32-Bit Microcontroller

SAM L22G / L22J / L22N

Introduction

The SAM L22 is a series of Ultra low-power segment LCD microcontrollers using the 32-bit ARM® Cortex®-M0+ processor, ranging from 48- to 100-pins with up to 256KB Flash and 32KB of SRAM and to drive up to 320 LCD segments. The SAM L22 devices operate at a maximum frequency of 32MHz and reach 2.46 CoreMark®/MHz. With sophisticated power management technologies the SAM L22 devices run down to 39µA/MHz (CPU running CoreMark) in active mode and down to 490nA in ultra low-power backup mode with RTC.

Features

- Processor
  - ARM Cortex-M0+ CPU running at up to 32MHz
    - Single-cycle hardware multiplier
    - Micro Trace Buffer
    - Memory Protection Unit (MPU)
- Memories
  - 64/128/256KB in-system self-programmable Flash
  - 2/4/8KB Flash Read-While-Write section
  - 8/16/32KB SRAM Main Memory
- System
  - Power-on reset (POR) and programmable brown-out detection (BOD)
  - Internal and external clock options
  - External Interrupt Controller (EIC)
    - 16 external interrupts that can use any I/O-Pin
    - One non-maskable interrupt on one I/O-Pin
  - Two-pin Serial Wire Debug (SWD)
- Low Power
  - Idle, Standby, Backup, and Off sleep modes
  - SleepWalking peripherals
  - Battery backup support
  - Two runtime selectable power/performance levels
  - Embedded Buck/LDO regulator supporting on-the-fly selection
  - Active mode: <50µA/MHz
  - Standby with full retention, RTC and LCD = 3.47µA
    - 2.1µs wake-up time

© 2017 Microchip Technology Inc.

  - Standby with full retention and RTC: 1.87µA
    - 2.1µs wake-up time
  - Ultra low power Backup mode with RTC: 490nA
    - 90µs wake-up time

- Peripherals
  - Segment LCD controller
    - Up to 8 (4) common and 40 (44) segment terminals to drive 320 (176) segments
    - Static, ½, 1/3, ¼ bias
    - Internal charge pump able to generate VLCD higher than VDDIO
  - 16-channel Direct Memory Access Controller (DMAC)
  - 8-channel Event System
  - Up to four 16-bit Timer/Counters (TC), each configurable as:
    - 16-bit TC with two compare/capture channels
    - 8-bit TC with two compare/capture channels
    - 32-bit TC with two compare/capture channels, by using two TCs
  - One 24-bit Timer/Counters for Control (TCC), with extended functions:
    - Four compare channels with optional complementary output
    - Generation of synchronized pulse width modulation (PWM) pattern across port pins
    - Deterministic fault protection, fast decay and configurable dead-time between complementary output
    - Dithering that increase resolution with up to 5 bit and reduce quantization error
  - Frequency Meter
  - 32-bit Real Time Counter (RTC) with clock/calendar function
    - 8x32-bit Backup Register
    - Tamper Detection
  - Watchdog Timer (WDT)
  - CRC-32 generator
  - One full-speed (12Mbps) Universal Serial Bus (USB) 2.0 Device
    - Eight endpoints
    - Crystal less operation
  - Up to six Serial Communication Interfaces (SERCOM), each configurable as:
    - USART with full-duplex and single-wire half-duplex configuration
    - ISO7816
    - I^2C up to 3.4MHz[1]
    - SPI
  - One AES encryption engine
  - One True Random Generator (TRNG)
  - One Configurable Custom Logic (CCL)
  - One 12-bit, 1MSPS Analog-to-Digital Converter (ADC) with up to 20 channels
    - Differential and single-ended input
    - Oversampling and decimation in hardware to support 13-, 14-, 15-, or 16-bit resolution
  - Two Analog Comparators (AC) with window compare function

[1] Max 1 high-speed mode and max 3 fast mode I^2C

  - Peripheral Touch Controller (PTC)
    - Up to 256-Channel capacitive touch sensing
      - Maximum Mutual-Cap up to 16x16 channels
      - Maximum Self-Cap up to 24 channels
    - Wake-up on touch in standby mode

- Oscillators
  - 32.768kHz crystal oscillator (XOSC32K)
  - 0.4-32MHz crystal oscillator (XOSC)
  - 32.768kHz ultra-low-power internal oscillator (OSCULP32K)
  - 16/12/8/4MHz high-accuracy internal oscillator (OSC16M)
  - 48MHz Digital Frequency Locked Loop (DFLL48M)
  - 96MHz Fractional Digital Phased Locked Loop (FDPLL96M)
- I/O
  - Up to 82 programmable I/O pins
  - Up to 52 segment LCD pins can be used as GPIO/GPI
  - Up to 5 wake-up pins with optional debouncing
  - Up to 5 tamper input pins
  - 1 tamper output pin
- Pin and code compatible with SAM D and SAM L Cortex-M0+ Families[2]
- Packages
  - 100-pin TQFP, UFBGA
  - 64-pin TQFP, QFN
  - 49-pin WLCSP
  - 48-pin TQFP, QFN
- Operating Voltage
  - 1.62V – 3.63V

[2] except the VLCD

Table of Contents

Introduction....1

Features 1

1. Description....14

2. Configuration Summary....15

3. Ordering Information....18
   - 3.1. SAM L22N....18
   - 3.2. SAM L22J....18
   - 3.3. SAM L22G....19
   - 3.4. Device Identification....19

4. Block Diagram....21

5. Pinout....23
   - 5.1. SAM L22G....23
   - 5.2. SAM L22J....25
   - 5.3. SAM L22N....26

6. Signal Descriptions List 28

7. I/O Multiplexing and Considerations....31
   - 7.1. Multiplexed Signals....31
   - 7.2. Other Functions....33

8. Power Supply and Start-Up Considerations....36
   - 8.1. Power Domain Overview....36
   - 8.2. Power Supply Considerations....36
   - 8.3. Power-Up....39
   - 8.4. Power-On Reset and Brown-Out Detector....39
   - 8.5. Performance Level Overview....40

9. Product Mapping....42

10. Memories....43
    - 10.1. Embedded Memories....43
    - 10.2. Physical Memory Map....43
    - 10.3. NVM User Row Mapping....44
    - 10.4. NVM Software Calibration Area Mapping....45
    - 10.5. Serial Number....45

11. Processor and Architecture....46
    - 11.1. Cortex M0+ Processor....46
    - 11.2. Nested Vector Interrupt Controller....48

11.3. Micro Trace Buffer....49
11.4. High-Speed Bus System....50

12. PAC - Peripheral Access Controller....54
12.1. Overview....54
12.2. Features....54
12.3. Block Diagram....54
12.4. Product Dependencies....54
12.5. Functional Description....55
12.6. Register Summary....59
12.7. Register Description....60

13. Peripherals Configuration Summary....71

14. DSU - Device Service Unit....74
14.1. Overview....74
14.2. Features....74
14.3. Block Diagram....75
14.4. Signal Description....75
14.5. Product Dependencies....75
14.6. Debug Operation....76
14.7. Chip Erase....78
14.8. Programming....78
14.9. Intellectual Property Protection....79
14.10. Device Identification....80
14.11. Functional Description....81
14.12. Register Summary....87
14.13. Register Description....89

15. Clock System....111
15.1. Clock Distribution....111
15.2. Synchronous and Asynchronous Clocks....112
15.3. Register Synchronization....113
15.4. Enabling a Peripheral....116
15.5. On Demand Clock Requests....116
15.6. Power Consumption vs. Speed....117
15.7. Clocks after Reset....117

16. GCLK - Generic Clock Controller....118
16.1. Overview....118
16.2. Features....118
16.3. Block Diagram....118
16.4. Signal Description....119
16.5. Product Dependencies....119
16.6. Functional Description....120
16.7. Register Summary....126
16.8. Register Description....129

17. MCLK – Main Clock....137

17.1. Overview ... 137
17.2. Features ... 137
17.3. Block Diagram ... 137
17.4. Signal Description ... 137
17.5. Product Dependencies ... 137
17.6. Functional Description ... 139
17.7. Register Summary - MCLK ... 144
17.8. Register Description ... 144

18. FREQM – Frequency Meter ... 156
18.1. Overview ... 156
18.2. Features ... 156
18.3. Block Diagram ... 156
18.4. Signal Description ... 156
18.5. Product Dependencies ... 156
18.6. Functional Description ... 158
18.7. Register Summary ... 161
18.8. Register Description ... 161

19. RSTC – Reset Controller ... 167
19.1. Overview ... 167
19.2. Features ... 167
19.3. Block Diagram ... 167
19.4. Signal Description ... 167
19.5. Product Dependencies ... 168
19.6. Functional Description ... 168
19.7. Register Summary ... 171
19.8. Register Description ... 171

20. PM – Power Manager ... 173
20.1. Overview ... 173
20.2. Features ... 173
20.3. Block Diagram ... 173
20.4. Signal Description ... 173
20.5. Product Dependencies ... 174
20.6. Functional Description ... 175
20.7. Register Summary ... 184
20.8. Register Description ... 184

21. OSCCTRL – Oscillators Controller ... 189
21.1. Overview ... 189
21.2. Features ... 189
21.3. Block Diagram ... 190
21.4. Signal Description ... 190
21.5. Product Dependencies ... 190
21.6. Functional Description ... 191
21.7. Register Summary ... 204
21.8. Register Description ... 205

22. OSC32KCTRL – 32KHz Oscillators Controller....231
22.1. Overview....231
22.2. Features....231
22.3. Block Diagram....231
22.4. Signal Description....231
22.5. Product Dependencies....231
22.6. Functional Description....233
22.7. Register Summary....238
22.8. Register Description....238
23. SUPC – Supply Controller....248
23.1. Overview....248
23.2. Features....248
23.3. Block Diagram....249
23.4. Signal Description....249
23.5. Product Dependencies....249
23.6. Functional Description....251
23.7. Register Summary....259
23.8. Register Description....260
24. WDT – Watchdog Timer....276
24.1. Overview....276
24.2. Features....276
24.3. Block Diagram....277
24.4. Signal Description....277
24.5. Product Dependencies....277
24.6. Functional Description....278
24.7. Register Summary....284
24.8. Register Description....284
25. RTC – Real-Time Counter....291
25.1. Overview....291
25.2. Features....291
25.3. Block Diagram....292
25.4. Signal Description....293
25.5. Product Dependencies....293
25.6. Functional Description....295
25.7. Register Summary - COUNT32....306
25.8. Register Description - COUNT32....308
25.9. Register Summary - COUNT16....325
25.10. Register Description - COUNT16....327
25.11. Register Summary - CLOCK....344
25.12. Register Description - CLOCK....346
26. DMAC – Direct Memory Access Controller....364
26.1. Overview....364
26.2. Features....364
26.3. Block Diagram....366

26.4. Signal Description ... 366
26.5. Product Dependencies ... 366
26.6. Functional Description ... 367
26.7. Register Summary ... 387
26.8. Register Description ... 388
26.9. Register Summary - SRAM ... 413
26.10. Register Description - SRAM ... 413

27. EIC – External Interrupt Controller ... 420
27.1. Overview ... 420
27.2. Features ... 420
27.3. Block Diagram ... 420
27.4. Signal Description ... 421
27.5. Product Dependencies ... 421
27.6. Functional Description ... 422
27.7. Register Summary ... 427
27.8. Register Description ... 427

28. NVMCTRL – Non-Volatile Memory Controller ... 437
28.1. Overview ... 437
28.2. Features ... 437
28.3. Block Diagram ... 437
28.4. Signal Description ... 438
28.5. Product Dependencies ... 438
28.6. Functional Description ... 439
28.7. Register Summary ... 446
28.8. Register Description ... 446

29. PORT - I/O Pin Controller ... 456
29.1. Overview ... 456
29.2. Features ... 456
29.3. Block Diagram ... 457
29.4. Signal Description ... 457
29.5. Product Dependencies ... 457
29.6. Functional Description ... 459
29.7. Register Summary ... 465
29.8. PORT Pin Groups and Register Repetition ... 467
29.9. Register Description ... 467

30. EVSYS – Event System ... 484
30.1. Overview ... 484
30.2. Features ... 484
30.3. Block Diagram ... 484
30.4. Signal Description ... 485
30.5. Product Dependencies ... 485
30.6. Functional Description ... 486
30.7. Register Summary ... 490
30.8. Register Description ... 491

31. SERCOM – Serial Communication Interface....505
31.1. Overview....505
31.2. Features....505
31.3. Block Diagram....506
31.4. Signal Description....506
31.5. Product Dependencies....506
31.6. Functional Description....508

32. SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter....514
32.1. Overview....514
32.2. USART Features....514
32.3. Block Diagram....515
32.4. Signal Description....515
32.5. Product Dependencies....515
32.6. Functional Description....517
32.7. Register Summary....531
32.8. Register Description....532

33. SERCOM SPI – SERCOM Serial Peripheral Interface....551
33.1. Overview....551
33.2. Features....551
33.3. Block Diagram....552
33.4. Signal Description....552
33.5. Product Dependencies....552
33.6. Functional Description....554
33.7. Register Summary....563
33.8. Register Description....564

34. SERCOM I^2C – SERCOM Inter-Integrated Circuit....577
34.1. Overview....577
34.2. Features....577
34.3. Block Diagram....578
34.4. Signal Description....578
34.5. Product Dependencies....578
34.6. Functional Description....580
34.7. Register Summary - I2C Slave....598
34.8. Register Description - I^2C Slave....598
34.9. Register Summary - I2C Master....612
34.10. Register Description - I^2C Master....613

35. TC – Timer/Counter....629
35.1. Overview....629
35.2. Features....629
35.3. Block Diagram....630
35.4. Signal Description....630
35.5. Product Dependencies....631
35.6. Functional Description....632

35.7. Register Description....647

36. TCC – Timer/Counter for Control Applications....698
36.1. Overview....698
36.2. Features....698
36.3. Block Diagram....699
36.4. Signal Description....699
36.5. Product Dependencies....700
36.6. Functional Description....701
36.7. Register Summary....735
36.8. Register Description....737

37. TRNG – True Random Number Generator....773
37.1. Overview....773
37.2. Features....773
37.3. Block Diagram....773
37.4. Signal Description....773
37.5. Product Dependencies....773
37.6. Functional Description....774
37.7. Register Summary....777
37.8. Register Description....777

38. AES – Advanced Encryption Standard....781
38.1. Overview....781
38.2. Features....781
38.3. Block Diagram....782
38.4. Signal Description....783
38.5. Product Dependencies....783
38.6. Functional Description....784
38.7. Register Summary....793
38.8. Register Description....795

39. USB – Universal Serial Bus....809
39.1. Overview....809
39.2. Features....809
39.3. USB Block Diagram....810
39.4. Signal Description....810
39.5. Product Dependencies....810
39.6. Functional Description....812
39.7. Register Summary....820
39.8. Register Description....822

40. CCL – Configurable Custom Logic....851
40.1. Overview....851
40.2. Features....851
40.3. Block Diagram....852
40.4. Signal Description....852
40.5. Product Dependencies....852
40.6. Functional Description....854

40.7. Register Summary....864
40.8. Register Description....864

41. ADC – Analog-to-Digital Converter....868
41.1. Overview....868
41.2. Features....868
41.3. Block Diagram....869
41.4. Signal Description....869
41.5. Product Dependencies....869
41.6. Functional Description....871
41.7. Register Summary....881
41.8. Register Description....882

42. AC – Analog Comparators....900
42.1. Overview....900
42.2. Features....900
42.3. Block Diagram....901
42.4. Signal Description....901
42.5. Product Dependencies....901
42.6. Functional Description....903
42.7. Register Summary....912
42.8. Register Description....912

43. SLCD - Segment Liquid Crystal Display Controller....924
43.1. Overview....924
43.2. Features....924
43.3. Block Diagram....925
43.4. Signal Description....925
43.5. Product Dependencies....925
43.6. Functional Description....927
43.7. Register Summary....952
43.8. Register Description....955

44. PTC - Peripheral Touch Controller....994
44.1. Overview....994
44.2. Features....994
44.3. Block Diagram....995
44.4. Signal Description....996
44.5. Product Dependencies....996
44.6. Functional Description....997

45. Electrical Characteristics....999
45.1. Disclaimer....999
45.2. Absolute Maximum Ratings....999
45.3. General Operating Ratings....999
45.4. Supply Characteristics....1000
45.5. Maximum Clock Frequencies....1001
45.6. Power Consumption....1002
45.7. Wake-up Timing....1006

45.8. IO Pin Characteristics....1007
45.9. Injection Current....1008
45.10. Analog Characteristics....1009
45.11. NVM Characteristics....1019
45.12. Oscillators Characteristics....1020
45.13. USB Characteristics....1026
45.14. SLCD Characteristics....1027
45.15. External Reset Pin....1030

46. Typical Characteristics....1031
46.1. Power Consumption over Temperature in Sleep Modes....1031

47. Packaging Information....1033
47.1. Thermal Considerations....1033
47.2. Package Drawings....1034
47.3. Soldering Profile....1042

48. Schematic Checklist....1043
48.1. Introduction....1043
48.2. Power Supply....1043
48.3. External Analog Reference Connections....1046
48.4. External Reset Circuit....1048
48.5. Unused or Unconnected Pins....1049
48.6. Clocks and Crystal Oscillators....1049
48.7. Programming and Debug Ports....1051
48.8. USB Interface....1055
48.9. LCD....1056
48.10. SERCOM I^2C Pins....1057
48.11. Pin Characteristics....1057
48.12. Reference Schematic....1057

49. Errata....1058
49.1. Die Revision A....1058
49.2. Die Revision B....1066

50. Conventions....1073
50.1. Numerical Notation....1073
50.2. Memory Size and Type....1073
50.3. Frequency and Time....1073
50.4. Registers and Bits....1074

51. Acronyms and Abbreviations....1075

52. Datasheet Revision History....1078
52.1. Rev.A - 03/2017....1078
52.2. Rev.E - 07/2016....1080
52.3. Rev.D - 05/2016....1080
52.4. Rev.C - 01/2016....1083
52.5. Rev.B - 11/2015....1085

52.6. Rev A - 08/2015..... 1087

The Microchip Web Site..... 1088

Customer Change Notification Service.....1088

Customer Support..... 1088

Product Identification System.....1089

Microchip Devices Code Protection Feature..... 1089

Legal Notice.....1090

Trademarks..... 1090

Quality Management System Certified by DNV.....1091

Worldwide Sales and Service.....1092

1. Description

The SAM L22 is a series of Ultra low-power segment LCD microcontrollers using the 32-bit ARM® Cortex®-M0+ processor, ranging from 48- to 100-pins with up to 256KB Flash and 32KB of SRAM and can drive up to 320 LCD segments. The SAM L22 devices operate at a maximum frequency of 32MHz and reach 2.46 Coremark/MHz. They are designed for simple and intuitive migration with identical peripheral modules, hex compatible code, identical linear address map and pin compatible migration paths between all devices in the product series. All devices include intelligent and flexible peripherals, an Event System for inter-peripheral signaling, and support for capacitive touch button, slider and wheel user interfaces.

The SAM L22 devices provide the following features: Segment LCD (SLCD) controller with up to 48 selectable SLCD pins from max. 52 pins to drive up to 320 segments, all SLCD Pins can be used also as GPIOs (100-pin package: 8 of the SLCD pins can be used only as GP input), in-system programmable Flash, sixteen-channel direct memory access (DMA) controller, 8 channel Event System, programmable interrupt controller, up to 82 programmable I/O pins, 32-bit real-time clock and calendar, up to four 16-bit Timer/Counters (TC) and one 24-bit Timer/Counters for Control (TCC), where each TC can be configured to perform frequency and waveform generation, accurate program execution timing or input capture with time and frequency measurement of digital signals. The TCs can operate in 8- or 16-bit mode, selected TCs can be cascaded to form a 32-bit TC, and the TCC has extended functions optimized for motor, lighting and other control applications. The series provide one full-speed USB 2.0 device interface; up to six Serial Communication Modules (SERCOM) that each can be configured to act as an USART, UART, SPI, I^2C up to 3.4MHz, SMBus, PMBus, and ISO7816 smart card interface; up to twenty channel 1Msps 12-bit ADC with optional oversampling and decimation supporting up to 16-bit resolution, two analog comparators with window mode, Peripheral Touch Controller supporting up to 256 buttons, sliders, wheels and proximity sensing; programmable Watchdog Timer, brown-out detector and power-on reset and two-pin Serial Wire Debug (SWD) program and debug interface.

All devices have accurate and low-power external and internal oscillators. All oscillators can be used as a source for the system clock. Different clock domains can be independently configured to run at different frequencies, enabling power saving by running each peripheral at its optimal clock frequency, and thus maintaining a high CPU frequency while reducing power consumption.

The SAM L22 devices have four software-selectable sleep modes, idle, standby, backup and off. In idle mode the CPU is stopped while all other functions can be kept running. In standby all clocks and functions are stopped expect those selected to continue running. In this mode all RAMs and logic contents are retained. The device supports SleepWalking. This feature allows the peripheral to wake up from sleep based on predefined conditions, and thus allows some internal operation like DMA transfer and/or the CPU to wake up only when needed, e.g. when a threshold is crossed or a result is ready. The Event System supports synchronous and asynchronous events, allowing peripherals to receive, react to and send events even in standby mode.

The SAM L22 devices have two software-selectable performance level (PL0 and PL2) allowing the user to scale the lowest core voltage level that will support the operating frequency.

The Flash program memory can be reprogrammed in-system through the SWD interface. The same interface can be used for nonintrusive on-chip debugging of application code. A boot loader running in the device can use any communication interface to download and upgrade the application program in the Flash memory.

The SAM L22 devices are supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators, programmers and evaluation kits.

2. Configuration Summary

	SAM L22N	SAM L22J	SAM L22G
Pins	100	64	48 (QFN and TQFP) 49 (WLCSP)
General Purpose I/O-pins (GPIOs)(1)	82	50	36
Flash	256/128/64KB	256/128/64KB	256/128/64KB
Flash RWW section	8/4/2KB	8/4/2KB	8/4/2KB
System SRAM	32/16/8KB	32/16/8KB	32/16/8KB
Segment LCD (SLCD) Pins(1)	48 selectable from 52	31	23
Timer Counter (TC) instances	4	4	4
Waveform output channels per TC instance	2	2	2
Timer Counter for Control (TCC) instances	1	1	1
Waveform output channels per TCC	4	4	4
DMA channels	16	16	16
USB interface	1	1	1
AES engine	1	1	1
Configurable Custom Logic (CCL) (LUTs)	4	4	4
True Random Generator (TRNG)	1	1	1
Serial Communication Interface (SERCOM) instances	6	4(2)	4(2)
Analog-to-Digital Converter (ADC) channels	20	16	10
Two Analog Comparators (AC) with number of external input channels	4	4	2
Tamper Input Pins	5	3	2

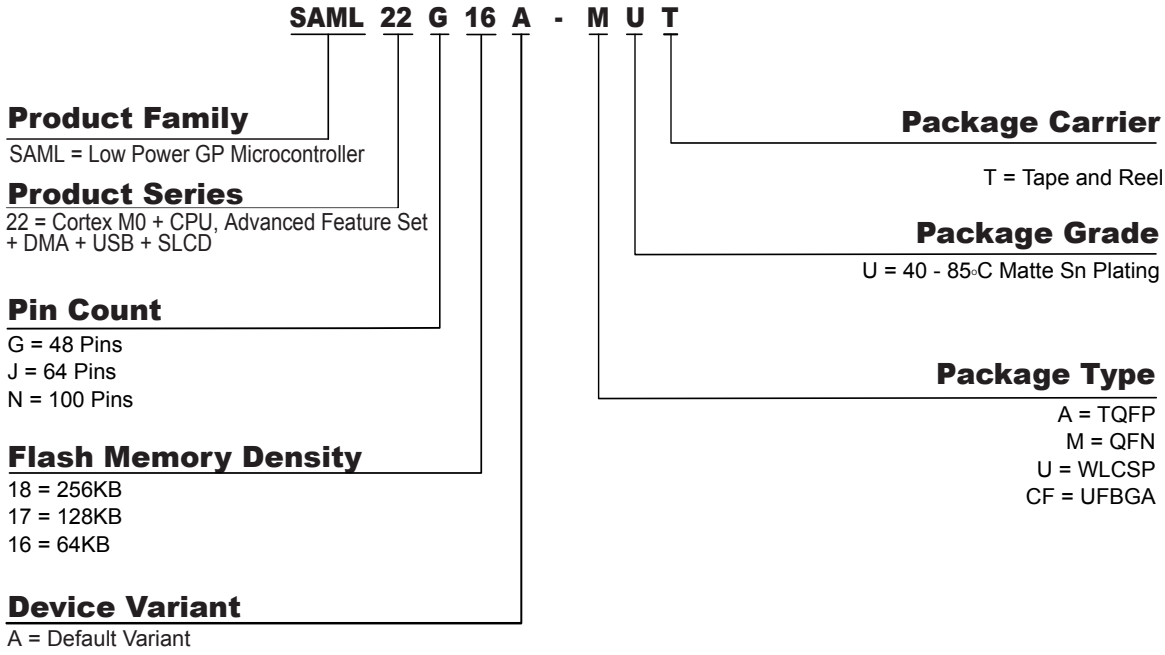
	SAM L22N	SAM L22J	SAM L22G
Wake-up Pins with debouncing	5	3	2
Real-Time Counter (RTC)	Yes	Yes	Yes
RTC alarms	1	1	1
RTC compare values	One 32-bit value or two 16-bit values	One 32-bit value or two 16-bit values	One 32-bit value or two 16-bit values
External Interrupt lines	16	16	16
Peripheral Touch Controller (PTC) channels (X- x Y-lines) for mutual capacitance(3)	256 (16x16)	182 (13x14)	132 (11x12)
Peripheral Touch Controller (PTC) channels for self capacitance (Y-lines only)(4)	24	19	15
Maximum CPU frequency	32MHz	32MHz	32MHz
Packages	TQFP UFBGA	QFN TQFP	QFN TQFP WLCSP
Oscillators	32.768kHz crystal oscillator (XOSC32K) 0.4-32MHz crystal oscillator (XOSC) 32KHz ultra-low-power internal oscillator (OSCULP32K) 16/12/8/4MHz high-accuracy internal oscillator (OSC16M) 48MHz Digital Frequency Locked Loop (DFLL48M) 96MHz Fractional Digital Phased Locked Loop (FDPLL96M)		
Event System channels	8	8	8
SW Debug Interface	Yes	Yes	Yes
Watchdog Timer (WDT)	Yes	Yes	Yes

Note:

1. L22J, L22G: All SLCD Pins can be configured also as GPIOs. L22N: 44 SLCD Pins can be configured as GPIOs, 8 SLCD Pins can be used as GP input.
2. SAM L22N: SERCOM[5:0]. L22G, L22J: SERCOM[3:0].
3. The number of X- and Y-lines depends on the configuration of the device, as some I/O lines can be configured as either X-lines or Y-lines.

4. The number of Y-lines depends on the configuration of the device, as some I/O lines can be configured as either X-lines or Y-lines. The number given here is the maximum number of Y-lines that can be obtained.

3. Ordering Information



Note: The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

3.1 SAM L22N

Table 3-1. SAM L22N Ordering Codes

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML22N16A-AUT	64K	8K	TQFP100	Tape & Reel
ATSAML22N16A-CFUT			UFBGA100	
ATSAML22N17A-AUT	128K	16K	TQFP100	Tape & Reel
ATSAML22N17A-CFUT			UFBGA100	
ATSAML22N18A-AUT	256K	32K	TQFP100	Tape & Reel
ATSAML22N18A-CFUT			UFBGA100	

3.2 SAM L22J

Table 3-2. SAM L22J Ordering Codes

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML22J16A-AUT	64K	8K	TQFP64	Tape & Reel
ATSAML22J16A-MUT			QFN64	

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML22J17A-AUT	128K	16K	TQFP64	Tape & Reel
ATSAML22J17A-MUT			QFN64	
ATSAML22J18A-AUT	256K	32K	TQFP64	Tape & Reel
ATSAML22J18A-MUT			QFN64	

3.3 SAM L22G

Table 3-3. SAM L22G Ordering Codes

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML22G16A-AUT	64K	8K	TQFP48	Tape & Reel
ATSAML22G16A-MUT			QFN48	
ATSAML22G17A-AUT	128K	16K	TQFP48	Tape & Reel
ATSAML22G17A-MUT			QFN48	
ATSAML22G17A-UUT			WLCSP49	
ATSAML22G18A-AUT	256K	32K	TQFP48	Tape & Reel
ATSAML22G18A-MUT			QFN48	
ATSAML22G18A-UUT			WLCSP49	

3.4 Device Identification

The DSU - Device Service Unit peripheral provides the Device Selection bits in the Device Identification register (DID.DEVSEL) in order to identify the device by software. The SAM L22 variants have a reset value of DID=0x10820xxx, with the last digits identifying the variant:

Table 3-4. SAM L22 Device Identification Values

DSU DID.DEVSEL	Device
0x0	L22N18
0x1	L22N17
0x2	L22N16
0x3-0x4	Reserved
0x5	L22J18
0x6	L22J17
0x7	L22J16
0x8-0x9	Reserved
0xA	L22G18
0xB	L22G17

DSU DID.DEVSEL	Device
0xC	L22G16
0xD-0xFF	Reserved

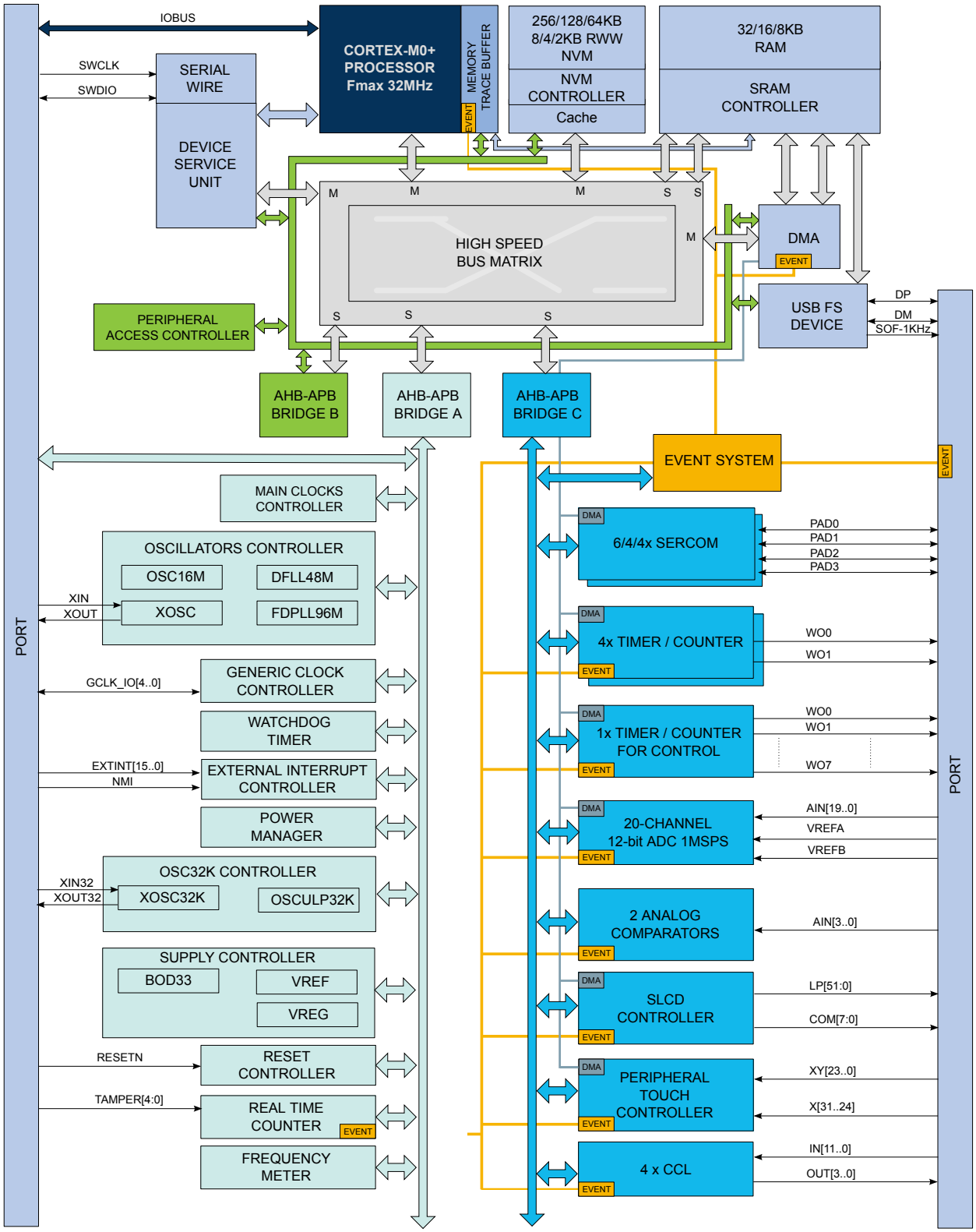
Note: The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

Related Links

DSU - Device Service Unit

DID

4. Block Diagram



Note:

1. Some device configurations have different number of SERCOM instances, Timer/Counter instances, PTC signals and ADC signals. The number of PTC X and Y signals is configurable.

Related Links

Peripherals Configuration Summary

5. Pinout

5.1 SAM L22G

Figure 5-1. 48-Pin QFN, TQFP

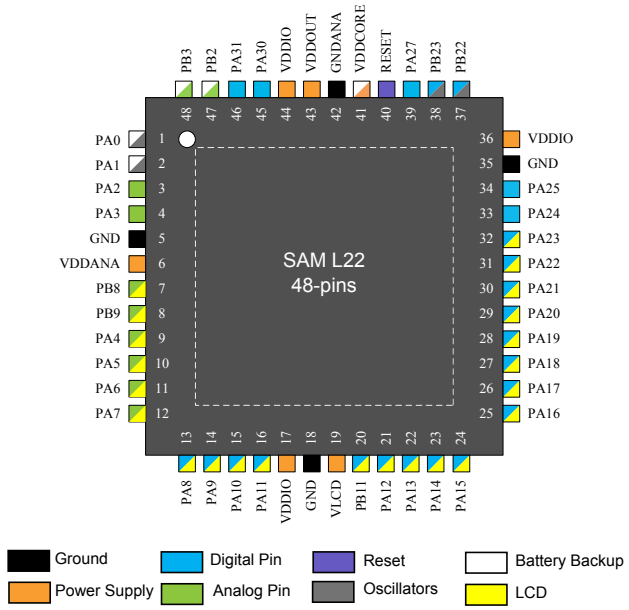
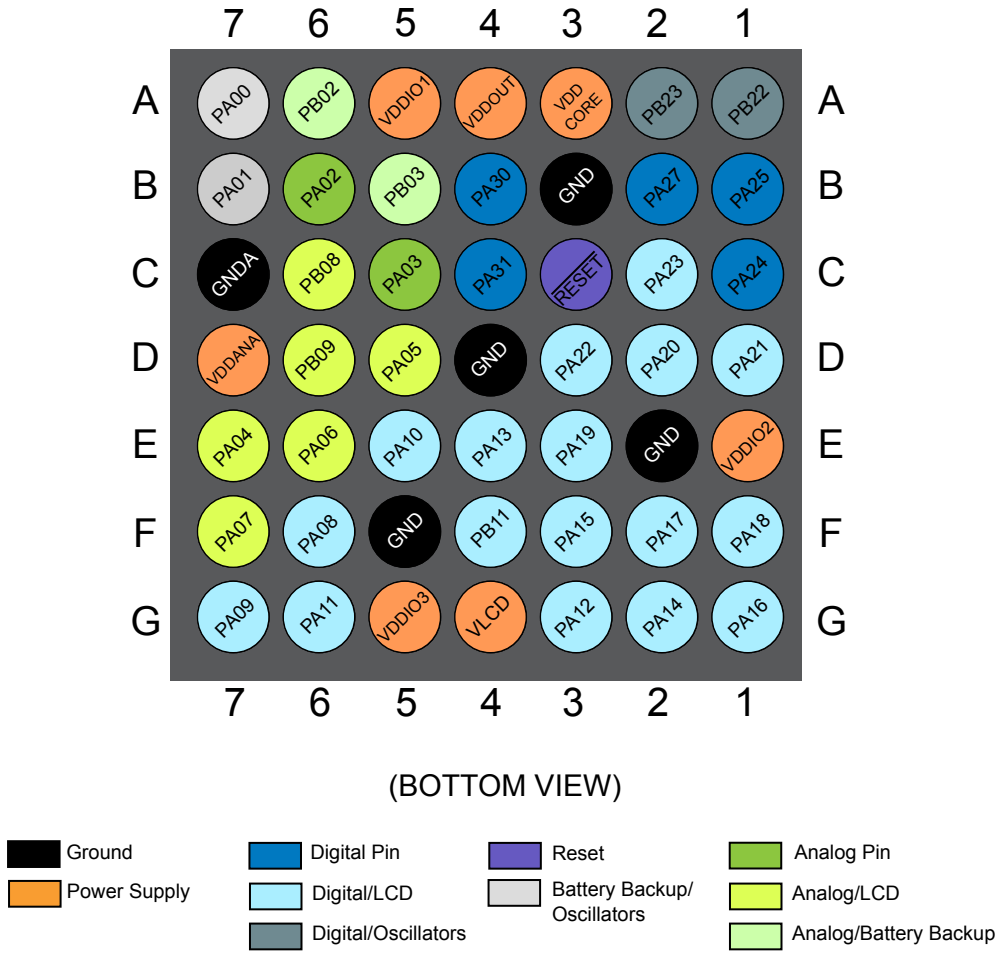
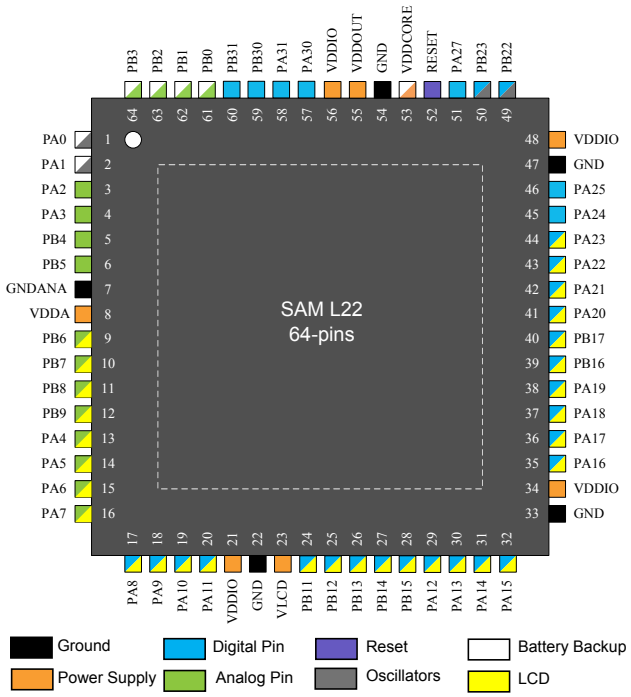


Figure 5-2. 49-Pin WLCSP



5.2 SAM L22J


PB3 64
PB2 63
PB1 62
PB0 61
PB31 60
PB30 59
PA31 58
PA30 57
VDDIO 56
VDDOUT 55
GND 54
VDDCORE 53
RESET 52
PA27 51
PB23 50
PB22 49
PA0 1
PA1 2
PA2 3
PA3 4
PB4 5
PB5 6
GNDANA 7
VDDA 8
PB6 9
PB7 10
PB8 11
PB9 12
PA4 13
PA5 14
PA6 15
PA7 16
48 VDDIO
47 GND
46 PA25
45 PA24
44 PA23
43 PA22
42 PA21
41 PA20
40 PB17
39 PB16
38 PA19
37 PA18
36 PA17
35 PA16
34 VDDIO
33 GND
SAM L22
64-pins
17 PA8
18 PA9
19 PA10
20 PA11
21 VDDIO
22 GND
23 VLCD
24 PB11
25 PB12
26 PB13
27 PB14
28 PB15
29 PA12
30 PA13
31 PA14
32 PA15
Ground
Power Supply
Digital Pin
Analog Pin
Reset
Oscillators
Battery Backup
LCD

5.3 SAM L22N

Figure 5-3. TQFP100

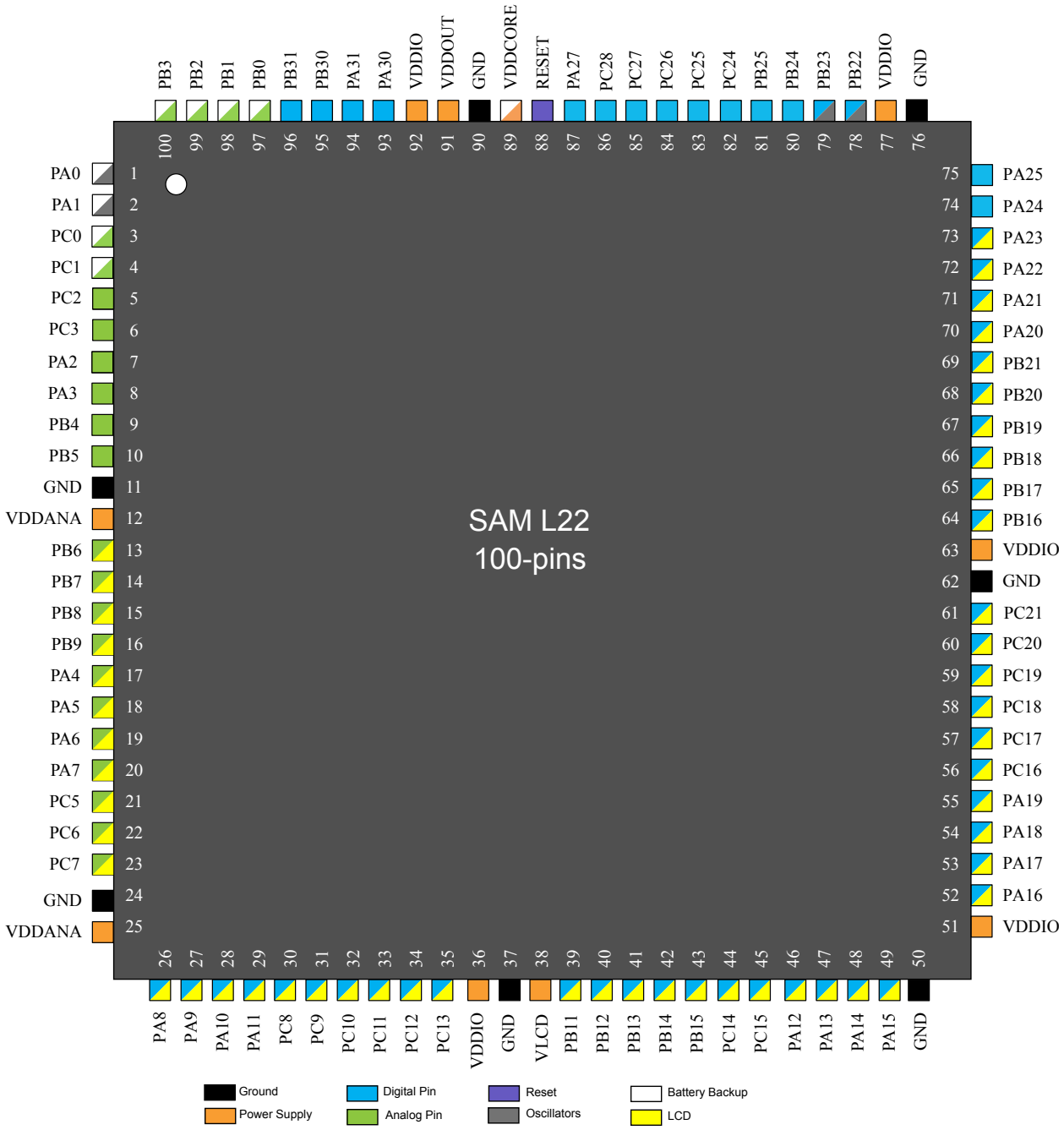
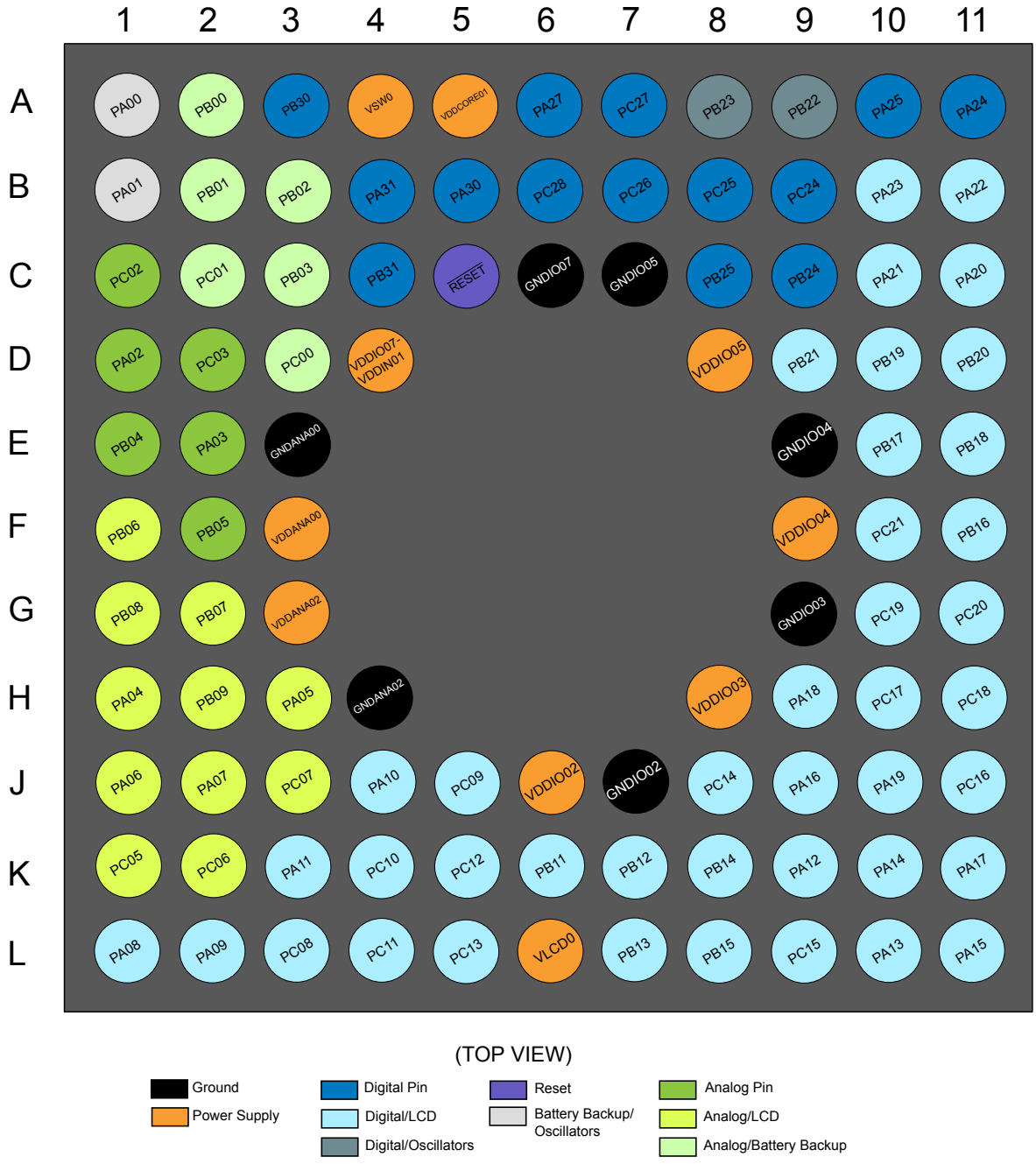


Figure 5-4. BGA100



(TOP VIEW)

Ground
Power Supply
Digital Pin
Digital/LCD
Digital/Oscillators
Reset
Battery Backup/ Oscillators
Analog Pin
Analog/LCD
Analog/Battery Backup

6. Signal Descriptions List

The following table gives details on signal names classified by peripheral.

Table 6-1. Signal Descriptions List

Signal Name	Function	Type	Active Level
Analog Comparators - AC			
AIN[3:0]	AC Analog Inputs	Analog	
CMP[1:0]	AC Analog Output	Analog	
Analog Digital Converter - ADC			
AIN[19:0]	ADC Analog Inputs	Analog	
VREFA	ADC Voltage External Reference A	Analog	
VREFB	ADC Voltage External Reference B	Analog	
External Interrupt Controller - EIC			
EXTINT[15:0]	External Interrupts inputs	Digital	
NMI	External Non-Maskable Interrupt input	Digital	
Generic Clock Generator - GCLK			
GCLK_IO[4:0]	Generic Clock (source clock inputs or generic clock generator output)	Digital	
Custom Control Logic - CCL			
IN[11:0]	Logic Inputs	Digital	
OUT[3:0]	Logic Outputs	Digital	
Supply Controller - SUPC			
VBAT	External battery supply Inputs	Analog	
PSOK	Main Power Supply OK input	Digital	
OUT[1:0]	Logic Outputs	Digital	
Power Manager - PM			
RESETN	Reset input	Digital	Low
Serial Communication Interface - SERCOMx			
PAD[3:0]	SERCOM Inputs/Outputs Pads	Digital	
Oscillators Control - OSCCTRL			
XIN	Crystal or external clock Input	Analog/Digital	
XOUT	Crystal Output	Analog	

Signal Name	Function	Type	Active Level
32KHz Oscillators Control - OSC32KCTRL			
XIN32	32KHz Crystal or external clock Input	Analog/Digital	
XOUT32	32KHz Crystal Output	Analog	
Timer Counter - TCx			
WO[1:0]	Waveform Outputs	Digital	
Timer Counter - TCCx			
WO[7:0]	Waveform Outputs	Digital	
Peripheral Touch Controller - PTC			
X[7:0]	PTC Input/Output	Analog	
Y[23:0]	PTC Input/Output	Analog	
X[31:24]	PTC Output	Analog	
General Purpose I/O - PORT			
PA25 - PA00	Parallel I/O Controller I/O Port A	Digital	
PA27	Parallel I/O Controller I/O Port A	Digital	
PA31 - PA30	Parallel I/O Controller I/O Port A	Digital	
PB09 - PB00	Parallel I/O Controller I/O Port B	Digital	
PB25 - PB11	Parallel I/O Controller I/O Port B	Digital	
PB31 - PB30	Parallel I/O Controller I/O Port B	Digital	
PC03 - PC00	Parallel I/O Controller I/O Port C	Digital	
PC07 - PC05	Parallel I/O Controller I/O Port C	Digital	
PC17 - PC12	Parallel I/O Controller I/O Port C	Digital	
PC28 - PC24	Parallel I/O Controller I/O Port C	Digital	
General Purpose input - PORT			
PC11 - PC08	Parallel I/O Controller input Port C	Digital	
PC21 - PC18	Parallel I/O Controller input Port C	Digital	
Segment LCD			
SLCD51 - SLCD00	Segment LCD	Analog	
VLCD	Bias Voltage	Analog	
Universal Serial Bus - USB			
DP	DP for USB	Digital	

Signal Name	Function	Type	Active Level
DM	DM for USB	Digital	
SOF 1kHz	USB Start of Frame	Digital	
Real Timer Clock - RTC			
RTC_IN[4:0]	Tamper or external wake-up pins	Digital	
RTC_OUT	Tamper output	Digital	

7. I/O Multiplexing and Considerations

7.1 Multiplexed Signals

Each pin is by default controlled by the PORT as a general purpose I/O and alternatively it can be assigned a different peripheral functions. To enable a peripheral function on a pin, the Peripheral Multiplexer Enable bit in the Pin Configuration register corresponding to that pin (PINCFGn.PMUXEN, n = 0-31) in the PORT must be written to '1'. The selection of peripheral function A to I is done by writing to the Peripheral Multiplexing Odd and Even bits in the Peripheral Multiplexing register (PMUXn.PMUXE/O) of the PORT.

This table describes the peripheral signals multiplexed to the PORT I/O pins.

Table 7-1. PORT Function Multiplexing

Function	-				A	B					C	D	E		F	H	I
Type	L22G(5)	L22J	L22N	Pad Name	EIC	ANAREF	ADC	AC	PTC	SLCD	SERCOM(6)	SERCOM(6)	TC/TCC	TCC/RTC	COM/RTC	AC/ GCLK/ SUPC	CCL
Battery backup	1	1	1	PA00	EIC/EXTINT[0]							SERCOM1/ PAD[0]					
	2	2	2	PA01	EIC/EXTINT[1]							SERCOM1/ PAD[1]					
			3	PC00	EIC/EXTINT[8]		ADC/ AIN[16]								RTC/IN[3]		
			4	PC01	EIC/EXTINT[9]		ADC/ AIN[17]								RTC/IN[4]		
			5	PC02	EIC/EXTINT[10]		ADC/ AIN[18]		PTC/ XY[6]								
			6	PC03	EIC/EXTINT[11]		ADC/ AIN[19]		PTC/ XY[7]								
	3	3	7	PA02	EIC/EXTINT[2]	ADC/ VREFB	ADC/ AIN[0]	AC/ AIN[0]	PTC/ XY[8]						RTC/IN[2]		
	4	4	8	PA03	EIC/EXTINT[3]	ADC/ VREFA	ADC/ AIN[1]	AC/ AIN[1]	PTC/ XY[9]								
		5	9	PB04	EIC/EXTINT[4]		ADC/ AIN[12]	AC/ AIN[2]	PTC/ XY[10]								
		6	10	PB05	EIC/EXTINT[5]		ADC/ AIN[13]	AC/ AIN[3]	PTC/ XY[11]								
		9	13	PB06	EIC/EXTINT[6]		ADC/ AIN[14]		PTC/ XY[12]	SLCD/ LP[0]							CCL/IN[6]
		10	14	PB07	EIC/EXTINT[7]		ADC/ AIN[15]		PTC/ XY[13]	SLCD/ LP[1]							CCL/IN[7]
	7	11	15	PB08	EIC/EXTINT[8]		ADC/ AIN[2]		PTC/ XY[14]	SLCD/ LP[2]		SERCOM3/ PAD[0]	TC/0/ WO[0]				CCL/IN[8]
	8	12	16	PB09	EIC/EXTINT[9]		ADC/ AIN[3]		PTC/ XY[15]	SLCD/ LP[3]		SERCOM3/ PAD[1]	TC/0/ WO[1]				CCL/ OUT[2]
	9	13	17	PA04	EIC/EXTINT[4]		ADC/ AIN[4]		PTC/ X[24]	SLCD/ LP[4]		SERCOM0/ PAD[0]	TCC/ WO[0]				CCL/IN[0]
	10	14	18	PA05	EIC/EXTINT[5]		ADC/ AIN[5]		PTC/ X[25]	SLCD/ LP[5]		SERCOM0/ PAD[1]	TCC/ WO[1]				CCL/IN[1]
	11	15	19	PA06	EIC/EXTINT[6]		ADC/ AIN[6]		PTC/ X[26]	SLCD/ LP[6]		SERCOM0/ PAD[2]					CCL/IN[2]
	12	16	20	PA07	EIC/EXTINT[7]		ADC/ AIN[7]		PTC/ X[27]	SLCD/ LP[7]		SERCOM0/ PAD[3]					CCL/ OUT[0]
			21	PC05	EIC/EXTINT[13]				PTC/ XY[4]	SLCD/ LP[8]							
			22	PC06	EIC/EXTINT[14]				PTC/ XY[5]	SLCD/ LP[9]							
			23	PC07	EIC/EXTINT[15]					SLCD/ LP[10]							
	13	17	26	PA08	EIC/NMI				PTC/ XY[3]	SLCD/ LP[11]	SERCOM0/ PAD[0]	SERCOM4/ PAD[0]	TCC/ WO[0]				CCL/IN[3]
	14	18	27	PA09	EIC/EXTINT[9]				PTC/ XY[2]	SLCD/ LP[12]	SERCOM0/ PAD[1]	SERCOM4/ PAD[1]	TCC/ WO[1]				CCL/IN[4]
	15	19	28	PA10	EIC/EXTINT[10]				PTC/ XY[1]	SLCD/ LP[13]	SERCOM0/ PAD[2]	SERCOM4/ PAD[2]		TCC/ WO[2]		GCLK/ IO[4]	CCL/IN[5]
	16	20	29	PA11	EIC/EXTINT[11]				PTC/ XY[0]	SLCD/ LP[14]	SERCOM0/ PAD[3]	SERCOM4/ PAD[3]		TCC/ WO[3]			CCL/ OUT[1]
digital: input only			30	PC08	EIC/EXTINT[0]					SLCD/ LP[15]							
			31	PC09	EIC/EXTINT[1]					SLCD/ LP[16]							
			32	PC10	EIC/EXTINT[2]					SLCD/ LP[17]	SERCOM1/ PAD[2]						

Function	-				A	B					C	D	E		F	H	I
Type	L22G(5)	L22J	L22N	Pad Name	EIC	ANAREF	ADC	AC	PTC	SLCD	SERCOM(6)	SERCOM(6)	TC/TCC	TCC/RTC	COM/RTC	AC/ GCLK/ SUPC	CCL
			33	PC11	EIC/EXTINT[3]					SLCD/ LP[18]	SERCOM1/ PAD[3]						
			34	PC12	EIC/EXTINT[4]					SLCD/ LP[19]	SERCOM1/ PAD[0]						
			35	PC13	EIC/EXTINT[5]					SLCD/ LP[20]	SERCOM1/ PAD[1]						
	19	23	38	VLCD													
	20	24	39	PB11	EIC/EXTINT[11]					SLCD/ LP[21]		SERCOM3/ PAD[3]	TC/1/ WO[1]	TCC/ WO[5]			CCL/ OUT[1]
I2C: full Fm+. Limited currents for Sm, Fm		25	40	PB12	EIC/EXTINT[12]					SLCD/ LP[22]	SERCOM3/ PAD[0]		TC/0/ WO[0]	TCC/ WO[6]			
		26	41	PB13	EIC/EXTINT[13]					SLCD/ LP[23]	SERCOM3/ PAD[1]		TC/0/ WO[1]	TCC/ WO[7]			
		27	42	PB14	EIC/EXTINT[14]					SLCD/ LP[24]	SERCOM3/ PAD[2]		TC/1/ WO[0]			GCLK/ IO[0]	CCL/IN[9]
		28	43	PB15	EIC/EXTINT[15]					SLCD/ LP[25]	SERCOM3/ PAD[3]		TC/1/ WO[1]			GCLK/ IO[1]	CCL/ IN[10]
			44	PC14	EIC/EXTINT[6]					SLCD/ LP[26]							
			45	PC15	EIC/EXTINT[7]					SLCD/ LP[27]							
I2C: Sm, Fm, Fm+	21	29	46	PA12	EIC/EXTINT[12]					SLCD/ LP[28]	SERCOM4/ PAD[0]	SERCOM3/ PAD[0]		TCC/ WO[6]		AC/ CMP[0]	
	22	30	47	PA13	EIC/EXTINT[13]					SLCD/ LP[29]	SERCOM4/ PAD[1]	SERCOM3/ PAD[1]		TCC/ WO[7]		AC/ CMP[1]	
	23	31	48	PA14	EIC/EXTINT[14]					SLCD/ LP[30]	SERCOM4/ PAD[2]	SERCOM3/ PAD[2]		TCC/ WO[4]		GCLK/ IO[0]	
	24	32	49	PA15	EIC/EXTINT[15]					SLCD/ LP[31]	SERCOM4/ PAD[3]	SERCOM3/ PAD[3]		TCC/ WO[5]		GCLK/ IO[1]	
	25	35	52	PA16	EIC/EXTINT[0]				PTC/ X[28]	SLCD/ LP[32]	SERCOM1/ PAD[0]	SERCOM2/ PAD[0]		TCC/ WO[6]		GCLK/ IO[2]	CCL/IN[0]
	26	36	53	PA17	EIC/EXTINT[1]				PTC/ X[29]	SLCD/ LP[33]	SERCOM1/ PAD[1]	SERCOM2/ PAD[1]		TCC/ WO[7]		GCLK/ IO[3]	CCL/IN[1]
	27	37	54	PA18	EIC/EXTINT[2]				PTC/ X[30]	SLCD/ LP[34]	SERCOM1/ PAD[2]	SERCOM2/ PAD[2]		TCC/ WO[2]		AC/ CMP[0]	CCL/IN[2]
	28	38	55	PA19	EIC/EXTINT[3]				PTC/ X[31]	SLCD/ LP[35]	SERCOM1/ PAD[3]	SERCOM2/ PAD[3]		TCC/ WO[3]		AC/ CMP[1]	CCL/ OUT[0]
			56	PC16	EIC/EXTINT[8]					SLCD/ LP[36]							
			57	PC17	EIC/EXTINT[9]					SLCD/ LP[37]							
digital: input only			58	PC18	EIC/EXTINT[10]					SLCD/ LP[38]							
			59	PC19	EIC/EXTINT[11]					SLCD/ LP[39]							
			60	PC20	EIC/EXTINT[12]					SLCD/ LP[40]							CCL/IN[9]
			61	PC21	EIC/EXTINT[13]					SLCD/ LP[41]							CCL/ IN[10]
		39	64	PB16	EIC/EXTINT[0]					SLCD/ LP[42]	SERCOM5/ PAD[0]		TC/2/ WO[0]	TCC/ WO[4]		GCLK/ IO[2]	CCL/IN[11]
		40	65	PB17	EIC/EXTINT[1]					SLCD/ LP[43]	SERCOM5/ PAD[1]		TC/2/ WO[1]	TCC/ WO[5]		GCLK/ IO[3]	CCL/ OUT[3]
			66	PB18	EIC/EXTINT[2]					SLCD/ LP[44]	SERCOM5/ PAD[2]	SERCOM3/ PAD[2]		TCC/ WO[0]			
			67	PB19	EIC/EXTINT[3]					SLCD/ LP[45]	SERCOM5/ PAD[3]	SERCOM3/ PAD[3]		TCC/ WO[1]			
			68	PB20	EIC/EXTINT[4]					SLCD/ LP[46]	SERCOM3/ PAD[0]	SERCOM5/ PAD[0]		TCC/ WO[2]			
			69	PB21	EIC/EXTINT[5]					SLCD/ LP[47]	SERCOM3/ PAD[1]	SERCOM5/ PAD[1]		TCC/ WO[3]			
	29	41	70	PA20	EIC/EXTINT[4]				PTC/ XY[16]	SLCD/ LP[48]	SERCOM0/ PAD[0]	SERCOM2/ PAD[2]	TC/3/ WO[0]	TCC/ WO[6]		GCLK/ IO[4]	
	30	42	71	PA21	EIC/EXTINT[5]				PTC/ XY[17]	SLCD/ LP[49]	SERCOM0/ PAD[1]	SERCOM2/ PAD[3]	TC/3/ WO[1]	TCC/ WO[7]			
I2C: Sm, Fm, Fm+	31	43	72	PA22	EIC/EXTINT[6]				PTC/ XY[18]	SLCD/ LP[50]	SERCOM0/ PAD[2]	SERCOM2/ PAD[0]	TC/0/ WO[0]	TCC/ WO[4]			CCL/IN[6]
	32	44	73	PA23	EIC/EXTINT[7]				PTC/ XY[19]	SLCD/ LP[51]	SERCOM0/ PAD[3]	SERCOM2/ PAD[1]	TC/0/ WO[1]	TCC/ WO[5]	USB/SOF_1KHZ		CCL/IN[7]
	33	45	74	PA24	EIC/EXTINT[12]						SERCOM2/ PAD[2]	SERCOM5/ PAD[0]	TC/1/ WO[0]	TCC/ WO[0]	USB/DM		CCL/IN[8]
	34	46	75	PA25	EIC/EXTINT[13]						SERCOM2/ PAD[3]	SERCOM5/ PAD[1]	TC/1/ WO[1]	TCC/ WO[1]	USB/DP		CCL/ OUT[2]
	37	49	78	PB22	EIC/EXTINT[6]						SERCOM0/ PAD[2]	SERCOM5/ PAD[2]	TC/3/ WO[0]	TCC/ WO[2]	USB/SOF_1KHZ	GCLK/ IO[0]	CCL/IN[0]
	38	50	79	PB23	EIC/EXTINT[7]						SERCOM0/ PAD[3]	SERCOM5/ PAD[3]	TC/3/ WO[1]	TCC/ WO[3]		GCLK/ IO[1]	CCL/ OUT[0]
			80	PB24	EIC/EXTINT[8]						SERCOM0/ PAD[0]	SERCOM4/ PAD[0]		TCC/ WO[6]		AC/ CMP[0]	
			81	PB25	EIC/EXTINT[9]						SERCOM0/ PAD[1]	SERCOM4/ PAD[1]		TCC/ WO[7]		AC/ CMP[1]	

Function	-				A	B					C	D	E		F	H	I
Type	L22G(5)	L22J	L22N	Pad Name	EIC	ANAREF	ADC	AC	PTC	SLCD	SERCOM(6)	SERCOM(6)	TC/TCC	TCC/RTC	COM/RTC	AC/ GCLK/ SUPC	CCL
			82	PC24	EIC/EXTINT[0]						SERCOM0/ PAD[2]	SERCOM4/ PAD[2]	TC/2/ WO[0]	TCC/ WO[0]			
			83	PC25	EIC/EXTINT[1]						SERCOM0/ PAD[3]	SERCOM4/ PAD[3]	TC/2/ WO[1]	TCC/ WO[1]			
			84	PC26	EIC/EXTINT[2]								TC/3/ WO[0]	TCC/ WO[2]			
			85	PC27	EIC/EXTINT[3]							SERCOM1/ PAD[0]	TC/3/ WO[1]	TCC/ WO[3]			CCL/IN[4]
			86	PC28	EIC/EXTINT[4]				PTC/ XY[20]			SERCOM1/ PAD[1]		TCC/ WO[4]			CCL/IN[5]
recommended for GCLK IO	39	51	87	PA27	EIC/EXTINT[15]				PTC/ XY[21]					TCC/ WO[5]		GCLK/ IO[0]	
	40	52	88	RESET_N													
	45	57	93	PA30	EIC/EXTINT[10]				PTC/ XY[22]			SERCOM1/ PAD[2]			CORTEX_M0P/ SWCLK	GCLK/ IO[0]	CCL/IN[3]
	46	58	94	PA31	EIC/EXTINT[11]				PTC/ XY[23]			SERCOM1/ PAD[3]			SWDIO		CCL/ OUT[1]
I2C: Sm, Fm, Fm +, Hs		59	95	PB30	EIC/EXTINT[14]						SERCOM1/ PAD[0]	SERCOM5/ PAD[0]	TCC/ WO[0]				
		60	96	PB31	EIC/EXTINT[15]						SERCOM1/ PAD[1]	SERCOM5/ PAD[1]	TCC/ WO[1]				
Battery backup		61	97	PB00	EIC/EXTINT[0]		ADC/ AIN[8]				SERCOM3/ PAD[2]	SERCOM5/ PAD[2]	TC/3/ WO[0]		RTC/IN[0]	SUPC/ PSOK	CCL/IN[1]
		62	98	PB01	EIC/EXTINT[1]		ADC/ AIN[9]				SERCOM3/ PAD[3]	SERCOM5/ PAD[3]	TC/3/ WO[1]	RTC/IN[2]	RTC/OUT	SUPC/ OUT[0]	CCL/IN[2]
	47	63	99	PB02	EIC/EXTINT[2]		ADC/ AIN[10]				SERCOM3/ PAD[0]	SERCOM5/ PAD[0]	TC/2/ WO[0]		RTC/IN[1]	SUPC/ OUT[1]	CCL/ OUT[0]
	48	64	100	PB03	EIC/EXTINT[3]		ADC/ AIN[11]				SERCOM3/ PAD[1]	SERCOM5/ PAD[1]	TC/2/ WO[1]			SUPC/ VBAT	

Note:

1. All analog pin functions are on peripheral function B. Peripheral function B must be selected to disable the digital control of the pin.
2. Only some pins can be used in SERCOM I^2C mode. See the Type column for supported I^2C modes.
   - Sm: Standard mode, up to 100kHz
   - Fm: Fast mode, up to 400kHz
   - Fm+: Fast mode Plus, up to 1MHz
   - Hs: High-speed mode, up to 3.4MHz
3. These pins are High Sink pins and have different properties than regular pins: PA12, PA13, PA22, PA23, PA27, PA31, PB30, PB31.
4. Clusters of multiple GPIO pins are sharing the same supply pin.
5. The 49th pin of the WLCSP49 package is an additional GND pin.
6. SAM L22N: SERCOM[0:5]. SAM L22G, L22J: SERCOM[0:3].

Related Links

Configuration Summary

SERCOM USART and I2C Configurations

7.2 Other Functions

7.2.1 Oscillator Pinout

The oscillators are not mapped to the normal PORT functions and their multiplexing is controlled by registers in the Oscillators Controller (OSCCTRL) and in the 32K Oscillators Controller (OSC32KCTRL).

Table 7-2. Oscillator Pinout

Oscillator	Supply	Signal	I/O pin
XOSC	VDDIO	XIN	PB22
		XOUT	PB23
XOSC32K	VSWOUT	XIN32	PA00
		XOUT32	PA01

Note: In order to minimize the cycle-to-cycle jitter of the external oscillator, keep the neighboring pins as steady as possible. For neighboring pin details, refer to the Oscillator Pinout section.

Table 7-3. XOSC32K Jitter Minimization

Package	Steady Signal Recommended
L22N	PB00, PB01, PB02, PB03, PC00, PC01
L22J	PB00, PB01, PB02, PB03, PA02, PA03
L22G	PB02, PB03, PA02, PA03

Related Links

External Real Time Oscillator

7.2.2 Serial Wire Debug Interface Pinout

Only the SWCLK pin is mapped to the normal PORT functions. A debugger cold-plugging or hot-plugging detection will automatically switch the SWDIO port to the SWDIO function.

Table 7-4. Serial Wire Debug Interface Pinout

Signal	Supply	I/O pin
SWCLK	VDDIO	PA30
SWDIO	VDDIO	PA31

7.2.3 SERCOM USART and I²C Configurations

The SAM L22 has up to six instances of the serial communication interface (SERCOM) peripheral. The following table lists the supported communication protocols for each SERCOM instance.

Table 7-5. SERCOM USART and I²C Protocols

	SERCOM Instance					
Protocol	SERCOM0	SERCOM1	SERCOM2	SERCOM3	SERCOM4	SERCOM5
I²C	no	yes	yes	yes	yes	yes
I²C at 3.4MHz	no	yes	no	no	no	yes
USART	yes	yes	yes	yes	yes	yes

	SERCOM Instance					
Protocol	SERCOM0	SERCOM1	SERCOM2	SERCOM3	SERCOM4	SERCOM5
including RS485 and ISO 7816						
SPI	yes	yes	yes	yes	yes	yes

Note: Not all available I^2C pins support I^2C mode at 3.4MHz.

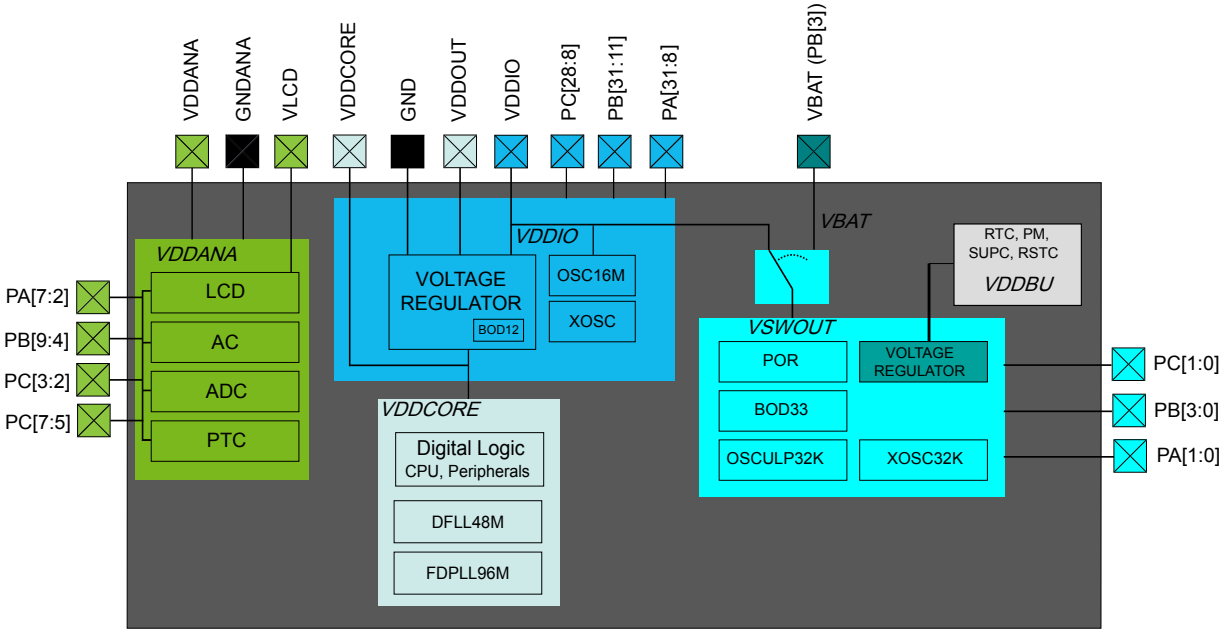
7.2.4 GPIO Pin Clusters

Table 7-6. GPIO Clusters

Package	Cluster	GPIO	Supplies Pin connected to the cluster	
100 pins	1	PA02, PA03, PB04, PB05, PC02, PC03	V_{DDANA} pin12	GNDANA pin11
	2	PA04, PA05, PA06, PA07, PB06, PB07, PB08, PB09, PC05, PC06, PC07	V_{DDANA} pin12, V_{DDANA} pin25	GNDANA pin11, GNDANA pin24
	3	PA08, PA09, PA10, PA11, PC08, PC09, PC10, PC11, PC12, PC13	V_{DDIO} pin36	GND pin37
	4	PA12, PA13, PA14, PA15, PB11, PB12, PB13, PB14, PB15, PC14, PC15	V_{DDIO} pin36, V_{DDIO} pin51	GND pin37, GND pin50
	5	PA16, PA17, PA18, PA19, PC16, PC17, PC18, PC19, PC20, PC21	V_{DDIO} pin51, V_{DDIO} pin63	GND pin50, GND pin62
	6	PA20, PA21, PA22, PA23, PA24, PA25, PB16, PB17, PB18, PB19, PB20, PB21	V_{DDIO} pin63, V_{DDIO} pin77	GND pin62, GND pin76
	7	PA27, PB22, PB23, PB24, PB25, PC24, PC25, PC26, PC27, PC28	V_{DDIO} pin77, V_{DDIO} pin92	GND pin76, GND pin90
	8	PA00, PA01, PA30, PA31, PB00, PB01, PB02, PB03, PB30, PB31, PC00, PC01	V_{DDIO} pin92	GND pin90
64 pins	1	PA02, PA03, PA04, PA05, PA06, PA07, PB04, PB05, PB06, PB07, PB08, PB09	V_{DDANA} pin8	GNDANA pin7
	2	PA08, PA09, PA10, PA11	V_{DDIO} pin21	GND pin22
	3	PA12, PA13, PA14, PA15, PB11, PB12, PB13, PB14, PB15	V_{DDIO} pin21, V_{DDIO} pin34	GND pin22, GND pin33
	4	PA16, PA17, PA18, PA19, PA20, PA21, PA22, PA23, PA24, PA25, PB16, PB17	V_{DDIO} pin34, V_{DDIO} pin48	GND pin33, GND pin47
	5	PA27, PB22, PB23	V_{DDIO} pin48, V_{DDIO} pin56	GND pin47, GND pin54
	6	PA00, PA01, PA30, PA31, PB00, PB01, PB02, PB03, PB30, PB31	V_{DDIO} pin56	GND pin54
48 pins	1	PA02, PA03, PA04, PA05, PA06, PA07, PB08, PB09	V_{DDANA} pin6	GNDANA pin5
	2	PA08, PA09, PA10, PA11	V_{DDIO} pin17	GND pin18
	3	PA12, PA13, PA14, PA15, PA16, PA17, PA18, PA19, PA20, PA21, PA22, PA23, PA24, PA25, PB11	V_{DDIO} pin17, V_{DDIO} pin36	GND pin18, GND pin35
	4	PA27, PB22, PB23	V_{DDIO} pin36, V_{DDIO} pin44	GND pin35, GND pin42
	5	PA00, PA01, PA30, PA31, PB02, PB03	V_{DDIO} pin44	GND pin42
49 pins	1	PA02, PA03, PA04, PA05, PA06, PA07, PB08, PB09	V_{DDANA} pin D7	GNDANA pin C7
	2	PA08, PA09, PA10, PA11	V_{DDIO} pin G5	GND pin F5
	3	PA12, PA13, PA14, PA15, PA16, PA17, PA18, PA19, PB11	V_{DDIO} pin G5, V_{DDIO} pin E1	GND pin F5, GND pin E2
	4	PA20, PA21, PA22, PA23, PA24, PA25	V_{DDIO} pin E1, V_{DDIO} pin A5	GND pin E2, GND pin D4
	4	PA27, PB22, PB23	V_{DDIO} pin E1, V_{DDIO} pin A5	GND pin D4, GND pin B3
	5	PA00, PA01, PA30, PA31, PB02, PB03	V_{DDIO} pin A5	GND pin B3

8. Power Supply and Start-Up Considerations

8.1 Power Domain Overview



The SAM L22 power domains are not independent of each other:

- VDDCORE and VDDIO share GND, whereas VDDANA refers to GNDANA.
- VDDCORE serves as the internal voltage regulator output.
- VSWOUT and VDDBU are internal power domains.

8.2 Power Supply Considerations

8.2.1 Power Supplies

The SAM L22 has several different power supply pins:

- VDDIO powers I/O lines and OSC16M, XOSC, the internal regulator for VDDCORE and the Automatic Power Switch. Voltage is 1.62V to 3.63V
- VDDANA powers I/O lines and the ADC, AC, LCD, and PTC. Voltage is 1.62V to 3.63V
- VLCD has two alternative functions:
  - Output of the LCD voltage pump when VLCD is generated internally. Output voltage is 2.5V to 3.5V.
  - Supply input for the bias generator when VLCD is provided externally by the application. Input voltage is 2.4 to 3.6V.
- VBAT powers the Automatic Power Switch. Voltage is 1.62V to 3.63V
- VDDCORE serves as the internal voltage regulator output. It powers the core, memories, peripherals, DFLL48M and FDPLL96M. Voltage is 0.9V to 1.2V typical.
- The Automatic Power Switch is a configurable switch that selects between VDDIO and VBAT as supply for the internal output VSWOUT, see the figure in Power Domain Overview.

The same voltage must be applied to both VDDIO and VDDANA. This common voltage is referred to as VDD in the datasheet.

The ground pins, GND, are common to VDDCORE, and VDDIO. The ground pin for VDDANA is GNDANA.

For decoupling recommendations for the different power supplies, refer to the schematic checklist.

8.2.2 Voltage Regulator

The SAM L22 internal Voltage Regulator has four different modes:

- Linear mode : This is the default mode when CPU and peripherals are running. It does not require an external inductor.
- Switching mode. This is the most efficient mode when the CPU and peripherals are running. This mode can be selected by software on the fly.
- Low Power (LP) mode. This is the default mode used when the chip is in standby mode.
- Shutdown mode. When the chip is in backup mode, the internal regulator is off.

Note that the Voltage Regulator modes are controlled by the Power Manager.

8.2.3 Typical Powering Schematic

The SAM L22 uses a single supply from 1.62V to 3.63V.

The following figure shows the recommended power supply connection.

Figure 8-1. Power Supply Connection for Linear Mode Only

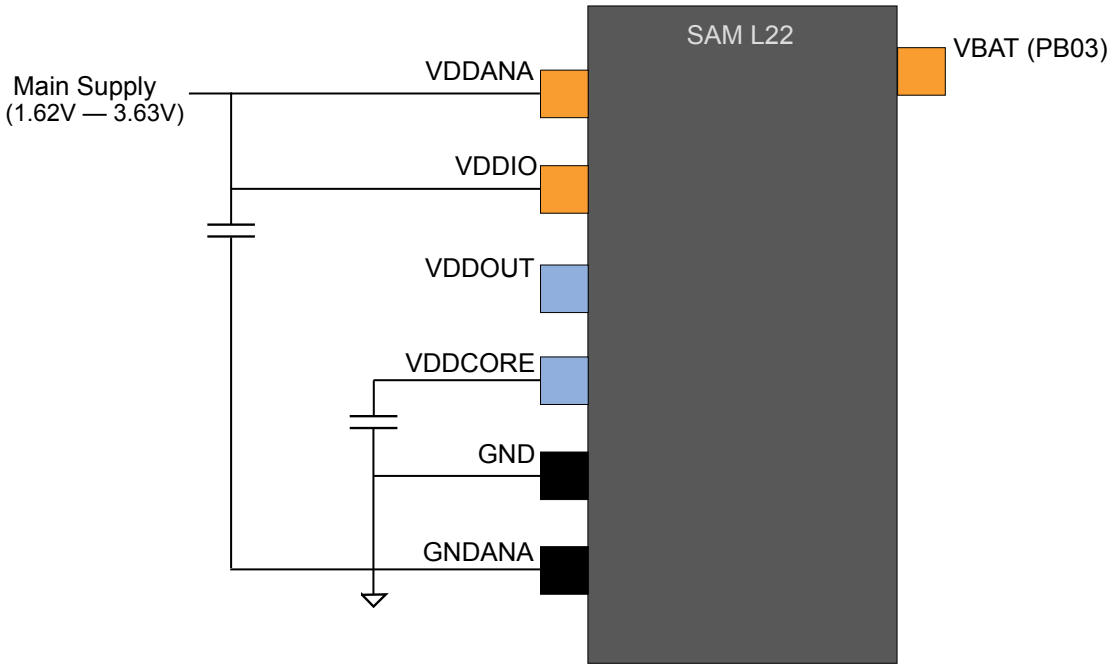


Figure 8-2. Power Supply Connection for Switching/Linear Mode

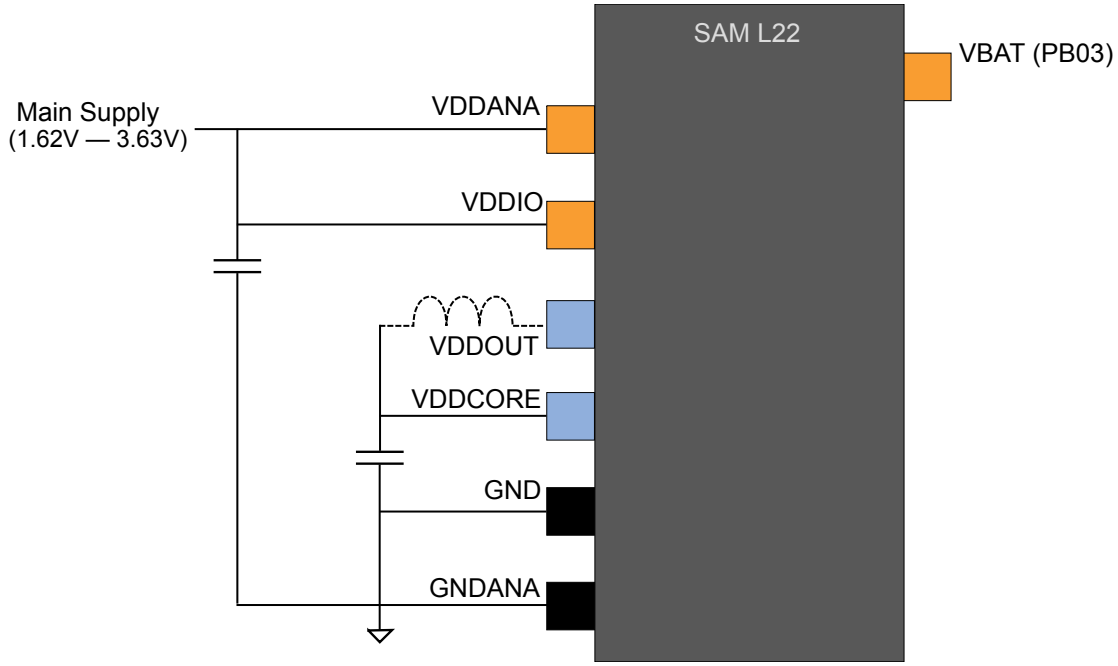
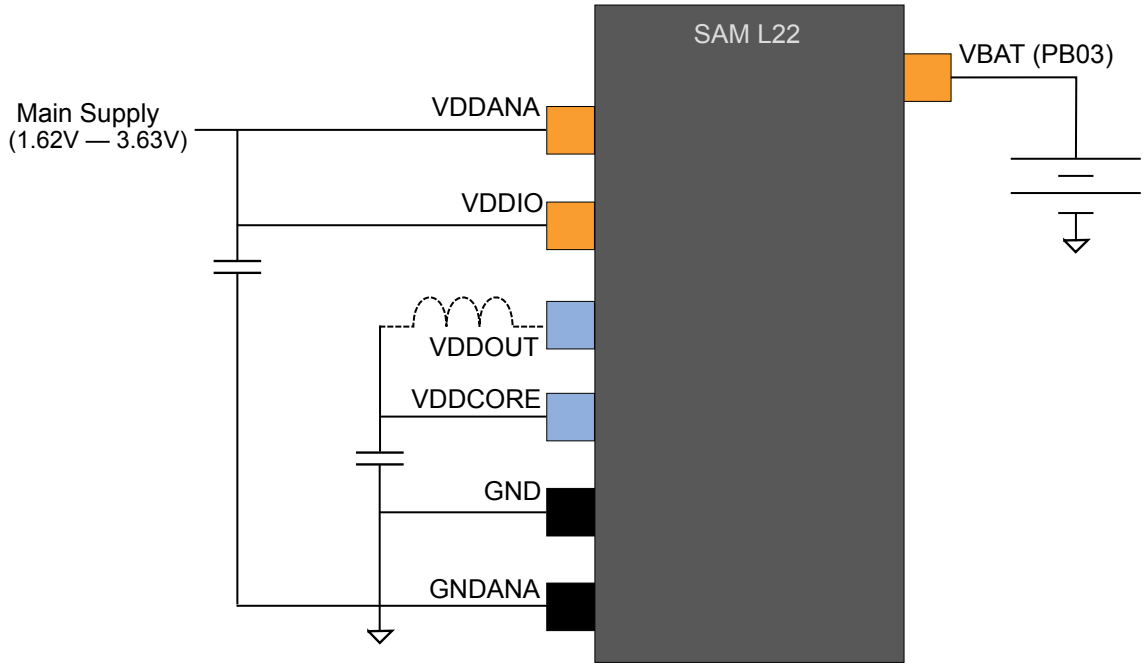


Figure 8-3. Power Supply Connection for Battery Backup



8.2.4 Power-Up Sequence

8.2.4.1 Supply Order

VDDIO and VDDANA must have the same supply sequence. Ideally, they must be connected together.

8.2.4.2 Minimum Rise Rate

One integrated power-on reset (POR) circuits monitoring VDDIO requires a minimum rise rate.

8.2.4.3 Maximum Rise Rate

The rise rate of the power supplies must not exceed the values described in Electrical Characteristics.

8.3 Power-Up

This section summarizes the power-up sequence of the SAM L22. The behavior after power-up is controlled by the Power Manager.

Related Links

PM – Power Manager

8.3.1 Starting of Internal Regulator

After power-up, the device is set to its initial state and kept in Reset, until the power has stabilized throughout the device. The default performance level after power-up is PL0.

The internal regulator provides the internal VDDCORE corresponding to this performance level. Once the external voltage VDDIO and the internal VDDCORE reach a stable value, the internal Reset is released.

Related Links

PM – Power Manager

8.3.2 Starting of Clocks

Once the power has stabilized and the internal Reset is released, the device will use a 4MHz clock by default. The clock source for this clock signal is OSC16M, which is enabled and configured at 4MHz after a reset by default. This is also the default time base for Generic Clock Generator 0. In turn, Generator 0 provides the main clock GCLK_MAIN which is used by the Power Manager (PM).

Some synchronous system clocks are active after Start-Up, allowing software execution. Refer to the “Clock Mask Register” section in the PM-Power Manager documentation for the list of clocks that are running by default. Synchronous system clocks that are running receive the 4MHz clock from Generic Clock Generator 0. Other generic clocks are disabled.

Related Links

PM – Power Manager

8.3.3 I/O Pins

After power-up, the I/O pins are tri-stated except PA30, which is pull-up enabled and configured as input.

8.3.4 Fetching of Initial Instructions

After Reset has been released, the CPU starts fetching PC and SP values from the Reset address, 0x00000000. This points to the first executable address in the internal Flash memory. The code read from the internal Flash can be used to configure the clock system and clock sources. See the related peripheral documentation for details. Refer to the ARM Architecture Reference Manual for more information on CPU startup (http://www.arm.com).

Related Links

PM – Power Manager

GCLK - Generic Clock Controller

OSCCTRL – Oscillators Controller

OSC32KCTRL – 32KHz Oscillators Controller

8.4 Power-On Reset and Brown-Out Detector

The SAM L22 embeds three features to monitor, warn and/or reset the device:

- POR: Power-on Reset on VSWOUT and VDDIO

- BOD33: Brown-out detector on VSWOUT/VBAT
- Brown-out detector internal to the voltage regulator for VDDCORE. BOD12 is calibrated in production and its calibration parameters are stored in the NVM User Row. This data should not be changed if the User Row is written to in order to assure correct behavior.

8.4.1 Power-On Reset on VSWOUT

VSWOUT is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VSWOUT goes below the threshold voltage, the entire chip is reset.

8.4.2 Power-On Reset on VDDIO

VDDIO is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VDDIO goes below the threshold voltage, all I/Os supplied by VDDIO are reset.

8.4.3 Brown-Out Detector on VSWOUT/VBAT

BOD33 monitors VSWOUT or VBAT depending on configuration.

Related Links

SUPC – Supply Controller

8.4.4 Brown-Out Detector on VDDCORE

Once the device has started up, BOD12 monitors the internal VDDCORE.

Related Links

SUPC – Supply Controller

8.5 Performance Level Overview

By default, the device will start in Performance Level 0. This PL0 is aiming for the lowest power consumption by limiting logic speeds and the CPU frequency. As a consequence, all GCLK will have limited capabilities, and some peripherals and clock sources will not work or with limited capabilities:

List of peripherals/clock sources not available in PL0:

- USB (limited by logic frequency)
- DFLL48M

List of peripherals/clock sources with limited capabilities in PL0:

- All AHB/APB peripherals are limited by CPU frequency
- DPLL96M: may be able to generate 48MHz internally, but the output cannot be used by logic
- GCLK: the maximum frequency is by factor 4 compared to PL2
- SW interface: the maximum frequency is by factor 4 compared to PL2
- TC: the maximum frequency is by factor 4 compared to PL2
- TCC:the maximum frequency is by factor 4 compared to PL2
- SERCOM: the maximum frequency is by factor 4 compared to PL2

List of peripherals/clock sources with full capabilities in PL0:

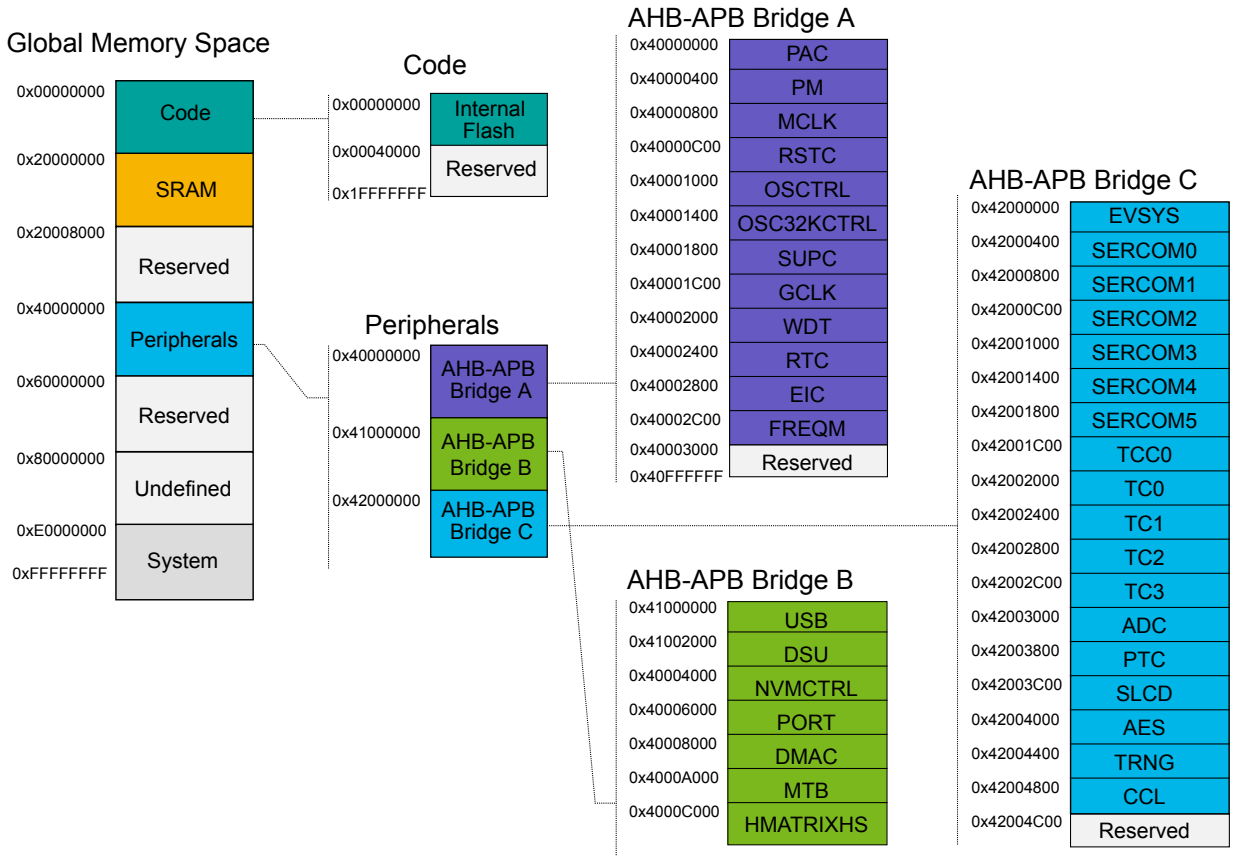
- AC
- ADC
- EIC
- OSC16M
- PTC

- All 32KHz clock sources and peripherals

Full functionality and capability will be ensured in PL2. When transitioning between performance levels, the Supply Controller (SUPC) will provide a configurable smooth voltage scaling transition.

9. Product Mapping

Figure 9-1. SAM L22 Product Mapping



10. Memories

10.1 Embedded Memories

- Internal high-speed Flash with Read-While-Write (RWW) capability on a section of the array
- Internal high-speed RAM, single-cycle access at full speed

10.2 Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed, and they are never remapped in any way, even during boot. The 32-bit physical address space is mapped as follows:

Table 10-1. SAM L22 Physical Memory Map

Memory	Start address	Size [KB]		
		SAML22x18(1)	SAML22x17(1)	SAML22x16(1)
Embedded Flash	0x00000000	256	128	64
Embedded RWW section	0x00400000	8	4	2
Embedded SRAM	0x20000000	32	16	8
Peripheral Bridge A	0x40000000	64	64	64
Peripheral Bridge B	0x41000000	64	64	64
Peripheral Bridge C	0x42000000	64	64	64
IOBUS	0x60000000	0.5	0.5	0.5

Note: 1. x = G, J, or E.

Table 10-2. Flash Memory Parameters

Device	Flash size [KB]	Number of pages	Page size [Bytes]
SAML22x18(1)	256	4096	64
SAML22x17(1)	128	2048	64
SAML22x16(1)	64	1024	64

Note: 1. x = G, J, or E.

Table 10-3. RWW Section Parameters(1)

Device	Flash size [KB]	Number of pages	Page size [Bytes]
SAML22x18(1)	8	128	64
SAML22x17(1)	4	64	64
SAML22x16(1)	2	32	64

Note: 1. x = G, J, or E.

10.3 NVM User Row Mapping

The Non Volatile Memory (NVM) User Row contains calibration data that are automatically read at device power-on.

The NVM User Row can be read at address 0x00804000.

To write the NVM User Row refer to the documentation of the NVMCTRL - Non-Volatile Memory Controller.

Note: When writing to the User Row, the new values do not get loaded by the other peripherals on the device until a device Reset occurs.

Table 10-4. NVM User Row Mapping

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
2:0	BOOTPROT	Used to select one of eight different bootloader sizes.	0x7	NVMCTRL
3	Reserved	—	0x1	—
6:4	EEPROM	Used to select one of eight different EEPROM sizes.	0x7	NVMCTRL
7	Reserved	—	0x1	—
13:8	BOD33 Level	BOD33 threshold level at power-on.	0x06	SUPC.BOD33
14	BOD33 Disable	BOD33 Disable at power-on.	0x0	SUPC.BOD33
16:15	BOD33 Action	BOD33 Action at power-on.	0x1	SUPC.BOD33
25:17	Reserved	Factory settings - do not change.	0x08F	-
26	WDT Enable	WDT Enable at power-on.	0x0	WDT.CTRLA
27	WDT Always-On	WDT Always-On at power-on.	0x0	WDT.CTRLA
31:28	WDT Period	WDT Period at power-on.	0xB	WDT.CONFIG
35:32	WDT Window	WDT Window mode time-out at power-on.	0xB	WDT.CONFIG
39:36	WDT EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on.	0xB	WDT.EWCTRL
40	WDT WEN	WDT Timer Window Mode Enable at power-on.	0x0	WDT.CTRLA
41	BOD33 Hysteresis	BOD33 Hysteresis configuration at power-on.	0x0	SUPC.BOD33
47:42	Reserved	Factory settings - do not change.	0x3E	—
63:48	LOCK	NVM Region Lock Bits.	0xFFFF	NVMCTRL

Related Links

NVMCTRL – Non-Volatile Memory Controller

SUPC – Supply Controller

BOD33
WDT – Watchdog Timer
CTRLA
CONFIG
EWCTRL

10.4 NVM Software Calibration Area Mapping

The NVM Software Calibration Area contains calibration data that are determined and written during production test. These calibration values should be read by the application software and written back to the corresponding register.

The NVM Software Calibration Area can be read at address 0x00806020.

The NVM Software Calibration Area can not be written.

Table 10-5. NVM Software Calibration Area Mapping

Bit Position	Name	Description
2:0	ADC LINEARITY	ADC Linearity Calibration. Should be written to CALIB register.
5:3	ADC BIASCAL	ADC Bias Calibration. Should be written to CALIB register.
12:6	Reserved	Reserved for future use.
17:13	USB TRANSN	USB TRANSN calibration value. Should be written to the USB PADCAL register.
22:18	USB TRANSP	USB TRANSP calibration value. Should be written to the USB PADCAL register.
25:23	USB TRIM	USB TRIM calibration value. Should be written to the USB PADCAL register.
31:26	DFLL48M COARSE CAL	DFLL48M Coarse calibration value. Should be written to the OSCCTRL DFLLVAL register.

10.5 Serial Number

Each device has a unique 128-bit serial number which is a concatenation of four 32-bit words contained at the following addresses:

Word 0: 0x0080A00C

Word 1: 0x0080A040

Word 2: 0x0080A044

Word 3: 0x0080A048

The uniqueness of the serial number is guaranteed only when using all 128 bits.

11. Processor and Architecture

11.1 Cortex M0+ Processor

The SAM L22 devices implement the ARM®Cortex™-M0+ processor, based on the ARMv6 Architecture and Thumb®-2 ISA. The Cortex M0+ is 100% instruction set compatible with its predecessor, the Cortex-M0 core, and upward compatible to Cortex-M3 and M4 cores. The implemented ARM Cortex-M0+ is revision r0p1. For more information refer to http://www.arm.com

11.1.1 Cortex M0+ Configuration

Table 11-1. Cortex M0+ Configuration

Features	Cortex-M0+ options	SAM L22 configuration
Interrupts	External interrupts 0-32	27
Data endianness	Little-endian or big-endian	Little-endian
SysTick timer	Present or absent	Present
Number of watchpoint comparators	0, 1, 2	2
Number of breakpoint comparators	0, 1, 2, 3, 4	4
Halting debug support	Present or absent	Present
Multiplier	Fast or small	Fast (single cycle)
Single-cycle I/O port	Present or absent	Present
Wake-up interrupt controller	Supported or not supported	Not supported
Vector Table Offset Register	Present or absent	Present
Unprivileged/Privileged support	Present or absent	Present
Memory Protection Unit	Not present or 8-region	8-region
Reset all registers	Present or absent	Absent
Instruction fetch width	16-bit only or mostly 32-bit	32-bit

The ARM Cortex-M0+ core has two bus interfaces:

- Single 32-bit AMBA-3 AHB-Lite system interface that provides connections to peripherals and all system memory, which includes flash and RAM.
- Single 32-bit I/O port bus interfacing to the PORT and DIVAS with 1-cycle loads and stores.

11.1.2 Cortex M0+ Peripherals

- System Control Space (SCS)
  - The processor provides debug through registers in the SCS. Refer to the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com)
- Nested Vectored Interrupt Controller (NVIC)
  - External interrupt signals connect to the NVIC, and the NVIC prioritizes the interrupts. Software can set the priority of each interrupt. The NVIC and the Cortex-M0+ processor core are closely coupled, providing low latency interrupt processing and efficient processing of late

arriving interrupts. Refer to NVIC-Nested Vector Interrupt Controller and the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com).
Note: When the CPU frequency is much higher than the APB frequency it is recommended to insert a memory read barrier after each CPU write to registers mapped on the APB. Failing to do so in such conditions may lead to unexpected behavior such as e.g. re-entering a peripheral interrupt handler just after leaving it.

- System Timer (SysTick)
  - The System Timer is a 24-bit timer clocked by CLK_CPU that extends the functionality of both the processor and the NVIC. Refer to the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com).
- System Control Block (SCB)
  - The System Control Block provides system implementation information, and system control. This includes configuration, control, and reporting of the system exceptions. Refer to the Cortex-M0+ Devices Generic User Guide for details (http://www.arm.com).
- Micro Trace Buffer (MTB)
  - The CoreSight MTB-M0+ (MTB) provides a simple execution trace capability to the Cortex-M0+ processor. Refer to section MTB-Micro Trace Buffer and the CoreSight MTB-M0+ Technical Reference Manual for details (http://www.arm.com).
- Memory Protection Unit (MPU)
  - The Memory Protection Unit divides the memory map into a number of regions, and defines the location, size, access permissions and memory attributes of each region. Refer to the Cortex-M0+ Devices Generic User Guide for details (http://www.arm.com)

11.1.3 Cortex M0+ Address Map

Table 11-2. Cortex-M0+ Address Map

Address	Peripheral
0xE000E000	System Control Space (SCS)
0xE000E010	System Timer (SysTick)
0xE000E100	Nested Vectored Interrupt Controller (NVIC)
0xE000ED00	System Control Block (SCB)
0x41006000	Micro Trace Buffer (MTB)

Related Links

Product Mapping

11.1.4 I/O Interface

The device allows direct access to PORT registers. Accesses to the AMBA® AHB-Lite™ and the single cycle I/O interface can be made concurrently, so the Cortex M0+ processor can fetch the next instructions while accessing the I/Os. This enables single cycle I/O access to be sustained for as long as necessary.

Related Links

PORT: IO Pin Controller

11.2 Nested Vector Interrupt Controller

11.2.1 Overview

The Nested Vectored Interrupt Controller (NVIC) in the SAM L22 supports 32 interrupts with four different priority levels. For more details, refer to the Cortex-M0+ Technical Reference Manual (http://www.arm.com).

11.2.2 Interrupt Line Mapping

Each of the interrupt lines is connected to one peripheral instance, as shown in the table below. Each peripheral can have one or more interrupt flags, located in the peripheral's Interrupt Flag Status and Clear (INTFLAG) register.

An interrupt flag is set when the interrupt condition occurs. Each interrupt in the peripheral can be individually enabled by writing a '1' to the corresponding bit in the peripheral's Interrupt Enable Set (INTENSET) register, and disabled by writing '1' to the corresponding bit in the peripheral's Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated from the peripheral when the interrupt flag is set and the corresponding interrupt is enabled.

The interrupt requests for one peripheral are ORed together on system level, generating one interrupt request for each peripheral. An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in ISPR/ICPR).

For the NVIC to activate the interrupt, it must be enabled in the NVIC interrupt enable register (SETENA/CLRENA bits in ISER/ICER). The NVIC interrupt priority registers IPR0-IPR7 provide a priority field for each interrupt.

Table 11-3. Interrupt Line Mapping

Peripheral source	NVIC line
EIC NMI – External Interrupt Controller	NMI
PM – Power Manager MCLK - Main Clock OSCCTRL - Oscillators Controller OSC32KCTRL - 32KHz Oscillators Controller PAC - Peripheral Access Controller SUPC - Supply Controller	0
WDT – Watchdog Timer	1
RTC – Real Time Counter	2
EIC – External Interrupt Controller	3
FREQM - Frequency Meter	4
USB - Universal Serial Bus	5
NVMCTRL – Non-Volatile Memory Controller	6
DMAC - Direct Memory Access Controller	7

Peripheral source	NVIC line
EVSYS – Event System	8
SERCOM0 – Serial Communication Interface 0	9
SERCOM1 – Serial Communication Interface 1	10
SERCOM2 – Serial Communication Interface 2	11
SERCOM3 – Serial Communication Interface 3	12
SERCOM4 – Serial Communication Interface 4	13
SERCOM5 – Serial Communication Interface 5	14
TCC0 – Timer Counter for Control 0	15
TC0 – Timer Counter 0	16
TC1 – Timer Counter 1	17
TC2 – Timer Counter 2	18
TC3 – Timer Counter 3	19
ADC – Analog-to-Digital Converter	20
AC – Analog Comparator	21
PTC – Peripheral Touch Controller	22
SLCD - Segmented LCD Controller	23
AES - Advanced Encryption Standard module	24
TRNG - True Random Number Generator	25

11.3 Micro Trace Buffer

11.3.1 Features

- Program flow tracing for the Cortex-M0+ processor
- MTB SRAM can be used for both trace and general purpose storage by the processor
- The position and size of the trace buffer in SRAM is configurable by software
- CoreSight compliant

11.3.2 Overview

When enabled, the MTB records the changes in program flow that are reported by the Cortex-M0+ processor over the execution trace interface. This interface is shared between the Cortex-M0+ processor and the CoreSight MTB-M0+. The information is stored by the MTB in the SRAM as trace packets. An off-chip debugger can extract the trace information using the Debug Access Port to read the trace information from the SRAM. The debugger can then reconstruct the program flow from this information.

The MTB stores trace information into the SRAM and gives the processor access to the SRAM simultaneously. The MTB ensures that trace write accesses have priority over processor accesses.

An execution trace packet consists of a pair of 32-bit words that the MTB generates when it detects a non-sequential change of the program pounter (PC) value. A non-sequential PC change can occur during

branch instructions or during exception entry. See the CoreSight MTB-M0+ Technical Reference Manual for more details on the MTB execution trace packet format.

Tracing is enabled when the MASTER.EN bit in the Master Trace Control Register is 1. There are various ways to set the bit to 1 to start tracing, or to 0 to stop tracing. See the CoreSight Cortex-M0+ Technical Reference Manual for more details on the Trace start and stop and for a detailed description of the MTB's MASTER register. The MTB can be programmed to stop tracing automatically when the memory fills to a specified watermark level or to start or stop tracing by writing directly to the MASTER.EN bit. If the watermark mechanism is not being used and the trace buffer overflows, then the buffer wraps around overwriting previous trace packets.

The base address of the MTB registers is 0x41006000; this address is also written in the CoreSight ROM Table. The offset of each register from the base address is fixed and as defined by the CoreSight MTB-M0+ Technical Reference Manual. The MTB has four programmable registers to control the behavior of the trace features:

- POSITION: Contains the trace write pointer and the wrap bit
- MASTER: Contains the main trace enable bit and other trace control fields
- FLOW: Contains the WATERMARK address and the AUTOSTOP and AUTOHALT control bits
- BASE: Indicates where the SRAM is located in the processor memory map. This register is provided to enable auto discovery of the MTB SRAM location by a debug agent

See the CoreSight MTB-M0+ Technical Reference Manual for a detailed description of these registers.

11.4 High-Speed Bus System

11.4.1 Overview

11.4.2 Features

High-Speed Bus Matrix has the following features:

- Symmetric crossbar bus switch implementation
- Allows concurrent accesses from different masters to different slaves
- 32-bit data bus
- Operation at a one-to-one clock frequency with the bus masters

11.4.3 Configuration

Figure 11-1. Master-Slave Relations High-Speed Bus Matrix

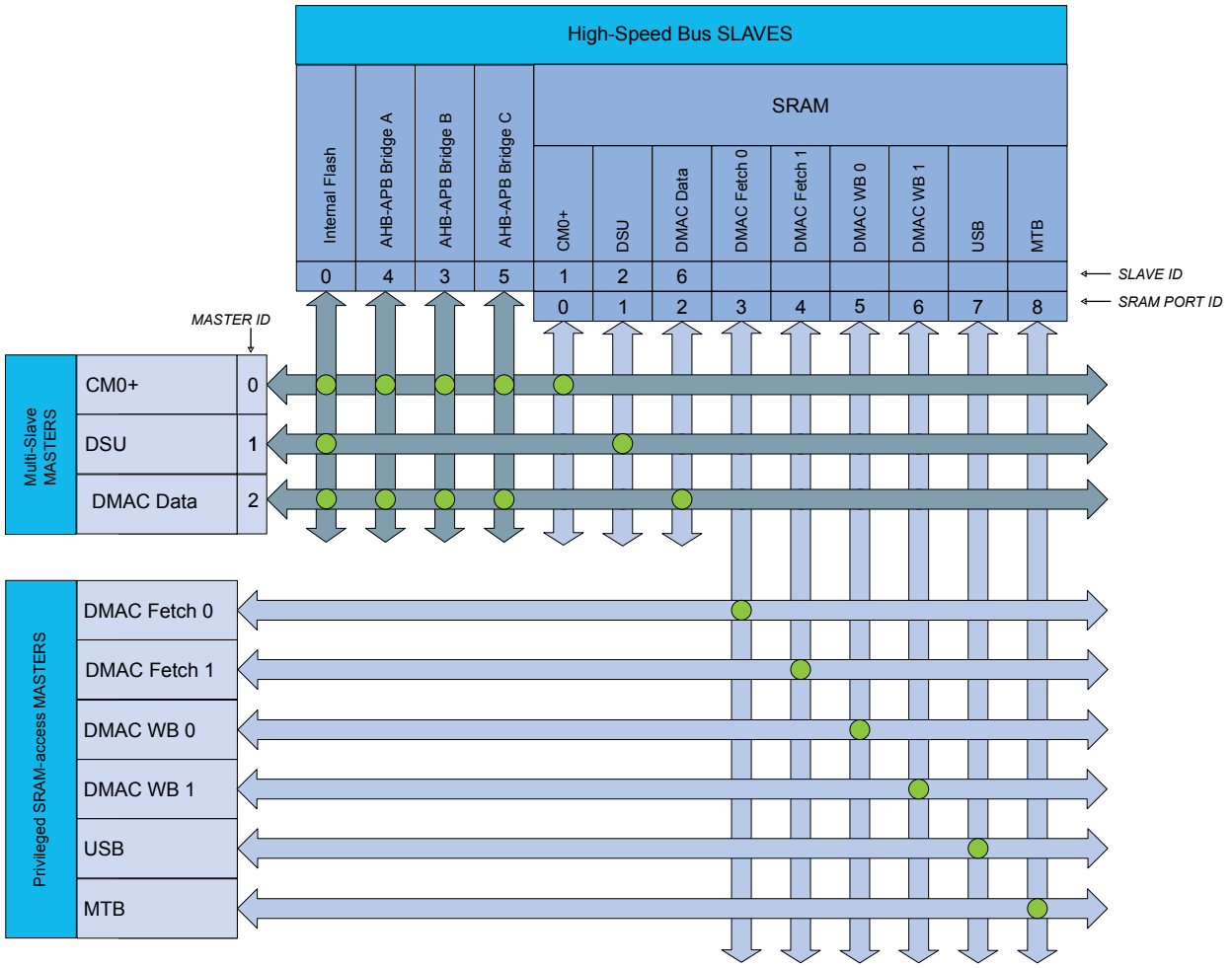


Table 11-4. High Speed Bus Matrix Masters

High-Speed Bus Matrix Masters	Master ID
CM0+ - Cortex M0+ Processor	0
DSU - Device Service Unit	1
DMAC - Direct Memory Access Controller / Data Access	2

Table 11-5. High-Speed Bus Matrix Slaves

High-Speed Bus Matrix Slaves	Slave ID
Internal Flash Memory	0
SRAM Port 0 - CM0+ Access	1
SRAM Port 1 - DSU Access	2
AHB-APB Bridge B	3
AHB-APB Bridge A	4

High-Speed Bus Matrix Slaves	Slave ID
AHB-APB Bridge C	5
SRAM Port 2 - DMAC Data Access	6

11.4.4 SRAM Quality of Service

To ensure that masters with latency requirements get sufficient priority when accessing RAM, priority levels can be assigned to the masters for different types of access.

The Quality of Service (QoS) level is independently selected for each master accessing the RAM. For any access to the RAM, the RAM also receives the QoS level. The QoS levels and their corresponding bit values for the QoS level configuration is shown in the table below.

Table 11-6. Quality of Service

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

If a master is configured with QoS level DISABLE (0x0) or LOW (0x1) there will be a minimum latency of one cycle for the RAM access.

The priority order for concurrent accesses are decided by two factors. First, the QoS level for the master and second, a static priority given by the port ID. The lowest port ID has the highest static priority. See the tables below for details.

The MTB has a fixed QoS level HIGH (0x3).

The CPU QoS level can be written/read, using 32-bit access only, at address 0x4100C114, bits [1:0]. Its reset value is 0x3.

Refer to different master QOSCTRL registers for configuring QoS for the other masters (USB, DMAC).

Table 11-7. SRAM Port Connections QoS

SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
CM0+ - Cortex M0+ Processor	0	Bus Matrix	0x4100C114, bits[1:0](1)	0x3
DSU - Device Service Unit	1	Bus Matrix	0x4100201C, bits[1:0](1)	0x2
DMAC - Direct Memory Access Controller - Data Access	2	Bus Matrix	IP-QOSCTRL.DQOS	0x2
DMAC - Direct Memory Access	3, 4	Direct	IP-QOSCTRL.FQOS	0x2

SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
Controller - Fetch Access				
DMAC - Direct Memory Access Controller - Write-Back Access	5, 6	Direct	IP-QOSCTRL.WRBQOS	0x2
USB - Universal Serial Bus	7	Direct	IP-QOSCTRL	0x3
MTB - Micro Trace Buffer	8	Direct	STATIC-3	0x3

Note: 1. Using 32-bit access only.

12. PAC - Peripheral Access Controller

12.1 Overview

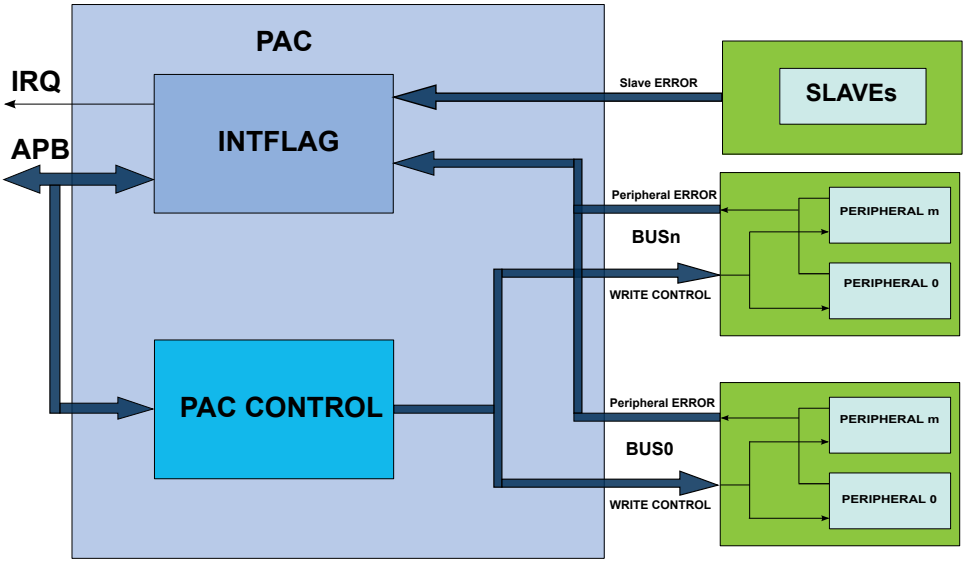
The Peripheral Access Controller provides an interface for the locking and unlocking of peripheral registers within the device. It reports all violations that could happen when accessing a peripheral: write protected access, illegal access, enable protected access, access when clock synchronization or software reset is on-going. These errors are reported in a unique interrupt flag for a peripheral. The PAC module also reports errors occurring at the slave bus level, when an access to a non-existing address is detected.

12.2 Features

- Manages write protection access and reports access errors for the peripheral modules or bridges

12.3 Block Diagram

Figure 12-1. PAC Block Diagram



12.4 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

12.4.1 IO Lines

Not applicable.

12.4.2 Power Management

The PAC can continue to operate in any sleep mode where the selected source clock is running. The PAC interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

12.4.3 Clocks

The PAC bus clock (CLK_PAC_APB) can be enabled and disabled in the Main Clock module. The default state of CLK_PAC_APB can be found in the related links.

Related Links

MCLK – Main Clock

Peripheral Clock Masking

12.4.4 DMA

Not applicable.

12.4.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the PAC interrupt requires the Interrupt Controller to be configured first.

Table 12-1. Interrupt Lines

Instances	NVIC Line
PAC	PACERR

Related Links

Nested Vector Interrupt Controller

12.4.6 Events

The events are connected to the Event System, which may need configuration.

Related Links

EVSYS – Event System

12.4.7 Debug Operation

When the CPU is halted in debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

12.4.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Write Control (WRCTRL) register
- AHB Slave Bus Interrupt Flag Status and Clear (INTFLAGAHB) register
- Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

12.5 Functional Description

12.5.1 Principle of Operation

The Peripheral Access Control module allows the user to set a write protection on peripheral modules and generate an interrupt in case of a peripheral access violation. The peripheral’s protection can be set,

cleared or locked for user convenience. A set of Interrupt Flag and Status registers informs the user on the status of the violation in the peripherals. In addition, slaves bus errors can be also reported in the cases where reserved area is accessed by the application.

12.5.2 Basic Operation

12.5.2.1 Initialization

After reset, the PAC is enabled.

12.5.2.2 Enabling and Resetting

The PAC is always enabled after reset.

Only a hardware reset will reset the PAC module.

12.5.2.3 Operations

The PAC module allows the user to set, clear or lock the write protection status of all peripherals on all Peripheral Bridges.

If a peripheral register violation occurs, the Peripheral Interrupt Flag n registers (INTFLAGn) are updated to inform the user on the status of the violation in the peripherals connected to the Peripheral Bridge n (n = A,B,C ...). The corresponding Peripheral Write Control Status n register (STATUSn) gives the state of the write protection for all peripherals connected to the corresponding Peripheral Bridge n. Refer to the Peripheral Access Errors for details.

The PAC module reports also the errors occurring at slave bus level when an access to reserved area is detected. AHB Slave Bus Interrupt Flag register (INTFLAGAHB) informs the user on the status of the violation in the corresponding slave. Refer to the AHB Slave Bus Errors for details.

12.5.2.4 Peripheral Access Errors

The following events will generate a Peripheral Access Error:

- Protected write: To avoid unexpected writes to a peripheral's registers, each peripheral can be write protected. Only the registers denoted as “PAC Write-Protection” in the module’s datasheet can be protected. If a peripheral is not write protected, write data accesses are performed normally. If a peripheral is write protected and if a write access is attempted, data will not be written and peripheral returns an access error. The corresponding interrupt flag bit in the INTFLAGn register will be set.
- Illegal access: Access to an unimplemented register within the module.
- Synchronized write error: For write-synchronized registers an error will be reported if the register is written while a synchronization is ongoing.

When any of the INTFLAGn registers bit are set, an interrupt will be requested if the PAC interrupt enable bit is set.

Related Links

Register Synchronization

12.5.2.5 Write Access Protection Management

Peripheral access control can be enabled or disabled by writing to the WRCTRL register.

The data written to the WRCTRL register is composed of two fields; WRCTRL.PERID and WRCTRL.KEY. The WRCTRL.PERID is an unique identifier corresponding to a peripheral. The WRCTRL.KEY is a key value that defines the operation to be done on the control access bit. These operations can be “clear protection”, “set protection” and “set and lock protection bit”.

The “clear protection” operation will remove the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.

The “set protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.

The “set and lock protection” operation will permanently set the write access protection for the peripheral selected by WRCTRL.PERID. The write access protection will only be cleared by a hardware reset.

The peripheral access control status can be read from the corresponding STATUSn register.

12.5.2.6 Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register will effectively change the access protection. Other type of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGn.PAC bit corresponding to the PAC module.

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected then the PAC returns an error, and similarly for a double clear protection operation. In addition, an error is generated when writing a “set and lock” protection to a write-protected peripheral or when a write access is done to a locked set protection.

This can be used to ensure that the application follows the intended program flow by always following a write protect with an unprotect and conversely. However in applications where a write protected peripheral is used in several contexts, e.g. interrupt, care should be taken so that either the interrupt can not happen while the main application or other interrupt levels manipulates the write protection status or when the interrupt handler needs to unprotect the peripheral based on the current protection status by reading the STATUS register.

The errors generated while accessing the PAC module registers (eg. key error, double protect error...) will set the INTFLAGn.PAC flag.

12.5.2.7 AHB Slave Bus Errors

The PAC module reports errors occurring at the AHB Slave bus level. These errors are generated when an access is performed at an address where no slave (bridge or peripheral) is mapped . These errors are reported in the corresponding bits of the INTFLAGAHB register.

12.5.2.8 Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set.

12.5.3 DMA Operation

Not applicable.

12.5.4 Interrupts

The PAC has the following interrupt source:

- Error (ERR): Indicates that a peripheral access violation occurred in one of the peripherals controlled by the PAC module, or a bridge error occurred in one of the bridges reported by the PAC
  - This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the PAC is reset. All interrupt requests from the peripheral are ORed together

on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAGAHB and INTFLAGn registers to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller

Sleep Mode Controller

12.5.5 Events

The PAC can generate the following output event:

- Error (ERR): Generated when one of the interrupt flag registers bits is set

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.ERREO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event.

12.5.6 Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available master (CPU, DMA) is running. The PAC will continue to catch access errors from module and generate interrupts or events.

12.5.7 Synchronization

Not applicable.

12.6 Register Summary

Offset	Name	Bit Pos.								
0x00	WRCTRL	7:0	PERID[7:0]							
0x01		15:8	PERID[15:8]							
0x02		23:16	KEY[7:0]							
0x03		31:24								
0x04	EVCTRL	7:0								ERREO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								ERR
0x09	INTENSET	7:0								ERR
0x0A ... 0x0F	Reserved									
0x10	INTFLAGAHB	7:0		HSRAMDMAC	HPB2	HPB0	HPB1	HSRAMDSU	HSRAMCM0P	FLASH
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	INTFLAGA	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x15		15:8					FREQM	EIC	RTC	WDT
0x16		23:16								
0x17		31:24								
0x18	INTFLAGB	7:0			MTB	DMAC	PORT	NVMCTRL	DSU	USB
0x19		15:8								
0x1A		23:16								
0x1B		31:24								
0x1C	INTFLAGC	7:0	TCC	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
0x1D		15:8	SLCD	PTC	AC	ADC	TC3	TC2	TC1	TC0
0x1E		23:16						CCL	TRNG	AES
0x1F		31:24								
0x20 ... 0x33	Reserved									
0x34	STATUSA	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x35		15:8					FREQM	EIC	RTC	WDT
0x36		23:16								
0x37		31:24								
0x38	STATUSB	7:0			MTB	DMAC	PORT	NVMCTRL	DSU	USB
0x39		15:8								
0x3A		23:16								
0x3B		31:24								
0x3C	STATUSC	7:0	TCC	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
0x3D		15:8	SLCD	PTC	AC	ADC	TC3	TC2	TC1	TC0

Offset	Name	Bit Pos.								
0x3E		23:16						CCL	TRNG	AES
0x3F		31:24								

12.7 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the related links.

Related Links

Register Synchronization

12.7.1 Write Control

Name: WRCTRL
Offset: 0x0 [ID-00000a18]
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – KEY[7:0]: Peripheral Access Control Key
These bits define the peripheral access control key:

Value	Name	Description
0x0	OFF	No action
0x1	CLEAR	Clear the peripheral write control
0x2	SET	Set the peripheral write control
0x3	LOCK	Set and lock the peripheral write control until the next hardware reset

Bits 15:0 – PERID[15:0]: Peripheral Identifier
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY. The Peripheral Identifier is calculated following formula:

$PERID = 32^{*}\,\mathrm{BridgeNumber} + \mathrm{N}$

Where BridgeNumber represents the Peripheral Bridge Number (0 for Peripheral Bridge A, 1 for Peripheral Bridge B, etc). N represents the peripheral index from the respective Bridge Number:

Table 12-2. PERID Values

Periph. Bridge Name	BridgeNumber	PERID Values
A	0	0+N
B	1	32+N
C	2	64+N
D	3	96+N
E	4	128+N

12.7.2 Event Control

Name: EVCTRL
Offset: 0x04 [ID-00000a18]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
								ERREO
Access								R/W
Reset								0

Bit 0 – ERREO: Peripheral Access Error Event Output
This bit indicates if the Peripheral Access Error Event Output is enabled or not. When enabled, an event will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Value	Description
0	Peripheral Access Error Event Output is disabled.
1	Peripheral Access Error Event Output is enabled.

12.7.3 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x08 [ID-00000a18]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/W
Reset								0

Bit 0 – ERR: Peripheral Access Error Interrupt Disable
This bit indicates that the Peripheral Access Error Interrupt is disabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Peripheral Access Error interrupt Enable bit and disables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

12.7.4 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENCLR).

Name: INTENSET
Offset: 0x09 [ID-00000a18]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/W
Reset								0

Bit 0 – ERR: Peripheral Access Error Interrupt Enable
This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Peripheral Access Error interrupt Enable bit and enables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

12.7.5 AHB Slave Bus Interrupt Flag Status and Clear

This flag is cleared by writing a '1' to the flag.

This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Name: INTFLAGAHB
Offset: 0x10 [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		HSRAMDMAC	HPB2	HPB0	HPB1	HSRAMDSU	HSRAMCM0P	FLASH
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – HSRAMDMAC: Interrupt Flag for SLAVE HSRAMDMAC

Bit 5 – HPB2: Interrupt Flag for SLAVE HPB2

Bit 4 – HPB0: Interrupt Flag for SLAVE HPB0

Bit 3 – HPB1: Interrupt Flag for SLAVE HPB1

Bit 2 – HSRAMDSU: Interrupt Flag for SLAVE HSRAMDSU

Bit 1 – HSRAMCM0P: Interrupt Flag for SLAVE HSRAMCM0P

Bit 0 – FLASH: Interrupt Flag for SLAVE FLASH

12.7.6 Peripheral Interrupt Flag Status and Clear A

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGA bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGA interrupt flag.

Name: INTFLAGA
Offset: 0x14 [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					FREQM	EIC	RTC	WDT
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 11 – FREQM: Interrupt Flag for FREQM

Bit 10 – EIC: Interrupt Flag for EIC

Bit 9 – RTC: Interrupt Flag for RTC

Bit 8 – WDT: Interrupt Flag for WDT

Bit 7 – GCLK: Interrupt Flag for GCLK

Bit 6 – SUPC: Interrupt Flag for SUPC

Bit 5 – OSC32KCTRL: Interrupt Flag for OSC32KCTRL

Bit 4 – OSCCTRL: Interrupt Flag for OSCCTRL

Bit 3 – RSTC: Interrupt Flag for RSTC

Bit 2 – MCLK: Interrupt Flag for MCLK

Bit 1 – PM: Interrupt Flag for PM

Bit 0 – PAC: Interrupt Flag for PAC

12.7.7 Peripheral Interrupt Flag Status and Clear B

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGB bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGB interrupt flag.

Name: INTFLAGB
Offset: 0x18 [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			MTB	DMAC	PORT	NVMCTRL	DSU	USB
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – MTB: Interrupt Flag for MTB

Bit 4 – DMAC: Interrupt Flag for DMAC

Bit 3 – PORT: Interrupt Flag for PORT

Bit 2 – NVMCTRL: Interrupt Flag for NVMCTRL

Bit 1 – DSU: Interrupt Flag for DSU

Bit 0 – USB: Interrupt Flag for USB

12.7.8 Peripheral Interrupt Flag Status and Clear C

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGC bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGC interrupt flag.

Name: INTFLAGC
Offset: 0x1C [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CCL	TRNG	AES
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	SLCD	PTC	AC	ADC	TC3	TC2	TC1	TC0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TCC	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 18 – CCL: Interrupt Flag for CCL

Bit 17 – TRNG: Interrupt Flag for TRNG

Bit 16 – AES: Interrupt Flag for AES

Bit 15 – SLCD: Interrupt Flag for SLCD

Bit 14 – PTC: Interrupt Flag for PTC

Bit 13 – AC: Interrupt Flag for AC

Bit 12 – ADC: Interrupt Flag for ADC

Bits 8, 9, 10, 11 – TC: Interrupt Flag for TCn [n = 3..0]

Bit 7 – TCC: Interrupt Flag for TCC

Bits 1, 2, 3, 4, 5, 6 – SERCOM: Interrupt Flag for SERCOMn [n = 5..0]

Bit 0 – EVSYS: Interrupt Flag for EVSYS

12.7.9 Peripheral Write Protection Status A

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSA
Offset: 0x34 [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					FREQM	EIC	RTC	WDT
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 11 – FREQM: Peripheral FREQM Write Protection Status

Bit 10 – EIC: Peripheral EIC Write Protection Status

Bit 9 – RTC: Peripheral RTC Write Protection Status

Bit 8 – WDT: Peripheral WDT Write Protection Status

Bit 7 – GCLK: Peripheral GCLK Write Protection Status

Bit 6 – SUPC: Peripheral SUPC Write Protection Status

Bit 5 – OSC32KCTRL: Peripheral OSC32KCTRL Write Protection Status

Bit 4 – OSCCTRL: Peripheral OSCCTRL Write Protection Status

Bit 3 – RSTC: Peripheral RSTC Write Protection Status

Bit 2 – MCLK: Peripheral MCLK Write Protection Status

Bit 1 – PM: Peripheral PM Write Protection Status

Bit 0 – PAC: Peripheral PAC Write Protection Status

12.7.10 Peripheral Write Protection Status B

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSB
Offset: 0x38 [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			MTB	DMAC	PORT	NVMCTRL	DSU	USB
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – MTB: Peripheral MTB Write Protection Status

Bit 4 – DMAC: Peripheral DMAC Write Protection Status

Bit 3 – PORT: Peripheral PORt Write Protection Status

Bit 2 – NVMCTRL: Peripheral NVMCTRL Write Protection Status

Bit 1 – DSU: Peripheral DSU Write Protection Status

Bit 0 – USB: Peripheral USB Write Protection Status

12.7.11 Peripheral Write Protection Status C

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSC
Offset: 0x3C [ID-00000a18]
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CCL	TRNG	AES
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	SLCD	PTC	AC	ADC	TC3	TC2	TC1	TC0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TCC	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 18 – CCL: Peripheral CCL Write Protection Status

Bit 17 – TRNG: Peripheral TRNG Write Protection Status

Bit 16 – AES: Peripheral AES Write Protection Status

Bit 15 – SLCD: Peripheral SLCD Write Protection Status

Bit 14 – PTC: Peripheral PTC Write Protection Status

Bit 13 – AC: Peripheral ADC Write Protection Status

Bit 12 – ADC: Peripheral ADC Write Protection Status

Bits 8, 9, 10, 11 – TC: Peripheral TCn Write Protection Status [n = 3..0]

Bit 7 – TCC: Peripheral TCC Write Protection Status

Bits 1, 2, 3, 4, 5, 6 – SERCOM: Peripheral SERCOMn Write Protection Status [n = 5..0]

Bit 0 – EVSYS: Peripheral EVSYS Write Protection Status

13. Peripherals Configuration Summary

Table 13-1. Peripherals Configuration Summary

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
AHB-APB Bridge A	0x40000000	—	0	Y	—	—	CPU	—	—	—	—	—	—	N/A	PDTOP
PAC	0x40000000	0	7	Y	0	Y	CPU	—	0	—	—	69 : ACCERR	—	N/A	PDTOP
PM	0x40000400	0	—	—	1	Y	Backup	—	1	N	—	—	—	N/A	PDBACKUP
MCLK	0x40000800	0	—	—	2	Y	CPU	—	2	N	—	—	—	Y	PDTOP
RSTC	0x40000C00	—	—	—	3	Y	Backup	—	3	N	—	—	—	N/A	PDBACKUP
OSCCTRL	0x40001000	0	—	—	4	Y	CPU	0: DFLL48M reference 1: FDPLL96M clk source 2: FDPLL96M 32kHz	4	N	—	0: CFD	—	Y	PDTOP
OSC32KCTRL	0x40001400	0	—	—	5	Y	Backup	—	5	N	—	1: CFD	—	—	PDBACKUP
SUPC	0x40001800	0	—	—	6	Y	Backup	—	6	N	—	—	—	N/A	PDBACKUP
GCLK	0x40001C00	—	—	—	7	Y	CPU	—	7	N	—	—	—	N/A	PDTOP
WDT	0x40002000	1	—	—	8	Y	CPU	—	8	N	—	—	—	Y	PDTOP
RTC	0x40002400	2	—	—	9	Y	Backup	—	9	N	0: TAMPEVT	2: CMP0/ALARM0 3: CMP1 4: TAMPER 5: OVF 6-13: PER0-7	1: TIMESTAMP	Y	PDBACKUP
EIC	0x40002800	3, NMI	—	—	10	Y	CPU	3	10	N	—	14-29: EXTINT0-15	—	Y	PDTOP
FREQM	0x40002C00	4	—	—	11	Y	CPU	4: FREQM_MSR 5: FREQM_REF	11	N	—	4: DONE	—	Y	PDTOP
AHB-APB Bridge B	0x41000000	—	1	Y	—	—	CPU	—	—	—	—	—	—	N/A	PDTOP
USB	0x41000000	5	4	Y	0	Y	CPU	6	0	N	—	—	—	Y	PDTOP
DSU	0x41002000	—	5	Y	1	Y	CPU	—	1	Y	—	—	—	N/A	PDTOP
NVMCTRL	0x41004000	6	8	Y	2	Y	CPU	—	2	N	—	—	—	Y	PDTOP

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
PORT	0x41006000	—	—	—	10	Y	CPU	—	3	N	1-4 : EV0-3	—	—	Y	PDTOP
DMAC	0x41008000	7	4	Y	—	—	CPU	—	4	—	5-8: CH0-4	30-33: CH0-4	—	Y	PDTOP
MTB	0x410A0000	—	—	—	—	—	CPU	—	—	—	—	—	—	N/A	PDTOP
AHB-APB Bridge C	0x42000000	—	2	Y	—	—	CPU	—	—	—	—	—	—	N/A	PDTOP
EVSYS	0x42000000	8	—	—	0	Y	CPU	7-14: one per CHANNEL	0	N	—	—	—	Y	PDTOP
SERCOM0	0x42000400	9	—	—	1	Y	CPU	16: CORE 15: SLOW	1	N	—	—	2: RX 3: TX	Y	PDTOP
SERCOM1	0x42000800	10	—	—	2	Y	CPU	17: CORE 15: SLOW	2	N	—	—	4: RX 5: TX	Y	PDTOP
SERCOM2	0x42000A00	11	—	—	3	Y	CPU	18: CORE 15: SLOW	3	N	—	—	6: RX 7: TX	Y	PDTOP
SERCOM3	0x42001000	12	—	—	4	Y	CPU	19: CORE 15: SLOW	4	N	—	—	8: RX 9: TX	Y	PDTOP
SERCOM4	0x42001400	13	—	—	5	Y	CPU	20: CORE 15: SLOW	5	N	—	—	10: RX 11: TX	Y	PDTOP
SERCOM5	0x42001800	14	—	—	6	Y	CPU	21: CORE 15: SLOW	6	N	—	—	12: RX 13: TX	Y	PDTOP
TCC0	0x42001C00	15	—	—	7	Y	CPU	22	7	N	9-10: EV0-1 11-14: MC0-3	34: OVF 35: TRG 36: CNT 37-40: MC0-3	14: OVF 15-18: MC0-3	Y	PDTOP
TC0	0x42002000	16	—	—	8	Y	CPU	23	8	N	15: EVU	41: OVF 42-43: MC0-1	19: OVF 20-21: MC0-1	Y	PDTOP
TC1	0x42002400	17	—	—	9	Y	CPU	23	9	N	16: EVU	44: OVF 45-46: MC0-1	22: OVF 23-24: MC0-1	Y	PDTOP
TC2	0x42002800	18	—	—	10	Y	CPU	24	10	N	17: EVU	47: OVF 48-49: MC0-1	25: OVF 26-27: MC0-1	Y	PDTOP
TC3	0x42002C00	19	—	—	11	Y	CPU	24	11	N	18: EVU	50: OVF 51-52: MC0-1	28: OVF 29-30: MC0-1	Y	PDTOP
ADC	0x42003000	20	—	—	12	Y	CPU	25	12	N	19: START	53: RESRDY	31: RESRDY	Y	PDTOP

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
											20: SYNC	54: WINMON			
AC	0x42003400	21	—	—	13	Y	CPU	26	13	N	21-22: SOC0-1	55-56: COMP0-1 57: WIN0	—	Y	PDTOP
PTC	0x42003800	22	—	—	14	Y	CPU	27	14	N	23: STCONV	58: EOC 59: WCOMP	—	—	PDTOP
SLCD	0x42003C00	23	—	—	15	Y	CPU	—	15	N	—	60-62: FC0-2 63: DT	32: DMU 33: ACMDRDY 34: ABMRDY	—	PDTOP
AES	0x42004000	24	—	—	16	Y	CPU	—	16	N	—	—	35 : WR 36 : RD	Y	PDTOP
TRNG	0x42004400	25	—	—	17	Y	CPU	—	17	N	—	64 : READY	—	Y	PDTOP
CCL	0x42004800	—	—	—	18	Y	CPU	28	18	N	24 : LUTIN0 25 : LUTIN1 26: LUTIN2 27: LUTIN3	65 : LUTOUT0 66 : LUTOUT1 67: LUTOUT2 68: LUTOUT3	—	Y	PDTOP

14. DSU - Device Service Unit

14.1 Overview

The Device Service Unit (DSU) provides a means to detect debugger probes. This enables the ARM Debug Access Port (DAP) to have control over multiplexed debug pads and CPU reset. The DSU also provides system-level services to debug adapters in an ARM debug system. It implements a CoreSight Debug ROM that provides device identification as well as identification of other debug components within the system. Hence, it complies with the ARM Peripheral Identification specification. The DSU also provides system services to applications that need memory testing, as required for IEC60730 Class B compliance, for example. The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix. For security reasons, some of the DSU features will be limited or unavailable when the device is protected by the NVMCTRL security bit.

Related Links

NVMCTRL – Non-Volatile Memory Controller

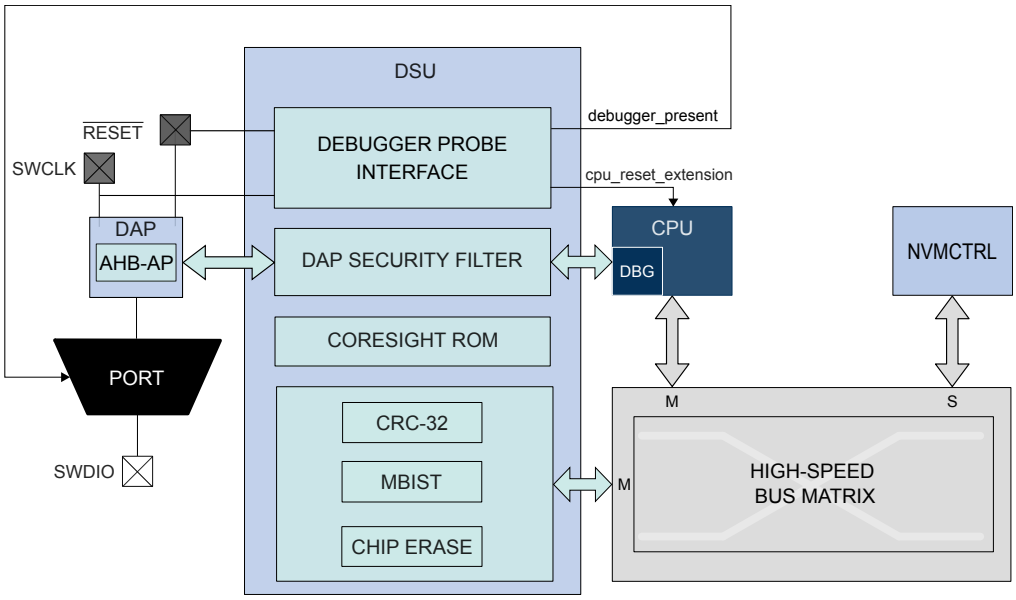
Security Bit

14.2 Features

- CPU reset extension
- Debugger probe detection (Cold- and Hot-Plugging)
- Chip-Erase command and status
- 32-bit cyclic redundancy check (CRC32) of any memory accessible through the bus matrix
- ARM® CoreSight™ compliant device identification
- Two debug communications channels
- Debug access port security filter
- Onboard memory built-in self-test (MBIST)

14.3 Block Diagram

Figure 14-1. DSU Block Diagram



14.4 Signal Description

The DSU uses three signals to function.

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital Input	External reset
SWCLK	Digital Input	SW clock
SWDIO	Digital I/O	SW bidirectional data pin

Related Links

I/O Multiplexing and Considerations

14.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

14.5.1 IO Lines

The SWCLK pin is by default assigned to the DSU module to allow debugger probe detection and to stretch the CPU reset phase. For more information, refer to Debugger Probe Detection. The Hot-Plugging feature depends on the PORT configuration. If the SWCLK pin function is changed in the PORT or if the PORT_MUX is disabled, the Hot-Plugging feature is disabled until a power-reset or an external reset.

14.5.2 Power Management

The DSU will continue to operate in any sleep mode where the selected source clock is running.

Related Links

PM – Power Manager

14.5.3 Clocks

The DSU bus clocks (CLK_DSU_APB and CLK_DSU_AHB) can be enabled and disabled by the Main Clock Controller.

Related Links

PM – Power Manager

MCLK – Main Clock

Peripheral Clock Masking

14.5.4 Interrupts

Not applicable.

14.5.5 Events

Not applicable.

14.5.6 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Debug Communication Channel 0 register (DCC0)
- Debug Communication Channel 1 register (DCC1)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

14.5.7 Analog Connections

Not applicable.

14.6 Debug Operation

14.6.1 Principle of Operation

The DSU provides basic services to allow on-chip debug using the ARM Debug Access Port and the ARM processor debug resources:

- CPU reset extension
- Debugger probe detection

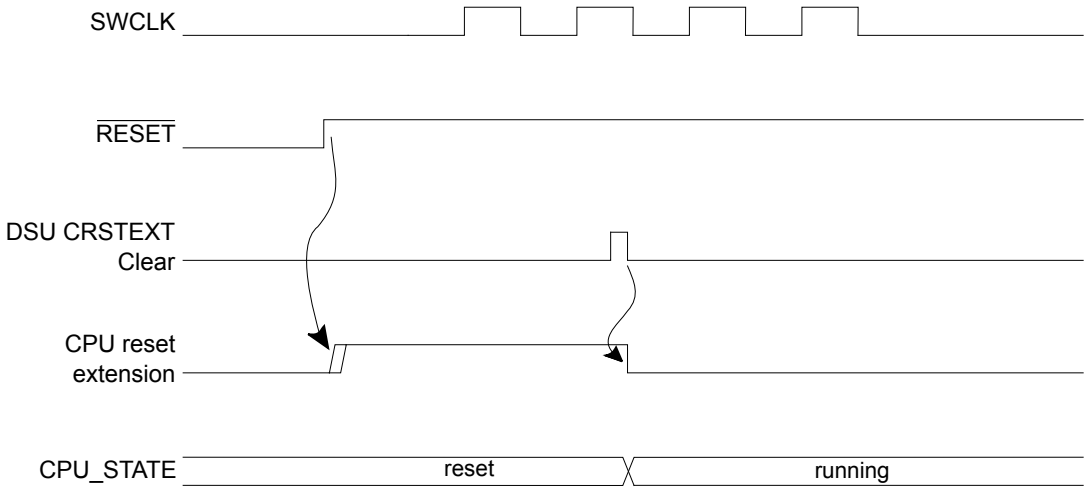
For more details on the ARM debug components, refer to the ARM Debug Interface v5 Architecture Specification.

14.6.2 CPU Reset Extension

“CPU reset extension” refers to the extension of the reset phase of the CPU core after the external reset is released. This ensures that the CPU is not executing code at startup while a debugger connects to the system. It is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At startup, SWCLK is internally pulled up to avoid false detection of a debugger if SWCLK is left unconnected. When the CPU is held in the reset extension phase, the CPU Reset Extension bit of the Status A register (STATUSA.CRSTEXT) is

set. To release the CPU, write a '1' to STATUSA.CRSTEXT. STATUSA.CRSTEXT will then be set to zero. Writing a '0' to STATUSA.CRSTEXT has no effect. For security reasons, it is not possible to release the CPU reset extension when the device is protected by the NVMCTRL security bit. Trying to do so sets the Protection Error bit (PERR) of the Status A register (STATUSA.PERR).

Figure 14-2. Typical CPU Reset Extension Set and Clear Timing Diagram



Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

14.6.3 Debugger Probe Detection

14.6.3.1 Cold Plugging

Cold-Plugging is the detection of a debugger when the system is in reset. Cold-Plugging is detected when the CPU reset extension is requested, as described above.

14.6.3.2 Hot Plugging

Hot-Plugging is the detection of a debugger probe when the system is not in reset. Hot-Plugging is not possible under reset because the detector is reset when POR or $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK function is changed, the Hot-Plugging feature is disabled until a power-reset or external reset occurs. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 14-3. Hot-Plugging Detection Timing Diagram



The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when the device is protected by the NVMCTRL security bit.

This detection requires that pads are correctly powered. Thus, at cold startup, this detection cannot be done until POR is released. If the device is protected, Cold-Plugging is the only way to detect a debugger probe, and so the external reset timing must be longer than the POR timing. If external reset is deasserted before POR release, the user must retry the procedure above until it gets connected to the device.

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

14.7 Chip Erase

Chip-Erase consists of removing all sensitive information stored in the chip and clearing the NVMCTRL security bit. Therefore, all volatile memories and the Flash memory (including the EEPROM emulation area) will be erased. The Flash auxiliary rows, including the user row, will not be erased.

When the device is protected, the debugger must reset the device in order to be detected. This ensures that internal registers are reset after the protected state is removed. The Chip-Erase operation is triggered by writing a '1' to the Chip-Erase bit in the Control register (CTRL.CE). This command will be discarded if the DSU is protected by the Peripheral Access Controller (PAC). Once issued, the module clears volatile memories prior to erasing the Flash array. To ensure that the Chip-Erase operation is completed, check the Done bit of the Status A register (STATUSA.DONE).

The Chip-Erase operation depends on clocks and power management features that can be altered by the CPU. For that reason, it is recommended to issue a Chip- Erase after a Cold-Plugging procedure to ensure that the device is in a known and safe state.

The recommended sequence is as follows:

1. Issue the Cold-Plugging procedure (refer to Cold Plugging). The device then:
    1.1. Detects the debugger probe.
    1.2. Holds the CPU in reset.
2. Issue the Chip-Erase command by writing a '1' to CTRL.CE. The device then:
    2.1. Clears the system volatile memories.
    2.2. Erases the whole Flash array (including the EEPROM emulation area, not including auxiliary rows).
    2.3. Erases the lock row, removing the NVMCTRL security bit protection.
3. Check for completion by polling STATUSA.DONE (read as one when completed).
4. Reset the device to let the NVMCTRL update fuses.

14.8 Programming

Programming the Flash or RAM memories is only possible when the device is not protected by the NVMCTRL security bit. The programming procedure is as follows:

1. At power up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold. The system continues to be held in this static state until the internally regulated supplies have reached a safe operating state.

2. The PM starts, clocks are switched to the slow clock (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal resets are maintained due to the external reset.
3. The debugger maintains a low level on SWCLK. $\overline{\text{RESET}}$ is released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. The CPU remains in Reset due to the Cold-Plugging procedure; meanwhile, the rest of the system is released.
6. A Chip-Erase is issued to ensure that the Flash is fully erased prior to programming.
7. Programming is available through the AHB-AP.
8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$, toggling power, or writing a '1' to the Status A register CPU Reset Phase Extension bit (STATUSA.CRSTEXT). Make sure that the SWCLK pin is high when releasing $\overline{\text{RESET}}$ to prevent extending the CPU reset.

Related Links

Electrical Characteristics

NVMCTRL – Non-Volatile Memory Controller

Security Bit

14.9 Intellectual Property Protection

Intellectual property protection consists of restricting access to internal memories from external tools when the device is protected, and this is accomplished by setting the NVMCTRL security bit. This protected state can be removed by issuing a Chip-Erase (refer to Chip Erase). When the device is protected, read/write accesses using the AHB-AP are limited to the DSU address range and DSU commands are restricted. When issuing a Chip-Erase, sensitive information is erased from volatile memory and Flash.

The DSU implements a security filter that monitors the AHB transactions generated by the ARM AHB-AP inside the DAP. If the device is protected, then AHB-AP read/write accesses outside the DSU external address range are discarded, causing an error response that sets the ARM AHB-AP sticky error bits (refer to the ARM Debug Interface v5 Architecture Specification on http://www.arm.com).

The DSU is intended to be accessed either:

- Internally from the CPU, without any limitation, even when the device is protected
- Externally from a debug adapter, with some restrictions when the device is protected

For security reasons, DSU features have limitations when used from a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map have been replicated at offset 0x100:

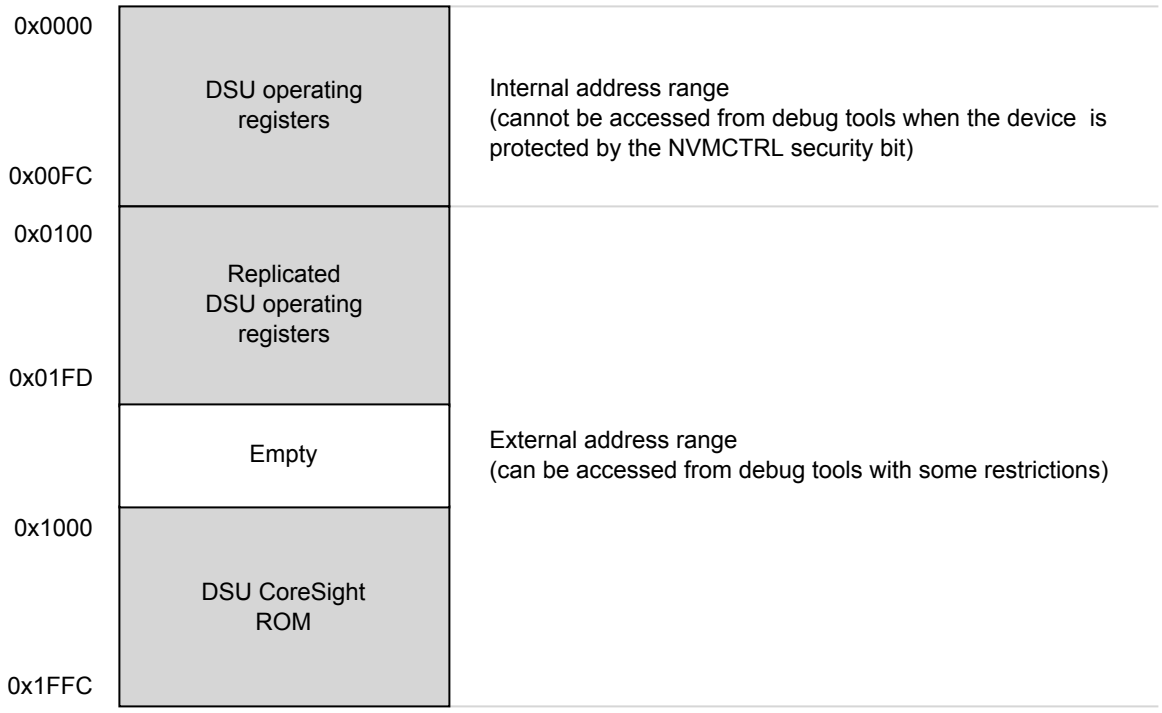
- The first 0x100 bytes form the internal address range
- The next 0x100 bytes form the external address range

When the device is protected, the DAP can only issue MEM-AP accesses in the DSU address range limited to the 0x100- 0x2000 offset range.

The DSU operating registers are located in the 0x00-0xFF area and remapped in 0x100-0x1FF to differentiate accesses coming from a debugger and the CPU. If the device is protected and an access is

issued in the region 0x100-0x1FF, it is subject to security restrictions. For more information, refer to the Table 14-1.

Figure 14-4. APB Memory Mapping



Some features not activated by APB transactions are not available when the device is protected:

Table 14-1. Feature Availability Under Protection

Features	Availability when the device is protected
CPU Reset Extension	Yes
Clear CPU Reset Extension	No
Debugger Cold-Plugging	Yes
Debugger Hot-Plugging	No

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

14.10 Device Identification

Device identification relies on the ARM CoreSight component identification scheme, which allows the chip to be identified as a SAM device implementing a DSU. The DSU contains identification registers to differentiate the device.

14.10.1 CoreSight Identification

A system-level ARM CoreSight ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the ARM Debug

Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

Figure 14-5. Conceptual 64-bit Peripheral ID

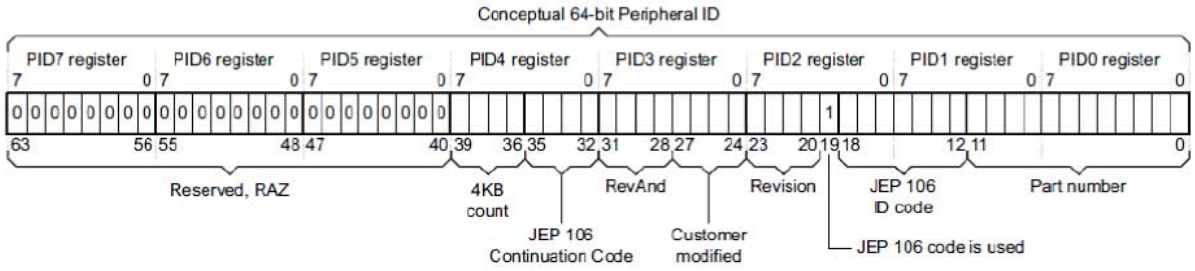


Table 14-2. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Continuation code: 0x0	PID4
JEP-106 ID code	7	Device ID: 0x1F	PID1+PID2
4KB count	4	Indicates that the CoreSight component is a ROM: 0x0	PID4
RevAnd	4	Not used; read as 0	PID3
CUSMOD	4	Not used; read as 0	PID3
PARTNUM	12	Contains 0xCD0 to indicate that DSU is present	PID0+PID1
REVISION	4	DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. If 0x0, this indicates that device identification can be completed by reading the Device Identification register (DID)	PID3

For more information, refer to the ARM Debug Interface Version 5 Architecture Specification.

14.10.2 Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Product family identification
- Product series identification
- Device select

14.11 Functional Description

14.11.1 Principle of Operation

The DSU provides memory services such as CRC32 or MBIST that require almost the same interface. Hence, the Address, Length and Data registers (ADDR, LENGTH, DATA) are shared. These shared registers must be configured first; then a command can be issued by writing the Control register. When a command is ongoing, other commands are discarded until the current operation is completed. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

14.11.2 Basic Operation

14.11.2.1 Initialization

The module is enabled by enabling its clocks. For more details, refer to Clocks. The DSU registers can be PAC write-protected.

Related Links

PAC - Peripheral Access Controller

14.11.2.2 Operation From a Debug Adapter

Debug adapters should access the DSU registers in the external address range 0x100 – 0x2000. If the device is protected by the NVMCTRL security bit, accessing the first 0x100 bytes causes the system to return an error. Refer to Intellectual Property Protection.

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

14.11.2.3 Operation From the CPU

There are no restrictions when accessing DSU registers from the CPU. However, the user should access DSU registers in the internal address range (0x0 – 0x100) to avoid external security restrictions. Refer to Intellectual Property Protection.

14.11.3 32-bit Cyclic Redundancy Check CRC32

The DSU unit provides support for calculating a cyclic redundancy check (CRC32) value for a memory area (including Flash and AHB RAM).

When the CRC32 command is issued from:

- The internal range, the CRC32 can be operated at any memory location
- The external range, the CRC32 operation is restricted; DATA, ADDR, and LENGTH values are forced (see below)

Table 14-3. AMOD Bit Descriptions when Operating CRC32

AMOD[1:0]	Short name	External range restrictions
0	ARRAY	CRC32 is restricted to the full Flash array area (EEPROM emulation area not included) DATA forced to 0xFFFFFFFF before calculation (no seed)
1	EEPROM	CRC32 of the whole EEPROM emulation area DATA forced to 0xFFFFFFFF before calculation (no seed)
2-3	Reserved	

The algorithm employed is the industry standard CRC32 algorithm using the generator polynomial 0xEDB88320 (reversed representation).

14.11.3.1 Starting CRC32 Calculation

CRC32 calculation for a memory range is started after writing the start address into the Address register (ADDR) and the size of the memory range into the Length register (LENGTH). Both must be word-aligned.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). This value will usually be 0xFFFFFFFF, but can be, for example, the result of a previous CRC32 calculation if generating a common CRC32 of separate memory blocks.

Once completed, the calculated CRC32 value can be read out of the Data register. The read value must be complemented to match standard CRC32 implementations or kept non-inverted if used as starting point for subsequent CRC32 calculations.

If the device is in protected state by the NVMCTRL security bit, it is only possible to calculate the CRC32 of the whole flash array when operated from the external address space. In most cases, this area will be the entire onboard non-volatile memory. The Address, Length and Data registers will be forced to predefined values once the CRC32 operation is started, and values written by the user are ignored. This allows the user to verify the contents of a protected device.

The actual test is started by writing a '1' in the 32-bit Cyclic Redundancy Check bit of the Control register (CTRL.CRC). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

14.11.3.2 Interpreting the Results

The user should monitor the Status A register. When the operation is completed, STATUSA.DONE is set. Then the Bus Error bit of the Status A register (STATUSA.BERR) must be read to ensure that no bus error occurred.

14.11.4 Debug Communication Channels

The Debug Communication Channels (DCCO and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger even if the device is protected by the NVMCTRL security bit. The registers can be used to exchange data between the CPU and the debugger, during run time as well as in debug mode. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are accessible when the protected state is active. When the device is protected, however, it is not possible to connect a debugger while the CPU is running (STATUSA.CRSTEXT is not writable and the CPU is held under Reset).

Two Debug Communication Channel status bits in the Status B registers (STATUS.DCCDx) indicate whether a new value has been written in DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the on-board memory testing logic (MBIST). Accordingly, DCC0 and DCC1 must not be used while performing MBIST operations.

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

14.11.5 Testing of On-Board Memories MBIST

The DSU implements a feature for automatic testing of memory also known as MBIST (memory built-in self test). This is primarily intended for production test of on-board memories. MBIST cannot be operated from the external address range when the device is protected by the NVMCTRL security bit. If an MBIST command is issued when the device is protected, a protection error is reported in the Protection Error bit in the Status A register (STATUSA.PERR).

1. Algorithm
   The algorithm used for testing is a type of March algorithm called "March LR". This algorithm is able to detect a wide range of memory defects, while still keeping a linear run time. The algorithm is:

1.1. Write entire memory to '0', in any order.
1.2. Bit for bit read '0', write '1', in descending order.
1.3. Bit for bit read '1', write '0', read '0', write '1', in ascending order.
1.4. Bit for bit read '1', write '0', in ascending order.
1.5. Bit for bit read '0', write '1', read '1', write '0', in ascending order.
1.6. Read '0' from entire memory, in ascending order.

The specific implementation used has a run time which depends on the CPU clock frequency and the number of bytes tested in the RAM. The detected faults are:

- Address decoder faults
- Stuck-at faults
- Transition faults
- Coupling faults
- Linked Coupling faults

2. Starting MBIST
To test a memory, you need to write the start address of the memory to the ADDR.ADDR bit field, and the size of the memory into the Length register.

For best test coverage, an entire physical memory block should be tested at once. It is possible to test only a subset of a memory, but the test coverage will then be somewhat lower.

The actual test is started by writing a '1' to CTRL.MBIST. A running MBIST operation can be canceled by writing a '1' to CTRL.SWRST.

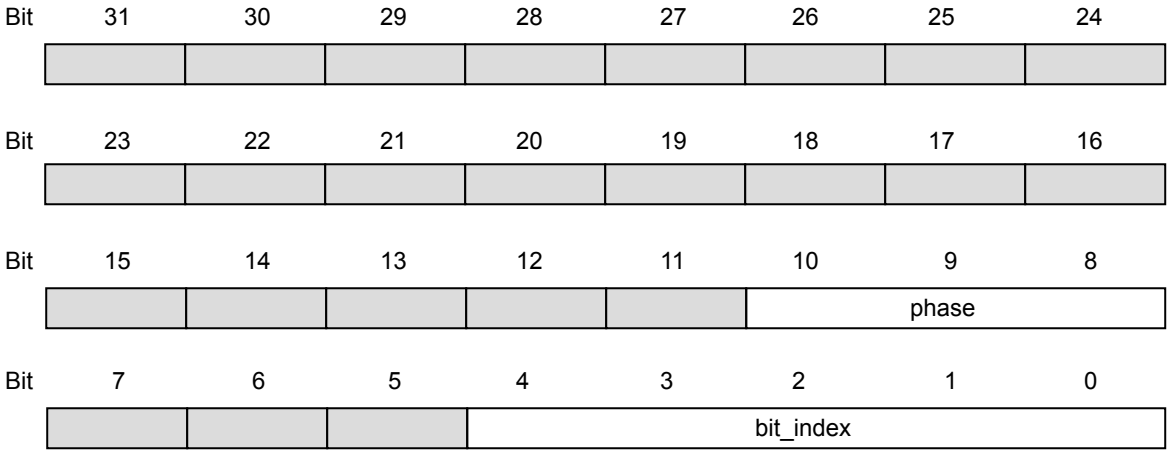
3. Interpreting the Results
The tester should monitor the STATUSA register. When the operation is completed, STATUSA.DONE is set. There are two different modes:

- ADDR.AMOD=0: exit-on-error (default)
  In this mode, the algorithm terminates either when a fault is detected or on successful completion. In both cases, STATUSA.DONE is set. If an error was detected, STATUSA.FAIL will be set. User then can read the DATA and ADDR registers to locate the fault.
- ADDR.AMOD=1: pause-on-error
  In this mode, the MBIST algorithm is paused when an error is detected. In such a situation, only STATUSA.FAIL is asserted. The state machine waits for user to clear STATUSA.FAIL by writing a '1' in STATUSA.FAIL to resume. Prior to resuming, user can read the DATA and ADDR registers to locate the fault.

4. Locating Faults
If the test stops with STATUSA.FAIL set, one or more bits failed the test. The test stops at the first detected error. The position of the failing bit can be found by reading the following registers:

- ADDR: Address of the word containing the failing bit
- DATA: contains data to identify which bit failed, and during which phase of the test it failed. The DATA register will in this case contains the following bit groups:

Figure 14-6. DATA bits Description When MBIST Operation Returns an Error



- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

Table 14-4. MBIST Operation Phases

Phase	Test actions
0	Write all bits to zero. This phase cannot fail.
1	Read '0', write '1', increment address
2	Read '1', write '0'
3	Read '0', write '1', decrement address
4	Read '1', write '0', decrement address
5	Read '0', write '1'
6	Read '1', write '0', decrement address
7	Read all zeros. bit_index is not used

Table 14-5. AMOD Bit Descriptions for MBIST

AMOD[1:0]	Description
0x0	Exit on Error
0x1	Pause on Error
0x2, 0x3	Reserved

Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

Product Mapping

14.11.6 System Services Availability when Accessed Externally

External access: Access performed in the DSU address offset 0x200-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x0-0x100 range.

Table 14-6. Available Features when Operated From The External Address Range and Device is Protected

Features	Availability From The External Address Range and Device is Protected
Chip-Erase command and status	Yes
CRC32	Yes, only full array or full EEPROM
CoreSight Compliant Device identification	Yes
Debug communication channels	Yes
Testing of onboard memories (MBIST)	No
STATUSA.CRSTEXT clearing	No (STATUSA.PERR is set when attempting to do so)

14.12 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0				CE	MBIST	CRC		SWRST
0x01	STATUSA	7:0				PERR	FAIL	BERR	CRSTEXT	DONE
0x02	STATUSB	7:0				HPE	DCCDx	DCCDx	DBGPRES	PROT
0x03	Reserved									
0x04	ADDR	7:0	ADDR[5:0]						AMOD[1:0]	
0x05		15:8	ADDR[13:6]							
0x06		23:16	ADDR[21:14]							
0x07		31:24	ADDR[29:22]							
0x08	LENGTH	7:0	LENGTH[5:0]							
0x09		15:8	LENGTH[13:6]							
0x0A		23:16	LENGTH[21:14]							
0x0B		31:24	LENGTH[29:22]							
0x0C	DATA	7:0	DATA[7:0]							
0x0D		15:8	DATA[15:8]							
0x0E		23:16	DATA[23:16]							
0x0F		31:24	DATA[31:24]							
0x10	DCC0	7:0	DATA[7:0]							
0x11		15:8	DATA[15:8]							
0x12		23:16	DATA[23:16]							
0x13		31:24	DATA[31:24]							
0x14	DCC1	7:0	DATA[7:0]							
0x15		15:8	DATA[15:8]							
0x16		23:16	DATA[23:16]							
0x17		31:24	DATA[31:24]							
0x18	DID	7:0	DEVSEL[7:0]							
0x19		15:8	DIE[3:0]				REVISION[3:0]			
0x1A		23:16	FAMILY[0:0]		SERIES[5:0]					
0x1B		31:24	PROCESSOR[3:0]				FAMILY[4:1]			
0x1C ... 0x0FFF	Reserved									
0x1000	ENTRY0	7:0							FMT	EPRES
0x1001		15:8	ADDOFF[3:0]							
0x1002		23:16	ADDOFF[11:4]							
0x1003		31:24	ADDOFF[19:12]							
0x1004	ENTRY1	7:0							FMT	EPRES
0x1005		15:8	ADDOFF[3:0]							
0x1006		23:16	ADDOFF[11:4]							
0x1007		31:24	ADDOFF[19:12]							
0x1008	END	7:0	END[7:0]							
0x1009		15:8	END[15:8]							
0x100A		23:16	END[23:16]							
0x100B		31:24	END[31:24]							
0x100C	Reserved									

Offset	Name	Bit Pos.								
... 0x1FCB										
0x1FCC	MEMTYPE	7:0								SMEMP
0x1FCD		15:8								
0x1FCE		23:16								
0x1FCF		31:24								
0x1FD0	PID4	7:0	FKBC[3:0]				JEPCC[3:0]			
0x1FD1		15:8								
0x1FD2		23:16								
0x1FD3		31:24								
0x1FD4 ... 0x1FDF	Reserved									
0x1FE0	PID0	7:0	PARTNBL[7:0]							
0x1FE1		15:8								
0x1FE2		23:16								
0x1FE3		31:24								
0x1FE4	PID1	7:0	JEPIDCL[3:0]				PARTNBH[3:0]			
0x1FE5		15:8								
0x1FE6		23:16								
0x1FE7		31:24								
0x1FE8	PID2	7:0	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
0x1FE9		15:8								
0x1FEA		23:16								
0x1FEB		31:24								
0x1FEC	PID3	7:0	REVAND[3:0]				CUSMOD[3:0]			
0x1FED		15:8								
0x1FEE		23:16								
0x1FEF		31:24								
0x1FF0	CID0	7:0	PREAMBLEB0[7:0]							
0x1FF1		15:8								
0x1FF2		23:16								
0x1FF3		31:24								
0x1FF4	CID1	7:0	CCLASS[3:0]				PREAMBLE[3:0]			
0x1FF5		15:8								
0x1FF6		23:16								
0x1FF7		31:24								
0x1FF8	CID2	7:0	PREAMBLEB2[7:0]							
0x1FF9		15:8								
0x1FFA		23:16								
0x1FFB		31:24								
0x1FFC	CID3	7:0	PREAMBLEB3[7:0]							
0x1FFD		15:8								
0x1FFE		23:16								
0x1FFF		31:24								

14.13 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

14.13.1 Control

Name: CTRL
Offset: 0x0000 [ID-00001c14]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				CE	MBIST	CRC		SWRST
Access				W	W	W		W
Reset				0	0	0		0

Bit 4 – CE: Chip Erase
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the Chip-Erase operation.

Bit 3 – MBIST: Memory Built-In Self-Test
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the memory BIST algorithm.

Bit 2 – CRC: 32-bit Cyclic Redundancy Check
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the cyclic redundancy check algorithm.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the module.

14.13.2 Status A

Name: STATUSA
Offset: 0x0001 [ID-00001c14]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				PERR	FAIL	BERR	CRSTEXT	DONE
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – PERR: Protection Error
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Protection Error bit.

This bit is set when a command that is not allowed in protected state is issued.

Bit 3 – FAIL: Failure
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Failure bit.

This bit is set when a DSU operation failure is detected.

Bit 2 – BERR: Bus Error
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Bus Error bit.

This bit is set when a bus error is detected.

Bit 1 – CRSTEXT: CPU Reset Phase Extension
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the CPU Reset Phase Extension bit.

This bit is set when a debug adapter Cold-Plugging is detected, which extends the CPU reset phase.

Bit 0 – DONE: Done
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Done bit.

This bit is set when a DSU operation is completed.

14.13.3 Status B

Name: STATUSB
Offset: 0x0002 [ID-00001c14]
Reset: 0x1X
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				HPE	DCCDx	DCCDx	DBGPRES	PROT
Access				R	R	R	R	R
Reset				1	0	0	x	x

Bit 4 – HPE: Hot-Plugging Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when Hot-Plugging is enabled.

This bit is cleared when Hot-Plugging is disabled. This is the case when the SWCLK function is changed. Only a power-reset or a external reset can set it again.

Bits 3,2 – DCCDx: Debug Communication Channel x Dirty [x=1..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when DCCx is written.

This bit is cleared when DCCx is read.

Bit 1 – DBGPRES: Debugger Present
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when a debugger probe is detected.

This bit is never cleared.

Bit 0 – PROT: Protected
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set at power-up when the device is protected.

This bit is never cleared.

14.13.4 Address

Name: ADDR
Offset: 0x0004 [ID-00001c14]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[5:0]						AMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:2 – ADDR[29:0]: Address
Initial word start address needed for memory operations.

Bits 1:0 – AMOD[1:0]: Address Mode
The functionality of these bits is dependent on the operation mode.

Bit description when operating CRC32: refer to 32-bit Cyclic Redundancy Check CRC32

Bit description when testing onboard memories (MBIST): refer to Testing of On-Board Memories MBIST

14.13.5 Length

Name: LENGTH
Offset: 0x0008 [ID-00001c14]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	LENGTH[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LENGTH[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LENGTH[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LENGTH[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – LENGTH[29:0]: Length
Length in words needed for memory operations.

14.13.6 Data

Name: DATA
Offset: 0x000C [ID-00001c14]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data
Memory operation initial value or result value.

14.13.7 Debug Communication Channel 0

Name: DCC0
Offset: 0x0010 [ID-00001c14]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data
Data register.

14.13.8 Debug Communication Channel 1

Name: DCC1
Offset: 0x0014 [ID-00001c14]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data
Data register.

14.13.9 Device Identification

The information in this register is related to the *Ordering Information*.

Name: DID
Offset: 0x0018 [ID-00001c14]
Reset: see related links
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	PROCESSOR[3:0]				FAMILY[4:1]			
Access	R	R	R	R	R	R	R	R
Reset	p	p	p	p	f	f	f	f

Bit	23	22	21	20	19	18	17	16
	FAMILY[0:0]		SERIES[5:0]					
Access	R		R	R	R	R	R	R
Reset	f		s	s	s	s	s	s

Bit	15	14	13	12	11	10	9	8
	DIE[3:0]				REVISION[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	d	d	d	d	r	r	r	r

Bit	7	6	5	4	3	2	1	0
	DEVSEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:28 – PROCESSOR[3:0]: Processor
The value of this field defines the processor used on the device.

Bits 27:23 – FAMILY[4:0]: Product Family
The value of this field corresponds to the Product Family part of the ordering code.

Bits 21:16 – SERIES[5:0]: Product Series
The value of this field corresponds to the Product Series part of the ordering code.

Bits 15:12 – DIE[3:0]: Die Number
Identifies the die family.

Bits 11:8 – REVISION[3:0]: Revision Number
Identifies the die revision number. 0x0=rev.A, 0x1=rev.B etc.

Note: The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

Bits 7:0 – DEVSEL[7:0]: Device Selection
This bit field identifies a device within a product family and product series. Refer to the Ordering Information for device configurations and corresponding values for Flash memory density, pin count and device variant.

14.13.10 CoreSight ROM Table Entry 0

Name: ENTRY0
Offset: 0x1000 [ID-00001c14]
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0]: Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT: Format
Always reads as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES: Entry Present
This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

14.13.11 CoreSight ROM Table Entry 1

Name: ENTRY1
Offset: 0x1004 [ID-00001c14]
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0]: Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT: Format
Always read as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES: Entry Present
This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

14.13.12 CoreSight ROM Table End

Name: END
Offset: 0x1008 [ID-00001c14]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	END[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	END[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	END[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	END[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – END[31:0]: End Marker
Indicates the end of the CoreSight ROM table entries.

14.13.13 CoreSight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC [ID-00001c14]
Reset: 0x0000000X
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SMEMP
Access								R
Reset								x

Bit 0 – SMEMP: System Memory Present
This bit indicates whether system memory is present on the bus that connects to the ROM table.

This bit is set at power-up if the device is not protected, indicating that the system memory is accessible from a debug adapter.

This bit is cleared at power-up if the device is protected, indicating that the system memory is not accessible from a debug adapter.

14.13.14 Peripheral Identification 4

Name: PID4
Offset: 0x1FD0 [ID-00001c14]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FKBC[3:0]				JEPCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – FKBC[3:0]: 4KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

Bits 3:0 – JEPCC[3:0]: JEP-106 Continuation Code
These bits will always return zero when read.

14.13.15 Peripheral Identification 0

Name: PID0
Offset: 0x1FE0 [ID-00001c14]
Reset: 0x000000D0
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PARTNBL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	1	0	0	0	0

Bits 7:0 – PARTNBL[7:0]: Part Number Low
These bits will always return 0xD0 when read, indicating that this device implements a DSU module instance.

14.13.16 Peripheral Identification 1

Name: PID1
Offset: 0x1FE4 [ID-00001c14]
Reset: 0x000000FC
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	JEPIDCL[3:0]				PARTNBH[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	0	0

Bits 7:4 – JEPIDCL[3:0]: Low part of the JEP-106 Identity Code
These bits will always return 0xF when read (JEP-106 identity code is 0x1F).

Bits 3:0 – PARTNBH[3:0]: Part Number High
These bits will always return 0xC when read, indicating that this device implements a DSU module instance.

14.13.17 Peripheral Identification 2

Name: PID2
Offset: 0x1FE8 [ID-00001c14]
Reset: 0x00000009
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bits 7:4 – REVISION[3:0]: Revision Number
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEPU: JEP-106 Identity Code is used
This bit will always return one when read, indicating that JEP-106 code is used.

Bits 2:0 – JEPIDCH[2:0]: JEP-106 Identity Code High
These bits will always return 0x1 when read, (JEP-106 identity code is 0x1F).

14.13.18 Peripheral Identification 3

Name: PID3
Offset: 0x1FEC [ID-00001c14]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVAND[3:0]				CUSMOD[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – REVAND[3:0]: Revision Number
These bits will always return 0x0 when read.

Bits 3:0 – CUSMOD[3:0]: ARM CUSMOD
These bits will always return 0x0 when read.

14.13.19 Component Identification 0

Name: CID0
Offset: 0x1FF0 [ID-00001c14]
Reset: 0x0000000D
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

Bits 7:0 – PREAMBLEB0[7:0]: Preamble Byte 0
These bits will always return 0x0D when read.

14.13.20 Component Identification 1

Name: CID1
Offset: 0x1FF4 [ID-00001c14]
Reset: 0x00000010
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCLASS[3:0]				PREAMBLE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 7:4 – CCLASS[3:0]: Component Class
These bits will always return 0x1 when read indicating that this ARM CoreSight component is ROM table (refer to the ARM Debug Interface v5 Architecture Specification at http://www.arm.com).

Bits 3:0 – PREAMBLE[3:0]: Preamble
These bits will always return 0x0 when read.

14.13.21 Component Identification 2

Name: CID2
Offset: 0x1FF8 [ID-00001c14]
Reset: 0x00000005
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	0	1

Bits 7:0 – PREAMBLEB2[7:0]: Preamble Byte 2
These bits will always return 0x05 when read.

14.13.22 Component Identification 3

Name: CID3
Offset: 0x1FFC [ID-00001c14]
Reset: 0x000000B1
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

Bits 7:0 – PREAMBLEB3[7:0]: Preamble Byte 3

These bits will always return 0xB1 when read.

15. Clock System

This chapter summarizes the clock distribution and terminology in the SAM L22 device. It will not explain every detail of its configuration. For in-depth documentation, see the respective peripherals descriptions and the *Generic Clock* documentation.

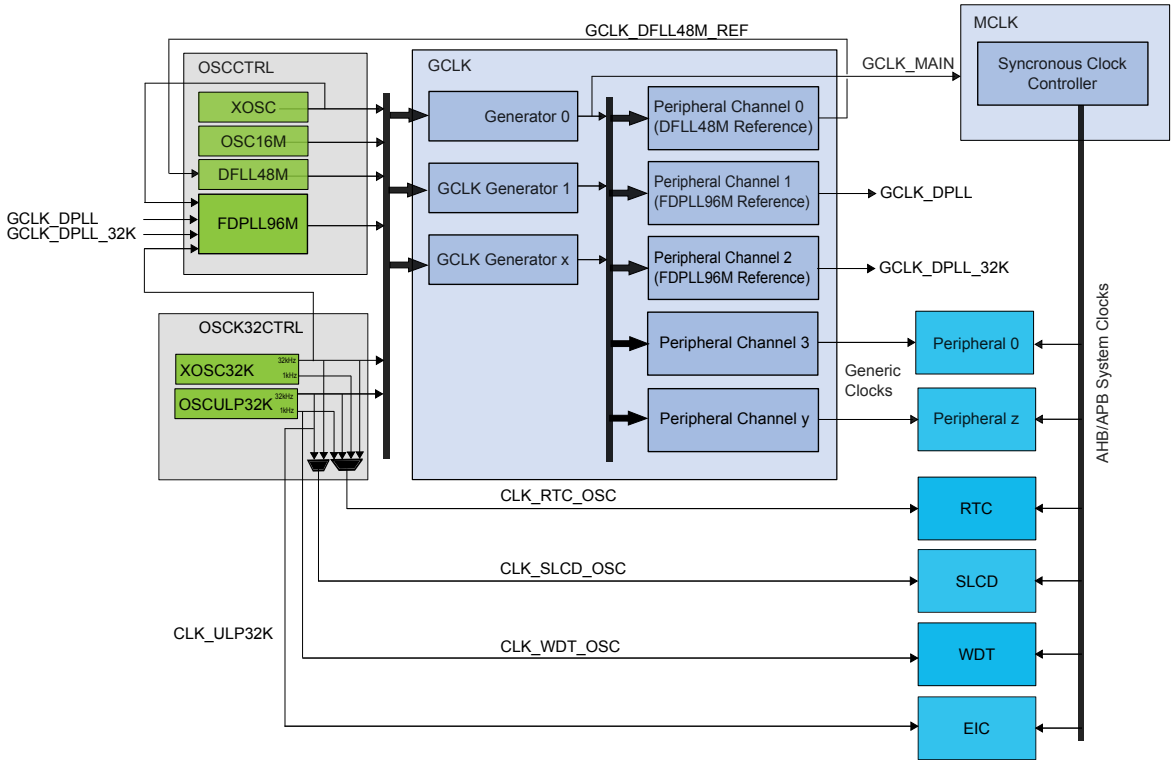
Related Links

MCLK – Main Clock

GCLK - Generic Clock Controller

15.1 Clock Distribution

Figure 15-1. Clock Distribution



The SAM L22 clock system consists of:

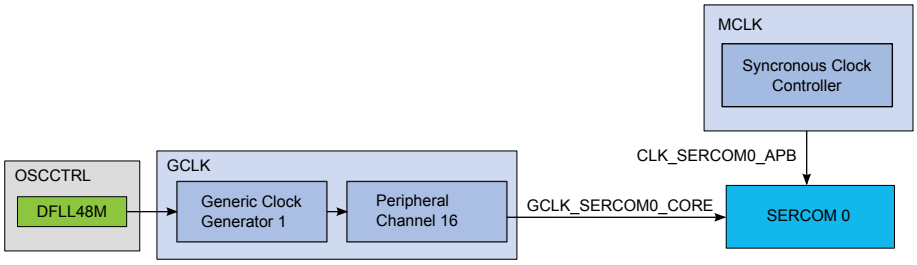
- *Clock sources*, i.e. oscillators controlled by OSCCTRL and OSC32KCTRL
  - A clock source provides a time base that is used by other components, such as Generic Clock Generators. Example clock sources are the internal 16MHz oscillator (OSC16M), external crystal oscillator (XOSC) and the Digital Frequency Locked Loop (DFLL48M).
- *Generic Clock Controller (GCLK),* which generates, controls and distributes the asynchronous clock consisting of:
  - *Generic Clock Generators:* These are programmable prescalers that can use any of the system clock sources as a time base. The Generic Clock Generator 0 generates the clock signal GCLK_MAIN, which is used by the Power Manager and the Main Clock (MCLK) module, which in turn generates synchronous clocks.
  - *Generic Clocks:* These are clock signals generated by Generic Clock Generators and output by the Peripheral Channels, and serve as clocks for the peripherals of the system. Multiple

instances of a peripheral will typically have a separate Generic Clock for each instance. Generic Clock 0 serves as the clock source for the DFLL48M clock input (when multiplying another clock source).

- *Main Clock Controller (MCLK)*
  - The MCLK generates and controls the synchronous clocks on the system. This includes the CPU, bus clocks (APB, AHB) as well as the synchronous (to the CPU) user interfaces of the peripherals. It contains clock masks that can turn on/off the user interface of a peripheral as well as prescalers for the CPU and bus clocks.

The next figure shows an example where SERCOM0 is clocked by the DFLL48M in open loop mode. The DFLL48M is enabled, the Generic Clock Generator 1 uses the DFLL48M as its clock source and feeds into Peripheral Channel 16. The Generic Clock 16, also called GCLK_SERCOM0_CORE, is connected to SERCOM0. The SERCOM0 interface, clocked by CLK_SERCOM0_APB, has been unmasked in the APBC Mask register in the MCLK.

Figure 15-2. Example of SERCOM Clock



To customize the clock distribution, refer to these registers and bit fields:

- The source oscillator for a generic clock generator n is selected by writing to the Source bit field in the Generator Control n register (GCLK.GENCTRLn.SRC).
- A Peripheral Channel m can be configured to use a specific Generic Clock Generator by writing to the Generic Clock Generator bit field in the respective Peripheral Channel m register (GCLK.PCHCTRLm.GEN)
- The Peripheral Channel number, *m*, is fixed for a given peripheral. See the Mapping table in the description of GCLK.PCHCTRLm.
- The AHB clocks are enabled and disabled by writing to the respective bit in the AHB Mask register (MCLK.AHBMASK).
- The APB clocks are enabled and disabled by writing to the respective bit in the APB x Mask registers (MCLK.APBxMASK).

15.2 Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, i.e. they are clocked from different clock sources and/or with different clock speeds, some peripheral accesses by the CPU need to be synchronized. In this case the peripheral includes a Synchronization Busy (SYNCBUSY) register that can be used to check if a sync operation is in progress.

For a general description, see Register Synchronization. Some peripherals have specific properties described in their individual sub-chapter “Synchronization”.

In the datasheet, references to Synchronous Clocks are referring to the CPU and bus clocks (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).

Related Links
Synchronization

15.3 Register Synchronization

15.3.1 Overview

All peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

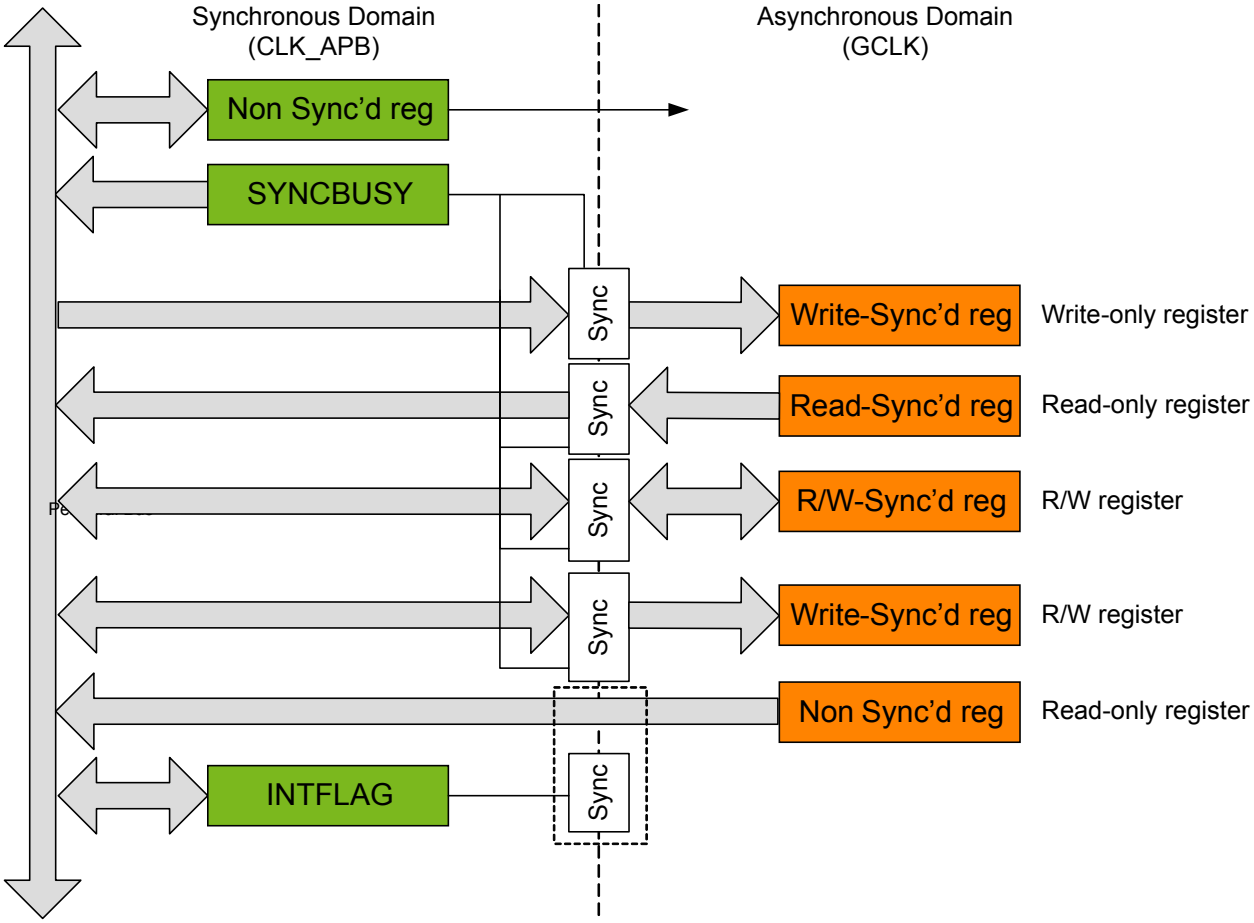
All registers in the bus interface are accessible without synchronization.

All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

Each individual register description will have the properties "Read-Synchronized" and/or "Write-Synchronized" if a register is synchronized.

As shown in the figure below, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).
Note: For registers requiring both read- and write-synchronization, the corresponding bit in SYNCBUSY is shared.

Figure 15-3. Register Synchronization Overview



15.3.2 General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain. The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer to Synchronization Delay for details on the synchronization delay.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported.

Example:

REGA, REGB are 8-bit core registers. REGC is a 16-bit core register.

Offset	Register
0x00	REGA
0x01	REGB
0x02	REGC
0x03	

Synchronization is per register, so multiple registers can be synchronized in parallel. Consequently, after REGA (8-bit access) was written, REGB (8-bit access) can be written immediately without error.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

Related Links

PAC - Peripheral Access Controller

15.3.3 General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.

Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.

Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

15.3.4 Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

15.3.5 Enable Write Synchronization

Setting the Enable bit in a module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronisation Ready interrupt (if available) cannot be used to enable write-synchronization.

15.3.6 Software Reset Write-Synchronization

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a ‘1’ to the CTRLA.SWRST bit it will immediately read as ‘1’.

CTRL.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRL.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

15.3.7 Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay *D* is within the range of:

$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$

Where P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

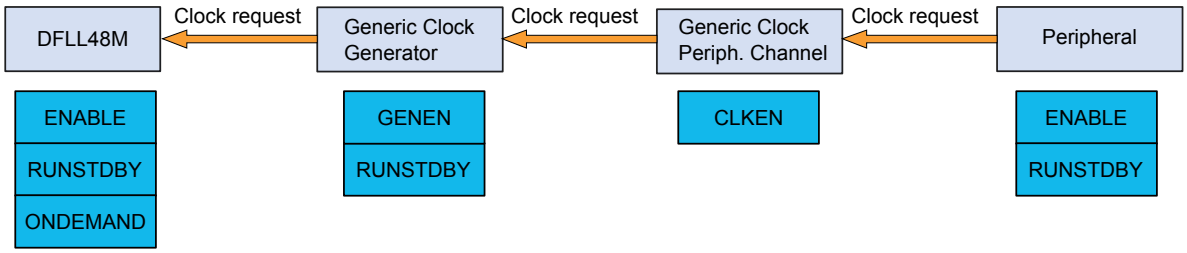
15.4 Enabling a Peripheral

In order to enable a peripheral that is clocked by a Generic Clock, the following parts of the system needs to be configured:

- A running Clock Source
- A clock from the Generic Clock Generator must be configured to use one of the running Clock Sources, and the Generator must be enabled.
- The Peripheral Channel that provides the Generic Clock signal to the peripheral must be configured to use a running Generic Clock Generator, and the Generic Clock must be enabled.
- The user interface of the peripheral needs to be unmasked in the PM. If this is not done the peripheral registers will read all 0's and any writing attempts to the peripheral will be discarded.

15.5 On Demand Clock Requests

Figure 15-4. Clock Request Routing



All clock sources in the system can be run in an on-demand mode: the clock source is in a stopped state unless a peripheral is requesting the clock source. Clock requests propagate from the peripheral, via the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started/kept running. As soon as the clock source is no longer needed and no peripheral has an active request, the clock source will be stopped until requested again.

The clock request can reach the clock source only if the peripheral, the generic clock and the clock from the Generic Clock Generator in-between are enabled. The time taken from a clock request being asserted to the clock source being ready is dependent on the clock source startup time, clock source frequency as well as the divider used in the Generic Clock Generator. The total startup time T_{start} from a clock request until the clock is available for the peripheral is between:

T_{start_max} = Clock source startup time + 2 × clock source periods + 2 × divided clock source periods

T_{start_min} = Clock source startup time + 1 × clock source period + 1 × divided clock source period

The time between the last active clock request stopped and the clock is shut down, T_{stop}, is between:

T_{stop_min} = 1 × divided clock source period + 1 × clock source period

T_{stop_max} = 2 × divided clock source periods + 2 × clock source periods

The On-Demand function can be disabled individually for each clock source by clearing the ONDEMAND bit located in each clock source controller. Consequently, the clock will always run whatever the clock request status is. This has the effect of removing the clock source startup time at the cost of power consumption.

The clock request mechanism can be configured to work in standby mode by setting the RUNSDTBY bits of the modules, see Figure 15-4.

15.6 Power Consumption vs. Speed

When targeting for either a low-power or a fast acting system, some considerations have to be taken into account due to the nature of the asynchronous clocking of the peripherals:

If clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent on the peripheral clock speed, and will take longer with a slower peripheral clock. This will cause worse response times and longer synchronization delays.

15.7 Clocks after Reset

On any Reset the synchronous clocks start to their initial state:

- OSC16M is enabled and configured to run at 4MHz
- Generic Generator 0 uses OSC16M as source and generates GCLK_MAIN
- CPU and BUS clocks are undivided

On a Power-on Reset, the 32KHz clock sources are reset and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except
  - Generator 0 is using OSC16M at 4MHz as source and generates GCLK_MAIN
- All Peripheral Channels in GCLK are disabled.

On a User Reset the GCLK module starts to its initial state, except for:

- Generic Clocks that are write-locked, i.e., the according WRTLOCK is set to 1 prior to Reset

Related Links

RSTC – Reset Controller

16. GCLK - Generic Clock Controller

16.1 Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller GCLK provides five Generic Clock Generators [4:0] that can provide a wide range of clock frequencies.

Generators can be set to use different external and internal oscillators as source. The clock of each Generator can be divided. The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules, as shown in Figure 16-2. The number of Peripheral Clocks depends on how many peripherals the device has.
Note: The Generator 0 is always the direct source of the GCLK_MAIN signal.

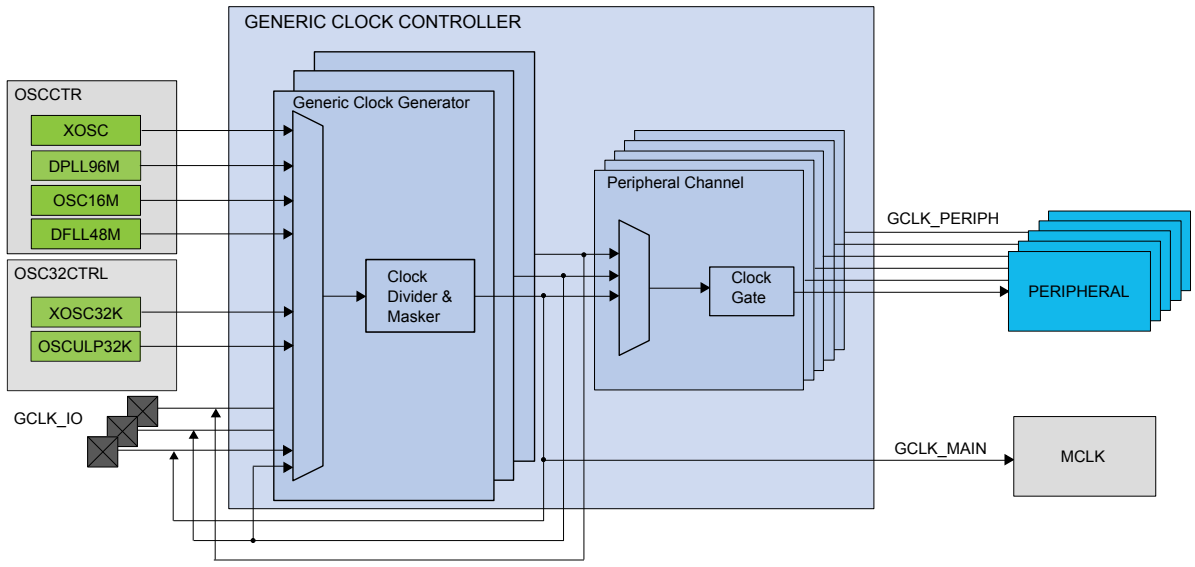
16.2 Features

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range

16.3 Block Diagram

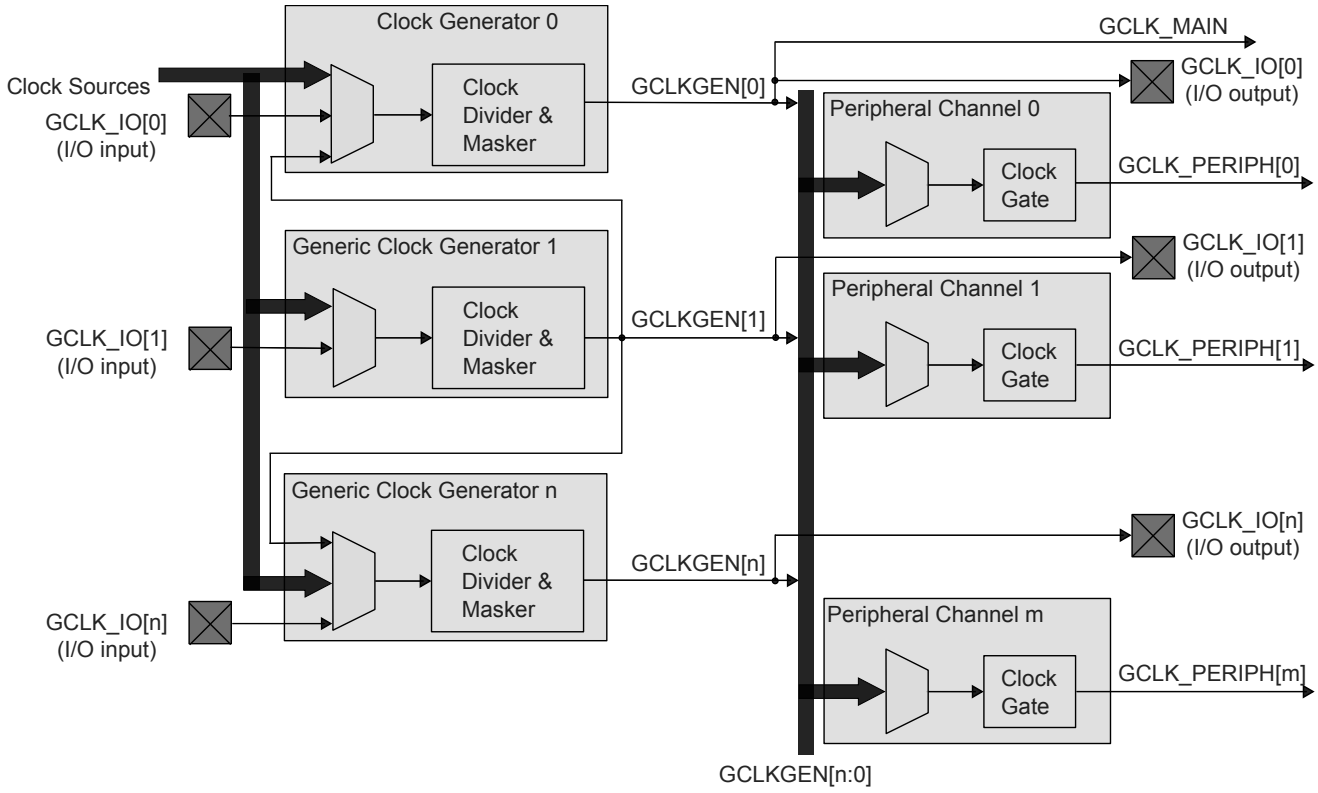
The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in Device Clocking Diagram.

Figure 16-1. Device Clocking Diagram



The GCLK block diagram is shown below:

Figure 16-2. Generic Clock Controller Block Diagram



16.4 Signal Description

Table 16-1. GCLK Signal Description

Signal Name	Type	Description
GCLK_IO[7:0]	Digital I/O	Clock source for Generators when input Generic Clock signal when output

Note: One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

16.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

16.5.1 I/O Lines

Using the GCLK I/O lines requires the I/O pins to be configured.

Related Links

PORT - I/O Pin Controller

16.5.2 Power Management

The GCLK can operate in all sleep modes, if required.

Related Links

PM – Power Manager

16.5.3 Clocks

The GCLK bus clock (CLK_GCLK_APB) can be enabled and disabled in the Main Clock Controller.

Related Links

Peripheral Clock Masking

OSC32KCTRL – 32KHz Oscillators Controller

16.5.4 DMA

Not applicable.

16.5.5 Interrupts

Not applicable.

16.5.6 Events

Not applicable.

16.5.7 Debug Operation

When the CPU is halted in debug mode the GCLK continues normal operation. If the GCLK is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

16.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

16.5.9 Analog Connections

Not applicable.

16.6 Functional Description

16.6.1 Principle of Operation

The GCLK module is comprised of five Generic Clock Generators (Generators) sourcing up to 64 Peripheral Channels and the Main Clock signal GCLK_MAIN.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH) to the peripherals.

16.6.2 Basic Operation

16.6.2.1 Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

1. The Generator must be enabled (GENCTRLn.GENEN=1) and the division factor must be set (GENTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).
2. The Generic Clock for a peripheral must be configured by writing to the respective Peripheral Channel Control register (PCHCTRLm). The Generator used as the source for the Peripheral Clock must be written to the GEN bit field in the Peripheral Channel Control register (PCHCTRLm.GEN).

Note: Each Generator n is configured by one dedicated register GENCTRLn.

Note: Each Peripheral Channel m is configured by one dedicated register PCHCTRLm.

16.6.2.2 Enabling, Disabling, and Resetting

The GCLK module has no enable/disable bit to enable or disable the whole module.

The GCLK is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST) to 1. All registers in the GCLK will be reset to their initial state, except for Peripheral Channels and associated Generators that have their Write Lock bit set to 1 (PCHCTRLm.WRTLOCK). For further details, refer to Configuration Lock.

16.6.2.3 Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of nine different clock sources except GCLK_GEN[1], which can be set to run from one of eight sources. GCLK_GEN[1] is the only Generator that can be selected as source to others Generators.

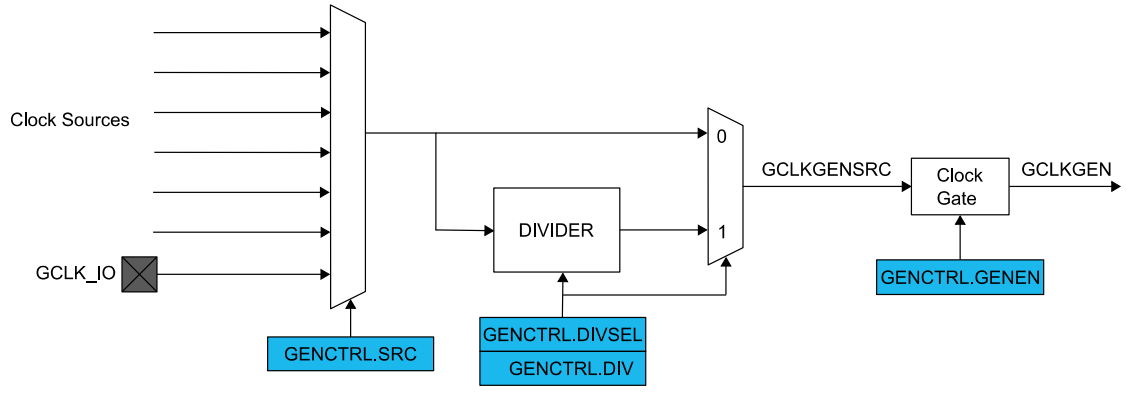
Each generator GCLK_GEN[x] can be connected to one specific pin (GCLK_IO[y]). The GCLK_IO[y] can be set to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as clock source for Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN[0] is used as GCLK_MAIN for the synchronous clock controller inside the Main Clock Controller. Refer to the Main Clock Controller description for details on the synchronous clock generation.

Figure 16-3. Generic Clock Generator



Related Links

MCLK – Main Clock

16.6.2.4 Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

16.6.2.5 Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GEN[n] clock is disabled and gated.

16.6.2.6 Selecting a Clock Source for the Generator

Each Generator can individually select a clock source by setting the Source Select bit group in the Generator Control register (GENCTRLn.SRC).

Changing from one clock source, for example A, to another clock source, B, can be done on the fly: If clock source B is not ready, the Generator will continue using clock source A. As soon as source B is ready, the Generator will switch to it. During the switching operation, the Generator maintains clock requests to both clock sources A and B, and will release source A as soon as the switch is done. The according bit in SYNCBUSY register (SYNCBUSY.GENCTRLn) will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, DPLL, and DFLL). Only Generator 1 can be used as a common source for all other generators.

16.6.2.7 Changing the Clock Frequency

The selected source for a Generator can be divided by writing a division value in the Division Factor bit field of the Generator Control register (GENCTRLn.DIV). How the actual division factor is calculated is depending on the Divide Selection bit (GENCTRLn.DIVSEL).

If GENCTRLn.DIVSEL=0 and GENCTRLn.DIV is either 0 or 1, the output clock will be undivided.

Note: The number of DIV bits for each Generator is device dependent.

16.6.2.8 Duty Cycle

When dividing a clock with an odd division factor, the duty-cycle will not be 50/50. Setting the Improve Duty Cycle bit of the Generator Control register (GENCTRLn.IDC) will result in a 50/50 duty cycle.

16.6.2.9 External Clock

The output clock (GCLK_GEN) of each Generator can be sent to I/O pins (GCLK_IO).

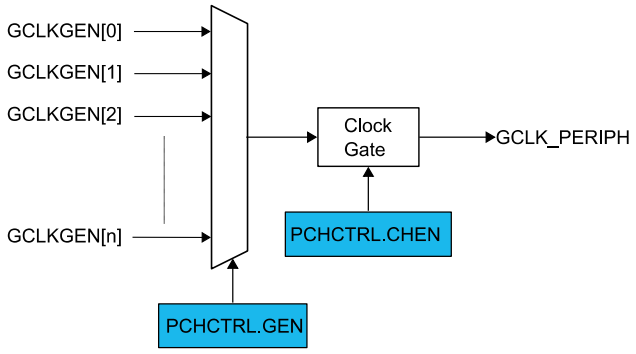
If the Output Enable bit in the Generator Control register is set (GENCTRLn.OE = 1) and the generator is enabled (GENCTRLn.GENEN=1), the Generator requests its clock source and the GCLK_GEN clock is output to an I/O pin.

If GENCTRLn.OE is 0, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV: If GENCTRLn.OOV is '0', the output clock will be low when turned off. If this bit is '1', the output clock will be high when turned off.

In Standby mode, if the clock is output (GENCTRLn.OE=1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero. If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.

16.6.3 Peripheral Clock

Figure 16-4. Peripheral Clock



16.6.3.1 Enabling a Peripheral Clock

Before a Peripheral Clock is enabled, one of the Generators must be enabled (GENCTRLn.GENEN) and selected as source for the Peripheral Channel by setting the Generator Selection bits in the Peripheral Channel Control register (PCHCTRL.GEN). Any available Generator can be selected as clock source for each Peripheral Channel.

When a Generator has been selected, the peripheral clock is enabled by setting the Channel Enable bit in the Peripheral Channel Control register, PCHCTRLm.CHEN = 1. The PCHCTRLm.CHEN bit must be synchronized to the generic clock domain. PCHCTRLm.CHEN will continue to read as its previous state until the synchronization is complete.

16.6.3.2 Disabling a Peripheral Clock

A Peripheral Clock is disabled by writing PCHCTRLm.CHEN=0. The PCHCTRLm.CHEN bit must be synchronized to the Generic Clock domain. PCHCTRLm.CHEN will stay in its previous state until the synchronization is complete. The Peripheral Clock is gated when disabled.

Related Links

PCHCTRLm

16.6.3.3 Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to PCHCTRLm.GEN, the peripheral clock must be disabled before re-enabling it with the new clock source setting. This prevents glitches during the transition:

1. Disable the Peripheral Channel by writing PCHCTRLm.CHEN=0
2. Assert that PCHCTRLm.CHEN reads '0'
3. Change the source of the Peripheral Channel by writing PCHCTRLm.GEN
4. Re-enable the Peripheral Channel by writing PCHCTRLm.CHEN=1

Related Links

PCHCTRLm

16.6.3.4 Configuration Lock

The peripheral clock configuration can be locked for further write accesses by setting the Write Lock bit in the Peripheral Channel Control register PCHCTRLm.WRTLOCK=1). All writing to the PCHCTRLm register will be ignored. It can only be unlocked by a Power Reset.

The Generator source of a locked Peripheral Channel will be locked, too: The corresponding GENCTRLn register is locked, and can be unlocked only by a Power Reset.

There is one exception concerning the Generator 0. As it is used as GCLK_MAIN, it cannot be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRLA.SWRST) can not unlock the registers.

In case of an external Reset, the Generator source will be disabled. Even if the WRTLOCK bit is written to '1' the peripheral channels are disabled (PCHCTRLm.CHEN set to '0') until the Generator source is enabled again. Then, the PCHCTRLm.CHEN are set to '1' again.

Related Links

CTRLA

PCHCTRLm

16.6.4 Additional Features

16.6.4.1 Peripheral Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a Reset. That means that the configuration of the Generators and Peripheral Channels after Reset is device-dependent.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to PCHCTRLm.SRC for details on PCHCTRLm reset.

16.6.5 Sleep Mode Operation

16.6.5.1 SleepWalking

The GCLK module supports the SleepWalking feature.

If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and Peripheral Channel stages successively, and delivers the clock to the peripheral.

The RUNSTDBY bit in the Generator Control register controls clock output to pin during standby sleep mode. If the bit is cleared, the Generator output is not available on pin. When set, the GCLK can continuously output the generator output to GCLK_IO. Refer to External Clock for details.

Related Links

PM – Power Manager

16.6.5.2 Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby Mode, minimizing the power consumption:

Table 16-2. Clock Generator n Activity in Standby Mode

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
yes	-	-	active
no	1	1	active
no	1	0	OFF

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
no	0	1	OFF
no	0	0	OFF

16.6.5.3 Entering Standby Mode

There may occur a delay when the device is put into Standby, until the power is turned off. This delay is caused by running Clock Generators: if the Run in Standby bit in the Generator Control register (GENCTRLn.RUNSTDBY) is '0', GCLK must verify that the clock is turned of properly. The duration of this verification is frequency-dependent.

Related Links

PM – Power Manager

16.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- Generic Clock Generator Control register (GENCTRLn)
- Control A register (CTRLA)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

CTRLA

PCHCTRLm

16.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SYNCBUSY	7:0		GENCTRLx	GENCTRLx	GENCTRLx	GENCTRLx	GENCTRLx		SWRST
0x05		15:8								
0x06		23:16								
0x07		31:24								
0x08 ... 0x1F	Reserved									
0x20	GENCTRLn0	7:0					SRC[3:0]			
0x21		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
0x22		23:16	DIV[7:0]							
0x23		31:24	DIV[15:8]							
0x24	GENCTRLn1	7:0					SRC[3:0]			
0x25		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
0x26		23:16	DIV[7:0]							
0x27		31:24	DIV[15:8]							
0x28	GENCTRLn2	7:0					SRC[3:0]			
0x29		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
0x2A		23:16	DIV[7:0]							
0x2B		31:24	DIV[15:8]							
0x2C	GENCTRLn3	7:0					SRC[3:0]			
0x2D		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
0x2E		23:16	DIV[7:0]							
0x2F		31:24	DIV[15:8]							
0x30	GENCTRLn4	7:0					SRC[3:0]			
0x31		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
0x32		23:16	DIV[7:0]							
0x33		31:24	DIV[15:8]							
0x34 ... 0x7F	Reserved									
0x80	PCHCTRLm0	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x81		15:8								
0x82		23:16								
0x83		31:24								
0x84	PCHCTRLm1	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x85		15:8								
0x86		23:16								
0x87		31:24								
0x88	PCHCTRLm2	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x89		15:8								

Offset	Name	Bit Pos.								
0x8A		23:16								
0x8B		31:24								
0x8C	PCHCTRLm3	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x8D		15:8								
0x8E		23:16								
0x8F		31:24								
0x90	PCHCTRLm4	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x91		15:8								
0x92		23:16								
0x93		31:24								
0x94	PCHCTRLm5	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x95		15:8								
0x96		23:16								
0x97		31:24								
0x98	PCHCTRLm6	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x99		15:8								
0x9A		23:16								
0x9B		31:24								
0x9C	PCHCTRLm7	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x9D		15:8								
0x9E		23:16								
0x9F		31:24								
0xA0	PCHCTRLm8	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xA1		15:8								
0xA2		23:16								
0xA3		31:24								
0xA4	PCHCTRLm9	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xA5		15:8								
0xA6		23:16								
0xA7		31:24								
0xA8	PCHCTRLm10	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xA9		15:8								
0xAA		23:16								
0xAB		31:24								
0xAC	PCHCTRLm11	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xAD		15:8								
0xAE		23:16								
0xAF		31:24								
0xB0	PCHCTRLm12	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xB1		15:8								
0xB2		23:16								
0xB3		31:24								
0xB4	PCHCTRLm13	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xB5		15:8								
0xB6		23:16								
0xB7		31:24								
0xB8	PCHCTRLm14	7:0	WRTLOCK	CHEN			GEN[3:0]			

Offset	Name	Bit Pos.								
0xB9		15:8								
0xBA		23:16								
0xBB		31:24								
0xBC	PCHCTRLm15	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xBD		15:8								
0xBE		23:16								
0xBF		31:24								
0xC0	PCHCTRLm16	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xC1		15:8								
0xC2		23:16								
0xC3		31:24								
0xC4	PCHCTRLm17	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xC5		15:8								
0xC6		23:16								
0xC7		31:24								
0xC8	PCHCTRLm18	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xC9		15:8								
0xCA		23:16								
0xCB		31:24								
0xCC	PCHCTRLm19	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xCD		15:8								
0xCE		23:16								
0xCF		31:24								
0xD0	PCHCTRLm20	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xD1		15:8								
0xD2		23:16								
0xD3		31:24								
0xD4	PCHCTRLm21	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xD5		15:8								
0xD6		23:16								
0xD7		31:24								
0xD8	PCHCTRLm22	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xD9		15:8								
0xDA		23:16								
0xDB		31:24								
0xDC	PCHCTRLm23	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xDD		15:8								
0xDE		23:16								
0xDF		31:24								
0xE0	PCHCTRLm24	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xE1		15:8								
0xE2		23:16								
0xE3		31:24								
0xE4	PCHCTRLm25	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xE5		15:8								
0xE6		23:16								
0xE7		31:24								

Offset	Name	Bit Pos.								
0xE8	PCHCTRLm26	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xE9		15:8								
0xEA		23:16								
0xEB		31:24								
0xEC	PCHCTRLm27	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xED		15:8								
0xEE		23:16								
0xEF		31:24								
0xF0	PCHCTRLm28	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xF1		15:8								
0xF2		23:16								
0xF3		31:24								

16.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

16.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-000008c2]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								R/W
Reset								0

Bit 0 – SWRST: Software Reset
Writing a zero to this bit has no effect.

Setting this bit to 1 will reset all registers in the GCLK to their initial state after a Power Reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.

Refer to GENCTRL Reset Value for details on GENCTRL register reset.

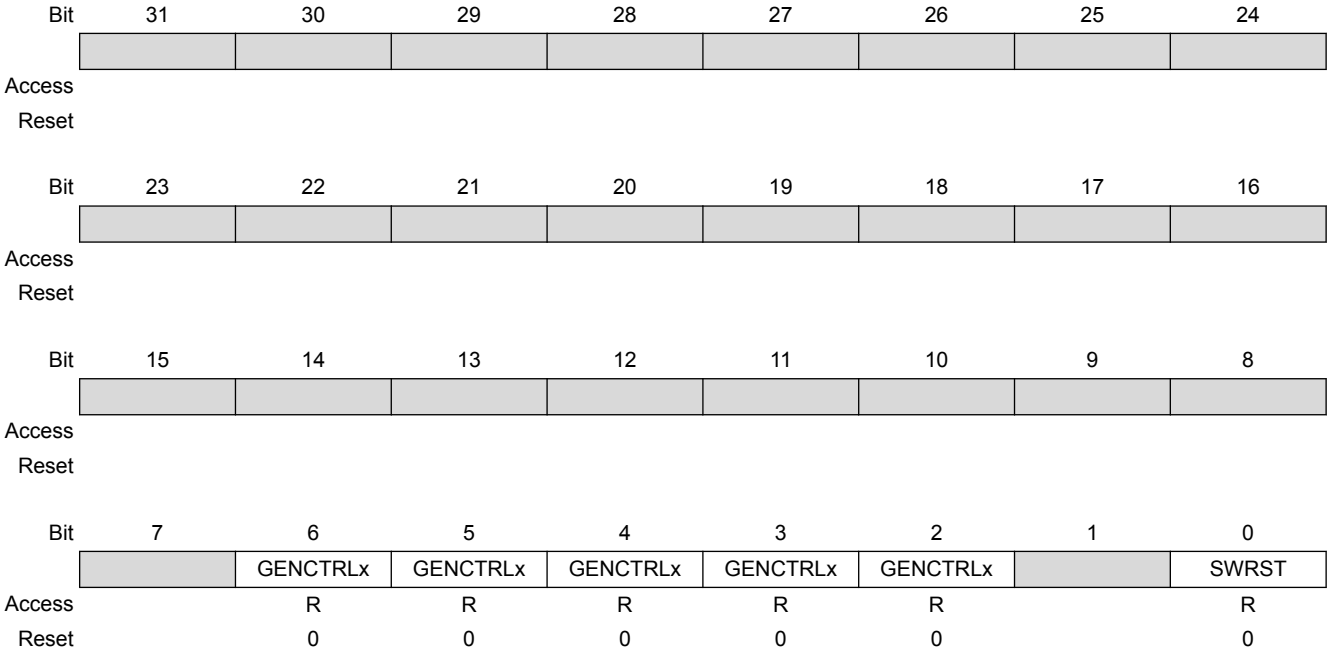
Refer to PCHCTRL Reset Value for details on PCHCTRL register reset.

Due to synchronization, there is a waiting period between setting CTRLA.SWRST and a completed Reset. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

16.8.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04 [ID-000008c2]
Reset: 0x00000000
Property: –



Bits 2,3,4,5,6 – GENCTRLx: Generator Control x Synchronization Busy
This bit is cleared when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is complete, or when clock switching operation is complete.

This bit is set when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST register bit between clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST register bit between clock domains is started.

16.8.3 Generator Control

GENCTRLn controls the settings of Generic Generator n (n=0..4).

Name: GENCTRLn
Offset: 0x20 + n*0x04 [n=0..4]
Reset: 0x00010005 for Generator n=0, else 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	DIV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					SRC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:16 – DIV[15:0]: Division Factor
These bits represent a division value for the corresponding Generator. The actual division factor is dependent on the state of DIVSEL. The number of relevant DIV bits for each Generator can be seen in this table. Written bits outside of the specified range will be ignored.

Table 16-3. Division Factor Bits

Generic Clock Generator	Division Factor Bits
Generator 0	8 division factor bits - DIV[7:0]
Generator 1	16 division factor bits - DIV[15:0]
Generator 2 - 4	8 division factor bits - DIV[7:0]

Bit 13 – RUNSTDBY: Run in Standby
This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IO pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

Value	Description
0	The Generator is stopped in Standby and the GCLK_IO pin state (one or zero) will be dependent on the setting in GENCTRL.OOV.
1	The Generator is kept running and output to its dedicated GCLK_IO pin during Standby mode.

Bit 12 – DIVSEL: Divide Selection
This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either 0 or 1.

Value	Description
0	The Generator clock frequency equals the clock source frequency divided by GENCTRLn.DIV.
1	The Generator clock frequency equals the clock source frequency divided by 2^(GENCTRLn.DIV+1).

Bit 11 – OE: Output Enable
This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO), as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	No Generator clock signal on pin GCLK_IO.
1	The Generator clock signal is output on the corresponding GCLK_IO, unless GCLK_IO is selected as a generator source in the GENCTRLn.SRC bit field.

Bit 10 – OOV: Output Off Value
This bit is used to control the clock output value on pin (GCLK_IO) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	The GCLK_IO will be LOW when generator is turned off or when the OE bit is zero.
1	The GCLK_IO will be HIGH when generator is turned off or when the OE bit is zero.

Bit 9 – IDC: Improve Duty Cycle
This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

Value	Description
0	Generator output clock duty cycle is not balanced to 50/50 for odd division factors.
1	Generator output clock duty cycle is 50/50.

Bit 8 – GENEN: Generator Enable
This bit is used to enable and disable the Generator.

Value	Description
0	Generator is disabled.
1	Generator is enabled.

Bits 3:0 – SRC[3:0]: Generator Clock Source Selection
These bits select the Generator clock source, as shown in this table.

Table 16-4. Generator Clock Source Selection

Value	Name	Description
0x0	XOSC	XOSC oscillator output
0x1	GCLK_IN	Generator input pad (GCLK_IO)
0x2	GCLK_GEN1	Generic clock generator 1 output

Value	Name	Description
0x03	OSCULP32K	OSCULP32K oscillator output
0x4	XOSC32K	XOSC32K oscillator output
0x5	OSC16M	OSC16M oscillator output
0x6	DFLL48M	DFLL48M output
0x7	DPLL96M	DPLL96M output
0x8-0xF	Reserved	Reserved for future use

A Power Reset will reset all GENCTRLn registers. the Reset values of the GENCTRLn registers are shown in table below.

Table 16-5. GENCTRLn Reset Value after a Power Reset

GCLK Generator	Reset Value after a Power Reset
0	0x00010005
others	0x00000000

A User Reset will reset the associated GENCTRL register unless the Generator is the source of a locked Peripheral Channel (PCHCTRLm.WRTLOCK=1). The reset values of the GENCTRL register are as shown in the table below.

Table 16-6. GENCTRLn Reset Value after a User Reset

GCLK Generator	Reset Value after a User Reset
0	0x00000105
others	No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK=1 else 0x00000000

16.8.4 Peripheral Channel Control

PCHTRLm controls the settings of Peripheral Channel number m (m=0..28).

Name: PCHCTRLm
Offset: 0x80 + n*0x04 [n=0..28]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WRTLOCK	CHEN			GEN[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 7 – WRTLOCK: Write Lock
After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.

Note that Generator 0 cannot be locked.

Value	Description
0	The Peripheral Channel register and the associated Generator register are not locked
1	The Peripheral Channel register and the associated Generator register are locked

Bit 6 – CHEN: Channel Enable
This bit is used to enable and disable a Peripheral Channel.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 3:0 – GEN[3:0]: Generator Selection
This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

Table 16-7. Generator Selection

Value	Description
0x0	Generic Clock Generator 0
0x1	Generic Clock Generator 1
0x2	Generic Clock Generator 2
0x3	Generic Clock Generator 3

Value	Description
0x4	Generic Clock Generator 4
0x5 - 0xF	Reserved

Table 16-8. Reset Value after a User Reset or a Power Reset

Reset	PCHCTRLm.GEN	PCHCTRLm.CHEN	PCHCTRLm.WRTLOCK
Power Reset	0x0	0x0	0x0
User Reset	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	No change

A Power Reset will reset all the PCHCTRLm registers.

A User Reset will reset a PCHCTRL if WRTLOCK=0, or else, the content of that PCHCTRL remains unchanged.

PCHCTRL register Reset values are shown in the table PCHCTRLm Mapping.

Table 16-9. PCHCTRLm Mapping

index(m)	Name	Description
0	GCLK_DFLL48M_REF	DFLL48M Reference
1	GCLK_DPLL	FDPLL96M input clock source for reference
2	GCLK_DPLL_32K	FDPLL96M 32kHz clock for FDPLL96M internal lock timer
3	GCLK_EIC	EIC
4	GCLK_FREQM_MSR	FREQM Measure
5	GCLK_FREQM_REF	FREQM Reference
6	USB	USB
7	GCLK_EVSYS_CHANNEL_0	EVSYS_CHANNEL_0
8	GCLK_EVSYS_CHANNEL_1	EVSYS_CHANNEL_1
9	GCLK_EVSYS_CHANNEL_2	EVSYS_CHANNEL_2
10	GCLK_EVSYS_CHANNEL_3	EVSYS_CHANNEL_3
11	GCLK_EVSYS_CHANNEL_4	EVSYS_CHANNEL_4
12	GCLK_EVSYS_CHANNEL_5	EVSYS_CHANNEL_5
13	GCLK_EVSYS_CHANNEL_6	EVSYS_CHANNEL_6
14	GCLK_EVSYS_CHANNEL_7	EVSYS_CHANNEL_7
15	GCLK_SERCOM[0,1,2,3,4,5]_SLOW	SERCOM[0,1,2,3,4,5]_SLOW
16	GCLK_SERCOM0_CORE	SERCOM0_CORE
17	GCLK_SERCOM1_CORE	SERCOM1_CORE

index(m)	Name	Description
18	GCLK_SERCOM2_CORE	SERCOM2_CORE
19	GCLK_SERCOM3_CORE	SERCOM3_CORE
20	GCLK_SERCOM4_CORE	SERCOM4_CORE
21	GCLK_SERCOM5_CORE	SERCOM5_CORE
22	GCLK_TCC0	TCC0
23	GCLK_TC0, GCLK_TC1	TC0, TC1
24	GCLK_TC2, GCLK_TC3	TC2, TC3
25	ADC	ADC
26	AC	AC
27	PTC	PTC
28	CCL	CCL

17. MCLK – Main Clock

17.1 Overview

The Main Clock (MCLK) controls the synchronous clock generation of the device.

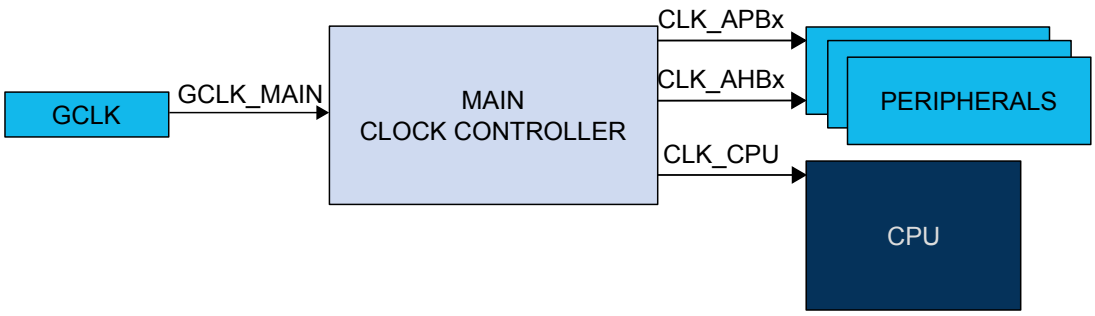
Using a clock provided by the Generic Clock Module (GCLK_MAIN), the Main Clock Controller provides synchronous system clocks to the CPU and the modules connected to the AHBx and the APBx bus. The synchronous system clocks are divided into a number of clock domains. Each clock domain can run at different frequencies, enabling the user to save power by running peripherals at a relatively low clock frequency, while maintaining high CPU performance or vice versa. In addition, the clock can be masked for individual modules, enabling the user to minimize power consumption.

17.2 Features

- Generates CPU, AHB, and APB system clocks
  - Clock source and division factor from GCLK
  - Clock prescaler with 1x to 128x division
- Safe run-time clock switching from GCLK
- Module-level clock gating through maskable peripheral clocks

17.3 Block Diagram

Figure 17-1. MCLK Block Diagram



17.4 Signal Description

Not applicable.

17.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

17.5.1 I/O Lines

Not applicable.

17.5.2 Power Management

The MCLK will operate in all sleep modes if a synchronous clock is required in these modes.

Related Links

PM – Power Manager

17.5.3 Clocks

The MCLK bus clock (CLK_MCLK_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_MCLK_APB can be found in the Peripheral Clock Masking section. If this clock is disabled, it can only be re-enabled by a reset.

The Generic Clock GCLK_MAIN is required to generate the Main Clocks. GCLK_MAIN is configured in the Generic Clock Controller, and can be re-configured by the user if needed.

Related Links

GCLK - Generic Clock Controller

Peripheral Clock Masking

17.5.3.1 Main Clock

The main clock GCLK_MAIN is the common source for the synchronous clocks. This is fed into the common 8-bit prescaler that is used to generate synchronous clocks to the CPU, AHBx, and APBx modules.

17.5.3.2 CPU Clock

The CPU clock (CLK_CPU) is routed to the CPU. Halting the CPU clock inhibits the CPU from executing instructions.

17.5.3.3 APBx and AHBx Clock

The APBx clocks (CLK_APBx) and the AHBx clocks (CLK_AHBx) are the root clock source used by modules requiring a clock on the APBx and the AHBx bus. These clocks are always synchronous to the CPU clock, but can be divided by a prescaler, and can run even when the CPU clock is turned off in sleep mode. A clock gater is inserted after the common APB clock to gate any APBx clock of a module on APBx bus, as well as the AHBx clock.

17.5.3.4 Clock Domains

The device has these synchronous clock domains:

- High-Speed synchronous clock domain (HS Clock Domain). Frequency is f_{hs}.
- CPU synchronous clock domain (CPU Clock Domain). Frequency is f_{CPU}.
- Backup synchronous clock domain. (BUP Clock Domain). Frequency is f_{BUP}.

See also the related links for the clock domain partitioning.

Related Links

Peripheral Clock Masking

17.5.4 DMA

Not applicable.

17.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the MCLK interrupt requires the Interrupt Controller to be configured first.

17.5.6 Events

Not applicable.

17.5.7 Debug Operation

When the CPU is halted in debug mode, the MCLK continues normal operation. In sleep mode, the clocks generated from the MCLK are kept running to allow the debugger accessing any module. As a consequence, power measurements are incorrect in debug mode.

17.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

17.5.9 Analog Connections

Not applicable.

17.6 Functional Description

17.6.1 Principle of Operation

The GCLK_MAIN clock signal from the GCLK module is the source for the main clock, which in turn is the common root for the synchronous clocks for the CPU, APBx, and AHBx modules. The GCLK_MAIN is divided by an 8-bit prescaler. Each of the derived clocks can run from any divided or undivided main clock, ensuring synchronous clock sources for each clock domain. Each clock domain (CPU, BUP) can be changed on the fly to respond to variable load in the application as long as $f_{CPU} \geq f_{BUP}$. The clocks for each module in a clock domain can be masked individually to avoid power consumption in inactive modules. Depending on the sleep mode, some clock domains can be turned off.

17.6.2 Basic Operation

17.6.2.1 Initialization

After a Reset, the default clock source of the GCLK_MAIN clock is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

By default, only the necessary clocks are enabled.

Related Links

Peripheral Clock Masking

17.6.2.2 Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

17.6.2.3 Selecting the Main Clock Source

Refer to the Generic Clock Controller description for details on how to configure the clock source of the GCLK_MAIN clock.

Related Links

GCLK - Generic Clock Controller

17.6.2.4 Selecting the Synchronous Clock Division Ratio

The main clock GCLK_MAIN feeds an 8-bit prescaler, which can be used to generate the synchronous clocks. By default, the synchronous clocks run on the undivided main clock. The user can select a prescaler division for the CPU clock domain by writing the Division (DIV) bits in the CPU Clock Division register CPUDIV, resulting in a CPU clock domain frequency determined by this equation:

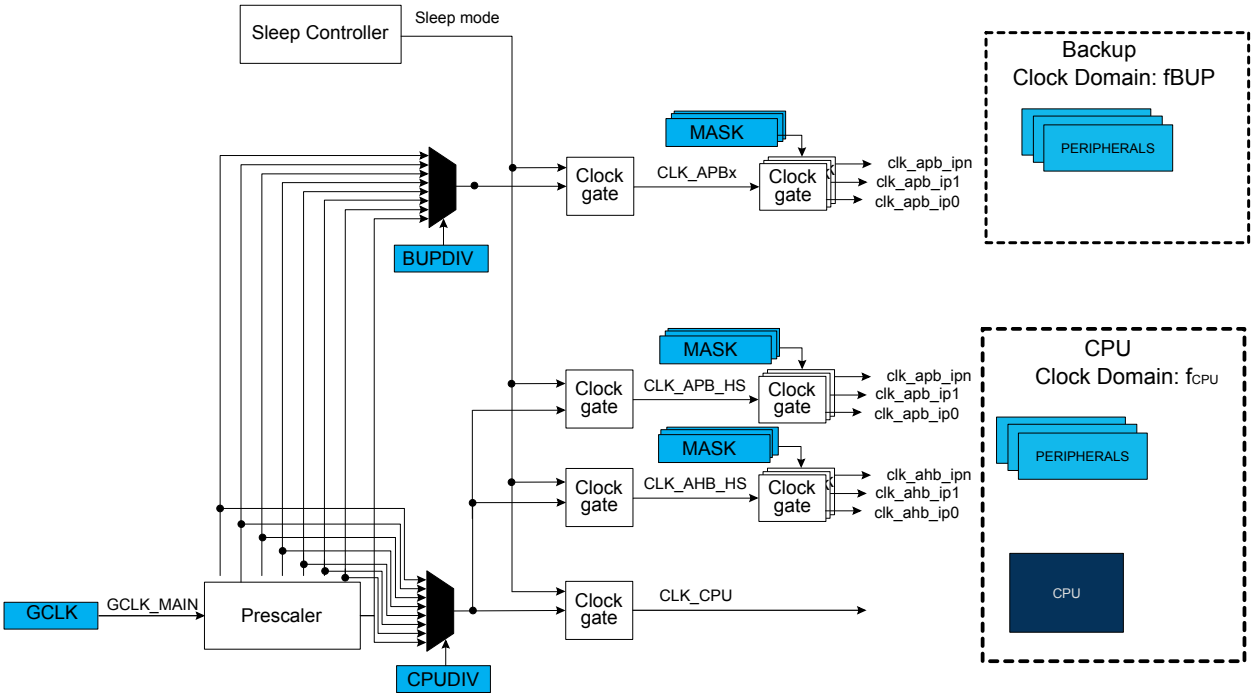
$$f_{CPU} = \frac{f_{main}}{CPUDIV}$$

Similarly, the clock for the Backup Clock Domain can be divided by writing the BUPDIV register. To ensure correct operation, frequencies must be selected so that $f_{CPU} \geq f_{BUP}$. Also, frequencies must never exceed the specified maximum frequency for each clock domain given in the electrical characteristics specifications.

If the application attempts to write forbidden values in CPUDIV or BUPDIV registers, registers are written but these bad values are not used and a violation is reported to the PAC module.

Division bits (DIV) can be written without halting or disabling peripheral modules. Writing DIV bits allows a new clock setting to be written to all synchronous clocks belonging to the corresponding clock domain at the same time. Each clock domain can be changed without changing others. This way, it is possible to, for example, scale the CPU clock domain speed according to the required performance, while keeping the Backup Clock Domain frequency constant.

Figure 17-2. Synchronous Clock Selection and Prescaler



Related Links

PAC - Peripheral Access Controller

Electrical Characteristics

17.6.2.5 Clock Ready Flag

There is a slight delay between writing to CPUDIV and BUPDIV until the new clock settings become effective.

During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will return zero when read. If CKRDY in the INTENSET register is set to '1', the Clock Ready interrupt will be triggered when the new clock setting is effective. The clock settings (CLKCFG) must not be re-written while INTFLAG. CKRDY reads '0'. The system may become unstable or hang, and a violation is reported to the PAC module.

Related Links

PAC - Peripheral Access Controller

17.6.2.6 Peripheral Clock Masking

It is possible to disable/enable the AHB or APB clock for a peripheral by writing the corresponding bit in the Clock Mask registers (APBxMASK) to '0'/'1'. The default state of the peripheral clocks is shown here.

Table 17-1. Peripheral Clock Default State

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_AC_APB	Enabled
CLK_ADC_APB	Enabled
CLK_AES_APB	Enabled
CLK_BRIDGE_A_AHB	Enabled
CLK_BRIDGE_B_AHB	Enabled
CLK_BRIDGE_C_AHB	Enabled
CLK_CCL_APB	Enabled
CLK_DMAC_AHB	Enabled
CLK_DSU_AHB	Enabled
CLK_DSU_APB	Enabled
CLK_EIC_APB	Enabled
CLK_EVSYS_APB	Enabled
CLK_FREQM_APB	Enabled
CLK_GCLK_APB	Enabled
CLK_MCLK_APB	Enabled
CLK_NVMCTRL_AHB	Enabled
CLK_NVMCTRL_APB	Enabled
CLK_OSCCTRL_APB	Enabled
CLK_PAC_AHB	Enabled
CLK_PAC_APB	Enabled
CLK_PORT_APB	Enabled
CLK_PTC_APB	Enabled
CLK_SERCOM0_APB	Enabled

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_SERCOM1_APB	Enabled
CLK_SERCOM2_APB	Enabled
CLK_SERCOM3_APB	Enabled
CLK_SERCOM4_APB	Enabled
CLK_SERCOM5_APB	Enabled
CLK_SLCD_APB	Enabled
CLK_TC0_APB	Enabled
CLK_TC1_APB	Enabled
CLK_TC2_APB	Enabled
CLK_TC3_APB	Enabled
CLK_TCC0_APB	Enabled
CLK_TRNG_APB	Enabled
CLK_USB_AHB	Enabled
CLK_USB_APB	Enabled
CLK_WDT_APB	Enabled

Backup Clock Domain	
Peripheral Clock	**Default State**
CLK_OSC32KCTRL_APB	Enabled
CLK_PM_APB	Enabled
CLK_SUPC_APB	Enabled
CLK_RSTC_APB	Enabled
CLK_RTC_APB	Enabled

When the APB clock is not provided to a module, its registers cannot be read or written. The module can be re-enabled later by writing the corresponding mask bit to '1'.

A module may be connected to several clock domains (for instance, AHB and APB), in which case it will have several mask bits.

Note that clocks should only be switched off if it is certain that the module will not be used: Switching off the clock for the NVM Controller (NVMCTRL) will cause a problem if the CPU needs to read from the Flash Memory. Switching off the clock to the MCLK module (which contains the mask registers) or the corresponding APBx bridge, will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

17.6.3 DMA Operation

Not applicable.

17.6.4 Interrupts

The peripheral has the following interrupt sources:

- Clock Ready (CKRDY): indicates that CPU and BUP clocks are ready. This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be enabled individually by writing a '1' to the corresponding enabling bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding clearing bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset. An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources.If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

Related Links

PM – Power Manager

Sleep Mode Controller

17.6.5 Events

Not applicable.

17.6.6 Sleep Mode Operation

In IDLE sleep mode, the MCLK is still running on the selected main clock.

In STANDBY sleep mode, the MCLK is frozen if no synchronous clock is required.

17.7 Register Summary - MCLK

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								
0x01	INTENCLR	7:0								CKRDY
0x02	INTENSET	7:0								CKRDY
0x03	INTFLAG	7:0								CKRDY
0x04	Reserved									
0x05	CPUDIV	7:0	CPUDIV[7:0]							
0x06	BUPDIV	7:0	BUPDIV[7:0]							
0x07 ... 0x0F	Reserved									
0x10	AHBMASK	7:0	PAC	Reserved	DSU	USB	DMAC	APBC	APBB	APBA
0x11		15:8						Reserved	Reserved	NVMCTRL
0x12		23:16								
0x13		31:24								
0x14	APBAMASK	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x15		15:8				Reserved	FREQM	EIC	RTC	WDT
0x16		23:16								
0x17		31:24								
0x18	APBBMASK	7:0					PORT	NVMCTRL	DSU	USB
0x19		15:8								
0x1A		23:16								
0x1B		31:24								
0x1C	APBCMASK	7:0	TCC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
0x1D		15:8	SLCD	PTC	AC	ADC	TC3	TC2	TC1	TC0
0x1E		23:16						CCL	TRNG	AES
0x1F		31:24								

17.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers can be write-protected optionally by the Peripheral Access Controller (PAC). This is denoted by the property "PAC Write-Protection" in each individual register description. Refer to the Register Access Protection for details.

17.8.1 Control A

All bits in this register are reserved.

Name: CTRLA
Offset: 0x00 [ID-00001086]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

17.8.2 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x01 [ID-00001086]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY: Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled and will generate an interrupt request when the Clock Ready Interrupt Flag is set.

17.8.3 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x02 [ID-00001086]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY: Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

17.8.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x03 [ID-00001086]
Reset: 0x01
Property: –

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								1

Bit 0 – CKRDY: Clock Ready
This flag is cleared by writing a '1' to the flag.

This flag is set when the synchronous CPU, APBx, and AHBx clocks have frequencies as indicated in the CLKCFG registers and will generate an interrupt if INTENCLR/SET.CKRDY is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Clock Ready interrupt flag.

17.8.5 CPU Clock Division

Name: CPUDIV
Offset: 0x05 [ID-00001086]
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	CPUDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – CPUDIV[7:0]: CPU Clock Division Factor
These bits define the division ratio of the main clock prescaler related to the CPU clock domain.

To ensure correct operation, frequencies must be selected so that $F_{HS} \geq F_{CPU} \geq F_{BUP}$ (i.e. BUPDIV ≥ CPUDIV ≥ HSDIV).

Frequencies must never exceed the specified maximum frequency for each clock domain.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	-	Reserved

17.8.6 Backup Clock Division

Name: BUPDIV
Offset: 0x06 [ID-00001086]
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BUPDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BUPDIV[7:0]: Backup Clock Division Factor
These bits define the division ratio of the main clock prescaler (2^n) related to the Backup clock domain. To ensure correct operation, frequencies must be selected so that $F_{CPU} \geq F_{BUP}$ (i.e. BUPDIV ≥ CPUDIV). Also, frequencies must never exceed the specified maximum frequency for each clock domain.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	-	Reserved

17.8.7 AHB Mask

Name: AHBMASK
Offset: 0x10 [ID-00001086]
Reset: 0x000007FF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						Reserved	Reserved	NVMCTRL
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	7	6	5	4	3	2	1	0
	PAC	Reserved	DSU	USB	DMAC	APBC	APBB	APBA
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 10,9,6 – Reserved: Reserved bits
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 8 – NVMCTRL: NVMCTRL AHB Clock Enable

Value	Description
0	The AHB clock for the NVMCTRL is stopped
1	The AHB clock for the NVMCTRL is enabled

Bit 7 – PAC: PAC AHB Clock Enable

Value	Description
0	The AHB clock for the PAC is stopped.
1	The AHB clock for the PAC is enabled.

Bit 5 – DSU: DSU AHB Clock Enable

Value	Description
0	The AHB clock for the DSU is stopped.
1	The AHB clock for the DSU is enabled.

Bit 4 – USB: USB AHB Clock Enable

Value	Description
0	The AHB clock for the USB is stopped.
1	The AHB clock for the USB is enabled.

Bit 3 – DMAC: DMAC AHB Clock Enable

Value	Description
0	The AHB clock for the DMAC is stopped.
1	The AHB clock for the DMAC is enabled.

Bit 2 – APBC: APBC AHB Clock Enable

Value	Description
0	The AHB clock for the APBC is stopped.
1	The AHB clock for the APBC is enabled

Bit 1 – APBB: APBB AHB Clock Enable

Value	Description
0	The AHB clock for the APBB is stopped.
1	The AHB clock for the APBB is enabled.

Bit 0 – APBA: APBA AHB Clock Enable

Value	Description
0	The AHB clock for the APBA is stopped.
1	The AHB clock for the APBA is enabled.

17.8.8 APBA Mask

Name: APBAMASK
Offset: 0x14 [ID-00001086]
Reset: 0x00001FFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				Reserved	FREQM	EIC	RTC	WDT
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 12 – Reserved: For future use

Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 11 – FREQM: FREQM APBA Clock Enable

Value	Description
0	The APBA clock for the FREQM is stopped.
1	The APBA clock for the FREQM is enabled.

Bit 10 – EIC: EIC APBA Clock Enable

Value	Description
0	The APBA clock for the EIC is stopped.
1	The APBA clock for the EIC is enabled.

Bit 9 – RTC: RTC APBA Clock Enable

Value	Description
0	The APBA clock for the RTC is stopped.
1	The APBA clock for the RTC is enabled.

Bit 8 – WDT: WDT APBA Clock Enable

Value	Description
0	The APBA clock for the WDT is stopped.
1	The APBA clock for the WDT is enabled.

Bit 7 – GCLK: GCLK APBA Clock Enable

Value	Description
0	The APBA clock for the GCLK is stopped.
1	The APBA clock for the GCLK is enabled.

Bit 6 – SUPC: SUPC APBA Clock Enable

Value	Description
0	The APBA clock for the SUPC is stopped.
1	The APBA clock for the SUPC is enabled.

Bit 5 – OSC32KCTRL: OSC32KCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSC32KCTRL is stopped.
1	The APBA clock for the OSC32KCTRL is enabled.

Bit 4 – OSCCTRL: OSCCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSCCTRL is stopped.
1	The APBA clock for the OSCCTRL is enabled.

Bit 3 – RSTC: RSTC APBA Clock Enable

Value	Description
0	The APBA clock for the RSTC is stopped.
1	The APBA clock for the RSTC is enabled.

Bit 2 – MCLK: MCLK APBA Clock Enable

Value	Description
0	The APBA clock for the MCLK is stopped.
1	The APBA clock for the MCLK is enabled.

Bit 1 – PM: PM APBA Clock Enable

Value	Description
0	The APBA clock for the PM is stopped.
1	The APBA clock for the PM is enabled.

Bit 0 – PAC: PAC APBA Clock Enable

Value	Description
0	The APBA clock for the PAC is stopped.
1	The APBA clock for the PAC is enabled.

17.8.9 APBB Mask

Name: APBBMASK
Offset: 0x18 [ID-00001086]
Reset: 0x0000004F
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					PORT	NVMCTRL	DSU	USB
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit 3 – PORT: PORT APBB Clock Enable

Value	Description
0	The APBB clock for the PORT is stopped.
1	The APBB clock for the PORT is enabled.

Bit 2 – NVMCTRL: NVMCTRL APBB Clock Enable

Value	Description
0	The APBB clock for the NVMCTRL is stopped
1	The APBB clock for the NVMCTRL is enabled

Bit 1 – DSU: DSU APBB Clock Enable

Value	Description
0	The APBB clock for the DSU is stopped
1	The APBB clock for the DSU is enabled

Bit 0 – USB: USB APBB Clock Enable

Value	Description
0	The APBB clock for the USB is stopped
1	The APBB clock for the USB is enabled

Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

17.8.10 APBC Mask

Name: APBCMASK
Offset: 0x1C
Reset: 0x0007 FFFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CCL	TRNG	AES
Access						R	R	R
Reset						1	1	1

Bit	15	14	13	12	11	10	9	8
	SLCD	PTC	AC	ADC	TC3	TC2	TC1	TC0
Access	R/W	R/W	R/W	R/W	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	TCC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 18 – CCL: CCL APBC Clock Enable

Value	Description
0	The APBC clock for the CCL is stopped.
1	The APBC clock for the CCL is enabled.

Bit 17 – TRNG: TRNG APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TRNG is stopped.
1	The APBC clock for the TRNG is enabled.

Bit 16 – AES: AES APBC Mask Clock Enable

Value	Description
0	The APBC clock for the AES is stopped.
1	The APBC clock for the AES is enabled.

Bit 15 – SLCD: SLCD APBC Clock Enable

Value	Description
0	The APBC clock for the SLCD is stopped.
1	The APBC clock for the SLCD is enabled.

Bit 14 – PTC: PTC APBC Clock Enable

Value	Description
0	The APBC clock for the PTC is stopped.
1	The APBC clock for the PTC is enabled.

Bit 13 – AC: AC APBC Clock Enable

Value	Description
0	The APBC clock for the AC is stopped.
1	The APBC clock for the AC is enabled.

Bit 12 – ADC: ADC APBC Clock Enable

Value	Description
0	The APBC clock for the ADC is stopped.
1	The APBC clock for the ADC is enabled.

Bit 11 – TC3: TC3 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC3 is stopped.
1	The APBC clock for the TC3 is enabled.

Bit 10 – TC2: TC2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC2 is stopped.
1	The APBC clock for the TC2 is enabled.

Bit 9 – TC1: TC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC1 is stopped.
1	The APBC clock for the TC1 is enabled.

Bit 8 – TC0: TC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC0 is stopped.
1	The APBC clock for the TC0 is enabled.

Bit 7 – TCC0: TCC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC0 is stopped.
1	The APBC clock for the TCC0 is enabled.

Bit 6 – SERCOM5: SERCOM5 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM5 is stopped.
1	The APBC clock for the SERCOM5 is enabled.

Bit 5 – SERCOM4: SERCOM4 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM4 is stopped.
1	The APBC clock for the SERCOM4 is enabled.

Bit 4 – SERCOM3: SERCOM3 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM3 is stopped.
1	The APBC clock for the SERCOM3 is enabled.

Bit 3 – SERCOM2: SERCOM2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM2 is stopped.
1	The APBC clock for the SERCOM2 is enabled.

Bit 2 – SERCOM1: SERCOM1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM1 is stopped.
1	The APBC clock for the SERCOM1 is enabled.

Bit 1 – SERCOM0: SERCOM0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM0 is stopped.
1	The APBC clock for the SERCOM0 is enabled.

Bit 0 – EVSYS: EVSYS APBC Clock Enable

Value	Description
0	The APBC clock for the EVSYS is stopped.
1	The APBC clock for the EVSYS is enabled.

18. FREQM – Frequency Meter

18.1 Overview

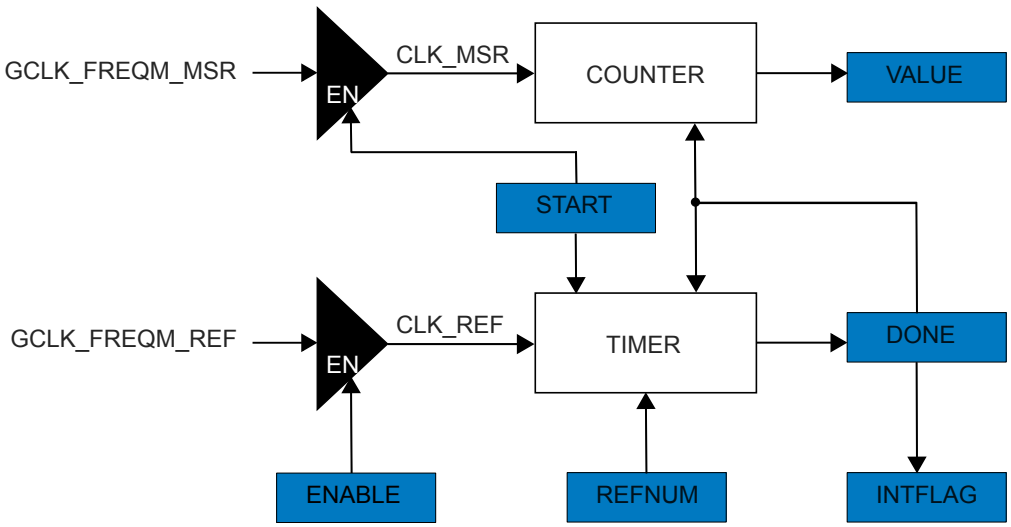
The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

18.2 Features

- Accurately measures a clock frequency
- A selectable reference clock from GCLK_FREQM_REF sources
- A selectable clock from GCLK_FREQM_MSR sources can be measured
- Ratio can be measured with 24-bit accuracy

18.3 Block Diagram

Figure 18-1. FREQM Block Diagram



18.4 Signal Description

Not applicable.

18.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

18.5.1 I/O Lines

Not applicable.

18.5.2 Power Management

The FREQM will continue to operate in idle sleep modes where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

18.5.3 Clocks

The clock for the FREQM bus interface (CLK_APB_FREQM) is enabled and disabled by the Main Clock Controller, the default state of CLK_APB_FREQM can be found in the Peripheral Clock Masking section.

Two generic clocks are used by the FREQM(GCLK_FREQM_REF and GCLK_FREQM_MSR). The reference clock (GCLK_FREQM_REF) is required to clock the internal reference timer while operating as a frequency reference, while the measurement clock (GCLK_FREQM_MSR) is required to clock a ripple counter for frequency measurement. These clocks must be configured and enabled in the generic clock controller before using the FREQM.

Related Links

MCLK – Main Clock

Peripheral Clock Masking

GCLK - Generic Clock Controller

18.5.4 DMA

Not applicable.

18.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using FREQM interrupt requires the interrupt controller to be configured first.

18.5.6 Events

Not applicable

18.5.7 Debug Operation

When the CPU is halted in debug mode the FREQM continues normal operation. If the FREQM is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

18.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Control B register (CTRLB)
- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

18.6 Functional Description

18.6.1 Principle of Operation

During a period of $\mathrm{REFNUM}/f_{\mathrm{CLK_REF}}$, the FREQM is counting the number of periods of the clock to be measured, VALUE. Here, REFNUM is the Number of Reference Clock Cycles selected in the Configuration A register (CFGA.REFNUM), VALUE is the Measurement result stored to the Value register (VALUE.VALUE), and $f_{\mathrm{CLK_REF}}$ is the frequency of the reference clock.

The frequency of the measured clock, $f_{\mathrm{CLK_MSR}}$, is calculated by

$$f_{\mathrm{CLK_MSR}} = \left(\frac{\mathrm{VALUE}}{\mathrm{REFNUM}}\right) f_{\mathrm{CLK_REF}}$$

18.6.2 Basic Operation

18.6.2.1 Initialization

Before enabling FREQM, the device and peripheral must be configured:

- Each of the generic clocks (GCLK_FREQM_REF and GCLK_FREQM_MSR) should be configured and enabled. Note that the reference clock should be slower than the measurement clock.
- The Number of Reference Clock Cycles value in the Configuration A register (CFGA.REFNUM) must be written to a value greater than 0x00.

The following register is enable-protected, meaning that it can only be written when the FREQM is disabled (CTRLA.ENABLE is zero):

- Configuration A register (CFGA)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Related Links

GCLK - Generic Clock Controller

18.6.2.2 Enabling, Disabling and Resetting

The FREQM is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The peripheral is disabled by writing CTRLA.ENABLE=0.

The FREQM is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the FREQM will be reset to their initial state, and the FREQM will be disabled.

18.6.2.3 Measurement

In the Configuration A register, the Number of Reference Clock Cycles field (CFGA.REFNUM) selects the duration of the measurement. The measurement is given in number of GCLK_FREQM_REF periods.
Note: The REFNUM field must be written before the FREQM is enabled.

After the FREQM is enabled, writing a '1' to the START bit in the Control B register (CTRLB.START) starts the measurement. The BUSY bit in Status register (STATUS.BUSY) is cleared when the measurement is done.

There is also an interrupt request for Measurement Done: When the Measurement Done bit in Interrupt Enable Set register (INTENSET.DONE) is '1' and a measurement is finished, the Measurement Done bit in the Interrupt Flag Status and Clear register (INTFLAG.DONE) will be set and an interrupt request is generated.

The result of the measurement can be read from the Value register (VALUE.VALUE). The frequency of the measured clock GCLK_FREQM_MSR is then:

$$f_{CLK_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK_REF}$$

Note: In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) should be checked.

In case an overflow condition occurred, indicated by the Overflow bit in the STATUS register (STATUS.OVF), either the number of reference clock cycles must be reduced (CFGA.REFNUM), or a faster reference clock must be configured. Once the configuration is adjusted, clear the overflow status by writing a '1' to STATUS.OVF. Then another measurement can be started by writing a '1' to CTRLB.START.

18.6.3 DMA Operation

Not applicable.

18.6.4 Interrupts

The FREQM has one interrupt source:

- DONE: A frequency measurement is done

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the FREQM is reset. See INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

18.6.5 Events

Not applicable.

18.6.6 Sleep Mode Operation

The FREQM will continue to operate in idle sleep modes where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep modes.

For lowest chip power consumption in sleep modes, FREQM should be disabled before entering a sleep mode.

Related Links

PM – Power Manager

18.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits and registers are write-synchronized:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization

18.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0								START
0x02	CFGA	7:0	REFNUM[7:0]							
0x03		15:8								
0x04 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DONE
0x09	INTENSET	7:0								DONE
0x0A	INTFLAG	7:0								DONE
0x0B	STATUS	7:0							OVF	BUSY
0x0C	SYNCBUSY	7:0							ENABLE	SWRST
0x0D		15:8								
0x0E		23:16								
0x0F		31:24								
0x10	VALUE	7:0	VALUE[7:0]							
0x11		15:8	VALUE[15:8]							
0x12		23:16	VALUE[23:16]							
0x13		31:24								

18.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description.

18.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000e03]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	R/W
Reset							0	0

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the FREQM to their initial state, and the FREQM will be disabled. Writing a '1' to this bit will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

18.8.2 Control B

Name: CTRLB
Offset: 0x01 [ID-00000e03]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								0

Bit 0 – START: Start Measurement

Value	Description
0	Writing a '0' has no effect.
1	Writing a '1' starts a measurement.

18.8.3 Configuration A

Name: CFGA
Offset: 0x02 [ID-00000e03]
Reset: 0x0000
Property: PAC Write-Protection, Enable-protected

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REFNUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – REFNUM[7:0]: Number of Reference Clock Cycles
Selects the duration of a measurement in number of CLK_FREQM_REF cycles. This must be a non-zero value, i.e. 0x01 (one cycle) to 0xFF (255 cycles).

18.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08 [ID-00000e03]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE: Measurement Done Interrupt Enable
Writing a '1' to this bit has no effect.

Writing a '1' to this bit will clear the Measurement Done Interrupt Enable bit, which disables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

18.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x09 [ID-00000e03]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE: Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Measurement Done Interrupt Enable bit, which enables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

18.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A [ID-00000e03]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R/W
Reset								0

Bit 0 – DONE: Mesurement Done
This flag is cleared by writing a '1' to it.

This flag is set when the STATUS.BUSY bit has a one-to-zero transition.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the DONE interrupt flag.

18.8.7 Status

Name: STATUS
Offset: 0x0B [ID-00000e03]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							OVF	BUSY
Access							R/W	R
Reset							0	0

Bit 1 – OVF: Sticky Count Value Overflow
This bit is cleared by writing a '1' to it.

This bit is set when an overflow condition occurs to the value counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the OVF status.

Bit 0 – BUSY: FREQM Status

Value	Description
0	No frequency measurement ongoing.
1	Frequency measurement is ongoing.

18.8.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x0C [ID-00000e03]
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE: Enable
This bit is cleared when the synchronization of CTRLA.ENABLE is complete.

This bit is set when the synchronization of CTRLA.ENABLE is started.

Bit 0 – SWRST: Synchronization Busy
This bit is cleared when the synchronization of CTRLA.SWRST is complete.

This bit is set when the synchronization of CTRLA.SWRST is started.

18.8.9 Value

Name: VALUE
Offset: 0x10 [ID-00000e03]
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	VALUE[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – VALUE[23:0]: Measurement Value
Result from measurement.

19. RSTC – Reset Controller

19.1 Overview

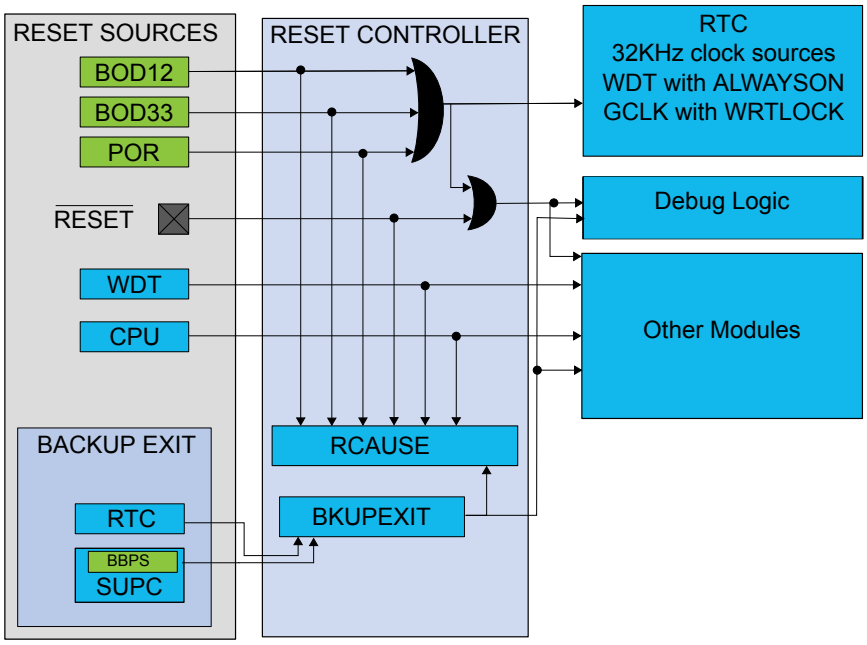
The Reset Controller (RSTC) manages the reset of the microcontroller. It issues a microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

19.2 Features

- Reset the microcontroller and set it to an initial state according to the reset source
- Reset cause register for reading the reset source from the application code
- Multiple reset sources
  - Power supply reset sources: POR, BOD12, BOD33
  - User reset sources: External reset ($\overline{\text{RESET}}$), Watchdog reset, and System Reset Request
  - Backup exit sources: Real-Time Counter (RTC) and Battery Backup Power Switch (BBPS)

19.3 Block Diagram

Figure 19-1. Reset System



19.4 Signal Description

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital input	External reset

One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

19.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

19.5.1 I/O Lines

Not applicable.

19.5.2 Power Management

The Reset Controller module is always on.

19.5.3 Clocks

The RSTC bus clock (CLK_RSTC_APB) can be enabled and disabled in the Main Clock Controller.

Related Links

MCLK – Main Clock

Peripheral Clock Masking

19.5.4 DMA

Not applicable.

19.5.5 Interrupts

Not applicable.

19.5.6 Events

Not applicable.

19.5.7 Debug Operation

When the CPU is halted in debug mode, the RSTC continues normal operation.

19.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

19.5.9 Analog Connections

Not applicable.

19.6 Functional Description

19.6.1 Principle of Operation

The Reset Controller collects the various Reset sources and generates Reset for the device.

19.6.2 Basic Operation

19.6.2.1 Initialization

After a power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the POR source.

19.6.2.2 Enabling, Disabling, and Resetting

The RSTC module is always enabled.

19.6.2.3 Reset Causes and Effects

The latest Reset cause is available in RCAUSE register, and can be read during the application boot sequence in order to determine proper action.

These are the groups of Reset sources:

- Power supply Reset: Resets caused by an electrical issue. It covers POR and BODs Resets
- User Reset: Resets caused by the application. It covers external Resets, system Reset requests and watchdog Resets
- Backup reset: Resets caused by a Backup Mode exit condition

The following table lists the parts of the device that are reset, depending on the Reset type.

Table 19-1. Effects of the Different Reset Causes

	Power Supply Reset		User Reset		Backup Reset
	POR, BOD33	BOD12	External Reset	WDT Reset, System Reset Request	RTC, BBPS
RTC, OSC32KCTRL, RSTC, CTRLA.IORET bit of PM	Y	N	N	N	N
GCLK with WRTLOCK	Y	Y	N	N	Y
Debug logic	Y	Y	Y	N	Y
Others	Y	Y	Y	Y	Y

The external Reset is generated when pulling the $\overline{\text{RESET}}$ pin low.

The POR, BOD12, and BOD33 Reset sources are generated by their corresponding module in the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer.

The System Reset Request is a Reset generated by the CPU when asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details refer to the ARM® Cortex™ Technical Reference Manual on http://www.arm.com).

From Backup Mode, the chip can be waken-up upon these conditions:

- Battery Backup Power Switch (BBPS): generated by the SUPC controller when the 3.3V VDDIO is restored.
- Real-Time Counter interrupt. For details refer to the applicable INTFLAG in the RTC for details.

If one of these conditions is triggered in Backup Mode, the RCAUSE.BACKUP bit is set and the Backup Exit Register (BKUPEXIT) is updated.

Related Links

SUPC – Supply Controller

Battery Backup Power Switch

19.6.3 Additional Features

Not applicable.

19.6.4 DMA Operation

Not applicable.

19.6.5 Interrupts

Not applicable.

19.6.6 Events

Not applicable.

19.6.7 Sleep Mode Operation

The RSTC module is active in all sleep modes.

19.7 Register Summary

Offset	Name	Bit Pos.								
0x00	RCAUSE	7:0	BACKUP	SYST	WDT	EXT		BOD33	BOD12	POR
0x01	Reserved									
0x02	BKUPEXIT	7:0						BBPS	RTC	

19.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

19.8.1 Reset Cause

When a Reset occurs, the bit corresponding to the Reset source is set to '1' and all other bits are written to '0'.

Name: RCAUSE
Offset: 0x00 [ID-0000052b]
Reset: Latest Reset Source
Property: –

Bit	7	6	5	4	3	2	1	0
	BACKUP	SYST	WDT	EXT		BOD33	BOD12	POR
Access	R	R	R	R		R	R	R
Reset	x	x	x	x		x	x	x

Bit 7 – BACKUP: Backup Reset
This bit is set if a Backup Reset has occurred. Refer to BKUPEXIT register to identify the source of the Backup Reset.

Bit 6 – SYST: System Reset Request
This bit is set if a System Reset Request has occurred. Refer to the Cortex processor documentation for more details.

Bit 5 – WDT: Watchdog Reset
This bit is set if a Watchdog Timer Reset has occurred.

Bit 4 – EXT: External Reset
This bit is set if an external Reset has occurred.

Bit 2 – BOD33: Brown Out 33 Detector Reset
This bit is set if a BOD33 Reset has occurred.

Bit 1 – BOD12: Brown Out 12 Detector Reset
This bit is set if a BOD12 Reset has occurred.

Bit 0 – POR: Power On Reset
This bit is set if a POR has occurred.

19.8.2 Backup Exit Source

When a Backup Reset occurs, the bit corresponding to the exit condition is set to '1', the other bits are written to '0'.

In some specific cases, the RTC and BBPS bits can be set together, e.g. when the device leaves the battery Backup Mode caused by a BBPS condition, and a RTC event was generated during the Battery Backup Mode period.

Name: BKUPEXIT
Offset: 0x02 [ID-0000052b]
Reset: Latest Backup Exit Source
Property: –

Bit	7	6	5	4	3	2	1	0
						BBPS	RTC	
Access						R	R	
Reset						x	x	

Bit 2 – BBPS: Battery Backup Power Switch
This bit is set if the Battery Backup Power Switch of the Supply Controller changes back from battery mode to main power mode.

Bit 1 – RTC: Real Timer Counter Interrupt
This bit is set if an RTC interrupt flag is set in Backup Mode.

20. PM – Power Manager

20.1 Overview

The Power Manager (PM) controls the sleep modes of the device.

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

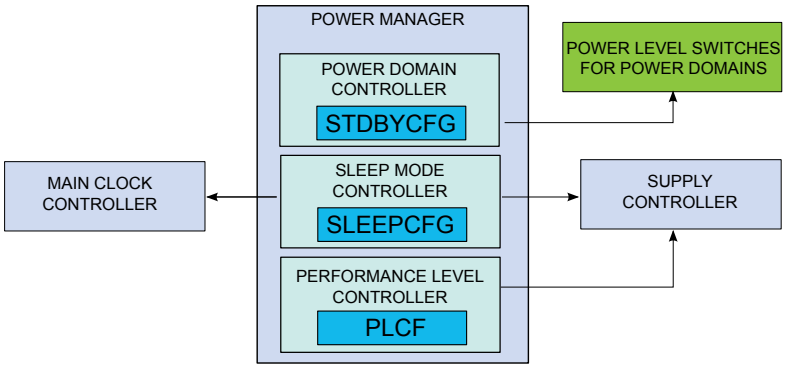
In backup mode, the PM allows retaining the state of the I/O lines, preventing I/O lines from toggling during wake-up.

20.2 Features

- Power management control
  - Sleep modes: Idle, Standby, Backup, and Off
  - Performance levels: PL0 and PL2
  - SleepWalking available in Standby mode.
  - I/O lines retention in Backup mode

20.3 Block Diagram

Figure 20-1. PM Block Diagram



20.4 Signal Description

Not applicable.

20.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

20.5.1 I/O Lines

Not applicable.

20.5.2 Clocks

The PM bus clock (CLK_PM_APB) can be enabled and disabled in the Main Clock module. If this clock is disabled, it can only be re-enabled by a system reset.

20.5.3 DMA

Not applicable.

20.5.4 Interrupts

The interrupt request line is connected to the interrupt controller. Using the PM interrupt requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

Interrupt Line Mapping

20.5.5 Events

Not applicable.

20.5.6 Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If standby sleep mode is requested by the system while in debug mode, the power domains are not turned off. As a consequence, power measurements while in debug mode are not relevant.

If Backup sleep mode is requested by the system while in debug mode, the core domains are kept on, and the debug modules are kept running to allow the debugger to access internal registers. When exiting the backup mode upon a reset condition, the core domains are reset except the debug logic, allowing users to keep using their current debug session.

Hot plugging in standby mode is supported.

Cold or Hot plugging in OFF or Backup mode is not supported.

20.5.7 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag register (INTFLAG). Refer to INTFLAG for details

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

20.5.8 Analog Connections

Not applicable.

20.6 Functional Description

20.6.1 Terminology

The following is a list of terms used to describe the Power Managemement features of this microcontroller.

20.6.1.1 Performance Levels

To help balance between performance and power consumption, the device has two performance levels. Each of the performance levels has a maximum operating frequency and a corresponding maximum consumption in µA/MHz.

It is the application's responsibility to configure the appropriate PL depending on the application activity level. When the application selects a new PL, the voltage applied on the full logic area moves from one value to another. This voltage scaling technique allows to reduce the active power consumption while decreasing the maximum frequency of the device.

PL0
Performance Level 0 (PL0) provides the maximum energy efficiency configuration.

Refer to Electrical Characteristics for details on energy consumption and maximum operating frequency.

PL2
Performance Level 2 (PL2) provides the maximum operating frequency.

Refer to Electrical Characteristics for details on energy consumption and maximum operating frequency.

20.6.1.2 Power Domains

In addition to the supply domains, such as VDDIO and VDDANA, the device provides these power domains:

- PDTOP
- PDBACKUP

Related Links

Power Domain Overview

PDTOP
PDTOP contains all controllers located in the core domain. It is powered when in Active, Idle or Standby mode. When in Backup of Off mode, this domain is completely powered down.

PDBACKUP
The Backup Power Domain (PDBACKUP) is always on, except in the off sleep mode. It contains the 32KHz oscillator sources, the Supply Controller, the Reset Controller, the Real Time Counter, and the Power Manager itself.

20.6.1.3 Sleep Modes

The device can be set in a sleep mode. In sleep mode, the CPU is stopped and the peripherals are either active or idle, according to the sleep mode depth:

- Idle sleep mode: The CPU is stopped. Synchronous clocks are stopped except when requested. The logic is retained.
- Standby sleep mode: The CPU is stopped as well as the peripherals.
- Backup sleep mode: Only the backup domain is kept powered to allow few features to run (RTC, 32KHz clock sources, and wake-up from external pins).
- Off sleep mode: The entire device is powered off.

20.6.2 Principle of Operation

In active mode, all clock domains and power domains are active, allowing software execution and peripheral operation. The PM Sleep Mode Controller allows to save power by choosing between different sleep modes depending on application requirements, see Sleep Mode Controller.

The PM Performance Level Controller allows to optimize either for low power consumption or high performance.

The PM Power Domain Controller allows to reduce the power consumption in standby mode even further.

20.6.3 Basic Operation

20.6.3.1 Initialization

After a power-on reset, the PM is enabled, the device is in ACTIVE mode, the performance level is PL0 (the lowest power consumption) and all the power domains are in active state.

20.6.3.2 Enabling, Disabling and Resetting

The PM is always enabled and can not be reset.

20.6.3.3 Sleep Mode Controller

A Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE) select the level of the sleep mode.

Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software must ensure that the SLEEPCFG register reads the desired value before issuing a WFI instruction.

Note: After power-up, the MAINVREG low power mode takes some time to stabilize. Once stabilized, the INTFLAG.SLEEPRDY bit is set. Before entering Standby or Backup mode, software must ensure that the INTFLAG.SLEEPRDY bit is set.

Table 20-1. Sleep Mode Entry and Exit Table

Mode	Mode Entry	Wake-Up Sources
IDLE	SLEEPCFG.SLEEPMODE = IDLE	Synchronous (2) (APB, AHB), asynchronous (1)
STANDBY	SLEEPCFG.SLEEPMODE = STANDBY	Synchronous(3), Asynchronous
BACKUP	SLEEPCFG.SLEEPMODE = BACKUP	Backup reset detected by the RSTC
OFF	SLEEPCFG.SLEEPMODE = OFF	External Reset

Note:

1. Asynchronous: interrupt generated on generic clock, external clock, or external event.
2. Synchronous: interrupt generated on the APB clock.
3. Synchronous interrupt only for peripherals configured to run in standby.

Note: The type of wake-up sources (synchronous or asynchronous) is given in each module interrupt section.

The sleep modes (idle, standby, backup, and off) and their effect on the clocks activity, the regulator and the NVM state are described in the table and the sections below.

Table 20-2. Sleep Mode Overview

Mode	Main clock	CPU	AHBx and APBx clock	GCLK clocks	Oscillators		Regulator	NVM
					ONDEMAND = 0	ONDEMAND = 1		
Active	Run	Run	Run	Run(3)	Run	Run if requested	MAINVREG	active
IDLE	Run	Stop	Stop(1)	Run(3)	Run	Run if requested	MAINVREG	active
STANDBY	Stop(1)	Stop	Stop(1)	Stop(1)	Run if requested or RUNSTDBY=1	Run if requested	MAINVREG in low power mode	Ultra Low power
BACKUP	Stop	Stop	Stop	Stop	Stop	Stop	Backup regulator (ULPVREG)	OFF
OFF	Stop	Stop	Stop	OFF	OFF	OFF	OFF	OFF

Note:

1. Running if requested by peripheral during SleepWalking.
2. Running during SleepWalking.
3. Following On-Demand Clock Request principle.

IDLE Mode

The IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

- Entering IDLE mode: The IDLE mode is entered by executing the WFI instruction. Additionally, if the SLEEPONEXIT bit in the ARM Cortex System Control register (SCR) is set, the IDLE mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine, see ARM Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Before entering the IDLE mode, the user must select the idle Sleep Mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=IDLE).
- Exiting IDLE mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry. The system goes back to the ACTIVE mode. The CPU and affected modules are restarted.

GCLK clocks, regulators and RAM are not affected by the idle sleep mode and operate in normal mode.

STANDBY Mode

The STANDBY mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

In this mode, all clocks are stopped except those configured to be running sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in sleepwalking tasks. This is the case when for example the SERCOM RUNSTDBY bit is written to '1'.

- Entering STANDBY mode: This mode is entered by executing the WFI instruction after writing the Sleep Mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=STANDBY). The SLEEPONEXIT feature is also available as in IDLE mode.
- Exiting STANDBY mode: Any peripheral able to generate an asynchronous interrupt can wake up the system. For example, a peripheral running on a GCLK clock can trigger an interrupt. When the enabled asynchronous wake-up event occurs and the system is woken up, the device will either

execute the interrupt service routine or continue the normal program execution according to the Priority Mask Register (PRIMASK) configuration of the CPU.

Refer to Regulators, RAMs, and NVM State in Sleep Mode for the RAM state.

The regulator operates in low-power mode by default and switches automatically to the normal mode in case of a sleepwalking task requiring more power. It returns automatically to low power mode when the sleepwalking task is completed.

BACKUP Mode

The BACKUP mode allows achieving the lowest power consumption aside from OFF. The device is entirely powered off except for the backup domain. All peripherals in backup domain are allowed to run, e.g. the RTC can be clocked by a 32.768kHz oscillator. All PM registers are reset except the CTRLA.IORET bit.

- Entering Backup mode: This mode is entered by executing the WFI instruction after selecting the Backup mode by writing the Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=BACKUP).
- Exiting Backup mode: is triggered when a Backup Reset is detected by the Reset Controller (RSTC).

OFF Mode

In OFF mode, the device is entirely powered-off.

- Entering OFF mode: This mode is entered by selecting the OFF mode in the Sleep Configuration register by writing the Sleep Mode bits (SLEEPCFG.SLEEPMODE=OFF), and subsequent execution of the WFI instruction.
- Exiting OFF mode: This mode is left by pulling the $\overline{\text{RESET}}$ pin low, or when a power Reset is done.

20.6.3.4 I/O Lines Retention in BACKUP Mode

When entering BACKUP mode, the PORT is powered off but the pin configuration is retained. When the device exits the BACKUP mode, the I/O line configuration can either be released or stretched, based on the I/O Retention bit in the CTRLA register (CTRLA.IORET).

- If IORET=0 when exiting BACKUP mode, the I/O lines configuration is released and driven by the reset value of the PORT.
- If the IORET=1 when exiting BACKUP mode, the configuration of the I/O lines is retained until the IORET bit is written to 0. It allows the I/O lines to be retained until the application has programmed the PORT.

20.6.3.5 Performance Level

The application can change the performance level on the fly writing to the by Performance Level Select bit in the Performance Level Configuration register (PLCFG.PLSEL).

When changing to a lower performance level, the bus frequency must be reduced before writing PLCFG.PLSEL in order to avoid exceeding the limit of the target performance level.

When changing to a higher performance level, the bus frequency can be increased only after the Performance Level Ready flag in the Interrupt Flag Status and Clear (INTFLAG.PLRDY) bit set to '1', indicating that the performance level transition is complete.

After a reset, the device starts in the lowest PL (lowest power consumption and lowest max frequency). The application can then switch to another PL at anytime without any stop in the code execution. As shown in Figure 20-2, performance level transition is possible only when the device is in active mode.

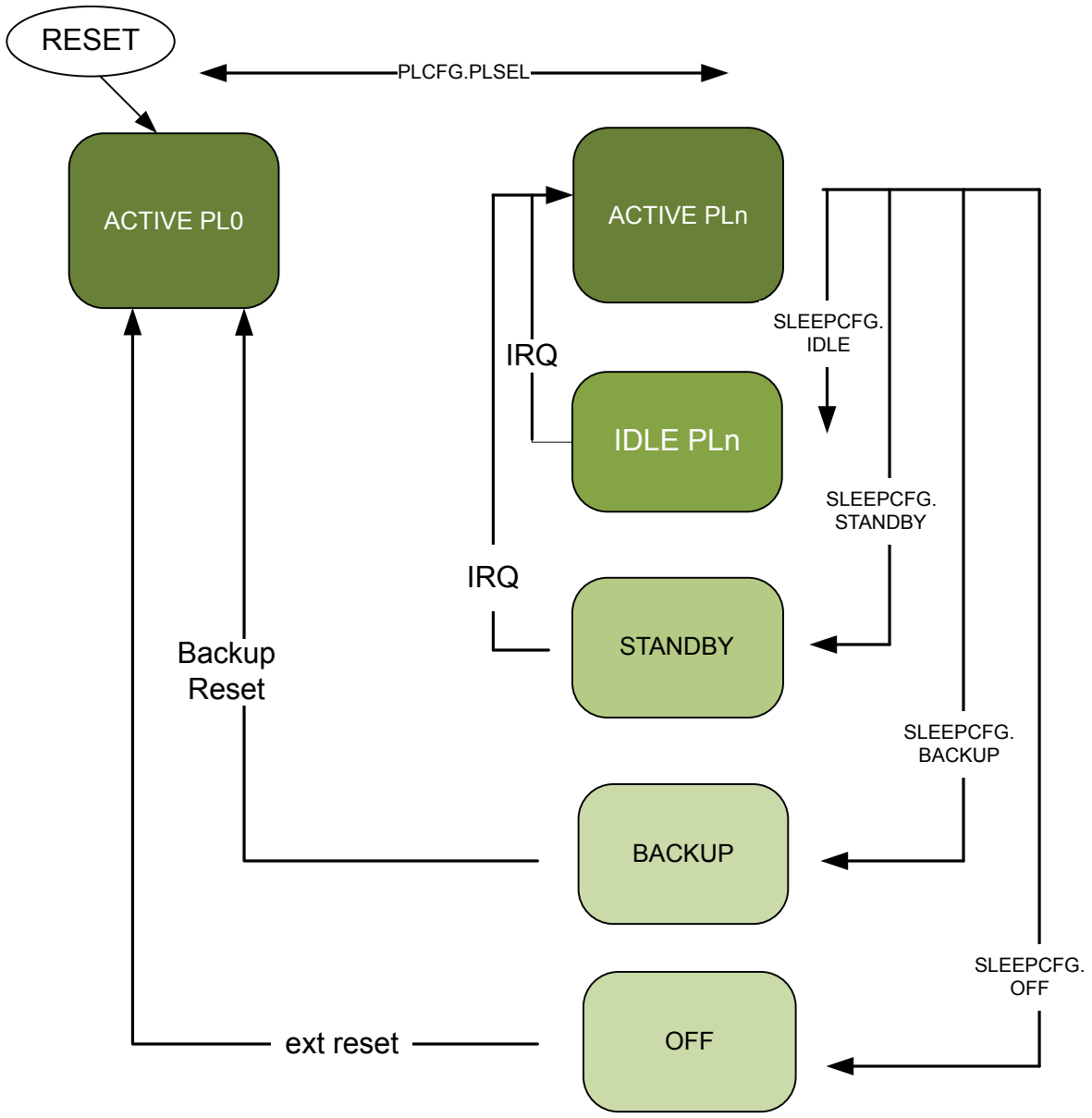
The Performance Level Disable bit in the Performance Level Configuration register (PLCFG.PLDIS) can be used to freeze the performance level to PL0. This disables the performance level hardware

mechanism in order to reduce both the power consumption and the wake-up startup time from standby sleep mode.

Note: This bit PLCFG.PLDIS must be changed only when the current performance level is PL0.

Any attempt to modify this bit while the performance level is not PL0 is discarded and a violation is reported to the PAC module. Any attempt to change the performance level to PLn (with n>0) while PLCFG.PLDIS=1 is discarded and a violation is reported to the PAC module.

Figure 20-2. Sleep Modes and Performance Level Transitions



20.6.3.6 Regulators, RAMs, and NVM State in Sleep Mode

By default, in standby sleep mode and backup sleep mode, the RAMs, NVM, and regulators are automatically set in low-power mode in order to reduce power consumption:

- The RAM is in low-power mode if the device is in standby mode. Refer to *RAM Automatic Low Power Mode* for details.
- Non-Volatile Memory - the NVM is automatically set in low power mode in these conditions:

  - When the device is in standby sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
  - When the device is in idle sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
- Regulators: by default, in standby sleep mode, the PM analyzes the device activity to use either the main or the low-power voltage regulator to supply the VDDCORE.

GCLK clocks, regulators and RAM are not affected in idle sleep mode and will operate as normal.

Table 20-3. Regulators, RAMs, and NVM state in Sleep Mode

Sleep Mode	SRAM Mode(1)	NVM	Regulators		
			VDDCORE		VDDBU
			main	ULP	
Active	normal	normal	on	on	on
Idle	auto(2)	on	on	on	on
Standby - case 1	normal	auto(2)	auto(3)	on	on
Standby - case 2	low power	low power	auto(3)	on	on
Standby - case 3	low power	low power	auto(3)	on	on
Standby - case 4	low power	low power	off	on	on
Backup	off	off	off	off	on
OFF	off	off	off	off	off

Note:

1. RAMs mode by default: STDBYCFG.BBIAS bits are set to their default value.
2. auto: by default, NVM is in low-power mode if not accessed.
3. auto: by default, the main voltage regulator is on if GCLK, APBx, or AHBx clock is running during SleepWalking.

Related Links

RAM Automatic Low Power Mode

Regulator Automatic Low Power Mode

20.6.4 Advanced Features

20.6.4.1 RAM Automatic Low Power Mode

The RAM is by default put in low power mode (back-biased) if the device is in standby sleep mode.

This behavior can be changed by configuring the Back Bias bit groups in the Standby Configuration register (STDBYCFG.BBIASxx), refer to the table below for details.

Note: In standby sleep mode, the RAM is put in low-power mode by default. This means that the RAM is back-biased, and the DMAC cannot access it. The DMAC can only access the RAM when it is not back biased (PM.STDBYCFG.BBIASxx=0x0).

Table 20-4. RAM Back-Biasing Mode

STBYCDFG.BBIASxx config		RAM
0x0	No Back Biasing mode	No Back Biasing in Standby mode
0x1	Standby Back Biasing mode	RAM is back-biased if the device is in standby sleep mode
0x2	Standby OFF mode	RAM is OFF if the device is in standby sleep mode
0x3	Always OFF mode	RAM is OFF if its power domain is in retention state

20.6.4.2 Regulator Automatic Low Power Mode

In standby mode, the PM selects either the main or the low power voltage regulator to supply the VDDCORE.

If a sleepwalking task is working on either asynchronous clocks (generic clocks) or synchronous clock (APB/AHB clocks), the main voltage regulator is used. This behavior can be changed by writing the Voltage Regulator Standby Mode bits in the Standby Configuration register (STDBYCFG.VREGSMOD). Refer to the following table for details.

Table 20-5. Regulator State in Sleep Mode

Sleep Modes	STDBYCFG. VREGSMOD	SleepWalking(1)	Regulator state for VDDCORE
Active	-	-	main voltage regulator
Idle	-	-	main voltage regulator
Standby	0x0: AUTO	NO	low power regulator
		YES	main voltage regulator
	0x1: PERFORMANCE	-	main voltage regulator
	0x2: LP(2)	-(2)	low power regulator

Note:

1. SleepWalking is running on GCLK clock or synchronous clock. This is not related to XOSC32K or OSCULP32K clocks.
2. Must only be used when SleepWalking is running on GCLK with 32KHz source.

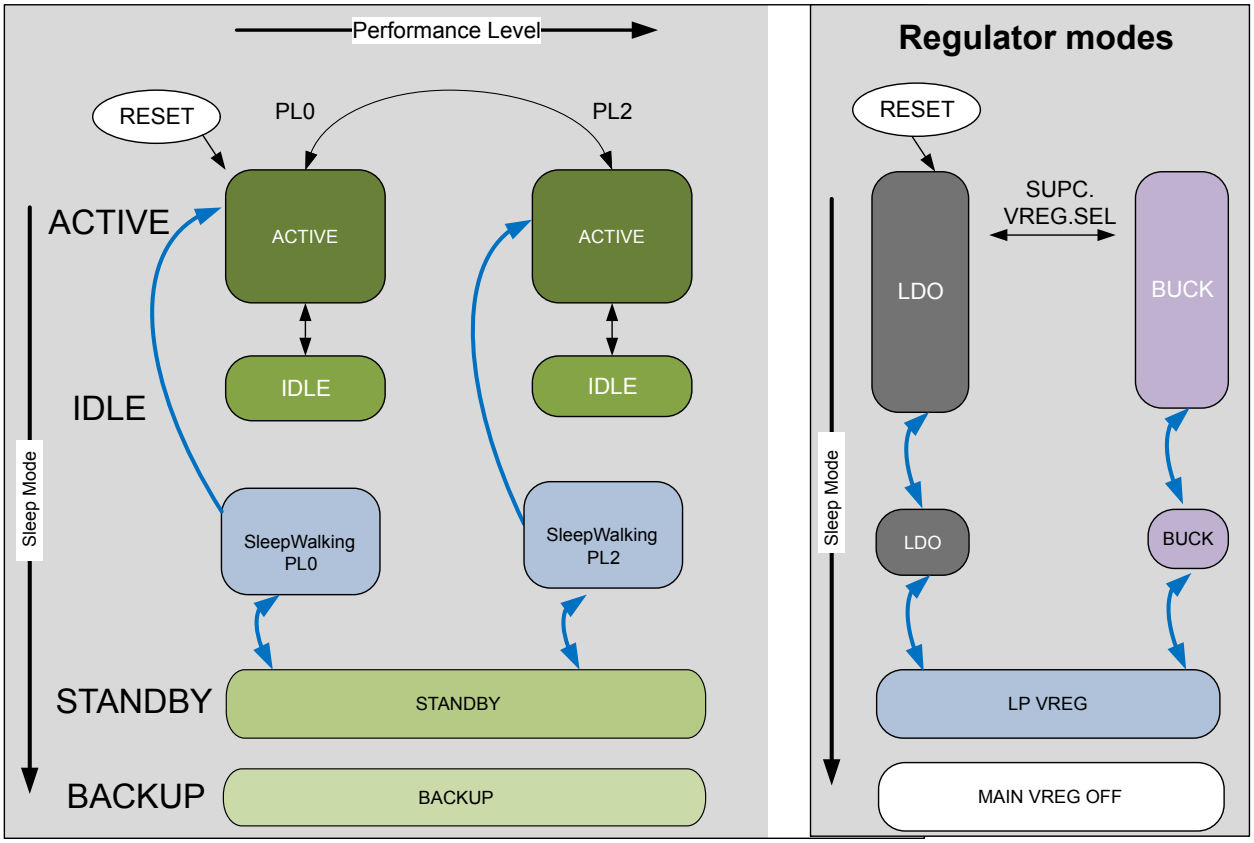
20.6.4.3 SleepWalking and Performance Level

SleepWalking is the capability for a device to temporarily wake up clocks for a peripheral to perform a task without waking up the CPU from STANDBY sleep mode. At the end of the sleepwalking task, the device can either be woken up by an interrupt (from a peripheral involved in SleepWalking) or enter again into STANDBY sleep mode. In this device, SleepWalking is supported only on GCLK clocks by using the on-demand clock principle of the clock sources.

In standby mode, when SleepWalking is ongoing, the performance level used to execute the sleepwalking task is the current configured performance level (used in active mode), and the main voltage regulator used to execute the sleepwalking task is the selected regulator used in active mode (LDO or Buck converter).

These are illustrated in the figure below.

Figure 20-3. Operating Conditions and SleepWalking



20.6.4.4 Wake-Up Time

The total wake-up time depends on:

- Latency due to Performance Level and Regulator effect:
  Performance Level has to be taken into account for the global wake-up time. As example, if PL2 is selected and the device is in standby sleep mode, the voltage level supplied by the ULP voltage regulator is lower than the one used in active mode. When the device wakes up, it takes a certain amount of time for the main regulator to transition to the voltage level corresponding to PL2, causing additional wake-up time.
- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM memory access.
- Latency due to Switchable Power Domain back-bias wake-up time:
  If back-bias is enabled, and the device wakes up from retention, it takes a certain amount of time for the regulator to settle.

20.6.5 DMA Operation

Not applicable.

20.6.6 Interrupts

The peripheral has the following interrupt sources:

- Performance Level Ready (PLRDY)
  This interrupt is a synchronous wake-up source. See Table 20-1 for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be

individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. Refer to the Nested Vector Interrupt Controller (NVIC) for details. If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

Related Links

Nested Vector Interrupt Controller

Interrupt Line Mapping

20.6.7 Events

Not applicable.

20.6.8 Sleep Mode Operation

The Power Manager is always active.

20.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0						IORET		
0x01	SLEEPCFG	7:0						SLEEPMODE[2:0]		
0x02	PLCFG	7:0	PLDIS						PLSEL[1:0]	
0x03	Reserved									
0x04	INTENCLR	7:0								PLRDY
0x05	INTENSET	7:0								PLRDY
0x06	INTFLAG	7:0								PLRDY
0x07	Reserved									
0x08	STDBYCFG	7:0	VREGSMOD[1:0]							
0x09		15:8					BBIASHS[1:0]			

20.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

20.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000a2f]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						IORET		
Access						R/W		
Reset						0		

Bit 2 – IORET: I/O Retention
Note: This bit is not reset by a backup reset.

Value	Description
0	After waking up from Backup mode, I/O lines are not held.
1	After waking up from Backup mode, I/O lines are held until IORET is written to 0.

20.8.2 Sleep Configuration

Name: SLEEPCFG
Offset: 0x01 [ID-00000a2f]
Reset: 0x2
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SLEEPMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SLEEPMODE[2:0]: Sleep Mode
Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software has to make sure the SLEEPCFG register reads the wanted value before issuing WFI instruction.

Value	Name	Definition
0x0	Reserved	Reserved
0x1	Reserved	Reserved
0x2	IDLE	CPU, AHBx, and APBx clocks are OFF
0x3	Reserved	Reserved
0x4	STANDBY	ALL clocks are OFF, unless requested by sleepwalking peripheral
0x5	BACKUP	Only Backup domain is powered ON
0x6	OFF	All power domains are powered OFF
0x7	Reserved	Reserved

20.8.3 Performance Level Configuration

Name: PLCFG
Offset: 0x02 [ID-00000a2f]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	PLDIS						PLSEL[1:0]	
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – PLDIS: Performance Level Disable
Disabling the automatic PL selection forces the device to run in PL0 , reducing the power consumption and the wake-up time from standby sleep mode.

Changing this bit when the current performance level is not PL0 is discarded and a violation is reported to the PAC module.

Value	Description
0	The Performance Level mechanism is enabled.
1	The Performance Level mechanism is disabled.

Bits 1:0 – PLSEL[1:0]: Performance Level Select

Value	Name	Definition
0x0	PL0	Performance Level 0
0x1	Reserved	Reserved
0x2	PL2	Performance Level 2
0x3	Reserved	Reserved

20.8.4 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x04 [ID-00000a2f]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY: Performance Level Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Performance Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled and will generate an interrupt request when the Performance Ready Interrupt Flag is set.

20.8.5 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x05 [ID-00000a2f]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY: Performance Level Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Performance Ready Interrupt Enable bit and enable the Performance Ready interrupt.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled.

20.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06 [ID-00000a2f]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY: Performance Level Ready
This flag is set when the performance level is ready and will generate an interrupt if INTENCLR/SET.PLRDY is '1'.

Writing a '1' to this bit has no effect.

Writing a '1' to this bit clears the Performance Ready interrupt flag.

20.8.7 Standby Configuration

Name: STDBYCFG
Offset: 0x08 [ID-00000a2f]
Reset: 0x0400
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					BBIASHS[1:0]			
Access					R	R		
Reset					0	0		

Bit	7	6	5	4	3	2	1	0
	VREGSMOD[1:0]							
Access	R	R						
Reset	0	0						

Bits 11:10 – BBIASHS[1:0]: Back Bias for HMCRAMCHS
Refer to Table 20-4 for details.

Value	Description
0	No Back Biasing in Standby mode
1	Back Biasing in Standby mode
2	Standby OFF mode
3	Always OFF mode

Bits 7:6 – VREGSMOD[1:0]: VREG Switching Mode
Refer to Regulator Automatic Low Power Mode for details.

Value	Name	Description
0x0	AUTO	Automatic Mode
0x1	PERFORMANCE	Performance oriented
0x2	LP	Low Power consumption oriented

21. OSCCTRL – Oscillators Controller

21.1 Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSC, OSC16M, DFLL48M, and FDPLL96M.

Through the interface registers, it is possible to enable, disable, calibrate, and monitor the OSCCTRL sub-peripherals.

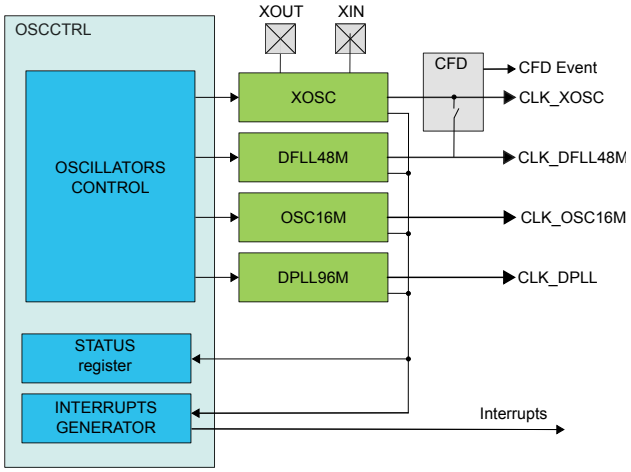
All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

21.2 Features

- 0.4-32MHz Crystal Oscillator (XOSC)
  - Tunable gain control
  - Programmable start-up time
  - Crystal or external input clock on XIN I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- 16MHz Internal Oscillator (OSC16M)
  - Fast startup
  - 4/8/12/16MHz output frequencies available
- Digital Frequency Locked Loop (DFLL48M)
  - Internal oscillator with no external components
  - 48MHz output frequency
  - Operates stand-alone as a high-frequency programmable oscillator in open loop mode
  - Operates as an accurate frequency multiplier against a known frequency in closed loop mode
- Fractional Digital Phase Locked Loop (FDPLL96M)
  - 48MHz to 96MHz output frequency
  - 32kHz to 2MHz reference clock
  - A selection of sources for the reference clock
  - Adjustable proportional integral controller
  - Fractional part used to achieve 1/16th of reference clock step

21.3 Block Diagram

Figure 21-1. OSCCTRL Block Diagram



21.4 Signal Description

Signal	Description	Type
XIN	Multipurpose Crystal Oscillator or external clock generator input	Analog input
XOUT	Multipurpose Crystal Oscillator output	Analog output

The I/O lines are automatically selected when XOSC is enabled.

21.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

21.5.1 I/O Lines

I/O lines are configured by OSCCTRL when XOSC is enabled, and need no user configuration.

21.5.2 Power Management

The OSCCTRL can continue to operate in any sleep mode where the selected source clock is running. The OSCCTRL interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

21.5.3 Clocks

The OSCCTRL gathers controls for all device oscillators and provides clock sources to the Generic Clock Controller (GCLK). The available clock sources are: XOSC, OSC16M, DFLL48M, and FDPLL96M.

The OSCCTRL bus clock (CLK_OSCCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

The DFLL48M control logic uses the DFLL oscillator output, which is also asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

MCLK – Main Clock

Peripheral Clock Masking

21.5.4 DMA

Not applicable.

21.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the OSCCTRL interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

21.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System

21.5.7 Debug Operation

When the CPU is halted in debug mode the OSCCTRL continues normal operation. If the OSCCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

21.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

21.5.9 Analog Connections

The 0.4-32MHz crystal must be connected between the XIN and XOUT pins, along with any required load capacitors.

21.6 Functional Description

21.6.1 Principle of Operation

XOSC, OSC16M, DFLL48M, and FDPLL96M are configured via OSCCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The Status register gathers different status signals coming from the sub-peripherals controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

21.6.2 External Multipurpose Crystal Oscillator (XOSC) Operation

The XOSC can operate in two different modes:

- External clock, with an external clock signal connected to the XIN pin
- Crystal oscillator, with an external 0.4-32MHz crystal

The XOSC can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller.

At reset, the XOSC is disabled, and the XIN/XOUT pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSC is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN and XOUT pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN pin will be overridden and controlled by the OSCCTRL, while the XOUT pin can still be used as a GPIO pin.

The XOSC is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRL.ENABLE).

To enable XOSC as an external crystal oscillator, the XTAL Enable bit (XOSCCTRL.XTALEN) must written to '1'. If XOSCCTRL.XTALEN is zero, the external clock input on XIN will be enabled.

When in crystal oscillator mode (XOSCCTRL.XTALEN=1), the External Multipurpose Crystal Oscillator Gain (XOSCCTRL.GAIN) must be set to match the external crystal oscillator frequency. If the External Multipurpose Crystal Oscillator Automatic Amplitude Gain Control (XOSCCTRL.AMPGC) is '1', the oscillator amplitude will be automatically adjusted, and in most cases result in a lower power consumption.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRL.RUNSTDBY, XOSCCTRL.ONDEMAND, and XOSCCTRL.ENABLE. If XOSCCTRL.ENABLE=0, the XOSC will be always stopped. For XOSCCTRL.ENABLE=1, this table is valid:

Table 21-1. XOSC Sleep Behavior

CPU Mode	XOSCCTRL.RUNSTDBY	XOSCCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

After a hard reset, or when waking up from a sleep mode where the XOSC was disabled, the XOSC will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRL.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

The External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set once the external clock or crystal oscillator is stable and ready to be used as a clock source. An interrupt is generated on a zero-to-one transition on STATUS.XOSCRDY if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set.

Related Links

GCLK - Generic Clock Controller

21.6.3 Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC). The CFD detects failing operation of the XOSC clock with reduced latency, and allows to switch to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC in case of recovery. The safe clock is derived from the OSC16M oscillator with a configurable prescaler. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral. See the Sleep Behavior table above when this is the case.

The user interface registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can optionally trigger an interrupt or an event when a failure is detected.

Clock Failure Detection

The CFD is reset only at power-on (POR). The CFD does not monitor the XOSC clock when the oscillator is disabled (XOSCCTRL.ENABLE=0).

Before starting CFD operation, the user must start and enable the safe clock source (OSC16M oscillator).

CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (XOCCTRL.CFDEN). After starting or restarting the XOSC, the CFD does not detect failure until the start-up time has elapsed. The start-up time is configured by the Oscillator Start-Up Time in the External Multipurpose Crystal Oscillator Control register (XOSCCTRL.STARTUP). Once the XOSC Start-Up Time is elapsed, the XOSC clock is constantly monitored.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC. There must be at least one rising and one falling XOSC clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) and the Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) are set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) is set, an interrupt is generated as well. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated, too.

After a clock failure was issued the monitoring of the XOSC clock is continued, and the Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC activity.

Clock Switch

When a clock failure is detected, the XOSC clock is replaced by the safe clock in order to maintain an active clock during the XOSC clock failure. The safe clock source is the OSC16M oscillator clock. The safe clock source can be scaled down by a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.CLKSW) is set.

When the CFD has switched to the safe clock, the XOSC is not disabled. If desired, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations.

In the case the application can recover the XOSC, the application can switch back to the XOSC clock by writing a '1' to Switch Back Enable bit in the Clock Failure Control register (XOSCCTRL.SWBACK). Once the XOSC clock is switched back, the Switch Back bit (XOSCCTRL.SWBACK) is cleared by hardware.

Prescaler

The CFD has an internal configurable prescaler to generate the safe clock from the OSC16M oscillator. The prescaler size allows to scale down the OSC16M oscillator so the safe clock frequency is not higher than the XOSC clock frequency monitored by the CFD. The division factor is 2^P, with P being the value of the CFD Prescaler bits in the CFD Prescaler Register (CFDPRESC.CFDPRESC).

> **Example**
>
> For an external crystal oscillator at 0.4MHz and the OSC16M frequency at 16MHz, the CFDPRESC.CFDPRESC value should be set scale down by more than factor 16/0.4=80, e.g. to 128, for a safe clock of adequate frequency.

Event

If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

Sleep Mode

The CFD is halted depending on configuration of the XOSC and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

21.6.4 16MHz Internal Oscillator (OSC16M) Operation

The OSC16M is an internal oscillator operating in open-loop mode and generating 4, 8, 12, or 16MHz frequency. The OSC16M frequency is selected by writing to the Frequency Select field in the OSC16M register (OSC16MCTRL.FSEL). OSC16M is enabled by writing '1' to the Oscillator Enable bit in the OSC16M Control register (OSC16MCTRL.ENABLE), and disabled by writing a '0' to this bit. Frequency selection must be done when OSC16M is disabled.

After enabling OSC16M, the OSC16M clock is output as soon as the oscillator is ready (STATUS.OSC16MRDY=1). User must ensure that the OSC16M is fully disabled before enabling it by reading STATUS.OSC16MRDY=0.

After reset, OSC16M is enabled and serves as the default clock source at 4MHz.

OSC16M will behave differently in different sleep modes based on the settings of OSC16MCTRL.RUNSTDBY, OSC16MCTRL.ONDEMAND, and OSC16MCTRL.ENABLE. If OSC16MCTRL.ENABLE=0, the OSC16M will be always stopped. For OSC16MCTRL.ENABLE=1, this table is valid:

Table 21-2. OSC16M Sleep Behavior

CPU Mode	OSC16MCTRL.RUNSTDBY	OSC16MCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

OSC16M is used as a clock source for the generic clock generators. This is configured by the Generic Clock Generator Controller.

Related Links

GCLK - Generic Clock Controller

21.6.5 Digital Frequency Locked Loop (DFLL48M) Operation

The DFLL48M can operate in both open-loop mode and closed-loop mode. In closed-loop mode, a low-frequency clock with high accuracy should be used as the reference clock to get high accuracy on the output clock (CLK_DFLL48M).

The DFLL48M can be used as a source for the generic clock generators.

Related Links

GCLK - Generic Clock Controller

21.6.5.1 Basic Operation

Open-Loop Operation

After any reset, the open-loop mode is selected. When operating in open-loop mode, the output frequency of the DFLL48M clock, CLK_DFLL48M, will be determined by the values written to the DFLL Coarse Value bit group and the DFLL Fine Value bit group (DFLLVAL.COARSE and DFLLVAL.FINE) in the DFLL Value register. Using "DFLL48M COARSE CAL" value from the Non Volatile Memory Software Calibration Area in DFLL.COARSE helps to output a frequency close to 48MHz.

It is possible to change the values of DFLLVAL.COARSE and DFLLVAL.FINE while the DFLL48M is enabled and in use, and thereby to adjust the output frequency of CLK_DFLL48M.

Related Links

NVM User Row Mapping

Closed-Loop Operation

In closed-loop operation, the DFLL48M output frequency is continuously regulated against a precise reference clock of relatively low frequency. This will improve the accuracy and stability of the CLK_DFLL48M clock in comparison to the open-loop (free-running) configuration.

Before closed-loop operation can be enabled, the DFLL48M must be enabled and configured in the following way:

1. Enable and select a reference clock (CLK_DFLL48M_REF). CLK_DFLL48M_REF is Generic Clock Channel 0 (DFLL48M_Reference).
2. Select the maximum step size allowed for finding the Coarse and Fine values by writing the appropriate values to the DFLL Coarse Maximum Step and DFLL Fine Maximum Step bit groups (DFLLMUL.CSTEP and DFLLMUL.FSTEP) in the DFLL Multiplier register.
   A small step size will ensure low overshoot on the output frequency, but it will typically take longer until locking is achieved. A high value might give a large overshoot, but will typically provide faster locking.

   DFLLMUL.CSTEP and DFLLMUL.FSTEP should not be higher than 50% of the maximum value of DFLLVAL.COARSE and DFLLVAL.FINE, respectively.
3. Select the multiplication factor in the DFLL Multiply Factor bit group (DFLLMUL.MUL) in the DFLL Multiplier register.
   Note: When choosing DFLLMUL.MUL, the output frequency must not exceed the maximum frequency of the device.

If the target frequency is below the minimum frequency of the DFLL48M, the output frequency will be equal to the DFLL minimum frequency.

4. Start the closed loop mode by writing '1' to the DFLL Mode Selection bit in the DFLL Control register (DFLLCTRL.MODE). See Frequency Locking for details.

The frequency of CLK_DFLL48M ($F_{clkdfll48m}$) is given by:

$$F_{clkdfll48m} = DFLLMUL \cdot MUL \times F_{clkdfll48m_ref}$$

where $F_{clkdfll48m_ref}$ is the frequency of the reference clock (CLK_DFLL48M_REF).

Related Links

GCLK - Generic Clock Controller

Frequency Locking

After enabling closed-loop operation by writing DFLLCTRL.MODE=1, the Coarse Value and the Fine Value bit fields in the DFLL48M Value register (DFLLVAL.COARSE and DFLLVAL.FINE) are used as starting parameters for the locking procedure.

Note: DFLLVAL.COARSE and DFLLVAL.FINE are read-only in closed-loop mode, and are controlled by the frequency tuner to meet user specified frequency.

The frequency locking is divided into two stages: coarse and fine lock.

Coarse Lock. Starting from the original DFLLVAL.COARSE and DFLLVAL.FINE, the control logic quickly finds the correct value for DFLLVAL.COARSE and sets the output frequency to a value close to the correct frequency. On coarse lock, the DFLL Locked on Coarse Value bit (STATUS.DFLLLCKC) in the Status register will be set.

Fine Lock. In this stage, the control logic tunes the value in DFLLVAL.FINE so that the output frequency is very close to the desired frequency. On fine lock, the DFLL Locked on Fine Value bit (STATUS.DFLLLCKF) in the Status register will be set.

Interrupts are generated by STATUS.DFLLLCKC and STATUS.DFLLLCKF, if INTENSET.DFLLLCKC or INTENSET.DFLLLCKF, respectively, are written to '1'.

The accuracy of the output frequency depends on which locks are set.

Note: Writing DFLLVAL.COARSE to a value close to the final value before entering closed-loop mode will reduce the time needed to get a lock on Coarse.

For a DFLL48M output frequency of 48MHz, the bit field "DFLL48M COARSE CAL" in the NVM Software Calibration Area provides a matching value for DFLL.COARSE, and will start DFLL with a frequency close to 48MHz.

This procedure will reduce the locking time to only the DFLL Fine Lock time:

1. Load the "DFLL48M COARSE CAL" value from the NVM Software Calibration Area into the DFLL.COARSE bit field.
2. Enable the Bypass Coarse Lock (DFLLCTRL.BPLCKC=1).
3. Start DFLL close loop (DFLLCTRL.MODE=1).

Related Links

NVM User Row Mapping

NVM Software Calibration Area Mapping

Frequency Error Measurement
The ratio between CLK_DFLL48M_REF and CLK48M_DFLL is measured automatically when the DFLL48M is in closed-loop mode. The difference between this ratio and the value in DFLLMUL.MUL is stored in the DFLL Multiplication Ratio Difference bit group (DFLLVAL.DIFF) in the DFLL Value register.

The relative error of CLK_DFLL48M with respect to the target frequency is calculated as follows:

$$ERROR = \frac{\text{DFLLVAL.DIFF}}{\text{DFLLMUL.MUL}}$$

Drift Compensation
If the Stable DFLL Frequency bit (DFLLCTRL.STABLE) in the DFLL Control register is '0', the frequency tuner will automatically compensate for drift in the CLK_DFLL48M without losing either of the locks.

Note: This means that DFLLVAL.FINE can change after every measurement of CLK_DFLL48M.

The DFLLVAL.FINE value may overflow or underflow in closed-loop mode due to large drift/instability of the clock source reference, and the DFLL Out Of Bounds bit (STATUS.DFLLOOB) in the Status register will be set. After an Out of Bounds error condition, the user must rewrite DFLLMUL.MUL to ensure correct CLK_DFLL48M frequency.

A zero-to-one transition of STATUS.DFLLOOB will generate an interrupt, if the DFLL Out Of Bounds bit in the Interrupt Enable Set register (INTENSET.DFLLOOB) is '1'. This interrupt will also be set if the tuner is not able to lock on the correct Coarse value.

To avoid this out-of-bounds error, the reference clock must be stable; an external oscillator XOSC32K is recommended.

Reference Clock Stop Detection
If CLK_DFLL48M_REF stops or is running at a very low frequency (slower than CLK_DFLL48M/(2 * MUL_{MAX})), the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will be set.

Detecting a stopped reference clock can take a long time, in the order of 217 CLK_DFLL48M cycles.

When the reference clock is stopped, the DFLL48M will operate as if in open-loop mode. Closed-loop mode operation will automatically resume when the CLK_DFLL48M_REF is restarted.

A zero-to-one transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will generate an interrupt, if the DFLL Reference Clock Stopped bit in the Interrupt Enable Set register (INTENSET.DFLLRCS) is '1'.

21.6.5.2 Additional Features

Dealing with Settling Time in Closed-Loop Mode
The time from selecting a new CLK_DFLL48M output frequency until this frequency is output by the DFLL48M can be up to several microseconds. A small value in DFLLMUL.MUL can lead to instability in the DFLL48M locking mechanism, which can prevent the DFLL48M from achieving locks.

To avoid this, a chill cycle can be enabled, during which the CLK_DFLL48M frequency is not measured. The chill cycle is enabled by default, but can be disabled by writing '1' to the DFLL Chill Cycle Disable bit in the DFLL Control register (DFLLCTRL.CCDIS). Enabling chill cycles might double the lock time.

Another solution to this problem is using less strict lock requirements. This is called Quick Lock (QL). QL is enabled by default as well, but it can be disabled by writing '1' to the Quick Lock Disable bit in the DFLL Control register (DFLLCTRL.QLDIS). The Quick Lock might lead to a larger spread in the output frequency than chill cycles, but the average output frequency is the same.

USB Clock Recovery Module

USB Clock Recovery mode can be used to create the 48MHz USB clock from the USB Start Of Frame (SOF). This mode is enabled by writing a '1' to both the USB Clock Recovery Mode bit and the Mode bit in DFLL Control register (DFLLCTRL.USBCRM and DFLLCTRL.MODE).

The SOF signal from USB device will be used as reference clock (CLK_DFLL_REF), ignoring the selected generic clock reference. When the USB device is connected, a SOF will be sent every 1ms, thus DFLLVAL.MUX bits should be written to 0xBB80 to obtain a 48MHz clock.

In USB clock recovery mode, the DFLLCTRL.BPLCKC bit state is ignored, and the value stored in the DFLLVAL.COARSE will be used as final Coarse Value. The COARSE calibration value can be loaded from NVM OTP row by software. The locking procedure will also go instantaneously to the fine lock search.

The DFLLCTRL.QLDIS bit must be cleared and DFLLCTRL.CCDIS should be set to speed up the lock phase. The DFLLCTRL.STABLE bit state is ignored, an auto jitter reduction mechanism is used instead.

Wake from Sleep Modes

DFLL48M can optionally reset its lock bits when it is disabled. This is configured by the Lose Lock After Wake bit in the DFLL Control register (DFLLCTRL.LLAW).

If DFLLCTRL.LLAW is zero, the DFLL48M will be re-enabled and start running with the same configuration as before being disabled, even if the reference clock is not available. The locks will not be lost. After the reference clock has restarted, the fine lock tracking will quickly compensate for any frequency drift during sleep if DFLLCTRL.STABLE is zero.

If DFLLCTRL.LLAW is '1' when disabling the DFLL48M, the DFLL48M will lose all its locks, and needs to regain these through the full lock sequence.

Accuracy

There are three main factors that determine the accuracy of $F_{clkdfll48m}$. These can be tuned to obtain maximum accuracy when fine lock is achieved.

- Fine resolution. The frequency step between two Fine values. This is relatively smaller for higher output frequencies.
- Resolution of the measurement: If the resolution of the measured $F_{clkdfll48m}$ is low, i.e., the ratio between the CLK_DFLL48M frequency and the CLK_DFLL48M_REF frequency is small, the DFLL48M might lock at a frequency that is lower than the targeted frequency. It is recommended to use a reference clock frequency of 32KHz or lower to avoid this issue for low target frequencies.
- The accuracy of the reference clock.

21.6.6 Digital Phase Locked Loop (DPLL) Operation

The task of the DPLL is to maintain coherence between the input (reference) signal and the respective output frequency, CLK_DPLL, via phase comparison. The DPLL controller supports three independent sources of reference clocks:

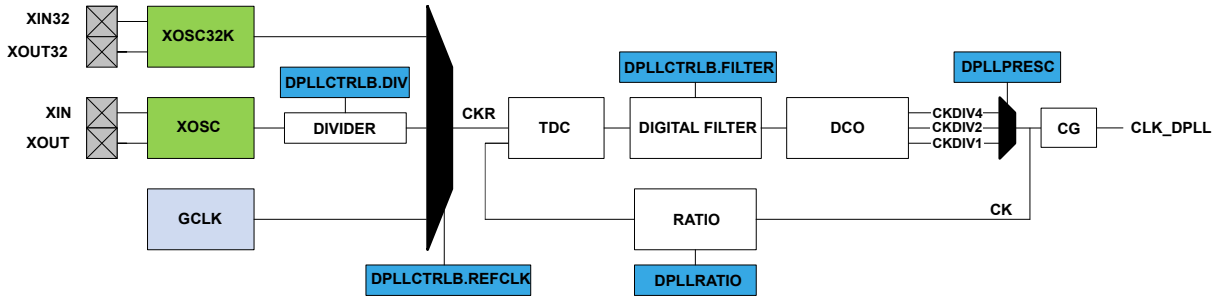
- XOSC32K: this clock is provided by the 32K External Crystal Oscillator (XOSC32K).
- XOSC: this clock is provided by the External Multipurpose Crystal Oscillator (XOSC).
- GCLK: this clock is provided by the Generic Clock Controller.

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is:

$$f_{CK} = f_{CKR} \times \left(LDR + 1 + \frac{LDRFRAC}{16}\right) \times \frac{1}{2^{PRESC}}$$

Where f_{CK} is the frequency of the DPLL output clock, LDR is the loop divider ratio integer part, LDRFRAC is the loop divider ratio fractional part, f_{CKR} is the frequency of the selected reference clock, and PRESC is the output prescaler value.

Figure 21-2. DPLL Block Diagram



When the controller is disabled, the output clock is low. If the Loop Divider Ratio Fractional part bit field in the DPLL Ratio register (DPLLRATIO.LDRFRAC) is zero, the DPLL works in integer mode. Otherwise, the fractional mode is activated. Note that the fractional part has a negative impact on the jitter of the DPLL.

Example (integer mode only): assuming F_{CKR} = 32kHz and F_{CK} = 48MHz, the multiplication ratio is 1500. It means that LDR shall be set to 1499.

Example (fractional mode): assuming F_{CKR} = 32kHz and F_{CK} = 48.006MHz, the multiplication ratio is 1500.1875 (1500 + 3/16). Thus LDR is set to 1499 and LDRFRAC to 3.

Related Links

GCLK - Generic Clock Controller

OSC32KCTRL – 32KHz Oscillators Controller

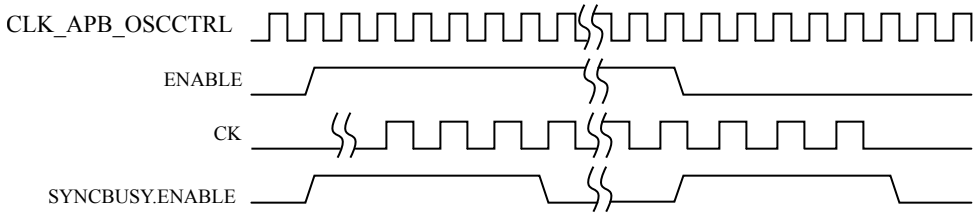
21.6.6.1 Basic Operation

Initialization, Enabling, Disabling, and Resetting

The DPLLC is enabled by writing a '1' to the Enable bit in the DPLL Control A register (DPLLCTRLA.ENABLE). The DPLLC is disabled by writing a zero to this bit.

The DPLLSYNCBUSY.ENABLE is set when the DPLLCTRLA.ENABLE bit is modified. It is cleared when the DPLL output clock CK has sampled the bit at the high level after enabling the DPLL. When disabling the DPLL, DPLLSYNCBUSY.ENABLE is cleared when the output clock is no longer running.

Figure 21-3. Enable Synchronization Busy Operation



The frequency of the DPLL output clock CK is stable when the module is enabled and when the Lock bit in the DPLL Status register is set (DPLLSTATUS.LOCK).

When the Lock Time bit field in the DPLL Control B register (DPLLCTRLB.LTIME) is non-zero, a user defined lock time is used to validate the lock operation. In this case the lock time is constant. If

DPLLCTRLB.LTIME=0, the lock signal is linked with the status bit of the DPLL, and the lock time varies depending on the filter selection and the final target frequency.

When the Wake Up Fast bit (DPLLCTRLB.WUF) is set, the wake up fast mode is activated. In this mode the clock gating cell is enabled at the end of the startup time. At this time the final frequency is not stable, as it is still during the acquisition period, but it allows to save several milliseconds. After first acquisition, the Lock Bypass bit (DPLLCTRLB.LBYPASS) indicates if the lock signal is discarded from the control of the clock gater (CG) generating the output clock CLK_DPLL.

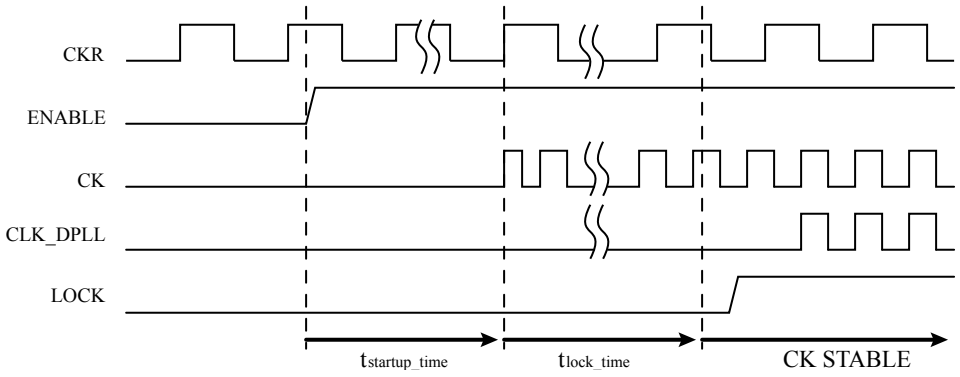
Table 21-3. CLK_DPLL Behavior from Startup to First Edge Detection

WUF	LTIME	CLK_DPLL Behavior
0	0	Normal Mode: First Edge when lock is asserted
0	Not Equal To Zero	Lock Timer Timeout mode: First Edge when the timer down-counts to 0.
1	X	Wake Up Fast Mode: First Edge when CK is active (startup time)

Table 21-4. CLK_DPLL Behavior after First Edge Detection

LBYPASS	CLK_DPLL Behavior
0	Normal Mode: the CLK_DPLL is turned off when lock signal is low.
1	Lock Bypass Mode: the CLK_DPLL is always running, lock is irrelevant.

Figure 21-4. CK and CLK_DPLL Output from DPLL Off Mode to Running Mode



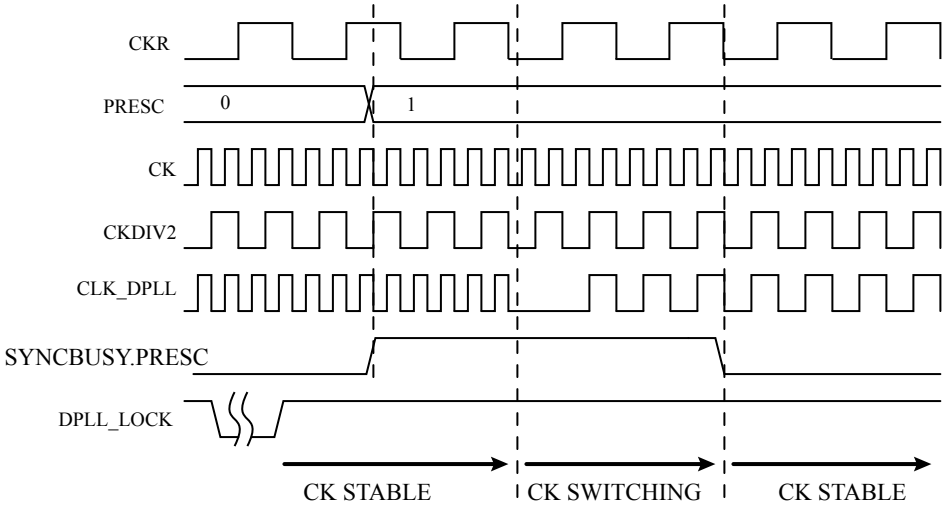
Reference Clock Switching

When a software operation requires reference clock switching, the recommended procedure is to turn the DPLL into the standby mode, modify the DPLLCTRLB.REFCLK to select the desired reference source, and activate the DPLL again.

Output Clock Prescaler

The DPLL controller includes an output prescaler. This prescaler provides three selectable output clocks CK, CKDIV2 and CKDIV4. The Prescaler bit field in the DPLL Prescaler register (DPLLPRESC.PRESC) is used to select a new output clock prescaler. When the prescaler field is modified, the DPLLSYNCBUSY.DPLLPRESC bit is set. It will be cleared by hardware when the synchronization is over.

Figure 21-5. Output Clock Switching Operation

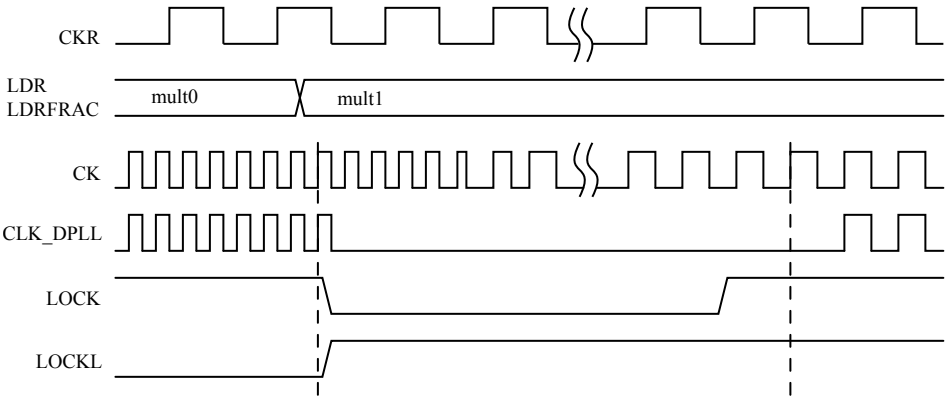


Loop Divider Ratio Updates

The DPLL Controller supports on-the-fly update of the DPLL Ratio Control (DPLLRATIO) register, allowing to modify the loop divider ratio and the loop divider ratio fractional part when the DPLL is enabled.

STATUS.DPLLLDRTO is set when the DPLLRATIO register has been modified and the DPLL analog cell has successfully sampled the updated value. At that time the DPLLSTATUS.LOCK bit is cleared and set again by hardware when the output frequency reached a stable state.

Figure 21-6. RATIOCTRL register update operation



Digital Filter Selection

The PLL digital filter (PI controller) is automatically adjusted in order to provide a good compromise between stability and jitter. Nevertheless a software operation can override the filter setting using the Filter bit field in the DPLL Control B register (DPLLCTRLB.FILTER). The Low Power Enable bit (DPLLCTRLB.LPEN) can be use to bypass the Time to Digital Converter (TDC) module.

21.6.7 DMA Operation

Not applicable.

21.6.8 Interrupts

The OSCCTRL has the following interrupt sources:

- XOSCRDY - Multipurpose Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSCRDY bit is detected

- CLKFAIL - Clock Failure. A 0-to-1 transition on the STATUS.CLKFAIL bit is detected
- OSC16MRDY - 16MHz Internal Oscillator Ready: A 0-to-1 transition on the STATUS.OSC16MRDY bit is detected
- DFLL-related:
  - DFLLRDY - DFLL48M Ready: A 0-to-1 transition of the STATUS.DFLLRDY bit is detected
  - DFLLOOB - DFLL48M Out Of Boundaries: A 0-to-1 transition of the STATUS.DFLLOOB bit is detected
  - DFLLLOCKF - DFLL48M Fine Lock: A 0-to-1 transition of the STATUS.DFLLLOCKF bit is detected
  - DFLLLOCKC - DFLL48M Coarse Lock: A 0-to-1 transition of the STATUS.DFLLLOCKC bit is detected
  - DFLLRCS - DFLL48M Reference Clock has Stopped: A 0-to-1 transition of the STATUS.DFLLRCS bit is detected
- DPLL-related:
  - DPLLLOCKR - DPLL Lock Rise: A 0-to-1 transition of the STATUS.DPLLLOCKR bit is detected
  - DPLLLOCKF - DPLL Lock Fall: A 0-to-1 transition of the STATUS.DPLLLOCKF bit is detected
  - DPLLLTTO - DPLL Lock Timer Time-out: A 0-to-1 transition of the STATUS.DPLLLTTO bit is detected
  - DPLLLDRTO - DPLL Loop Divider Ratio Update Complete. A 0-to-1 transition of the STATUS.DPLLLDRTO bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the OSCCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: The interrupts must be globally enabled for interrupt requests to be generated.

21.6.9 Events

The CFD can generate the following output event:

- Clock Failure (CLKFAIL): Generated when the Clock Failure status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the Clock Switch bit (STATUS.CLKSW) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.

21.6.10 Synchronization

DFLL48M

Due to the multiple clock domains, values in the DFLL48M control registers need to be synchronized to other clock domains.

Once the DFLL is enabled, any read and write operation requires the DFLL Ready bit in the Status register (STATUS.DFLLRDY) to read '1'.

Note: Once the DFLL48M is enabled in on-demand mode (DFLLCTRL.ONDEMAND=1), the STATUS.DFLLRDY bit will keep to '0' until the DFLL48M is requested by a peripheral.

Before writing to any of the DFLL48M control registers, the user must check that the DFLL Ready bit (STATUS.DFLLRDY) is set to '1'. When this bit is set, the DFLL48M can be configured and CLK_DFLL48M is ready to be used. Any write to any of the DFLL48M control registers while DFLLRDY is '0' will be ignored.

In order to read from the DFLLVAL register in closed loop mode, the user must request a read synchronization by writing a '1' to the Read Request bit in the DFLL Synchronization register (DFLLSYNC.READREQ). This is required because the DFLL controller may change the content of the DFLLVAL register any time. If a read operation is issued while the DFLL controller is updating the DFLLVAL content, a zero will be returned.

Note: Issuing a read on any register while a write-synchronization is still on-going will return a zero.

Read-Synchronized registers using DFLLSYNC.READREQ:

- DFLL48M Value register (DFLLVAL)

Write-Synchronized registers:

- DFLL48M Control register (DFLLCTRL)
- DFLL48M Value register (DFLLVAL)
- DFLL48M Multiplier register (DFLLMUL)

DPLL96M

Due to the multiple clock domains, some registers in the DPLL96M must be synchronized when accessed.

When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DPLLSYNCBUSY) will be set immediately, and cleared when synchronization is complete.

The following bits need synchronization when written:

- Enable bit in control register A (DPLLCTRLA.ENABLE)
- DPLL Ratio register (DPLLRATIO)
- DPLL Prescaler register (DPLLPRESC)

Related Links

Register Synchronization

21.7 Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0				OSC16MRDY			CLKFAIL	XOSCRDY
0x01		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x02		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x03		31:24								
0x04	INTENSET	7:0				OSC16MRDY			CLKFAIL	XOSCRDY
0x05		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x06		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x07		31:24								
0x08	INTFLAG	7:0				OSC16MRDY			CLKFAIL	XOSCRDY
0x09		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x0A		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x0B		31:24								
0x0C	STATUS	7:0				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
0x0D		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x0E		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x0F		31:24								
0x10	XOSCCTRL	7:0	ONDEMAND	RUNSTDBY		SWBACK	CFDEN	XTALEN	ENABLE	
0x11		15:8	STARTUP[3:0]				AMPGC	GAIN[2:0]		
0x12	CFDPRESC	7:0						CFDPRESC[2:0]		
0x13	EVCTRL	7:0								CFDEO
0x14	OSC16MCTRL	7:0	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
0x15 ... 0x17	Reserved									
0x18	DFLLCTRL	7:0	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	
0x19		15:8					WAITLOCK	BPLCKC	QLDIS	CCDIS
0x1A ... 0x1B	Reserved									
0x1C	DFLLVAL	7:0	FINE[7:0]							
0x1D		15:8	COARSE[5:0]						FINE[9:8]	
0x1E		23:16	DIFF[7:0]							
0x1F		31:24	DIFF[15:8]							
0x20	DFLLMUL	7:0	MUL[7:0]							
0x21		15:8	MUL[15:8]							
0x22		23:16	FSTEP[7:0]							
0x23		31:24	CSTEP[5:0]						FSTEP[9:8]	
0x24	DFLLSYNC	7:0	READREQ							
0x25 ... 0x27	Reserved									
0x28	DPLLCTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	
0x29 ...	Reserved									

Offset	Name	Bit Pos.								
0x2B										
0x2C	DPLLRATIO	7:0	LDR[7:0]							
0x2D		15:8					LDR[11:8]			
0x2E		23:16					LDRFRAC[3:0]			
0x2F		31:24								
0x30	DPLLCTRLB	7:0			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
0x31		15:8				LBYPASS		LTIME[2:0]		
0x32		23:16	DIV[7:0]							
0x33		31:24						DIV[10:8]		
0x34	DPLLPRESC	7:0							PRESC[1:0]	
0x35 ... 0x37	Reserved									
0x38	DPLLSYNCBUSY	7:0					DPLLPRESC	DPLLRATIO	ENABLE	
0x39 ... 0x3B	Reserved									
0x3C	DPLLSTATUS	7:0							CLKRDY	LOCK

21.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to the Register Access Protection section and the PAC - Peripheral Access Controller chapter for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" or "Write.Synchronized" property in each individual register description. Refer to the Synchronization section for details.

21.8.1 Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x04 [ID-00001eee]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			CLKFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Loop Ratio Update Complete Interrupt Enable bit, which enables the DPLL Loop Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO: DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Timeout Interrupt Enable bit, which enables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF: DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Fall Interrupt Enable bit, which enables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR: DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Rise Interrupt Enable bit, which enables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Reference Clock Stopped Interrupt Enable bit, which enables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLLCKC: DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Lock Coarse Interrupt Enable bit, which enables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 10 – DFLLLCKF: DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Lock Fine Interrupt Disable/Enable bit, disable the DFLL Lock Fine interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 9 – DFLLOOB: DFLL Out Of Bounds Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Out Of Bounds Interrupt Enable bit, which enables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 8 – DFLLRDY: DFLL Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Ready Interrupt Enable bit, which enables the DFLL Ready interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 4 – OSC16MRDY: OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the OSC16M Ready Interrupt Enable bit, which enables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 1 – CLKFAIL: XOSC Clock Failure Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the XOSC Clock Failure Interrupt Enable bit, which enables the XOSC Clock Failure Interrupt.

Value	Description
0	The XOSC Clock Failure Interrupt is disabled.
1	The XOSC Clock Failure Interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 0 – XOSCRDY: XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the XOSC Ready Interrupt Enable bit, which enables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

21.8.2 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x00 [ID-00001eee]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			CLKFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Loop Divider Ratio Update Complete Interrupt Enable bit, which disables the DPLL Loop Divider Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Divider Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO: DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Timeout Interrupt Enable bit, which disables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF: DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Fall Interrupt Enable bit, which disables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR: DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Rise Interrupt Enable bit, which disables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLLCKC: DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Lock Coarse Interrupt Enable bit, which disables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 10 – DFLLLCKF: DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Lock Fine Interrupt Enable bit, which disables the DFLL Lock Fine interrupt.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 9 – DFLLOOB: DFLL Out Of Bounds Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Out Of Bounds Interrupt Enable bit, which disables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 8 – DFLLRDY: DFLL Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Ready Interrupt Enable bit, which disables the DFLL Ready interrupt.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 4 – OSC16MRDY: OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the OSC16M Ready Interrupt Enable bit, which disables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 1 – CLKFAIL: Clock Failure Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the XOSC Clock Failure Interrupt Enable bit, which disables the XOSC Clock Failure interrupt.

Value	Description
0	The XOSC Clock Failure interrupt is disabled.
1	The XOSC Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 0 – XOSCRDY: XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

21.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08 [ID-00001eee]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			CLKFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Loop Divider Ratio Update Complete bit in the Status register (STATUS.DPLLLDRTO) and will generate an interrupt request if INTENSET.DPLLLDRTO is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Loop Divider Ratio Update Complete interrupt flag.

Bit 18 – DPLLLTO: DPLL Lock Timeout
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Timeout bit in the Status register (STATUS.DPLLLTO) and will generate an interrupt request if INTENSET.DPLLLTO is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Timeout interrupt flag.

Bit 17 – DPLLLCKF: DPLL Lock Fall
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Fall bit in the Status register (STATUS.DPLLLCKF) and will generate an interrupt request if INTENSET.DPLLLCKF is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Fall interrupt flag.

Bit 16 – DPLLLCKR: DPLL Lock Rise
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Rise bit in the Status register (STATUS.DPLLLCKR) and will generate an interrupt request if INTENSET.DPLLLCKR is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Rise interrupt flag.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) and will generate an interrupt request if INTENSET.DFLLRCS is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Reference Clock Stopped interrupt flag.

Bit 11 – DFLLLCKC: DFLL Lock Coarse
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Lock Coarse bit in the Status register (STATUS.DFLLLCKC) and will generate an interrupt request if INTENSET.DFLLLCKC is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Lock Coarse interrupt flag.

Bit 10 – DFLLLCKF: DFLL Lock Fine
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Lock Fine bit in the Status register (STATUS.DFLLLCKF) and will generate an interrupt request if INTENSET.DFLLLCKF is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Lock Fine interrupt flag.

Bit 9 – DFLLOOB: DFLL Out Of Bounds
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Out Of Bounds bit in the Status register (STATUS.DFLLOOB) and will generate an interrupt request if INTENSET.DFLLOOB is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Out Of Bounds interrupt flag.

Bit 8 – DFLLRDY: DFLL Ready
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Ready bit in the Status register (STATUS.DFLLRDY) and will generate an interrupt request if INTENSET.DFLLRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Ready interrupt flag.

Bit 4 – OSC16MRDY: OSC16M Ready
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the OSC16M Ready bit in the Status register (STATUS.OSC16MRDY) and will generate an interrupt request if INTENSET.OSC16MRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the OSC16M Ready interrupt flag.

Bit 1 – CLKFAIL: XOSC Failure Detection
This flag is cleared by writing '1' to it.

This flag is set on a 0-to-1 transition of the XOSC Clock Failure bit in the Status register (STATUS.CLKFAIL) and will generate an interrupt request if INTENSET.CLKFAIL is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the XOSC Clock Fail interrupt flag.

Bit 0 – XOSCRDY: XOSC Ready
This flag is cleared by writing '1' to it.

This flag is set on a 0-to-1 transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the XOSC Ready interrupt flag.

21.8.4 Status

Name: STATUS
Offset: 0x0C [ID-00001eee]
Reset: 0x00000100
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R	R	R	R	R
Reset				0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
Access				R		R	R	R
Reset				0		0	0	0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete

Value	Description
0	DPLL Loop Divider Ratio Update Complete not detected.
1	DPLL Loop Divider Ratio Update Complete detected.

Bit 18 – DPLLLTO: DPLL Lock Timeout

Value	Description
0	DPLL Lock time-out not detected.
1	DPLL Lock time-out detected.

Bit 17 – DPLLLCKF: DPLL Lock Fall

Value	Description
0	DPLL Lock fall edge not detected.
1	DPLL Lock fall edge detected.

Bit 16 – DPLLLCKR: DPLL Lock Rise

Value	Description
0	DPLL Lock rise edge not detected.
1	DPLL Lock fall edge detected.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped

Value	Description
0	DFLL reference clock is running.
1	DFLL reference clock has stopped.

Bit 11 – DFLLLCKC: DFLL Lock Coarse

Value	Description
0	No DFLL coarse lock detected.
1	DFLL coarse lock detected.

Bit 10 – DFLLLCKF: DFLL Lock Fine

Value	Description
0	No DFLL fine lock detected.
1	DFLL fine lock detected.

Bit 9 – DFLLOOB: DFLL Out Of Bounds

Value	Description
0	No DFLL Out Of Bounds detected.
1	DFLL Out Of Bounds detected.

Bit 8 – DFLLRDY: DFLL Ready

Value	Description
0	DFLL registers update is ongoing. Registers update is requested through DFLLSYNC.READREQ, or after a write access in DFLLCTRL, DFLLVAL or DFLLMUL register.
1	DFLL registers are stable and ready for read/write access.

Bit 4 – OSC16MRDY: OSC16M Ready

Value	Description
0	OSC16M is not ready.
1	OSC16M is stable and ready to be used as a clock source.

Bit 2 – CLKSW: XOSC Clock Switch

Value	Description
0	XOSC is not switched and provides the external clock or crystal oscillator clock.
1	XOSC is switched and provides the safe clock.

Bit 1 – CLKFAIL: XOSC Clock Failure

Value	Description
0	No XOSC failure detected.
1	A XOSC failure was detected.

Bit 0 – XOSCRDY: XOSC Ready

Value	Description
0	XOSC is not ready.
1	XOSC is stable and ready to be used as a clock source.

21.8.5 Clock Failure Detector Prescaler

Name: CFDPRESC
Offset: 0x12
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CFDPRESC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – CFDPRESC[2:0]: Clock Failure Detector Prescaler
These bits select the prescaler for the clock failure detector.

The OSC16M oscillator is used to clock the CFD prescaler. The CFD safe clock frequency is the OSC16M frequency divided by 2^CFDPRESC.

21.8.6 Event Control

Name: EVCTRL
Offset: 0x13
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								CFDEO
Access								R/W
Reset								0

Bit 0 – CFDEO: Clock Failure Detector Event Out
This bit indicates whether the Clock Failure detector event output is enabled or not and an output event will be generated when the Clock Failure detector detects a clock failure

Value	Description
0	Clock Failure detector event output is disabled and no event will be generated.
1	Clock Failure detector event output is enabled and an event will be generated.

21.8.7 16MHz Internal Oscillator (OSC16M) Control

Name: OSC16MCTRL
Offset: 0x14 [ID-00001eee]
Reset: 0x82
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	1	0			0	0	1	

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the oscillator to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the oscillator will only be running when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the OSC16M behaves during standby sleep mode.

Value	Description
0	The OSC16M is disabled in standby sleep mode if no peripheral requests the clock.
1	The OSC16M is not stopped in standby sleep mode. If ONDEMAND=1, the OSC16M will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bits 3:2 – FSEL[1:0]: Oscillator Frequency Selection
These bits control the oscillator frequency range.

Value	Description
0x00	4MHz
0x01	8MHz
0x10	12MHz
0x11	16MHz

Bit 1 – ENABLE: Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

21.8.8 External Multipurpose Crystal Oscillator (XOSC) Control

Name: XOSCCTRL
Offset: 0x10 [ID-00001eee]
Reset: 0x0080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	STARTUP[3:0]				AMPGC	GAIN[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		SWBACK	CFDEN	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bits 15:12 – STARTUP[3:0]: Start-Up Time
These bits select start-up time for the oscillator.

The OSCULP32K oscillator is used to clock the start-up counter.

Table 21-5. Start-Up Time for External Multipurpose Crystal Oscillator

STARTUP[3:0]	Number of OSCULP32K Clock Cycles	Number of XOSC Clock Cycles	Approximate Equivalent Time [µs]
0x0	1	3	31
0x1	2	3	61
0x2	4	3	122
0x3	8	3	244
0x4	16	3	488
0x5	32	3	977
0x6	64	3	1953

STARTUP[3:0]	Number of OSCULP32K Clock Cycles	Number of XOSC Clock Cycles	Approximate Equivalent Time [µs]
0x7	128	3	3906
0x8	256	3	7813
0x9	512	3	15625
0xA	1024	3	31250
0xB	2048	3	62500µs
0xC	4096	3	125000
0xD	8192	3	250000
0xE	16384	3	500000
0xF	32768	3	1000000

Note:

1. Actual startup time is 1 OSCULP32K cycle + 3 XOSC cycles.
2. The given time neglects the three XOSC cycles before OSCULP32K cycle.

Bit 11 – AMPGC: Automatic Amplitude Gain Control

Note: This bit must be set only after the XOSC has settled, indicated by the XOSC Ready flag in the Status register (STATUS.XOSCRDY).

Value	Description
0	The automatic amplitude gain control is disabled.
1	The automatic amplitude gain control is enabled. Amplitude gain will be automatically adjusted during Crystal Oscillator operation.

Bits 10:8 – GAIN[2:0]: Oscillator Gain

These bits select the gain for the oscillator. The listed maximum frequencies are recommendations, and might vary based on capacitive load and crystal characteristics. Those bits must be properly configured even when the Automatic Amplitude Gain Control is active.

Value	Recommended Max Frequency [MHz]
0x0	2
0x1	4
0x2	8
0x3	16
0x4	30
0x5-0x7	Reserved

Bit 7 – ONDEMAND: On Demand Control

The On Demand operation mode allows the oscillator to be enabled or disabled, depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the oscillator will be running only when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled, the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the XOSC behaves during standby sleep mode, together with the ONDEMAND bit:

Value	Description
0	The XOSC is not running in Standby sleep mode if no peripheral requests the clock.
1	The XOSC is running in Standby sleep mode. If ONDEMAND=1, the XOSC will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in Standby sleep mode.

Bit 4 – SWBACK: Clock Switch Back
This bit controls the XOSC output switch back to the external clock or crystal oscillator in case of clock recovery:

Value	Description
0	The clock switch back is disabled.
1	The clock switch back is enabled. This bit is reset once the XOSC putput clock is switched back to the external clock or crystal oscillator.

Bit 3 – CFDEN: Clock Failure Detector Enable
This bit controls the clock failure detector:

Value	Description
0	The Clock Failure Detector is disabled.
1	the Clock Failure Detector is enabled.

Bit 2 – XTALEN: Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator:

Value	Description
0	External clock connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN/XOUT.

Bit 1 – ENABLE: Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

21.8.9 DFLL48M Control

Name: DFLLCTRL
Offset: 0x18 [ID-00001eee]
Reset: 0x0080
Property: PAC Write-Protection, Write-Synchronized using STATUS.DFLLRDY=1

Bit	15	14	13	12	11	10	9	8
					WAITLOCK	BPLCKC	QLDIS	CCDIS
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	0	0	0	0	0	0	

Bit 11 – WAITLOCK: Wait Lock
This bit controls the DFLL output clock, depending on lock status.

Value	Description
0	Output clock before the DFLL is locked.
1	Output clock when DFLL is locked.

Bit 10 – BPLCKC: Bypass Coarse Lock
This bit controls the coarse lock procedure.

Value	Description
0	Bypass coarse lock is disabled.
1	Bypass coarse lock is enabled.

Bit 9 – QLDIS: Quick Lock Disable

Value	Description
0	Quick Lock is enabled.
1	Quick Lock is disabled.

Bit 8 – CCDIS: Chill Cycle Disable

Value	Description
0	Chill Cycle is enabled.
1	Chill Cycle is disabled.

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the DFLL will only be running when requested by a peripheral. If there is no peripheral requesting the DFLL clock source, the DFLL will be in a disabled state.

If On Demand is disabled, the DFLL will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DFLL is always on, if enabled.
1	The DFLL is enabled when a peripheral is requesting the DFLL to be used as a clock source. The DFLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the DFLL behaves during standby sleep mode:

Value	Description
0	The DFLL is disabled in standby sleep mode if no peripheral requests the clock.
1	The DFLL is not stopped in standby sleep mode. If ONDEMAND is one, the DFLL will be running when a peripheral is requesting the clock. If ONDEMAND is zero, the clock source will always be running in standby sleep mode.

Bit 5 – USBCRM: USB Clock Recovery Mode

Value	Description
0	USB Clock Recovery Mode is disabled.
1	USB Clock Recovery Mode is enabled.

Bit 4 – LLAW: Lose Lock After Wake

Value	Description
0	Locks will not be lost after waking up from sleep modes if the DFLL clock has been stopped.
1	Locks will be lost after waking up from sleep modes if the DFLL clock has been stopped.

Bit 3 – STABLE: Stable DFLL Frequency

Value	Description
0	FINE calibration tracks changes in output frequency.
1	FINE calibration register value will be fixed after a fine lock.

Bit 2 – MODE: Operating Mode Selection

Value	Description
0	The DFLL operates in open-loop operation.
1	The DFLL operates in closed-loop operation.

Bit 1 – ENABLE: DFLL Enable
Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRL.ENABLE will read back immediately after written.

Value	Description
0	The DFLL oscillator is disabled.
1	The DFLL oscillator is enabled.

21.8.10 DFLL48M Value

Name: DFLLVAL
Offset: 0x1C [ID-00001eee]
Reset: 0x00000000
Property: PAC Write-Protection, Read-Synchronized using DFLLSYNC.READREQ, Write-Synchronized using STATUS.DFLLRDY=1

Bit	31	30	29	28	27	26	25	24
	DIFF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIFF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COARSE[5:0]						FINE[9:8]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FINE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DIFF[15:0]: Multiplication Ratio Difference
In closed-loop mode (DFLLCTRL.MODE=1), this bit group indicates the difference between the ideal number of DFLL cycles and the counted number of cycles. In open-loop mode, this value is not updated and hence, invalid.

Bits 15:10 – COARSE[5:0]: Coarse Value
Set the value of the Coarse Calibration register. In closed-loop mode, this field is read-only.

Bits 9:0 – FINE[9:0]: Fine Value
Set the value of the Fine Calibration register. In closed-loop mode, this field is read-only.

21.8.11 DFLL48M Multiplier

Name: DFLLMUL
Offset: 0x20 [ID-00001eee]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized using STATUS.DFLLRDY=1

Bit	31	30	29	28	27	26	25	24
	CSTEP[5:0]						FSTEP[9:8]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FSTEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MUL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MUL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – CSTEP[5:0]: Coarse Maximum Step
This bit group indicates the maximum step size allowed during coarse adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 25:16 – FSTEP[9:0]: Fine Maximum Step
This bit group indicates the maximum step size allowed during fine adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 15:0 – MUL[15:0]: DFLL Multiply Factor
This field determines the ratio of the CLK_DFLL output frequency to the CLK_DFLL_REF input frequency. Writing to the MUL bits will cause locks to be lost and the fine calibration value to be reset to its midpoint.

21.8.12 DFLL48M Synchronization

Name: DFLLSYNC
Offset: 0x24 [ID-00001eee]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	READREQ							
Access	W							
Reset	0							

Bit 7 – READREQ: Read Request
To be able to read the current value of the DFLLVAL register in closed-loop mode, this bit must be written to '1'.

21.8.13 DPLL Control A

Name: DPLLCTRLA
Offset: 0x28 [ID-00001eee]
Reset: 0x80
Property: PAC Write-Protection, Write-Synchronized (ENABLE)

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	
Access	R/W	R/W					R/W	
Reset	1	0					0	

Bit 7 – ONDEMAND: On Demand Clock Activation
The On Demand operation mode allows the DPLL to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the DPLL will only be running when requested by a peripheral. If there is no peripheral requesting the DPLL's clock source, the DPLL will be in a disabled state.

If On Demand is disabled the DPLL will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DPLL is always on, if enabled.
1	The DPLL is enabled when a peripheral is requesting the DPLL to be used as a clock source. The DPLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the DPLL behaves during standby sleep mode:

Value	Description
0	The DPLL is disabled in standby sleep mode if no peripheral requests the clock.
1	The DPLL is not stopped in standby sleep mode. If ONDEMAND=1, the DPLL will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bit 1 – ENABLE: DPLL Enable
The software operation of enabling or disabling the DPLL takes a few clock cycles, so the DPLLSYNCBUSY.ENABLE status bit indicates when the DPLL is successfully enabled or disabled.

Value	Description
0	The DPLL is disabled.
1	The DPLL is enabled.

21.8.14 DPLL Ratio Control

Name: DPLLRATIO
Offset: 0x2C [ID-00001eee]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					LDRFRAC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					LDR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – LDRFRAC[3:0]: Loop Divider Ratio Fractional Part
Writing these bits selects the fractional part of the frequency multiplier. Due to synchronization there is a delay between writing these bits and the effect on the DPLL output clock. The value written will read back immediately and the DPLLRATIO bit in the DPLL Synchronization Busy register (DPLLSYNCBUSY.DPLLRATIO) will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

Bits 11:0 – LDR[11:0]: Loop Divider Ratio
Writing these bits selects the integer part of the frequency multiplier. The value written to these bits will read back immediately, and the DPLLRATIO bit in the DPLL Synchronization busy register (DPLLSYNCBUSY.DPLLRATIO), will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

21.8.15 DPLL Control B

Name: DPLLCTRLB
Offset: 0x30 [ID-00001eee]
Reset: 0x00
Property: Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
						DIV[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				LBYPASS		LTIME[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 26:16 – DIV[10:0]: Clock Divider
These bits set the XOSC clock division factor and can be calculated with following formula:

$$f_{DIV} = \frac{f_{XOSC}}{2x(DIV + 1)}$$

Bit 12 – LBYPASS: Lock Bypass

Value	Description
0	DPLL Lock signal drives the DPLL controller internal logic.
1	DPLL Lock signal is always asserted.

Bits 10:8 – LTIME[2:0]: Lock Time
These bits select the lock time-out value:

Value	Name	Description
0x0	Default	No time-out. Automatic lock.
0x1	Reserved	
0x2	Reserved	
0x3	Reserved	
0x4	8MS	Time-out if no lock within 8ms
0x5	9MS	Time-out if no lock within 9ms
0x6	10MS	Time-out if no lock within 10ms
0x7	11MS	Time-out if no lock within 11ms

Bits 5:4 – REFCLK[1:0]: Reference Clock Selection
Write these bits to select the DPLL clock reference:

Value	Name	Description
0x0	XOSC32K	XOSC32K clock reference
0x1	XOSC	XOSC clock reference

Value	Name	Description
0x2	GCLK	GCLK clock reference
0x3	Reserved	

Bit 3 – WUF: Wake Up Fast

Value	Description
0	DPLL clock is output after startup and lock time.
1	DPLL clock is output after startup time.

Bit 2 – LPEN: Low-Power Enable

Value	Description
0	The low-power mode is disabled. Time to Digital Converter is enabled.
1	The low-power mode is enabled. Time to Digital Converter is disabled. This will improve power consumption but increase the output jitter.

Bits 1:0 – FILTER[1:0]: Proportional Integral Filter Selection
These bits select the DPLL filter type:

Value	Name	Description
0x0	DEFAULT	Default filter mode
0x1	LBFILT	Low bandwidth filter
0x2	HBFILT	High bandwidth filter
0x3	HDFILT	High damping filter

21.8.16 DPLL Prescaler

Name: DPLLPRESC
Offset: 0x34 [ID-00001eee]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							PRESC[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – PRESC[1:0]: Output Clock Prescaler
These bits define the output clock prescaler setting.

Value	Name	Description
0x0	DIV1	DPLL output is divided by 1
0x1	DIV2	DPLL output is divided by 2
0x2	DIV4	DPLL output is divided by 4
0x3	Reserved	

21.8.17 DPLL Synchronization Busy

Name: DPLLSYNCBUSY
Offset: 0x38 [ID-00001eee]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
					DPLLPRESC	DPLLRATIO	ENABLE	
Access					R	R	R	
Reset					0	0	0	

Bit 3 – DPLLPRESC: DPLL Prescaler Synchronization Status

Value	Description
0	The DPLLRESC register has been synchronized.
1	The DPLLRESC register value has changed and its synchronization is in progress.

Bit 2 – DPLLRATIO: DPLL Loop Divider Ratio Synchronization Status

Value	Description
0	The DPLLRATIO register has been synchronized.
1	The DPLLRATIO register value has changed and its synchronization is in progress.

Bit 1 – ENABLE: DPLL Enable Synchronization Status

Value	Description
0	The DPLLCTRLA.ENABLE bit has been synchronized.
1	The DPLLCTRLA.ENABLE bit value has changed and its synchronization is in progress.

21.8.18 DPLL Status

Name: DPLLSTATUS
Offset: 0x3C [ID-00001eee]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							CLKRDY	LOCK
Access							R	R
Reset							0	0

Bit 1 – CLKRDY: Output Clock Ready

Value	Description
0	The DPLL output clock is off.
1	The DPLL output clock in on.

Bit 0 – LOCK: DPLL Lock status bit

Value	Description
0	The DPLL Lock signal is cleared, when the DPLL is disabled or when the DPLL is trying to reach the target frequency.
1	The DPLL Lock signal is asserted when the desired frequency is reached.

22. OSC32KCTRL – 32KHz Oscillators Controller

22.1 Overview

The 32KHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768kHz oscillators: XOSC32K and OSCULP32K.

The OSC32KCTRL sub-peripherals can be enabled, disabled, calibrated, and monitored through interface registers.

All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

22.2 Features

- 32.768kHz Crystal Oscillator (XOSC32K)
  - Programmable start-up time
  - Crystal or external input clock on XIN32 I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- 32.768kHz Ultra Low Power Internal Oscillator (OSCULP32K)
  - Ultra low power, always-on oscillator
  - Frequency fine tuning
- Calibration value loaded from Flash factory calibration at reset
- 1.024kHz clock outputs available

22.3 Block Diagram

22.4 Signal Description

Signal	Description	Type
XIN32	Analog Input	32.768kHz Crystal Oscillator or external clock generator input
XOUT32	Analog Output	32.768kHz Crystal Oscillator output

The I/O lines are automatically selected when XOSC32K is enabled.

Note: The signal of the external crystal oscillator may affect the jitter of neighboring pads.

22.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

22.5.1 I/O Lines

I/O lines are configured by OSC32KCTRL when XOSC32K is enabled, and need no user configuration.

22.5.2 Power Management

The OSC32KCTRL will continue to operate in any sleep mode where a 32KHz oscillator is running as source clock. The OSC32KCTRL interrupts can be used to wake up the device from sleep modes.

Related Links

PM – Power Manager

22.5.3 Clocks

The OSC32KCTRL gathers controls for all 32KHz oscillators and provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), Segment Liquid Crystal Controller (SLCD) and Watchdog Timer (WDT).

The available clock sources are: XOSC32K and OSCULP32K.

The OSC32KCTRL bus clock (CLK_OSC32KCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

Related Links

Peripheral Clock Masking

22.5.4 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the OSC32KCTRL interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

22.5.5 Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System

22.5.6 Debug Operation

When the CPU is halted in debug mode, OSC32KCTRL will continue normal operation. If OSC32KCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

22.5.7 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

22.5.8 Analog Connections

The external 32.768kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors. For details on recommended oscillator characteristics and capacitor load, refer to the related links.

Related Links

Electrical Characteristics

22.6 Functional Description

22.6.1 Principle of Operation

XOSC32K and OSCULP32K are configured via OSC32KCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The STATUS register gathers different status signals coming from the sub-peripherals of OSC32KCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

22.6.2 32KHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in two different modes:

- External clock, with an external clock signal connected to XIN32
- Crystal oscillator, with an external 32.768kHz crystal connected between XIN32 and XOUT32

At reset, the XOSC32K is disabled, and the XIN32/XOUT32 pins can either be used as General Purpose I/O (GPIO) pins or by other peripherals in the system.

When XOSC32K is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN32 and XOUT32 pins are controlled by the OSC32KCTRL, and GPIO functions are overridden on both pins. When in external clock mode, the only XIN32 pin will be overridden and controlled by the OSC32KCTRL, while the XOUT32 pin can still be used as a GPIO pin.

The XOSC32K is enabled by writing a '1' to the Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.ENABLE=1). The XOSC32K is disabled by writing a '0' to the Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.ENABLE=0).

To enable the XOSC32K as a crystal oscillator, the XTALEN bit in the 32KHz External Crystal Oscillator Control register must be set (XOSC32K.XTALEN=1). If XOSC32K.XTALEN is '0', the external clock input will be enabled.

The XOSC32K 32.768kHz output is enabled by setting the 32KHz Output Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.EN32K=1). The XOSC32K also has a 1.024kHz clock output. This is enabled by setting the 1KHz Output Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.EN1K=1).

It is also possible to lock the XOSC32K configuration by setting the Write Lock bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.WRTLOCK=1). If set, the XOSC32K configuration is locked until a Power-On Reset (POR) is detected.

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.RUNSTDBY, XOSC32K.ONDEMAND, and XOSC32K.ENABLE. If XOSC32KCTRL.ENABLE=0, the XOSC32K will be always stopped. For XOS32KCTRL.ENABLE=1, this table is valid:

Table 22-1. XOSC32K Sleep Behavior

CPU Mode	XOSC32KCTRL.RUNSTDBY	XOSC32KCTRL.ONDEMAND	Sleep Behavior of XOSC32K and CFD
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

As a crystal oscillator usually requires a very long start-up time, the 32KHz External Crystal Oscillator will keep running across resets when XOSC32K.ONDEMAND=0, except for power-on reset (POR). After a reset or when waking up from a sleep mode where the XOSC32K was disabled, the XOSC32K will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP) in the 32KHz External Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the XOSC32K Ready bit in the Status register is set (STATUS.XOSC32KRDY=1). The transition of STATUS.XOSC32KRDY from '0' to '1' generates an interrupt if the XOSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.XOSC32KRDY=1).

The XOSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (XOSC32K.EN32K or XOSC32K.EN1K) in order to ensure proper operation. In the same way, the GCLK or RTC modules must be disabled before the clock selection is changed. For details on RTC clock configuration, refer also to Real-Time Counter Clock Selection.

Related Links

GCLK - Generic Clock Controller

RTC – Real-Time Counter

22.6.3 Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC32K). The CFD detects failing operation of the XOSC32K clock with reduced latency, and allows to switch to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC32K in case of recovery. The safe clock is derived from the OSCULP32K oscillator with a configurable prescaler. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral. See the Sleep Behavior table above when this is the case.

The user interface registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can optionally trigger an interrupt or an event when a failure is detected.

Clock Failure Detection

The CFD is reset only at power-on (POR). The CFD does not monitor the XOSC32K clock when the oscillator is disabled (XOSC32K.ENABLE=0).

Before starting CFD operation, the user must start and enable the safe clock source (OSCULP32K oscillator).

CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (CFDCTRL.CFDEN). After starting or restarting the XOSC32K, the CFD does not detect failure until the start-up time has elapsed. The start-up time is configured by the Oscillator Start-Up Time in the External Multipurpose Crystal Oscillator Control register (XOSC32K.STARTUP). Once the XOSC32K Start-Up Time is elapsed, the XOSC32K clock is constantly monitored.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC32K. There must be at least one rising and one falling XOSC32K clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) and the Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) are set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) is set, an interrupt is generated as well. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated, too.

After a clock failure was issued the monitoring of the XOSC32K clock is continued, and the Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC32K activity.

Clock Switch

When a clock failure is detected, the XOSC32K clock is replaced by the safe clock in order to maintain an active clock during the XOSC32K clock failure. The safe clock source is the OSCULP32K oscillator clock. Both 32KHz and 1KHz outputs of the XOSC32K are replaced by the respective OSCULP32K 32KHz and 1KHz outputs. The safe clock source can be scaled down by a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC32K clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.CLKSW) is set.

When the CFD has switched to the safe clock, the XOSC32K is not disabled. If desired, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations. In the case the application can recover the XOSC32K, the application can switch back to the XOSC32K clock by writing a '1' to Switch Back Enable bit in the Clock Failure Control register (CFDCTRL.SWBACK). Once the XOSC32K clock is switched back, the Switch Back bit (CFDCTRL.SWBACK) is cleared by hardware.

Prescaler

The CFD has an internal configurable prescaler to generate the safe clock from the OSCULP32K oscillator. The prescaler size allows to scale down the OSCULP32K oscillator so the safe clock frequency is not higher than the XOSC32K clock frequency monitored by the CFD. The maximum division factor is 2.

The prescaler is applied on both outputs (32KHz and 1KHz) of the safe clock.

> **Example**
>
> For an external crystal oscillator at 32KHz and the OSCULP32K frequency is 32KHz, the XOSC32K.CFDPRESC should be set to 0 for a safe clock of equal frequency.

Event

If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

Sleep Mode

The CFD is halted depending on configuration of the XOSC32K and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

22.6.4 32KHz Ultra Low Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra-low-power clock source. The OSCULP32K is factory-calibrated under typical voltage and temperature conditions.

The OSCULP32K is enabled by default after a power-on reset (POR) and will always run except during POR. The frequency of the OSCULP32K oscillator is controlled by the value in the 32KHz Ultra Low Power Internal Oscillator Calibration bits in the 32KHz Ultra Low Power Internal Oscillator Control register (OSCULP32K.CALIB). This data is used to compensate for process variations.

OSCULP32K.CALIB is automatically loaded from Flash Factory Calibration during start-up. The calibration value can be overridden by the user by writing to OSCULP32K.CALIB.

It is also possible to lock the OSCULP32K configuration by setting the Write Lock bit in the 32KHz Ultra Low Power Internal Oscillator Control register (OSCULP32K.WRTLOCK=1). If set, the OSCULP32K configuration is locked until a power-on reset (POR) is detected.

The OSCULP32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). To ensure proper operation, the GCLK or RTC modules must be disabled before the clock selection is changed.

Related Links

RTC – Real-Time Counter

Real-Time Counter Clock Selection

GCLK - Generic Clock Controller

22.6.5 Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1.024kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

Related Links

WDT – Watchdog Timer

22.6.6 Real-Time Counter Clock Selection

Before enabling the RTC module, the RTC clock must be selected first. All oscillator outputs are valid as RTC clock. The selection is done in the RTC Control register (RTCCTRL). To ensure a proper operation, it is highly recommended to disable the RTC module first, before the RTC clock source selection is changed.

Related Links

RTC – Real-Time Counter

22.6.7 SLCD Clock Selection

Before enabling the SLCD module, the SLCD clock must be selected first. the 32.768kHz outputs of OSCULP32K and XOSC32K are valid as SLCD clock. The selection is done by the SLCD Selection bit in the SLCD Control register (SLCDCTRL.SLCDSEL).

To ensure proper operation, it is highly recommended to first disable the SLCD module before the SLCD clock source is selected.

changed.

Related Links

Clocks

22.6.8 Interrupts

The OSC32KCTRL has the following interrupt sources:

- XOSC32KRDY - 32KHz Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected
- CLKFAIL - Clock Failure Detector: A 0-to-1 transition on the STATUS.CLKFAIL bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be enabled individually by setting the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the OSC32KCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSC32KCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

PM – Power Manager

Nested Vector Interrupt Controller

22.6.9 Events

The CFD can generate the following output event:

- Clock Failure Detector (CLKFAIL): Generated when the Clock Failure Detector status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the Clock Switch bit (STATUS.SWBACK) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.

22.7 Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0						CLKFAIL		XOSC32KRDY
0x01		15:8								
0x02		23:16								
0x03		31:24								
0x04	INTENSET	7:0						CLKFAIL		XOSC32KRDY
0x05		15:8								
0x06		23:16								
0x07		31:24								
0x08	INTFLAG	7:0						CLKFAIL		XOSC32KRDY
0x09		15:8								
0x0A		23:16								
0x0B		31:24								
0x0C	STATUS	7:0					CLKSW	CLKFAIL		XOSC32KRDY
0x0D		15:8								
0x0E		23:16								
0x0F		31:24								
0x10	RTCCTRL	7:0						RTCSEL[2:0]		
0x11	SLCDCTRL	7:0								SLCDSEL
0x12 ... 0x13	Reserved									
0x14	XOSC32K	7:0	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
0x15		15:8				WRTLOCK		STARTUP[2:0]		
0x16	CFDCTRL	7:0						CFDPRESC	SWBACK	CFDEN
0x17	EVCTRL	7:0								CFDEO
0x18 ... 0x1B	Reserved									
0x1C	OSCULP32K	7:0								
0x1D		15:8	WRTLOCK			CALIB[4:0]				

22.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Optional Write-Protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in the register description. Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

22.8.1 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x00 [ID-00001010]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL: XOSC32K Clock Failure Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC32K Clock Failure Interrupt Enable bit, which disables the XOSC32K Clock Failure interrupt.

Value	Description
0	The XOSC32K Clock Failure Detection is disabled.
1	The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY: XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.8.2 Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x04 [ID-00001010]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL: XOSC32K Clock Failure Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the XOSC32K Clock Failure Interrupt Enable bit, which enables the XOSC32K Clock Failure interrupt.

Value	Description
0	The XOSC32K Clock Failure Detection is disabled.
1	The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

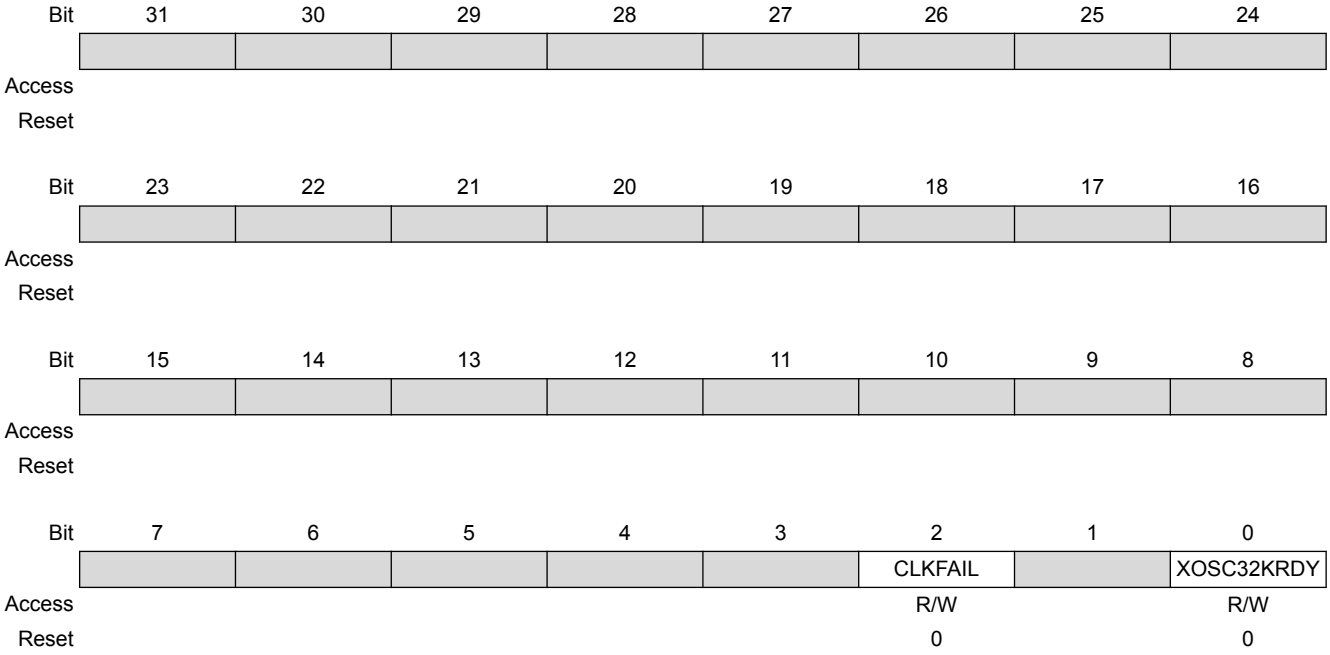
Bit 0 – XOSC32KRDY: XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the XOSC32K Ready Interrupt Enable bit, which enables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08 [ID-00001010]
Reset: 0x00000000
Property: –



Bit 2 – CLKFAIL: XOSC32K Clock Failure Detection
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the XOSC32K Clock Failure Detection bit in the Status register (STATUS.CLKFAIL) and will generate an interrupt request if INTENSET.CLKFAIL is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC32K Clock Failure Detection flag.

Bit 0 – XOSC32KRDY: XOSC32K Ready
This flag is cleared by writing a '1' to it.

This flag is set by a zero-to-one transition of the XOSC32K Ready bit in the Status register (STATUS.XOSC32KRDY), and will generate an interrupt request if INTENSET.XOSC32KRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the XOSC32K Ready interrupt flag.

22.8.4 Status

Name: STATUS
Offset: 0x0C [ID-00001010]
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CLKSW	CLKFAIL		XOSC32KRDY
Access					R	R		R
Reset					0	0		0

Bit 3 – CLKSW: XOSC32K Clock Switch

Value	Description
0	XOSC32K is not switched and provided the crystal oscillator.
1	XOSC32K is switched to be provided by the safe clock.

Bit 2 – CLKFAIL: XOSC32K Clock Failure Detector

Value	Description
0	XOSC32K is passing failure detection.
1	XOSC32K is not passing failure detection.

Bit 0 – XOSC32KRDY: XOSC32K Ready

Value	Description
0	XOSC32K is not ready.
1	XOSC32K is stable and ready to be used as a clock source.

22.8.5 RTC Clock Selection Control

Name: RTCCTRL
Offset: 0x10 [ID-00001010]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						RTCSEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – RTCSEL[2:0]: RTC Clock Selection
These bits select the source for the RTC.

Value	Name	Description
0x0	ULP1K	1.024kHz from 32KHz internal ULP oscillator
0x1	ULP32K	32.768kHz from 32KHz internal ULP oscillator
0x2, 0x3	Reserved	-
0x4	XOSC1K	1.024kHz from 32KHz external oscillator
0x5	XOSC32K	32.768kHz from 32KHz external crystal oscillator
0x6	Reserved	
0x7	Reserved	

22.8.6 SLCD Clock Selection Control

Name: SLCDCTRL
Offset: 0x11
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								SLCDSEL
Access								R/W
Reset								0

Bit 0 – SLCDSEL: SLCD Clock Source Selection
This bit selects the clock source for the SLCD

Value	Name	Description
0	ULP32K	32.768kHz from 32KHz internal ULP oscillator
1	XOSC32K	32.768kHz from external oscillator

22.8.7 32KHz External Crystal Oscillator (XOSC32K) Control

Name: XOSC32K
Offset: 0x14 [ID-00001010]
Reset: 0x00000080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
				WRTLOCK		STARTUP[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bit 12 – WRTLOCK: Write Lock
This bit locks the XOSC32K register for future writes, effectively freezing the XOSC32K configuration.

Value	Description
0	The XOSC32K configuration is not locked.
1	The XOSC32K configuration is locked.

Bits 10:8 – STARTUP[2:0]: Oscillator Start-Up Time
These bits select the start-up time for the oscillator.

The OSCULP32K oscillator is used to clock the start-up counter.

Table 22-2. Start-Up Time for 32KHz External Crystal Oscillator

STARTUP[2:0]	Number of OSCULP32K Clock Cycles	Number of XOSC32K Clock Cycles	Approximate Equivalent Time [s]
0x0	2048	3	0.06
0x1	4096	3	0.13
0x2	16384	3	0.5
0x3	32768	3	1
0x4	65536	3	2
0x5	131072	3	4
0x6	262144	3	8
0x7	-	-	Reserved

Note:
1. Actual Start-Up time is 1 OSCULP32K cycle + 3 XOSC32K cycles.
2. The given time assumes an XTAL frequency of 32.768kHz.

Bit 7 – ONDEMAND: On Demand Control
This bit controls how the XOSC32K behaves when a peripheral clock request is detected. For details, refer to XOSC32K Sleep Behavior.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the XOSC32K behaves during standby sleep mode. For details, refer to XOSC32K Sleep Behavior.

Bit 4 – EN1K: 1KHz Output Enable

Value	Description
0	The 1KHz output is disabled.
1	The 1KHz output is enabled.

Bit 3 – EN32K: 32KHz Output Enable

Value	Description
0	The 32KHz output is disabled.
1	The 32KHz output is enabled.

Bit 2 – XTALEN: Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator.

Value	Description
0	External clock connected on XIN32. XOUT32 can be used as general-purpose I/O.
1	Crystal connected to XIN32/XOUT32.

Bit 1 – ENABLE: Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

22.8.8 Clock Failure Detector Control

Name: CFDCTRL
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CFDPRESC	SWBACK	CFDEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – CFDPRESC: Clock Failure Detector Prescaler
This bit selects the prescaler for the Clock Failure Detector.

Value	Description
0	The CFD safe clock frequency is the OSCULP32K frequency
1	The CFD safe clock frequency is the OSCULP32K frequency divided by 2

Bit 1 – SWBACK: Clock Switch Back
This bit clontrols the XOSC32K output switch back to the external clock or crystal scillator in case of clock recovery.

Value	Description
0	The clock switch is disabled.
1	The clock switch is enabled. This bit is reset when the XOSC32K output is switched back to the external clock or crystal oscillator.

Bit 0 – CFDEN: Clock Failure Detector Enable
This bit selects the Clock Failure Detector state.

Value	Description
0	The CFD is disabled.
1	The CFD is enabled.

22.8.9 Event Control

Name: EVCTRL
Offset: 0x17
Reset: 0x0
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								CFDEO
Access								R/W
Reset								0

Bit 0 – CFDEO: Clock Failure Detector Event Out
This bit controls whether the Clock Failure Detector event output is enabled and an event will be generated when the CFD detects a clock failure.

Value	Description
0	Clock Failure Detector Event output is disabled, no event will be generated.
1	Clock Failure Detector Event output is enabled, an event will be generated.

22.8.10 32KHz Ultra Low Power Internal Oscillator (OSCULP32K) Control

Name: OSCULP32K
Offset: 0x1C [ID-00001010]
Reset: 0x0000XX06
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	WRTLOCK			CALIB[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – WRTLOCK: Write Lock
This bit locks the OSCULP32K register for future writes to fix the OSCULP32K configuration.

Value	Description
0	The OSCULP32K configuration is not locked.
1	The OSCULP32K configuration is locked.

Bits 12:8 – CALIB[4:0]: Oscillator Calibration
These bits control the oscillator calibration.

These bits are loaded from Flash Calibration at startup.

23. SUPC – Supply Controller

23.1 Overview

The Supply Controller (SUPC) manages the voltage reference, power supply, and supply monitoring of the device. It is also able to control two output pins.

The SUPC controls the voltage regulators for the core (VDDCORE) and backup (VDDBU) domains. It sets the voltage regulators according to the sleep modes, or the user configuration. In active mode, the voltage regulators can be selected on the fly between LDO (low-dropout) type regulator or Buck converter.

The SUPC supports connection of a battery backup to the VBAT power pin. It includes functionality that enables automatic power switching between main power and battery backup power. This ensures power to the backup domain when the main battery or power source is unavailable.

The SUPC embeds two Brown-Out Detectors. BOD33 monitors the voltage applied to the device (VDD or VBAT) and BOD12 monitors the internal voltage to the core (VDDCORE). The BOD can monitor the supply voltage continuously (continuous mode) or periodically (sampling mode).

The SUPC generates also a selectable reference voltage which can be used by analog modules like the ADC.

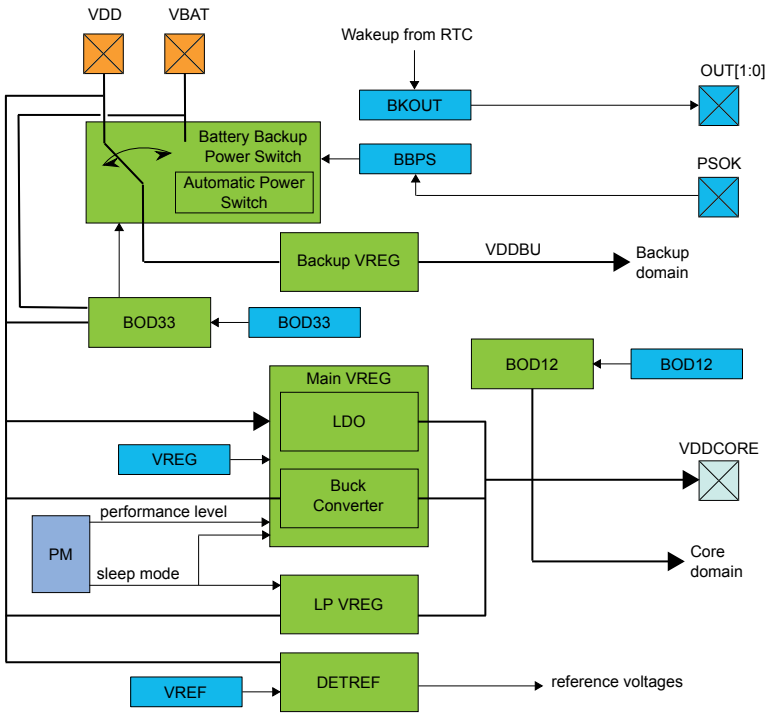
23.2 Features

- Voltage Regulator System
  - Main voltage regulator: LDO or Buck Converter in active mode (MAINVREG)
  - Low Power voltage regulator in standby mode (LPVREG)
  - Backup voltage regulator for backup domains
  - Adjustable VDDCORE to the sleep mode or the performance level
  - Controlled VDDCORE voltage slope when changing VDDCORE
- Battery Backup Power Switch
  - Automatic switching from main power to battery backup power
    - Automatic entry to backup mode when switched to battery backup power
  - Automatic switching from battery backup power to main power
    - Automatic exit from backup mode when switched back to main power
    - Stay in backup mode when switched back to main power
  - Main power request upon wake-up sources from backup mode
- Voltage Reference System
  - Reference voltage for ADC
- 3.3V Brown-Out Detector (BOD33)
  - Programmable threshold
  - Threshold value loaded from NVM User Row at startup
  - Triggers resets, interrupts, or Battery Backup Power Switch. Action loaded from NVM User Row
  - Operating modes:
    - Continuous mode

- Sampled mode for low power applications with programmable sample frequency
  - Hysteresis value from Flash User Calibration
  - Monitor VDD or VBAT
- 1.2V Brown-Out Detector (BOD12)
  - Internal non-configurable Brown-Out Detector
- Output pins
  - Pin toggling on RTC event

23.3 Block Diagram

Figure 23-1. SUPC Block Diagram



23.4 Signal Description

Signal Name	Type	Description
OUT[1:0]	Digital Output	SUPC Outputs
PSOK	Digital Input	Main Power Supply OK

One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

23.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

23.5.1 I/O Lines

I/O lines are configured by SUPC either when the SUPC output (signal OUT) is enabled or when the PSOK input is enabled. The I/O lines need no user configuration.

23.5.2 Power Management

The SUPC can operate in all sleep modes except backup sleep mode. BOD33 and Battery backup Power Switch can operate in backup mode.

Related Links

PM – Power Manager

23.5.3 Clocks

The SUPC bus clock (CLK_SUPC_APB) can be enabled and disabled in the Main Clock module.

A 32KHz clock, asynchronous to the user interface clock (CLK_SUPC_APB), is required to run BOD33 and BOD12 in sampled mode. Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

OSC32KCTRL – 32KHz Oscillators Controller

Peripheral Clock Masking

23.5.4 DMA

Not applicable.

23.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the SUPC interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

23.5.6 Events

Not applicable.

23.5.7 Debug Operation

When the CPU is halted in debug mode, the SUPC continues normal operation. If the SUPC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

If debugger cold-plugging is detected by the system, BOD33 and BOD12 resets will be masked. The BOD resets keep running under hot-plugging. This allows to correct a BOD33 user level too high for the available supply.

23.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

Note: Not all registers with write-access can be write-protected.

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Related Links

PAC - Peripheral Access Controller

23.5.9 Analog Connections

Not applicable.

23.6 Functional Description

23.6.1 Voltage Regulator System Operation

23.6.1.1 Enabling, Disabling, and Resetting

The LDO main voltage regulator is enabled after any Reset. The main voltage regulator (MAINVREG) can be disabled by writing the Enable bit in the VREG register (VREG.ENABLE) to zero. The main voltage regulator output supply level is automatically defined by the performance level or the sleep mode selected in the Power Manager module.

Related Links

PM – Power Manager

23.6.1.2 Initialization

After a Reset, the LDO voltage regulator supplying VDDCORE is enabled.

23.6.1.3 Selecting a Voltage Regulator

In active mode, the type of the main voltage regulator supplying VDDCORE can be switched on the fly. The two alternatives are a LDO regulator and a Buck converter.

The main voltage regulator switching sequence:

- The user changes the value of the Voltage Regulator Selection bit in the Voltage Regulator System Control register (VREG.SEL)
- The start of the switching sequence is indicated by clearing the Voltage Regulator Ready bit in the STATUS register (STATUS.VREGRDY=0)
- Once the switching sequence is completed, STATUS.VREGRDY will read '1'

The Voltage Regulator Ready (VREGRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.VREGRDY bit.

23.6.1.4 Voltage Scaling Control

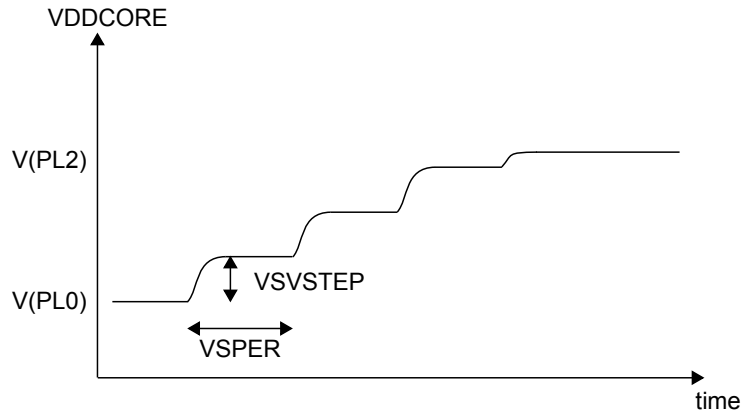
The VDDCORE supply will change under certain circumstances:

- When a new performance level (PL) is set
- When the standby sleep mode is entered or left
- When a sleepwalking task is requested in standby sleep mode

To prevent high peak current on the main power supply and to have a smooth transition of VDDCORE, both the voltage scaling step size and the voltage scaling frequency can be controlled: VDDCORE is changed by the selected step size of the selected period until the target voltage is reached.

The Voltage Scaling Voltage Step field is in the VREG register, VREG.VSVSTEP. The Voltage Scaling Period field is VREG.VSPER.

The following waveform shows an example of changing performance level from PL0 to PL2.



Setting VREG.VSVSTEP to the maximum value allows to transition in one voltage step.

The STATUS.VCORERDY bit is set to '1' as soon as the VDDCORE voltage has reached the target voltage. During voltage transition, STATUS.VCORERDY will read '0'. The Voltage Ready interrupt (VCORERDY) can be used to detect a 0-to-1 transition of STATUS.VCORERDY, see also Interrupts.

When entering the standby sleep mode and when no sleepwalking task is requested, the VDDCORE Voltage scaling control is not used.

23.6.1.5 Sleep Mode Operation

In standby mode, the low power voltage regulator (LPVREG) is used to supply VDDCORE.

When the Run in Standby bit in the VREG register (VREG.RUNSTDBY) is written to '1', VDDCORE is supplied by the main voltage regulator. Depending on the Standby in PL0 bit in the Voltage Regulator register (VREG.STDBYPL0), the VDDCORE level is either set to the PL0 voltage level, or remains in the current performance level.

Table 23-1. VDDCORE Level in Standby Mode

VREG.RUNSTDBY	VREG.STDBYPL0	VDDCORE Supply in Standby Mode
0	-	LPVREG
1	0	MAINVREG in current performance level(1)
1	1	MAINVREG in PL0

Note:

1. When the device is in PL0 but VREG.STDBYPL0=0, the MAINVREG is operating in normal power mode. To minimize power consumption, operate MAINVREG in PL0 mode by selecting VREG.STDBYPL0=1.

By writing the Low Power mode Efficiency bit in the VREG register (VREG.LPEFF) to '1', the efficiency of the regulator in LPVREG can be improved when the application uses a limited VDD range (2.5 to 3.6V). It is also possible to use the BOD33 in order to monitor the VDD and change this LPEFF value on the fly according to VDD level.

Related Links

Sleep Mode Controller

23.6.2 Voltage Reference System Operation

The reference voltages are generated by a functional block DETREF inside of the SUPC. DETREF is providing a fixed-voltage source, BANDGAP=1V, and a variable voltage, INTREF.

23.6.2.1 Initialization

The voltage reference output is disabled after any Reset.

23.6.2.2 Enabling, Disabling, and Resetting

The voltage reference output is enabled/disabled by setting/clearing the Voltage Reference Output Enable bit in the Voltage Reference register (VREF.VREFOE).

23.6.2.3 Selecting a Voltage Reference

The Voltage Reference Selection bit field in the VREF register (VREF.SEL) selects the voltage of INTREF to be applied to analog modules, e.g. the ADC.

23.6.2.4 Sleep Mode Operation

The Voltage Reference output behavior during sleep mode can be configured using the Run in Standby bit and the On Demand bit in the Voltage Reference register (VREF.RUNSTDBY, VREF.ONDEMAND), see the following table:

Table 23-2. VREF Sleep Mode Operation

VREF.ONDEMAND	VREF.RUNSTDBY	Voltage Reference Sleep behavior
-	-	Disable
0	0	Always run in all sleep modes *except* standby sleep mode
0	1	Always run in all sleep modes *including* standby sleep mode
1	0	Only run if requested by the ADC, in all sleep modes *except* standby sleep mode
1	1	Only run if requested by the ADC, in all sleep modes *including* standby sleep mode

23.6.3 Battery Backup Power Switch

23.6.3.1 Initialization

The Battery Backup Power Switch (BBPS) is disabled at power-up, and the backup domain is supplied by main power.

23.6.3.2 Forced Battery Backup Power Switch

The Backup domain is always supplied by the VBAT supply pin when the Configuration bit field in the Battery Backup Power Switch Control register (BBPS.CONF) is written to 0x2 (FORCED).

23.6.3.3 Automatic Battery Backup Power Switch

The supply of the backup domain can be switched automatically to VBAT supply pin by the Automatic Power Switch or by using the BOD33.

The supply of the backup domain can be switched automatically to VDD supply pin either by the Automatic Power Switch or the Main Power Pin when VDD and VDDCORE are restored.

Automatic Power Switch (APWS)

When the Configuration bit field in the Battery Backup Power Switch register (BBPS.CONF) is selecting the APWS, the Automatic Power Switch will function as Battery Backup Power Switch.

The Automatic Power switch allows to switch the supply of the backup domain from VDD to VBAT power and vice-versa.

When the Automatic Power Switch configuration is selected, the Automatic Power Switch Ready bit in the Status register (STATUS.APWSRDY) is set when the Automatic Power Switch is ready to operate. The

Automatic Power Switch Ready bit in the Interrupt Flag Status and Clear (INTFLAG.APSWRDY) will be set at the same time.

Related Links

Electrical Characteristics

BOD33 Power Switch

When the Configuration bit field in the Battery Backup Power Switch register (BBPS.CONF) are selecting the BOD33, BOD33 will function as Battery Backup Power Switch. In this case, when the VDD voltage is below the BOD33 threshold, the backup domain supply is switched to VBAT.

Main Power Supply OK (PSOK) Pin Enable

The state of the Main Power VDD can be used to switch between supply sources as long as the Battery Backup Power Switch is not configured as Automatic Power Switch (i.e., BBPS.CONF not set to APWS): when the Main Power Supply OK Pin Enable bit in the BBPS register is written to '1' (BBPS.PSOKEN), restoring VDD will form a low-to-high transition on the PSOK pin. This low-to-high transition will switch the Backup Power Supply back to VDD.

Note: With BBPS.PSOKEN=0 and BBPS.CONF not configured to APWS, the device can not be restarted.

Backup Battery Power Switch Status

The Battery Backup Power Switch bit in the Status register (STATUS.BBPS) indicates whether the backup domain is currently powered by VDD or VBAT.

23.6.3.4 Sleep Mode Operation

The Battery Backup Power Switch is not stopped in any sleep mode.

Entering Battery Backup Mode

Entering backup mode can be triggered by either:

- Wait-for-interrupt (WFI) instruction.
- Automatic Power Switch (BBPS.CONF=APWS). When the Automatic Power Switch detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode.
- BOD33 detection: When the BOD33 detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode. For this trigger, the following register configuration is required: BOD33.ACTION=BKUP, BOD33.VMON=VDD, and BBPS.CONF=BOD33.

Related Links

PM – Power Manager

Leaving Battery Backup Mode

Leaving backup mode can be triggered by either:

- RTC requests and externally triggered RSTC requests, under one of these conditions:
  - The Backup Domain is supplied by Main Power, and the Battery Backup Power Switch is *not* forced (BBPS.CONF not set to FORCED)
  - The Battery Backup Power Switch *is* forced (BBPS.CONF is FORCED)

  The device is kept in battery-powered backup mode until Main Power is restored to supply the device. Then, the backup domain will be powered by Main Power.
- Automatic Power Switch. Leaving backup mode will happen when Main Power is restored and the Battery Backup Power Switch configuration (BBPS.CONF) is set to APWS: When BBPS.WAKEEN=1, the device will leave backup mode and wake up.

When BBPS.WAKEEN=0, the backup domain will be powered by Main Power, but the device will stay in backup mode.

- PSOK pin. A low-to-high transition on PSOK will wake up the device if BBPS.PSOKEN=1, BBPS.WAKEEN=1, and the Battery Backup Power Switch is different from APWS (BBPS.CONF is not APWS).
  When BBPS.WAKEEN=0, the backup domain will be powered by Main Power, but the device will stay in backup mode.

23.6.4 Output Pins

The SUPC can drive two outputs. By writing a '1' to the corresponding Output Enable bit in the Backup Output Control register (BKOUT.EN), the OUTx pin is driven by the SUPC.

The OUT pin can be set by writing a '1' to the corresponding Set Output bit in the Backup Output Control register (BKOUT.SETx).

The OUT pin can be cleared by writing a '1' to the corresponding CLR bit (BKOUT.CLRx).

If a RTC Toggle Enable bit is written to '1' (BKOUT.RTCTGLx), the corresponding OUTx pin will toggle when an RTC event occurs.

23.6.5 Brown-Out Detectors

23.6.5.1 Initialization

Before a Brown-Out Detector (BOD33) is enabled, it must be configured, as outlined by the following:

- Set the BOD threshold level (BOD33.LEVEL)
- Set the configuration in active, standby, backup modes (BOD33.ACTCDG, BOD33.STDBYCFG, BODVDD.BKUP)
- Set the prescaling value if the BOD will run in sampling mode (BOD33.PSEL)
- Set the action and hysteresis (BOD33.ACTION and BOD33.HYST)

The BOD33 register is Enable-Protected, meaning that they can only be written when the BOD is disabled (BOD33.ENABLE=0 and SYNCBUSY.BOD33EN=0). As long as the Enable bit is '1', any writes to Enable-Protected registers will be discarded, and an APB error will be generated. The Enable bits are not Enable-Protected.

23.6.5.2 Enabling, Disabling, and Resetting

After power or user reset, the BOD33 and BOD12 register values are loaded from the NVM User Row.

The BODVDD is enabled by writing a '1' to the Enable bit in the BOD control register (BOD33.ENABLE). The BOD is disabled by writing a '0' to the BODVDD.ENABLE.

Related Links

NVM User Row Mapping

23.6.5.3 3.3V Brown-Out Detector (BOD33)

The 3.3V Brown-Out Detector (BOD33) is able to monitor either the VDD or the VBAT supply .

The Voltage Monitored bit in the BOD33 Control register (BOD33.VMON) selects which supply is monitored in active and standby mode. In backup mode, BOD33 will always monitor the supply of the backup domain, i.e. either VDD or VBAT.

If VDD is monitored, the BOD33 compares the voltage with the brown-out threshold level. This level is set in the BOD33 Level field in the BOD33 register (BOD33.LEVEL). This level is used in all modes except the backup sleep modes. In backup sleep modes, a different voltage reference is used, which is configured by the BOD33.BKUPLEVEL bits.

When VDD crosses below the brown-out threshold level, the BOD33 can generate either an interrupt, a Reset, or an Automatic Battery Backup Power Switch, depending on the BOD33 Action bit field (BOD33.ACTION).

If VBAT is monitored, the BOD33 compares the voltage with the brown-out threshold level set in the BOD33 Backup Level field in the BOD33 register (BOD33.BKUPLEVEL).

When VBAT crosses below the backup brown-out threshold level, the BOD33 can generate either an interrupt or a Reset.

The BOD33 detection status can be read from the BOD33 Detection bit in the Status register (STATUS.BOD33DET).

At start-up or at Power-On Reset (POR), the BOD33 register values are loaded from the NVM User Row.

Related Links

NVM User Row Mapping

23.6.5.4 1.2V Brown-Out Detector (BOD12)

The BOD12 is calibrated in production and its calibration configuration is stored in the NVM User Row. This configuration must not be changed to assure the correct behavior of the BOD12. The BOD12 generates a reset when 1.2V crosses below the preset brown-out level. The BODCORE is always disabled in standby sleep mode.

Related Links

NVM User Row Mapping

23.6.5.5 Continuous Mode

Continuous mode is the default mode for BOD33.

The BOD33 is continuously monitoring the supply voltage (VDD or VBAT, depending on BOD33.VMON) if it is enabled (BOD33.ENABLE=1) and if the BOD33 Configuration bit in the BOD33 register is cleared (BOD33.ACTCFG=0 for active mode, BOD33.STDBYCFG=0 for standby mode).

23.6.5.6 Sampling Mode

The Sampling Mode is a low-power mode where the BOD33 is being repeatedly enabled on a sampling clock's ticks. The BOD33 will monitor the supply voltage for a short period of time and then go to a low-power disabled state until the next sampling clock tick.

Sampling mode is enabled in Active mode for BOD33 by writing the ACTCFG bit (BOD33.ACTCFG=1). Sampling mode is enabled in Standby mode by writing to the STDBYCFG bit (BOD33.STBYCFG=1). The frequency of the clock ticks ($F_{clksampling}$) is controlled by the Prescaler Select bit groups in the BOD33 register (BOD33.PSEL).

$$F_{clksampling} = \frac{F_{clkprescaler}}{2^{(\mathrm{PSEL}+1)}}$$

The prescaler signal ($F_{clkprescaler}$) is a 1KHz clock, output by the 32KHz Ultra Low Power Oscillator OSCULP32K.

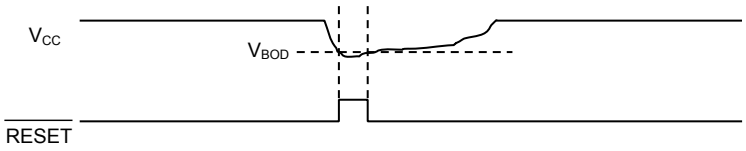
As the sampling clock is different from the APB clock domain, synchronization among the clocks is necessary. See also Synchronization.

23.6.5.7 Hysteresis

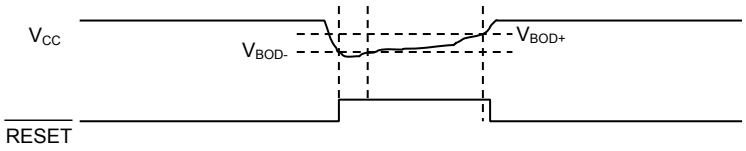
A hysteresis on the trigger threshold of a BOD will reduce the sensitivity to ripples on the monitored voltage: instead of switching $\overline{\text{RESET}}$ at each crossing of V_{BOD}, the thresholds for switching $\overline{\text{RESET}}$ on and off are separated (V_{BOD-} and V_{BOD+}, respectively).

Figure 23-2. BOD Hysteresis Principle

Hysteresis OFF:



Hysteresis ON:



Enabling the BOD33 hysteresis by writing the Hysteresis bit in the BOD33 register (BOD33.HYST) to '1' will add hysteresis to the BOD33 threshold level.

The hysteresis functionality can be used in both Continuous and Sampling Mode.

23.6.5.8 Sleep Mode Operation

Standby Mode

The BOD33 can be used in standby mode if the BOD is enabled and the corresponding Run in Standby bit is written to '1' (BOD33.RUNSTDBY).

The BOD33 can be configured to work in either Continuous or Sampling Mode by writing a '1' to the Configuration in Standby Sleep Mode bit (BOD33.STDBYCFG).

Backup Mode

In Backup mode, the BOD12 is automatically disabled.

If the BOD33 is enabled and the Run in Backup sleep mode bit in the BOD33 register (BOD33.RUNBKUP) is written to '1', the BOD33 will operate in Sampling mode. In this state, the voltage monitored by BOD33 is always the supply of the backup domain, i.e. VDD or VBAT.

23.6.6 Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- VDDCORE Voltage Ready (VCORERDY), asynchronous
- Automatic Power Switch Ready Ready (APSWRDY), asynchronous
- Voltage Regulator Ready (VREGRDY) asynchronous
- BOD33 Ready (BOD33RDY), synchronous
- BOD33 Detection (BOD33DET), asynchronous
- BOD33 Synchronization Ready (B33SRDY), synchronous

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or

the SUPC is reset. See the INTFLAG register for details on how to clear interrupt flags. The SUPC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller

Sleep Mode Controller

23.6.7 Synchronization

The prescaler counters that are used to trigger brown-out detections operate asynchronously from the peripheral bus. As a consequence, the BOD33 Enable bit (BOD33.ENABLE) need synchronization when written.

The Write-Synchronization of the Enable bit is triggered by writing a '1' to the Enable bit of the BOD33 Control register. The Synchronization Ready bit (STATUS.B33SRDY) in the STATUS register will be cleared when the Write-Synchronization starts, and set again when the Write-Synchronization is complete. Writing to the same register while the Write-Synchronization is ongoing (STATUS.B33SRDY is '0') will generate an error without stalling the APB bus.

23.7 Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0						B33SRDY	BOD33DET	BOD33RDY
0x01		15:8						VCORERDY	APWSRDY	VREGRDY
0x02		23:16								
0x03		31:24								
0x04	INTENSET	7:0						B33SRDY	BOD33DET	BOD33RDY
0x05		15:8						VCORERDY	APWSRDY	VREGRDY
0x06		23:16								
0x07		31:24								
0x08	INTFLAG	7:0						B33SRDY	BOD33DET	BOD33RDY
0x09		15:8						VCORERDY	APWSRDY	VREGRDY
0x0A		23:16								
0x0B		31:24								
0x0C	STATUS	7:0						B33SRDY	BOD33DET	BOD33RDY
0x0D		15:8					BBPS	VCORERDY	APWSRDY	VREGRDY
0x0E		23:16								
0x0F		31:24								
0x10	BOD33	7:0	RUNBKUP	RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
0x11		15:8	PSEL[3:0]					VMON		ACTCFG
0x12		23:16			LEVEL[5:0]					
0x13		31:24			BKUPLEVEL[5:0]					
0x14 ... 0x17	Reserved									
0x18	VREG	7:0		RUNSTDBY	STDBYPL0			SEL	ENABLE	
0x19		15:8								LPEFF
0x1A		23:16					VSVSTEP[3:0]			
0x1B		31:24	VSPER[7:0]							
0x1C	VREF	7:0	ONDEMAND	RUNSTDBY				VREFOE		
0x1D		15:8								
0x1E		23:16					SEL[3:0]			
0x1F		31:24								
0x20	BBPS	7:0					PSOKEN	WAKEEN	CONF[1:0]	
0x21		15:8								
0x22		23:16								
0x23		31:24								
0x24	BKOUT	7:0							EN[1:0]	
0x25		15:8							CLR[1:0]	
0x26		23:16							SET[1:0]	
0x27		31:24							RTCTGL[1:0]	
0x28	BKIN	7:0						BKIN[2:0]		
0x29		15:8								
0x2A		23:16								
0x2B		31:24								

23.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). PAC Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to Register Access Protection for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. Refer to Synchronization for details.

23.8.1 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x00 [ID-00001e33]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 10 – VCORERDY: VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the VDDCORE Ready Interrupt Enable bit, which disables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Bit 9 – APWSRDY: Automatic Power Switch Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Automatic Power Switch Ready Interrupt Enable bit, which disables the Automatic Power Switch Ready interrupt.

Value	Description
0	The Automatic Power Switch Ready interrupt is disabled.
1	The Automatic Power Switch Ready interrupt is enabled and an interrupt request will be generated when the APWSRDY Interrupt Flag is set.

Bit 8 – VREGRDY: Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Voltage Regulator Ready Interrupt Enable bit, which disables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 2 – B33SRDY: BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Synchronization Ready Interrupt Enable bit, which disables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET: BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Detection Interrupt Enable bit, which disables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY: BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

23.8.2 Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x04 [ID-00001e33]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 10 – VCORERDY: VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the VDDCORE Ready Interrupt Enable bit, which enables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Bit 9 – APWSRDY: Automatic Power Switch Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Automatic Power Switch Ready Interrupt Enable bit, which enables the Automatic Power Switch Ready interrupt.

Value	Description
0	The Automatic Power Switch Ready interrupt is disabled.
1	The Automatic Power Switch Ready interrupt is enabled and an interrupt request will be generated when the Automatic Power Switch Ready Interrupt Flag is set.

Bit 8 – VREGRDY: Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Voltage Regulator Ready Interrupt Enable bit, which enables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 2 – B33SRDY: BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Synchronization Ready Interrupt Enable bit, which enables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET: BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Detection Interrupt Enable bit, which enables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY: BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Ready Interrupt Enable bit, which enables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

23.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08 [ID-00001e33]
Reset: 0x0000010X - X= determined from NVM User Row (0xX=0bx00y)
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	1

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	y

Bit 10 – VCORERDY: VDDCORE Voltage Ready
This flag is cleared by writing a '1 to it.

This flag is set on a zero-to-one transition of the VDDCORE Ready bit in the Status register (STATUS.VCORERDY) and will generate an interrupt request if INTENSET.VCORERDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VCORERDY interrupt flag.

Bit 9 – APWSRDY: Automatic Power Switch Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the Automatic Power Switch Ready bit in the Status register (STATUS.APWSRDY) and will generate an interrupt request if INTENSET.APWSRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the APWSRDY interrupt flag.

Bit 8 – VREGRDY: Voltage Regulator Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the Voltage Regulator Ready bit in the Status register (STATUS.VREGRDY) and will generate an interrupt request if INTENSET.VREGRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VREGRDY interrupt flag.

Bit 2 – B33SRDY: BOD33 Synchronization Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Synchronization Ready bit in the Status register (STATUS.B33SRDY) and will generate an interrupt request if INTENSET.B33SRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Synchronization Ready interrupt flag.

Bit 1 – BOD33DET: BOD33 Detection
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Detection bit in the Status register (STATUS.BOD33DET) and will generate an interrupt request if INTENSET.BOD33DET=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Detection interrupt flag.

Bit 0 – BOD33RDY: BOD33 Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Ready bit in the Status register (STATUS.BOD33RDY) and will generate an interrupt request if INTENSET.BOD33RDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Ready interrupt flag.

The BOD33 can be enabled.

23.8.4 Status

Name: STATUS
Offset: 0x0C [ID-00001e33]
Reset: Determined from NVM User Row
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					BBPS	VCORERDY	APWSRDY	VREGRDY
Access					R	R	R	R
Reset					0	1	0	1

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R	R	R
Reset						0	0	y

Bit 11 – BBPS: Battery Backup Power Switch

Value	Description
0	the backup domain is supplied by VDD.
1	the backup domain is supplied by VBAT.

Bit 10 – VCORERDY: VDDCORE Voltage Ready

Value	Description
0	the VDDCORE voltage is not as expected.
1	the VDDCORE voltage is the target voltage.

Bit 9 – APWSRDY: Automatic Power Switch Ready

Value	Description
0	The Automatic Power Switch is not ready.
1	The Automatic Power Switch is ready.

Bit 8 – VREGRDY: Voltage Regulator Ready

Value	Description
0	The selected voltage regulator in VREG.SEL is not ready.
1	The voltage regulator selected in VREG.SEL is ready and the core domain is supplied by this voltage regulator.

Bit 2 – B33SRDY: BOD33 Synchronization Ready

Value	Description
0	BOD33 synchronization is ongoing.
1	BOD33 synchronization is complete.

Bit 1 – BOD33DET: BOD33 Detection

Value	Description
0	No BOD33 detection.
1	BOD33 has detected that the I/O power supply is going below the BOD33 reference value.

Bit 0 – BOD33RDY: BOD33 Ready
The BOD33 can be enabled at start-up from NVM User Row.

Value	Description
0	BOD33 is not ready.
1	BOD33 is ready.

23.8.5 3.3V Brown-Out Detector (BOD33) Control

Name: BOD33
Offset: 0x10 [ID-00001e33]
Reset: Determined from NVM User Row
Property: Write-Synchronized, Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
			BKUPLEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			LEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	PSEL[3:0]					VMON		ACTCFG
Access	R/W	R/W	R/W	R/W		R/W		R/W
Reset	0	0	0	0		0		0

Bit	7	6	5	4	3	2	1	0
	RUNBKUP	RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	y	y	0	z	

Bits 29:24 – BKUPLEVEL[5:0]: BOD33 Threshold Level on VBAT or in Backup Sleep Mode
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors VBAT or in backup sleep mode.

This bit field is not synchronized.

Bits 21:16 – LEVEL[5:0]: BOD33 Threshold Level on VDD
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors the VDD except in backup sleep mode.

These bits are loaded from NVM User Row at start-up.

This bit field is not synchronized.

Bits 15:12 – PSEL[3:0]: Prescaler Select
Selects the prescaler divide-by output for the BOD33 sampling mode. The input clock comes from the OSCULP32K 1KHz output.

Value	Name	Description
0x0	DIV2	Divide clock by 2
0x1	DIV4	Divide clock by 4
0x2	DIV8	Divide clock by 8
0x3	DIV16	Divide clock by 16
0x4	DIV32	Divide clock by 32
0x5	DIV64	Divide clock by 64
0x6	DIV128	Divide clock by 128
0x7	DIV256	Divide clock by 256
0x8	DIV512	Divide clock by 512
0x9	DIV1024	Divide clock by 1024
0xA	DIV2048	Divide clock by 2048
0xB	DIV4096	Divide clock by 4096
0xC	DIV8192	Divide clock by 8192
0xD	DIV16384	Divide clock by 16384
0xE	DIV32768	Divide clock by 32768
0xF	DIV65536	Divide clock by 65536

Bit 10 – VMON: Voltage Monitored in Active and Standby Mode
This bit is not synchronized.

Value	Description
0	The BOD33 monitors the VDD power pin in active and standby mode.
1	The BOD33 monitors the VBAT power pin in active and standby mode.

Bit 8 – ACTCFG: BOD33 Configuration in Active Sleep Mode
This bit is not synchronized.

Value	Description
0	In active mode, the BOD33 operates in continuous mode.
1	In active mode, the BOD33 operates in sampling mode.

Bit 7 – RUNBKUP: BOD33 Configuration in Backup Sleep Mode
This bit is not synchronized.

Value	Description
0	In backup sleep mode, the BOD33 is disabled.
1	In backup sleep mode, the BOD33 is enabled and configured in sampling mode.

Bit 6 – RUNSTDBY: Run in Standby
This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is disabled.
1	In standby sleep mode, the BOD33 is enabled.

Bit 5 – STDBYCFG: BOD33 Configuration in Standby Sleep Mode
If the RUNSTDBY bit is set to '1', the STDBYCFG bit sets the BOD33 configuration in standby sleep mode.

This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is enabled and configured in continuous mode.
1	In standby sleep mode, the BOD33 is enabled and configured in sampling mode.

Bits 4:3 – ACTION[1:0]: BOD33 Action
These bits are used to select the BOD33 action when the supply voltage crosses below the BOD33 threshold.

These bits are loaded from NVM User Row at start-up.

This bit field is not synchronized.

Value	Name	Description
0x0	NONE	No action
0x1	RESET	The BOD33 generates a reset
0x2	INT	The BOD33 generates an interrupt
0x3	BKUP	The BOD33 puts the device in backup sleep mode if VMON=0. No action if VMON=1.

Bit 2 – HYST: Hysteresis
This bit indicates whether hysteresis is enabled for the BOD33 threshold voltage.

This bit is loaded from NVM User Row at start-up.

This bit is not synchronized.

Value	Description
0	No hysteresis.
1	Hysteresis enabled.

Bit 1 – ENABLE: Enable
This bit is loaded from NVM User Row at start-up.

This bit is not enable-protected.

Value	Description
0	BOD33 is disabled.
1	BOD33 is enabled.

23.8.6 Voltage Regulator System (VREG) Control

Name: VREG
Offset: 0x18 [ID-00001e33]
Reset: 0x00000002
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	VSPER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					VSVSTEP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
								LPEFF
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	STDBYPL0			SEL	ENABLE	
Access		R/W	R/W			R/W	R/W	
Reset		0	1			0	1	

Bits 31:24 – VSPER[7:0]: Voltage Scaling Period
This bitfield sets the period between the voltage steps when the VDDCORE voltage is changing in µs.

If VSPER=0, the period between two voltage steps is 1µs.

Bits 19:16 – VSVSTEP[3:0]: Voltage Scaling Voltage Step
This field sets the voltage step height when the VDDCORE voltage is changing to reach the target VDDCORE voltage.

The voltage step is equal to $2^{VSVSTEP*}$ min_step.

See the Electrical Characteristics chapter for the min_step voltage level.

Bit 8 – LPEFF: Low power Mode Efficiency

Value	Description
0	The voltage regulator in Low power mode has the default efficiency and supports the whole VDD range (1.62V to 3.6V).
1	The voltage regulator in Low power mode has the highest efficiency and supports a limited VDD range (2.5V to 3.6V).

Bit 6 – RUNSTDBY: Run in Standby

Value	Description
0	The voltage regulator is in low power mode in Standby sleep mode.
1	The voltage regulator is in normal mode in Standby sleep mode.

Bit 5 – STDBYPL0: Standby in PL0
This bit selects the performance level (PL) of the main voltage regulator for the Standby sleep mode. This bit is only considered when RUNSTDBY=1.

Value	Description
0	In Standby sleep mode, the voltage regulator remains in the current performance level.
1	In Standby sleep mode, the voltage regulator is used in PL0.

Bit 2 – SEL: Voltage Regulator Selection
This bit is loaded from NVM User Row at start-up.

Value	Description
0	The voltage regulator in active mode is a LDO voltage regulator.
1	The voltage regulator in active mode is a buck converter.

Bit 1 – ENABLE: Enable

Value	Description
0	The voltage regulator is disabled.
1	The voltage regulator is enabled.

23.8.7 Voltage References System (VREF) Control

Name: VREF
Offset: 0x1C [ID-00001e33]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				VREFOE		
Access	R/W	R/W				R/W		
Reset	0	0				0		

Bits 19:16 – SEL[3:0]: Voltage Reference Selection
These bits select the Voltage Reference for the ADC.

Value	Description
0x0	1.024V voltage reference typical value.
0x2	2.048V voltage reference typical value.

Value	Description
0x3	4.096V voltage reference typical value.
Others	Reserved

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows to enable or disable the voltage reference depending on peripheral requests.

Value	Description
0	The voltage reference is always on, if enabled.
1	The voltage reference is enabled when a peripheral is requesting it. The voltage reference is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY: Run In Standby
The bit controls how the voltage reference behaves during standby sleep mode.

Value	Description
0	The voltage reference is halted during standby sleep mode.
1	The voltage reference is not stopped in standby sleep mode. If VREF.ONDEMAND=1, the voltage reference will be running when a peripheral is requesting it. If VREF.ONDEMAND=0, the voltage reference will always be running in standby sleep mode.

Bit 2 – VREFOE: Voltage Reference Output Enable

Value	Description
0	The Voltage Reference output is not available as an ADC input channel.
1	The Voltage Reference output is routed to an ADC input channel.

23.8.8 Battery Backup Power Switch (BBPS) Control

Name: BBPS
Offset: 0x20 [ID-00001e33]
Reset: 0x0000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					PSOKEN	WAKEEN	CONF[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – PSOKEN: Power Supply OK Enable

Value	Description
0	The PSOK pin is not used.
1	The PSOK pin is used to determine the status of the Main Power Supply.

Bit 2 – WAKEEN: Wake Enable

Value	Description
0	The device is not woken up when switched from battery backup power to Main Power.
1	The device is woken up when switched from battery backup power to Main Power.

Bits 1:0 – CONF[1:0]: Battery Backup Power Switch Configuration

Value	Name	Description
0x0	NONE	The backup domain is always supplied by Main Power.
0x1	APWS	The power switch is handled by the Automatic Power Switch.
0x2	FORCED	The backup domain is always supplied by Battery Backup Power.
0x3	BOD33	The power switch is handled by the BOD33.

23.8.9 Backup Output (BKOUT) Control

Name: BKOUT
Offset: 0x24 [ID-00001e33]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
							RTCTGL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
							SET[1:0]	
Access							W	W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							CLR[1:0]	
Access							W	W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
							EN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 25:24 – RTCTGL[1:0]: RTC Toggle Output

Value	Description
0	The output will not toggle on RTC event.
1	The output will toggle on RTC event.

Bits 17:16 – SET[1:0]: Set Output
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will set the corresponding output.

Reading this bit returns '0'.

Bits 9:8 – CLR[1:0]: Clear Output
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will clear the corresponding output.

Reading this bit returns '0'.

Bits 1:0 – EN[1:0]: Enable Output

Value	Description
0	The output is not enabled.
1	The output is enabled and driven by the SUPC.

23.8.10 Backup Input (BKIN) Value

Name: BKIN
Offset: 0x28 [ID-00001e33]
Reset: 0x0000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						BKIN[2:0]		
Access						R	R	R
Reset						0	0	0

Bits 2:0 – BKIN[2:0]: Backup I/O Data Input Value

These bits are cleared when the corresponding backup I/O pin detects a logical low level on the input pin or when the backup I/O is not enabled.

These bits are set when the corresponding backup I/O pin detects a logical high level on the input pin when the backup I/O is enabled.

BKIN[2:0]	PAD	Description
BKIN[0]	PSOK	If BBPS.PSOKEN=1, BKIN[0] will give the input value of the PSOK pin
BKIN[1]	OUT[0]	If BKOUT.EN[0]=1, BKIN[1] will give the input value of the OUT[0] pin
BKIN[2]	OUT[1]	If BKOUT.EN[1]=1, BKIN[2] will give the input value of the OUT[1] pin

24. WDT – Watchdog Timer

24.1 Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

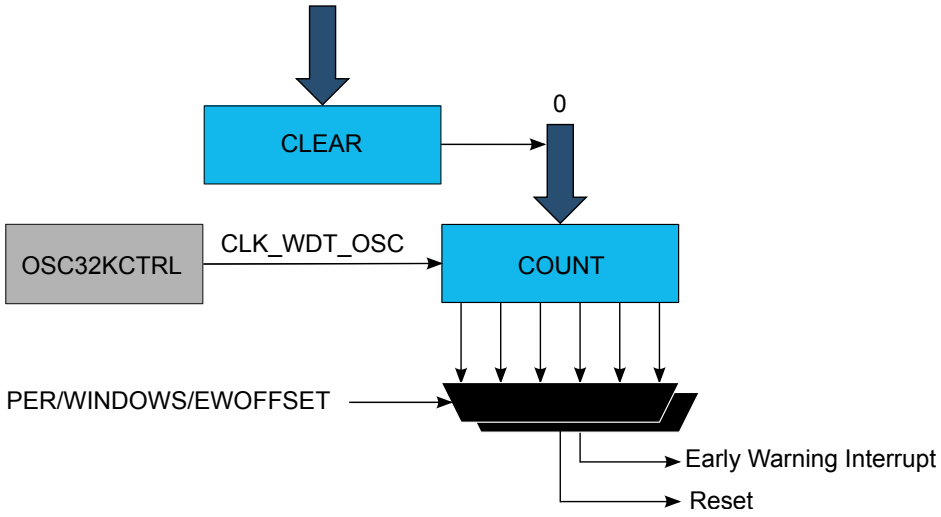
When enabled, the WDT will run in active mode and all sleep modes. It is asynchronous and runs from a CPU-independent clock source. The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail.

24.2 Features

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
  - Normal
  - Window mode
- Selectable time-out periods
  - From 8 cycles to 16,384 cycles in Normal mode
  - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

24.3 Block Diagram

Figure 24-1. WDT Block Diagram



24.4 Signal Description

Not applicable.

24.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

24.5.1 I/O Lines

Not applicable.

24.5.2 Power Management

The WDT can continue to operate in any sleep mode where the selected source clock is running. The WDT interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

24.5.3 Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1KHz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter. This clock must be configured and enabled in the 32KHz Oscillator Controller (OSC32KCTRL) before using the WDT.

CLK_WDT_OSC is normally sourced from the clock of the internal ultra-low-power oscillator, OSCULP32K. Due to the ultra-low-power design, the oscillator is not very accurate, and so the exact time-out period may vary from device to device. This variation must be kept in mind when designing software that uses the WDT to ensure that the time-out periods used are valid for all devices.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

Peripheral Clock Masking

OSC32KCTRL – 32KHz Oscillators Controller

Electrical Characteristics

24.5.4 DMA

Not applicable.

24.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the WDT interrupt(s) requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

Interrupt Line Mapping

24.5.6 Events

Not applicable.

24.5.7 Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

24.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

24.5.9 Analog Connections

Not applicable.

24.6 Functional Description

24.6.1 Principle of Operation

The Watchdog Timer (WDT) is a system for monitoring correct program operation, making it possible to recover from error situations such as runaway code, by issuing a Reset. When enabled, the WDT is a constantly running timer that is configured to a predefined time-out period. Before the end of the time-out period, the WDT should be set back, or else, a system Reset is issued.

The WDT has two modes of operation, Normal mode and Window mode. Both modes offer the option of Early Warning interrupt generation. The description for each of the basic modes is given below. The settings in the Control A register (CTRLA) and the Interrupt Enable register (handled by INTENCLR/INTENSET) determine the mode of operation:

Table 24-1. WDT Operating Modes

CTRLA.ENABLE	CTRLA.WEN	Interrupt Enable	Mode
0	x	x	Stopped
1	0	0	Normal mode
1	0	1	Normal mode with Early Warning interrupt
1	1	0	Window mode
1	1	1	Window mode with Early Warning interrupt

24.6.2 Basic Operation

24.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the WDT is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except the Enable bit (CTRLA.ENABLE)
- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

The WDT can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Enable bit in the Control A register must be set (CTRLA.WEN=1) and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

24.6.2.2 Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE
- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Watchdog Timer Windows Mode Enable bit in the Control A register, CTRLA.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period bits in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

Related Links

NVM User Row Mapping

24.6.2.3 Enabling, Disabling, and Resetting

The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The WDT is disabled by writing a '0' to CTRLA.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control A register (CTRLA.ALWAYSON) is '0'.

24.6.2.4 Normal Mode

In Normal mode operation, the length of a time-out period is configured in CONFIG.PER. The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). Once enabled, the WDT will issue a system reset if a time-out occurs. This can be prevented by clearing the WDT at any time during the time-out period.

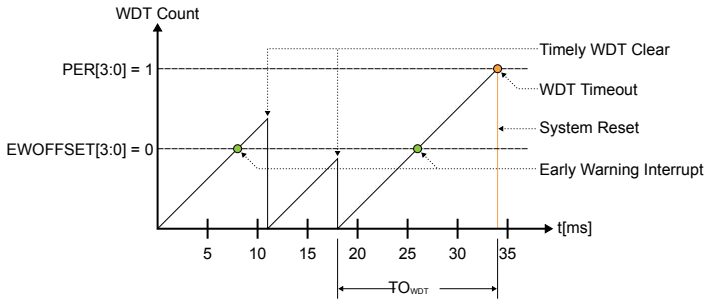
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

There are 12 possible WDT time-out (TO_{WDT}) periods, selectable from 8ms to 16s.

By default, the early warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear register (INTENCLR.EW).

If the Early Warning Interrupt is enabled, an interrupt is generated prior to a WDT time-out condition. In Normal mode, the Early Warning Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET, define the time when the early warning interrupt occurs. The Normal mode operation is illustrated in the figure Normal-Mode Operation.

Figure 24-2. Normal-Mode Operation



24.6.2.5 Window Mode

In Window mode operation, the WDT uses two different time specifications: the WDT can only be cleared by writing 0xA5 to the CLEAR register *after* the closed window time-out period (TO_{WDTW}), during the subsequent Normal time-out period (TO_{WDT}). If the WDT is cleared before the time window opens (before TO_{WDTW} is over), the WDT will issue a system reset.

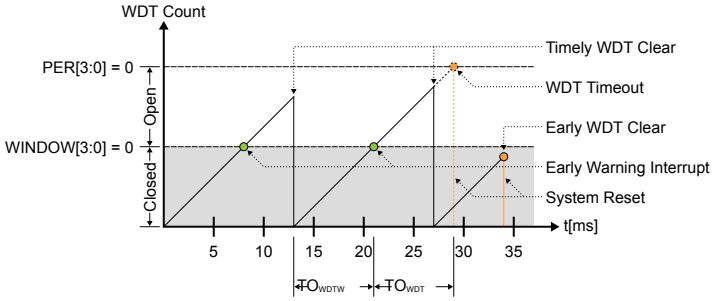
Both parameters TO_{WDTW} and TO_{WDT} are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear (INTENCLR.EW) register.

If the Early Warning interrupt is enabled in Window mode, the interrupt is generated at the start of the open window period, i.e. after TO_{WDTW}. The Window mode operation is illustrated in figure Window-Mode Operation.

Figure 24-3. Window-Mode Operation



24.6.3 DMA Operation

Not applicable.

24.6.4 Interrupts

The WDT has the following interrupt source:

- Early Warning (EW): Indicates that the counter is approaching the time-out condition.
  - This interrupt is an asynchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the WDT is reset. See the INTFLAG register description for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller

Interrupt Line Mapping

PM – Power Manager

Sleep Mode Controller

24.6.5 Events

Not applicable.

24.6.6 Sleep Mode Operation

The WDT will continue to operate in any sleep mode where the source clock is active except backup mode. The WDT interrupts can be used to wake up the device from a sleep mode. An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering an interrupt. In this case, the CPU will continue executing from the instruction following the entry into sleep.

Related Links

CTRLA

24.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following registers are synchronized when written:

- Enable bit in Control A register (CTRLA.ENABLE)
- Window Enable bit in Control A register (CTRLA.WEN)
- Always-On bit in control Control A (CTRLA.ALWAYSON)

The following registers are synchronized when read:

- Watchdog Clear register (CLEAR)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

24.6.8 Additional Features

24.6.8.1 Always-On Mode

The Always-On mode is enabled by setting the Always-On bit in the Control A register (CTRLA.ALWAYSON=1). When the Always-On mode is enabled, the WDT runs continuously, regardless of the state of CTRLA.ENABLE. Once written, the Always-On bit can only be cleared by a power-on reset. The Configuration (CONFIG) and Early Warning Control (EWCTRL) registers are read-only registers while the CTRLA.ALWAYSON bit is set. Thus, the time period configuration bits (CONFIG.PER, CONFIG.WINDOW, EWCTRL.EWOFFSET) of the WDT cannot be changed.

Enabling or disabling Window mode operation by writing the Window Enable bit (CTRLA.WEN) is allowed while in Always-On mode, but note that CONFIG.PER cannot be changed.

The Interrupt Clear and Interrupt Set registers are accessible in the Always-On mode. The Early Warning interrupt can still be enabled or disabled while in the Always-On mode, but note that EWCTRL.EWOFFSET cannot be changed.

Table WDT Operating Modes With Always-On shows the operation of the WDT for CTRLA.ALWAYSON=1.

Table 24-2. WDT Operating Modes With Always-On

WEN	Interrupt Enable	Mode
0	0	Always-on and normal mode
0	1	Always-on and normal mode with Early Warning interrupt
1	0	Always-on and window mode
1	1	Always-on and window mode with Early Warning interrupt

24.6.8.2 Early Warning

The Early Warning interrupt notifies that the WDT is approaching its time-out condition. The Early Warning interrupt behaves differently in Normal mode and in Window mode.

In Normal mode, the Early Warning interrupt generation is defined by the Early Warning Offset in the Early Warning Control register (EWCTRL.EWOFFSET). The Early Warning Offset bits define the number of CLK_WDT_OSC clocks before the interrupt is generated, relative to the start of the watchdog time-out period.

The user must take caution when programming the Early Warning Offset bits. If these bits define an Early Warning interrupt generation time greater than the watchdog time-out period, the watchdog time-out system reset is generated prior to the Early Warning interrupt. Consequently, the Early Warning interrupt will never be generated.

In window mode, the Early Warning interrupt is generated at the start of the open window period. In a typical application where the system is in sleep mode, the Early Warning interrupt can be used to wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

> If the WDT is operating in Normal mode with CONFIG.PER = 0x2 and EWCTRL.EWOFFSET = 0x1, the Early Warning interrupt is generated 16 CLK_WDT_OSC clock cycles after the start of the time-out period. The time-out system reset is generated 32 CLK_WDT_OSC clock cycles after the start of the watchdog time-out period.

24.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ALWAYSON					WEN	ENABLE	
0x01	CONFIG	7:0	WINDOW[3:0]				PER[3:0]			
0x02	EWCTRL	7:0					EWOFFSET[3:0]			
0x03	Reserved									
0x04	INTENCLR	7:0								EW
0x05	INTENSET	7:0								EW
0x06	INTFLAG	7:0								EW
0x07	Reserved									
0x08	SYNCBUSY	7:0				CLEAR	ALWAYSON	WEN	ENABLE	
0x09		15:8								
0x0A		23:16								
0x0B		31:24								
0x0C	CLEAR	7:0	CLEAR[7:0]							

24.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

24.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-0000067a]
Reset: Loaded from NVM User Row at start-up
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ALWAYSON					WEN	ENABLE	
Access	R/W					R/W	R/W	
Reset	0					0	0	

Bit 7 – ALWAYSON: Always-On
This bit allows the WDT to run continuously. After being set, this bit cannot be written to '0', and the WDT will remain enabled until a power-on Reset is received. When this bit is '1', the Control A register

(CTRLA), the Configuration register (CONFIG) and the Early Warning Control register (EWCTRL) will be read-only, and any writes to these registers are not allowed.

Writing a '0' to this bit has no effect.

This bit is not Enable-Protected.

This bit is loaded from NVM User Row at start-up.

Value	Description
0	The WDT is enabled and disabled through the ENABLE bit.
1	The WDT is enabled and can only be disabled by a power-on reset (POR).

Bit 2 – WEN: Watchdog Timer Window Mode Enable
This bit enables Window mode. It can only be written if the peripheral is disabled unless CTRLA.ALWAYSON=1. The initial value of this bit is loaded from Flash Calibration.

This bit is loaded from NVM User Row at startup.

Value	Description
0	Window mode is disabled (normal operation).
1	Window mode is enabled.

Bit 1 – ENABLE: Enable
This bit enables or disables the WDT. It can only be written if CTRLA.ALWAYSON=0.

Due to synchronization, there is delay between writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not Enable-Protected.

This bit is loaded from NVM User Row at startup.

Value	Description
0	The WDT is disabled.
1	The WDT is enabled.

24.8.2 Configuration

Name: CONFIG
Offset: 0x01 [ID-0000067a]
Reset: Loaded from NVM User Row at start-up
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PER[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – WINDOW[3:0]: Window Mode Time-Out Period
In Window mode, these bits determine the watchdog closed window period as a number of cycles of the 1.024kHz CLK_WDT_OSC clock.

These bits are loaded from NVM User Row at start-up.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

Bits 3:0 – PER[3:0]: Time-Out Period
These bits determine the watchdog time-out period as a number of 1.024kHz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period.

These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

24.8.3 Early Warning Control

Name: EWCTRL
Offset: 0x02 [ID-0000067a]
Reset: N/A - Loaded from NVM User Row at startup
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
					EWOFFSET[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – EWOFFSET[3:0]: Early Warning Interrupt Time Offset
These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

24.8.4 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x04 [ID-0000067a]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW: Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning Interrupt Enable bit, which disables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

24.8.5 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x05 [ID-0000067a]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW: Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Early Warning Interrupt Enable bit, which enables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

24.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06 [ID-0000067a]
Reset: 0x00
Property: N/A

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW: Early Warning
This flag is cleared by writing a '1' to it.

This flag is set when an Early Warning interrupt occurs, as defined by the EWOFFSET bit group in EWCTRL.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning interrupt flag.

24.8.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08 [ID-0000067a]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CLEAR	ALWAYSON	WEN	ENABLE	
Access				R	R	R	R	
Reset				0	0	0	0	

Bit 4 – CLEAR: Clear Synchronization Busy

Value	Description
0	Write synchronization of the CLEAR register is complete.
1	Write synchronization of the CLEAR register is ongoing.

Bit 3 – ALWAYSON: Always-On Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ALWAYSON bit is complete.
1	Write synchronization of the CTRLA.ALWAYSON bit is ongoing.

Bit 2 – WEN: Window Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.WEN bit is complete.
1	Write synchronization of the CTRLA.WEN bit is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ENABLE bit is complete.
1	Write synchronization of the CTRLA.ENABLE bit is ongoing.

24.8.8 Clear

Name: CLEAR
Offset: 0x0C [ID-0000067a]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CLEAR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CLEAR[7:0]: Watchdog Clear
In Normal mode, writing 0xA5 to this register during the watchdog time-out period will clear the Watchdog Timer and the watchdog time-out period is restarted.

In Window mode, any writing attempt to this register before the time-out period started (i.e., during TO_{WDTW}) will issue an immediate system Reset. Writing 0xA5 during the time-out period TO_{WDT} will clear the Watchdog Timer and the complete time-out sequence (first TO_{WDTW} then TO_{WDT}) is restarted.

In both modes, writing any other value than 0xA5 will issue an immediate system Reset.

25. RTC – Real-Time Counter

25.1 Overview

The Real-Time Counter (RTC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, or overflow wake up mechanisms, or from the wake inputs.

The RTC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and peripheral event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the minimum counter tick interval is 30.5µs, and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

25.2 Features

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- One 32-bit or two 16-bit compare values
- Clock/Calendar mode
  - Time in seconds, minutes, and hours (12/24)
  - Date in day of month, month, and year
  - Leap year correction
- Digital prescaler correction/tuning for increased accuracy
- Overflow, alarm/compare match and prescaler interrupts and events
  - Optional clear on alarm/compare match
- 8 backup registers with retention capability
- Tamper Detection
  - Timestamp on event or up to 5 inputs with debouncing
  - Active layer protection

25.3 Block Diagram

Figure 25-1. RTC Block Diagram (Mode 0 — 32-Bit Counter)

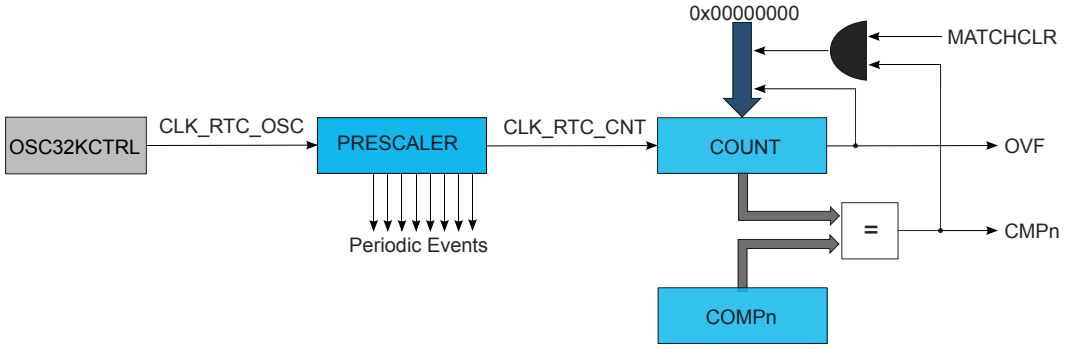


Figure 25-2. RTC Block Diagram (Mode 1 — 16-Bit Counter)

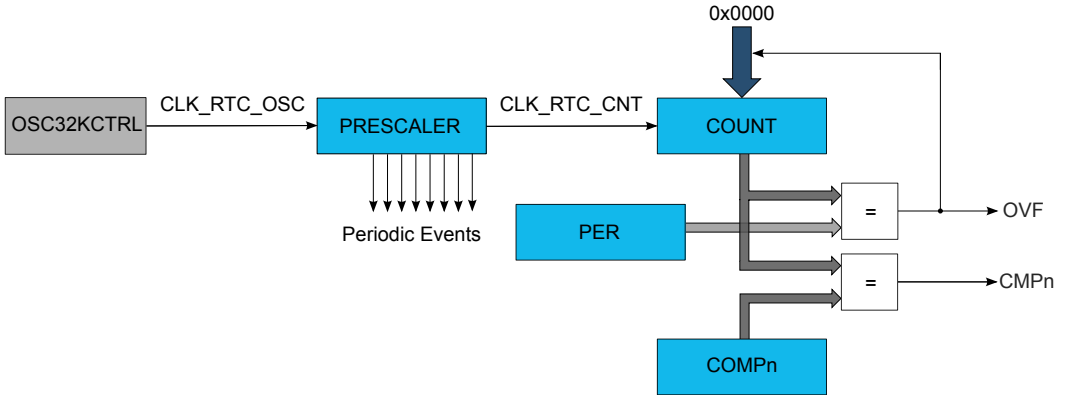


Figure 25-3. RTC Block Diagram (Mode 2 — Clock/Calendar)

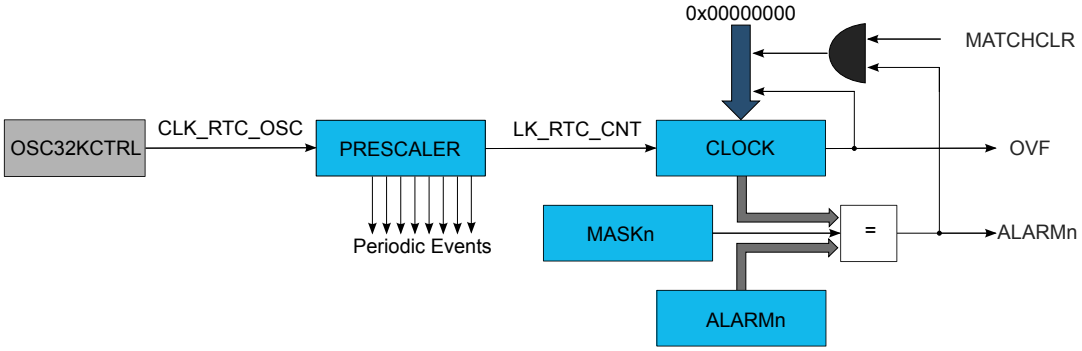
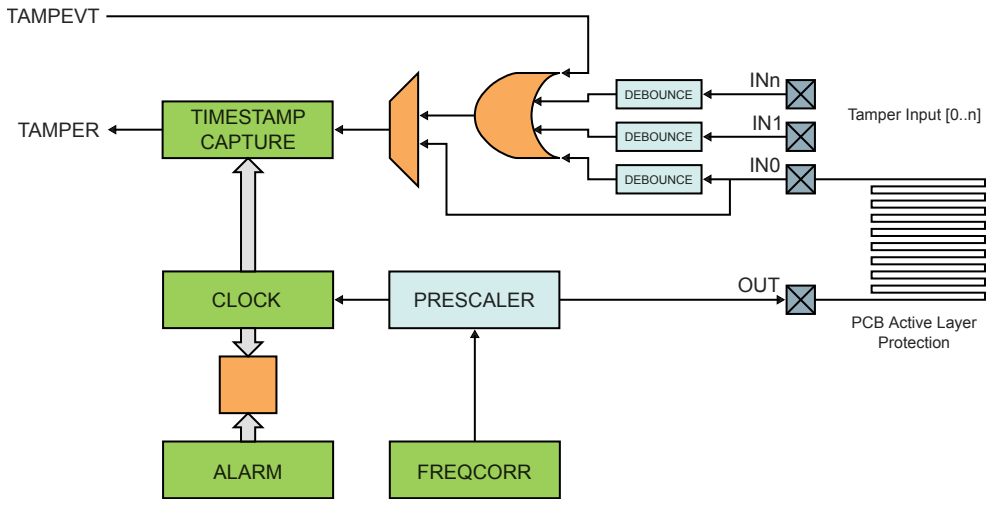


Figure 25-4. RTC Block Diagram (Tamper Detection)



Related Links

32-Bit Counter (Mode 0)

16-Bit Counter (Mode 1)

Clock/Calendar (Mode 2)

Tamper Detection

25.4 Signal Description

Table 25-1. Signal Description

Signal	Description	Type
INn [n=0..4]	Tamper / Wake / Activelayer input	Digital input
OUT	Active layer protection output	Digital output

One signal can be mapped to one of several pins.

Related Links

I/O Multiplexing and Considerations

25.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

25.5.1 I/O Lines

Not applicable.

25.5.2 Power Management

The RTC will continue to operate in any sleep mode where the selected source clock is running. The RTC interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to the *Power Manager* for details on the different sleep modes.

The RTC will be reset only at power-on (POR) or by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1).

Related Links

PM – Power Manager

25.5.3 Clocks

The RTC bus clock (CLK_RTC_APB) can be enabled and disabled in the Main Clock module MCLK, and the default state of CLK_RTC_APB can be found in Peripheral Clock Masking section.

A 32KHz or 1KHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32KHz oscillator controller (OSC32KCTRL) before using the RTC.

This oscillator clock is asynchronous to the bus clock (CLK_RTC_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

OSC32KCTRL – 32KHz Oscillators Controller

Peripheral Clock Masking

25.5.4 DMA

The DMA request lines (or line if only one request) are connected to the DMA Controller (DMAC). Using the RTC DMA requests requires the DMA Controller to be configured first.

Related Links

DMAC – Direct Memory Access Controller

25.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the RTC interrupt requires the Interrupt Controller to be configured first.

Related Links

Nested Vector Interrupt Controller

25.5.6 Events

The events are connected to the *Event System*.

Related Links

EVSYS – Event System

25.5.7 Debug Operation

When the CPU is halted in debug mode the RTC will halt normal operation. The RTC can be forced to continue operation during debugging. Refer to DBGCTRL for details.

25.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Write-protection is denoted by the "PAC Write-Protection" property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the PAC - Peripheral Access Controller for details.

Related Links

PAC - Peripheral Access Controller

25.5.9 Analog Connections

A 32.768kHz crystal can be connected to the XIN32 and XOUT32 pins, along with any required load capacitors. See Electrical Characteristics for details on recommended crystal characteristics and load capacitors.

Related Links

Electrical Characteristics

25.6 Functional Description

25.6.1 Principle of Operation

The RTC keeps track of time in the system and enables periodic events, as well as interrupts and events at a specified time. The RTC consists of a 10-bit prescaler that feeds a 32-bit counter. The actual format of the 32-bit counter depends on the RTC operating mode.

The RTC can function in one of these modes:

- Mode 0 - COUNT32: RTC serves as 32-bit counter
- Mode 1 - COUNT16: RTC serves as 16-bit counter
- Mode 2 - CLOCK: RTC serves as clock/calendar with alarm functionality

25.6.2 Basic Operation

25.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE=0):

- Operating Mode bits in the Control A register (CTRLA.MODE)
- Prescaler bits in the Control A register (CTRLA.PRESCALER)
- Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- Clock Representation bit in the Control A register (CTRLA.CLKREP)

The following registers are enable-protected:

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- Tamper Control register (TAMPCTRL)

Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE=0). If the RTC is enabled (CTRLA.ENABLE=1), these operations are necessary: first write CTRLA.ENABLE=0 and check whether the write synchronization has finished, then change the desired bit field value. Enable-protected bits in can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

Note: In Clock/Calendar mode, the prescaler must be configured to provide a 1Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$$f_{CLK_RTC_CNT} = \frac{f_{CLK_RTC_OSC}}{2^{PRESCALER}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{CLK_RTC_OSC}$, and $f_{CLK_RTC_CNT}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

25.6.2.2 Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE=1). The RTC is disabled by writing CTRLA.ENABLE=0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

25.6.2.3 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x0, the counter operates in 32-bit Counter mode. The block diagram of this mode is shown in Figure 25-1. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare register (COMP0). When a compare match occurs, the Compare 0 Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next 0-to-1 transition of CLK_RTC_CNT.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMP0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMP0 and INTFLAG.OVF will both be set simultaneously on a compare match with COMP0.

25.6.2.4 16-Bit Counter (Mode 1)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x1, the counter operates in 16-bit Counter mode as shown in Figure 25-2. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. In 16-bit Counter mode, the 16-bit Period register (PER) holds the maximum value of the counter. The counter will increment until it reaches the PER value, and then wrap to 0x0000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 16-bit format.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT.

25.6.2.5 Clock/Calendar (Mode 2)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x2, the counter operates in Clock/Calendar mode, as shown in Figure 25-3. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm register (ALARM0). When an alarm match occurs, the Alarm 0 Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARM0) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm 0 Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm 0 Mask register (MASK0.SEL). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARM0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see Periodic Intervals).
Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARM0 and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARM0.

25.6.3 DMA Operation

The RTC generates the following DMA request:

- Tamper (TAMPER): The request is set on capture of the timestamp. The request is cleared when the Timestamp register is read.

If the CPU accesses the registers which are source for DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

25.6.4 Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Indicates detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to Periodic Intervals for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is raised and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the RTC is reset. See the description of the INTFLAG registers for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the Nested Vector Interrupt Controller for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to the Nested Vector Interrupt Controller for details.

Related Links

Nested Vector Interrupt Controller

25.6.5 Events

The RTC can generate the following output events:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Generated on detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to Periodic Intervals for details.

Setting the Event Output bit in the Event Control Register (EVCTRL.xxxEO=1) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the EVSYS - Event System for details on configuring the event system.

The RTC can take the following actions on an input event:

- Tamper (TAMPEVT): Capture the RTC counter to the timestamp register. See *Tamper Detection*.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.xxxEI) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event.

Related Links

EVSYS – Event System

25.6.6 Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU will continue executing right from the first instruction that followed the entry into sleep.

The periodic events can also wake up the CPU through the interrupt function of the Event System. In this case, the event must be enabled and connected to an event channel with its interrupt enabled. See *Event System* for more information.

25.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE
- Count Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)
- Clock Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn
- The General Purpose n registers (GPn)

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'
- The Timestamp Value register (TIMESTAMP)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

Register Synchronization

25.6.8 Additional Features

25.6.8.1 Periodic Intervals

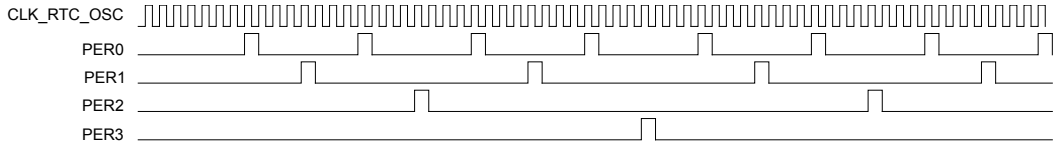
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

$$f_{PERIODIC(n)} = \frac{f_{CLK_RTC_OSC}}{2^{n+3}}$$

$f_{CLK_RTC_OSC}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREOn bit. For example, PER0 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

Figure 25-5. Example Periodic Events



25.6.8.2 Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 8192 CLK_RTC_OSC cycles. The Value bit group in the Frequency Correction register (FREQCORR.VALUE) determines the number of times the adjustment is applied over 128 of these periods. The resulting correction is as follows:

$$\text{Correction in ppm} = \frac{\text{FREQCORR.VALUE}}{8192 \cdot 128} \cdot 10^6 \text{ppm}$$

This results in a resolution of 0.95367ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

25.6.8.3 Backup Registers

The RTC includes eight Backup registers (BKUPn). These registers maintain their content in Backup sleep mode. They can be used to store user-defined values.

If more user-defined data must be stored than the eight Backup registers can hold, the General Purpose registers (GPn) can be used.

Related Links

PM – Power Manager

25.6.8.4 General Purpose Registers

The RTC includes two General Purpose registers (GPn). These registers are reset only when the RTC is reset or when tamper detection occurs while CTRLA.GPTRST=1, and remain powered while the RTC is powered. They can be used to store user-defined values while other parts of the system are powered off.

It is recommended to use the eight Backup registers (BKUPn) first to store user-defined values, and use the GPn only when the user-defined values exceed the capacity of the provided BKUPn.

An example procedure to write the general purpose registers GP0 and GP1 is:

1. Wait for any ongoing write to COMP0 to complete (SYNCBUSY.COMP0 = 0). If the RTC is operating in Mode 1, wait for any ongoing write to COMP1 to complete as well (SYNCBUSY.COMP1 = 0).
2. Write CTRLB.GP0EN = 1 if GP0 is needed.
3. Write GP0 if needed.
4. Wait for any ongoing write to GP0 to complete (SYNCBUSY.GP0 = 0). Note that GP1 will also show as busy when GP0 is busy.
5. Write GP1 if needed.

Table 25-2. General Purpose Registers vs Compare/Alarm Registers

Register	Mode 0	Mode 1	Mode 2	Write Before
GP0	COMP0	COMP0 / COMP1	ALARM0	GP1
GP1	COMP0	COMP0 / COMP1	ALARM0	-

The GPn registers share internal resources with the compare/alarm features. Each pair of 32-bit GPn are associated with one 32-bit compare/alarm or a pair of 16-bit compare registers as shown in the table above. Before using an even GPn, the associated compare/alarm feature must be disabled by writing a '1' to the General Purpose Enable bit in the Control B register (CTRLB.GPnEN). To re-enable the compare/alarm, CTRLB.GPnEN must be written to zero and the associated COMPn/ALARMn must be written with the correct value. Each even GPn must also be written prior to writing the odd GPn if both will be used. Odd GPn can be used without affecting the compare/alarm functions; however, any writes to the associated COMPn/ALARMn register must be completed before writing the odd GPn.

25.6.8.5 Tamper Detection

The RTC provides up to five selectable polarity external inputs (INn) that can be used for tamper detection. The RTC also supports an input event (TAMPEVT) for generating a tamper condition from within the Event System.

A single interrupt request (TAMPER) is available for all tamper sources. The polarity for each input is selected with the Tamper Level bits in the Tamper Control register (TAMPCTRL.TAMPLVLn). The tamper input event is enabled by the Tamper Input Event Enable in the Event Control register (EVCTRL.TAMPEVIE). The action of each input pin is configured using the Input n Action bits in the Tamper Control register (TAMPCTRL.INnACT). Tamper inputs support the following actions:

- Off: Detection for INn is disabled.
- Wake: A transition on INn matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will not be captured in the TIMESTAMP register
- Capture: A transition on INn matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.
- Active Layer Protection: A mismatch between INn and OUT will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.

In order to determine which tamper source caused a tamper event, the Tamper ID register (TAMPID) provides the detection status of each input pin and the input event. These bits remain active until cleared by software. Separate debouncers are embedded for each external input. The debouncer for each input is enabled/disabled with the Debounce Enable bits in the Tamper Control register (TAMPCTRL.DEBNCn). The debouncer configuration is fixed for all inputs as set by the Control B register (CTRLB). The debouncing period duration is configurable using the Debounce Frequency field in the Control B register

(CTRLB.DEBF). The period is set for all debouncers (i.e., the duration cannot be adjusted separately for each debouncer).

When TAMPCTRL.DEBNCn = 0, INn is detected asynchronously. See Figure 25-6 for an example.

When TAMPCTRL.DEBNCn = 1, the detection time depends on whether the debouncer operates synchronously or asynchronously, and whether majority detection is enabled or not. Refer to the table below for more details. Synchronous versus asynchronous stability debouncing is configured by the Debounce Asynchronous Enable bit in the Control B register (CTRLB.DEBASYNC):

- Synchronous (CTRLB.DEBASYNC = 0): INn is synchronized in two CLK_RTC periods and then must remain stable for four CLK_RTC_DEB periods before a valid detection occurs. See Figure 25-7 for an example.
- Asynchronous (CTRLB.DEBASYNC = 1): The first edge on INn is detected. Further detection is blanked until INn remains stable for four CLK_RTC_DEB periods. See Figure 25-8 for an example.

Majority debouncing is configured by the Debounce Majority Enable bit in the Control B register (CTRLB.DEBMAJ). INn must be valid for two out of three CLK_RTC_DEB periods. See Figure 25-9 for an example.

Table 25-3. Debouncer Configuration

TAMPCTRL.DEBNCn	CTRLB.DEBMAJ	CTRLB.DEBASYNC	Description
0	X	X	Detect edge on INn with no debouncing. Every edge detected is immediately triggered.
1	0	0	Detect edge on INn with synchronous stability debouncing. Edge detected is only triggered when INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	0	1	Detect edge on INn with asynchronous stability debouncing. First detected edge is triggered immediately. All subsequent detected edges are ignored until INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	1	X	Detect edge on INn with majority debouncing. Pin INn is sampled for 3 consecutive CLK_RTC_DEB periods. Signal level is determined by majority-rule (LLL, LLH, LHL, HLL = '0' and LHH, HLH, HHL, HHH = '1').

Figure 25-6. Edge Detection with Debouncer Disabled

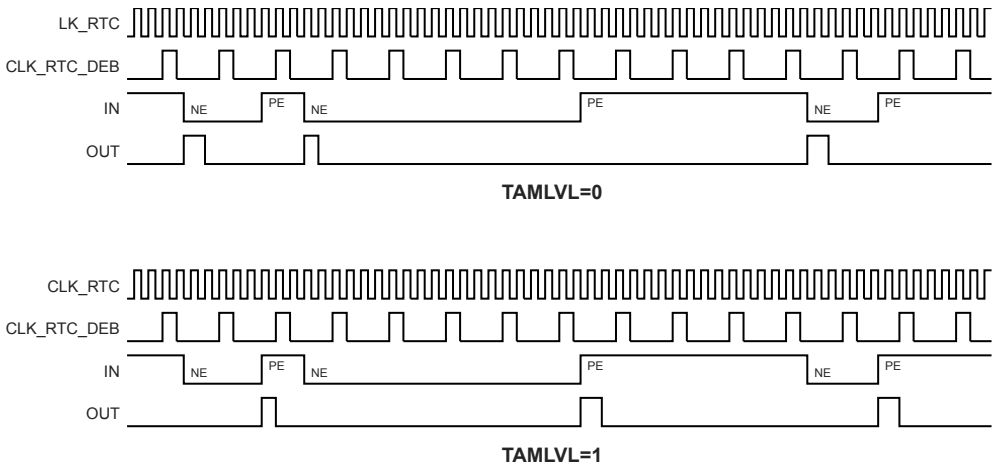


Figure 25-7. Edge Detection with Synchronous Stability Debouncing

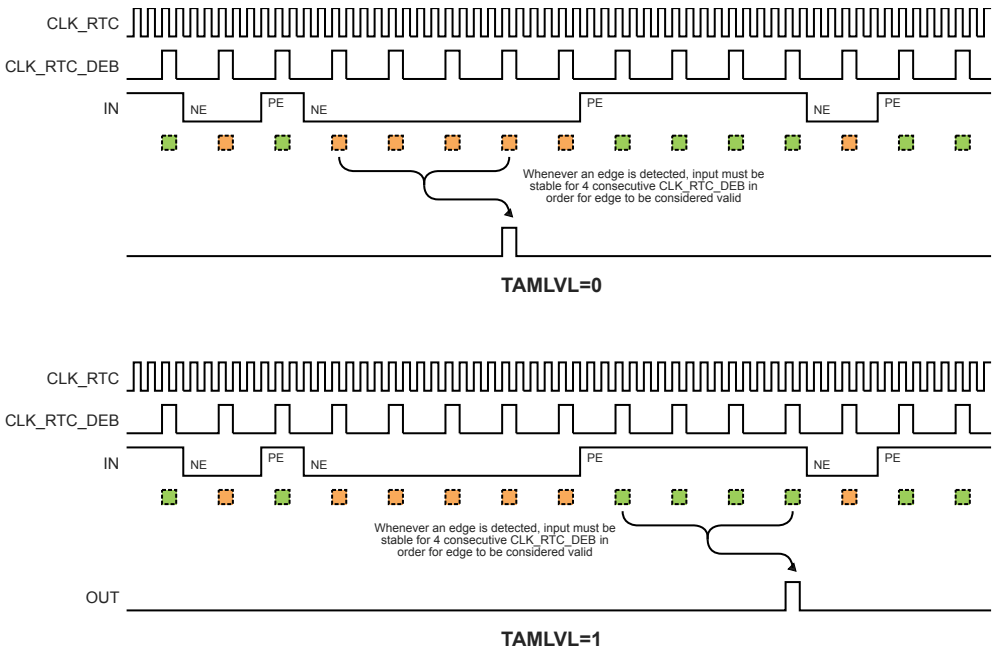


Figure 25-8. Edge Detection with Asynchronous Stability Debouncing

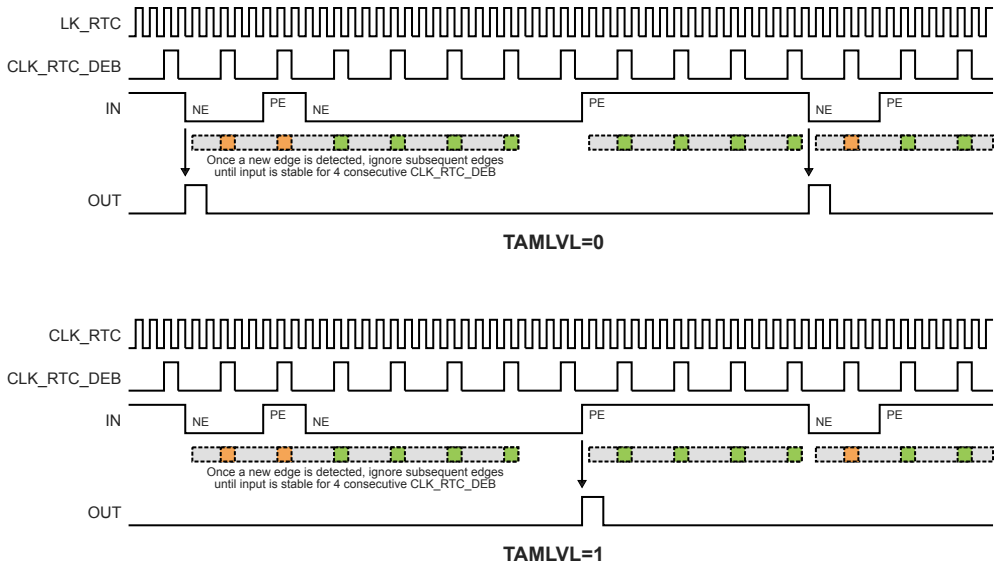
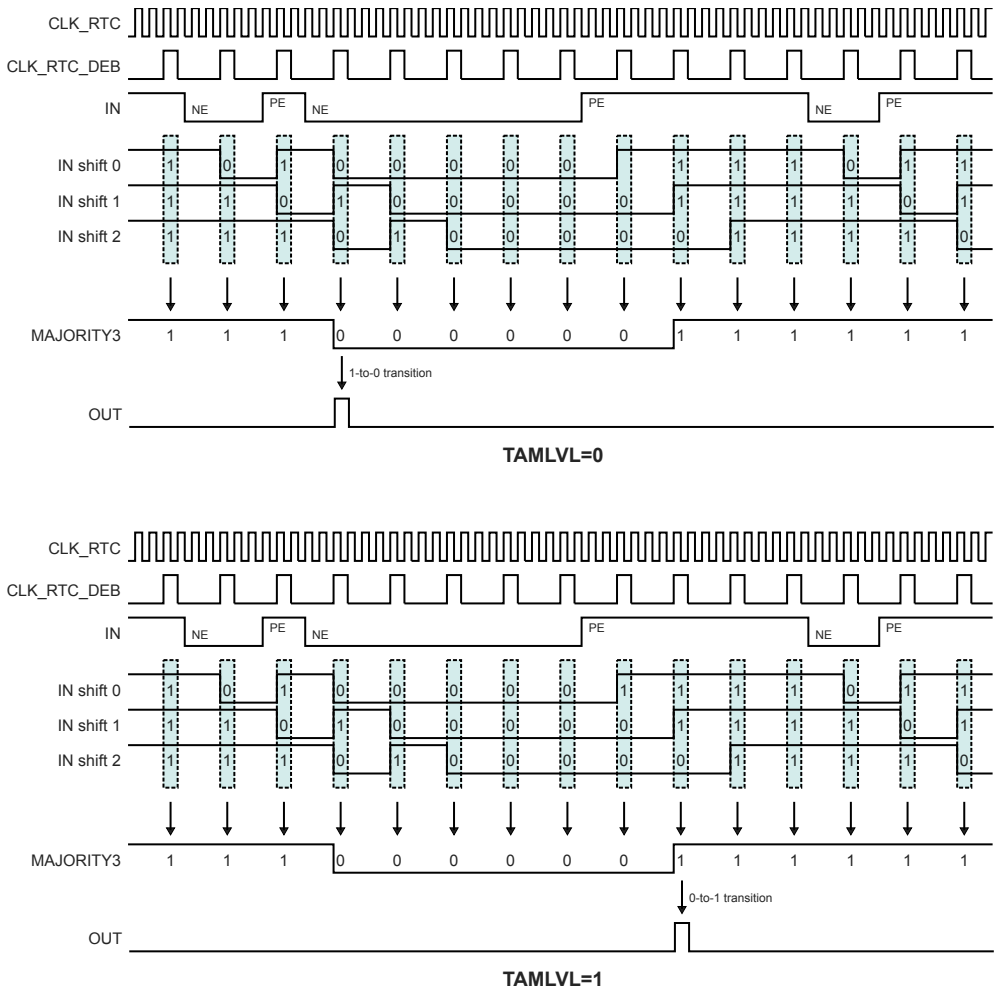


Figure 25-9. Edge Detection with Majority Debouncing



Related Links

Block Diagram
Active Layer Protection

25.6.8.6 Active Layer Protection

The RTC provides a means of detecting broken traces on the PCB, also known as Active Layer Protection. In this mode an RTC output signal is routed over critical components on the board and fed back to one of the RTC inputs. The input and output signals are compared and a tamper condition is detected when they do not match.

Enabling active layer protection requires the following steps:

- Enable the RTC prescaler output by writing a one to the RTC Out bit in the Control B register (CTRLB.RTCOUT). The I/O pins must also be configured to correctly route the signal to the external pins.
- Select the frequency of the output signal by configuring the RTC Active Layer Frequency field in the Control B register (CTRLB.ACTF).

  $$\mathrm{GCLK_RTC_OUT} = \frac{\mathrm{CLK_RTC}}{2^{\mathrm{CTRLB.ACTF}+1}}$$

- Enable one of the tamper inputs (INn) in active layer mode by writing 3 to the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT). When active layer protection is enabled, the value of INn is sampled on the falling edge of CLK_RTC and compared to the expected value of OUT. Therefore up to one half of a CLK_RTC period is available for propagation delay through the trace.

Related Links

Tamper Detection

25.7 Register Summary - COUNT32

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
0x01		15:8	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
0x03		15:8		ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
0x05		15:8	OVFEO	TAMPEREO						CMPEO0
0x06		23:16								TAMPEVEI
0x07		31:24								
0x08	INTENCLR	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x09		15:8	OVF	TAMPER						CMP0
0x0A	INTENSET	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0B		15:8	OVF	TAMPER						CMP0
0x0C	INTFLAG	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0D		15:8	OVF	TAMPER						CMP0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			COMP0		COUNT	FREQCORR	ENABLE	SWRST
0x11		15:8	COUNTSYNC							
0x12		23:16							GPn	GPn
0x13		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
0x19		15:8	COUNT[15:8]							
0x1A		23:16	COUNT[23:16]							
0x1B		31:24	COUNT[31:24]							
0x1C ... 0x1F	Reserved									
0x20	COMP0	7:0	COMP[7:0]							
0x21		15:8	COMP[15:8]							
0x22		23:16	COMP[23:16]							
0x23		31:24	COMP[31:24]							
0x24 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
0x41		15:8	GP[15:8]							
0x42		23:16	GP[23:16]							
0x43		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
0x45		15:8	GP[15:8]							

Offset	Name	Bit Pos.								
0x46		23:16	GP[23:16]							
0x47		31:24	GP[31:24]							
0x48 ... 0x5F	Reserved									
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
0x61		15:8							IN4ACT[1:0]	
0x62		23:16				TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn
0x63		31:24				DEBNCn	DEBNCn	DEBNCn	DEBNCn	DEBNCn
0x64	TIMESTAMP	7:0	COUNT[7:0]							
0x65		15:8	COUNT[15:8]							
0x66		23:16	COUNT[23:16]							
0x67		31:24	COUNT[31:24]							
0x68	TAMPID	7:0				TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn
0x69		15:8								
0x6A		23:16								
0x6B		31:24	TAMPEVT							
0x6C ... 0x7F	Reserved									
0x80	BKUP0	7:0	BKUP[7:0]							
0x81		15:8	BKUP[15:8]							
0x82		23:16	BKUP[23:16]							
0x83		31:24	BKUP[31:24]							
0x84	BKUP1	7:0	BKUP[7:0]							
0x85		15:8	BKUP[15:8]							
0x86		23:16	BKUP[23:16]							
0x87		31:24	BKUP[31:24]							
0x88	BKUP2	7:0	BKUP[7:0]							
0x89		15:8	BKUP[15:8]							
0x8A		23:16	BKUP[23:16]							
0x8B		31:24	BKUP[31:24]							
0x8C	BKUP3	7:0	BKUP[7:0]							
0x8D		15:8	BKUP[15:8]							
0x8E		23:16	BKUP[23:16]							
0x8F		31:24	BKUP[31:24]							
0x90	BKUP4	7:0	BKUP[7:0]							
0x91		15:8	BKUP[15:8]							
0x92		23:16	BKUP[23:16]							
0x93		31:24	BKUP[31:24]							
0x94	BKUP5	7:0	BKUP[7:0]							
0x95		15:8	BKUP[15:8]							
0x96		23:16	BKUP[23:16]							
0x97		31:24	BKUP[31:24]							
0x98	BKUP6	7:0	BKUP[7:0]							
0x99		15:8	BKUP[15:8]							
0x9A		23:16	BKUP[23:16]							

Offset	Name	Bit Pos.								
0x9B		31:24	BKUP[31:24]							
0x9C	BKUP7	7:0	BKUP[7:0]							
0x9D		15:8	BKUP[15:8]							
0x9E		23:16	BKUP[23:16]							
0x9F		31:24	BKUP[31:24]							

25.8 Register Description - COUNT32

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

25.8.1 Control A in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR				MODE[1:0]		ENABLE	SWRST
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 15 – COUNTSYNC: COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST: GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Bit 13 – BKTRST: GP Registers Reset On Tamper Enable
All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	BKUPn registers will not reset when a tamper condition occurs.
1	BKUPn registers will reset when a tamper condition occurs.

Bits 11:8 – PRESCALER[3:0]: Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR: Clear on Match
This bit defines if the counter is cleared or not on a match.

This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bits 3:2 – MODE[1:0]: Operating Mode
This bit group defines the operating mode of the RTC.

This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.8.2 Control B in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
		ACTF[2:0]				DEBF[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 14:12 – ACTF[2:0]: Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16

Value	Name	Description
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0]: Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN: DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT: RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC: Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ: Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – GP0EN: General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

25.8.3 Event Control in COUNT32 mode (CTRLA.MODE=0)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO						CMPEO0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – TAMPEVEI: Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled and incoming events will be ignored.
1	Tamper event input is enabled and incoming events will capture the COUNT value.

Bit 15 – OVFEO: Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO: Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled and will not be generated.
1	Tamper event output is enabled and will be generated for every tamper input.

Bit 8 – CMPEO0: Compare 0 Event Output Enable

Value	Description
0	Compare 0 event is disabled and will not be generated.
1	Compare 0 event is enabled and will be generated for every compare match.

Bits 7:0 – PEREOn: Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

25.8.4 Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER: Tamper Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this but will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bit 8 – CMP0: Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Compare 0 Interrupt Enable bit, which disables the Compare interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.8.5 Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER: Tamper Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bit 8 – CMP0: Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Compare 0 Interrupt Enable bit, which enables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.8.6 Interrupt Flag Status and Clear in COUNT32 mode (CTRLA.MODE=0)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER: Tamper event
This flag is set after a damper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bit 8 – CMP0: Compare 0
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMP0 is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Compare 0 interrupt flag.

Bits 7:0 – PERn: Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.8.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

25.8.8 Synchronization Busy in COUNT32 mode (CTRLA.MODE=0)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPn	GPn
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
			COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 17:16 – GPn: General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

Bit 15 – COUNTSYNC: Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bit 5 – COMP0: Compare 0 Synchronization Busy Status

Value	Description
0	Write synchronization for COMP0 register is complete.
1	Write synchronization for COMP0 register is ongoing.

Bit 3 – COUNT: Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR: Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for FREQCORR register is complete.
1	Read/write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.ENABLE bit is complete.
1	Read/write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.SWRST bit is complete.
1	Read/write synchronization for CTRLA.SWRST bit is ongoing.

25.8.9 Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN: Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0]: Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.8.10 Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0]: Counter Value
These bits define the value of the 32-bit RTC counter in mode 0.

25.8.11 Compare 0 Value in COUNT32 mode (CTRLA.MODE=0)

Name: COMP0
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	COMP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COMP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP[31:0]: Compare Value
The 32-bit value of COMP0 is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare 0 interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is '1'.

25.8.12 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0]: General Purpose
These bits are for user-defined general purpose use, see General Purpose Registers.

25.8.13 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				DEBNCn	DEBNCn	DEBNCn	DEBNCn	DEBNCn
Access								
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn
Access								
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							IN4ACT[1:0]	
Access								
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 28:24 – DEBNCn: Debounce Enable n

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 20:16 – TAMLVLn: Tamper Level Select n

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 8:9 – IN0ACT, IN1ACT, IN2ACT, IN3ACT, IN4ACT: Tamper Input n Action
These bits determine the action taken by Tamper Input INn.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare INn to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

25.8.14 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0]: Count Timestamp Value
The 32-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs

25.8.15 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 31 – TAMPEVT: Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 4:0 – TAMPIDn: Tamper Input n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input condition has not been detected on INn pin
1	A tamper input condition has been detected in INn pin

25.8.16 Backup n

Name: BKUP
Offset: 0x80 + n*0x04 [n=0..7]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	BKUP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BKUP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BKUP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BKUP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BKUP[31:0]: Backup
These bits are user-defined for general purpose use in the Backup domain.

25.9 Register Summary - COUNT16

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0					MODE[1:0]		ENABLE	SWRST
0x01		15:8	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
0x02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
0x03		15:8		ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
0x05		15:8	OVFEO	TAMPEREO				ALARMEOn	ALARMEOn	CMPEOn
0x06		23:16								TAMPEVEI
0x07		31:24								
0x08	INTENCLR	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x09		15:8	OVF	TAMPER					CMPn	CMPn
0x0A	INTENSET	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0B		15:8	OVF	TAMPER					CMPn	CMPn
0x0C	INTFLAG	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0D		15:8	OVF	TAMPER					CMPn	CMPn
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0		COMPn	COMPn	PER	COUNT	FREQCORR	ENABLE	SWRST
0x11		15:8	COUNTSYNC							
0x12		23:16								
0x13		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
0x19		15:8	COUNT[15:8]							
0x1A ... 0x1B	Reserved									
0x1C	PER	7:0	PER[7:0]							
0x1D		15:8	PER[15:8]							
0x1E ... 0x1F	Reserved									
0x20	COMP0	7:0	COMP[7:0]							
0x21		15:8	COMP[15:8]							
0x22	COMP1	7:0	COMP[7:0]							
0x23		15:8	COMP[15:8]							
0x24 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
0x41		15:8	GP[15:8]							
0x42		23:16	GP[23:16]							

Offset	Name	Bit Pos.									
0x43		31:24	GP[31:24]								
0x44	GP1	7:0	GP[7:0]								
0x45		15:8	GP[15:8]								
0x46		23:16	GP[23:16]								
0x47		31:24	GP[31:24]								
0x48 ... 0x5F	Reserved										
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]		
0x61		15:8							IN4ACT[1:0]		
0x62		23:16				TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	
0x63		31:24				DEBNCn	DEBNCn	DEBNCn	DEBNCn	DEBNCn	
0x64	TIMESTAMP	7:0	COUNT[7:0]								
0x65		15:8	COUNT[15:8]								
0x66		23:16									
0x67		31:24									
0x68	TAMPID	7:0				TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	
0x69		15:8									
0x6A		23:16									
0x6B		31:24	TAMPEVT								
0x6C ... 0x7F	Reserved										
0x80	BKUP0	7:0	BKUP[7:0]								
0x81		15:8	BKUP[15:8]								
0x82		23:16	BKUP[23:16]								
0x83		31:24	BKUP[31:24]								
0x84	BKUP1	7:0	BKUP[7:0]								
0x85		15:8	BKUP[15:8]								
0x86		23:16	BKUP[23:16]								
0x87		31:24	BKUP[31:24]								
0x88	BKUP2	7:0	BKUP[7:0]								
0x89		15:8	BKUP[15:8]								
0x8A		23:16	BKUP[23:16]								
0x8B		31:24	BKUP[31:24]								
0x8C	BKUP3	7:0	BKUP[7:0]								
0x8D		15:8	BKUP[15:8]								
0x8E		23:16	BKUP[23:16]								
0x8F		31:24	BKUP[31:24]								
0x90	BKUP4	7:0	BKUP[7:0]								
0x91		15:8	BKUP[15:8]								
0x92		23:16	BKUP[23:16]								
0x93		31:24	BKUP[31:24]								
0x94	BKUP5	7:0	BKUP[7:0]								
0x95		15:8	BKUP[15:8]								
0x96		23:16	BKUP[23:16]								
0x97		31:24	BKUP[31:24]								

Offset	Name	Bit Pos.								
0x98	BKUP6	7:0	BKUP[7:0]							
0x99		15:8	BKUP[15:8]							
0x9A		23:16	BKUP[23:16]							
0x9B		31:24	BKUP[31:24]							
0x9C	BKUP7	7:0	BKUP[7:0]							
0x9D		15:8	BKUP[15:8]							
0x9E		23:16	BKUP[23:16]							
0x9F		31:24	BKUP[31:24]							

25.10 Register Description - COUNT16

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

25.10.1 Control A in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MODE[1:0]		ENABLE	SWRST
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – COUNTSYNC: COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST: GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	GPn registers will not reset when a tamper condition occurs.
1	GPn registers will reset when a tamper condition occurs.

Bit 13 – BKTRST: GP Registers Reset On Tamper Enable
All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	BKUPn registers will not reset when a tamper condition occurs.
1	BKUPn registers will reset when a tamper condition occurs.

Bits 11:8 – PRESCALER[3:0]: Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bits 3:2 – MODE[1:0]: Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.10.2 Control B in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
		ACTF[2:0]				DEBF[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 14:12 – ACTF[2:0]: Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16

Value	Name	Description
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0]: Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN: DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT: RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC: Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ: Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 0 – GP0EN: General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0/GP1 disabled.
1	COMP0 compare function disabled. GP0/GP1 enabled.

25.10.3 Event Control in COUNT16 mode (CTRLA.MODE=1)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO				ALARMEOn	ALARMEOn	CMPEOn
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – TAMPEVEI: Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored
1	Tamper event input is enabled, and incoming events will capture the CLOCK value

Bit 15 – OVFEO: Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO: Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated.
1	Tamper event output is enabled, and will be generated for every tamper input.

Bits 10:9 – ALARMEOn: Alarm n Event Output Enable

Value	Description
0	Alarm n event is disabled, and will not be generated
	Alarm n event is enabled, and will be generated for every compare batch

Bits 9:8 – CMPEOn: Compare n Event Output Enable [n = 1..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 7:0 – PEREOn: Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated. [n = 7..0]
1	Periodic Interval n event is enabled and will be generated. [n = 7..0]

25.10.4 Interrupt Enable Clear in COUNT16 mode (CTRLA.MODE=1)

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER					CMPn	CMPn
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER: Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 9:8 – CMPn: Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.10.5 Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER					CMPn	CMPn
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER: Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 9:8 – CMPn: Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which and enables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.10.6 Interrupt Flag Status and Clear in COUNT16 mode (CTRLA.MODE=1)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER					CMPn	CMPn
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER: Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/ INTENSET.TAMPER is one.

Writing a '0' to this bit has no effect.

Writing a one to this bit clears the Tamper interrupt flag.

Bits 9:8 – CMPn: Compare n [n = 1..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMPx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 7:0 – PERn: Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.10.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

25.10.8 Synchronization Busy in COUNT16 mode (CTRLA.MODE=1)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
		COMPn	COMPn	PER	COUNT	FREQCORR	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 15 – COUNTSYNC: Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bits 6:5 – COMPn: Compare n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for COMPn register is complete.
1	Write synchronization for COMPn register is ongoing.

Bit 4 – PER: Period Synchronization Busy Status

Value	Description
0	Write synchronization for PER register is complete.
1	Write synchronization for PER register is ongoing.

Bit 3 – COUNT: Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR: Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for FREQCORR register is complete.
1	Read/write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.ENABLE bit is complete.
1	Read/write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.SWRST bit is complete.
1	Read/write synchronization for CTRLA.SWRST bit is ongoing.

25.10.9 Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN: Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0]: Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.10.10 Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0]: Counter Value
These bits define the value of the 16-bit RTC counter in COUNT16 mode (CTRLA.MODE=1).

25.10.11 Counter Period in COUNT16 mode (CTRLA.MODE=1)

Name: PER
Offset: 0x1C
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PER[15:0]: Counter Period
These bits define the value of the 16-bit RTC period in COUNT16 mode (CTRLA.MODE=1).

25.10.12 Compare n Value in COUNT16 mode (CTRLA.MODE=1)

Name: COMP
Offset: 0x20 + n*0x02 [n=0..1]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COMP[15:0]: Compare Value
The 16-bit value of COMPn is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle.

25.10.13 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0]: General Purpose
These bits are for user-defined general purpose use, see General Purpose Registers.

25.10.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				DEBNCn	DEBNCn	DEBNCn	DEBNCn	DEBNCn
Access								
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn
Access								
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							IN4ACT[1:0]	
Access								
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 28:24 – DEBNCn: Debounce Enable n

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 20:16 – TAMLVLn: Tamper Level Select n

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 8:9 – IN0ACT, IN1ACT, IN2ACT, IN3ACT, IN4ACT: Tamper Input n Action
These bits determine the action taken by Tamper Input INn.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare INn to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

25.10.15 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0]: Count Timestamp Value
The 16-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs.

25.10.16 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 31 – TAMPEVT: Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 4:0 – TAMPIDn: Tamper Input n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input condition has not been detected on INn pin
1	A tamper input condition has been detected in INn pin

25.10.17 Backup n

Name: BKUP
Offset: 0x80 + n*0x04 [n=0..7]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	BKUP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BKUP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BKUP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BKUP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BKUP[31:0]: Backup

These bits are user-defined for general purpose use in the Backup domain.

25.11 Register Summary - CLOCK

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
0x01		15:8	CLOCKSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
02	CTRLB	7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ			GP2EN	GP0EN
03		15:8		ACTF[2:0]				DEBF[2:0]		
0x04	EVCTRL	7:0	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
0x05		15:8	OVFEO	TAMPEREO						ALARMO0
0x06		23:16								TAMPEVEI
0x07		31:24								
0x08	INTENCLR	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x09		15:8	OVF	TAMPER						ALARM0
0x0A	INTENSET	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0B		15:8	OVF	TAMPER						ALARM0
0x0C	INTFLAG	7:0	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
0x0D		15:8	OVF	TAMPER						ALARM0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
0x11		15:8	CLOCKSYNC				MASK0			
0x12		23:16							GPn	GPn
0x13		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	CLOCK	7:0	MINUTE[1:0]		SECOND[5:0]					
0x19		15:8	HOUR[3:0]				MINUTE[5:2]			
0x1A		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x1B		31:24	YEAR[5:0]						MONTH[3:2]	
0x1C ... 0x1F	Reserved									
0x20	ALARM	7:0	MINUTE[1:0]		SECOND[5:0]					
0x21		15:8	HOUR[3:0]				MINUTE[5:2]			
0x22		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x23		31:24	YEAR[5:0]						MONTH[3:2]	
0x24	MASK	7:0						SEL[2:0]		
0x25 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
0x41		15:8	GP[15:8]							
0x42		23:16	GP[23:16]							
0x43		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							

Offset	Name	Bit Pos.									
0x45		15:8	GP[15:8]								
0x46		23:16	GP[23:16]								
0x47		31:24	GP[31:24]								
0x48 ... 0x5F	Reserved										
0x60	TAMPCTRL	7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]		
0x61		15:8							IN4ACT[1:0]		
0x62		23:16				TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	
0x63		31:24				DEBNCn	DEBNCn	DEBNCn	DEBNCn	DEBNCn	
0x64	TIMESTAMP	7:0	SECOND[2:0]								
0x65		15:8	MINUTE[4:0]					SECOND[5:3]			
0x66		23:16	DAY[2:0]			HOUR[4:0]					
0x67		31:24	YEAR[0:0]			MONTH[3:0]				DAY[4:3]	
0x68	TAMPID	7:0				TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	
0x69		15:8									
0x6A		23:16									
0x6B		31:24	TAMPEVT								
0x6C ... 0x7F	Reserved										
0x80	BKUP0	7:0	BKUP[7:0]								
0x81		15:8	BKUP[15:8]								
0x82		23:16	BKUP[23:16]								
0x83		31:24	BKUP[31:24]								
0x84	BKUP1	7:0	BKUP[7:0]								
0x85		15:8	BKUP[15:8]								
0x86		23:16	BKUP[23:16]								
0x87		31:24	BKUP[31:24]								
0x88	BKUP2	7:0	BKUP[7:0]								
0x89		15:8	BKUP[15:8]								
0x8A		23:16	BKUP[23:16]								
0x8B		31:24	BKUP[31:24]								
0x8C	BKUP3	7:0	BKUP[7:0]								
0x8D		15:8	BKUP[15:8]								
0x8E		23:16	BKUP[23:16]								
0x8F		31:24	BKUP[31:24]								
0x90	BKUP4	7:0	BKUP[7:0]								
0x91		15:8	BKUP[15:8]								
0x92		23:16	BKUP[23:16]								
0x93		31:24	BKUP[31:24]								
0x94	BKUP5	7:0	BKUP[7:0]								
0x95		15:8	BKUP[15:8]								
0x96		23:16	BKUP[23:16]								
0x97		31:24	BKUP[31:24]								
0x98	BKUP6	7:0	BKUP[7:0]								
0x99		15:8	BKUP[15:8]								

Offset	Name	Bit Pos.								
0x9A		23:16	BKUP[23:16]							
0x9B		31:24	BKUP[31:24]							
0x9C	BKUP7	7:0	BKUP[7:0]							
0x9D		15:8	BKUP[15:8]							
0x9E		23:16	BKUP[23:16]							
0x9F		31:24	BKUP[31:24]							

25.12 Register Description - CLOCK

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

25.12.1 Control A in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC	GPTRST	BKTRST		PRESCALER[3:0]			
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 15 – CLOCKSYNC: CLOCK Read Synchronization Enable
The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
This bit is not enable-protected.

Value	Description
0	CLOCK read synchronization is disabled
1	CLOCK read synchronization is enabled

Bit 14 – GPTRST: GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Bit 13 – BKTRST: GP Registers Reset On Tamper Enable
All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Value	Description
0	BKUPn registers will not reset when a tamper condition occurs.
1	BKUPn registers will reset when a tamper condition occurs.

Bits 11:8 – PRESCALER[3:0]: Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR: Clear on Match
This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP: Clock Representation
This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0]: Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.12.2 Control B in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLB
Offset: 2
Reset: 0x0
Property: Read-Write

Bit	15	14	13	12	11	10	9	8
		ACTF[2:0]				DEBF[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ			GP2EN	GP0EN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bits 14:12 – ACTF[2:0]: Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0]: Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN: DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT: RTC Out Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC: Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ: Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 1 – GP2EN: General Purpose 2 Enable

Value	Description
0	COMP1 compare function enabled. GP2 disabled.
1	COMP1 compare function disabled. GP2 enabled.

Bit 0 – GP0EN: General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0 disabled.
1	COMP0 compare function disabled. GP0 enabled.

25.12.3 Event Control in Clock/Calendar mode (CTRLA.MODE=2)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO						ALARMO0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – TAMPEVEI: Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored.
1	Tamper event input is enabled, and all incoming events will capture the CLOCK value.

Bit 15 – OVFEO: Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO: Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated
1	Tamper event output is enabled, and will be generated for every tamper input.

Bit 8 – ALARMO0: Alarm 0 Event Output Enable

Value	Description
0	Alarm 0 event is disabled and will not be generated.
1	Alarm 0 event is enabled and will be generated for every compare match.

Bits 7:0 – PEREOn: Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

25.12.4 Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER: Tamper Interrupt Enable

Bit 8 – ALARM0: Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm 0 Interrupt Enable bit, which disables the Alarm interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.12.5 Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER: Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt it disabled.
1	The Tamper interrupt is enabled.

Bit 8 – ALARM0: Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm 0 Interrupt Enable bit, which enables the Alarm 0 interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.12.6 Interrupt Flag Status and Clear in Clock/Calendar mode (CTRLA.MODE=2)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	PERn	PERn	PERn	PERn	PERn	PERn	PERn	PERn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER: Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bit 8 – ALARM0: Alarm 0
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.ALARM0 is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Alarm 0 interrupt flag.

Bits 7:0 – PERn: Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.12.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

25.12.8 Synchronization Busy in Clock/Calendar mode (CTRLA.MODE=2)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPn	GPn
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC				MASK0			
Access	R				R			
Reset	0				0			

Bit	7	6	5	4	3	2	1	0
			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 17:16 – GPn: General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

Bit 15 – CLOCKSYNC: Clock Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.CLOCKSYNC bit is complete.
1	Write synchronization for CTRLA.CLOCKSYNC bit is ongoing.

Bit 11 – MASK0: Mask 0 Synchronization Busy Status

Value	Description
0	Write synchronization for MASK0 register is complete.
1	Write synchronization for MASK0 register is ongoing.

Bit 5 – ALARM0: Alarm 0 Synchronization Busy Status

Value	Description
0	Write synchronization for ALARM0 register is complete.
1	Write synchronization for ALARM0 register is ongoing.

Bit 3 – CLOCK: Clock Register Synchronization Busy Status

Value	Description
0	Read/write synchronization for CLOCK register is complete.
1	Read/write synchronization for CLOCK register is ongoing.

Bit 2 – FREQCORR: Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for FREQCORR register is complete.
1	Read/write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.ENABLE bit is complete.
1	Read/write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.SWRST bit is complete.
1	Read/write synchronization for CTRLA.SWRST bit is ongoing.

25.12.9 Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN: Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0]: Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.12.10 Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0]: Year
The year offset with respect to the reference year (defined in software).

The year is considered a leap year if YEAR[1:0] is zero.

Bits 25:22 – MONTH[3:0]: Month
1 – January

2 – February

...

12 – December

Bits 21:17 – DAY[4:0]: Day
Day starts at 1 and ends at 28, 29, 30, or 31, depending on the month and year.

Bits 16:12 – HOUR[4:0]: Hour
When CTRLA.CLKREP=0, the Hour bit group is in 24-hour format, with values 0-23. When CTRLA.CLKREP=1, HOUR[3:0] has values 1-12, and HOUR[4] represents AM (0) or PM (1).

Bits 11:6 – MINUTE[5:0]: Minute
0 – 59

Bits 5:0 – SECOND[5:0]: Second
0 – 59

25.12.11 Alarm Value in Clock/Calendar mode (CTRLA.MODE=2)

The 32-bit value of ALARM is continuously compared with the 32-bit CLOCK value, based on the masking set by MASK.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARM) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

Name: ALARM
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0]: Year
The alarm year. Years are only matched if MASK.SEL is 6

Bits 25:22 – MONTH[3:0]: Month
The alarm month. Months are matched only if MASK.SEL is greater than 4.

Bits 21:17 – DAY[4:0]: Day
The alarm day. Days are matched only if MASK.SEL is greater than 3.

Bits 16:12 – HOUR[4:0]: Hour
The alarm hour. Hours are matched only if MASK.SEL is greater than 2.

Bits 11:6 – MINUTE[5:0]: Minute
The alarm minute. Minutes are matched only if MASK.SEL is greater than 1.

Bits 5:0 – SECOND[5:0]: Second
The alarm second. Seconds are matched only if MASK.SEL is greater than 0.

25.12.12 Alarm Mask in Clock/Calendar mode (CTRLA.MODE=2)

Name: MASK
Offset: 0x24
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						SEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SEL[2:0]: Alarm Mask Selection
These bits define which bit groups of ALARM are valid.

Value	Name	Description
0x0	OFF	Alarm Disabled
0x1	SS	Match seconds only
0x2	MMSS	Match seconds and minutes only
0x3	HHMMSS	Match seconds, minutes, and hours only
0x4	DDHHMMSS	Match seconds, minutes, hours, and days only
0x5	MMDDHHMMSS	Match seconds, minutes, hours, days, and months only
0x6	YYMMDDHHMMSS	Match seconds, minutes, hours, days, months, and years
0x7	-	Reserved

25.12.13 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0]: General Purpose
These bits are for user-defined general purpose use, see General Purpose Registers.

25.12.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				DEBNCn	DEBNCn	DEBNCn	DEBNCn	DEBNCn
Access								
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn	TAMLVLn
Access								
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							IN4ACT[1:0]	
Access								
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 28:24 – DEBNCn: Debounce Enable n

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 20:16 – TAMLVLn: Tamper Level Select n

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 8:9 – IN0ACT, IN1ACT, IN2ACT, IN3ACT, IN4ACT: Tamper Input n Action
These bits determine the action taken by Tamper Input INn.

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare INn to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

25.12.15 Timestamp Value

Name: TIMESTAMP
Offset: 0x64
Reset: 0
Property: R

Bit	31	30	29	28	27	26	25	24
	YEAR[0:0]			MONTH[3:0]				DAY[4:3]
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			HOUR[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MINUTE[4:0]					SECOND[5:3]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SECOND[2:0]							
Access	R	R	R					
Reset	0	0	0					

Bits 36:31 – YEAR[5:0]: Year
The year value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 28:25 – MONTH[3:0]: Month
The month value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 25:21 – DAY[4:0]: Day
The day value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 20:16 – HOUR[4:0]: Hour
The hour value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 16:11 – MINUTE[5:0]: Minute
The minute value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 10:5 – SECOND[5:0]: Second
The second value is captured by the TIMESTAMP when a tamper condition occurs.

25.12.16 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn	TAMPIDn
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 31 – TAMPEVT: Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 4:0 – TAMPIDn: Tamper Input n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input condition has not been detected on INn pin
1	A tamper input condition has been detected in INn pin

25.12.17 Backup n

Name: BKUP
Offset: 0x80 + n*0x04 [n=0..7]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	BKUP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BKUP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BKUP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BKUP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BKUP[31:0]: Backup

These bits are user-defined for general purpose use in the Backup domain.

26. DMAC – Direct Memory Access Controller

26.1 Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the Block Diagram. The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer.
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC.
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- The *write-back bus* is used to write the transfer descriptor back to SRAM.

All buses are AHB master interfaces but the AHB/APB Bridge bus, which is an APB slave interface.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

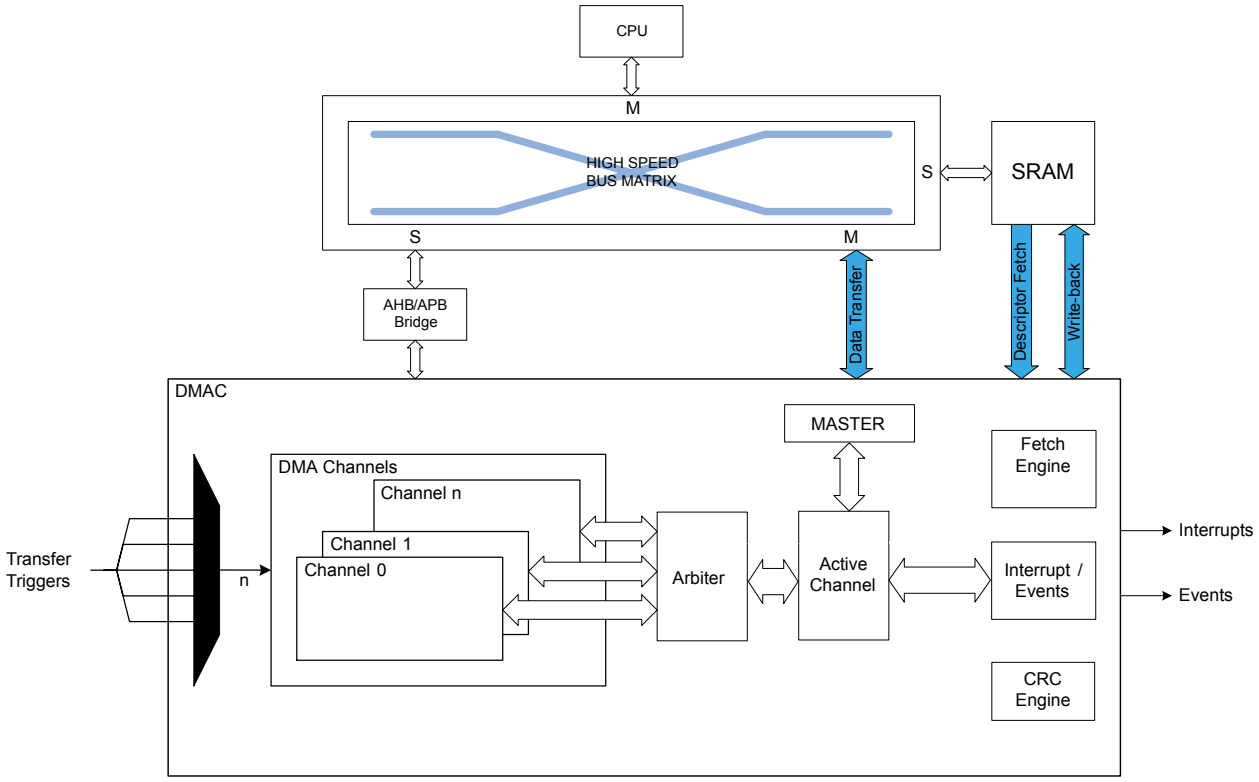
26.2 Features

- Data transfer from:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer trigger sources
  - Software
  - Events from Event System
  - Dedicated requests from peripherals
- SRAM based transfer descriptors
  - Single transfer using one descriptor
  - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 16channels

  - Enable 16 independent transfers
  - Automatic descriptor fetch for each channel
  - Suspend/resume operation support for each channel
- Flexible arbitration scheme
  - 4 configurable priority levels for each channel
  - Fixed or round-robin priority scheme within each priority level
- From 1 to 256KB data transfer in a single block transfer
- Multiple addressing modes
  - Static
  - Configurable increment scheme
- Optional interrupt generation
  - On block transfer complete
  - On error detection
  - On channel suspend
- 4 event inputs
  - One event input for each of the 4 least significant DMA channels
  - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
  - Can be selected to suspend or resume channel operation
- 4 event outputs
  - One output event for each of the 4 least significant DMA channels
  - Selectable generation on AHB, block, or transaction transfer complete
- Error management supported by write-back function
  - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC polynomial software selectable to
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE® 802.3)

26.3 Block Diagram

Figure 26-1. DMAC Block Diagram



26.4 Signal Description

Not applicable.

26.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

26.5.1 I/O Lines

Not applicable.

26.5.2 Power Management

The DMAC will continue to operate in any sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. On hardware or software reset, all registers are set to their reset value.

Related Links

PM – Power Manager

26.5.3 Clocks

The DMAC bus clock (CLK_DMAC_APB) must be configured and enabled in the Main Clock module before using the DMAC.

This bus clock (CLK_DMAC_APB) is always synchronous to the module clock (CLK_DMAC_AHB), but can be divided by a prescaler and may run even when the module clock is turned off.

Related Links

Peripheral Clock Masking

26.5.4 DMA

Not applicable.

26.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

26.5.6 Events

The events are connected to the event system.

Related Links

EVSYS – Event System

26.5.7 Debug Operation

When the CPU is halted in debug mode the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. Refer to DBGCTRL for details.

26.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Pending register (INTPEND)
- Channel ID register (CHID)
- Channel Interrupt Flag Status and Clear register (CHINTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

26.5.9 Analog Connections

Not applicable.

26.6 Functional Description

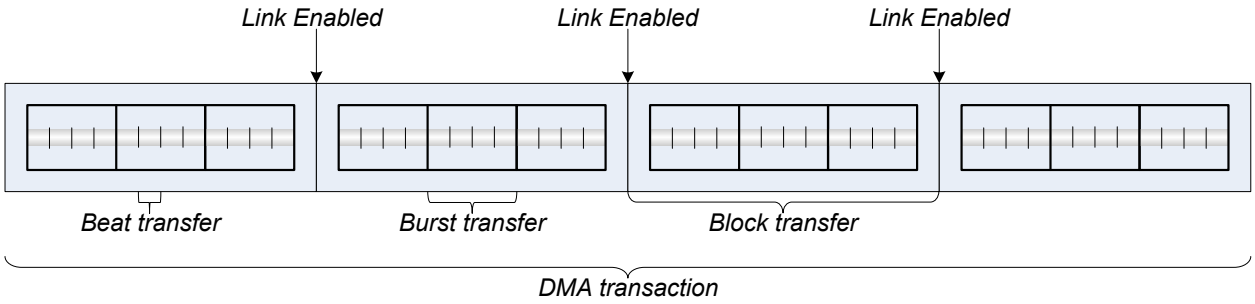
26.6.1 Principle of Operation

The DMAC consists of a DMA module and a CRC module.

26.6.1.1 DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. Figure 'DMA Transfer Sizes' shows the relationship between the different transfer sizes:

Figure 26-2. DMA Transfer Sizes



- Beat transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
- Block transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. A block transfer can be interrupted.
- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer should be carried out by the DMAC, and it must remain in SRAM. For further details on the transfer descriptor refer to Transfer Descriptors.

The figure above shows several block transfers linked together, which are called linked descriptors. For further information about linked descriptors, refer to Linked Descriptors.

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, dependent of the configuration, the DMA channel will either be suspended or disabled.

26.6.1.2 CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. Refer to CRC Operation for details.

26.6.2 Basic Operation

26.6.2.1 Initialization

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- Descriptor Base Memory Address register (BASEADDR)
- Write-Back Memory Base Address register (WRBADDR)

The following DMAC bit is enable-protected, meaning that it can only be written when both the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CTRL.CRCENABLE=0):

- Software Reset bit in Control register (CTRL.SWRST)

The following DMA channel register is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled (CHCTRLA.ENABLE=0):

- Channel Control B (CHCTRLB) register, except the Command bit (CHCTRLB.CMD) and the Channel Arbitration Level bit (CHCTRLB.LVL)

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- Channel Software Reset bit in Channel Control A register (CHCTRLA.SWRST)

The following CRC registers are enable-protected, meaning that they can only be written when the CRC is disabled (CTRL.CRCENABLE=0):

- CRC Control register (CRCCTRL)
- CRC Checksum register (CRCCHKSUM)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before the DMAC is enabled it must be configured, as outlined by the following steps:

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register
- The SRAM address of where the write-back section should be located must be written to the Write-Back Memory Base Address (WRBADDR) register
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA channel configurations
  - The channel number of the DMA channel to configure must be written to the Channel ID (CHID) register
  - Trigger action must be selected by writing the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT)
  - Trigger source must be selected by writing the Trigger Source bit group in the Channel Control B register (CHCTRLB.TRIGSRC)
- Transfer Descriptor
  - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
  - Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register
  - Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register
  - Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)

- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control register (CRCCTRL.CRCBEATSIZE)

26.6.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the reset to take effect.

26.6.2.3 Transfer Descriptors

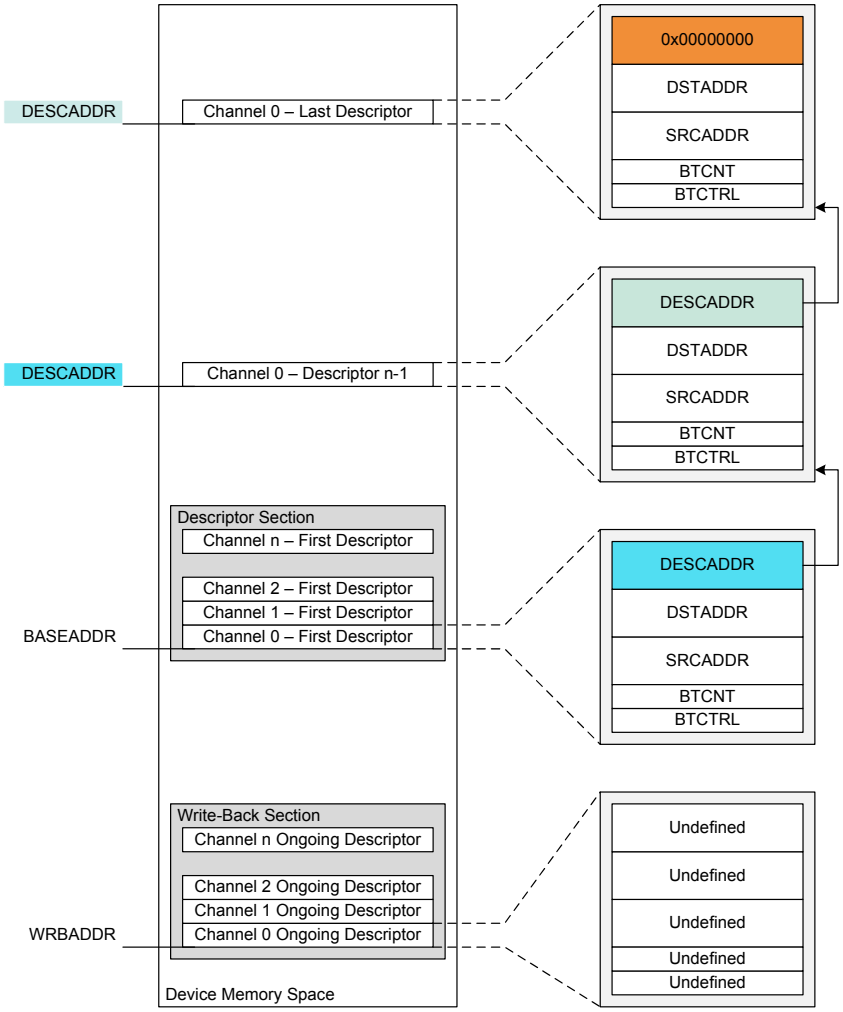
Together with the channel configurations the transfer descriptors decides how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one), and receives a transfer trigger, its first transfer descriptor has to be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel 0 (see figure below), all first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number. For further details on linked descriptors, refer to Linked Descriptors.

The write-back memory section is the section where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel 0. All ongoing transfer descriptors will be stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel 0. For further details on linked descriptors, refer to Linked Descriptors.

Figure 26-3. Memory Sections



The size of the descriptor and write-back memory sections is dependent on the number of the most significant enabled DMA channel *m*, as shown below:

$$Size = 128\text{bits} \cdot (m + 1)$$

For memory optimization, it is recommended to always use the less significant DMA channels if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor. The benefit of having descriptor memory and write-back memory in the same section is that it requires less SRAM.

26.6.2.4 Arbitration

If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The active channel is the DMA channel being granted access to perform its next transfer. When the arbiter has granted a DMA channel access to the DMAC, the corresponding bit PENDCH.PENDCHx will be cleared. See also the following figure.

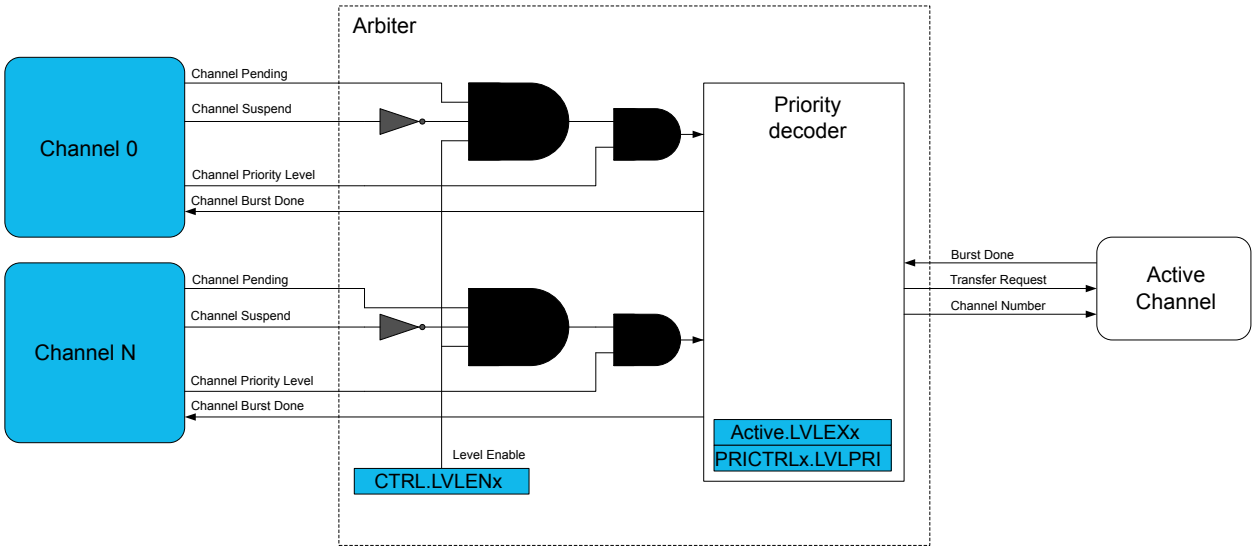
If the upcoming transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register will be set (BUSYCH.BUSYCHx=1), and it will remain '1' for the subsequent granted transfers.

When the channel has performed its granted transfer(s) it will be either fed into the queue of channels with pending transfers, set to be waiting for a new transfer trigger, suspended, or disabled. This depends on the channel and block transfer configuration. If the DMA channel is fed into the queue of channels with pending transfers, the corresponding BUSYCH.BUSYCHx will remain '1'. If the DMA channel is set to wait for a new transfer trigger, suspended, or disabled, the corresponding BUSYCH.BUSYCHx will be cleared.

If a DMA channel is suspended while it has a pending transfer, it will be removed from the queue of pending channels, but the corresponding PENDCH.PENDCHx will remain set. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE=0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

Figure 26-4. Arbiter Overview



Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

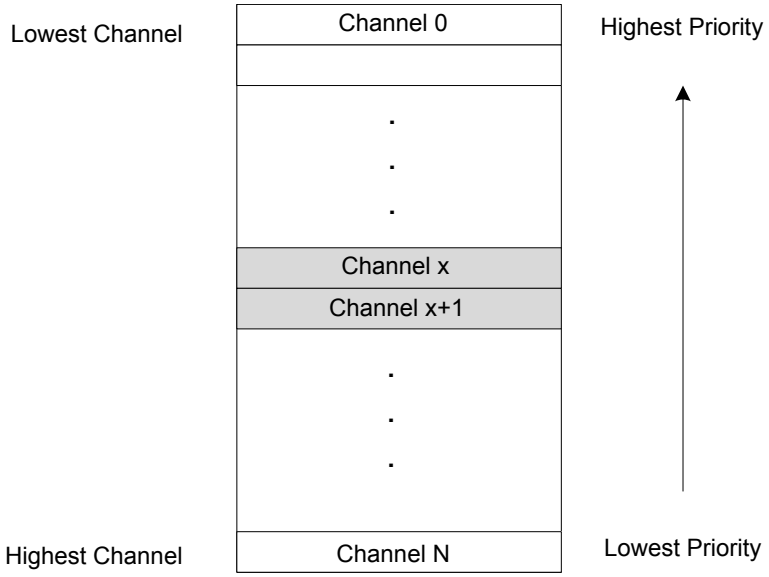
Each DMA channel supports a 4-level priority scheme. The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Control B register (CHCTRLB.LVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority level number. Each priority level x is enabled by setting the corresponding Priority Level x Enable bit in the Control register (CTRL.LVLENx=1).

Within each priority level the DMAC's arbiter can be configured to prioritize statically or dynamically:

Static Arbitration within a priority level is selected by writing a '0' to the Level x Round-Robin Scheduling Enable bit in the Priority Control 0 register (PRICTRL0.RRLVLENx).

When static arbitration is selected, the arbiter will prioritize a low channel number over a high channel number as shown in the figure below. When using the static arbitration there is a risk of high channel numbers never being granted access as the active channel. This can be avoided using a dynamic arbitration scheme.

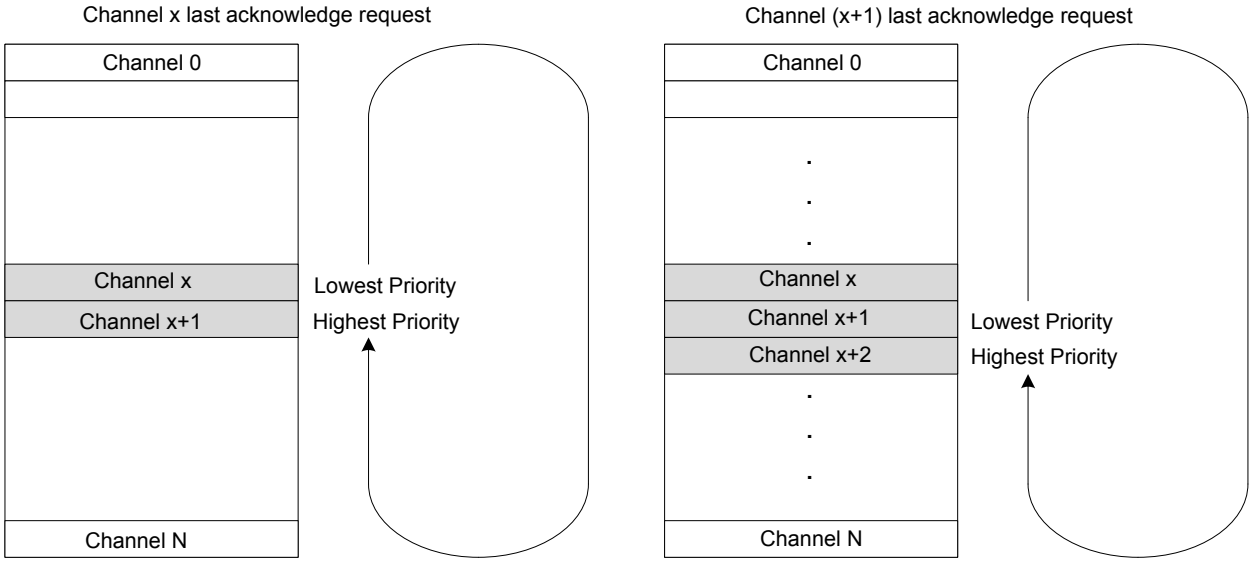
Figure 26-5. Static Priority Scheduling



Dynamic Arbitration within a priority level is selected by writing a '1' to PRICTRL0.RRLVLENx.

The dynamic arbitration scheme in the DMAC is round-robin. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in Figure 26-6. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control 0 register (PRICTRL0.LVLPRIx) for the corresponding priority level.

Figure 26-6. Dynamic (Round-Robin) Priority Scheduling



26.6.2.5 Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (refer to DMA Block Diagram section) the transfer descriptor for the DMA channel will be fetched from SRAM using the fetch bus, and

stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated, refer to the section on Addressing.

The arbitration procedure is performed after each transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

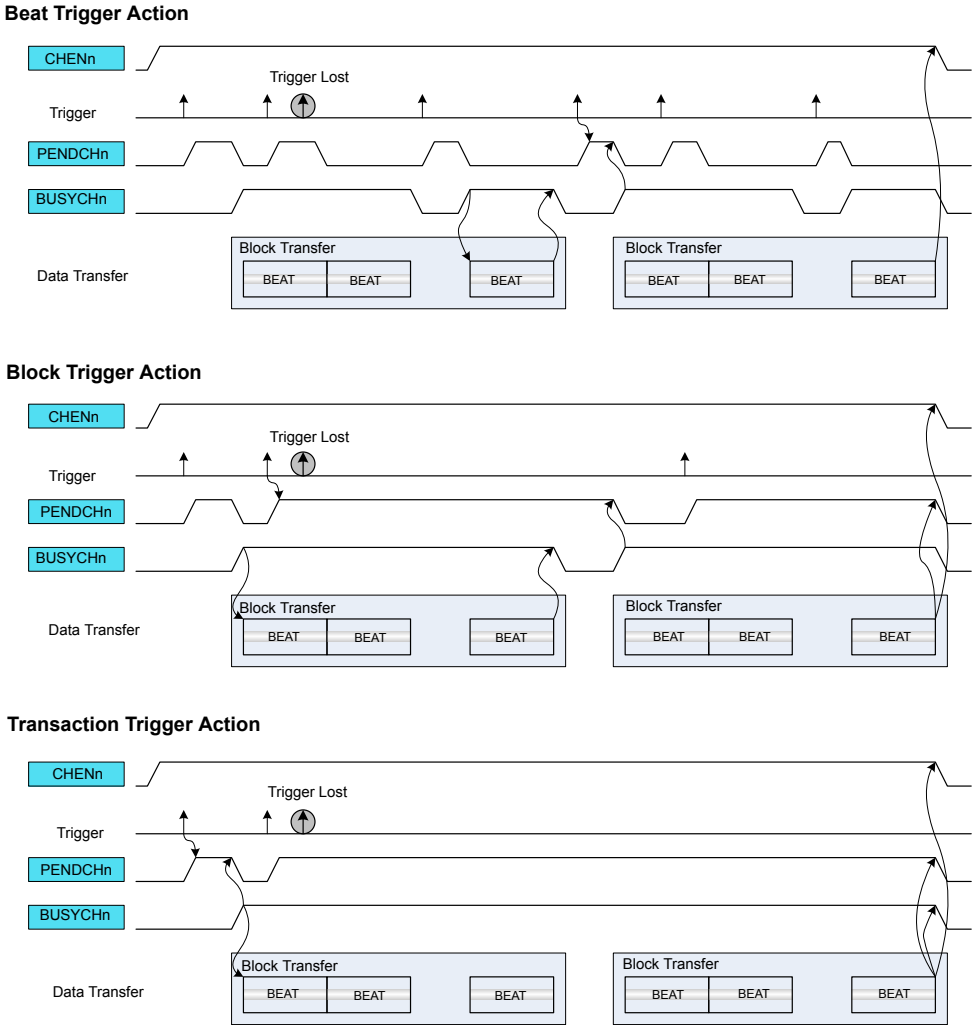
26.6.2.6 Transfer Triggers and Actions

A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. If the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a beat transfer (CHCTRLB.TRIGACT=0x2) or transaction transfer (CHCTRLB.TRIGACT=0x3) instead of a block transfer (CHCTRLB.TRIGACT=0x0).

Figure 26-7 shows an example where triggers are used with two linked block descriptors.

Figure 26-7. Trigger Action and Transfers



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUS.PEND=1), and the new transfer can start after the ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel Status register (CHSTATUS.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

26.6.2.7 Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (SRCADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address

Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\text{SRCADDR} = \text{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\text{STEPSIZE}}$$

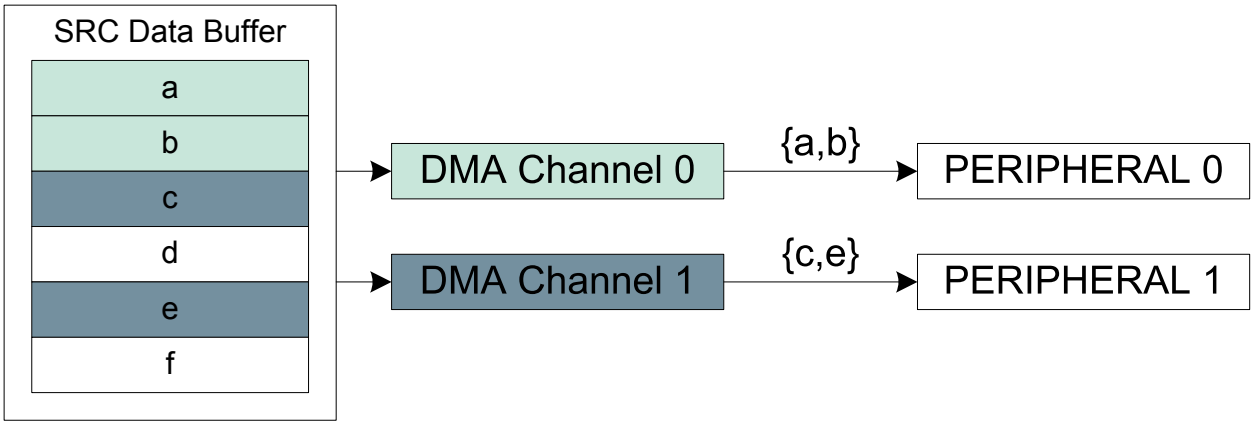
If BTCTRL.STEPSEL=0:

$$\text{SRCADDR} = \text{SRCADDR}_{START} + BTCNT \cdot (BEATSIZE + 1)$$

- $\text{SRCADDR}_{\text{START}}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1, and BTCTRL.STEPSIZE=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

Figure 26-8. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

When the destination address incrementation is configured (BTCTRL.DSTINC=1), SRCADDR must be set and calculated as follows:

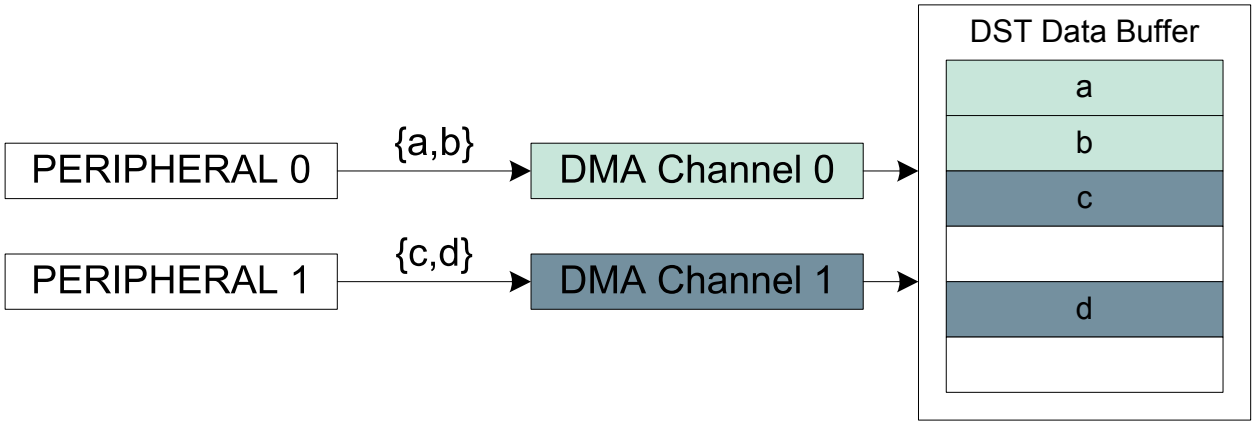
$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$	where BTCTRL.STEPSEL is one

- $\text{DSTADDR}_{\text{START}}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer

- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

Figure 26-9shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC=1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC=1, BTCTRL.STEPSEL=0, and BTCTRL.STEPSIZE=0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC=0).

Figure 26-9. Destination Address Increment



26.6.2.8 Error Handling

If a bus error is received from an AHB slave during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID=0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR=0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

26.6.3 Additional Features

26.6.3.1 Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consist of several block transfers it is called linked descriptors.

Figure Figure 26-3 illustrates how linked descriptors work. When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR=0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from SRAM, refer to section Data Transmission.

Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in SRAM, with DESCADDR=0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in SRAM to configure a new descriptor.
4. Configure the new descriptor:
   – Set the next descriptor address (DESCADDR)
   – Set the destination address (DSTADDR)
   – Set the source address (SRCADDR)
   – Configure the block transfer control (BTCTRL) including
      • Optionally enable the Suspend block action
      • Set the descriptor VALID bit
5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read DESCADDR from the Write-Back memory.
   – If the DMA has not already fetched the descriptor which requires changes (i.e., DESCADDR is wrong):
      • Update the DESCADDR location of the descriptor from the List
      • Optionally clear the Suspend block action
      • Set the descriptor VALID bit to '1'
      • Optionally enable the Resume software command
   – If the DMA is executing the same descriptor as the one which requires changes:
      • Set the Channel Suspend software command and wait for the Suspend interrupt
      • Update the next descriptor address (DESCRADDR) in the write-back memory
      • Clear the interrupt sources and set the Resume software command
      • Update the DESCADDR location of the descriptor from the List
      • Optionally clear the Suspend block action
      • Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow the steps:
   2.1. Set the descriptor A VALID bit to '0'.
   2.2. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
   2.3. Set the DESCADDR value of descriptor C to point to descriptor B.
   2.4. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:
   3.1. Apply the software suspend command to the channel and
   3.2. Perform steps 2.1 through 2.4.
   3.3. Apply the software resume command to the channel.

26.6.3.2 Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register(CHASTATUS.FERR) will be set.

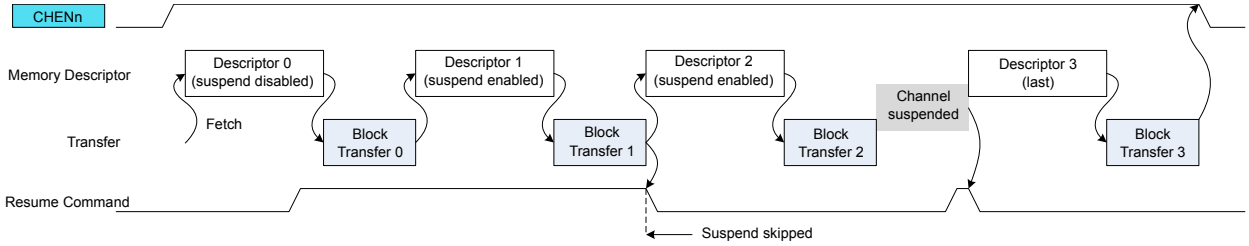
Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors, refer to section Transfer Descriptors.

26.6.3.3 Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

Figure 26-10. Channel Suspend/Resume Operation



26.6.3.4 Event Input Actions

The event input actions are available only on the least significant DMA channels. For details on channels with event input support, refer to the in the Event system documentation.

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to Events.

Table 26-1. Event Input Action

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
None	NOACT	-
Normal Transfer	TRIG	DISABLE
Conditional Transfer on Strobe	TRIG	any peripheral
Conditional Transfer	CTRIG	
Conditional Block Transfer	CBLOCK	
Channel Suspend	SUSPEND	

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
Channel Resume	RESUME	
Skip Next Block Suspend	SSKIP	

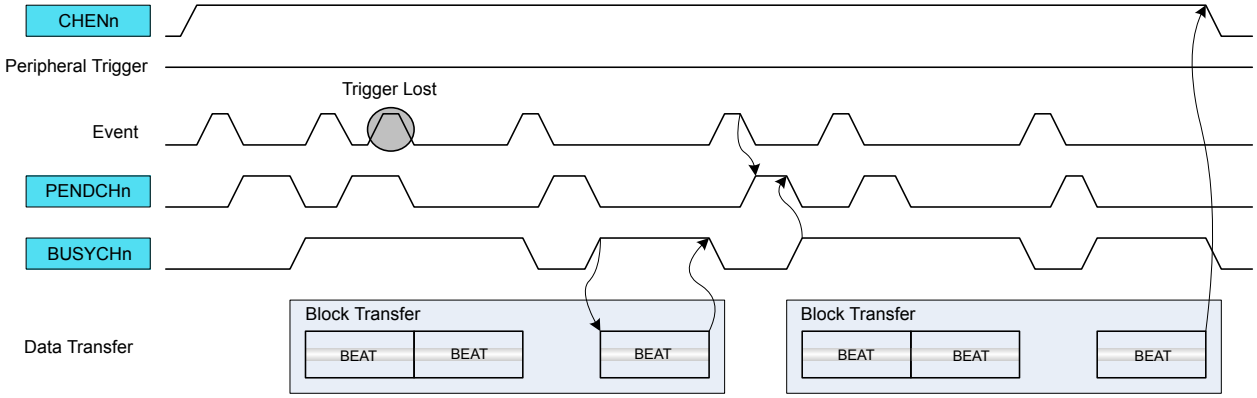
Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The figure below shows an example where beat transfers are enabled by internal events.

Figure 26-11. Beat Event Trigger Action



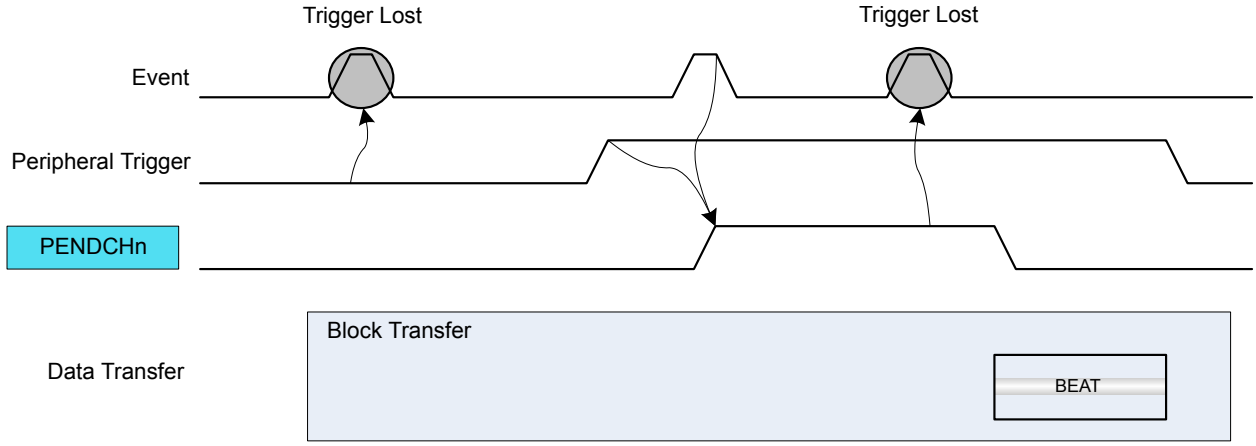
Conditional Transfer on Strobe

The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, e.g. for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (i.e. the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA will wait for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both CHSTATUS.PEND and PENDCH.PENDCHn are set. A software trigger will now trigger a transfer.

The figure below shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Figure 26-12. Periodic Event with Beat Peripheral Triggers



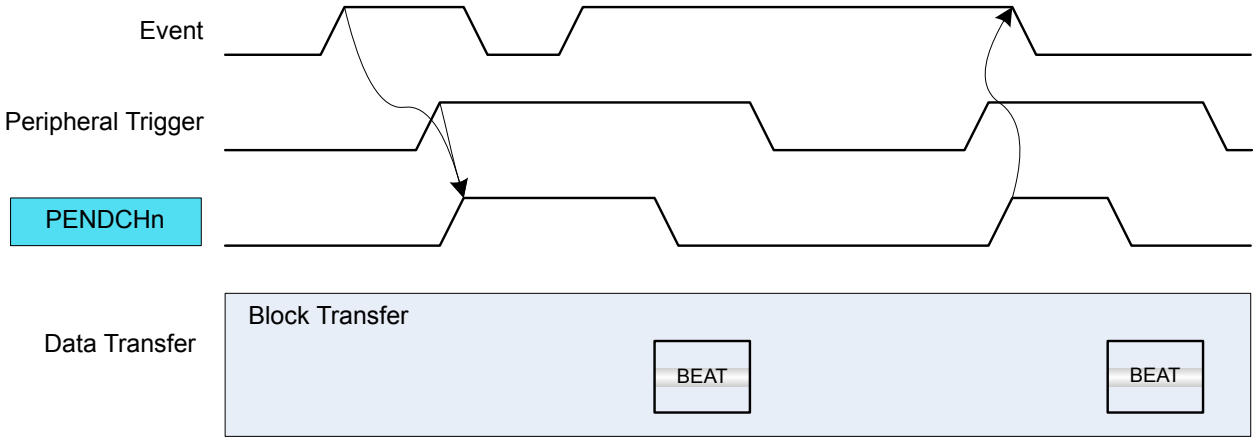
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 26-13. Conditional Event with Beat Peripheral Triggers



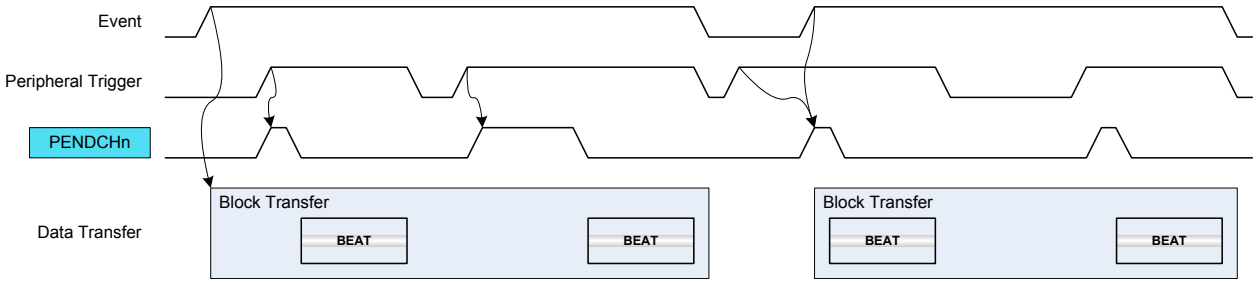
Conditional Block Transfer

The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Figure 26-14. Conditional Block Transfer with Beat Peripheral Triggers



Channel Suspend

The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For further details on Channel Suspend, refer to Channel Suspend.

Channel Resume

The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. For further details refer to Channel Suspend.

Skip Next Block Suspend

This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

26.6.3.5 Event Output Selection

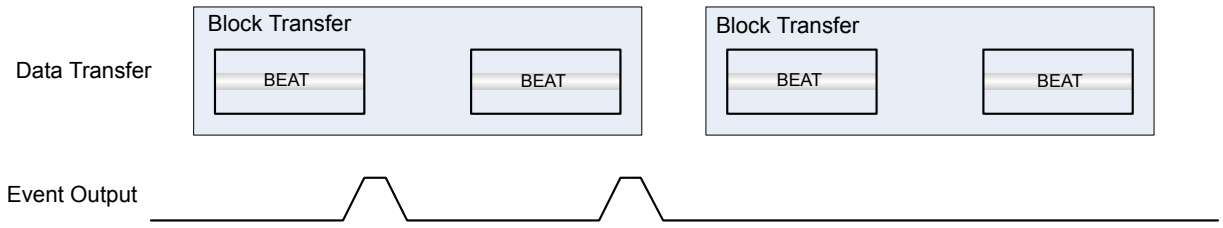
Event output selection is available only for the least significant DMA channels. The pulse width of an event output from a channel is one AHB clock cycle.

The output of channel events is enabled by writing a '1' to the Channel Event Output Enable bit in the Control B register (CHCTRLB.EVOE). The event output cause is selected by writing to the Event Output Selection bits in the Block Transfer Control register (BTCTRL.EVOSEL). It is possible to generate events after each block transfer (BTCTRL.EVOSEL=0x1) or beat transfer (BTCTRL.EVOSEL=0x3). To enable an event being generated when a transaction is complete, the block event selection must be set in the last transfer descriptor only.

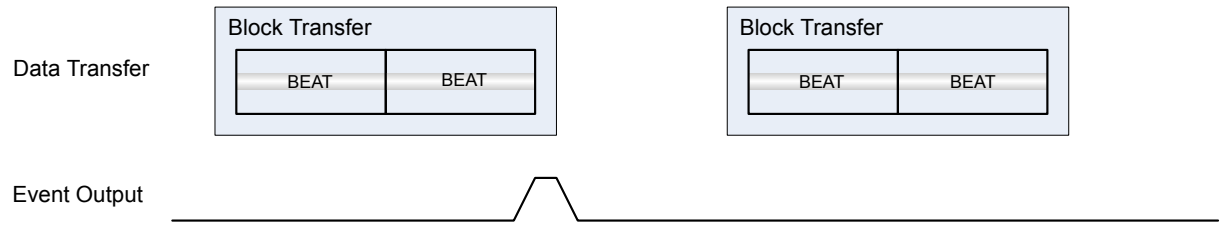
The figure Figure 26-15 shows an example where the event output generation is enabled in the first block transfer, and disabled in the second block.

Figure 26-15. Event Output Generation

Beat Event Output



Block Event Output



26.6.3.6 Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

26.6.3.7 CRC Operation

A cyclic redundancy check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation: If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

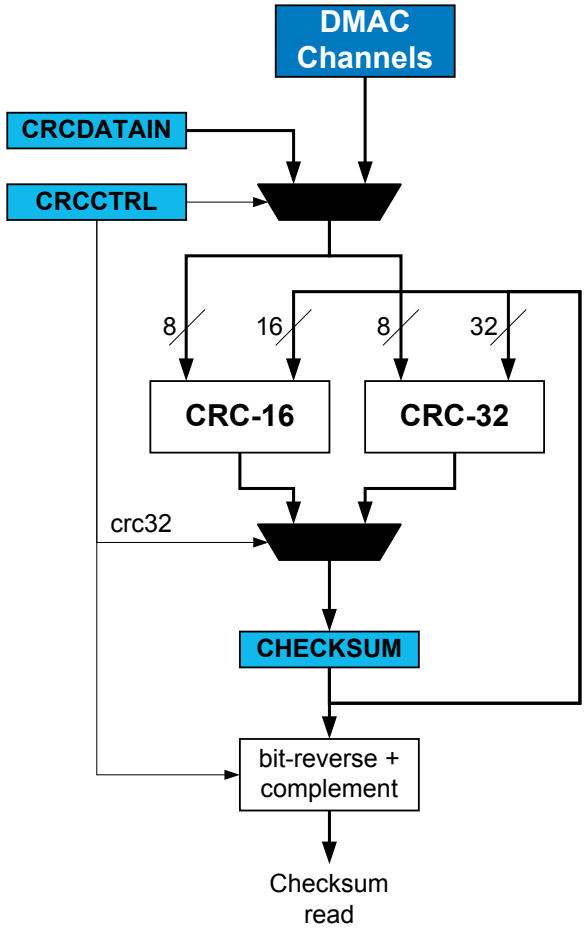
The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in Figure 26-16.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16-, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.

Figure 26-16. CRC Generator Block Diagram



CRC on DMA data CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.

CRC using the I/O interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

26.6.4 DMA Operation

Not applicable.

26.6.5 Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. Refer to Data Transmission for details.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to Error Handling for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. Refer to Channel Suspend and Data Transmission for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller

26.6.6 Events

The DMAC can generate the following output events:

- Channel (CH): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. Refer to Event Output Selection for details.

Setting the Channel Control B Event Output Enable bit (CHCTRLB.EVOE=1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHCTRLB.EVOE=0 disables the corresponding output event.

The DMAC can take the following actions on an input event:

- Transfer and Periodic Transfer Trigger (TRIG): normal transfer or periodic transfers on peripherals are enabled
- Conditional Transfer Trigger (CTRIG): conditional transfers on peripherals are enabled
- Conditional Block Transfer Trigger (CBLOCK): conditional block transfers on peripherals are enabled
- Channel Suspend Operation (SUSPEND): suspend a channel operation
- Channel Resume Operation (RESUME): resume a suspended channel operation
- Skip Next Block Suspend Action (SSKIP): skip the next block suspend transfer condition

Setting the Channel Control B Event Input Enable bit (CHCTRLB.EVIE=1) enables the corresponding action on input event. clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. For further details on event input actions, refer to Event Input Actions.

Related Links

EVSYS – Event System

26.6.7 Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to '1'. The DMAC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

Note: In standby sleep mode, the DMAC can only access RAM when it is not back biased (PM.STDBYCFG.BBIASxx=0x0)

26.6.8 Synchronization

Not applicable.

26.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0						CRCENABLE	DMAENABLE	SWRST
0x01		15:8					LVLENx3	LVLENx2	LVLENx1	LVLENx0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
0x03		15:8			CRCSRC[5:0]					
0x04	CRCDATAIN	7:0	CRCDATAIN[7:0]							
0x05		15:8	CRCDATAIN[15:8]							
0x06		23:16	CRCDATAIN[23:16]							
0x07		31:24	CRCDATAIN[31:24]							
0x08	CRCCHKSUM	7:0	CRCCHKSUM[7:0]							
0x09		15:8	CRCCHKSUM[15:8]							
0x0A		23:16	CRCCHKSUM[23:16]							
0x0B		31:24	CRCCHKSUM[31:24]							
0x0C	CRCSTATUS	7:0							CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E	QOSCTRL	7:0			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
0x0F	Reserved									
0x10	SWTRIGCTRL	7:0	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn
0x11		15:8	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn
0x12		23:16								
0x13		31:24								
0x14	PRICTRL0	7:0	RRLVLEN0				LVLPRI0[3:0]			
0x15		15:8	RRLVLEN1				LVLPRI1[3:0]			
0x16		23:16	RRLVLEN2				LVLPRI2[3:0]			
0x17		31:24	RRLVLEN3				LVLPRI3[3:0]			
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0					ID[3:0]			
0x21		15:8	PEND	BUSY	FERR			SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn
0x25		15:8	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn
0x26		23:16								
0x27		31:24								
0x28	BUSYCH	7:0	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn
0x29		15:8	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn
0x2A		23:16								
0x2B		31:24								
0x2C	PENDCH	7:0	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn
0x2D		15:8	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn
0x2E		23:16								
0x2F		31:24								

Offset	Name	Bit Pos.								
0x30	ACTIVE	7:0					LVLEXx	LVLEXx	LVLEXx	LVLEXx
0x31		15:8	ABUSY			ID[4:0]				
0x32		23:16	BTCNT[7:0]							
0x33		31:24	BTCNT[15:8]							
0x34	BASEADDR	7:0	BASEADDR[7:0]							
0x35		15:8	BASEADDR[15:8]							
0x36		23:16	BASEADDR[23:16]							
0x37		31:24	BASEADDR[31:24]							
0x38	WRBADDR	7:0	WRBADDR[7:0]							
0x39		15:8	WRBADDR[15:8]							
0x3A		23:16	WRBADDR[23:16]							
0x3B		31:24	WRBADDR[31:24]							
0x3C ... 0x3E	Reserved									
0x3F	CHID	7:0					ID[3:0]			
0x40	CHCTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x41 ... 0x43	Reserved									
0x44	CHCTRLB	7:0		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
0x45		15:8			TRIGSRC[5:0]					
0x46		23:16	TRIGACT[1:0]							
0x47		31:24							CMD[1:0]	
0x48 ... 0x4B	Reserved									
0x4C	CHINTENCLR	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS	7:0						FERR	BUSY	PEND

26.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

26.8.1 Control

Name: CTRL
Offset: 0x00 [ID-00001ece]
Reset: 0x00X0
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
					LVLENx3	LVLENx2	LVLENx1	LVLENx0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CRCENABLE	DMAENABLE	SWRST
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 8, 9, 10, 11 – LVLENx: Priority Level x Enable
When this bit is set, all requests with the corresponding level will be fed into the arbiter block. When cleared, all requests with the corresponding level will be ignored.

For details on arbitration schemes, refer to the Arbitration section.

These bits are not enable-protected.

Value	Description
0	Transfer requests for Priority level x will not be handled.
1	Transfer requests for Priority level x will be handled.

Bit 2 – CRCENABLE: CRC Enable
Writing a '0' to this bit will disable the CRC calculation when the CRC Status Busy flag is cleared (CRCSTATUS. CRCBUSY). The bit is zero when the CRC is disabled.

Writing a '1' to this bit will enable the CRC calculation.

Value	Description
0	The CRC calculation is disabled.
1	The CRC calculation is enabled.

Bit 1 – DMAENABLE: DMA Enable
Setting this bit will enable the DMA module.

Writing a '0' to this bit will disable the DMA module. When writing a '0' during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

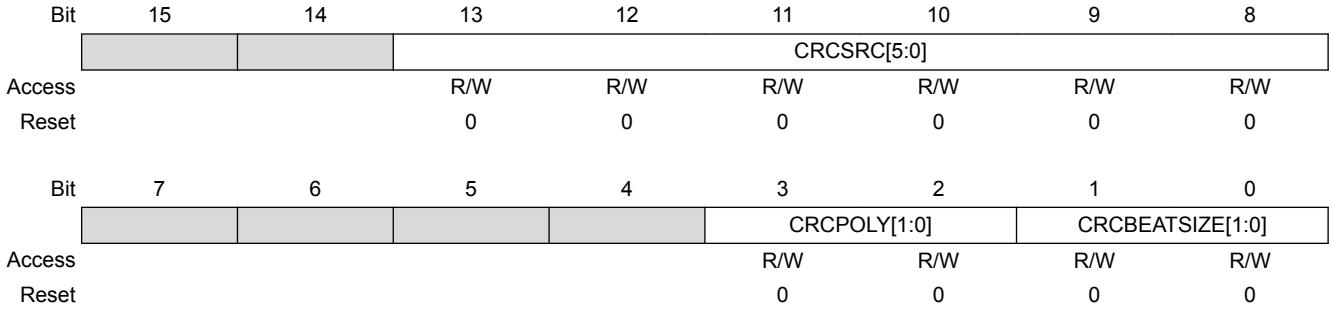
Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit when both the DMAC and the CRC module are disabled (DMAENABLE and CRCENABLE are '0') resets all registers in the DMAC (except DBGCTRL) to their initial state. If either the DMAC or CRC module is enabled, the Reset request will be ignored and the DMAC will return an access error.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

26.8.2 CRC Control

Name: CRCCTRL
Offset: 0x02 [ID-00001ece]
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected



Bits 13:8 – CRCSRC[5:0]: CRC Input Source
These bits select the input source for generating the CRC, as shown in the table below. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Value	Name	Description
0x00	NOACT	No action
0x01	IO	I/O interface
0x02-0x1F	-	Reserved
0x20	CHN	DMA channel 0
0x21	CHN	DMA channel 1
0x22	CHN	DMA channel 2
0x23	CHN	DMA channel 3
0x24	CHN	DMA channel 4
0x25	CHN	DMA channel 5
0x26	CHN	DMA channel 6
0x27	CHN	DMA channel 7
0x28	CHN	DMA channel 8
0x29	CHN	DMA channel 9
0x2A	CHN	DMA channel 10
0x2B	CHN	DMA channel 11

Value	Name	Description
0x2C	CHN	DMA channel 12
0x2D	CHN	DMA channel 13
0x2E	CHN	DMA channel 14
0x2F	CHN	DMA channel 15
0x30	CHN	DMA channel 16
0x31	CHN	DMA channel 17
0x32	CHN	DMA channel 18
0x33	CHN	DMA channel 19
0x34	CHN	DMA channel 20
0x35	CHN	DMA channel 21
0x36	CHN	DMA channel 22
0x37	CHN	DMA channel 23
0x38	CHN	DMA channel 24
0x39	CHN	DMA channel 25
0x3A	CHN	DMA channel 26
0x3B	CHN	DMA channel 27
0x3C	CHN	DMA channel 28
0x3D	CHN	DMA channel 29
0x3E	CHN	DMA channel 30
0x3F	CHN	DMA channel 31

Bits 3:2 – CRCPOLY[1:0]: CRC Polynomial Type
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)
0x2-0x3		Reserved

Bits 1:0 – CRCBEATSIZE[1:0]: CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3		Reserved

26.8.3 CRC Data Input

Name: CRCDATAIN
Offset: 0x04 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CRCDATAIN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCDATAIN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCDATAIN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCDATAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCDATAIN[31:0]: CRC Data Input
These bits store the data for which the CRC checksum is computed. A new CRC Checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.

26.8.4 CRC Checksum

The CRCCHKSUM represents the 16- or 32-bit checksum value and the generated CRC. The register is reset to zero by default, but it is possible to reset all bits to one by writing the CRCCHKSUM register directly. It is possible to write this register only when the CRC module is disabled. If CRC-32 is selected and the CRC Status Busy flag is cleared (i.e., CRC generation is completed or aborted), the bit reversed (bit 31 is swapped with bit 0, bit 30 with bit 1, etc.) and complemented result will be read from CRCCHKSUM. If CRC-16 is selected or the CRC Status Busy flag is set (i.e., CRC generation is ongoing), CRCCHKSUM will contain the actual content.

Name: CRCCHKSUM
Offset: 0x08 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	CRCCHKSUM[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCCHKSUM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCCHKSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCCHKSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCCHKSUM[31:0]: CRC Checksum
These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.

26.8.5 CRC Status

Name: CRCSTATUS
Offset: 0x0C [ID-00001ece]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							CRCZERO	CRCBUSY
Access							R	R/W
Reset							0	0

Bit 1 – CRCZERO: CRC Zero
This bit is cleared when a new CRC source is selected.

This bit is set when the CRC generation is complete and the CRC Checksum is zero.

When running CRC-32 and appending the checksum at the end of the packet (as little endian), the final checksum should be 0x2144df1c, and not zero. However, if the checksum is complemented before it is appended (as little endian) to the data, the final result in the checksum register will be zero. See the description of CRCCHKSUM to read out different versions of the checksum.

Bit 0 – CRCBUSY: CRC Module Busy
This flag is cleared by writing a one to it when used with I/O interface. When used with a DMA channel, the bit is set when the corresponding DMA channel is enabled, and cleared when the corresponding DMA channel is disabled. This register bit cannot be cleared by the application when the CRC is used with a DMA channel.

This bit is set when a source configuration is selected and as long as the source is using the CRC module.

26.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0D [ID-00001ece]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DMAC is halted when the CPU is halted by an external debugger.
1	The DMAC continues normal operation when the CPU is halted by an external debugger.

26.8.7 Quality of Service Control

Name: QOSCTRL
Offset: 0x0E [ID-00001ece]
Reset: 0x2A
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	1	0	1	0

Bits 5:4 – DQOS[1:0]: Data Transfer Quality of Service
These bits define the memory priority access during the data transfer operation.

DQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 3:2 – FQOS[1:0]: Fetch Quality of Service
These bits define the memory priority access during the fetch operation.

FQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 1:0 – WRBQOS[1:0]: Write-Back Quality of Service
These bits define the memory priority access during the write-back operation.

WRBQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

26.8.8 Software Trigger Control

Name: SWTRIGCTRL
Offset: 0x10 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn	SWTRIGn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SWTRIGn: Channel n Software Trigger [n = 15..0]
This bit is cleared when the Channel Pending bit in the Channel Status register (CHSTATUS.PEND) for the corresponding channel is either set, or by writing a '1' to it.

This bit is set if CHSTATUS.PEND is already '1' when writing a '1' to that bit.

Writing a '0' to this bit will clear the bit.

Writing a '1' to this bit will generate a DMA software trigger on channel x, if CHSTATUS.PEND=0 for channel x. CHSTATUS.PEND will be set and SWTRIGn will remain cleared.

26.8.9 Priority Control 0

Name: PRICTRL0
Offset: 0x14 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3				LVLPRI3[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RRLVLEN2				LVLPRI2[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RRLVLEN1				LVLPRI1[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RRLVLEN0				LVLPRI0[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 31 – RRLVLEN3: Level 3 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 3. For details on arbitration schemes, refer to Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 3 priority.
1	Round-robin arbitration scheme for channels with level 3 priority.

Bits 27:24 – LVLPRI3[3:0]: Level 3 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN3=1) for priority level 3, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 3.

When static arbitration is enabled (PRICTRL0.RRLVLEN3=0) for priority level 3, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN3 written to '0').

Bit 23 – RRLVLEN2: Level 2 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 2. For details on arbitration schemes, refer to Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 2 priority.
1	Round-robin arbitration scheme for channels with level 2 priority.

Bits 19:16 – LVLPRI2[3:0]: Level 2 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN2=1) for priority level 2, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 2.

When static arbitration is enabled (PRICTRL0.RRLVLEN2=0) for priority level 2, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN2 written to '0').

Bit 15 – RRLVLEN1: Level 1 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 1 priority.
1	Round-robin arbitration scheme for channels with level 1 priority.

Bits 11:8 – LVLPRI1[3:0]: Level 1 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN1=1) for priority level 1, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 1.

When static arbitration is enabled (PRICTRL0.RRLVLEN1=0) for priority level 1, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN1 written to '0').

Bit 7 – RRLVLEN0: Level 0 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 3:0 – LVLPRI0[3:0]: Level 0 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN0=1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.

When static arbitration is enabled (PRICTRL0.RRLVLEN0=0) for priority level 0, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN0 written to '0').

26.8.10 Interrupt Pending

This register allows the user to identify the lowest DMA channel with pending interrupt.

Name: INTPEND
Offset: 0x20 [ID-00001ece]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR			SUSP	TCMPL	TERR
Access	R	R	R			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – PEND: Pending
This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY: Busy
This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR: Fetch Error
This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 10 – SUSP: Channel Suspend
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Suspend interrupt flag.

Bit 9 – TCMPL: Transfer Complete
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Complete interrupt flag.

Bit 8 – TERR: Transfer Error
This bit is read one when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Error interrupt flag.

Bits 3:0 – ID[3:0]: Channel ID
These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is

refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

26.8.11 Interrupt Status

Name: INTSTATUS
Offset: 0x24 [ID-00001ece]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn	CHINTn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CHINTn: Channel n Pending Interrupt [n=15..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.

This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

26.8.12 Busy Channels

Name: BUSYCH
Offset: 0x28 [ID-00001ece]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn	BUSYCHn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BUSYCHn: Busy Channel n [x=15..0]
This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.

This bit is set when DMA channel n starts a DMA transfer.

26.8.13 Pending Channels

Name: PENDCH
Offset: 0x2C [ID-00001ece]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn	PENDCHn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PENDCHn: Pending Channel n [n=15..0]
This bit is cleared when trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.

This bit is set when a transfer is pending on DMA channel n.

26.8.14 Active Channel and Levels

Name: ACTIVE
Offset: 0x30 [ID-00001ece]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	BTCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BTCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ABUSY			ID[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					LVLEXx	LVLEXx	LVLEXx	LVLEXx
Access					R	R	R	R
Reset					0	0	0	0

Bits 31:16 – BTCNT[15:0]: Active Channel Block Transfer Count
These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel active busy flag (ABUSY) is set.

Bit 15 – ABUSY: Active Channel Busy
This bit is cleared when the active transfer count is written back in the write-back memory section.

This bit is set when the next descriptor transfer count is read from the write-back memory section.

Bits 12:8 – ID[4:0]: Active Channel ID
These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

Bits 3,2,1,0 – LVLEXx: Level x Channel Trigger Request Executing [x=3..0]
This bit is set when a level-x channel trigger request is executing or pending.

This bit is cleared when no request is pending or being executed.

26.8.15 Descriptor Memory Section Base Address

Name: BASEADDR
Offset: 0x34 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	BASEADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BASEADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BASEADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BASEADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BASEADDR[31:0]: Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 128-bit aligned.

26.8.16 Write-Back Memory Section Base Address

Name: WRBADDR
Offset: 0x38 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	WRBADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WRBADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WRBADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WRBADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WRBADDR[31:0]: Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 128-bit aligned.

26.8.17 Channel ID

Name: CHID
Offset: 0x3F [ID-00001ece]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – ID[3:0]: Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.

26.8.18 Channel Control A

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHCTRLA
Offset: 0x40 [ID-00001ece]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access	R	R/W	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 6 – RUNSTDBY: Channel run in standby
This bit is used to keep the DMAC channel running in standby mode.

This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in standby.
1	The DMAC channel continues to run in standby.

Bit 1 – ENABLE: Channel Enable
Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

Writing a '1' to this bit will enable the DMA channel.

This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST: Channel Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

26.8.19 Channel Control B

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHCTRLB
Offset: 0x44 [ID-00001ece]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	TRIGACT[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bit	15	14	13	12	11	10	9	8
			TRIGSRC[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 25:24 – CMD[1:0]: Software Command
These bits define the software commands. Refer to Channel Suspend and Channel Resume and Next Suspend Skip.

These bits are not enable-protected.

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Bits 23:22 – TRIGACT[1:0]: Trigger Action
These bits define the trigger action used for a transfer.

TRIGACT[1:0]	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1	-	Reserved
0x2	BEAT	One trigger required for each beat transfer
0x3	TRANSACTION	One trigger required for each transaction

Bits 13:8 – TRIGSRC[5:0]: Trigger Source
These bits define the peripheral trigger which is source of the transfer. For details on trigger selection and trigger modes, refer to Transfer Triggers and Actions and CHCTRLB.TRIGACT.

Value	Name	Description
0x00	DISABLE	Only software/event triggers

Value	Name	Description
0x01	RTC TIMESTAMP	RTC Timestamp Trigger
0x02	SERCOM0 RX	SERCOM0 RX Trigger
0x03	SERCOM0 TX	SERCOM0 TX Trigger
0x04	SERCOM1 RX	SERCOM1 RX Trigger
0x05	SERCOM1 TX	SERCOM1 TX Trigger
0x06	SERCOM2 RX	SERCOM2 RX Trigger
0x07	SERCOM2 TX	SERCOM2 TX Trigger
0x08	SERCOM3 RX	SERCOM3 RX Trigger
0x09	SERCOM3 TX	SERCOM3 TX Trigger
0x0A	SERCOM4 RX	SERCOM4 RX Trigger
0x0B	SERCOM4 TX	SERCOM4 TX Trigger
0c0C	SERCOM5 RX	SERCOM5 RX Trigger
0x0D	SERCOM5 TX	SERCOM5 TX Trigger
0x0E	TCC0 OVF	TCC0 Overflow Trigger
0x0F	TCC0 MC0	TCC0 Match/Compare 0 Trigger
0x10	TCC0 MC1	TCC0 Match/Compare 1 Trigger
0x11	TCC0 MC2	TCC0 Match/Compare 2 Trigger
0x12	TCC0 MC3	TCC0 Match/Compare 3 Trigger
0x13	TC0 OVF	TC0 Overflow Trigger
0x14	TC0 MC0	TC0 Match/Compare 0 Trigger
0x15	TC0 MC1	TC0 Match/Compare 1 Trigger
0x16	TC1 OVF	TC1 Overflow Trigger
0x17	TC1 MC0	TC1 Match/Compare 0 Trigger
0x18	TC1 MC1	TC1 Match/Compare 1 Trigger
0x19	TC2 OVF	TC2 Overflow Trigger
0x1A	TC2 MC0	TC2 Match/Compare 0 Trigger
0x1B	TC2 MC1	TC2 Match/Compare 1 Trigger
0x1C	TC3 OVF	TC3 Overflow Trigger
0x1D	TC3 MC0	TC3 Match/Compare 0 Trigger
0x1E	TC3 MC1	TC3 Match/Compare 1 Trigger
0x1F	ADC RESRDY	ADC Result Ready Trigger
0x20	SLCD DMU	SLCD Display Memory Update Trigger
0x21	SLCD ACMDRDY	SLCD Automated Character Mapping Data Ready Trigger
0x22	SLCD ABMDRDY	SLCD Automated Bit Mapping Data Ready Trigger
0x23	AES WR	AES Write Trigger
0x24	AES RD	AES Read Trigger
0x25	PTC EOC	PTC End of Conversion Trigger
0x26	PTC SEQ	PTC Sequence Trigger
0x27	PTC WCOMP	PTC Window Comparator Trigger

Bits 6:5 – LVL[1:0]: Channel Arbitration Level
These bits define the arbitration level used for the DMA channel, where a high level has priority over a low level. For further details on arbitration schemes, refer to Arbitration.

These bits are not enable-protected.

TRIGACT[1:0]	Name	Description
0x0	LVL0	Channel Priority Level 0
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3

Bit 4 – EVOE: Channel Event Output Enable
This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the descriptor Event Output Selection (BTCTRL.EVOSEL).

This bit is available only for the least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 3 – EVIE: Channel Event Input Enable
This bit is available only for the least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event action will not be executed on any incoming event.
1	Channel event action will be executed on any incoming event.

Bits 2:0 – EVACT[2:0]: Event Input Action
These bits define the event input action, as shown below. The action is executed only if the corresponding EVIE bit in CHCTRLB register of the channel is set.

These bits are available only for the least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

EVACT[2:0]	Name	Description
0x0	NOACT	No action
0x1	TRIG	Normal Transfer and Conditional Transfer on Strobe trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action
0x7	-	Reserved

26.8.20 Channel Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Set (CHINTENSET) register.

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHINTENCLR
Offset: 0x4C [ID-00001ece]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP: Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL: Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR: Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

26.8.21 Channel Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Clear (CHINTENCLR) register.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHINTENSET
Offset: 0x4D [ID-00001ece]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP: Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL: Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR: Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

26.8.22 Channel Interrupt Flag Status and Clear

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHINTFLAG
Offset: 0x4E [ID-00001ece]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP: Channel Suspend
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.

For details on available software commands, refer to CHCTRLB.CMD.

For details on available event input actions, refer to CHCTRLB.EVACT.

For details on available block actions, refer to BTCTRL.BLOCKACT.

Bit 1 – TCMPL: Channel Transfer Complete
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR: Channel Transfer Error
This flag is cleared by writing a '1' to it.

This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

26.8.23 Channel Status

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHSTATUS
Offset: 0x4F [ID-00001ece]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						FERR	BUSY	PEND
Access						R	R	R
Reset						0	0	0

Bit 2 – FERR: Channel Fetch Error
This bit is cleared when a software resume command is executed.

This bit is set when an invalid descriptor is fetched.

Bit 1 – BUSY: Channel Busy
This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.

This bit is set when the DMA channel starts a DMA transfer.

Bit 0 – PEND: Channel Pending
This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.

This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

26.9 Register Summary - SRAM

Offset	Name	Bit Pos.								
0x00	BTCTRL	7:0				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
0x01		15:8	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
0x02	BTCNT	7:0	BTCNT[7:0]							
0x03		15:8	BTCNT[15:8]							
0x04	SRCADDR	7:0	SRCADDR[7:0]							
0x05		15:8	SRCADDR[15:8]							
0x06		23:16	SRCADDR[23:16]							
0x07		31:24	SRCADDR[31:24]							
0x08	DSTADDR	7:0	DSTADDR[7:0]							
0x09		15:8	DSTADDR[15:8]							
0x0A		23:16	DSTADDR[23:16]							
0x0B		31:24	DSTADDR[31:24]							
0x0C	DESCADDR	7:0	DESCADDR[7:0]							
0x0D		15:8	DESCADDR[15:8]							
0x0E		23:16	DESCADDR[23:16]							
0x0F		31:24	DESCADDR[31:24]							

26.10 Register Description - SRAM

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

26.10.1 Block Transfer Control

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: BTCTRL
Offset: 0x00 [ID-00001ece]
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access								
Reset								

Bits 15:13 – STEPSIZE[2:0]: Address Increment Step Size
These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (Beat size in byte) * 1
0x1	X2	Next ADDR = ADDR + (Beat size in byte) * 2
0x2	X4	Next ADDR = ADDR + (Beat size in byte) * 4
0x3	X8	Next ADDR = ADDR + (Beat size in byte) * 8
0x4	X16	Next ADDR = ADDR + (Beat size in byte) * 16
0x5	X32	Next ADDR = ADDR + (Beat size in byte) * 32
0x6	X64	Next ADDR = ADDR + (Beat size in byte) * 64
0x7	X128	Next ADDR = ADDR + (Beat size in byte) * 128

Bit 12 – STEPSEL: Step Selection
This bit selects if source or destination addresses are using the step size settings.

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC: Destination Address Increment Enable
Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled.
1	The Destination Address Increment is enabled.

Bit 10 – SRCINC: Source Address Increment Enable
Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Source Address Increment is disabled.
1	The Source Address Increment is enabled.

Bits 9:8 – BEATSIZE[1:0]: Beat Size
These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
other		Reserved

Bits 4:3 – BLOCKACT[1:0]: Block Action
These bits define what actions the DMAC should take after a block transfer has completed.

BLOCKACT[1:0]	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0]: Event Output Selection
These bits define the event output selection.

EVOSEL[1:0]	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2		Reserved
0x3	BEAT	Event strobe when beat transfer complete

Bit 0 – VALID: Descriptor Valid
Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.

The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

Value	Description
0	The descriptor is not valid.
1	The descriptor is valid.

26.10.2 Block Transfer Count

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: BTCNT
Offset: 0x02 [ID-00001ece]
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
	BTCNT[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	BTCNT[7:0]							
Access								
Reset								

Bits 15:0 – BTCNT[15:0]: Block Transfer Count
This bit group holds the 16-bit block transfer count.

During a transfer, the internal counter value is decremented by one after each beat transfer. The internal counter is written to the corresponding write-back memory section for the DMA channel when the DMA channel loses priority, is suspended or gets disabled. The DMA channel can be disabled by a complete transfer, a transfer error or by software.

26.10.3 Block Transfer Source Address

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: SRCADDR
Offset: 0x04 [ID-00001ece]
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access								
Reset								

Bits 31:0 – SRCADDR[31:0]: Transfer Source Address
This bit group holds the source address corresponding to the last beat transfer address in the block transfer.

26.10.4 Block Transfer Destination Address

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: DSTADDR
Offset: 0x08 [ID-00001ece]
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	DSTADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DSTADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DSTADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DSTADDR[7:0]							
Access								
Reset								

Bits 31:0 – DSTADDR[31:0]: Transfer Destination Address
This bit group holds the destination address corresponding to the last beat transfer address in the block transfer.

26.10.5 Next Descriptor Address

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: DESCADDR
Offset: 0x0C [ID-00001ece]
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	DESCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DESCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DESCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DESCADDR[7:0]							
Access								
Reset								

Bits 31:0 – DESCADDR[31:0]: Next Descriptor Address

This bit group holds the SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

27. EIC – External Interrupt Controller

27.1 Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, falling, or both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Each external pin can also be configured to be asynchronous in order to wake up the device from sleep modes where all clocks have been disabled. External pins can also generate an event.

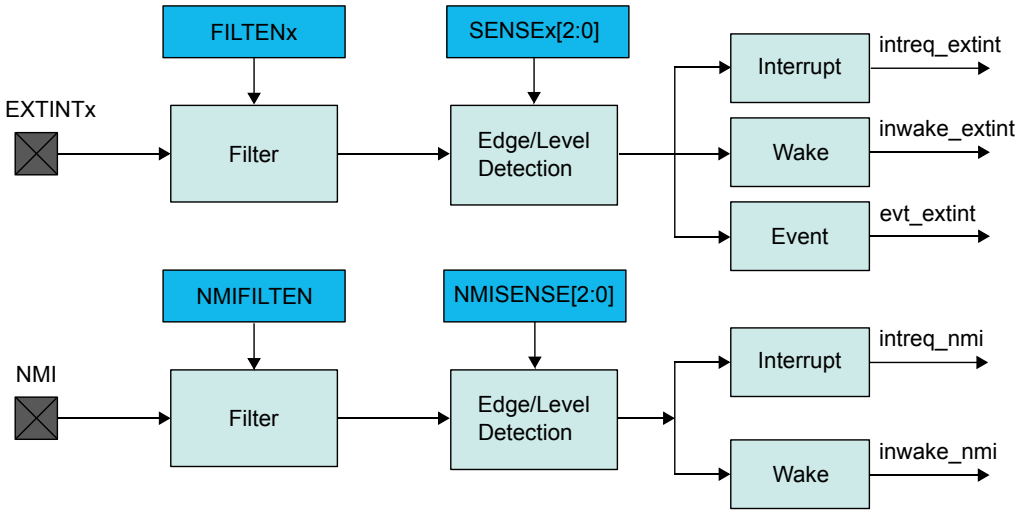
A separate non-maskable interrupt (NMI) is also supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other interrupt mode.

27.2 Features

- Up to 16 external pins, plus one non-maskable pin
- Dedicated, individually maskable interrupt for each pin
- Interrupt on rising, falling, or both edges
- Synchronous or asynchronous edge detection mode
- Interrupt on high or low levels
- Asynchronous interrupts for sleep modes without clock
- Filtering of external pins
- Event generation

27.3 Block Diagram

Figure 27-1. EIC Block Diagram



27.4 Signal Description

Signal Name	Type	Description
EXTINT[15..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

One signal can be mapped on several pins.

27.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

27.5.1 I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

Related Links

PORT - I/O Pin Controller

27.5.2 Power Management

All interrupts are available in all sleep modes, but the EIC can be configured to automatically mask some interrupts in order to prevent device wake-up.

The EIC will continue to operate in any sleep mode where the selected source clock is running. The EIC's interrupts can be used to wake up the device from sleep modes. Events connected to the Event System can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

27.5.3 Clocks

The EIC bus clock (CLK_EIC_APB) can be enabled and disabled by the Main Clock Controller, the default state of CLK_EIC_APB can be found in the Peripheral Clock Masking section.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK_EIC, for wider frequency selection) or a Ultra Low Power 32KHz clock (CLK_ULP32K, for highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral:

GCLK_EIC is configured and enabled in the Generic Clock Controller.

CLK_ULP32K is provided by the internal ultra-low-power (OSCULP32K) oscillator in the OSC32KCTRL module.

Both GCLK_EIC and CLK_ULP32K are asynchronous to the user interface clock (CLK_EIC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

MCLK – Main Clock

Peripheral Clock Masking

GCLK - Generic Clock Controller

OSC32KCTRL – 32KHz Oscillators Controller

27.5.4 DMA

Not applicable.

27.5.5 Interrupts

There are two interrupt request lines, one for the external interrupts (EXTINT) and one for non-maskable interrupt (NMI).

The EXTINT interrupt request line is connected to the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is also connected to the interrupt controller, but does not require the interrupt to be configured.

Related Links

Nested Vector Interrupt Controller

27.5.6 Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

Related Links

EVSYS – Event System

27.5.7 Debug Operation

When the CPU is halted in debug mode, the EIC continues normal operation. If the EIC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

27.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Non-Maskable Interrupt Flag Status and Clear register (NMIFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

27.5.9 Analog Connections

Not applicable.

27.6 Functional Description

27.6.1 Principle of Operation

The EIC detects edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K.

27.6.2 Basic Operation

27.6.2.1 Initialization

The EIC must be initialized in the following order:

1. Enable CLK_EIC_APB
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (NMICTRL)
3. When the NMI is used or synchronous edge detection or filtering are required, enable GCLK_EIC or CLK_ULP32K.
   GCLK_EIC is used when a frequency higher than 32KHz is required for filtering, CLK_ULP32K is recommended when power consumption is the priority. For CLK_ULP32K write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL). Optionally, enable the asynchronous mode.
4. Configure the EIC input sense and filtering by writing the Configuration n register (CONFIG0, CONFIG1).
5. Enable the EIC.

The following bits are enable-protected, meaning that it can only be written when the EIC is disabled (CTRLA.ENABLE=0):

- Clock Selection bit in Control A register (CTRLA.CKSEL)

The following registers are enable-protected:

- Event Control register (EVCTRL)
- Configuration n register (CONFIG0, CONFIG1...)
- External Interrupt Asynchronous Mode register (ASYNCH)

Enable-protected bits in the CTRLA register can be written at the same time when setting CTRLA.ENABLE to '1', but not at the same time as CTRLA.ENABLE is being cleared.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

27.6.2.2 Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

Refer to the CTRLA register description for details.

27.6.3 External Pin Processing

Each external pin can be configured to generate an interrupt/event on edge detection (rising, falling or both edges) or level detection (high or low). The sense of external interrupt pins is configured by writing the Input Sense x bits in the Config n register (CONFIGn.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met.

When the interrupt flag has been cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met. In level-sensitive mode, when interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or CLK_ULP32K. Filtering is enabled if bit Filter Enable x in the Configuration n register (CONFIGn.FILTENx) is written to '1'. The majority vote filter samples the external pin three times with GCLK_EIC or CLK_ULP32K and outputs the value when two or more samples are equal.

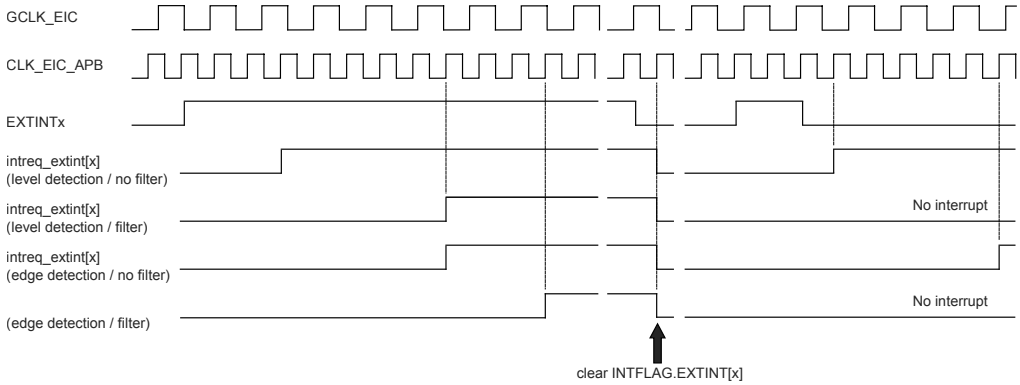
Table 27-1. Majority Vote Filter

Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Asynchronuous detection does not require GCLK_EIC or CLK_ULP32K, but interrupt and events can still be generated.

If filtering or edge detection is enabled, the EIC automatically requests GCLK_EIC or CLK_ULP32K to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (CTRLA.CKSEL). GCLK_EIC must be enabled in the GCLK module.

Figure 27-2. Interrupt Detections



The detection delay depends on the detection mode.

Table 27-2. Interrupt Latency

Detection mode	Latency (worst case)
Level without filter	Five CLK_EIC_APB periods
Level with filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge without filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge with filter	Six GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods

Related Links

GCLK - Generic Clock Controller

27.6.4 Additional Features

27.6.4.1 Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC is not required to be enabled.

When an NMI is detected, the non-maskable interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

27.6.4.2 Asynchronous Edge Detection Mode

The EXTINT edge detection can be operated synchronously or asynchronously, selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In *Synchronous Edge Detection Mode*, the external interrupt (EXTINT) or the non-maskable interrupt (NMI) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) or Non-Maskable Interrupt flag (NMIFLAG.NMI) is set when the last sampled state of the pin differs from the previously sampled state. In this mode, the EIC clock is required.

The Synchronous Edge Detection Mode can be used in Idle sleep mode.

In *Asynchronous Edge Detection Mode*, the external interrupt (EXTINT) pins or the non-maskable interrupt (NMI) pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in all sleep modes.

27.6.5 DMA Operation

Not applicable.

27.6.6 Interrupts

The EIC has the following interrupt sources:

- External interrupt pins (EXTINTx). See Basic Operation.
- Non-maskable interrupt pin (NMI). See Additional Features.

Each interrupt source has an associated interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an interrupt condition occurs (NMIFLAG for NMI). Each interrupt, except NMI, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC is reset. See the INTFLAG register for details on how to clear interrupt flags. The EIC has one common

interrupt request line for all the interrupt sources, and one interrupt request line for the NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Processor and Architecture

27.6.7 Events

The EIC can generate the following output events:

- External event from pin (EXTINTx).

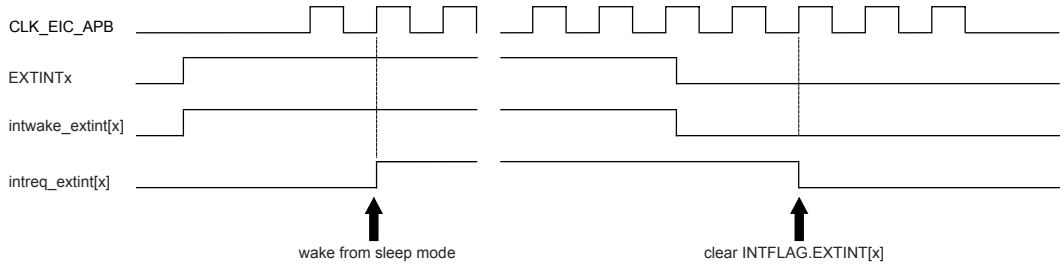
Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to *Event System* for details on configuring the Event System.

When the condition on pin EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

27.6.8 Sleep Mode Operation

In sleep modes, an EXTINTx pin can wake up the device if the corresponding condition matches the configuration in CONFIG0, CONFIG1 register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.

Figure 27-3. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



27.6.9 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

27.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0				CKSEL			ENABLE	SWRST
0x01	NMICTRL	7:0				ASYNCH	NMIFILTEN	NMISENSE[2:0]		
0x02	NMIFLAG	7:0								NMI
0x03		15:8								
0x04	SYNCBUSY	7:0							ENABLE	SWRST
0x05		15:8								
0x06		23:16								
0x07		31:24								
0x08	EVCTRL	7:0	EXTINTEO[7:0]							
0x09		15:8	EXTINTEO[15:8]							
0x0A		23:16								
0x0B		31:24								
0x0C	INTENCLR	7:0	EXTINT[7:0]							
0x0D		15:8	EXTINT[15:8]							
0x0E		23:16								
0x0F		31:24								
0x10	INTENSET	7:0	EXTINT[7:0]							
0x11		15:8	EXTINT[15:8]							
0x12		23:16								
0x13		31:24								
0x14	INTFLAG	7:0	EXTINT[7:0]							
0x15		15:8	EXTINT[15:8]							
0x16		23:16								
0x17		31:24								
0x18	ASYNCH	7:0	ASYNCH[7:0]							
0x19		15:8	ASYNCH[15:8]							
0x1A		23:16								
0x1B		31:24								
0x1C	CONFIG0	7:0	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x1D		15:8	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x1E		23:16	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x1F		31:24	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x20	CONFIG1	7:0	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x21		15:8	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x22		23:16	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
0x23		31:24	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		

27.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

27.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000c8b]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				R/W			R/W	W
Reset				0			0	0

Bit 4 – CKSEL: Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32KHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).

This bit is not Write-Synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by CLK_ULP32K.

Bit 1 – ENABLE: Enable
Due to synchronization there is a delay between writing to CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register will be set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not Enable-Protected.

Value	Description
0	The EIC is disabled.
1	The EIC is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the EIC to their initial state, and the EIC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

This bit is not Enable-Protected.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

27.8.2 Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01 [ID-00000c8b]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				ASYNCH	NMIFILTEN	NMISENSE[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – ASYNCH: Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

Value	Description
0	The NMI edge detection is synchronously operated.
1	The NMI edge detection is asynchronously operated.

Bit 3 – NMIFILTEN: Non-Maskable Interrupt Filter Enable

Value	Description
0	NMI filter is disabled.
1	NMI filter is enabled.

Bits 2:0 – NMISENSE[2:0]: Non-Maskable Interrupt Sense
These bits define on which edge or level the NMI triggers.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	-	Reserved

27.8.3 Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x02 [ID-00000c8b]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NMI
Access								R/W
Reset								0

Bit 0 – NMI: Non-Maskable Interrupt
This flag is cleared by writing a '1' to it.

This flag is set when the NMI pin matches the NMI sense configuration, and will generate an interrupt request.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the non-maskable interrupt flag.

27.8.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04 [ID-00000c8b]
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

27.8.5 Event Control

Name: EVCTRL
Offset: 0x08 [ID-00000c8b]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINTEO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINTEO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINTEO[15:0]: External Interrupt x Event Output
These bits enable the event associated with the EXTINTx pin.

Value	Description
0	Event from pin EXTINTx is disabled.
1	Event from pin EXTINTx is enabled and will be generated when EXTINTx pin matches the external interrupt sensing configuration.

27.8.6 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x0C [ID-00000c8b]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0]: External Interrupt x Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the External Interrupt x Enable bit, which disables the external interrupt.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

27.8.7 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x10 [ID-00000c8b]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0]: External Interrupt x Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the External Interrupt x Enable bit, which enables the external interrupt.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

27.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14 [ID-00000c8b]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0]: External Interrupt x
This flag is cleared by writing a '1' to it.

This flag is set when EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR/SET.EXTINT[x] is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the External Interrupt x flag.

27.8.9 External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18 [ID-00000c8b]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ASYNCH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ASYNCH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ASYNCH[15:0]: Asynchronous Edge Detection Mode

Value	Description
0	The EXTINT edge detection is synchronously operated.
1	The EXTINT edge detection is asynchronously operated.

27.8.10 Configuration n

Name: CONFIG0, CONFIG1
Offset: 0x1C + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FILTENx	SENSEx[2:0]			FILTENx	SENSEx[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 3,7,11,15,19,23,27,31 – FILTENx: Filter x Enable [x = 7..0]

Value	Description
0	Filter is disabled for EXTINT[n*8+1] input.
1	Filter is enabled for EXTINT[n*8+1] input.

Bits 0:2,4:6,8:10,12:14,16:18,20:22,24:26,28:30 – SENSEx: Input Sense x Configuration
These bits define on which edge or level the interrupt or event for EXTINT[n*8+x] will be generated.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	-	Reserved

28. NVMCTRL – Non-Volatile Memory Controller

28.1 Overview

Non-Volatile Memory (NVM) is a reprogrammable Flash memory that retains program and data storage even with power off. It embeds a main array and a separate smaller array intended for EEPROM emulation (RWWEE) that can be programmed while reading the main array. The NVM Controller (NVMCTRL) connects to the AHB and APB bus interfaces for system access to the NVM block. The AHB interface is used for reads and writes to the NVM block, while the APB interface is used for commands and configuration.

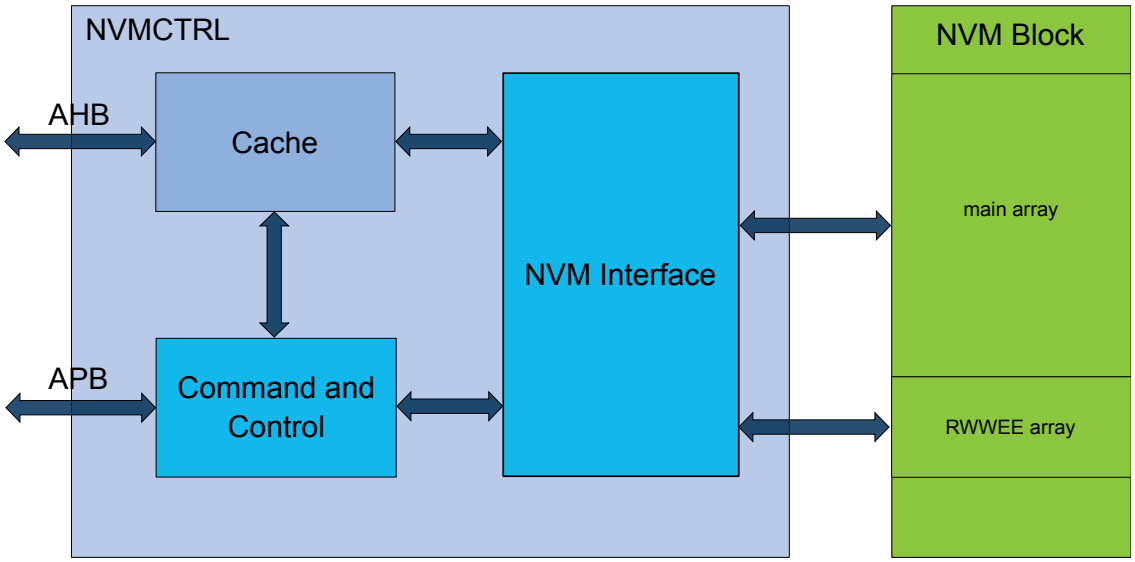
28.2 Features

- 32-bit AHB interface for reads and writes
- Read While Write EEPROM emulation area
- All NVM sections are memory mapped to the AHB, including calibration and system configuration
- 32-bit APB interface for commands and control
- Programmable wait states for read optimization
- 16 regions can be individually protected or unprotected
- Additional protection for boot loader
- Supports device protection through a security bit
- Interface to Power Manager for power-down of Flash blocks in sleep modes
- Can optionally wake up on exit from sleep or on first access
- Direct-mapped cache

Note: A register with property "Enable-Protected" may contain bits that are *not* enable-protected.

28.3 Block Diagram

Figure 28-1. Block Diagram



28.4 Signal Description

Not applicable.

28.5 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

28.5.1 Power Management

The NVMCTRL will continue to operate in any sleep mode where the selected source clock is running. The NVMCTRL interrupts can be used to wake up the device from sleep modes.

The Power Manager will automatically put the NVM block into a low-power state when entering sleep mode. This is based on the Control B register (CTRLB) SLEEPPRM bit setting. Refer to the CTRLB.SLEEPPRM register description for more details.

Related Links

PM – Power Manager

28.5.2 Clocks

Two synchronous clocks are used by the NVMCTRL. One is provided by the AHB bus (CLK_NVMCTRL_AHB) and the other is provided by the APB bus (CLK_NVMCTRL_APB). For higher system frequencies, a programmable number of wait states can be used to optimize performance. When changing the AHB bus frequency, the user must ensure that the NVM Controller is configured with the proper number of wait states. Refer to the Electrical Characteristics for the exact number of wait states to be used for a particular frequency range.

Related Links

Electrical Characteristics

28.5.3 Interrupts

The NVM Controller interrupt request line is connected to the interrupt controller. Using the NVMCTRL interrupt requires the interrupt controller to be programmed first.

28.5.4 Debug Operation

When an external debugger forces the CPU into debug mode, the peripheral continues normal operation.

Access to the NVM block can be protected by the security bit. In this case, the NVM block will not be accessible. See the section on the NVMCTRL Security Bit for details.

28.5.5 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Related Links

PAC - Peripheral Access Controller

28.5.6 Analog Connections

Not applicable.

28.6 Functional Description

28.6.1 Principle of Operation

The NVM Controller is a slave on the AHB and APB buses. It responds to commands, read requests and write requests, based on user configuration.

28.6.1.1 Initialization

After power up, the NVM Controller goes through a power-up sequence. During this time, access to the NVM Controller from the AHB bus is halted. Upon power-up completion, the NVM Controller is operational without any need for user configuration.

28.6.2 Memory Organization

Refer to the Physical Memory Map for memory sizes and addresses for each device.

The NVM is organized into rows, where each row contains four pages, as shown in the NVM Row Organization figure. The NVM has a row-erase granularity, while the write granularity is by page. In other words, a single row erase will erase all four pages in the row, while four write operations are used to write the complete row.

Figure 28-2. NVM Row Organization

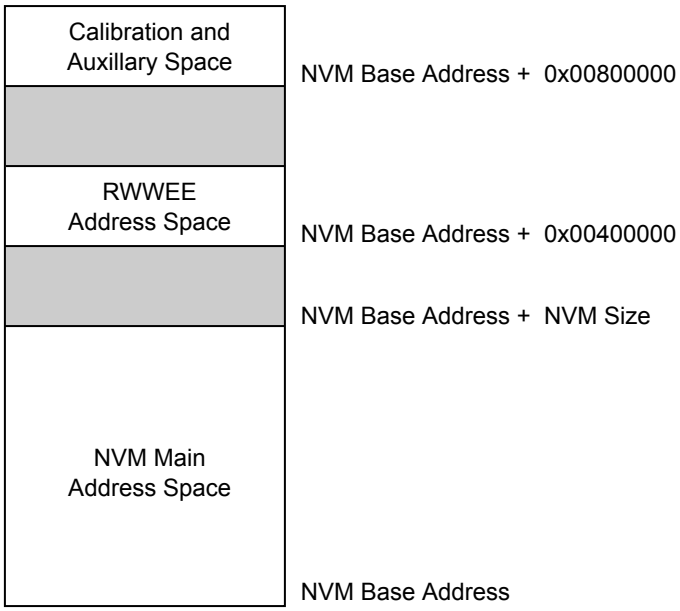
Row n	Page (n*4) + 3	Page (n*4) + 2	Page (n*4) + 1	Page (n*4) + 0

The NVM block contains a calibration and auxiliary space plus a dedicated EEPROM emulation space that are memory mapped. Refer to the NVM Organization figure below for details.

The calibration and auxiliary space contains factory calibration and system configuration information. These spaces can be read from the AHB bus in the same way as the main NVM main address space.

In addition, a boot loader section can be allocated at the beginning of the main array, and an EEPROM section can be allocated at the end of the NVM main address space.

Figure 28-3. NVM Memory Organization

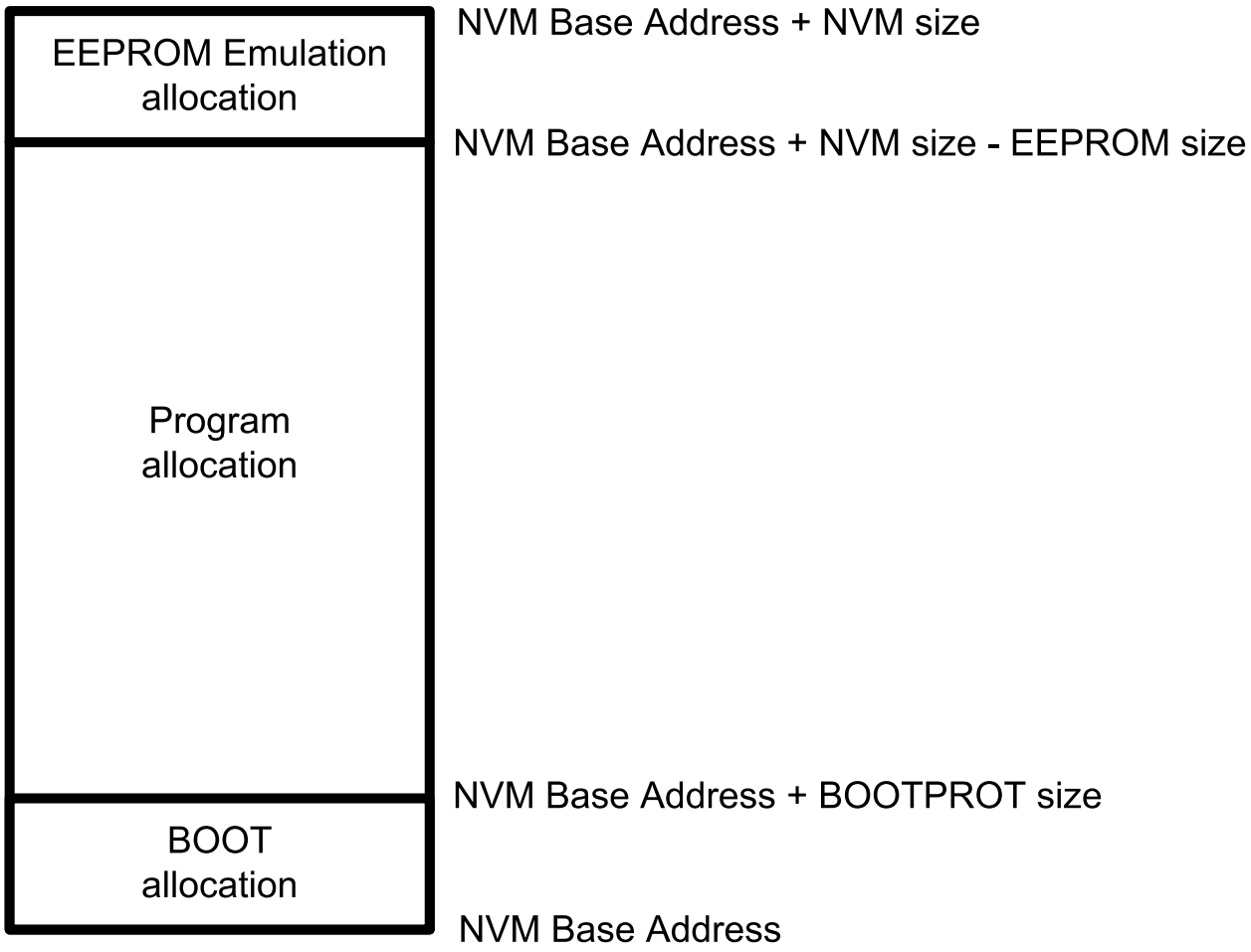


The lower rows in the NVM main address space can be allocated as a boot loader section by using the BOOTPROT fuses, and the upper rows can be allocated to EEPROM, as shown in the figure below.

The boot loader section is protected by the lock bit(s) corresponding to this address space and by the BOOTPROT[2:0] fuse. The EEPROM rows can be written regardless of the region lock status.

The number of rows protected by BOOTPROT is given in Boot Loader Size, the number of rows allocated to the EEPROM are given in EEPROM Size.

Figure 28-4. EEPROM and Boot Loader Allocation



Related Links

Physical Memory Map

28.6.3 Region Lock Bits

The NVM block is grouped into 16 equally sized regions. The region size is dependent on the Flash memory size, and is given in the table below. Each region has a dedicated lock bit preventing writing and erasing pages in the region. After production, all regions will be unlocked.

Table 28-1. Region Size

Memory Size [KB]	Region Size [KB]
256	16
128	8
64	4
32	2

To lock or unlock a region, the Lock Region and Unlock Region commands are provided. Writing one of these commands will temporarily lock/unlock the region containing the address loaded in the ADDR register. ADDR can be written by software, or the automatically loaded value from a write operation can be used. The new setting will stay in effect until the next Reset, or until the setting is changed again using

the Lock and Unlock commands. The current status of the lock can be determined by reading the LOCK register.

To change the default lock/unlock setting for a region, the user configuration section of the auxiliary space must be written using the Write Auxiliary Page command. Writing to the auxiliary space will take effect after the next Reset. Therefore, a boot of the device is needed for changes in the lock/unlock setting to take effect. Refer to the Physical Memory Map for calibration and auxiliary space address mapping.

Related Links

Physical Memory Map

28.6.4 Command and Data Interface

The NVM Controller is addressable from the APB bus, while the NVM main address space is addressable from the AHB bus. Read and automatic page write operations are performed by addressing the NVM main address space or the RWWEE address space directly, while other operations such as manual page writes and row erases must be performed by issuing commands through the NVM Controller.

To issue a command, the CTRLA.CMD bits must be written along with the CTRLA.CMDEX value. When a command is issued, INTFLAG.READY will be cleared until the command has completed. Any commands written while INTFLAG.READY is low will be ignored.

The CTRLB register must be used to control the power reduction mode, read wait states, and the write mode.

28.6.4.1 NVM Read

Reading from the NVM main address space is performed via the AHB bus by addressing the NVM main address space or auxiliary address space directly. Read data is available after the configured number of read wait states (CTRLB.RWS) set in the NVM Controller.

The number of cycles data are delayed to the AHB bus is determined by the read wait states. Examples of using zero and one wait states are shown in Figure Read Wait State Examples below.

Reading the NVM main address space while a programming or erase operation is ongoing on the NVM main array results in an AHB bus stall until the end of the operation. Reading the NVM main array does not stall the bus when the RWWEE array is being programmed or erased.

28.6.4.2 RWWEE Read

Reading from the RWW EEPROM address space is performed via the AHB bus by addressing the RWWEE address space directly.

Read timings are similar to regular NVM read timings when access size is Byte or half-Word. The AHB data phase is twice as long in case of full-Word-size access.

It is not possible to read the RWWEE area while the NVM main array is being written or erased, whereas the RWWEE area can be written or erased while the main array is being read.

The RWWEE address space is not cached, therefore it is recommended to limit access to this area for performance and power consumption considerations.

28.6.4.3 NVM Write

The NVM Controller requires that an erase must be done before programming. The entire NVM main address space and the RWWEE address space can be erased by a debugger Chip Erase command. Alternatively, rows can be individually erased by the Erase Row command or the RWWEE Erase Row command to erase the NVM main address space or the RWWEE address space, respectively.

After programming the NVM main array, the region that the page resides in can be locked to prevent spurious write or erase sequences. Locking is performed on a per-region basis, and so, locking a region will lock all pages inside the region.

Data to be written to the NVM block are first written to and stored in an internal buffer called the *page buffer*. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 16 or 32 bits. 8-bit writes to the page buffer are not allowed and will cause a system exception.

Both the NVM main array and the RWWEE array share the same page buffer. Writing to the NVM block via the AHB bus is performed by a load operation to the page buffer. For each AHB bus write, the address is stored in the ADDR register. After the page buffer has been loaded with the required number of bytes, the page can be written to the NVM main array or the RWWEE array by setting CTRLA.CMD to 'Write Page' or 'RWWEE Write Page', respectively, and setting the key value to CMDEX. The LOAD bit in the STATUS register indicates whether the page buffer has been loaded or not. Before writing the page to memory, the accessed row must be erased.

Automatic page writes are enabled by writing the manual write bit to zero (CTRLB.MANW=0). This will trigger a write operation to the page addressed by ADDR when the last location of the page is written.

Because the address is automatically stored in ADDR during the I/O bus write operation, the last given address will be present in the ADDR register. There is no need to load the ADDR register manually, unless a different page in memory is to be written.

Procedure for Manual Page Writes (CTRLB.MANW=1)
The row to be written to must be erased before the write command is given.

- Write to the page buffer by addressing the NVM main address space directly
- Write the page buffer to memory: CTRL.CMD='Write Page' and CMDEX
- The READY bit in the INTFLAG register will be low while programming is in progress, and access through the AHB will be stalled

Procedure for Automatic Page Writes (CTRLB.MANW=0)
The row to be written to must be erased before the last write to the page buffer is performed.

Note that partially written pages must be written with a manual write.

- Write to the page buffer by addressing the NVM main address space directly.
  When the last location in the page buffer is written, the page is automatically written to NVM main address space.
- INTFLAG.READY will be zero while programming is in progress and access through the AHB will be stalled.

28.6.4.4 Page Buffer Clear

The page buffer is automatically set to all '1' after a page write is performed. If a partial page has been written and it is desired to clear the contents of the page buffer, the Page Buffer Clear command can be used.

28.6.4.5 Erase Row

Before a page can be written, the row containing that page must be erased. The Erase Row command can be used to erase the desired row in the NVM main address space. The RWWEE Erase Row can be used to erase the desired row in the RWWEE array. Erasing the row sets all bits to '1'. If the row resides in a region that is locked, the erase will not be performed and the Lock Error bit in the Status register (STATUS.LOCKE) will be set.

Procedure for Erase Row

- Write the address of the row to erase to ADDR. Any address within the row can be used.

- Issue an Erase Row command.

Note: The NVM Address bit field in the Address register (ADDR.ADDR) uses 16-bit addressing.

28.6.4.6 Lock and Unlock Region

These commands are used to lock and unlock regions as detailed in section Region Lock Bits.

28.6.4.7 Set and Clear Power Reduction Mode

The NVM Controller and block can be taken in and out of power reduction mode through the Set and Clear Power Reduction Mode commands. When the NVM Controller and block are in power reduction mode, the Power Reduction Mode bit in the Status register (STATUS.PRM) is set.

28.6.5 NVM User Configuration

The NVM user configuration resides in the auxiliary space. Refer to the Physical Memory Map of the device for calibration and auxiliary space address mapping.

The bootloader resides in the main array starting at offset zero. The allocated boot loader section is write-protected.

Table 28-2. Boot Loader Size

BOOTPROT [2:0]	Rows Protected by BOOTPROT	Boot Loader Size in Bytes
0x7(1)	None	0
0x6	2	512
0x5	4	1024
0x4	8	2048
0x3	16	4096
0x2	32	8192
0x1	64	16384
0x0	128	32768

Note: 1) Default value is 0x7.

The EEPROM[2:0] bits indicate the EEPROM size, see the table below. The EEPROM resides in the upper rows of the NVM main address space and is writable, regardless of the region lock status.

Table 28-3. EEPROM Size

EEPROM[2:0]	Rows Allocated to EEPROM	EEPROM Size in Bytes
7	None	0
6	1	256
5	2	512
4	4	1024
3	8	2048
2	16	4096
1	32	8192
0	64	16384

Related Links

Physical Memory Map

28.6.6 Security Bit

The security bit allows the entire chip to be locked from external access for code security. The security bit can be written by a dedicated command, Set Security Bit (SSB). Once set, the only way to clear the security bit is through a debugger Chip Erase command. After issuing the SSB command, the PROGE error bit can be checked.

In order to increase the security level it is recommended to enable the internal BOD33 when the security bit is set.

Related Links

DSU - Device Service Unit

28.6.7 Cache

The NVM Controller cache reduces the device power consumption and improves system performance when wait states are required. Only the NVM main array address space is cached. It is a direct-mapped cache that implements 8 lines of 64 bits (i.e., 64 Bytes). NVM Controller cache can be enabled by writing a '0' to the Cache Disable bit in the Control B register (CTRLB.CACHEDIS).

The cache can be configured to three different modes using the Read Mode bit group in the Control B register (CTRLB.READMODE).

The INVALL command can be issued using the Command bits in the Control A register to invalidate all cache lines (CTRLA.CMD=INVALL). Commands affecting NVM content automatically invalidate cache lines.

28.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		CMD[6:0]						
0x01		15:8	CMDEX[7:0]							
0x02 ... 0x03	Reserved									
0x04	CTRLB	7:0				RWS[3:0]				
0x05		15:8							SLEEPPRM[1:0]	
0x06		23:16						CACHEDIS	READMODE[1:0]	
0x07		31:24								
0x08	PARAM	7:0	NVMP[7:0]							
0x09		15:8	NVMP[15:8]							
0x0A		23:16	RWWEEP[3:0]					PSZ[2:0]		
0x0B		31:24	RWWEEP[11:4]							
0x0C	INTENCLR	7:0							ERROR	READY
0x0D ... 0x0F	Reserved									
0x10	INTENSET	7:0							ERROR	READY
0x11 ... 0x13	Reserved									
0x14	INTFLAG	7:0							ERROR	READY
0x15 ... 0x17	Reserved									
0x18	STATUS	7:0				NVME	LOCKE	PROGE	LOAD	PRM
0x19		15:8								
0x1A ... 0x1B	Reserved									
0x1C	ADDR	7:0	ADDR[7:0]							
0x1D		15:8	ADDR[15:8]							
0x1E		23:16			ADDR[21:16]					
0x1F		31:24								
0x20	LOCK	7:0	LOCK[7:0]							
0x21		15:8	LOCK[15:8]							

28.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

28.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000b2c]
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	CMDEX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CMD[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 15:8 – CMDEX[7:0]: Command Execution
When this bit group is written to the key value 0xA5, the command written to CMD will be executed. If a value different from the key value is tried, the write will not be performed and the Programming Error bit in the Status register (STATUS.PROGE) will be set. PROGE is also set if a previously written command is not completed yet.

The key value must be written at the same time as CMD. If a command is issued through the APB bus on the same cycle as an AHB bus access, the AHB bus access will be given priority. The command will then be executed when the NVM block and the AHB bus are idle.

INTFLAG.READY must be '1' when the command is issued.

Bit 0 of the CMDEX bit group will read back as '1' until the command is issued.

Note: The NVM Address bit field in the Address register (ADDR.ADDR) uses 16-bit addressing.

Bits 6:0 – CMD[6:0]: Command
These bits define the command to be executed when the CMDEX key is written.

CMD[6:0]	Group Configuration	Description
0x00-0x01	-	Reserved
0x02	ER	Erase Row - Erases the row addressed by the ADDR register in the NVM main array.
0x03	-	Reserved
0x04	WP	Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register.
0x05	EAR	
0x06	WAP	

CMD[6:0]	Group Configuration	Description
0x07-0x0E	-	Reserved
0x0F	WL	Write Lockbits- write the LOCK register
0x1A-0x19	-	Reserved
0x1A	RWWEEER	RWWEE Erase Row - Erases the row addressed by the ADDR register in the RWWEE array.
0x1B	-	Reserved
0x1C	RWWEEWP	RWWEE Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register in the RWWEE array.
0x1D-0x3F	-	Reserved
0x40	LR	Lock Region - Locks the region containing the address location in the ADDR register.
0x41	UR	Unlock Region - Unlocks the region containing the address location in the ADDR register.
0x42	SPRM	Sets the Power Reduction Mode.
0x43	CPRM	Clears the Power Reduction Mode.
0x44	PBC	Page Buffer Clear - Clears the page buffer.
0x46	INVALL	Invalidates all cache lines.
0x47	LDR	Lock Data Region - Locks the data region containing the address location in the ADDR register. When the Security Extension is enabled, only secure access can lock secure regions.
0x48	UDR	Unlock Data Region - Unlocks the data region containing the address location in the ADDR register. When the Security Extension is enabled, only secure access can unlock secure regions.

28.8.2 Control B

Name: CTRLB
Offset: 0x04 [ID-00000b2c]
Reset: 0x00000080
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CACHEDIS	READMODE[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							SLEEPPRM[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
				RWS[3:0]				
Access				R/W	R/W	R/W	R/W	
Reset				0	0	0	0	

Bit 18 – CACHEDIS: Cache Disable
This bit is used to disable the cache.

Value	Description
0	The cache is enabled
1	The cache is disabled

Bits 17:16 – READMODE[1:0]: NVMCTRL Read Mode

Value	Name	Description
0x0	NO_MISS_PENALTY	The NVM Controller (cache system) does not insert wait states on a cache miss. Gives the best system performance.
0x1	LOW_POWER	Reduces power consumption of the cache system, but inserts a wait state each time there is a cache miss. This mode may not be relevant if CPU performance is required, as the application will be stalled and may lead to increased run time.
0x2	DETERMINISTIC	The cache system ensures that a cache hit or miss takes the same amount of time, determined by the number of programmed Flash wait states. This mode can be used for real-time applications that require deterministic execution timings.
0x3	Reserved	

Bits 9:8 – SLEEPPRM[1:0]: Power Reduction Mode during Sleep
Indicates the Power Reduction Mode during sleep.

Value	Name	Description
0x0	WAKEUPACCESS	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode upon first access.
0x1	WAKEUPINSTANT	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode when exiting sleep.

Value	Name	Description
0x2	Reserved	
0x3	DISABLED	Auto power reduction disabled.

Bits 4:1 – RWS[3:0]: NVM Read Wait States
These bits control the number of wait states for a read operation. '0' indicates zero wait states, '1' indicates one wait state, etc., up to 15 wait states.
This register is initialized to 0 wait states. Software can change this value based on the NVM access time and system frequency.

28.8.3 NVM Parameter

Name: PARAM
Offset: 0x08 [ID-00000b2c]
Reset: 0x000XXXXX
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RWWEEP[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RWWEEP[3:0]					PSZ[2:0]		
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		x	x	x

Bit	15	14	13	12	11	10	9	8
	NVMP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	NVMP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:20 – RWWEEP[11:0]: Read While Write EEPROM emulation area Pages
Indicates the number of pages in the RWW EEPROM emulation address space.

Bits 18:16 – PSZ[2:0]: Page Size
Indicates the page size. Not all devices of the device families will provide all the page sizes indicated in the table.

Value	Name	Description
0x0	8	8 bytes
0x1	16	16 bytes
0x2	32	32 bytes
0x3	64	64 bytes
0x4	128	128 bytes

Value	Name	Description
0x5	256	256 bytes
0x6	512	512 bytes
0x7	1024	1024 bytes

Bits 15:0 – NVMP[15:0]: NVM Pages
Indicates the number of pages in the NVM main address space.

28.8.4 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x0C [ID-00000b2c]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R/W
Reset							0	0

Bit 1 – ERROR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the ERROR interrupt enable.

This bit will read as the current value of the ERROR interrupt enable.

Bit 0 – READY: NVM Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the READY interrupt enable.

This bit will read as the current value of the READY interrupt enable.

28.8.5 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x10 [ID-00000b2c]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R/W
Reset							0	0

Bit 1 – ERROR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the ERROR interrupt enable.

This bit will read as the current value of the ERROR interrupt enable.

Bit 0 – READY: NVM Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the READY interrupt enable.

This bit will read as the current value of the READY interrupt enable.

28.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14 [ID-00000b2c]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R
Reset							0	0

Bit 1 – ERROR: Error
This flag is set on the occurrence of an NVME, LOCKE or PROGE error.

This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No errors have been received since the last clear.
1	At least one error has occurred since the last clear.

Bit 0 – READY: NVM Ready

Value	Description
0	The NVM controller is busy programming or erasing.
1	The NVM controller is ready to accept a new command.

28.8.7 Status

Name: STATUS
Offset: 0x18 [ID-00000b2c]
Reset: 0x0X00
Property: –

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				NVME	LOCKE	PROGE	LOAD	PRM
Access				R/W	R/W	R/W	R/W	R
Reset				0	0	0	0	0

Bit 4 – NVME: NVM Error
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No programming or erase errors have been received from the NVM controller since this bit was last cleared.
1	At least one error has been registered from the NVM Controller since this bit was last cleared.

Bit 3 – LOCKE: Lock Error Status
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No programming of any locked lock region has happened since this bit was last cleared.
1	Programming of at least one locked lock region has happened since this bit was last cleared.

Bit 2 – PROGE: Programming Error Status
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No invalid commands or bad keywords were written in the NVM Command register since this bit was last cleared.
1	An invalid command and/or a bad keyword was/were written in the NVM Command register since this bit was last cleared.

Bit 1 – LOAD: NVM Page Buffer Active Loading
This bit indicates that the NVM page buffer has been loaded with one or more words. Immediately after an NVM load has been performed, this flag is set. It remains set until a page write or a page buffer clear (PBCLR) command is given.

This bit can be cleared by writing a '1' to its bit location.

Bit 0 – PRM: Power Reduction Mode
This bit indicates the current NVM power reduction state. The NVM block can be set in power reduction mode in two ways: through the command interface or automatically when entering sleep with SLEEPPRM set accordingly.

PRM can be cleared in three ways: through AHB access to the NVM block, through the command interface (SPRM and CPRM) or when exiting sleep with SLEEPPRM set accordingly.

Value	Description
0	NVM is not in power reduction mode.
1	NVM is in power reduction mode.

28.8.8 Address

Name: ADDR
Offset: 0x1C [ID-00000b2c]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			ADDR[21:16]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 21:0 – ADDR[21:0]: NVM Address
ADDR drives the hardware (16-bit) address to the NVM when a command is executed using CMDEX. This register is also automatically updated when writing to the page buffer.

28.8.9 Lock Section

Name: LOCK
Offset: 0x20 [ID-00000b2c]
Reset: 0xXXXX
Property: –

Bit	15	14	13	12	11	10	9	8
	LOCK[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LOCK[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	x

Bits 15:0 – LOCK[15:0]: Region Lock Bits
In order to set or clear these bits, the CMD register must be used.

Default state after erase will be unlocked (0x0000).

Value	Description
0	The corresponding lock region is locked.
1	The corresponding lock region is not locked.

29. PORT - I/O Pin Controller

29.1 Overview

The IO Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a PORT group. Each PORT group can have up to 32 pins that can be configured and controlled individually or as a group. The number of PORT groups on a device may depend on the package/number of pins. Each pin may either be used for general-purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general-purpose I/O, each pin can be configured as input or output, with highly configurable driver and pull settings.

All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of one or more pins may be changed (set, reset or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-, 16- or 32-bit write.

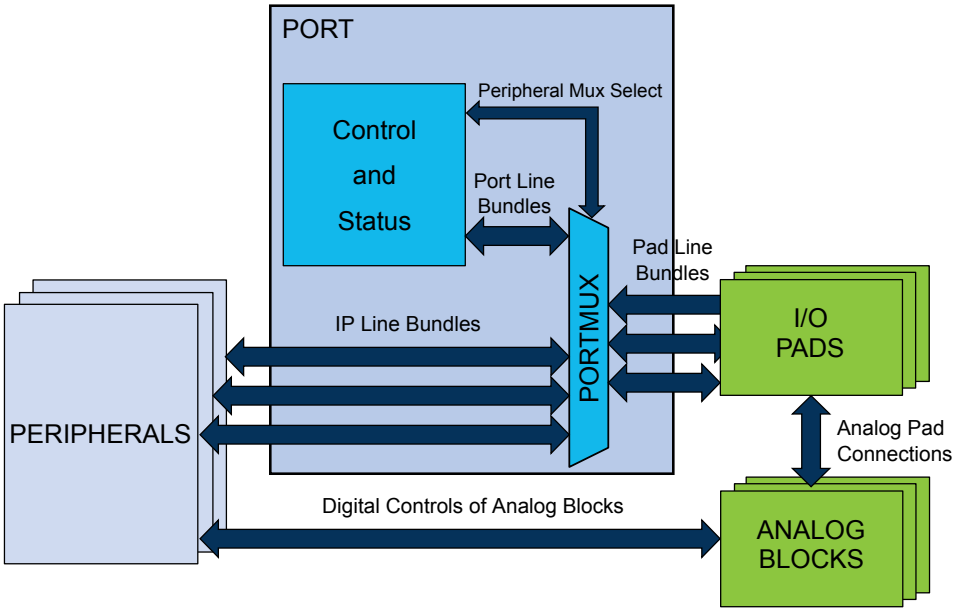
The PORT is connected to the high-speed bus matrix through an AHB/APB bridge. The Pin Direction, Data Output Value and Data Input Value registers may also be accessed using the low-latency CPU local bus (IOBUS; ARM® single-cycle I/O port).

29.2 Features

- Selectable input and output configuration for each individual pin
- Software-controlled multiplexing of peripheral functions on I/O pins
- Flexible pin configuration through a dedicated Pin Configuration register
- Configurable output driver and pull settings:
  - Totem-pole (push-pull)
  - Pull configuration
  - Driver strength
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down
  - Input sampling criteria
  - Input buffer can be disabled if not needed for lower power consumption
- Input event:
  - Up to four input event pins for each PORT group
  - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
  - Can be output to pin
- Power saving using STANDBY mode
  - No access to configuration registers
  - Possible access to data registers (DIR, OUT or IN)

29.3 Block Diagram

Figure 29-1. PORT Block Diagram



29.4 Signal Description

Table 29-1. Signal description for PORT

Signal name	Type	Description
Pxy	Digital I/O	General-purpose I/O pin y in group x

Refer to the *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

29.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly as following.

29.5.1 I/O Lines

The I/O lines of the PORT are mapped to pins of the physical device. The following naming scheme is used:

Each line bundle with up to 32 lines is assigned an identifier 'xy', with letter x=A, B, C… and two-digit number y=00, 01, …31. Examples: A24, C03.

PORT pins are labeled 'Pxy' accordingly, for example PA24, PC03. This identifies each pin in the device uniquely.

Each pin may be controlled by one or more peripheral multiplexer settings, which allow the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral

has control over the output state of the pad, as well as the ability to read the current physical pad state. Refer to *I/O Multiplexing and Considerations* for details.

Device-specific configurations may cause some lines (and the corresponding Pxy pin) not to be implemented.

Related Links

I/O Multiplexing and Considerations

29.5.2 Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled.

When the device is set to the BACKUP sleep mode, even if the PORT configuration registers and input synchronizers will lose their contents (these will not be restored when PORT is powered up again), the latches in the pads will keep their current configuration, such as the output value and pull settings. Refer to the Power Manager documentation for more features related to the I/O lines configuration in and out of BACKUP mode.

The PORT peripheral will continue operating in any sleep mode where its source clock is running.

29.5.3 Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the *Peripheral Clock Masking* section in *MCLK – Main Clock*.

The PORT is fed by two different clocks: a CPU main clock, which allows the CPU to access the PORT through the low latency CPU local bus (IOBUS); an APB clock, which is a divided clock of the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

The priority of IOBUS accesses is higher than event accesses and APB accesses. The EVSYS and APB will insert wait states in the event of concurrent PORT accesses.

The PORT input synchronizers use the CPU main clock so that the resynchronization delay is minimized with respect to the APB clock.

Related Links

MCLK – Main Clock

29.5.4 DMA

Not applicable.

29.5.5 Interrupts

Not applicable.

29.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System

29.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

29.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

Related Links

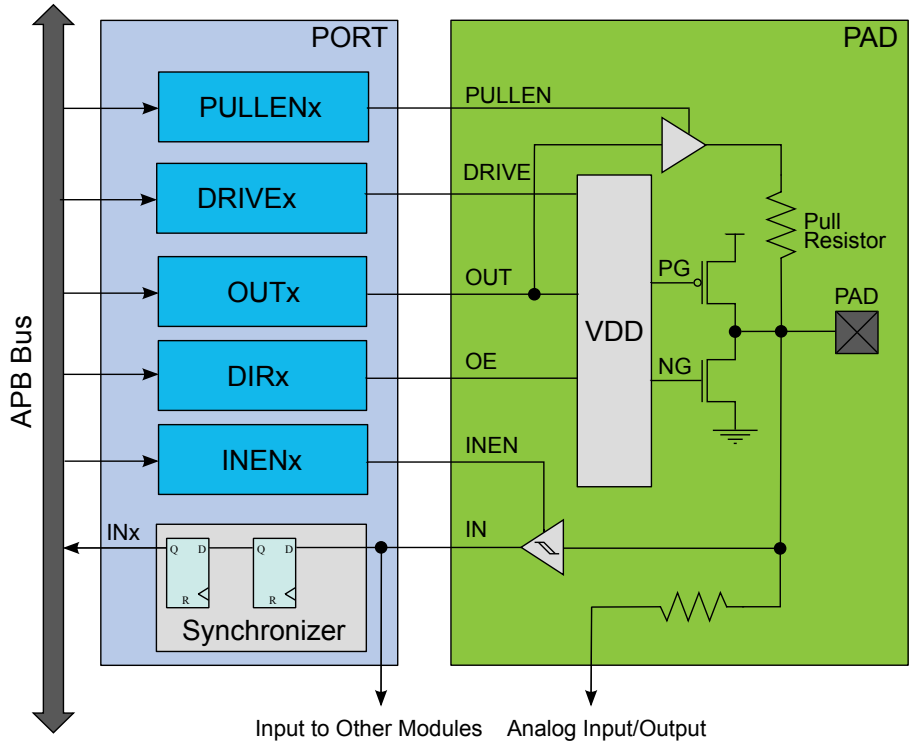
PAC - Peripheral Access Controller

29.5.9 Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

29.6 Functional Description

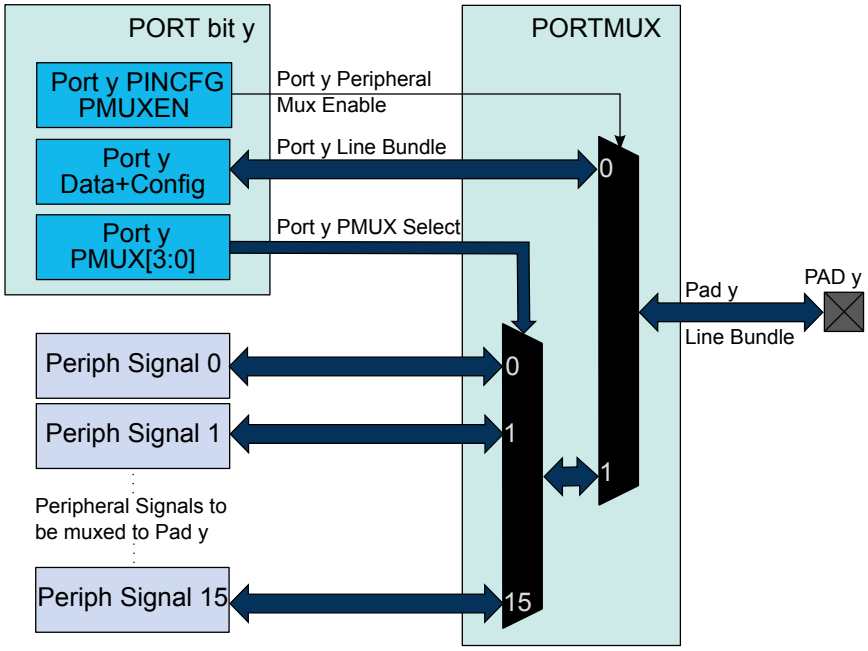
Figure 29-2. Overview of the PORT



29.6.1 Principle of Operation

Each PORT group of up to 32 pins is controlled by the registers in PORT, as described in the figure. These registers in PORT are duplicated for each PORT group, with increasing base addresses. The number of PORT groups may depend on the package/number of pins.

Figure 29-3. Overview of the peripheral functions multiplexing



The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the output state.

The direction of each pin in a PORT group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit y in OUT is written to '1', pin y is driven HIGH. If bit y in OUT is written to '0', pin y is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGy) registers, with y=00, 01, ..31 representing the bit position.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers are clocked only when system requires reading the input value. The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGy.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGy register (PINCFGy.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing n (PMUXn) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, and connect the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

29.6.2 Basic Operation

29.6.2.1 Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled, even if there is no clock running.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.

29.6.2.2 Operation

Each I/O pin y can be controlled by the registers in PORT. Each PORT group has its own set of PORT registers, the base address of the register set for pin y is at byte address PORT + ([y] * 0x4). The index within that register set is [y].

To use pin number y as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

To use pin y as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGy.INEN) is written to '1'.

Refer to *I/O Multiplexing and Considerations* for details on pin configuration and PORT groups.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two CLK_PORT cycles. To remove the delay, the input synchronizers for each PORT group of eight pins can be configured to be always active, but this will increase power consumption. This is enabled by writing '1' to the corresponding SAMPLINGn bit field of the CTRL register, see CTRL.SAMPLING for details.

To use pin y as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGy register must be '1'. The PINCFGy register for pin y is at byte offset (PINCFG0 + [y]).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXn register. The PMUXO/PMUXE is at byte offset PMUX0 + (y/2). The chosen peripheral must also be configured and enabled.

Related Links

I/O Multiplexing and Considerations

29.6.3 I/O Pin Configuration

The Pin Configuration register (PINCFGy) is used for additional I/O pin configuration. A pin can be set in a totem-pole or pull configuration.

As pull configuration is done through the Pin Configuration register, all intermediate PORT states during switching of pin direction and pin values are avoided.

The I/O pin configurations are described further in this chapter, and summarized in Table 29-2.

29.6.3.1 Pin Configurations Summary

Table 29-2. Pin Configurations Summary

DIR	INEN	PULLEN	OUT	Configuration
0	0	0	X	Reset or analog I/O: all digital disabled
0	0	1	0	Pull-down; input disabled
0	0	1	1	Pull-up; input disabled
0	1	0	X	Input
0	1	1	0	Input with pull-down

DIR	INEN	PULLEN	OUT	Configuration
0	1	1	1	Input with pull-up
1	0	X	X	Output; input disabled
1	1	X	X	Output; input enabled

29.6.3.2 Input Configuration

Figure 29-4. I/O configuration - Standard Input

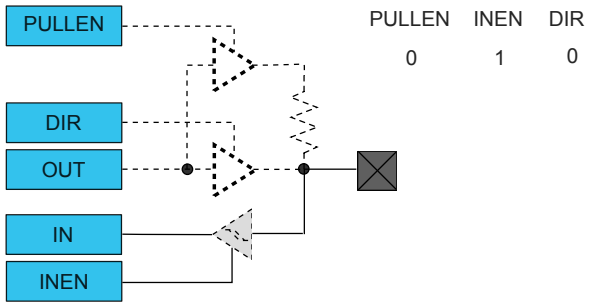
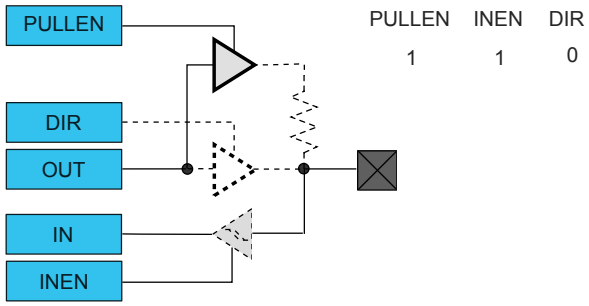


Figure 29-5. I/O Configuration - Input with Pull



Note: When pull is enabled, the pull value is defined by the OUT value.

29.6.3.3 Totem-Pole Output

When configured for totem-pole (push-pull) output, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull is connected.

Note: Enabling the output driver will automatically disable pull.

Figure 29-6. I/O Configuration - Totem-Pole Output with Disabled Input

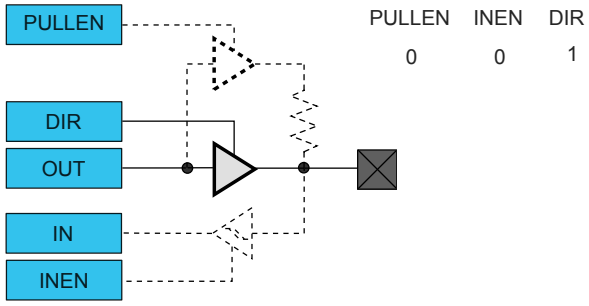


Figure 29-7. I/O Configuration - Totem-Pole Output with Enabled Input

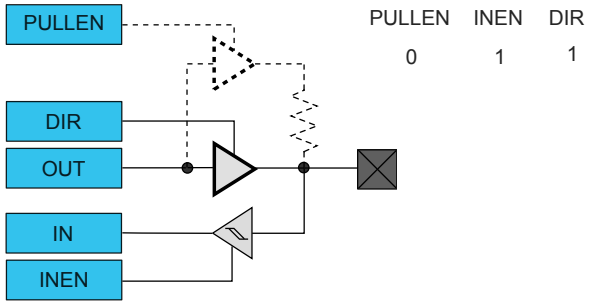
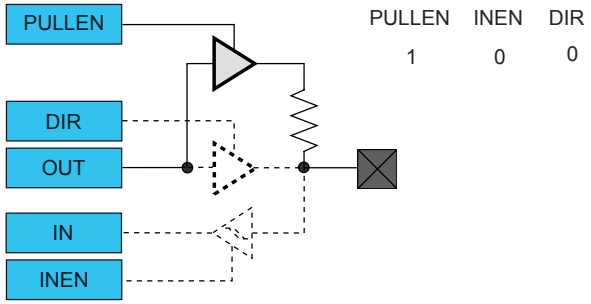


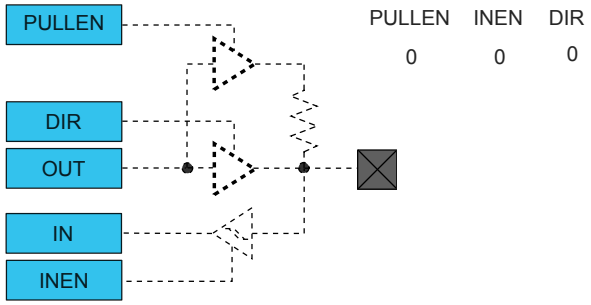
Figure 29-8. I/O Configuration - Output with Pull



29.6.3.4 Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

Figure 29-9. I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled



29.6.4 Events

The PORT allows input events to control individual I/O pins. These input events are generated by the EVSYS module and can originate from a different clock domain than the PORT module.

The PORT can perform the following actions:

- Output (OUT): I/O pin will be set when the incoming event has a high level ('1') and cleared when the incoming event has a low-level ('0').
- Set (SET): I/O pin will be set when an incoming event is detected.
- Clear (CLR): I/O pin will be cleared when an incoming event is detected.
- Toggle (TGL): I/O pin will toggle when an incoming event is detected.

The event is output to pin without any internal latency. For SET, CLEAR and TOGGLE event actions, the action will be executed up to three clock cycles after a rising edge.

The event actions can be configured with the Event Action m bit group in the Event Input Control register(EVCTRL.EVACTm). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIm) enables the corresponding action on input event. Writing '0' to this bit disables the

corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. Refer to *EVSYS – Event System*. for details on configuring the Event System.

Each event input can address one and only one I/O pin at a time. The selection of the pin is indicated by the PORT Event Pin Identifier of the Event Input Control register (EVCTR.PIDn). On the other hand, one I/O pin can be addressed by up to four different input events. To avoid action conflict on the output value of the register (OUT) of this particular I/O pin, only one action is performed according to the table below.

Note that this truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

Table 29-3. Priority on Simultaneous SET/CLR/TGL Event Actions

EVACT0	EVACT1	EVACT2	EVACT3	Executed Event Action
SET	SET	SET	SET	SET
CLR	CLR	CLR	CLR	CLR
All Other Combinations				TGL

Be careful when the event is output to pin. Due to the fact the events are received asynchronously, the I/O pin may have unpredictable levels, depending on the timing of when the events are received. When several events are output to the same pin, the lowest event line will get the access. All other events will be ignored.

Related Links

EVSYS – Event System

29.6.5 PORT Access Priority

The PORT is accessed by different systems:

- The ARM® CPU through the ARM® single-cycle I/O port (IOBUS)
- The ARM® CPU through the high-speed matrix and the AHB/APB bridge (APB)
- EVSYS through four asynchronous input events

The following priority is adopted:

1. ARM® CPU IOBUS (No wait tolerated)
2. APB
3. EVSYS input events

For input events that require different actions on the same I/O pin, refer to Events.

29.7 Register Summary

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Offset	Name	Bit Pos.								
0x00	DIR	7:0	DIR[7:0]							
0x01		15:8	DIR[15:8]							
0x02		23:16	DIR[23:16]							
0x03		31:24	DIR[31:24]							
0x04	DIRCLR	7:0	DIRCLR[7:0]							
0x05		15:8	DIRCLR[15:8]							
0x06		23:16	DIRCLR[23:16]							
0x07		31:24	DIRCLR[31:24]							
0x08	DIRSET	7:0	DIRSET[7:0]							
0x09		15:8	DIRSET[15:8]							
0x0A		23:16	DIRSET[23:16]							
0x0B		31:24	DIRSET[31:24]							
0x0C	DIRTGL	7:0	DIRTGL[7:0]							
0x0D		15:8	DIRTGL[15:8]							
0x0E		23:16	DIRTGL[23:16]							
0x0F		31:24	DIRTGL[31:24]							
0x10	OUT	7:0	OUT[7:0]							
0x11		15:8	OUT[15:8]							
0x12		23:16	OUT[23:16]							
0x13		31:24	OUT[31:24]							
0x14	OUTCLR	7:0	OUTCLR[7:0]							
0x15		15:8	OUTCLR[15:8]							
0x16		23:16	OUTCLR[23:16]							
0x17		31:24	OUTCLR[31:24]							
0x18	OUTSET	7:0	OUTSET[7:0]							
0x19		15:8	OUTSET[15:8]							
0x1A		23:16	OUTSET[23:16]							
0x1B		31:24	OUTSET[31:24]							
0x1C	OUTTGL	7:0	OUTTGL[7:0]							
0x1D		15:8	OUTTGL[15:8]							
0x1E		23:16	OUTTGL[23:16]							
0x1F		31:24	OUTTGL[31:24]							
0x20	IN	7:0	IN[7:0]							
0x21		15:8	IN[15:8]							
0x22		23:16	IN[23:16]							
0x23		31:24	IN[31:24]							
0x24	CTRL	7:0	SAMPLING[7:0]							
0x25		15:8	SAMPLING[15:8]							
0x26		23:16	SAMPLING[23:16]							
0x27		31:24	SAMPLING[31:24]							
0x28	WRCONFIG	7:0	PINMASK[7:0]							

Offset	Name	Bit Pos.								
0x29		15:8	PINMASK[15:8]							
0x2A		23:16		DRVSTR				PULLEN	INEN	PMUXEN
0x2B		31:24	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
0x2C	EVCTRL	7:0	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
0x2D		15:8	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
0x2E		23:16	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
0x2F		31:24	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
0x30	PMUX0	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x31	PMUX1	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x32	PMUX2	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x33	PMUX3	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x34	PMUX4	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x35	PMUX5	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x36	PMUX6	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x37	PMUX7	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x38	PMUX8	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x39	PMUX9	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3A	PMUX10	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3B	PMUX11	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3C	PMUX12	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3D	PMUX13	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3E	PMUX14	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3F	PMUX15	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x40	PINCFG0	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x41	PINCFG1	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x42	PINCFG2	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x43	PINCFG3	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x44	PINCFG4	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x45	PINCFG5	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x46	PINCFG6	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x47	PINCFG7	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x48	PINCFG8	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x49	PINCFG9	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4A	PINCFG10	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4B	PINCFG11	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4C	PINCFG12	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4D	PINCFG13	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4E	PINCFG14	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x4F	PINCFG15	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x50	PINCFG16	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x51	PINCFG17	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x52	PINCFG18	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x53	PINCFG19	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x54	PINCFG20	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x55	PINCFG21	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x56	PINCFG22	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x57	PINCFG23	7:0		DRVSTR				PULLEN	INEN	PMUXEN

Offset	Name	Bit Pos.								
0x58	PINCFG24	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x59	PINCFG25	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5A	PINCFG26	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5B	PINCFG27	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5C	PINCFG28	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5D	PINCFG29	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5E	PINCFG30	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5F	PINCFG31	7:0		DRVSTR				PULLEN	INEN	PMUXEN

29.8 PORT Pin Groups and Register Repetition

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

29.9 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

29.9.1 Data Direction

This register allows the user to configure one or more I/O pins as an input or output. This register can be manipulated without doing a read-modify-write operation by using the Data Direction Toggle (DIRTGL), Data Direction Clear (DIRCLR) and Data Direction Set (DIRSET) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: DIR
Offset: 0x00 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIR[31:0]: Port Data Direction
These bits set the data direction for the individual I/O pins in the PORT group.

Value	Description
0	The corresponding I/O pin in the PORT group is configured as an input.
1	The corresponding I/O pin in the PORT group is configured as an output.

29.9.2 Data Direction Clear

This register allows the user to set one or more I/O pins as an input, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Set (DIRSET) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: DIRCLR
Offset: 0x04 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRCLR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRCLR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRCLR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRCLR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRCLR[31:0]: Port Data Direction Clear
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will clear the corresponding bit in the DIR register, which configures the I/O pin as an input.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as input.

29.9.3 Data Direction Set

This register allows the user to set one or more I/O pins as an output, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Clear (DIRCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: DIRSET
Offset: 0x08 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRSET[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRSET[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRSET[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRSET[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRSET[31:0]: Port Data Direction Set
Writing '0' to a bit has no effect.

Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as an output.

29.9.4 Data Direction Toggle

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET) and Data Direction Clear (DIRCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: DIRTGL
Offset: 0x0C [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRTGL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRTGL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRTGL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRTGL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRTGL[31:0]: Port Data Direction Toggle
Writing '0' to a bit has no effect.

Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The direction of the corresponding I/O pin is toggled.

29.9.5 Data Output Value

This register sets the data output drive value for the individual I/O pins in the PORT.
This register can be manipulated without doing a read-modify-write operation by using the Data Output Value Clear (OUTCLR), Data Output Value Set (OUTSET), and Data Output Value Toggle (OUTTGL) registers.

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: OUT
Offset: 0x10 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUT[31:0]: Port Data Output Value
For pins configured as outputs via the Data Direction register (DIR), these bits set the logical output drive level.

For pins configured as inputs via the Data Direction register (DIR) and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN), these bits will set the input pull direction.

Value	Description
0	The I/O pin output is driven low, or the input is connected to an internal pull-down.
1	The I/O pin output is driven high, or the input is connected to an internal pull-up.

29.9.6 Data Output Value Clear

This register allows the user to set one or more output I/O pin drive levels low, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Set (OUTSET) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: OUTCLR
Offset: 0x14 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUTCLR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTCLR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTCLR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTCLR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTCLR[31:0]: PORT Data Output Value Clear
Writing '0' to a bit has no effect.

Writing '1' to a bit will clear the corresponding bit in the OUT register. Pins configured as outputs via the Data Direction register (DIR) will be set to low output drive level. Pins configured as inputs via DIR and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN) will set the input pull direction to an internal pull-down.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down.

29.9.7 Data Output Value Set

This register allows the user to set one or more output I/O pin drive levels high, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Clear (OUTCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: OUTSET
Offset: 0x18 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUTSET[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTSET[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTSET[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTSET[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTSET[31:0]: PORT Data Output Value Set
Writing '0' to a bit has no effect.

Writing '1' to a bit will set the corresponding bit in the OUT register, which sets the output drive level high for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will set the input pull direction to an internal pull-up.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin output is driven high, or the input is connected to an internal pull-up.

29.9.8 Data Output Value Toggle

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET) and Data Output Value Clear (OUTCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: OUTTGL
Offset: 0x1C [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUTTGL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTTGL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTTGL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTTGL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTTGL[31:0]: PORT Data Output Value Toggle
Writing '0' to a bit has no effect.

Writing '1' to a bit will toggle the corresponding bit in the OUT register, which inverts the output drive level for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will toggle the input pull direction.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding OUT bit value is toggled.

29.9.9 Data Input Value

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: IN
Offset: 0x20 [ID-000011ca]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	IN[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IN[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IN[31:0]: PORT Data Input Value
These bits are cleared when the corresponding I/O pin input sampler detects a logical low level on the input pin.

These bits are set when the corresponding I/O pin input sampler detects a logical high level on the input pin.

29.9.10 Control

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: CTRL
Offset: 0x24 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	SAMPLING[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPLING[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SAMPLING[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SAMPLING[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SAMPLING[31:0]: Input Sampling Mode
Configures the input sampling functionality of the I/O pin input samplers, for pins configured as inputs via the Data Direction register (DIR).

The input samplers are enabled and disabled in sub-groups of eight. Thus if any pins within a byte request continuous sampling, all pins in that eight pin sub-group will be continuously sampled.

Value	Description
0	The I/O pin input synchronizer is disabled.
1	The I/O pin input synchronizer is enabled.

29.9.11 Write Configuration

This write-only register is used to configure several pins simultaneously with the same configuration and/or peripheral multiplexing.

In order to avoid side effect of non-atomic access, 8-bit or 16-bit writes to this register will have no effect. Reading this register always returns zero.

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Name: WRCONFIG
Offset: 0x28 [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
Access	W	W		W	W	W	W	W
Reset	0	0		0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		DRVSTR				PULLEN	INEN	PMUXEN
Access		W				W	W	W
Reset		0				0	0	0

Bit	15	14	13	12	11	10	9	8
	PINMASK[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PINMASK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 31 – HWSEL: Half-Word Select
This bit selects the half-word field of a 32-PORT group to be reconfigured in the atomic write operation.

This bit will always read as zero.

Value	Description
0	The lower 16 pins of the PORT group will be configured.
1	The upper 16 pins of the PORT group will be configured.

Bit 30 – WRPINCFG: Write PINCFG
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGy) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVSTR, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN and WRCONFIG.PINMASK values.

This bit will always read as zero.

Value	Description
0	The PINCFGy registers of the selected pins will not be updated.
1	The PINCFGy registers of the selected pins will be updated.

Bit 28 – WRPMUX: Write PMUX
This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG. PMUX value.

This bit will always read as zero.

Value	Description
0	The PMUXn registers of the selected pins will not be updated.
1	The PMUXn registers of the selected pins will be updated.

Bits 27:24 – PMUX[3:0]: Peripheral Multiplexing
These bits determine the new value written to the Peripheral Multiplexing register (PMUXn) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.

These bits will always read as zero.

Bit 22 – DRVSTR: Output Driver Strength Selection
This bit determines the new value written to PINCFGy.DRVSTR for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 18 – PULLEN: Pull Enable
This bit determines the new value written to PINCFGy.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 17 – INEN: Input Enable
This bit determines the new value written to PINCFGy.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 16 – PMUXEN: Peripheral Multiplexer Enable
This bit determines the new value written to PINCFGy.PMUXEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bits 15:0 – PINMASK[15:0]: Pin Mask for Multiple Pin Configuration
These bits select the pins to be configured within the half-word group selected by the WRCONFIG.HWSEL bit.

These bits will always read as zero.

Value	Description
0	The configuration of the corresponding I/O pin in the half-word group will be left unchanged.
1	The configuration of the corresponding I/O pin in the half-word PORT group will be updated.

29.9.12 Event Input Control

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

Name: EVCTRL
Offset: 0x2C [ID-000011ca]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31,23,15,7 – PORTEIx: PORT Event Input x Enable [x = 3..0]

Value	Description
0	The event action x (EVACTx) will not be triggered on any incoming event.
1	The event action x (EVACTx) will be triggered on any incoming event.

Bits 30:29, 22:21,14:13,6:5 – EVACTx: PORT Event Action x [x = 3..0]
These bits define the event action the PORT will perform on event input x. See also Table 29-4.

Bits 28:24,20:16,12:8,4:0 – PIDx: PORT Event Pin Identifier x [x = 3..0]
These bits define the I/O pin on which the event action will be performed, according to Table 29-5.

Table 29-4. PORT Event x Action (x = [3..0])

Value	Name	Description
0x0	OUT	Output register of pin will be set to level of event.
0x1	SET	Set output register of pin on event.
0x2	CLR	Clear output register of pin on event.
0x3	TGL	Toggle output register of pin on event.

Table 29-5. PORT Event x Pin Identifier (x = [3..0])

Value	Name	Description
0x0	PIN0	Event action to be executed on PIN 0.
0x1	PIN1	Event action to be executed on PIN 1.
...	...	...
0x31	PIN31	Event action to be executed on PIN 31.

29.9.13 Peripheral Multiplexing n

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The n denotes the number of the set of I/O lines.

Name: PMUX
Offset: 0x30 + n*0x01 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	PMUXO[3:0]				PMUXE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – PMUXO[3:0]: Peripheral Multiplexing for Odd-Numbered Pin
These bits select the peripheral function for odd-numbered pins (2*n + 1) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations*.

PMUXO[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected

PMUXO[3:0]	Name	Description
0x5	F	Peripheral function F selected
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

Bits 3:0 – PMUXE[3:0]: Peripheral Multiplexing for Even-Numbered Pin
These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations.*

PMUXE[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	F	Peripheral function F selected
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

29.9.14 Pin Configuration

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Name: PINCFG
Offset: 0x40 + n*0x01 [n=0..31]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		DRVSTR				PULLEN	INEN	PMUXEN
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 6 – DRVSTR: Output Driver Strength Selection
This bit controls the output driver strength of an I/O pin configured as an output.

Value	Description
0	Pin drive strength is set to normal drive strength.
1	Pin drive strength is set to stronger drive strength.

Bit 2 – PULLEN: Pull Enable
This bit enables the internal pull-up or pull-down resistor of an I/O pin configured as an input.

Value	Description
0	Internal pull resistor is disabled, and the input is in a high-impedance configuration.
1	Internal pull resistor is enabled, and the input is driven to a defined logic level in the absence of external input.

Bit 1 – INEN: Input Enable
This bit controls the input buffer of an I/O pin configured as either an input or output.

Writing a zero to this bit disables the input buffer completely, preventing read-back of the physical pin state when the pin is configured as either an input or output.

Value	Description
0	Input buffer for the I/O pin is disabled, and the input value will not be sampled.
1	Input buffer for the I/O pin is enabled, and the input value will be sampled when required.

Bit 0 – PMUXEN: Peripheral Multiplexer Enable
This bit enables or disables the peripheral multiplexer selection set in the Peripheral Multiplexing register (PMUXn) to enable or disable alternative peripheral control over an I/O pin direction and output drive value.

Writing a zero to this bit allows the PORT to control the pad direction via the Data Direction register (DIR) and output drive value via the Data Output Value register (OUT). The peripheral multiplexer value in PMUXn is ignored. Writing '1' to this bit enables the peripheral selection in PMUXn to control the pad. In this configuration, the physical pin state may still be read from the Data Input Value register (IN) if PINCFGn.INEN is set.

Value	Description
0	The peripheral multiplexer selection is disabled, and the PORT registers control the direction and output drive value.
1	The peripheral multiplexer selection is enabled, and the selected peripheral function controls the direction and output drive value.

30. EVSYS – Event System

30.1 Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users.

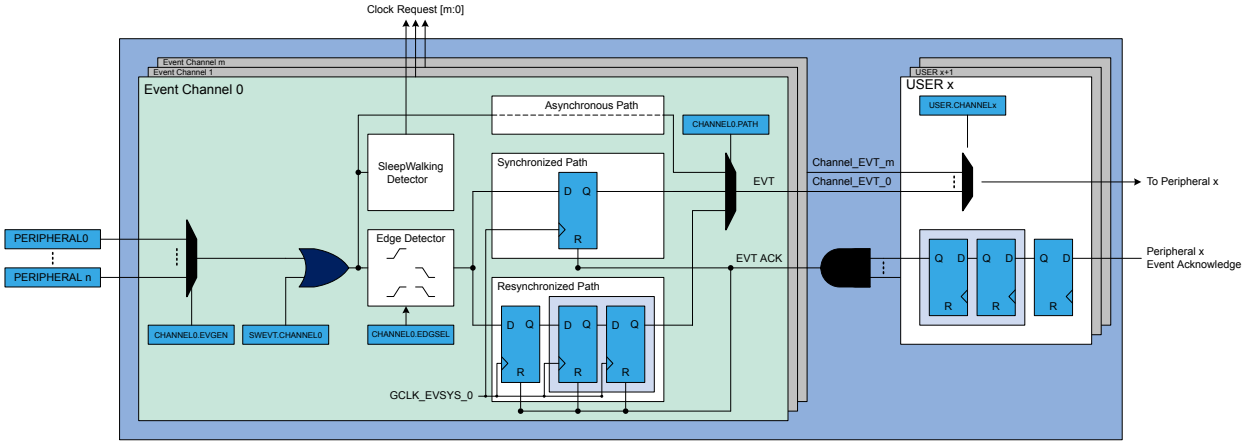
Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

30.2 Features

- 8 configurable event channels, where each channel can:
  - Be connected to any event generator.
  - Provide a pure asynchronous, resynchronized or synchronous path
- 69 event generators.
- 31 event users.
- Configurable edge detector.
- Peripherals can be event generators, event users, or both.
- SleepWalking and interrupt for operation in sleep modes.
- Software event generation.
- Each event user can choose which channel to respond to.

30.3 Block Diagram

Figure 30-1. Event System Block Diagram



30.4 Signal Description

Not applicable.

30.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

30.5.1 I/O Lines

Not applicable.

30.5.2 Power Management

The EVSYS can be used to wake up the CPU from all sleep modes, even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM – Power Manager* for details on the different sleep modes.

In all sleep modes, although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CHANNEL_n) will be restarted if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

Related Links

PM – Power Manager

30.5.3 Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_EVSYS_APB can be found in *Peripheral Clock Masking*.

Each EVSYS channel has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to *GCLK - Generic Clock Controller* for details.

Related Links

Peripheral Clock Masking

GCLK - Generic Clock Controller

30.5.4 DMA

Not applicable.

30.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the EVSYS interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

30.5.6 Events

Not applicable.

30.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or

data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

30.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Channel Status (CHSTATUS)
- Interrupt Flag Status and Clear register (INTFLAG)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

30.5.9 Analog Connections

Not applicable.

30.6 Functional Description

30.6.1 Principle of Operation

The Event System consists of several channels which route the internal events from peripherals (generators) to other internal peripherals or IO pins (users). Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous, synchronous or re-synchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using synchronous or resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on on event generators.

For further details, refer to “Channel Path” of this chapter.

30.6.2 Basic Operation

30.6.2.1 Initialization

Before enabling events routing within the system, the Event Users Multiplexer and Event Channels must be configured. The Event Users Multiplexer must be configured first.

For further details about the event user multiplexer configuration, refer to “User Multiplexer Setup”.

For further details about the event channels configuration, refer to “Event System Channel”.

30.6.2.2 Enabling, Disabling, and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS will be reset to their initial state and all ongoing events will be canceled.

Refer to CTRLA.SWRST register for details.

30.6.2.3 User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be

configured to select one of these channels, as shown in Figure 30-1. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all user multiplexers is found in the User (USERm) register description.

Related Links

USERn

30.6.2.4 Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in Figure 30-1.

30.6.2.5 Event Generators

Each event channel can receive the events form all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (ie, CHANNELn.EVGEN = 0)

30.6.2.6 Channel Path

There are three different ways to propagate the event from an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

Asynchronous Path

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CHANNEL_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel Status register (CHSTATUS) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

Synchronous Path

The synchronous path should be used when the event generator and the event channel share the same generator for the generic clock. If they do not share the same clock, a logic change from the event generator to the event channel might not be detected in the channel, which means that the event will not be propagated to the event user. For details on generic clock generators, refer to *GCLK - Generic Clock Controller*.

When using the synchronous path, the channel is able to generate interrupts. The channel busy n bit in the Channel Status register (CHSTATUS.CHBUSYn) are also updated and available for use.

Resynchronized Path

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel. For details on generic clock generators, refer to *GCLK - Generic Clock Controller*.

When the resynchronized path is used, the channel is able to generate interrupts. The channel busy n bits in the Channel Status register (CHSTATUS.CHBUSYn) are also updated and available for use.

Related Links

GCLK - Generic Clock Controller

30.6.2.7 Edge Detection

When synchronous or resynchronized paths are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

30.6.2.8 Event Latency

An event from an event generator is propagated to an event user with different latency, depending on event channel configuration.

- Asynchronous Path: The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.
- Synchronous Path: The maximum routing latency of an external event is one GCLK_EVSYS_CHANNEL_n clock cycle.
- Resynchronized Path: The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_n clock cycles.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

30.6.2.9 The Overrun Channel n Interrupt

The Overrun Channel n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVRn) will be set, and the optional interrupt will be generated in the following cases:

- One or more event users on channel n is not ready when there is a new event.
- An event occurs when the previous event on channel m has not been handled by all event users connected to that channel.

The flag will only be set when using synchronous or resynchronized paths. In the case of asynchronous path, the INTFLAG.OVRn is always read as zero.

30.6.2.10 The Event Detected Channel n Interrupt

The Event Detected Channel n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.EVDn) is set when an event coming from the event generator configured on channel n is detected.

The flag will only be set when using a synchronous or resynchronized paths. In the case of asynchronous path, the INTFLAG.EVDn is always zero.

30.6.2.11 Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a synchronous or resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUS.CHBUSYn bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel.
- The CHSTATUS.USRRDYn bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel.

30.6.2.12 Software Event

A software event can be initiated on a channel by setting the Channel n bit in the Software Event register (SWEVT.CHANNELn) to '1'. Then the software event can be serviced as any event generator; i.e., when the bit is set to '1', an event will be generated on the respective channel.

30.6.3 Interrupts

The EVSYS has the following interrupt sources:

- Overrun Channel n interrupt (OVRn): for details, refer to The Overrun Channel n Interrupt.
- Event Detected Channel n interrupt (EVDn): for details, refer to The Event Detected Channel n Interrupt.

These interrupts events are asynchronous wake-up sources. See *Sleep Mode Controller*. Each interrupt source has an interrupt flag which is in the Interrupt Flag Status and Clear (INTFLAG) register. The flag is set when the interrupt is issued. Each interrupt event can be individually enabled by setting a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by setting a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt event is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt event works until the interrupt flag is cleared, the interrupt is disabled, or the Event System is reset. See INTFLAG for details on how to clear interrupt flags.

All interrupt events from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the *Nested Vector Interrupt Controller* for details. The event user must read the INTFLAG register to determine what the interrupt condition is.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

Sleep Mode Controller

30.6.4 Sleep Mode Operation

The EVSYS can generate interrupts to wake up the device from any sleep mode.

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in Channel register (CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path will increase by two GCLK_EVSYS_CHANNEL_n clock (i.e., up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes regarding to CHANNELn.RUNSTDBY and CHANNELn.ONDEMAND, as shown in the table below:

Table 30-1. Event Channel Sleep Behavior

CHANNELn.ONDEMAND	CHANNELn.RUNSTDBY	Sleep Behavior
0	0	Only run in IDLE sleep mode if an event must be propagated. Disabled in STANDBY sleep mode.
0	1	Always run in IDLE and STANDBY sleep modes.
1	0	Only run in IDLE sleep mode if an event must be propagated. Disabled in STANDBY sleep mode. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.
1	1	Always run in IDLE and STANDBY sleep modes. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.

30.7 Register Summary

30.7.1 Common Registers

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								SWRST
0x01..0x0B	Reserved									
0x0C	CHSTATUS	7:0	USRRDY7	USRRDY6	USRRDY5	USRRDY4	USRRDY3	USRRDY2	USRRDY1	USRRDY0
0x0D		15:8								
0x0E		23:16	CHBUSY7	CHBUSY6	CHBUSY5	CHBUSY4	CHBUSY3	CHBUSY2	CHBUSY1	CHBUSY0
0x0F		31:24								
0x10	INTENCLR	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
0x11		15:8								
0x12		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
0x13		31:24								
0x14	INTENSET	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
0x15		15:8								
0x16		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
0x17		31:24								
0x18	INTFLAG	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
0x19		15:8								
0x1A		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
0x1B		31:24								
0x1C	SWEVT	7:0	CHANNEL[7:0]							
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								

30.7.2 CHANNELn

Offset	Name	Bit Pos.								
0x20 + 0x4*n	CHANNELn	7:0		EVGEN[6:0]						
0x21 + 0x4*n		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
0x22 + 0x4*n		23:16								
0x23 + 0x4*n		31:24								

30.7.3 USERm

Offset	Name	Bit Pos.								
0x80 + 0x4*m	USERm	7:0				CHANNEL[4:0]				
0x81 + 0x4*m		15:8								
0x82 + 0x4*m		23:16								
0x83 + 0x4*m		31:24								

30.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Refer to Register Access Protection and *PAC - Peripheral Access Controller*.

Related Links

PAC - Peripheral Access Controller

30.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-0000120d]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								0

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state.

Note: Before applying a Software Reset it is recommended to disable the event generators.

30.8.2 Channel Status

Name: CHSTATUS
Offset: 0x0C [ID-0000120d]
Reset: 0x000000FF
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CHBUSYn	CHBUSYn	CHBUSYn	CHBUSYn	CHBUSYn	CHBUSYn	CHBUSYn	CHBUSYn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	USRRDYn	USRRDYn	USRRDYn	USRRDYn	USRRDYn	USRRDYn	USRRDYn	USRRDYn
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bits 23:16 – CHBUSYn: Channel Busy n [n = 7..0]
This bit is cleared when channel n is idle.

This bit is set if an event on channel n has not been handled by all event users connected to channel n.

Bits 7:0 – USRRDYn: User Ready for Channel n [n = 7..0]
This bit is cleared when at least one of the event users connected to the channel is not ready.

This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.

30.8.3 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x10 [ID-0000120d]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	OVRn	OVRn	OVRn	OVRn	OVRn	OVRn	OVRn	OVRn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – EVDn: Event Detected Channel n Interrupt Enable [n = 7..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Event Detected Channel n Interrupt Enable bit, which disables the Event Detected Channel n interrupt.

Value	Description
0	The Event Detected Channel n interrupt is disabled.
1	The Event Detected Channel n interrupt is enabled.

Bits 7:0 – OVRn: Overrun Channel n Interrupt Enable[n = 7..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Overrun Channel n Interrupt Enable bit, which disables the Overrun Channel n interrupt.

Value	Description
0	The Overrun Channel n interrupt is disabled.
1	The Overrun Channel n interrupt is enabled.

30.8.4 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x14 [ID-0000120d]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVRn	OVRn	OVRn	OVRn	OVRn	OVRn	OVRn
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 23:16 – EVDn: Event Detected Channel n Interrupt Enable [n = 7..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Event Detected Channel n Interrupt Enable bit, which enables the Event Detected Channel n interrupt.

Value	Description
0	The Event Detected Channel n interrupt is disabled.
1	The Event Detected Channel n interrupt is enabled.

Bits 6:0 – OVRn: Overrun Channel n Interrupt Enable [n = 7..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Overrun Channel n Interrupt Enable bit, which disables the Overrun Channel n interrupt.

Value	Description
0	The Overrun Channel n interrupt is disabled.
1	The Overrun Channel n interrupt is enabled.

30.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18 [ID-0000120d]
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn	EVDn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	OVRn	OVRn	OVRn	OVRn	OVRn	OVRn	OVRn	OVRn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – EVDn: Event Detected Channel n [n=7..0]
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if INTENCLR/SET.EVDn is '1'.

When the event channel path is asynchronous, the EVDn interrupt flag will not be set.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Event Detected Channel n interrupt flag.

Bits 7:0 – OVRn: Overrun Channel n [n=7..0]
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if INTENCLR/SET.OVRn is '1'.

When the event channel path is asynchronous, the OVRn interrupt flag will not be set.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Overrun Detected Channel n interrupt flag.

30.8.6 Software Event

Name: SWEVT
Offset: 0x1C [ID-0000120d]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CHANNELn	CHANNELn	CHANNELn	CHANNELn	CHANNELn	CHANNELn	CHANNELn	CHANNELn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CHANNELn: Channel n Software [n=7..0] Selection
Writing '0' to this bit has no effect.

Writing '1' to this bit will trigger a software event for the channel n.

These bits will always return zero when read.

30.8.7 Channel

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

Name: CHANNELn
Offset: 0x20+n*0x4 [n=0..7] [ID-0000120d]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
		EVGEN[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 15 – ONDEMAND: Generic Clock On Demand

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY: Run in Standby

This bit is used to define the behavior during standby sleep mode.

Value	Description
0	The channel is disabled in standby sleep mode.
1	The channel is not stopped in standby sleep mode and depends on the CHANNEL.ONDEMAND

Bits 11:10 – EDGSEL[1:0]: Edge Detection Selection

These bits set the type of edge detection to be used on the channel.

These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized or synchronous path
0x1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator
0x2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator
0x3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator

Bits 9:8 – PATH[1:0]: Path Selection

These bits are used to choose which path will be used by the selected channel.

The path choice can be limited by the channel source, see the table in USERm.

Value	Name	Description
0x0	SYNCHRONOUS	Synchronous path
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path
0x3	-	Reserved

Bits 6:0 – EVGEN[6:0]: Event Generator
These bits are used to choose the event generator to connect to the selected channel.

Value	Event Generator	Description
0x00	NONE	No event generator selected
0x01	OSCCTRL_CFD	Clock Failure Detection
0x02	OSC32KCTRL_CFD	Clock Failure Detection
0x03	RTC CMP0	Compare 0 (mode 0 and 1) or Alarm 0 (mode 2)
0x04	RTC CMP1	Compare 1
0x05	RTC_TAMPER	Tamper Detection
0x06	RTC OVF	Overflow
0x07	RTC PER0	Period 0
0x08	RTC PER1	Period 1
0x09	RTC PER2	Period 2
0x0A	RTC PER3	Period 3
0x0B	RTC PER4	Period 4
0x0C	RTC PER5	Period 5
0x0D	RTC PER6	Period 6
0x0E	RTC PER7	Period 7
0x0F	EIC EXTINT0	External Interrupt 0
0x10	EIC EXTINT1	External Interrupt 1
0x11	EIC EXTINT2	External Interrupt 2
0x12	EIC EXTINT3	External Interrupt 3
0x13	EIC EXTINT4	External Interrupt 4
0x14	EIC EXTINT5	External Interrupt 5
0x15	EIC EXTINT6	External Interrupt 6
0x16	EIC EXTINT7	External Interrupt 7
0x17	EIC EXTINT8	External Interrupt 8
0x18	EIC EXTINT9	External Interrupt 9
0x19	EIC EXTINT10	External Interrupt 10

Value	Event Generator	Description
0x1A	EIC EXTINT11	External Interrupt 11
0x1B	EIC EXTINT12	External Interrupt 12
0x1C	EIC EXTINT13	External Interrupt 13
0x1D	EIC EXTINT14	External Interrupt 14
0x1E	EIC EXTINT15	External Interrupt 15
0x1F	DMAC CH0	Channel 0
0x20	DMAC CH1	Channel 1
0x21	DMAC CH2	Channel 2
0x22	DMAC CH3	Channel 3
0x23	TCC0_OVF	Overflow
0x24	TCC0_TRG	Trig
0x25	TCC0_CNT	Counter
0x26	TCC0_MCX0	Match/Capture 0
0x27	TCC0_MCX1	Match/Capture 1
0x28	TCC0_MCX2	Match/Capture 2
0x29	TCC0_MCX3	Match/Capture 3
0x2A	TC0 OVF	Overflow/Underflow
0x2B	TC0 MCX0	Match/Capture 0
0x2C	TC0 MCX1	Match/Capture 1
0x2D	TC1 OVF	Overflow/Underflow
0x2E	TC1 MCX0	Match/Capture 0
0x2F	TC1 MCX1	Match/Capture 1
0x30	TC2 OVF	Overflow/Underflow
0x31	TC2 MCX0	Match/Capture 0
0x32	TC2 MCX1	Match/Capture 1
0x33	TC3 OVF	Overflow/Underflow
0x34	TC3 MCX0	Match/Capture 0
0x35	TC3 MCX1	Match/Capture 1
0x36	ADC RESRDY	Result Ready
0x37	ADC WINMON	Window Monitor
0x38	AC COMP0	Comparator 0
0x39	AC COMP1	Comparator 1

Value	Event Generator	Description
0x3A	AC WIN0	Window 0
0x3B	PTC EOC	End of Conversion
0x3C	PTC WCOMP	Window Comparator
0x3D	SLCD_FC0	Frame Counter 0 overflow
0x3E	SCLD_FC1	Frame Counter 1 overflow
0x3F	SLCD_FC2	Frame Counter 2 overflow
0x41	TRNG READY	Data Ready
0x42	CCL LUTOUT0	CCL output
0x43	CCL LUTOUT1	CCL output
0x44	CCL LUTOUT2	CCL output
0x45	CCL LUTOUT3	CCL output
0x46	PAC EVT	Access Error
0x47-0x7F	Reserved	

30.8.8 Event User m

Name: USERm
Offset: 0x80+m*0x4 [m=0..41] [ID-0000120d]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			CHANNEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – CHANNEL[5:0]: Channel Event Selection

These bits are used to select the channel to connect to the event user.

Note that to select channel m, the value (m+1) must be written to the USER.CHANNEL bit group.

Value	Channel Number
0x00	No channel output selected
0x01	0
0x02	1
0x03	2
0x04	3
0x05	4
0x06	5
0x07	6
0x08	7
0x09-0xFF	Reserved

Table 30-2. User Multiplexer Number

USERm	User Multiplexer	Description	Path Type
m = 0	RTC TAMPEVT	Tamper Event	Asynchronous, synchronous, and resynchronized paths
m = 1	PORT EV0	Event 0	Asynchronous, synchronous, and resynchronized paths
m = 2	PORT EV1	Event 1	Asynchronous, synchronous, and resynchronized paths
m = 3	PORT EV2	Event 2	Asynchronous, synchronous, and resynchronized paths
m = 4	PORT EV3	Event 3	Asynchronous, synchronous, and resynchronized paths
m = 5	DMAC CH0	Channel 0	Synchronous, and resynchronized paths
m = 6	DMAC CH1	Channel 1	Synchronous, and resynchronized paths
m = 7	DMAC CH2	Channel 2	Synchronous, and resynchronized paths
m = 8	DMAC CH3	Channel 3	Synchronous, and resynchronized paths
m = 9	TCC0 EV0	-	Asynchronous, synchronous, and resynchronized paths
m = 10	TCC0 EV1	-	Asynchronous, synchronous, and resynchronized paths
m = 11	TCC0 MC0	Match/Capture 0	Asynchronous, synchronous, and resynchronized paths
m = 12	TCC0 MC1	Match/Capture 1	Asynchronous, synchronous, and resynchronized paths
m = 13	TCC0 MC2	Match/Capture 2	Asynchronous, synchronous, and resynchronized paths

USERm	User Multiplexer	Description	Path Type
m = 14	TCC0 MC3	Match/Capture 3	Asynchronous, synchronous, and resynchronized paths
m = 15	TC0	-	Asynchronous, synchronous, and resynchronized paths
m = 16	TC1	-	Asynchronous, synchronous, and resynchronized paths
m = 17	TC2	-	Asynchronous, synchronous, and resynchronized paths
m = 18	TC3	-	Asynchronous, synchronous, and resynchronized paths
m = 19	ADC START	ADC start conversion	Asynchronous, synchronous, and resynchronized paths
m = 20	ADC SYNC	Flush ADC	Asynchronous, synchronous, and resynchronized paths
m = 21	AC COMP0	Start comparator 0	Asynchronous, synchronous, and resynchronized paths
m = 22	AC COMP1	Start comparator 1	Asynchronous, synchronous, and resynchronized paths
m = 23	PTC STCONV	PTC start conversion	Asynchronous, synchronous, and resynchronized paths
m = 24	CCL LUTIN 0	CCL input	Asynchronous, synchronous, and resynchronized paths
m = 25	CCL LUTIN 1	CCL input	Asynchronous, synchronous, and resynchronized paths
m = 26	CCL LUTIN 2	CCL input	Asynchronous, synchronous, and resynchronized paths
m = 27	CCL LUTIN 3	CCL input	Asynchronous, synchronous, and resynchronized paths

USERm	User Multiplexer	Description	Path Type
m = 29	MTB START	Tracing start	Asynchronous, synchronous, and resynchronized paths
m = 30	MTB STOP	Tracing stop	Asynchronous, synchronous, and resynchronized paths
others	Reserved	-	-

31. SERCOM – Serial Communication Interface

31.1 Overview

There are up to six instances of the serial communication interface (SERCOM) peripheral.

A SERCOM can be configured to support a number of modes: I^2C, SPI, and USART. When SERCOM is configured and enabled, all SERCOM resources will be dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address matching functionality. It can use the internal generic clock or an external clock to operate in all sleep modes.

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

SERCOM SPI – SERCOM Serial Peripheral Interface

SERCOM I2C – SERCOM Inter-Integrated Circuit

SERCOM USART and I2C Configurations

31.2 Features

- Interface for configuring into one of the following:
  - I^2C – Two-wire serial interface
    SMBus™ compatible
  - SPI – Serial peripheral interface
  - USART – Universal synchronous and asynchronous serial receiver and transmitter
- Single transmit buffer and double receive buffer
- Baud-rate generator
- Address match/mask logic
- Operational in all sleep modes
- Can be used with DMA

See the Related Links for full feature lists of the interface configurations.

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

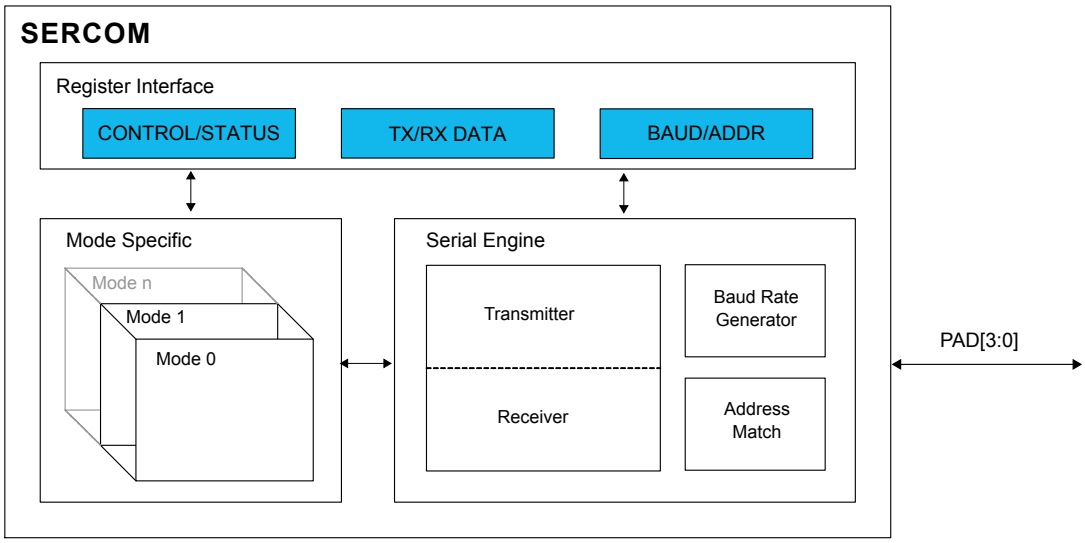
SERCOM SPI – SERCOM Serial Peripheral Interface

SERCOM I2C – SERCOM Inter-Integrated Circuit

SERCOM USART and I2C Configurations

31.3 Block Diagram

Figure 31-1. SERCOM Block Diagram



31.4 Signal Description

See the respective SERCOM mode chapters for details.

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

SERCOM SPI – SERCOM Serial Peripheral Interface

SERCOM I2C – SERCOM Inter-Integrated Circuit

31.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

31.5.1 I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using port configuration (PORT).

From *USART Block Diagram* one can see that the SERCOM has four internal pads, PAD[3:0]. The signals from I2C, SPI and USART are routed through these SERCOM pads via a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes. Refer to the mode specific chapters for details.

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

SERCOM SPI – SERCOM Serial Peripheral Interface

SERCOM I2C – SERCOM Inter-Integrated Circuit

PORT: IO Pin Controller

Block Diagram

31.5.2 Power Management

The SERCOM can operate in any sleep mode where the selected clock source is running. SERCOM interrupts can be used to wake up the device from sleep modes.

Related Links

PM – Power Manager

31.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock.

The SERCOM uses two generic clocks: GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a master. The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions. See specific mode chapters for details.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM.

The generic clocks are asynchronous to the user interface clock (CLK_SERCOMx_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for details.

Related Links

GCLK - Generic Clock Controller

MCLK – Main Clock

31.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used.

Related Links

DMAC – Direct Memory Access Controller

31.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). The NVIC must be configured before the SERCOM interrupts are used.

Related Links

Nested Vector Interrupt Controller

31.5.6 Events

Not applicable.

31.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

31.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

- Address register (ADDR)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

31.5.9 Analog Connections

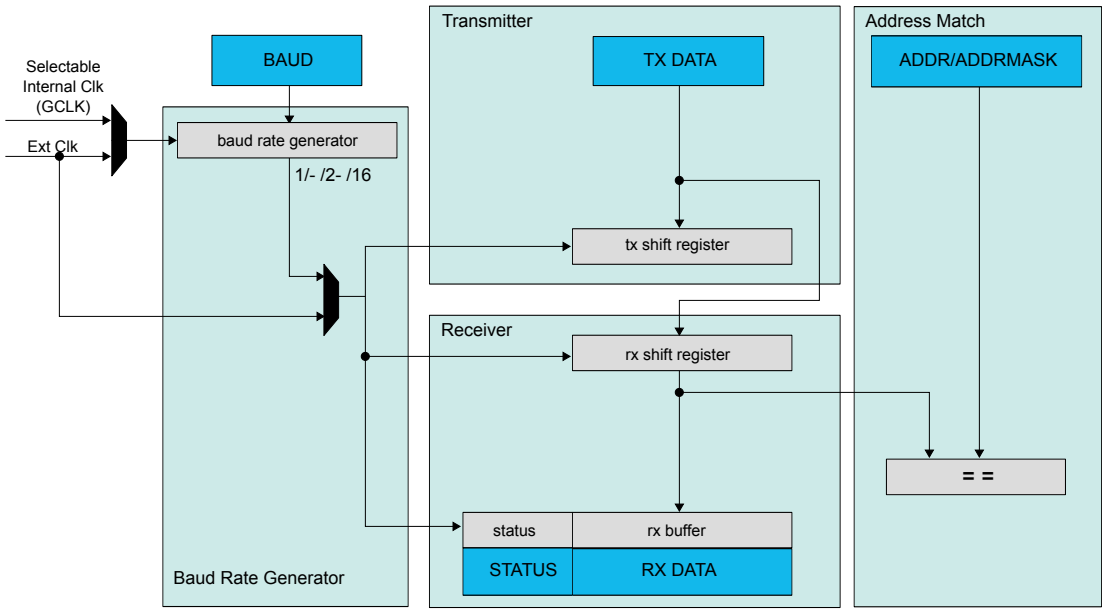
Not applicable.

31.6 Functional Description

31.6.1 Principle of Operation

The basic structure of the SERCOM serial engine is shown in Figure 31-2. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

Figure 31-2. SERCOM Serial Engine



The transmitter consists of a single write buffer and a shift register.

The receiver consists of a two-level receive buffer and a shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and I^2C operation.

31.6.2 Basic Operation

31.6.2.1 Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE). Refer to table SERCOM Modes for details.

Table 31-1. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in slave operation
0x3	SPI in master operation
0x4	I^2C slave operation
0x5	I^2C master operation
0x6-0x7	Reserved

For further initialization information, see the respective SERCOM mode chapters:

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

SERCOM SPI – SERCOM Serial Peripheral Interface

SERCOM I2C – SERCOM Inter-Integrated Circuit

31.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

31.6.2.3 Clock Generation – Baud-Rate Generator

The baud-rate generator, as shown in Figure 31-3, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

Figure 31-3. Baud Rate Generator

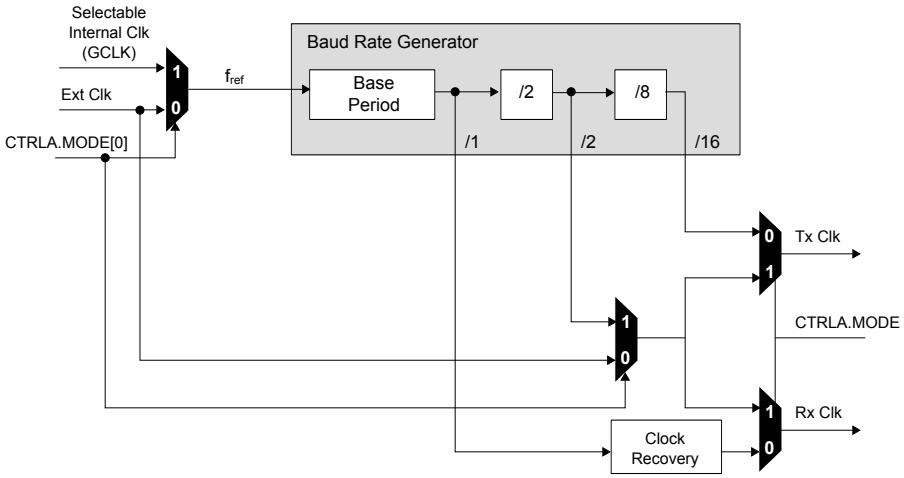


Table 31-2 contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there are two different modes: In *arithmetic mode*, the BAUD register value is 16 bits (0 to 65,535). In *fractional mode*, the BAUD register is 13 bits, while the fractional adjustment is 3 bits. In this mode the BAUD setting must be greater than or equal to 1.

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 31-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - S \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit. Can be 16, 8, or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

where

- *D* represent the data bits per frame
- *S* represent the sum of start and first stop bits, if present.

Table 31-3 shows the BAUD register value versus baud frequency f_{BAUD} at a serial engine frequency of 48MHz. This assumes a *D* value of 8 bits and an *S* value of 2 bits (10 bits, including start and stop bits).

Table 31-3. BAUD Register Value vs. Baud Frequency

BAUD Register Value	Serial Engine CPF	f_{BAUD} at 48MHz Serial Engine Frequency (f_{REF})
0 – 406	160	3MHz
407 – 808	161	2.981MHz
809 – 1205	162	2.963MHz
...	...	...
65206	31775	15.11kHz
65207	31871	15.06kHz
65208	31969	15.01kHz

31.6.3 Additional Features

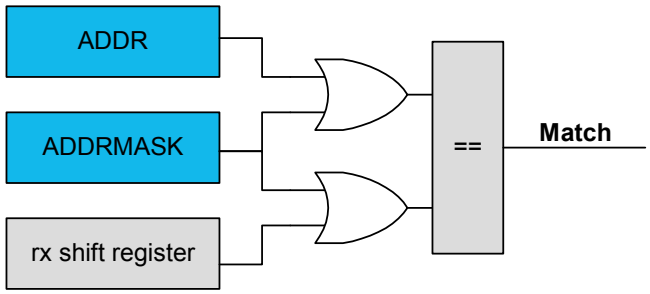
31.6.3.1 Address Match and Mask

The SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

Address With Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match. Note that writing the ADDR.ADDRMASK to 'all zeros' will match a single unique address, while writing ADDR.ADDRMASK to 'all ones' will result in all addresses being accepted.

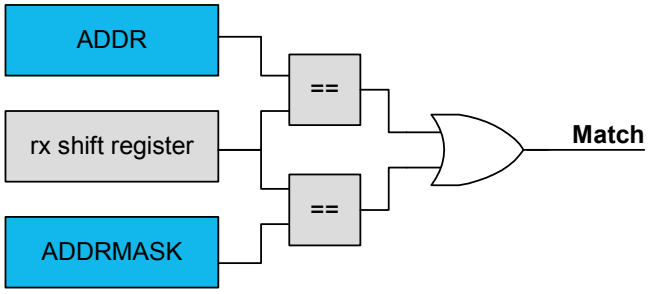
Figure 31-4. Address With Mask



Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

Figure 31-5. Two Unique Addresses



Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

Figure 31-6. Address Range



31.6.4 DMA Operation

Not applicable.

31.6.5 Interrupts

Interrupt sources are mode-specific. See the respective SERCOM mode chapters for details.

Each interrupt source has its own interrupt flag.

The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met.

Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SERCOM is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The SERCOM has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt condition occurred. The user must read the INTFLAG register to determine which interrupt condition is present.

Note:

Note that interrupts must be globally enabled for interrupt requests.

Related Links

Nested Vector Interrupt Controller

31.6.6 Events

Not applicable.

31.6.7 Sleep Mode Operation

The peripheral can operate in any sleep mode where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake up the device from sleep modes. Refer to the different SERCOM mode chapters for details.

31.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

Register Synchronization

32. SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

32.1 Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver, see Block Diagram. Labels in uppercase letters are synchronous to CLK_SERCOMx_APB and accessible for CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a single write buffer, a shift register, and control logic for different frame formats. The write buffer support data transmission without any delay between frames. The receiver consists of a two-level receive buffer and a shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

Related Links

SERCOM – Serial Communication Interface

SERCOM USART and I2C Configurations

32.2 USART Features

- Full-duplex operation
- Asynchronous (with clock reconstruction) or synchronous operation
- Internal or external clock source for asynchronous and synchronous operation
- Baud-rate generator
- Supports serial frames with 5, 6, 7, 8 or 9 data bits and 1 or 2 stop bits
- Odd or even parity generation and parity check
- Selectable LSB- or MSB-first data transfer
- Buffer overflow and frame error detection
- Noise filtering, including false start-bit detection and digital low-pass filter
- Collision detection
- Can operate in all sleep modes
- Operation at speeds up to half the system clock for internally generated clocks
- Operation at speeds up to the system clock for externally generated clocks
- RTS and CTS flow control
- IrDA modulation and demodulation up to 115.2kbps
- ISO 7816 T=0 or T=1 protocols for Smart Card interfacing
- RS485 Support
- Start-of-frame detection
- Can work with DMA

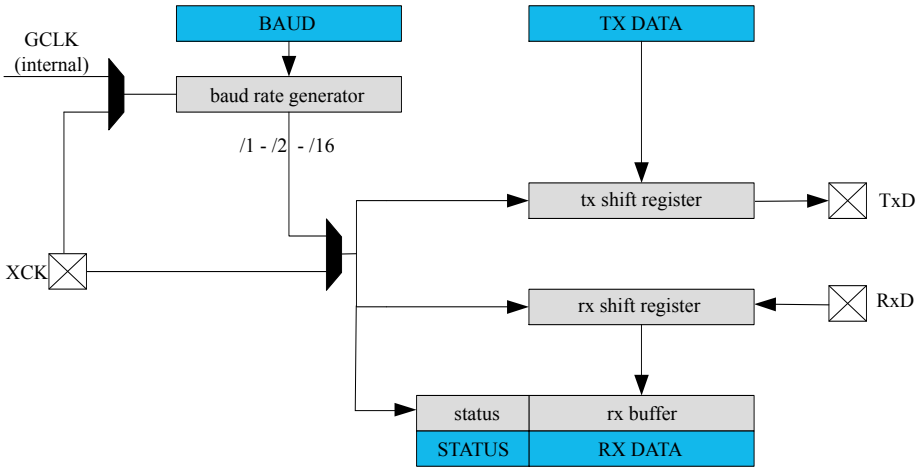
Related Links

SERCOM USART and I2C Configurations

Features

32.3 Block Diagram

Figure 32-1. USART Block Diagram



32.4 Signal Description

Table 32-1. SERCOM USART Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Related Links

I/O Multiplexing and Considerations

32.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

32.5.1 I/O Lines

Using the USART's I/O lines requires the I/O pins to be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in USART mode, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Table 32-2. USART Pin Configuration

Pin	Pin Configuration
TxD	Output
RxD	Input
XCK	Output or input

The combined configuration of PORT and the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) will define the physical position of the USART signals in Table 32-2.

Related Links

PORT: IO Pin Controller

32.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes.

Related Links

PM – Power Manager

32.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be disabled and enabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SERCOMx_CORE. This clock must be configured and enabled in the Generic Clock Controller before using the SERCOMx_CORE. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writing to certain registers will require synchronization to the clock domains. Refer to *Synchronization* for further details.

Related Links

Peripheral Clock Masking

Synchronization

GCLK - Generic Clock Controller

32.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller

32.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

32.5.6 Events

Not applicable.

32.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

32.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

32.5.9 Analog Connections

Not applicable.

32.6 Functional Description

32.6.1 Principle of Operation

The USART uses the following lines for data transfer:

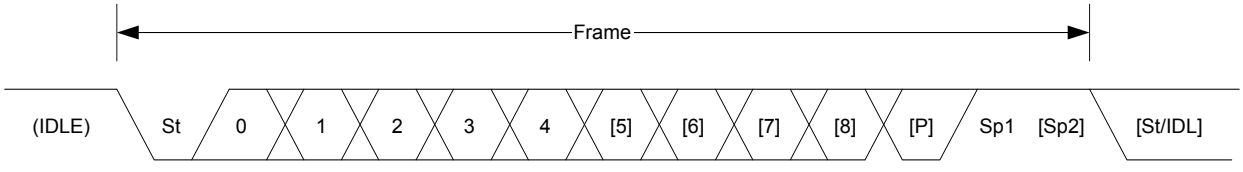
- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- No, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by one character of data bits. If enabled, the parity bit is inserted after the data bits and before the first stop bit. After the stop bit(s) of a frame, either the next frame can follow immediately, or the communication line can return to the idle (high) state. The figure below illustrates the possible frame formats. Brackets denote optional bits.

Figure 32-2. Frame Formats



St Start bit. Signal is always low.

n, [n] Data bits. 0 to [5..9]

[P] Parity bit. Either odd or even.

Sp, [Sp] Stop bit. Signal is always high.

IDLE No frame is transferred on the communication line. Signal is always high in this state.

32.6.2 Basic Operation

32.6.2.1 Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits.
- Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits.
- Baud register (BAUD)

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers will be discarded. If the peripheral is being disabled, writing to these registers will be executed after disabling is completed. Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the USART is enabled, it must be configured by these steps:

1. Select either external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either asynchronous (0) or or synchronous (1) communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).
3. Select pin for receive data by writing the Receive Data Pinout value in the CTRLA register (CTRLA.RXPO).
4. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the CTRLA register (CTRLA.TXPO).
5. Configure the Character Size field in the CTRLB register (CTRLB.CHSIZE) for character size.
6. Set the Data Order bit in the CTRLA register (CTRLA.DORD) to determine MSB- or LSB-first data transmission.
7. To use parity mode:
    7.1. Enable parity mode by writing 0x1 to the Frame Format field in the CTRLA register (CTRLA.FORM).
    7.2. Configure the Parity Mode bit in the CTRLB register (CTRLB.PMODE) for even or odd parity.
8. Configure the number of stop bits in the Stop Bit Mode bit in the CTRLB register (CTRLB.SBMODE).
9. When using an internal clock, write the Baud register (BAUD) to generate the desired baud rate.
10. Enable the transmitter and receiver by writing '1' to the Receiver Enable and Transmitter Enable bits in the CTRLB register (CTRLB.RXEN and CTRLB.TXEN).

32.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

32.6.2.3 Clock Generation and Selection

For both synchronous and asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.

The synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the asynchronous mode is selected by writing a zero to CTRLA.CMODE.

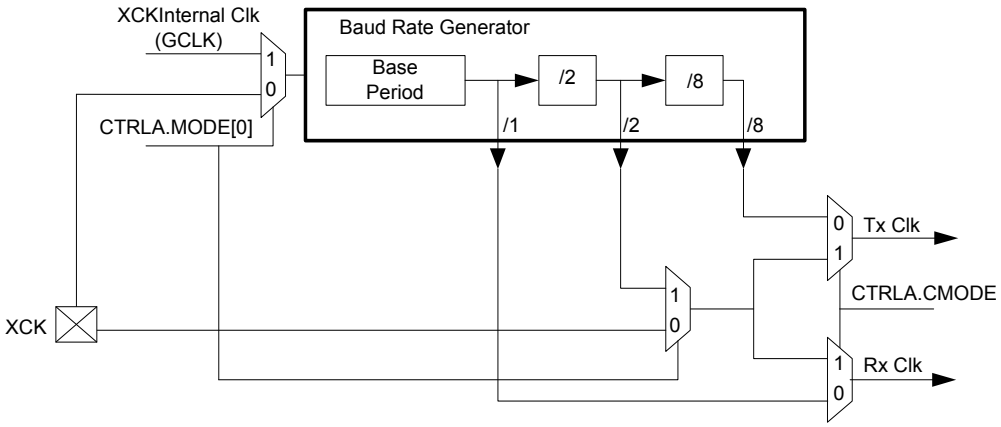
The internal clock source is selected by writing 0x1 to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing 0x0 to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the figure below.

In asynchronous mode (CTRLA.CMODE=0), the 16-bit Baud register value is used.

In synchronous mode (CTRLA.CMODE=1), the eight LSBs of the Baud register are used. Refer to *Clock Generation – Baud-Rate Generator* for details on configuring the baud rate.

Figure 32-3. Clock Generation



Related Links

Clock Generation – Baud-Rate Generator

Asynchronous Arithmetic Mode BAUD Value Selection

Synchronous Clock Operation

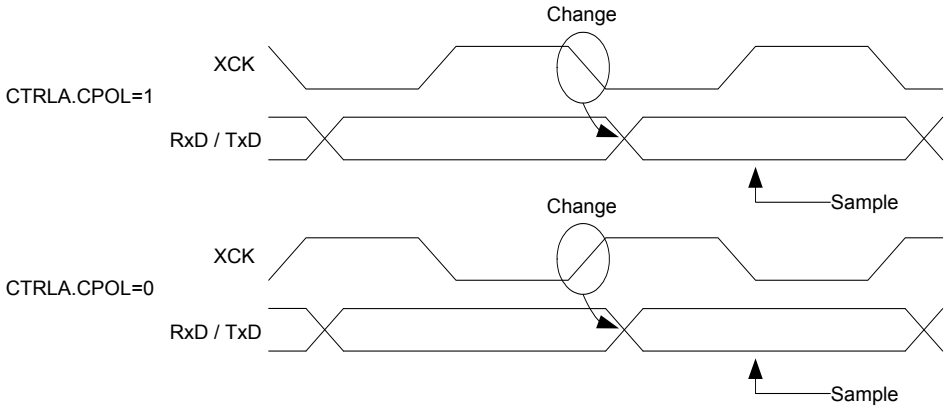
In synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling, and which is used for TxD change:

When CTRLA.CPOL is '0', the data will be changed on the rising edge of XCK, and sampled on the falling edge of XCK.

When CTRLA.CPOL is '1', the data will be changed on the falling edge of XCK, and sampled on the rising edge of XCK.

Figure 32-4. Synchronous Mode XCK Timing



When the clock is provided through XCK (CTRLA.MODE=0x0), the shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to the system frequency.

32.6.2.4 Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

32.6.2.5 Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the shift register when the shift register is empty and ready to send a new frame. After the shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including stop bit(s) has been transmitted and no new data was written to DATA, the Transmit Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt will be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter will complete only after any ongoing and pending transmissions are completed, i.e., there is no data in the transmit shift register and TxDATA to transmit.

32.6.2.6 Data Reception

The receiver accepts data when a valid start bit is detected. Each bit following the start bit will be sampled according to the baud rate or XCK clock, and shifted into the receive shift register until the first stop bit of a frame is received. The second stop bit will be ignored by the receiver.

When the first stop bit is received and a complete serial frame is present in the receive shift register, the contents of the shift register will be moved into the two-level receive buffer. Then, the Receive Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set, and the optional interrupt will be generated.

The received data can be read from the DATA register when the Receive Complete interrupt flag is set.

Disabling the Receiver
Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) will disable the receiver, flush the two-level receive buffer, and data from ongoing receptions will be lost.

Error Bits
The USART receiver has three error bits in the Status (STATUS) register: Frame Error (FERR), Buffer Overflow (BUFOVF), and Parity Error (PERR). Once an error happens, the corresponding error bit will be set until it is cleared by writing ‘1’ to it. These bits are also cleared automatically when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

When CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA, until the receiver complete interrupt flag (INTFLAG.RXC) is cleared.

When CTRLA.IBON=0, the buffer overflow condition is attending data through the receive FIFO. After the received data is read, STATUS.BUFOVF will be set along with INTFLAG.RXC.

Asynchronous Data Reception
The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock.

The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

Asynchronous Operational Range
The operational range of the asynchronous reception depends on the accuracy of the internal baud-rate clock, the rate of the incoming frames, and the frame size (in number of bits). In addition, the operational range of the receiver is depending on the difference between the received bit rate and the internally generated baud rate. If the baud rate of an external transmitter is too high or too low compared to the internally generated baud rate, the receiver will not be able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in baud rate: First, the reference clock will always have some minor instability. Second, the baud-rate generator cannot always do an exact division of the reference clock frequency to get the baud rate desired. In this case, the BAUD register value should be set to give the lowest possible error. Refer to *Clock Generation – Baud-Rate Generator* for details.

Recommended maximum receiver baud-rate errors for various character sizes are shown in the table below.

Table 32-3. Asynchronous Receiver Error for 16-fold Oversampling

D (Data bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. total error [%]	Recommended max. Rx error [%]
5	94.12	107.69	+5.88/-7.69	±2.5
6	94.92	106.67	+5.08/-6.67	±2.0
7	95.52	105.88	+4.48/-5.88	±2.0
8	96.00	105.26	+4.00/-5.26	±2.0
9	96.39	104.76	+3.61/-4.76	±1.5
10	96.70	104.35	+3.30/-4.35	±1.5

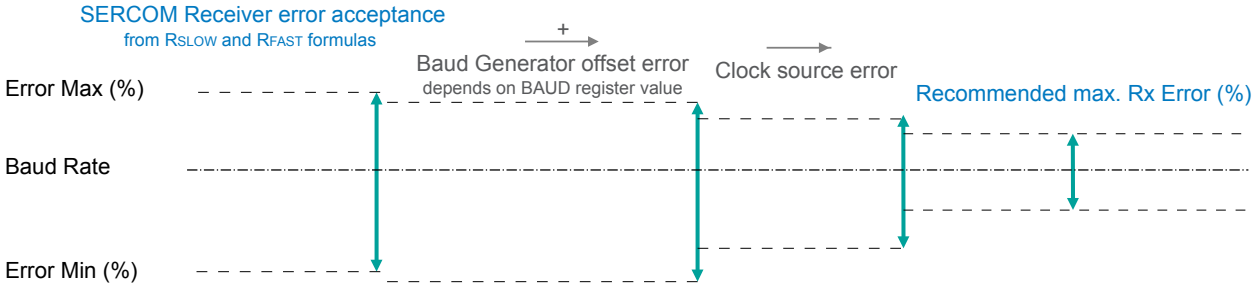
The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{SLOW} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \quad R_{FAST} = \frac{(D+2)S}{(D+1)S+S_M}$$

- R_{SLOW} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate
- R_{FAST} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate
- D is the sum of character size and parity size (D = 5 to 10 bits)
- S is the number of samples per bit (S = 16, 8 or 3)
- S_F is the first sample number used for majority voting (S_F = 7, 3, or 2) when CTRLA.SAMPA=0.
- S_M is the middle sample number used for majority voting (S_M = 8, 4, or 2) when CTRLA.SAMPA=0.

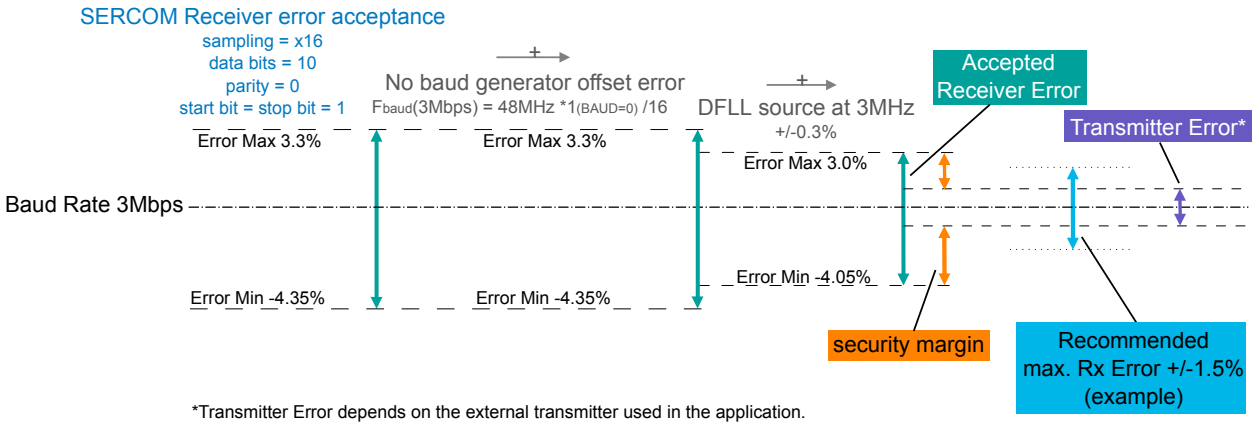
The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

Figure 32-5. USART Rx Error Calculation



The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure gives an example for a baud rate of 3Mbps:

Figure 32-6. USART Rx Error Calculation Example



*Transmitter Error depends on the external transmitter used in the application.
It is advised that it is within the Recommended max. Rx Error (+/-1.5% in this example).
Larger Transmitter Errors are acceptable but must lie within the Accepted Receiver Error.

Related Links

Clock Generation – Baud-Rate Generator

Asynchronous Arithmetic Mode BAUD Value Selection

32.6.3 Additional Features

32.6.3.1 Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

If *even parity* is selected (CTRLB.PMODE=0), the parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

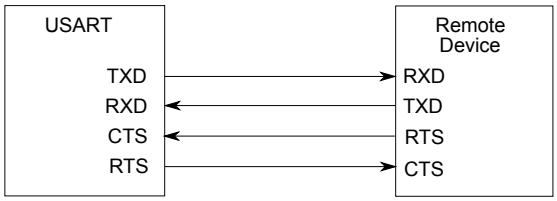
If *odd parity* is selected (CTRLB.PMODE=1), the parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

32.6.3.2 Hardware Handshaking

The USART features an out-of-band hardware handshaking flow control mechanism, implemented by connecting the RTS and CTS pins with the remote device, as shown in the figure below.

Figure 32-7. Connection with a Remote Device for Hardware Handshaking

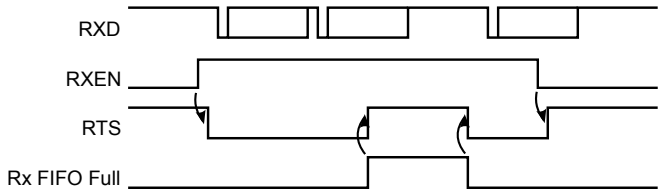


Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and Flow control pinout (CTRLA.TXPO=2).

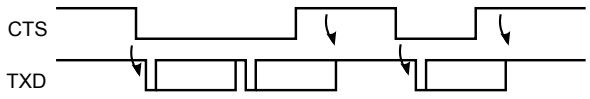
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS will be set immediately and the frame being received will be stored in the shift register until the receive FIFO is no longer full.

Figure 32-8. Receiver Behavior when Operating with Hardware Handshaking



The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission will start only if STATUS.CTS=0. When CTS is set, the transmitter will complete the ongoing transmission and stop transmitting.

Figure 32-9. Transmitter Behavior when Operating with Hardware Handshaking



32.6.3.3 IrDA Modulation and Demodulation

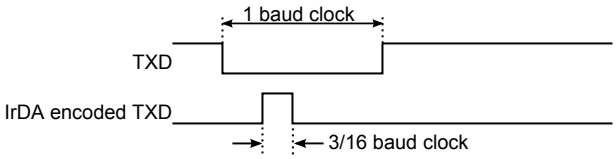
Transmission and reception can be encoded IrDA compliant up to 115.2 kb/s. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC=1),
- Asynchronous mode (CTRLA.CMODE=0),

- and 16x sample rate (CTRLA.SAMPR[0]=0).

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the figure below.

Figure 32-10. IrDA Transmit Encoding



The reception decoder has two main functions.

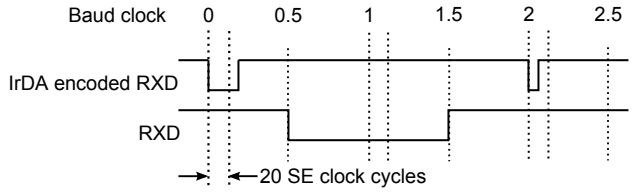
The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.

The second main function is to decode incoming Rx data. If a pulse width meets the minimum length set by configuration (RXPL.RXPL), it is accepted. When the baud rate counter reaches its middle value (1/2 bit length), it is transferred to the receiver.

Note: Note that the polarity of the transmitter and receiver are opposite: During transmission, a '0' bit is transmitted as a '1' pulse. During reception, an accepted '0' pulse is received as a '0' bit.

Example: The figure below illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width should be at least 20 SE clock cycles. When using BAUD=0xE666 or 160 SE cycles per bit, this corresponds to 2/16 baud clock as minimum pulse width required. In this case the first bit is accepted as a '0', the second bit is a '1', and the third bit is also a '1'. A low pulse is rejected since it does not meet the minimum requirement of 2/16 baud clock.

Figure 32-11. IrDA Receive Decoding



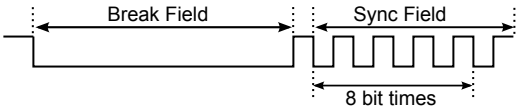
32.6.3.4 Break Character Detection and Auto-Baud

Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05),
- Asynchronous mode (CTRLA.CMODE = 0),
- and 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1).

The auto-baud follows the LIN format. All LIN Frames start with a Break Field followed by a Sync Field. The USART uses a break detection threshold of greater than 11 nominal bit times at the configured baud rate. At any time, if more than 11 consecutive dominant bits are detected on the bus, the USART detects a Break Field. When a Break Field has been detected, the Receive Break interrupt flag (INTFLAG.RXBRK) is set and the USART expects the Sync Field character to be 0x55. This field is used to update the actual baud rate in order to stay synchronized. If the received Sync character is not 0x55, then the Inconsistent Sync Field error flag (STATUS.ISF) is set along with the Error interrupt flag (INTFLAG.ERROR), and the baud rate is unchanged.

Figure 32-12. LIN Break and Sync Fields



After a break field is detected and the start bit of the Sync Field is detected, a counter is started. The counter is then incremented for the next 8 bit times of the Sync Field. At the end of these 8 bit times, the counter is stopped. At this moment, the 13 most significant bits of the counter (value divided by 8) give the new clock divider (BAUD.BAUD), and the 3 least significant bits of this value (the remainder) give the new Fractional Part (BAUD.FP).

When the Sync Field has been received, the clock divider (BAUD.BAUD) and the Fractional Part (BAUD.FP) are updated after a synchronization delay. After the Break and Sync Fields are received, multiple characters of data can be received.

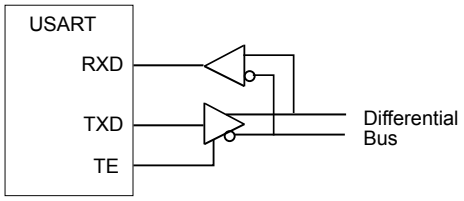
32.6.3.5 RS485

RS485 is available with the following configuration:

- USART frame format (CTRLA.FORM = 0x00 or 0x01)
- RS485 pinout (CTRLA.TXPO=0x3).

The RS485 feature enables control of an external line driver as shown in the figure below. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

Figure 32-13. RS485 Bus Connection



The TE pin will remain high for the complete frame including stop bit(s). If a Guard Time is programmed in the Control C register (CTRLC.GTIME), the line will remain driven after the last character completion. The following figure shows a transfer with one stop bit and CTRLC.GTIME=3.

Figure 32-14. Example of TE Drive with Guard Time



The Transmit Complete interrupt flag (INTFLAG.TXC) will be raised after the guard time is complete and TE goes low.

32.6.3.6 ISO 7816 for Smart Card Interfacing

The SERCOM USART features an ISO/IEC 7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO 7816 link. Both T=0 and T=1 protocols defined by the ISO 7816 specification are supported.

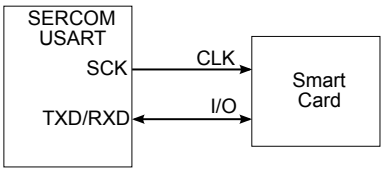
ISO 7816 is available with the following configuration:

- ISO 7816 format (CTRLA.FORM = 0x07)
- Inverse transmission and reception (CTRLA.RXINV=1 and CTRLA.TXINV=1)
- Single bidirectional data line (CTRLA.TXPO and CTRLA.RXPO configured to use the same data pin)
- Even parity (CTRLB.PMODE=0)

- 8-bit character size (CTRLB.CHSIZE=0)
- T=0 (CTRLA.CMODE=1) or T=1 (CTRLA.CMODE=0)

ISO 7816 is a half duplex communication on a single bidirectional line. The USART connects to a smart card as shown below. The output is only driven when the USART is transmitting. The USART is considered as the master of the communication as it generates the clock.

Figure 32-15. Connection of a Smart Card to the SERCOM USART



ISO 7816 characters are specified as 8 bits with even parity. The USART must be configured accordingly.

The USART cannot operate concurrently in both receiver and transmitter modes as the communication is unidirectional. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO 7816 mode may lead to unpredictable results.

The ISO 7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value (CTRLA.RXINV=1 and CTRLA.TXINV=1).

Protocol T=0

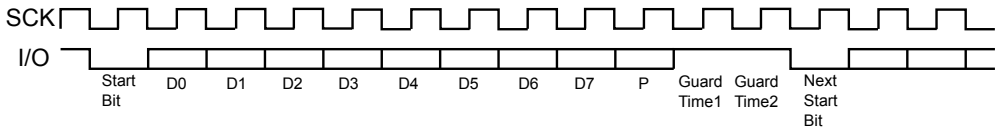
In T=0 protocol, a character is made up of:

- one start bit,
- eight data bits,
- one parity bit
- and one guard time, which lasts two bit times.

The transfer is synchronous (CTRLA.CMODE=1). The transmitter shifts out the bits and does not drive the I/O line during the guard time. Additional guard time can be added by programming the Guard Time (CTRLC.GTIME).

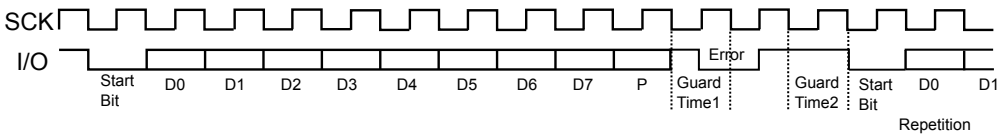
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the figure below.

Figure 32-16. T=0 Protocol without Parity Error



If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in the next figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time, which lasts 1 bit time.

Figure 32-17. T=0 Protocol with Parity Error



When the USART is the receiver and it detects a parity error, the parity error bit in the Status Register (STATUS.PERR) is set and the character is not written to the receive FIFO.

Receive Error Counter

The receiver also records the total number of errors (receiver parity errors and NACKs from the remote transmitter) up to a maximum of 255. This can be read in the Receive Error Count (RXERRCNT) register. RXERRCNT is automatically cleared on read.

Receive NACK Inhibit

The receiver can also be configured to inhibit error generation. This can be achieved by setting the Inhibit Not Acknowledge (CTRLC.INACK) bit. If CTRLC.INACK is 1, no error signal is driven on the I/O line even if a parity error is detected. Moreover, if CTRLC.INACK is set, the erroneous received character is stored in the receive FIFO, and the STATUS.PERR bit is set. Inhibit not acknowledge (CTRLC.INACK) takes priority over disable successive receive NACK (CTRLC.DSNACK).

Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next character. Repetition is enabled by writing the Maximum Iterations register (CTRLC.MAXITER) to a non-zero value. The USART repeats the character the number of times specified in CTRLC.MAXITER.

When the USART repetition number reaches the programmed value in CTRLC.MAXITER, the STATUS.ITER bit is set and the internal iteration counter is reset. If the repetition of the character is acknowledged by the receiver before the maximum iteration is reached, the repetitions are stopped and the iteration counter is cleared.

Disable Successive Receive NACK

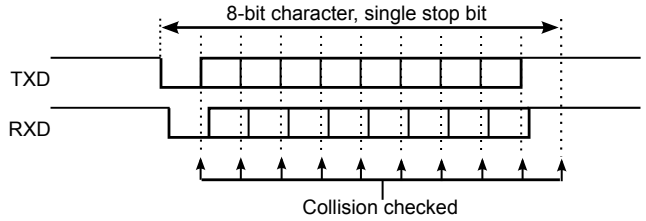
The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the Disable Successive NACK bit (CTRLC.DSNACK). The maximum number of NACKs transmitted is programmed in the CTRLC.MAXITER field. As soon as the maximum is reached, the character is considered as correct, an acknowledge is sent on the line, the STATUS.ITER bit is set and the internal iteration counter is reset.

32.6.3.7 Collision Detection

When the receiver and transmitter are connected either through pin configuration or externally, transmit collision can be detected after selecting the Collision Detection Enable bit in the CTRLB register (CTRLB.COLDEN=1). To detect collision, the receiver and transmitter must be enabled (CTRLB.RXEN=1 and CTRLB.TXEN=1).

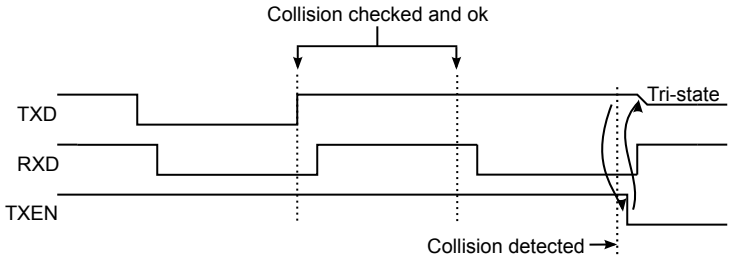
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as shown in the figure below. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 32-18. Collision Checking



The next figure shows the conditions for a collision detection. In this case, the start bit and the first data bit are received with the same value as transmitted. The second received data bit is found to be different than the transmitted bit at the detection point, which indicates a collision.

Figure 32-19. Collision Detected



When a collision is detected, the USART follows this sequence:

1. Abort the current transfer.
2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN=0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) will be set until this is complete.
   - After disabling, the TxD pin will be tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error interrupt flag (INTFLAG.ERROR).
5. Set the Transmit Complete interrupt flag (INTFLAG.TXC), since the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after assuring that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

32.6.3.8 Loop-Back Mode

For loop-back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

32.6.3.9 Start-of-Frame Detection

The USART start-of-frame detector can wake up the CPU when it detects a start bit. In standby sleep mode, the internal fast startup oscillator must be selected as the GCLK_SERCOMx_CORE source.

When a 1-to-0 transition is detected on RxD, the 8MHz Internal Oscillator is powered up and the USART clock is enabled. After startup, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast startup internal oscillator start-up time. Refer to *Electrical Characteristics* for details. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in asynchronous and synchronous modes. It is enabled by writing ‘1’ to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8MHz Internal Oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the Receive Complete interrupt is generated.

Related Links

Electrical Characteristics

32.6.3.10 Sample Adjustment

In asynchronous mode (CTRLA.CMODE=0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in Control A register (CTRLA.SAMPA). When CTRLA.SAMPA=0, samples 7-8-9 are used for 16x oversampling, and samples 3-4-5 are used for 8x oversampling.

32.6.4 DMA, Interrupts and Events

Table 32-4. Module Request for SERCOM USART

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Receive Start (RXS)	NA	Yes	
Clear to Send Input Change (CTSIC)	NA	Yes	
Receive Break (RXBRK)	NA	Yes	
Error (ERROR)	NA	Yes	

32.6.4.1 DMA Operation

The USART generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

32.6.4.2 Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is

disabled, or the USART is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The USART has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

32.6.4.3 Events

Not applicable.

32.6.5 Sleep Mode Operation

The behavior in sleep mode is depending on the clock source and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Internal clocking, CTRLA.RUNSTDBY=1: GCLK_SERCOMx_CORE can be enabled in all sleep modes. Any interrupt can wake up the device.
- External clocking, CTRLA.RUNSTDBY=1: The Receive Complete interrupt(s) can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY=0: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Complete interrupt(s) can wake up the device.
- External clocking, CTRLA.RUNSTDBY=0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

32.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also CTRLB for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

32.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8	SAMPR[2:0]					RXINV	TXINV	IBON
0x02		23:16	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
0x03		31:24		DORD	CPOL	CMODE	FORM[3:0]			
0x04	CTRLB	7:0		SBMODE				CHSIZE[2:0]		
0x05		15:8			PMODE			ENC	SFDE	COLDEN
0x06		23:16							RXEN	TXEN
0x07		31:24								
0x08	CTRLC	7:0						GTIME[2:0]		
0x09		15:8								
0x0A		23:16		MAXITER[2:0]					DSNACK	INACK
0x0B		31:24								
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D		15:8	BAUD[15:8]							
0x0E	RXPL	7:0	RXPL[7:0]							
0x0F ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0		ITER	COLL	ISF	CTS	BUFOVF	FERR	PERR
0x1B		15:8								
0x1C	SYNCBUSY	7:0					RXERRCNT	CTRLB	ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x20	RXERRCNT	7:0	RXERRCNT[7:0]							
0x21 ... 0x27	Reserved									
0x28	DATA	7:0	DATA[7:0]							
0x29		15:8								DATA[8:8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

32.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

32.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000fa7]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]					RXINV	TXINV	IBON
Access	R/W	R/W	R/W			R/W	R/W	R
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD: Data Order
This bit selects the data order when a character is shifted out from the Data register.

This bit is not synchronized.

Value	Description
0	MSB is transmitted first.
1	LSB is transmitted first.

Bit 29 – CPOL: Clock Polarity
This bit selects the relationship between data output change and data input sampling in synchronous mode.

This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE: Communication Mode
This bit selects asynchronous or synchronous communication.

This bit is not synchronized.

Value	Description
0	Asynchronous communication.
1	Synchronous communication.

Bits 27:24 – FORM[3:0]: Frame Format
These bits define the frame format.

These bits are not synchronized.

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity
0x2-0x3	Reserved
0x4	Auto-baud - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6	Reserved
0x7	ISO 7816
0x8-0xF	Reserved

Bits 23:22 – SAMPA[1:0]: Sample Adjustment
These bits define the sample adjustment.

These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9	3-4-5
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0]: Receive Data Pinout
These bits define the receive data (RxD) pin configuration.

These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Bits 17:16 – TXPO[1:0]: Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.

This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	N/A	N/A
0x1	SERCOM PAD[2]	SERCOM PAD[3]	N/A	N/A
0x2	SERCOM PAD[0]	N/A	SERCOM PAD[2]	SERCOM PAD[3]
0x3	SERCOM_PAD[0]	SERCOM_PAD[1]	SERCOM_PAD[2]	N/A

Bits 15:13 – SAMPR[2:0]: Sample Rate
These bits select the sample rate.

These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

Bit 10 – RXINV: Receive Data Invert
This bit controls whether the receive data (RxD) is inverted or not.
Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

Value	Description
0	RxD is not inverted.
1	RxD is inverted.

Bit 9 – TXINV: Transmit Data Invert
This bit controls whether the transmit data (TxD) is inverted or not.
Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

Value	Description
0	TxD is not inverted.
1	TxD is inverted.

Bit 8 – IBON: Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY: Run In Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device can wake up on Transfer Complete interrupt.
0x1	Wake on Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits select the USART serial communication interface of the SERCOM.

These bits are not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

32.8.2 Control B

Name: CTRLB
Offset: 0x04 [ID-00000fa7]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
			PMODE			ENC	SFDE	COLDEN
Access			R/W			R/W	R/W	R/W
Reset			0			0	0	0

Bit	7	6	5	4	3	2	1	0
		SBMODE				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN: Receiver Enable
Writing '0' to this bit will disable the USART receiver. Disabling the receiver will flush the receive buffer and clear the FERR, PERR and BUFOVF bits in the STATUS register.

Writing '1' to CTRLB.RXEN when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN will read back as '1'.

Writing '1' to CTRLB.RXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or will be enabled when the USART is enabled.

Bit 16 – TXEN: Transmitter Enable
Writing '0' to this bit will disable the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.

Writing '1' to CTRLB.TXEN when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN will read back as '1'.

Writing '1' to CTRLB.TXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.TXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or will be enabled when the USART is enabled.

Bit 13 – PMODE: Parity Mode
This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.

This bit is not synchronized.

Value	Description
0	Even parity.
1	Odd parity.

Bit 10 – ENC: Encoding Format
This bit selects the data encoding format.

This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

Bit 9 – SFDE: Start of Frame Detection Enable
This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.

This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved

SFDE	INTENSET.RXS	INTENSET.RXC	Description
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes.

Bit 8 – COLDEN: Collision Detection Enable
This bit enables collision detection.

This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE: Stop Bit Mode
This bit selects the number of stop bits transmitted.

This bit is not synchronized.

Value	Description
0	One stop bit.
1	Two stop bits.

Bits 2:0 – CHSIZE[2:0]: Character Size
These bits select the number of bits in a character.

These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

32.8.3 Control C

Name: CTRLC
Offset: 0x08 [ID-00000fa7]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		MAXITER[2:0]					DSNACK	INACK
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						GTIME[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 22:20 – MAXITER[2:0]: Maximum Iterations
These bits define the maximum number of retransmit iterations.

These bits also define the successive NACKs sent to the remote transmitter when CTRLC.DSNACK is set.

This field is only valid when using ISO7816 T=0 mode (CTRLA.MODE=0x7 and CTRLA.CMODE=0).

Value	Description
0	NACK is sent on the ISO line for every parity error received.
1	Successive parity errors are counted up to the value specified in CTRLC.MAXITER. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line.

Bit 17 – DSNACK: Disable Successive Not Acknowledge
This bit controls how many times NACK will be sent on parity error reception.

This bit is only valid in ISO7816 T=0 mode and when CTRLC.INACK=0.

Value	Description
0	NACK is sent on the ISO line for every parity error received.
1	Successive parity errors are counted up to the value specified in CTRLC.MAXITER. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line.

Bit 16 – INACK: Inhibit Not Acknowledge
This bit controls whether a NACK is transmitted when a parity error is received.

This bit is only valid in ISO7816 T=0 mode.

Value	Description
0	NACK is transmitted when a parity error is received.
1	NACK is not transmitted when a parity error is received.

Bits 2:0 – GTIME[2:0]: Guard Time
These bits define the guard time when using RS485 mode (CTRLA.TXPO=0x3) or ISO7816 mode (CTRLA.MODE=0x7).ISO7816 mode (CTRLA.TXPO=0x7).RS485 mode (CTRLA.TXPO=0x3).

For RS485 mode, the guard time is programmable from 0-7 bit times and defines the time that the transmit enable pin (TE) remains high after the last stop bit is transmitted and there is no remaining data to be transmitted.

For ISO7816 T=0 mode, the guard time is programmable from 2-9 bit times and defines the guard time between each transmitted byte.

32.8.4 Baud

Name: BAUD
Offset: 0x0C [ID-00000fa7]
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	BAUD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BAUD[15:0]: Baud Value
Arithmetic Baud Rate Generation (`CTRLA.SAMPR[0]=0`):

These bits control the clock generation, as described in the SERCOM Baud Rate section.

If Fractional Baud Rate Generation (`CTRLA.SAMPR[0]=1`) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.
- **Bits 12:0 - BAUD[21:0]: Baud Value**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.

32.8.5 Receive Pulse Length Register

Name: RXPL
Offset: 0x0E [ID-00000fa7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	RXPL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXPL[7:0]: Receive Pulse Length
When the encoding format is set to IrDA (CTRLB.ENC=1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per}.

$$PULSE \geq \left(\mathrm{RXPL} + 2\right) \cdot SE_{per}$$

32.8.6 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x14 [ID-00000fa7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK: Receive Break Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC: Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS: Receive Start Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start Interrupt Enable bit, which disables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

32.8.7 Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x16 [ID-00000fa7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK: Receive Break Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC: Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS: Receive Start Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

© 2017 Microchip Technology Inc.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

32.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18 [ID-00000fa7]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R	R/W	R
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR: Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. Errors that will set this flag are COLL, ISF, BUFOVF, FERR, and PERR.Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 5 – RXBRK: Receive Break
This flag is cleared by writing '1' to it.

This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 4 – CTSIC: Clear to Send Input Change
This flag is cleared by writing a '1' to it.

This flag is set when a change is detected on the CTS pin.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – RXS: Receive Start
This flag is cleared by writing '1' to it.

This flag is set when a start condition is detected on the RxD line and start-of-frame detection is enabled (CTRLB.SFDE is '1').

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start interrupt flag.

Bit 2 – RXC: Receive Complete
This flag is cleared by reading the Data register (DATA) or by disabling the receiver.

This flag is set when there are unread data in DATA.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 1 – TXC: Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.

This flag is set when the entire frame in the transmit shift register has been shifted out and there are no new data in DATA.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 0 – DRE: Data Register Empty
This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready to be written.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

32.8.9 Status

Name: STATUS
Offset: 0x1A [ID-00000fa7]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		ITER	COLL	ISF	CTS	BUFOVF	FERR	PERR
Access		R/W	R/W	R/W	R	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – ITER: Maximum Number of Repetitions Reached
This bit is set when the maximum number of NACK repetitions or retransmissions is met in ISO7816 T=0 mode.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 5 – COLL: Collision Detected
This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 4 – ISF: Inconsistent Sync Field
This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when the frame format is set to auto-baud (CTRLA.FORM) and a sync field not equal to 0x55 is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 3 – CTS: Clear to Send
This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 2 – BUFOVF: Buffer Overflow
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 1 – FERR: Frame Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set if the received character had a frame error, i.e., when the first stop bit is zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 0 – PERR: Parity Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5, or 0x7) and a parity error is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

32.8.10 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C [ID-00000fa7]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RXERRCNT	CTRLB	ENABLE	SWRST
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – RXERRCNT: Receive Error Count Synchronization Busy
The RXERRCNT register is automatically synchronized to the APB domain upon error. When returning from sleep, this bit will be raised until the new value is available to be read.

Value	Description
0	RXERRCNT synchronization is not busy.
1	RXERRCNT synchronization is busy.

Bit 2 – CTRLB: CTRLB Synchronization Busy
Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB bit will be set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST: Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

32.8.11 Receive Error Count

Name: RXERRCNT
Offset: 0x20 [ID-00000fa7]
Reset: 0x00
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	RXERRCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXERRCNT[7:0]: Receive Error Count
This register records the total number of parity errors and NACK errors combined in ISO7816 mode (CTRLA.FORM=0x7).

This register is automatically cleared on read.

32.8.12 Data

Name: DATA
Offset: 0x28 [ID-00000fa7]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
								DATA[8:8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – DATA[8:0]: Data
Reading these bits will return the contents of the Receive Data register. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS should be read before reading the DATA value in order to get any corresponding error.

Writing these bits will write the Transmit Data register. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

32.8.13 Debug Control

Name: DBGCTRL
Offset: 0x30 [ID-00000fa7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP: Debug Stop Mode
This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

33. SERCOM SPI – SERCOM Serial Peripheral Interface

33.1 Overview

The serial peripheral interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as shown in Block Diagram. Each side, master and slave, depicts a separate SPI containing a shift register, a transmit buffer and two receive buffers. In addition, the SPI master uses the SERCOM baud-rate generator, while the SPI slave can use the SERCOM address match logic. Labels in capital letters are synchronous to CLK_SERCOMx_APB and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

Related Links

SERCOM – Serial Communication Interface

33.2 Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (MISO, MOSI, SCK, $\overline{SS}$)
- Single-buffered transmitter, double-buffered receiver
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on MISO or MOSI pin
- Selectable LSB- or MSB-first data transfer
- Can be used with DMA
- Master operation:
  - Serial clock speed, $f_{SCK}=1/t_{SCK}$(1)
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
- Slave operation:
  - Serial clock speed, $f_{SCK}=1/t_{SSCK}$(1)
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

1. For t_{SCK} and t_{SSCK} values, refer to SPI Timing Characteristics.

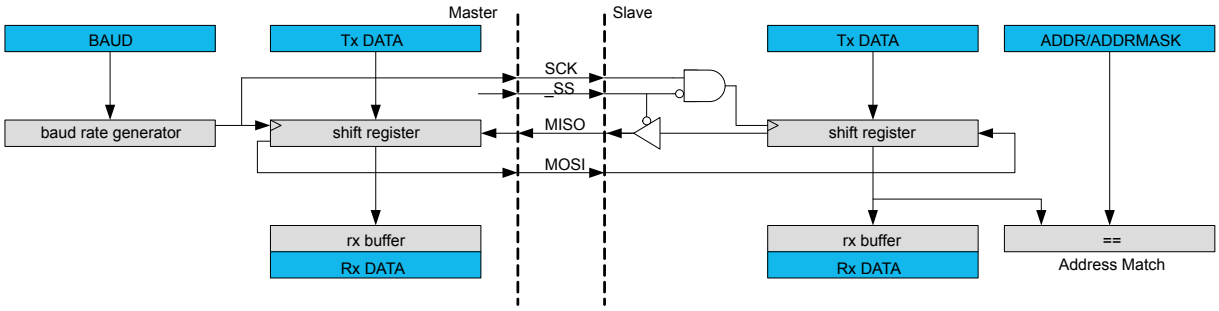
Related Links

SERCOM – Serial Communication Interface

Features

33.3 Block Diagram

Figure 33-1. Full-Duplex SPI Master Slave Interconnection



33.4 Signal Description

Table 33-1. SERCOM SPI Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Related Links

I/O Multiplexing and Considerations

33.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

33.5.1 I/O Lines

In order to use the SERCOM's I/O lines, the I/O pins must be configured using the IO Pin Controller (PORT).

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver is disabled, the data input pin can be used for other purposes. In master mode, the slave select line ($\overline{SS}$) is hardware controlled when the Master Slave Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

Table 33-2. SPI Pin Configuration

Pin	Master SPI	Slave SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
$\overline{SS}$	Output (CTRLB.MSSEN=1)	Input

The combined configuration of PORT, the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals in the table above.

Related Links

PORT: IO Pin Controller

33.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes.

Related Links

PM – Power Manager

33.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Generic Clock Controller before using the SPI.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writes to certain registers will require synchronization to the clock domains.

Related Links

GCLK - Generic Clock Controller

Peripheral Clock Masking

Synchronization

33.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller

33.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

33.5.6 Events

Not applicable.

33.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

33.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

33.5.9 Analog Connections

Not applicable.

33.6 Functional Description

33.6.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface It allows high-speed communication between the device and peripheral devices.

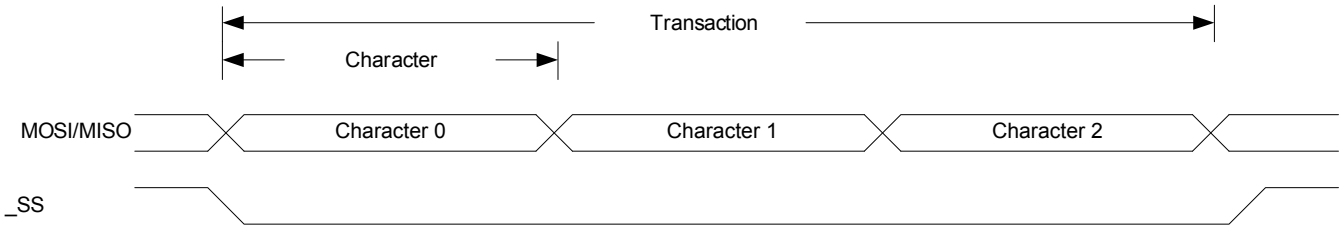
The SPI can operate as master or slave. As master, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

Figure 33-2. SPI Transaction Format



The SPI master must pull the slave select line ($\overline{SS}$) of the desired slave low to initiate a transaction. The master and slave prepare data to send via their respective shift registers, and the master generates the serial clock on the SCK line.

Data are always shifted from master to slave on the Master Output Slave Input line (MOSI); data is shifted from slave to master on the Master Input Slave Output line (MISO).

Each time character is shifted out from the master, a character will be shifted out from the slave simultaneously. To signal the end of a transaction, the master will pull the $\overline{SS}$ line high

33.6.2 Basic Operation

33.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB), except Receiver Enable (CTRLB.RXEN)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE=1), any writing to these registers will be discarded.

when the SPI is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in master / slave operation in the Operating Mode bit group in the CTRLA register (CTRLA.MODE= 0x2 or 0x3).
2. Select transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLA register (CTRLA.CPOL and CTRLA.CPHA) if desired.
3. Select the Frame Format value in the CTRLA register (CTRLA.FORM).
4. Configure the Data In Pinout field in the Control A register (CTRLA.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control A register (CTRLA.DOPO) for SERCOM pads of the transmitter.
6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).
7. Write the Data Order bit in the CTRLA register (CTRLA.DORD) for data direction.
8. If the SPI is used in master mode:
    8.1. Select the desired baud rate by writing to the Baud register (BAUD).
    8.2. If Hardware SS control is required, write '1' to the Master Slave Select Enable bit in CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN=1).

33.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

33.6.2.3 Clock Generation

In SPI master operation (CTRLA.MODE=0x3), the serial clock (SCK) is generated internally by the SERCOM baud-rate generator.

In SPI mode, the baud-rate generator is set to synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the shift register. Refer to *Clock Generation – Baud-Rate Generator* for more details.

In SPI slave operation (CTRLA.MODE is 0x2), the clock is provided by an external master on the SCK pin. This clock is used to directly clock the SPI shift register.

Related Links

Clock Generation – Baud-Rate Generator

Asynchronous Arithmetic Mode BAUD Value Selection

33.6.2.4 Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

33.6.2.5 SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI data transfer modes are shown in SPI Transfer Modes (Table) and SPI Transfer Modes (Figure).

SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

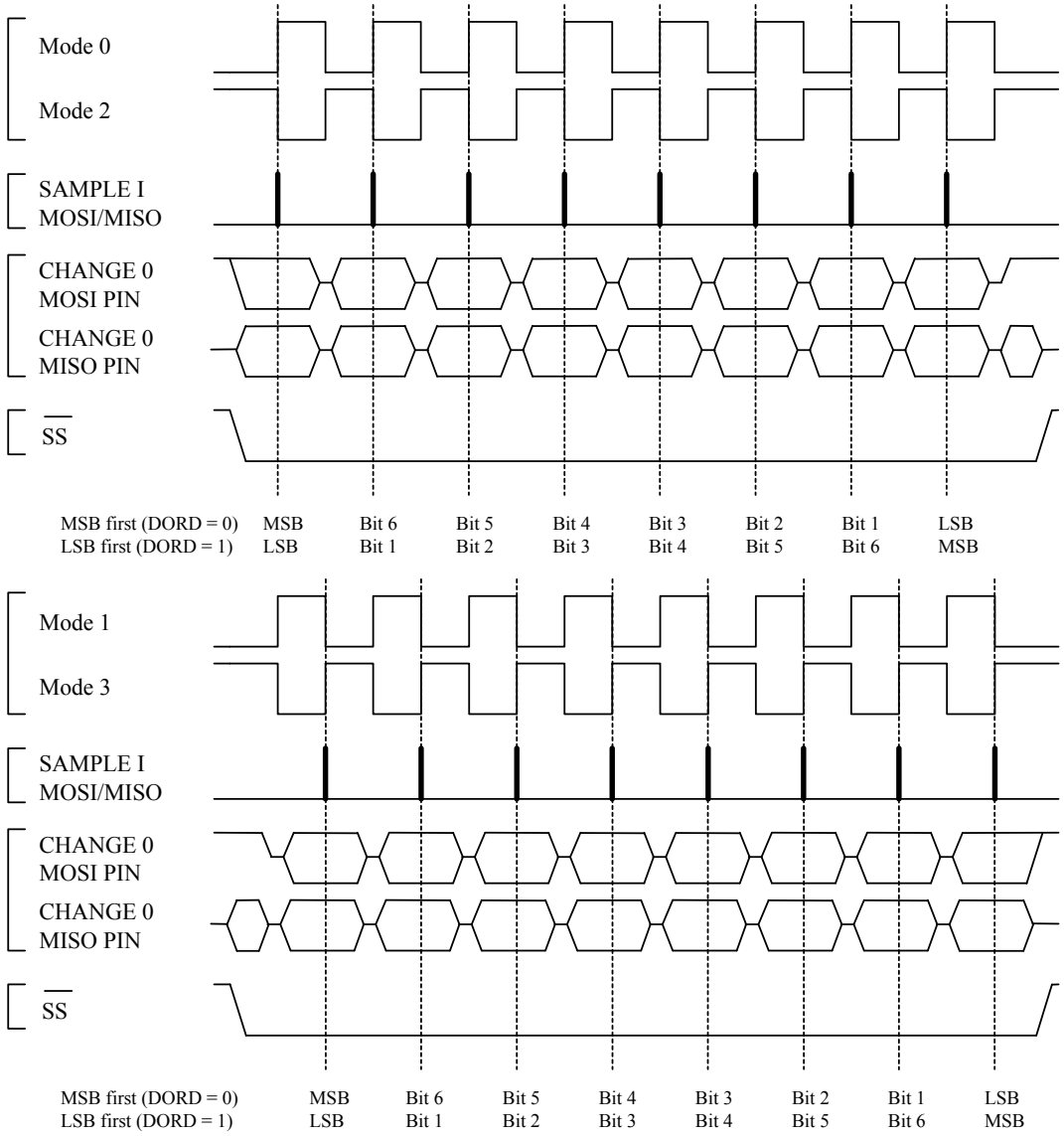
Table 33-3. SPI Transfer Modes

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

Note:
Leading edge is the first clock edge in a clock cycle.

Trailing edge is the second clock edge in a clock cycle.

Figure 33-3. SPI Transfer Modes



33.6.2.6 Transferring Data

Master

In master mode (CTRLA.MODE=0x3), when Master Slave Enable Select (CTRLB.MSSEN) is ‘1’, hardware will control the $\overline{SS}$ line.

When Master Slave Select Enable (CTRLB.MSSEN) is '0', the $\overline{SS}$ line must be configured as an output. $\overline{SS}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{SS}$ line low.

When writing a character to the Data register (DATA), the character will be transferred to the shift register. Once the content of TxDATA has been transferred to the shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) will be set. And a new character can be written to DATA.

Each time one character is shifted out from the master, another character will be shifted in from the slave simultaneously. If the receiver is enabled (CTRLA.RXEN=1), the contents of the shift register will be transferred to the two-level receive buffer. The transfer takes place in the same clock cycle as the last

data bit is shifted in. And the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set. The received data can be retrieved by reading DATA.

When the last character has been transmitted and there is no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set. When the transaction is finished, the master must pull the $\overline{SS}$ line high to notify the slave. If Master Slave Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

Slave

In slave mode (CTRLA.MODE=0x2), the SPI interface will remain inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the slave will sample and shift out data according to the transaction mode set. When the content of TxDATA has been loaded into the shift register, INTFLAG.DRE will be set, and new data can be written to DATA.

Similar to the master, the slave will receive one character for each character transmitted. A character will be transferred into the two-level receive buffer within the same clock cycle its last data bit is received. The received character can be retrieved from DATA when the Receive Complete interrupt flag (INTFLAG.RXC) is set.

When the master pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set.

After DATA is written it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. Refer to Preloading of the Slave Shift Register.

When transmitting several characters in one SPI transaction, the data has to be written into DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character will be transmitted.

Once the DATA register is empty, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

33.6.2.7 Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). Once an error happens, the bit will stay set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate buffer overflow notification bit in the Control A register (CTRLA.IBON):

If CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA until the receiver complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.

If CTRLA.IBON=0, the buffer overflow condition travels with data through the receive FIFO. After the received data is read, STATUS.BUFOVF and INTFLAG.ERROR will be set along with INTFLAG.RXC, and RxDATA will be zero.

33.6.3 Additional Features

33.6.3.1 Address Recognition

When the SPI is configured for slave operation (CTRLA.MODE=0x2) with address recognition (CTRLA.FORM is 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled, and the transaction is processed. If the device is in sleep mode, an address match can wake up the device in order to process the transaction.

If there is no match, the complete transaction is ignored.

If a 9-bit frame format is selected, only the lower 8 bits of the shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN=0) in order to use this mode.

Related Links

Address Match and Mask

33.6.3.2 Preloading of the Slave Shift Register

When starting a transaction, the slave will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

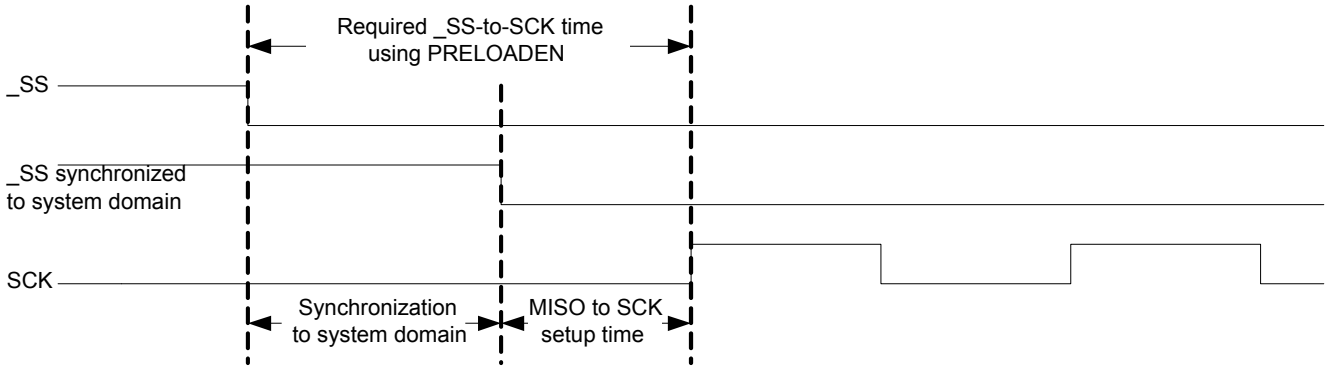
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as in Timing Using Preloading. See also *Electrical Characteristics* for timing details.

Preloading is enabled by writing '1' to the Slave Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

Figure 33-4. Timing Using Preloading



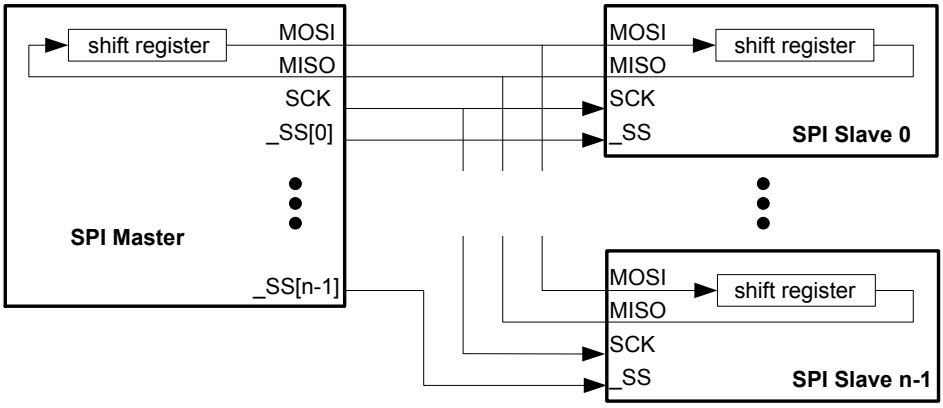
Related Links

Electrical Characteristics

33.6.3.3 Master with Several Slaves

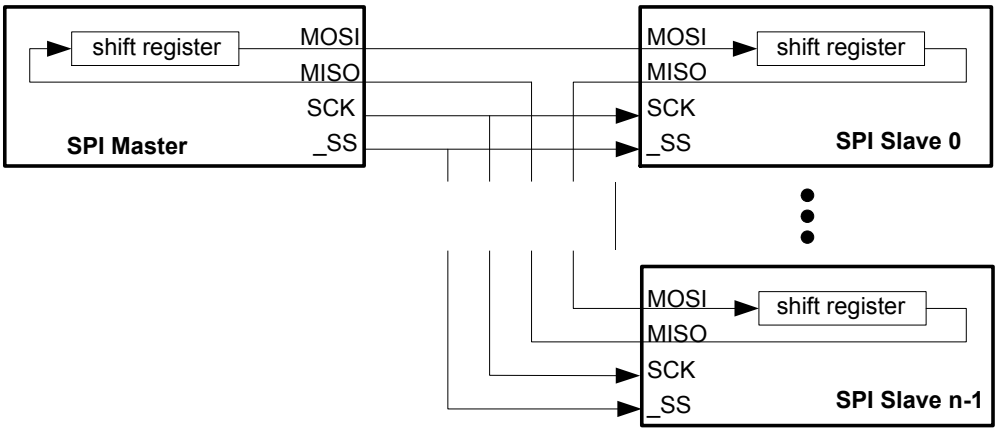
Master with multiple slaves in parallel is only available when Master Slave Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI slaves, an SPI master can use general purpose I/O pins to control the $\overline{SS}$ line to each of the slaves on the bus, as shown in Multiple Slaves in Parallel. In this configuration, the single selected SPI slave will drive the tri-state MISO line.

Figure 33-5. Multiple Slaves in Parallel



Another configuration is multiple slaves in series, as in Multiple Slaves in Series. In this configuration, all n attached slaves are connected in series. A common $\overline{SS}$ line is provided to all slaves, enabling them simultaneously. The master must shift n characters for a complete transaction. Depending on the Master Slave Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

Figure 33-6. Multiple Slaves in Series



33.6.3.4 Loop-Back Mode

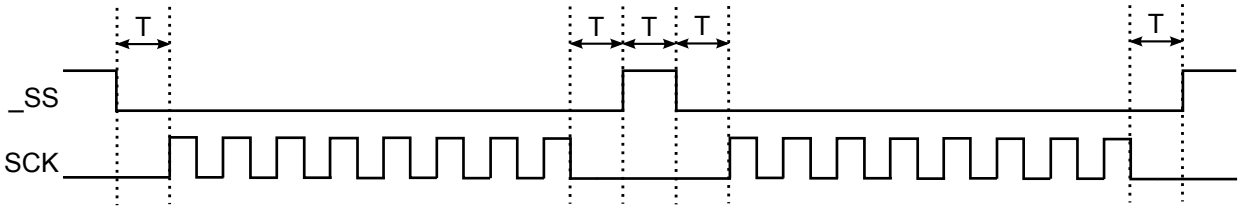
For loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

33.6.3.5 Hardware Controlled $\overline{SS}$

In master mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Master Slave Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In Hardware Controlled $\overline{SS}$, the time T is between one and two baud cycles depending on the SPI transfer mode.

Figure 33-7. Hardware Controlled $\overline{SS}$



T = 1 to 2 baud cycles

When CTRLB.MSSEN=0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

33.6.3.6 Slave Select Low Detection

In slave mode, the SPI can wake the CPU when the slave select ($\overline{SS}$) goes low. When the Slave Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the Slave Select Low interrupt flag (INTFLAG.SSL) and the device will wake up if applicable.

33.6.4 DMA, Interrupts, and Events

Table 33-4. Module Request for SERCOM SPI

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Slave Select low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

33.6.4.1 DMA Operation

The SPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

33.6.4.2 Interrupts

The SPI has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Slave Select Low (SSL)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually

enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SPI is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The SPI has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

33.6.4.3 Events

Not applicable.

33.6.5 Sleep Mode Operation

The behavior in sleep mode is depending on the master/slave configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Master operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_SERCOM_CORE will continue to run in idle sleep mode and in standby sleep mode. Any interrupt can wake up the device.
- Master operation, CTRLA.RUNSTDBY=0: GLK_SERCOMx_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction.

33.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also *CTRLB* register for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

33.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8								IBON
0x02		23:16			DIPO[1:0]				DOPO[1:0]	
0x03		31:24		DORD	CPOL	CPHA	FORM[3:0]			
0x04	CTRLB	7:0		PLOADEN				CHSIZE[2:0]		
0x05		15:8	AMODE[1:0]		MSSEN				SSDE	
0x06		23:16							RXEN	
0x07		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR				SSL	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR				SSL	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR				SSL	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0						BUFOVF		
0x1B		15:8								
0x1C	SYNCBUSY	7:0						CTRLB	ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[7:0]							
0x25		15:8								
0x26		23:16	ADDRMASK[7:0]							
0x27		31:24								
0x28	DATA	7:0	DATA[7:0]							
0x29		15:8								DATA[8:8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

33.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Refer to Synchronization

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Refer to Register Access Protection.

33.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000e74]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
								IBON
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD: Data Order
This bit selects the data order when a character is shifted out from the shift register.

This bit is not synchronized.

Value	Description
0	MSB is transferred first.
1	LSB is transferred first.

Bit 29 – CPOL: Clock Polarity
In combination with the Clock Phase bit (CPHA), this bit determines the SPI transfer mode.

This bit is not synchronized.

Value	Description
0	SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge.
1	SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge.

Bit 28 – CPHA: Clock Phase
In combination with the Clock Polarity bit (CPOL), this bit determines the SPI transfer mode.

This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Description
0	The data is sampled on a leading SCK edge and changed on a trailing SCK edge.
1	The data is sampled on a trailing SCK edge and changed on a leading SCK edge.

Bits 27:24 – FORM[3:0]: Frame Format
This bit field selects the various frame formats supported by the SPI in slave mode. When the 'SPI frame with address' format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	-	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	-	Reserved

Bits 21:20 – DIPO[1:0]: Data In Pinout
These bits define the data in (DI) pad configurations.

In master operation, DI is MISO.

In slave operation, DI is MOSI.

These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0]: Data Out Pinout
This bit defines the available pad configurations for data out (DO) and the serial clock (SCK). In slave operation, the slave select line ($\overline{SS}$) is controlled by DOPO, while in master operation the $\overline{SS}$ line is controlled by the port configuration.

In master operation, DO is MOSI.

In slave operation, DO is MISO.

These bits are not synchronized.

DOPO	DO	SCK	Slave $\overline{SS}$	Master $\overline{SS}$
0x0	PAD[0]	PAD[1]	PAD[2]	System configuration
0x1	PAD[2]	PAD[3]	PAD[1]	System configuration
0x2	PAD[3]	PAD[1]	PAD[2]	System configuration
0x3	PAD[0]	PAD[3]	PAD[1]	System configuration

Bit 8 – IBON: Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.

This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY: Run In Standby
This bit defines the functionality in standby sleep mode.

These bits are not synchronized.

RUNSTDBY	Slave	Master
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake up the device.
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in sleep modes. All interrupts can wake up the device.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits must be written to 0x2 or 0x3 to select the SPI serial communication interface of the SERCOM.

0x2: SPI slave operation

0x3: SPI master operation

These bits are not synchronized.

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing "1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY. SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

33.8.2 Control B

Name: CTRLB
Offset: 0x04 [ID-00000e74]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	
Access							R/W	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	R/W	R/W	R/W				R/W	
Reset	0	0	0				0	

Bit	7	6	5	4	3	2	1	0
		PLOADEN				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN: Receiver Enable

Writing '0' to this bit will disable the SPI receiver immediately. The receive buffer will be flushed, data from ongoing receptions will be lost and STATUS.BUFOVF will be cleared.

Writing '1' to CTRLB.RXEN when the SPI is disabled will set CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN will be cleared, SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled CTRLB.RXEN will read back as '1'.

Writing '1' to CTRLB.RXEN when the SPI is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or it will be enabled when SPI is enabled.

Bits 15:14 – AMODE[1:0]: Address Mode

These bits set the slave addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in master mode.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The slave responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit
0x3	-	Reserved

Bit 13 – MSSEN: Master Slave Select Enable

This bit enables hardware slave select ($\overline{SS}$) control.

Value	Description
0	Hardware $\overline{SS}$ control is disabled.
1	Hardware $\overline{SS}$ control is enabled.

Bit 9 – SSDE: Slave Select Low Detect Enable
This bit enables wake up when the slave select ($\overline{SS}$) pin transitions from high to low.

Value	Description
0	$\overline{SS}$ low detector is disabled.
1	$\overline{SS}$ low detector is enabled.

Bit 6 – PLOADEN: Slave Data Preload Enable
Setting this bit will enable preloading of the slave shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it will be transferred immediately to the shift register.

Bits 2:0 – CHSIZE[2:0]: Character Size

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	-	Reserved

33.8.3 Baud Rate

Name: BAUD
Offset: 0x0C [ID-00000e74]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BAUD[7:0]: Baud Register
These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator*.

33.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14 [ID-00000e74]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL: Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Slave Select Low Interrupt Enable bit, which disables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disable the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

33.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x16 [ID-00000e74]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL: Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Slave Select Low Interrupt Enable bit, which enables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

33.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18 [ID-00000e74]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R	R/W	R
Reset	0				0	0	0	0

Bit 7 – ERROR: Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The BUFOVF error will set this interrupt flag.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – SSL: Slave Select Low
This flag is cleared by writing '1' to it.

This bit is set when a high to low transition is detected on the _SS pin in slave mode and Slave Select Low Detect (CTRLB.SSDE) is enabled.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 2 – RXC: Receive Complete
This flag is cleared by reading the Data (DATA) register or by disabling the receiver.

This flag is set when there are unread data in the receive buffer. If address matching is enabled, the first data received in a transaction will be an address.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 1 – TXC: Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.

In master mode, this flag is set when the data have been shifted out and there are no new data in DATA.

In slave mode, this flag is set when the _SS pin is pulled high. If address matching is enabled, this flag is only set if the transaction was initiated with an address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 0 – DRE: Data Register Empty
This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready for new data to transmit.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

33.8.7 Status

Name: STATUS
Offset: 0x1A [ID-00000e74]
Reset: 0x0000
Property: –

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						BUFOVF		
Access						R/W		
Reset						0		

Bit 2 – BUFOVF: Buffer Overflow
Reading this bit before reading DATA will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. See also CTRLA.IBON for overflow handling.

When set, the corresponding RxDATA will be zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

33.8.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C [ID-00000e74]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLB	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLB: CTRLB Synchronization Busy
Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB=1 until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB=1, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE=1 until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST: Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST=1 until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

33.8.9 Address

Name: ADDR
Offset: 0x24 [ID-00000e74]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – ADDRMASK[7:0]: Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0]: Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

33.8.10 Data

Name: DATA
Offset: 0x28 [ID-00000e74]
Reset: 0x0000
Property: –

Bit	15	14	13	12	11	10	9	8
								DATA[8:8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – DATA[8:0]: Data
Reading these bits will return the contents of the receive data buffer. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.

Writing these bits will write the transmit data buffer. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

33.8.11 Debug Control

Name: DBGCTRL
Offset: 0x30 [ID-00000e74]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP: Debug Stop Mode
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

34. SERCOM I²C – SERCOM Inter-Integrated Circuit

34.1 Overview

The inter-integrated circuit (I²C) interface is one of the available modes in the serial communication interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in Figure 34-1. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM. Each master and slave have a separate I²C interface containing a shift register, a transmit buffer and a receive buffer. In addition, the I²C master uses the SERCOM baud-rate generator, while the I²C slave uses the SERCOM address match logic.

Related Links

SERCOM – Serial Communication Interface

SERCOM USART and I2C Configurations

34.2 Features

SERCOM I²C includes the following features:

- Master or slave operation
- Can be used with DMA
- Philips I²C compatible
- SMBus™ compatible
- PMBus compatible
- Support of 100kHz and 400kHz, 1MHz and 3.4MHz I²C mode low system clock frequencies
- Physical interface includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Slave operation:
  - Operation in all sleep modes
  - Wake-up on address match
  - 7-bit and 10-bit Address match in hardware for:
  -
    - Unique address and/or 7-bit general call address
    - Address range
    - Two unique addresses can be used with DMA

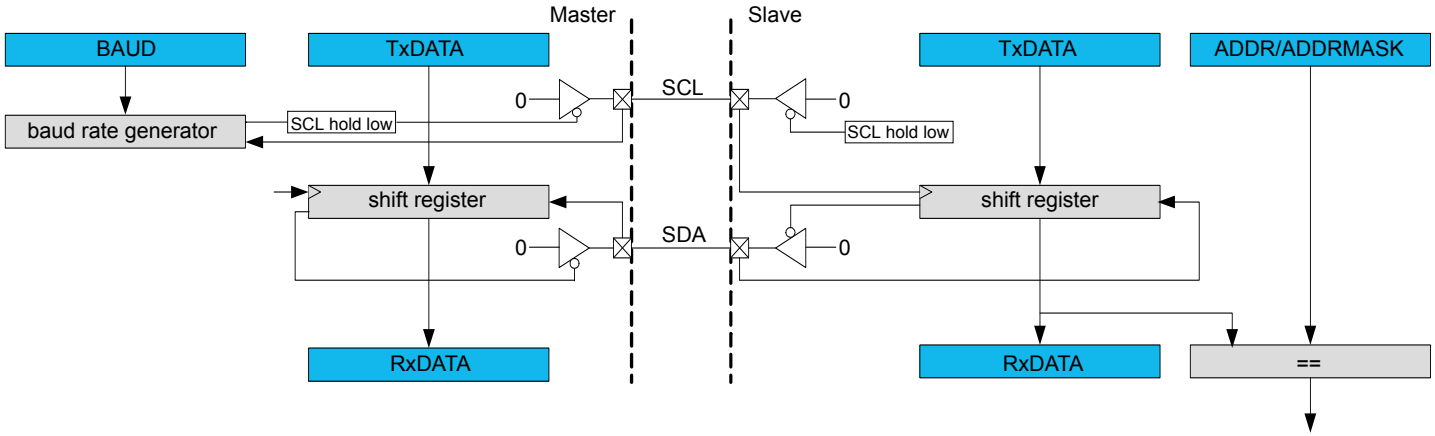
Related Links

SERCOM USART and I2C Configurations

Features

34.3 Block Diagram

Figure 34-1. I²C Single-Master Single-Slave Interconnection



34.4 Signal Description

Signal Name	Type	Description
PAD[0]	Digital I/O	SDA
PAD[1]	Digital I/O	SCL
PAD[2]	Digital I/O	SDA_OUT (4-wire)
PAD[3]	Digital I/O	SDC_OUT (4-wire)

One signal can be mapped on several pins.

Not all the pins are I²C pins.

Related Links

I/O Multiplexing and Considerations

34.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

34.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in I²C mode, the SERCOM controls the direction and value of the I/O pins. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Related Links

PORT: IO Pin Controller

34.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes.

Related Links

PM – Power Manager

34.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock Controller and the Power Manager.

Two generic clocks ared used by SERCOM, GCLK_SERCOMx_CORE and GCLK_SERCOM_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I^2C when working as a master. The slow clock (GCLK_SERCOM_SLOW) is required only for certain functions, e.g. SMBus timing. These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the I^2C.

These generic clocks are asynchronous to the bus clock (CLK_SERCOMx_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

GCLK - Generic Clock Controller

Peripheral Clock Masking

PM – Power Manager

34.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller

34.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

34.5.6 Events

Not applicable.

34.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

34.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)

- Data register (DATA)
- Address register (ADDR)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

34.5.9 Analog Connections

Not applicable.

34.6 Functional Description

34.6.1 Principle of Operation

The I^{2}C interface uses two physical lines for communication:

- Serial Data Line (SDA) for packet transfer
- Serial Clock Line (SCL) for the bus clock

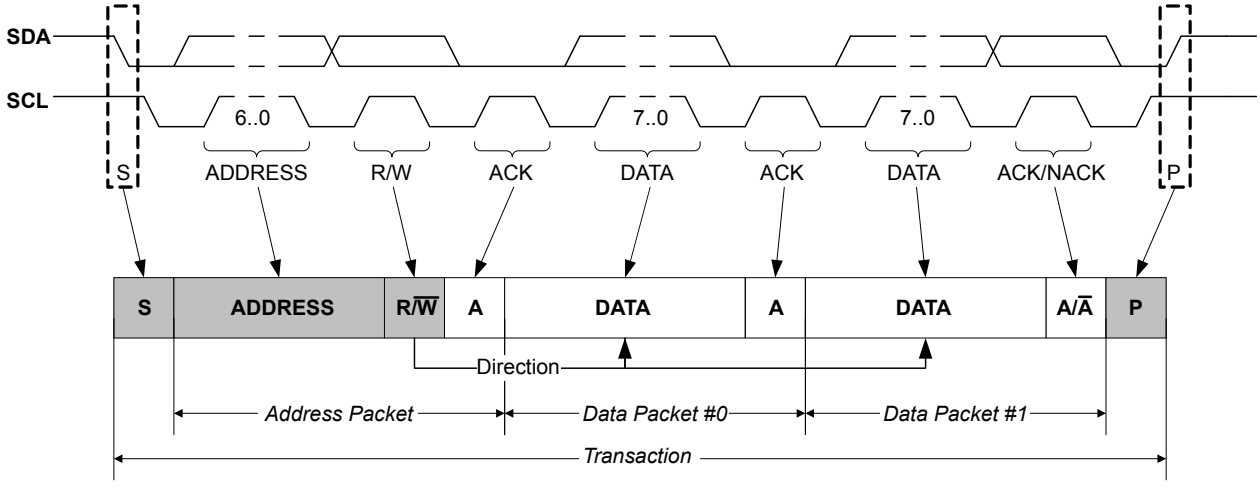
A transaction starts with the I^{2}C master sending the start condition, followed by a 7-bit address and a direction bit (read or write to/from the slave).

The addressed I^{2}C slave will then acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is not acknowledged (NACK), whether by the I^{2}C slave or master, the I^{2}C master takes action by either terminating the transaction by sending the stop condition, or by sending a repeated start to transfer more data.

The figure below illustrates the possible transaction formats and Transaction Diagram Symbols explains the transaction symbols. These symbols will be used in the following descriptions.

Figure 34-2. Basic I^{2}C Transaction Diagram



Transaction Diagram Symbols

Bus Driver


Master driving bus


Slave driving bus


Either Master or Slave driving bus

Special Bus Conditions



START condition



repeated START condition



STOP condition

Data Package Direction



Master Read '1'



Master Write '0'

Acknowledge



Acknowledge (ACK) '0'



Not Acknowledge (NACK) '1'

34.6.2 Basic Operation

34.6.2.1 Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE is ‘0’):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control B register (CTRLB), except Acknowledge Action (CTRLB.ACKACT) and Command (CTRLB.CMD) bits
- Baud register (BAUD)
- Address register (ADDR) in slave operation.

When the I²C is enabled or is being enabled (CTRLA.ENABLE=1), writing to these registers will be discarded. If the I²C is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the I²C is enabled it must be configured as outlined by the following steps:

1. Select I²C Master or Slave mode by writing 0x4 or 0x5 to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
2. If desired, select the SDA Hold Time value in the CTRLA register (CTRLA.SDAHOLD).
3. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
4. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUT).
5. In Master mode:
   - 5.1. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   - 5.2. Write the Baud Rate register (BAUD) to generate the desired baud rate.

   In Slave mode:

5.1. Configure the address match configuration by writing the Address Mode value in the CTRLB register (CTRLB.AMODE).

5.2. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

34.6.2.2 Enabling, Disabling, and Resetting

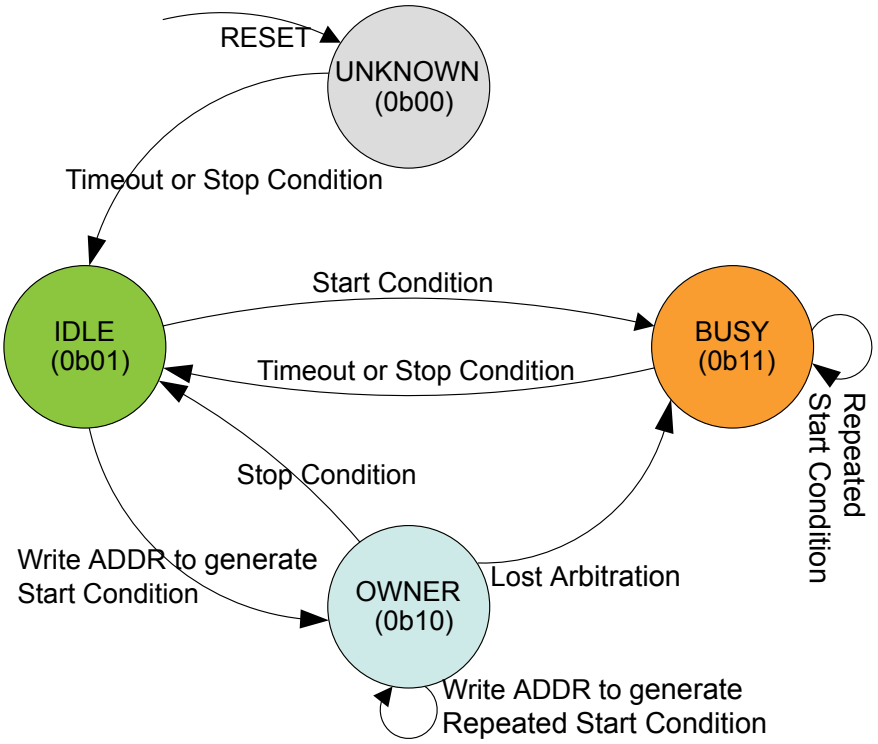
This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Refer to CTRLA for details.

34.6.2.3 I²C Bus State Logic

The bus state logic includes several logic blocks that continuously monitor the activity on the I²C bus lines in all sleep modes. The start and stop detectors and the bit counter are all essential in the process of determining the current bus state. The bus state is determined according to Bus State Diagram. Software can get the current bus state by reading the Master Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the figure is shown in binary.

Figure 34-3. Bus State Diagram



The bus state machine is active when the I²C master is enabled.

After the I²C master has been enabled, the bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by by writing 0b01 to STATUS.BUSSTATE
- A stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs.

Note: Once a known bus state is established, the bus state logic will not re-enter the UNKNOWN state.

When the bus is IDLE it is ready for a new transaction. If a start condition is issued on the bus by another I²C master in a multi-master setup, the bus becomes BUSY (0b11). The bus will re-enter IDLE either

when a stop condition is detected, or when a time-out occurs (inactive bus time-out needs to be configured).

If a start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, i.e., arbitration was not lost, the I^{2}C master can issue a stop condition, which will change the bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the bus state becomes BUSY until a stop condition is detected. A repeated start condition will change the bus state only if arbitration is lost while issuing a repeated start.

Regardless of winning or losing arbitration, the entire address will be sent. If arbitration is lost, only 'ones' are transmitted from the point of losing arbitration and the rest of the address length.

Note: Violating the protocol may cause the I^{2}C to hang. If this happens it is possible to recover from this state by a software reset (CTRLA.SWRST='1').

Related Links

CTRLA

34.6.2.4 I^{2}C Master Operation

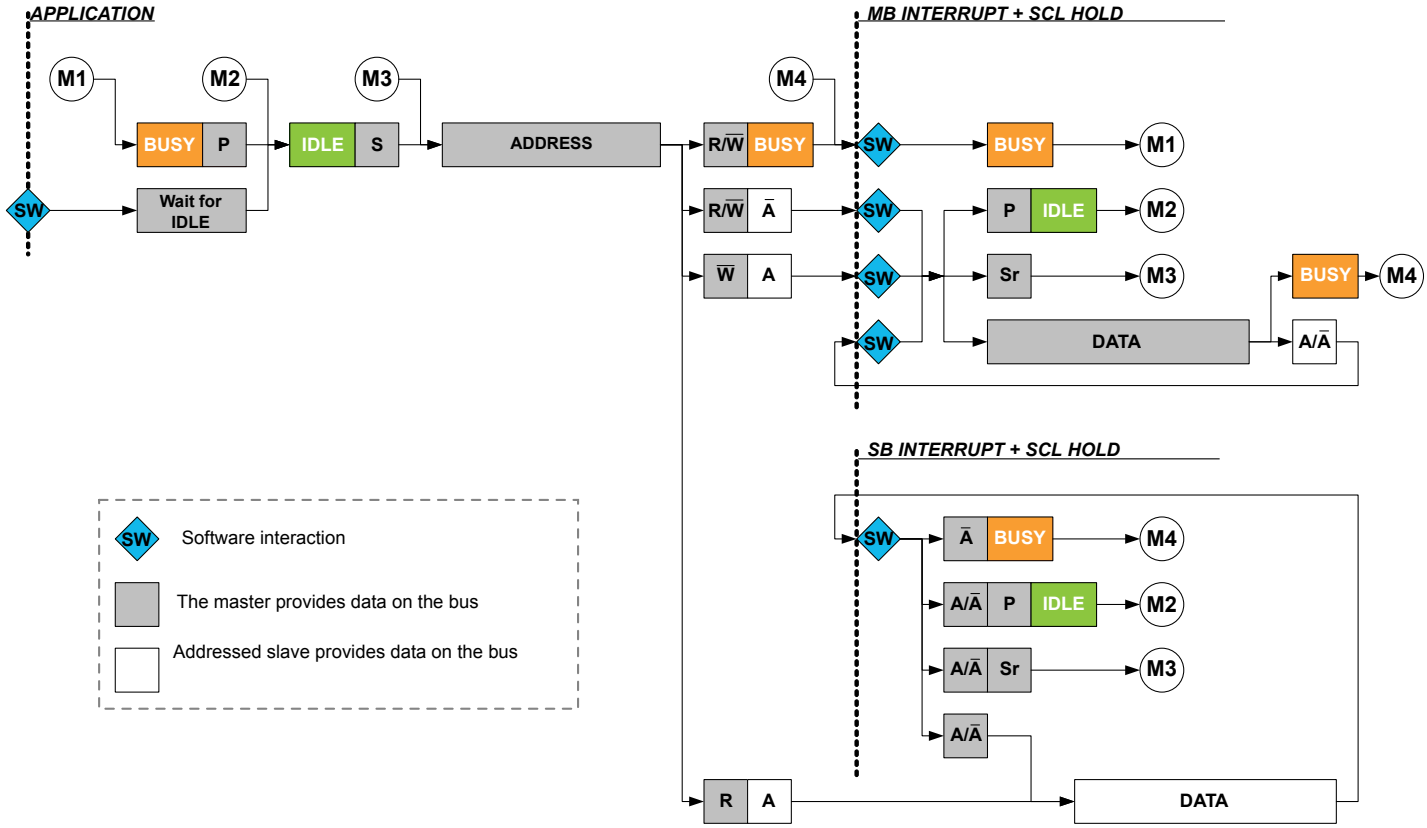
The I^{2}C master is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I^{2}C master has two interrupt strategies.

When SCL Stretch Mode (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit . In this mode the I^{2}C master operates according to Master Behavioral Diagram (SCLSM=0). The circles labelled "M*n*" (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I^{2}C master operation throughout the document.

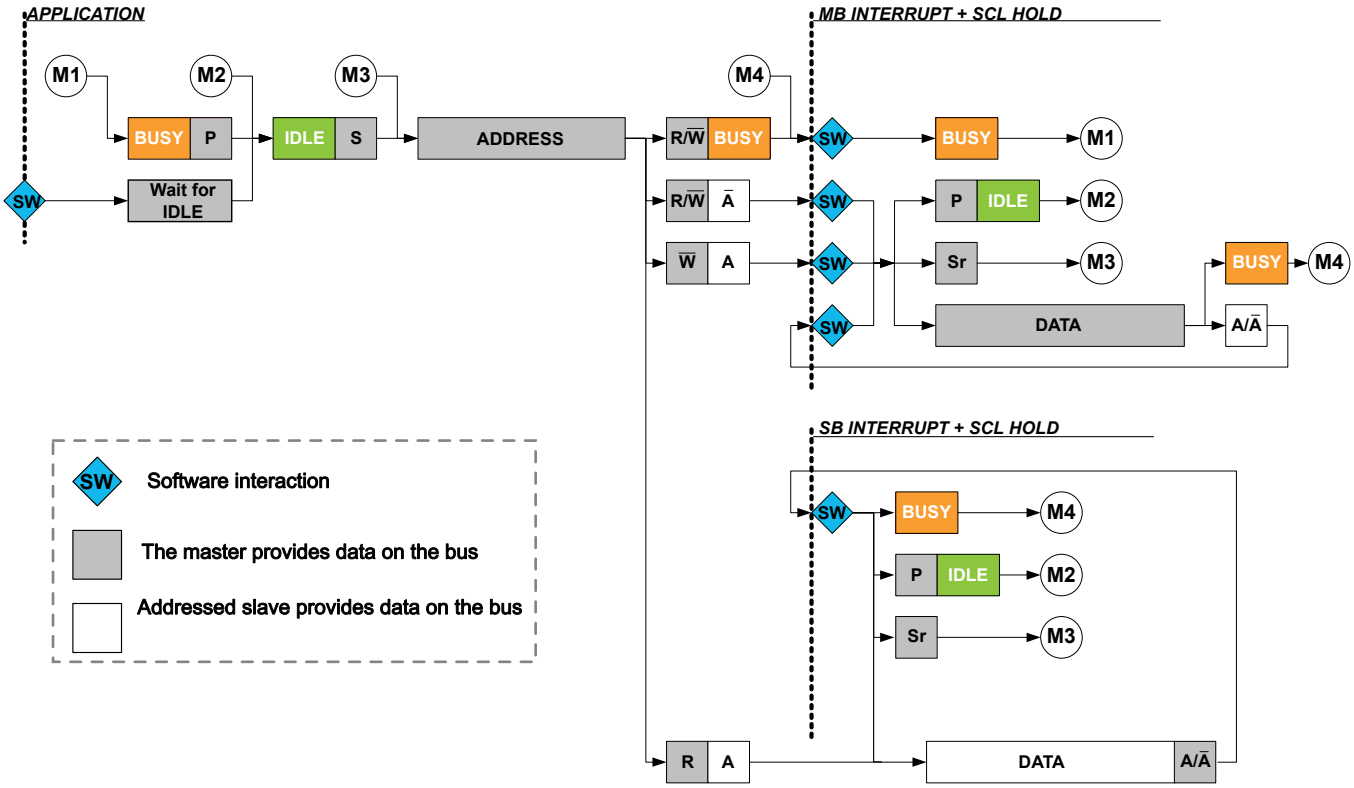
Figure 34-4. I²C Master Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit, as in Master Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging.

Note: I²C High-speed (*Hs*) mode requires CTRLA.SCLSM=1.

Figure 34-5. I²C Master Behavioral Diagram (SCLSM=1)



Master Clock Generation

The SERCOM peripheral supports several I²C bi-directional modes:

- Standard mode (*Sm*) up to 100kHz
- Fast mode (*Fm*) up to 400kHz
- Fast mode Plus (*Fm+*) up to 1MHz
- High-speed mode (*Hs*) up to 3.4MHz

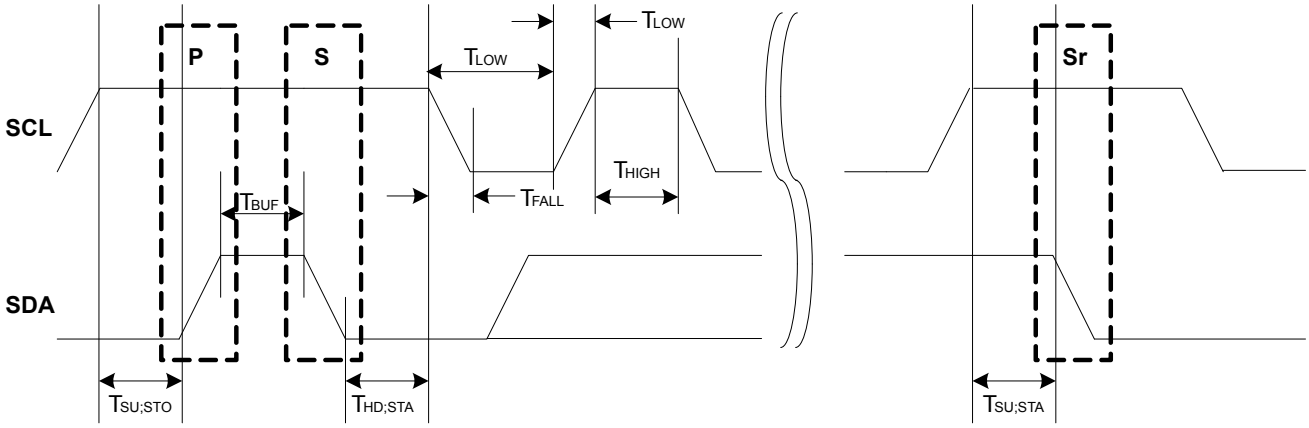
The Master clock configuration for *Sm, Fm,* and *Fm+* are described in Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus). For *Hs*, refer to Master Clock Generation (High-Speed Mode).

Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)

In I²C *Sm, Fm*, and *Fm+* mode, the Master clock (SCL) frequency is determined as described in this section:

The low (T_{LOW}) and high (T_{HIGH}) times are determined by the Baud Rate register (BAUD), while the rise (T_{RISE}) and fall (T_{FALL}) times are determined by the bus topology. Because of the wired-AND logic of the bus, T_{FALL} will be considered as part of T_{LOW}. Likewise, T_{RISE} will be in a state between T_{LOW} and T_{HIGH} until a high state has been detected.

Figure 34-6. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Master Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, or the Master Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- T_{LOW} – Low period of SCL clock
- $T_{SU;STO}$ – Set-up time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Set-up time for repeated start condition
- T_{HIGH} is timed using the SCL high time count from BAUD.BAUD
- T_{RISE} is determined by the bus impedance; for internal pull-ups. Refer to *Electrical Characteristics*.
- T_{FALL} is determined by the open-drain current limit and bus impedance; can typically be regarded as zero. Refer to *Electrical Characteristics* for details.

The SCL frequency is given by:

$$f_{\mathrm{SCL}} = \frac{1}{T_{\mathrm{LOW}} + T_{\mathrm{HIGH}} + T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case the following formula will give the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + 2BAUD + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + BAUD + BAUDLOW + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{\mathrm{LOW}} = \frac{BAUDLOW + 5}{f_{\mathrm{GCLK}}}$$

$$T_{\mathrm{HIGH}} = \frac{BAUD + 5}{f_{\mathrm{GCLK}}}$$

Note: The I²C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Startup Timing The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.

Note: When timing is controlled by user, the Smart Mode cannot be enabled.

Related Links

Electrical Characteristics

Master Clock Generation (High-Speed Mode)

For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency (f_{GCLK}) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\,BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\,BAUD + HSBAUDLOW}$$

Note: The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

Transmitting Address Packets

The I²C master starts a bus transaction by writing the I²C slave address to ADDR.ADDR and the direction bit, as described in Principle of Operation. If the bus is busy, the I²C master will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C master will issue a start condition on the bus. The I²C master will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C master, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Master on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C master is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the arbitration lost condition. In this case, the MB interrupt flag and Master Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Master Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C slave device responding to the address packet, then the INTFLAG.MB interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

The missing ACK response can indicate that the I2C slave is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a stop condition (recommended) or resending the address packet by a repeated start condition. When using SMBus logic, the slave must ACK the address. If there is no response, it means that the slave is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Master on Bus set

If the I2C master receives an acknowledge response from the I2C slave, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I2C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.
- Transmit a new address packet by writing ADDR.ADDR. A repeated start condition will automatically be inserted before the address packet.
- Issue a stop condition, consequently terminating the transaction.

Case 4: Address packet transmit complete – Read packet, Slave on Bus set

If the I2C master receives an ACK from the I2C slave, the I2C master proceeds to receive the next byte of data from the I2C slave. When the first data byte is received, the Slave on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I2C operation to continue:

- Let the I2C master continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a stop condition.

Note: An ACK or NACK will be automatically transmitted if smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK should be sent.

Transmitting Data Packets

When an address packet with direction Master Write (see Figure 34-2) was transmitted successfully , INTFLAG.MB will be set. The I2C master will start transmitting data via the I2C bus by writing to DATA.DATA, and monitor continuously for packet collisions. I

If a collision is detected, the I2C master will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I2C master will receive an ACK bit from the I2C slave, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I2C Master on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I2C master is not allowed to continue transmitting data packets if a NACK is received from the I2C slave.

Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I2C master will already have received one data packet. The I2C master must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when

arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I^2C master will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the smart mode.

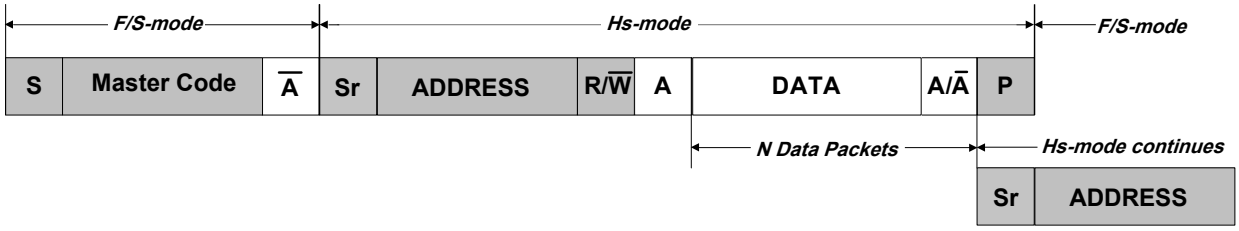
High-Speed Mode

High-speed transfers are a multi-step process, see High Speed Transfer.

First, a master code (0b00001nnn, where 'nnn' is a unique master code) is transmitted in Full-speed mode, followed by a NACK since no slaveshould acknowledge. Arbitration is performed only during the Full-speed Master Code phase. The master code is transmitted by writing the master code to the address register (ADDR.ADDR) and writing the high-speed bit (ADDR.HS) to '0'.

After the master code and NACK have been transmitted, the master write interrupt will be asserted. In the meanwhile, the slave address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the master will generate a repeated start, followed by the slave address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

Figure 34-7. High Speed Transfer



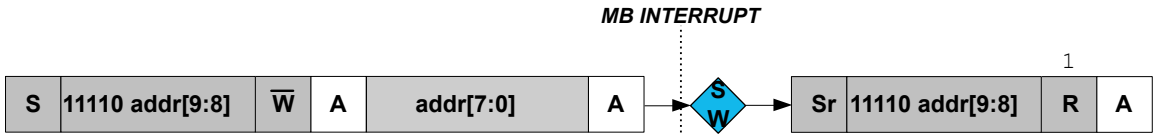
Transmitting in High-speed mode requires the I^2C master to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL clock stretch mode (CTRLA.SCLSM) bit set to '1'.

10-Bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN=1) and the Address bit field ADDR.ADDR is written, the two address bytes will be transmitted, see 10-bit Address Transmission for a Read Transaction. The addressed slave acknowledges the two address bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the master must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being zero.

If the master receives a NACK after the first byte, the write interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more slaves, then the master will proceed to transmit the second address byte and the master will first see the write interrupt flag after the second byte is transmitted. If the transaction direction is read-from-slave, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

Figure 34-8. 10-bit Address Transmission for a Read Transaction



This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).
2. Once the Master on Bus interrupt is asserted, Write ADDR[7:0] register to '11110 `address[9:8]` 1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

34.6.2.5 I²C Slave Operation

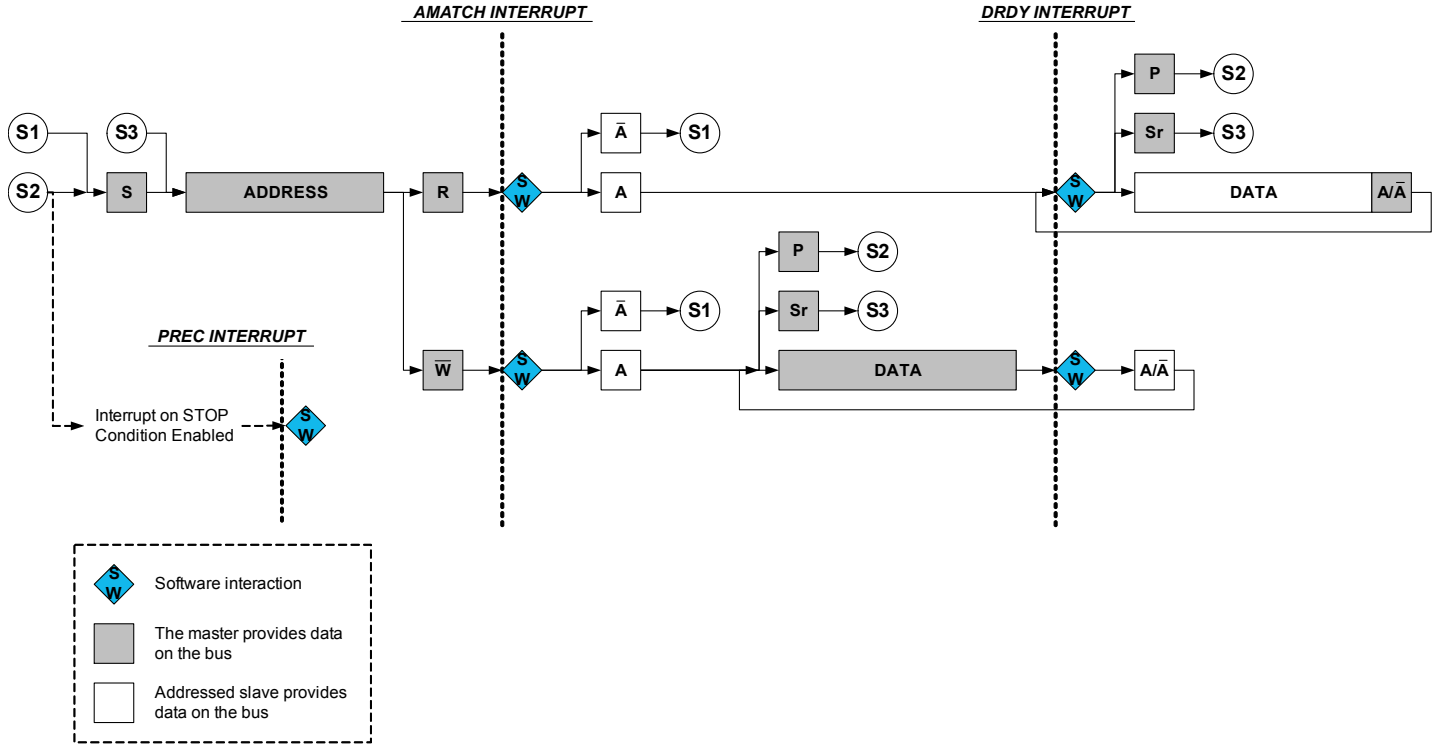
The I²C slave is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C slave has two interrupt strategies.

When SCL Stretch Mode bit (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C slave operates according to I²C Slave Behavioral Diagram (SCLSM=0). The circles labelled "S*n*" (S1, S2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C slave operation throughout the document.

Figure 34-9. I²C Slave Behavioral Diagram (SCLSM=0)

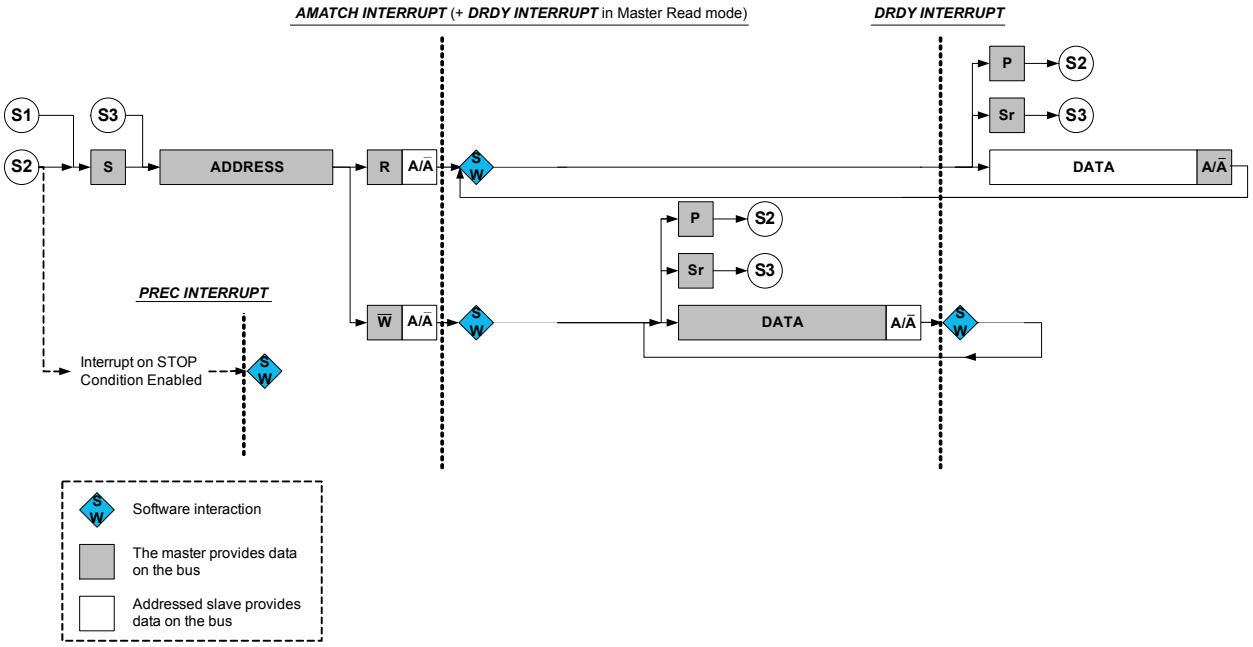


In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Slave Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check

DATA before acknowledging. For master reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for master writes, the first data interrupt will be seen after the first data byte has been received by the slave and the acknowledge bit has been sent to the master.

Note: For I²C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 34-10. I²C Slave Behavioral Diagram (SCLSM=1)



Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I2C slave stretches the SCL line according to Figure 34-9. When the I²C slave is properly configured, it will wait for a start condition.

When a start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C slave will wait for a new start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C slave clears INTFLAG.AMATCH. As the I²C slave holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read / Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C master, one of two cases will arise based on transfer direction.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is '1', indicating an I²C master read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C slave hardware will set the Data Ready bit in the Interrupt Flag

register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I2C slave will wait for a new start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I2C slave Command bit field in the Control B register (CTRLB.CMD) can be written to '0x3' for both read and write operations as the command execution is dependent on the STATUS.DIR bit. Writing ‘1’ to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I2C master write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I2C slave will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I2C slave will wait for a new start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I2C slave command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing ‘1’ to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I2C slave will stretch the SCL line only after an ACK, see Slave Behavioral Diagram (SCLSM=1). When the I2C slave is properly configured, it will wait for a start condition to be detected.

When a start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I2C slave will wait for a new start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be stretched until the I2C slave clears INTFLAG.AMATCH. As the I2C slave holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I2C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet’s collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet has been received from the I2C master, INTFLAG.AMATCH be set to ‘1’ to clear it.

Receiving and Transmitting Data Packets

After the I2C slave has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA. When a data packet is received or sent, INTFLAG.DRDY will be set. After receiving data, the I2C slave will send an acknowledge according to CTRLB.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set, and SCL is held low, pending for SW interaction.

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I^2C slave must expect a stop or a repeated start to be received. The I^2C slave must release the data line to allow the I^2C master to generate a stop or repeated start. Upon detecting a stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I^2C slave will return to IDLE state.

High-Speed Mode

When the I^2C slave is configured in High-speed mode (*Hs*, CTRLA.SPEED=0x2) and CTRLA.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the slave recognizes a START followed by a master code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The slave will then remain in High-speed mode until a STOP is received.

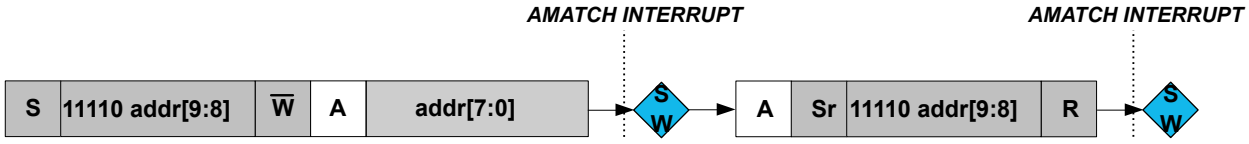
10-Bit Addressing

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit slave address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of '11110 `ADDR[9:8]` 1', and the second address interrupt will be received with the DIR bit set. The slave matches on the second address as it it was addressed by the previous 10-bit address.

Figure 34-11. 10-bit Addressing



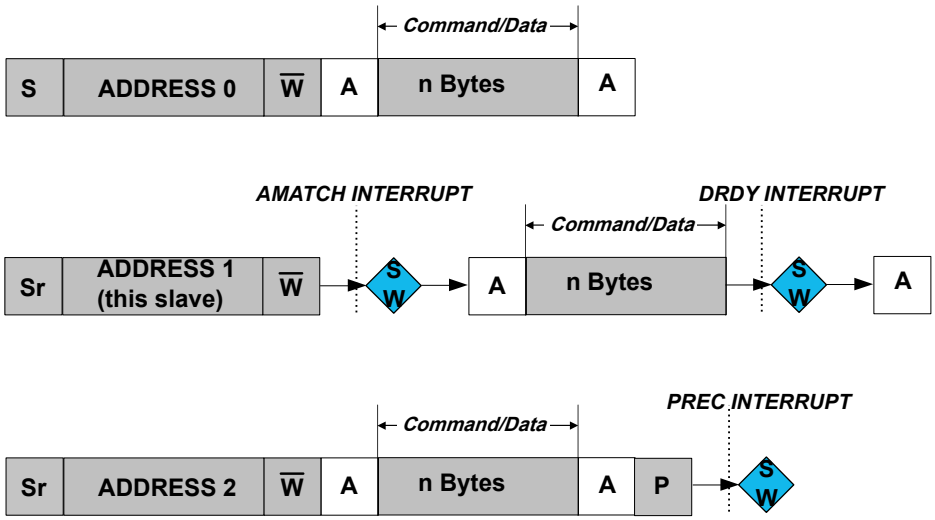
PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set when a STOP condition is detected on the bus. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the slaves addressed during the group command, they all begin executing the command they received.

PMBus Group Command Example shows an example where this slave, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple slaves addressed before and after this slave. Eventually, at the end of the group command, a single STOP is generated by the master. At this point a STOP interrupt is asserted.

Figure 34-12. PMBus Group Command Example



34.6.3 Additional Features

34.6.3.1 SMBus

The I^{2}C includes three hardware SCL low time-outs which allow a time-out to occur for SMBus SCL low time-out, master extend time-out, and slave extend time-out. This allows for SMBus functionality These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32KHz oscillator. The I^{2}C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- $T_{LOW:SEXT}$: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a slave device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- $T_{LOW:MEXT}$: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the master device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

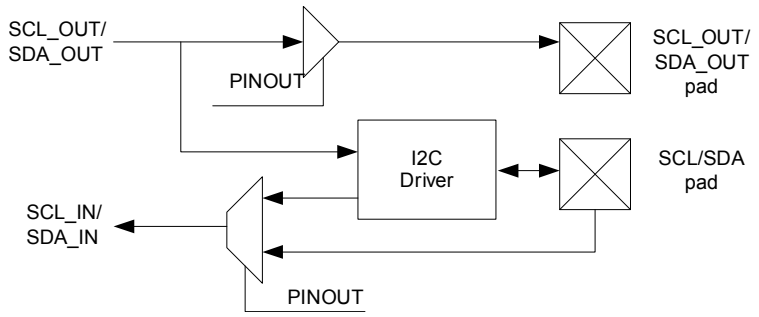
34.6.3.2 Smart Mode

The I^{2}C interface has a smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I^{2}C protocol. The smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

34.6.3.3 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) will enable 4-wire mode operation. In this mode, the internal I^{2}C tri-state drivers are bypassed, and an external I^{2}C compliant tri-state driver is needed when connecting to an I^{2}C bus.

Figure 34-13. I²C Pad Interface



34.6.3.4 Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the slave acknowledges the address. At this point, the software can either issue a stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

34.6.4 DMA, Interrupts and Events

Table 34-1. Module Request for SERCOM I²C Slave

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Slave transmit mode)	Yes (request cleared when data is written)		NA
Data received (RX) (Slave receive mode)	Yes (request cleared when data is read)		
Data Ready (DRDY)		Yes	
Address Match (AMATCH)		Yes	
Stop received (PREC)		Yes	
Error (ERROR)		Yes	

Table 34-2. Module Request for SERCOM I²C Master

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Master transmit mode)	Yes (request cleared when data is written)		NA
Data needed for transmit (RX) (Master transmit mode)	Yes (request cleared when data is read)		
Master on Bus (MB)		Yes	

Condition	Request		
	DMA	Interrupt	Event
Stop received (SB)		Yes	
Error (ERROR)		Yes	

34.6.4.1 DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

Slave DMA

When using the I^2C slave with DMA, an address match will cause the address interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt has been serviced, data transfer will be performed through DMA.

The I^2C slave generates the following requests:

- Write data received (RX): The request is set when master write data is received. The request is cleared when DATA is read.
- Read data needed for transmit (TX): The request is set when data is needed for a master read operation. The request is cleared when DATA is written.

Master DMA

When using the I^2C master with DMA, the ADDR register must be written with the desired address (ADDR.ADDR), transaction length (ADDR.LEN), and transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to 1 along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. DMA is then used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for master reads) and a STOP.

If a NACK is received by the slave for a master write transaction before ADDR.LEN bytes, a STOP will be automatically generated and the length error (STATUS.LENERR) will be raised along with the INTFLAG.ERROR interrupt.

The I^2C master generates the following requests:

- Read data received (RX): The request is set when master read data is received. The request is cleared when DATA is read.
- Write data needed for transmit (TX): The request is set when data is needed for a master write operation. The request is cleared when DATA is written.

34.6.4.2 Interrupts

The I^2C slave has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I^2C master has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)
- Slave on Bus (SB)
- Master on Bus (MB)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request active until the interrupt flag is cleared, the interrupt is disabled or the I^2C is reset. See INTFLAG register for details on how to clear interrupt flags.

The I^2C has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

34.6.4.3 Events

Not applicable.

34.6.5 Sleep Mode Operation

I^2C Master Operation

The generic clock (GCLK_SERCOMx_CORE) will continue to run in idle sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE will also run in standby sleep mode. Any interrupt can wake up the device.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I^2C Slave Operation

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

34.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in master operation.

The following registers are synchronized when written:

- Data (DATA) when in master operation

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

34.7 Register Summary - I2C Slave

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8								
0x02		23:16	SEXTTOEN		SDAHOLD[1:0]					PINOUT
0x03		31:24		LOWTOUT			SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
0x05		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
0x06		23:16						ACKACT	CMD[1:0]	
0x07		31:24								
0x08 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR					DRDY	AMATCH	PREC
0x15	Reserved									
0x16	INTENSET	7:0	ERROR					DRDY	AMATCH	PREC
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR					DRDY	AMATCH	PREC
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
0x1B		15:8						LENERR	SEXTTOUT	
0x1C	SYNCBUSY	7:0							ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[6:0]							GENCEN
0x25		15:8	TENBITEN					ADDR[9:7]		
0x26		23:16	ADDRMASK[6:0]							
0x27		31:24						ADDRMASK[9:7]		
0x28	DATA	7:0	DATA[7:0]							
0x29		15:8								

34.8 Register Description - I2C Slave

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

34.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00001bb3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN		SDAHOLD[1:0]					PINOUT
Access	R/W		R/W	R/W				R/W
Reset	0		0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT: SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the slave will release its clock hold, if enabled, and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bit 27 – SCLSM: SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-9
1	SCL stretch only after ACK bit according to Figure 34-10

Bits 25:24 – SPEED[1:0]: Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN: Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the slave will release its clock hold if enabled and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0]: SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75	50-100ns hold time
0x2	450	300-600ns hold time
0x3	600	400-800ns hold time

Bit 16 – PINOUT: Pin Usage
This bit sets the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled
1	4-wire operation enabled

Bit 7 – RUNSTDBY: Run in Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits must be written to 0x04 to select the I^2C slave serial communication interface of the SERCOM.

These bits are not synchronized.

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Enable Synchronization

Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.8.2 Control B

Name: CTRLB
Offset: 0x04 [ID-00001bb3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT: Acknowledge Action
This bit defines the slave's acknowledge behavior after an address or data byte is received from the master. The acknowledge action is executed when a command is written to the CMD bits. If smart mode is enabled (CTRLB.SMEN=1), the acknowledge action is performed when the DATA register is read.

This bit is not enable-protected.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0]: Command
This bit field triggers the slave operation as the below. The CMD bits are strobe bits, and always read as zero. The operation is dependent on the slave interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.

All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.

This bit is not enable-protected.

Table 34-3. Command Description

CMD[1:0]	DIR	Action
0x0	X	(No action)
0x1	X	(Reserved)
0x2	Used to complete a transaction in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by waiting for any start (S/Sr) condition
	1 (Master read)	Wait for any start (S/Sr) condition

CMD[1:0]	DIR	Action
0x3	Used in response to an address interrupt (AMATCH)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute acknowledge action succeeded by slave data interrupt
	Used in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute a byte read operation followed by ACK/NACK reception

Bits 15:14 – AMODE[1:0]: Address Mode
These bits set the addressing mode.

These bits are not write-synchronized.

Value	Name	Description
0x0	MASK	The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See *SERCOM – Serial Communication Interface* for additional information.
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The slave responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit.
0x3	-	Reserved.

Bit 10 – AACKEN: Automatic Acknowledge Enable
This bit enables the address to be automatically acknowledged if there is an address match.

This bit is not write-synchronized.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD: PMBus Group Command
This bit enables PMBus group command support. When enabled, the Stop Recived interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the slave has been addressed since the last STOP condition on the bus.

This bit is not write-synchronized.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN: Smart Mode Enable
When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.8.3 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x14 [ID-00001bb3]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY: Data Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready bit, which disables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH: Address Match Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC: Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

© 2017 Microchip Technology Inc.

34.8.4 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x16 [ID-00001bb3]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY: Data Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Ready bit, which enables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH: Address Match Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC: Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

34.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18 [ID-00001bb3]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR: Error
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The corresponding bits in STATUS are SEXTTOUT, LOWTOUT, COLL, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 2 – DRDY: Data Ready
This flag is set when a I^2C slave byte transmission is successfully completed.

The flag is cleared by hardware when either:

- Writing to the DATA register.
- Reading the DATA register with smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready interrupt flag.

Bit 1 – AMATCH: Address Match
This flag is set when the I^2C slave address match logic detects that a valid address has been received.

The flag is cleared by hardware when CTRL.CMD is written.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC: Stop Received
This flag is set when a stop condition is detected for a transaction being processed. A stop condition detected between a bus master and another slave will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).

This flag is cleared by hardware after a command is issued on the next address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received interrupt flag.

34.8.6 Status

Name: STATUS
Offset: 0x1A [ID-00001bb3]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	
Access						R/W	R/W	
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
Access	R	R/W		R	R	R	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 10 – LENERR: Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically when responding to a new start condition with ACK or NACK (CTRLB.CMD=0x3) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Bit 10 – HS: High-speed
This bit is set if the slave detects a START followed by a Master Code transmission.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status. However, this flag is automatically cleared when a STOP is received.

Bit 9 – SEXTTOUT: Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low extend time-out has occurred.
1	SCL low extend time-out has occurred.

Bit 7 – CLKHOLD: Clock Hold
The slave Clock Hold bit (STATUS.CLKHOLD) is set when the slave is holding the SCL line low, stretching the I2C clock. Software should consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.

This bit is automatically cleared when the corresponding interrupt is also cleared.

Bit 6 – LOWTOUT: SCL Low Time-out
This bit is set if an SCL low time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low time-out has occurred.
1	SCL low time-out has occurred.

Bit 4 – SR: Repeated Start
When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.

This flag is only valid while the INTFLAG.AMATCH flag is one.

Value	Description
0	Start condition on last address match
1	Repeated start condition on last address match

Bit 3 – DIR: Read / Write Direction
The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a master.

Value	Description
0	Master write operation is in progress.
1	Master read operation is in progress.

Bit 2 – RXNACK: Received Not Acknowledge
This bit indicates whether the last data packet sent was acknowledged or not.

Value	Description
0	Master responded with ACK.
1	Master responded with NACK.

Bit 1 – COLL: Transmit Collision
If set, the I2C slave was not able to transmit a high data or NACK bit, the I2C slave will immediately release the SDA and SCL lines and wait for the next packet addressed to it.

This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and should be treated as a bus error.

Note that this status will not trigger any interrupt, and should be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No collision detected on last data byte sent.
1	Collision detected on last data byte sent.

Bit 0 – BUSERR: Bus Error
The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set STATUS.BUSERR.

This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.

Writing a '1' to this bit will clear the status.

Writing a '0' to this bit has no effect.

Value	Description
0	No bus error detected.
1	Bus error detected.

34.8.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C [ID-00001bb3]
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST: Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.8.8 Address

Name: ADDR
Offset: 0x24 [ID-00001bb3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
						ADDRMASK[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[6:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
	TENBITEN					ADDR[9:7]		
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[6:0]							GENCEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:17 – ADDRMASK[9:0]: Address Mask
These bits act as a second address match register, an address mask register or the lower limit of an address range, depending on the CTRLB.AMODE setting.

Bit 15 – TENBITEN: Ten Bit Addressing Enable

Value	Description
0	10-bit address recognition disabled.
1	10-bit address recognition enabled.

Bits 10:1 – ADDR[9:0]: Address
These bits contain the I²C slave address used by the slave address match logic to determine if a master has addressed the slave.

When using 7-bit addressing, the slave address is represented by ADDR[6:0].

When using 10-bit addressing (ADDR.TENBITEN=1), the slave address is represented by ADDR[9:0]

When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

Bit 0 – GENCEN: General Call Address Enable
A general call address is an address consisting of all-zeroes, including the direction bit (master write).

Value	Description
0	General call address recognition disabled.
1	General call address recognition enabled.

34.8.9 Data

Name: DATA
Offset: 0x28 [ID-00001bb3]
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0]: Data
The slave data register I/O location (DATA.DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the slave (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.

Accessing DATA.DATA auto-triggers I²C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

34.9 Register Summary - I2C Master

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8								
0x02		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
0x03		31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
0x05		15:8							QCEN	SMEN
0x06		23:16						ACKACT	CMD[1:0]	
0x07		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D		15:8	BAUDLOW[7:0]							
0x0E		23:16	HSBAUD[7:0]							
0x0F		31:24	HSBAUDLOW[7:0]							
0x10 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR						SB	MB
0x15	Reserved									
0x16	INTENSET	7:0	ERROR						SB	MB
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR						SB	MB
0x18	DATA	7:0	DATA[7:0]							
0x19		15:8								
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
0x1B		15:8						LENERR	SEXTTOUT	MEXTTOUT
0x1C	SYNCBUSY	7:0						SYSOP	ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x21 ... 0x23	Reserved									
0x24	ADDR	7:0								
0x25		15:8	TENBITEN	HS	LENEN			ADDR[2:0]		
0x26		23:16	LEN[7:0]							
0x27		31:24								
0x28 ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

34.10 Register Description - I²C Master

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

34.10.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00001bb3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
Access		R/W	R/W	R/W	R/W		R/W	R/W
Reset		0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT: SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the master will release its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted.

INTFLAG.SB or INTFLAG.MB will be set as normal, but the clock hold will be released. The STATUS.LOWTOUT and STATUS.BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bits 29:28 – INACTOUT[1:0]: Inactive Time-Out
If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic will be set to idle. An inactive bus arise when either an I^2C master or slave is holding the SCL low.

Enabling this option is necessary for SMBus compatibility, but can also be used in a non-SMBus set-up.

Calculated time-out periods are based on a 100kHz baud rate.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	55US	5-6 SCL cycle time-out (50-60µs)
0x2	105US	10-11 SCL cycle time-out (100-110µs)
0x3	205US	20-21 SCL cycle time-out (200-210µs)

Bit 27 – SCLSM: SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-4.
1	SCL stretch only after ACK bit, Figure 34-5.

Bits 25:24 – SPEED[1:0]: Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN: Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be release. The MEXTTOUT and BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 22 – MEXTTOEN: Master SCL Low Extend Time-Out
This bit enables the master SCL low extend time-out. If SCL is cumulatively held low for greater than 10ms from START-to-ACK, ACK-to-ACK, or ACK-to-STOP the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0]: SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75NS	50-100ns hold time
0x2	450NS	300-600ns hold time
0x3	600NS	400-800ns hold time

Bit 16 – PINOUT: Pin Usage
This bit set the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled.
1	4-wire operation enabled.

Bit 7 – RUNSTDBY: Run in Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^2C master will not operate in standby sleep mode.
1	GCLK_SERCOMx_CORE is enabled in all sleep modes.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits must be written to 0x5 to select the I^2C master serial communication interface of the SERCOM.

These bits are not synchronized.

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.10.2 Control B

Name: CTRLB
Offset: 0x04 [ID-00001bb3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							QCEN	SMEN
Access							R	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT: Acknowledge Action
This bit defines the I^2C master's acknowledge behavior after a data byte is received from the I^2C slave. The acknowledge action is executed when a command is written to CTRLB.CMD, or if smart mode is enabled (CTRLB.SMEN is written to one), when DATA.DATA is read.

This bit is not enable-protected.

This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0]: Command
Writing these bits triggers a master operation as described below. The CMD bits are strobe bits, and always read as zero. The acknowledge action is only valid in master read mode. In master write mode, a command will only result in a repeated start or stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and then the acknowledge action will be updated before the command is triggered.

Commands can only be issued when either the Slave on Bus interrupt flag (INTFLAG.SB) or Master on Bus interrupt flag (INTFLAG.MB) is '1'.

If CMD 0x1 is issued, a repeated start will be issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This will trigger a repeated start followed by transmission of the new address.

Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 34-4. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a stop condition

These bits are not enable-protected.

Bit 9 – QCEN: Quick Command Enable
This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN: Smart Mode Enable
When smart mode is enabled, acknowledge action is sent when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.10.3 Baud Rate

Name: BAUD
Offset: 0x0C [ID-00001bb3]
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	HSBAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HSBAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – HSBAUDLOW[7:0]: High Speed Master Baud Rate Low

HSBAUDLOW non-zero: HSBAUDLOW indicates the SCL low time in High-speed mode according to

$$\mathrm{HSBAUDLOW} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{LOW}} - 1$$

HSBAUDLOW equal to zero: The HSBAUD register is used to time T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$. T_{BUF} is timed by the BAUD register.

Bits 23:16 – HSBAUD[7:0]: High Speed Master Baud Rate

This bit field indicates the SCL high time in High-speed mode according to the following formula. When HSBAUDLOW is zero, T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$ are derived using this formula. T_{BUF} is timed by the BAUD register.

$$\mathrm{HSBAUD} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{HIGH}} - 1$$

Bits 15:8 – BAUDLOW[7:0]: Master Baud Rate Low

If this bit field is non-zero, the SCL low time will be described by the value written.

For more information on how to calculate the frequency, see SERCOM Clock Generation – Baud-Rate Generator.

Bits 7:0 – BAUD[7:0]: Master Baud Rate

This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL.

For more information on how to calculate the frequency, see SERCOM Clock Generation – Baud-Rate Generator.

34.10.4 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x14 [ID-00001bb3]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB: Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Slave on Bus Interrupt Enable bit, which disables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB: Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Master on Bus Interrupt Enable bit, which disables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

34.10.5 Interrupt Enable Clear

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x16 [ID-00001bb3]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB: Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Slave on Bus Interrupt Enable bit, which enables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB: Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Master on Bus Interrupt Enable bit, which enables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

34.10.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18 [ID-00001bb3]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR: Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status bits in the STATUS register. These status bits are LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 1 – SB: Slave on Bus
The Slave on Bus flag (SB) is set when a byte is successfully received in master read mode, i.e., no arbitration lost or bus error occurred during the operation. When this flag is set, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and SB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

Bit 0 – MB: Master on Bus
This flag is set when a byte is transmitted in master write mode. The flag is set regardless of the occurrence of a bus error or an arbitration lost condition. MB is also set when arbitration is lost during sending of NACK in master read mode, or when issuing a start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and MB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

34.10.7 Status

Name: STATUS
Offset: 0x1A [ID-00001bb3]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	MEXTTOUT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
Access	R	R/W	R	R		R	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 10 – LENERR: Transaction Length Error
This bit is set when automatic length is used for a DMA transaction and the slave sends a NACK before ADDR.LEN bytes have been written by the master.

Writing '1' to this bit location will clear STATUS.LENERR. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 9 – SEXTTOUT: Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.

This bit is automatically cleared when writing to the ADDR register.

Writing '1' to this bit location will clear SEXTTOUT. Normal use of the I^2C interface does not require the SEXTTOUT flag to be cleared by this method.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 8 – MEXTTOUT: Master SCL Low Extend Time-Out
This bit is set if a master SCL low time-out occurs.

Writing '1' to this bit location will clear STATUS.MEXTTOUT. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 7 – CLKHOLD: Clock Hold
This bit is set when the master is holding the SCL line low, stretching the I^2C clock. Software should consider this bit when INTFLAG.SB or INTFLAG.MB is set.

This bit is cleared when the corresponding interrupt flag is cleared and the next operation is given.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

This bit is not write-synchronized.

Bit 6 – LOWTOUT: SCL Low Time-Out
This bit is set if an SCL low time-out occurs.

Writing '1' to this bit location will clear this bit. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bits 5:4 – BUSSTATE[1:0]: Bus State
These bits indicate the current I^2C bus state.

When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the IDLE state. The bus state cannot be forced into any other state.

Writing BUSSTATE to idle will set SYNCBUSY.SYSOP.

Value	Name	Description
0x0	UNKNOWN	The bus state is unknown to the I^2C master and will wait for a stop condition to be detected or wait to be forced into an idle state by software
0x1	IDLE	The bus state is waiting for a transaction to be initialized
0x2	OWNER	The I^2C master is the current owner of the bus
0x3	BUSY	Some other I^2C master owns the bus

Bit 2 – RXNACK: Received Not Acknowledge
This bit indicates whether the last address or data packet sent was acknowledged or not.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

This bit is not write-synchronized.

Value	Description
0	Slave responded with ACK.
1	Slave responded with NACK.

Bit 1 – ARBLOST: Arbitration Lost
This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit, or while issuing a start or repeated start condition on the bus. The Master on Bus interrupt flag (INTFLAG.MB) will be set when STATUS.ARBLOST is set.

Writing the ADDR.ADDR register will automatically clear STATUS.ARBLOST.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

This bit is not write-synchronized.

Bit 0 – BUSERR: Bus Error
This bit indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I^2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set BUSERR.

If the I^2C master is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB will be set in addition to BUSERR.

Writing the ADDR.ADDR register will automatically clear the BUSERR flag.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

This bit is not write-synchronized.

34.10.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C [ID-00001bb3]
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP: System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST: Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.10.9 Address

Name: ADDR
Offset: 0x24 [ID-00001bb3]
Reset: 0x0000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TENBITEN	HS	LENEN			ADDR[2:0]		
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:16 – LEN[7:0]: Transaction Length
These bits define the transaction length of a DMA transaction from 0 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order to use DMA.

Bit 15 – TENBITEN: Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 14 – HS: High Speed
This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

Value	Description
0	High-speed transfer disabled.
1	High-speed transfer enabled.

Bit 13 – LENEN: Transfer Length Enable

Value	Description
0	Automatic transfer length disabled.
1	Automatic transfer length enabled.

Bits 10:8 – ADDR[2:0]: Address
When ADDR is written, the consecutive operation will depend on the bus state:

UNKNOWN: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.

BUSY: The I^2C master will await further operation until the bus becomes IDLE.

IDLE: The I^2C master will issue a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low, and STATUS.CLKHOLD and INTFLAG.MB are set.

OWNER: A repeated start sequence will be performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set.

STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.

The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the master logic to perform any bus protocol related operations.

The I^2C master control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.

34.10.10 Data

Name: DATA
Offset: 0x18 [ID-00001bb3]
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0]: Data
The master data register I/O location (DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the master (STATUS.CLKHOLD is set). An exception is reading the last data byte after the stop condition has been sent.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

34.10.11 Debug Control

Name: DBGCTRL
Offset: 0x30 [ID-00001bb3]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP: Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

35. TC – Timer/Counter

35.1 Overview

There are up to four TC peripheral instances.

Each TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events, or clock pulses. The counter, together with the compare/capture channels, can be configured to timestamp input events or IO pin edges, allowing for capturing of frequency and/or pulse width.

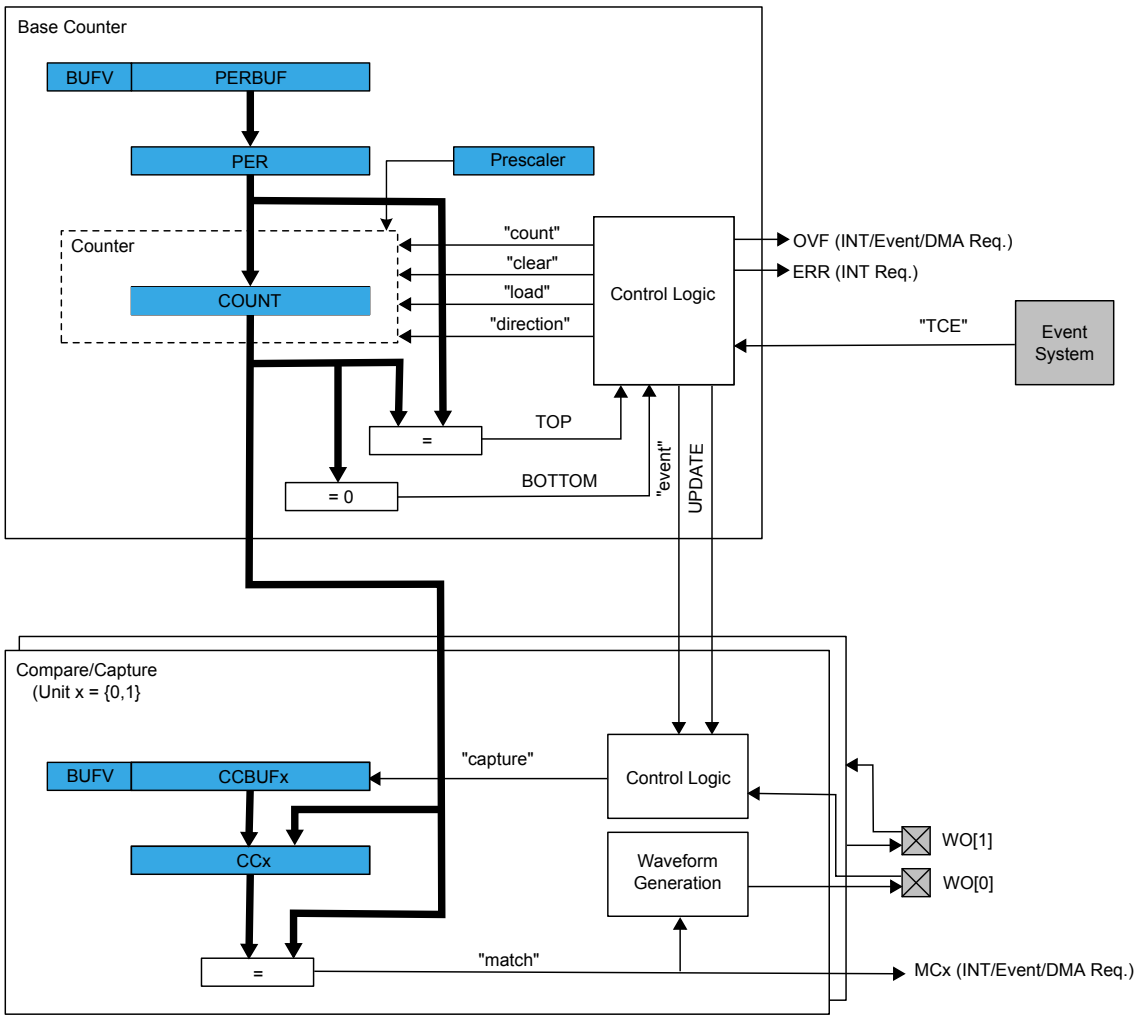
A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

35.2 Features

- Selectable configuration
  - 8-, 16- or 32-bit TC operation, with compare/capture channels
- 2 compare/capture channels (CC) with:
  - Double buffered timer period setting (in 8-bit mode only)
  - Double buffered compare channel
- Waveform generation
  - Frequency generation
  - Single-slope pulse-width modulation
- Input capture
  - Event / IO pin edge capture
  - Frequency capture
  - Pulse-width capture
  - Time-stamp capture
- One input event
- Interrupts/output events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal prescaler
- DMA support

35.3 Block Diagram

Figure 35-1. Timer/Counter Block Diagram



35.4 Signal Description

Table 35-1. Signal Description for TC.

Signal Name	Type	Description
WO[1:0]	Digital output	Waveform output
	Digital input	Capture input

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

35.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

35.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

Related Links

PORT: IO Pin Controller

35.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

35.5.3 Clocks

The TC bus clocks (CLK_TCx_APB) can be enabled and disabled in the Main Clock Module. The default state of CLK_TCx_APB can be found in the *Peripheral Clock Masking*.

The generic clocks (GCLK_TCx) are asynchronous to the user interface clock (CLK_TCx_APB). Due to this asynchronicity, accessing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Note that TC0 and TC1 share a peripheral clock channel, as do TC2 and TC3. For this reason they cannot be set to different clock frequencies.

Related Links

Peripheral Clock Masking

35.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller

35.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

35.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System

35.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

35.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)
- Count register (COUNT)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture Value registers and Compare/Capture Value Buffer registers (CCx, CCBUFx)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

35.5.9 Analog Connections

Not applicable.

35.6 Functional Description

35.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 35-2. Timer/Counter Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Operations.
ZERO	The counter is ZERO when it contains all zeroes
MAX	The counter reaches MAX when it contains all ones
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source
Counter	The clock control is handled externally (e.g. counting external events)
CC	For compare operations, the CC are referred to as “compare channels” For capture operations, the CC are referred to as “capture channels.”

Each TC instance has up to two compare/capture channels (CC0 and CC1).

The counter in the TC can either count events from the Event System, or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

The Counter register (COUNT), compare and capture registers with buffers (CCx and CCBUFx) can be configured as 8-, 16- or 32-bit registers, with according MAX values. Mode settings determine the maximum range of the counter. Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

In 8-bit mode, Period Value (PER) and Period Buffer Value (PERBUF) registers are also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match the TC can request DMA transactions, or generate interrupts or events for the Event System.

In compare operation, the counter value is continuously compared to the values in the CCx registers. In case of a match the TC can request DMA transactions, or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements, or to capture selectable edges from an IO pin or internal event from Event System.

35.6.2 Basic Operation

35.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TC is disabled (CTRLA.ENABLE =0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Drive Control register (DRVCTRL)
- Wave register (WAVE)
- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before enabling the TC, the peripheral must be configured by the following steps:

1. Enable the TC bus clock (CLK_TCx_APB).
2. Select 8-, 16- or 32-bit counter mode via the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.
3. Select one wave generation operation in the Waveform Generation Operation bit group in the WAVE register (WAVE.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled via the Prescaler bit group in the Control A register (CTRLA.PRESCALER).
    – If the prescaler is used, select a prescaler synchronization operation via the Prescaler and Counter Synchronization bit group in the Control A register (CTRLA.PRESYNC).

5. If desired, select one-shot operation by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT).
6. If desired, configure the counting direction 'down' (starting from the TOP value) by writing a '1' to the Counter Direction bit in the Control B register (CTRLBSET.DIR).
7. For capture operation, enable the individual channels to capture in the Capture Channel x Enable bit group in the Control A register (CTRLA.CAPTEN).
8. If desired, enable inversion of the waveform output or IO pin input signal for individual channels via the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

35.6.2.2 Enabling, Disabling, and Resetting

The TC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TC is disbled by writing a zero to CTRLA.ENABLE.

The TC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TC, except DBGCTRL, will be reset to their initial state. Refer to the CTRLA register for details.

The TC should be disabled before the TC is reset in order to avoid undefined behavior.

35.6.2.3 Prescaler Selection

The GCLK_TCx is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

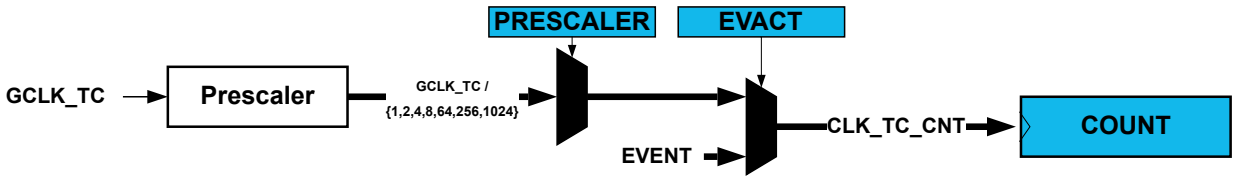
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TC_CNT.

Figure 35-2. Prescaler



35.6.2.4 Counter Mode

The counter mode is selected by the Mode bit group in the Control A register (CTRLA.MODE). By default, the counter is enabled in the 16-bit counter resolution. Three counter resolutions are available:

- COUNT8: The 8-bit TC has its own Period Value and Period Buffer Value registers (PER and PERBUF).
- COUNT16: 16-bit is the default counter mode. There is no dedicated period register in this mode.
- COUNT32: This mode is achieved by pairing two 16-bit TC peripherals. TC0 is paired with TC1, and TC2 is paired with TC3. TC4 does not support 32-bit resolution.
  When paired, the TC peripherals are configured using the registers of the even-numbered TC (TC0 or TC2 respectively). The odd-numbered partner (TC1 or TC3 respectively) will act as slave, and the Slave bit in the Status register (STATUS.SLAVE) will be set. The register values of a slave will

not reflect the registers of the 32-bit counter. Writing to any of the slave registers will not affect the 32-bit counter. Normal access to the slave COUNT and CCx registers is not allowed.

35.6.2.5 Counter Operations

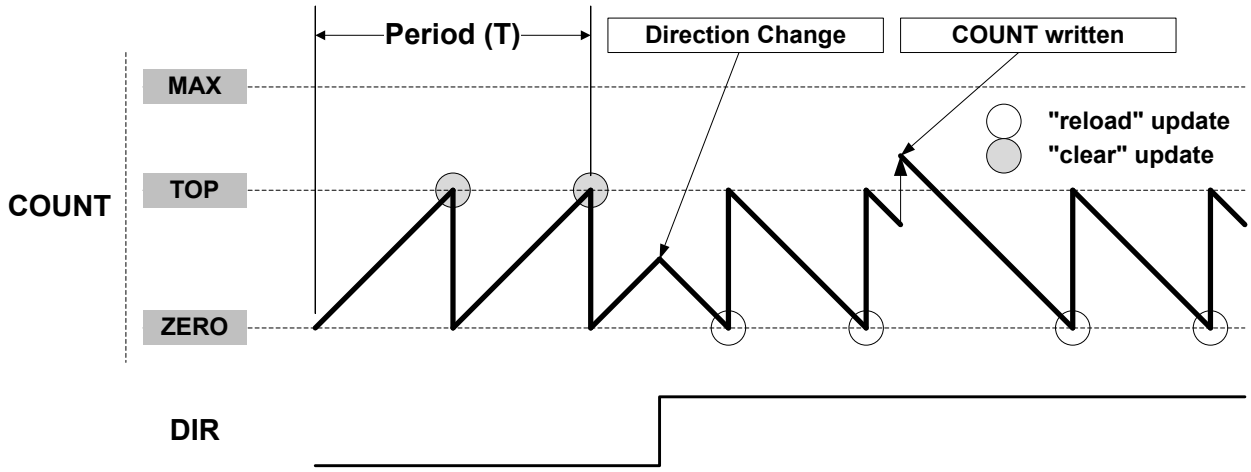
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TC_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR=1. The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When it is counting down, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e. a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. When starting the TC, the COUNT value will be either ZERO or TOP (depending on the counting direction set by CTRLBSET.DIR or CTRLBCLR.DIR), unless a different value has been written to it, or the TC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See also the figure below.

Figure 35-3. Counter Operation



Due to asynchronous clock domains, the internal counter settings are written when the synchronization is complete. Normal operation must be used when using the counter as timer base for the capture channels.

Stop Command and Event Action

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter will be loaded with the starting value (ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR). All waveforms are cleared and the Stop bit in the Status register is set (STATUS.STOP).

Re-Trigger Command and Event Action
A re-trigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER), or from event when a re-trigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

Count Event Action
The TC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR). The count event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x2, COUNT).

Start Event Action
The TC can start counting operation on an event when previously stopped. In this configuration, the event has no effect if the counter is already counting. When the peripheral is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

The Start TC on Event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x3, START).

35.6.2.6 Compare Operations

By default, the Compare/Capture channel is configured for compare operations.

When using the TC and the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare Buffer (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a forced update command (CTRLBSET.CMD=UPDATE). For further details, refer to Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Waveform Output Operations
The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Output Waveform x Invert Enable bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TC_CNT (see Normal Frequency Operation). An interrupt/and or event can be generated on comparison match if enabled. The same condition generates a DMA request.

There are four waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

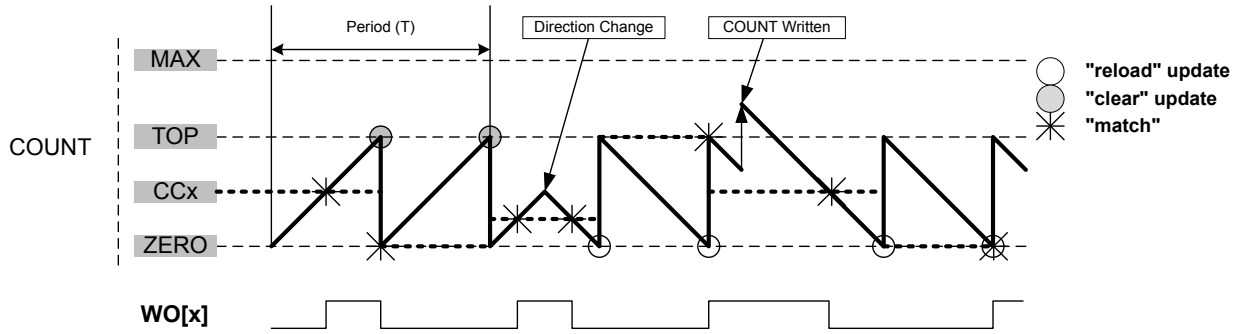
- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

When using NPWM or NFRQ configuration, the TOP will be determined by the counter resolution. In 8-bit counter mode, the Period register (PER) is used as TOP, and the TOP can be changed by writing to the PER register. In 16- and 32-bit counter mode, TOP is fixed to the maximum (MAX) value of the counter.

Normal Frequency Generation (NFRQ)

For Normal Frequency Generation, the period time (T) is controlled by the period register (PER) for 8-bit counter mode and MAX for 16- and 32-bit mode. The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (INTFLAG.MCx) will be set.

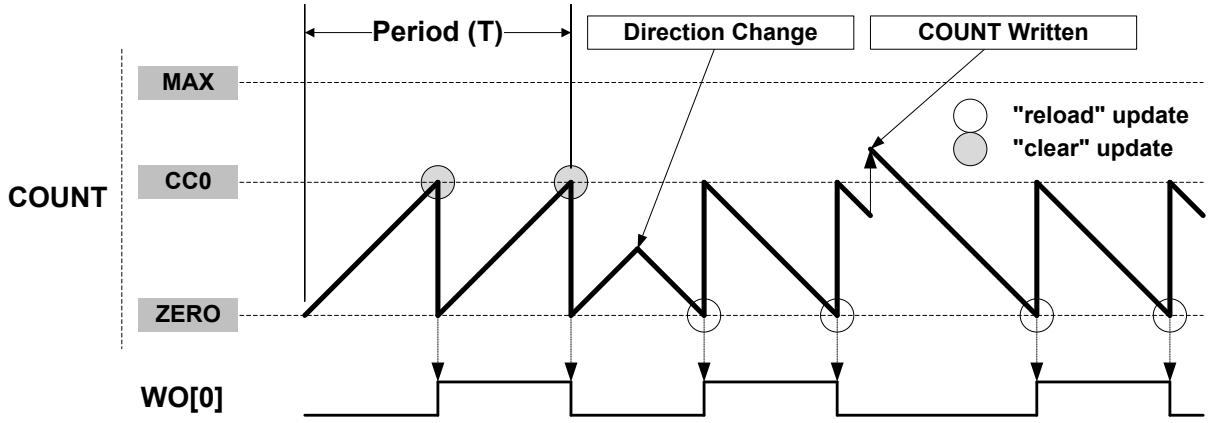
Figure 35-4. Normal Frequency Operation



Match Frequency Generation (MFRQ)

For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER or MAX. WO[0] toggles on each update condition.

Figure 35-5. Match Frequency Operation



Normal Pulse-Width Modulation Operation (NPWM)

NPWM uses single-slope PWM generation.

For single-slope PWM generation, the period time (T) is controlled by the TOP value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare

match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency (f_{PWM_SS}) depends on TOP value and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

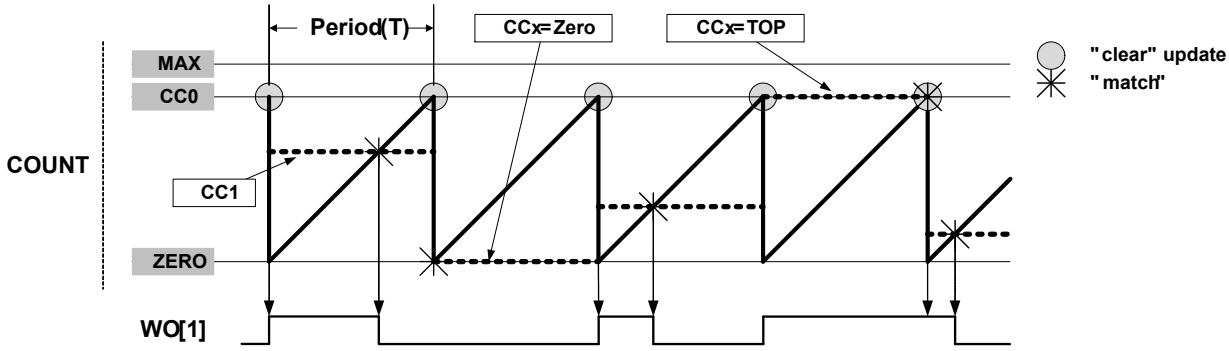
$$f_{PWM_SS} = \frac{f_{GCLK_TC}}{N(TOP+1)}$$

Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Match Pulse-Width Modulation Operation (MPWM)

In MPWM, the output of WO[1] is depending on CC1 as shown in the figure below. On on every overflow/ underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

Figure 35-6. Match PWM Operation



The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 35-3. Counter Update and Overflow Event/interrupt Conditions in TC

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See description above.		TOP	ZERO
MPWM	Single-slope PWM	CC0	TOP/ ZERO	Toggle	Toggle	TOP	ZERO

Related Links

PORT: IO Pin Controller

35.6.2.7 Double Buffering

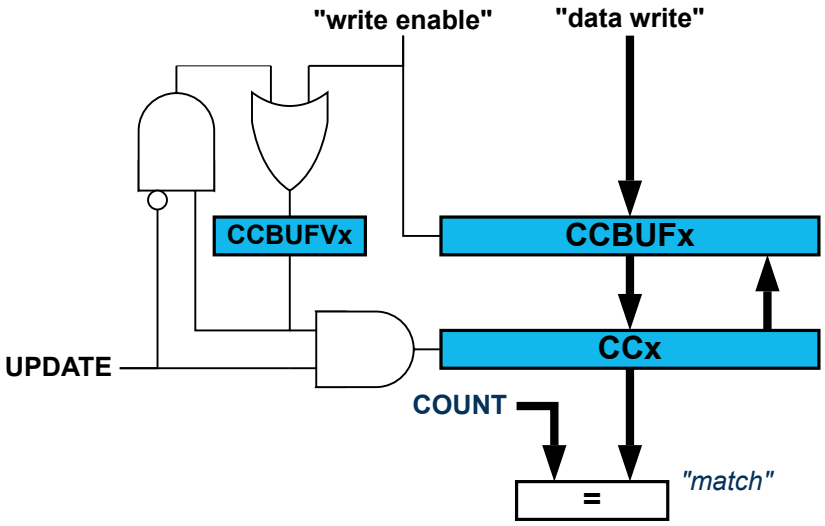
The Compare Channels (CCx) registers, and the Period (PER) register in 8-bit mode are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As long as the respective buffer valid status flag (PERBUFV or CCBUFVx) are set to '1', related

syncbusy bits are set (SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and access to the respective PER or CCx register is invalid.

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the buffer valid flags bit in the STATUS register are automatically cleared by hardware.
Note: The software update command (CTRLBSET.CMD=0x3) is acting independently of the LUPD value.

A compare register is double buffered as in the following figure.

Figure 35-7. Compare Channel Double Buffering



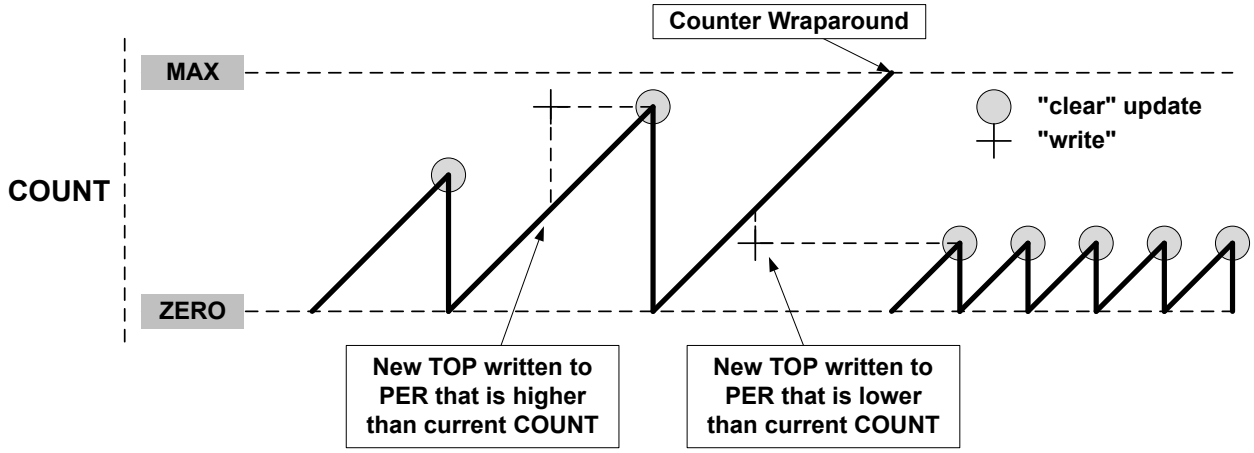
Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new TOP value to the Period register (PER or CC0, depending on the waveform generation mode), any period update on registers (PER or CCx) is effective after the synchronization delay, whatever double buffering enabling is.

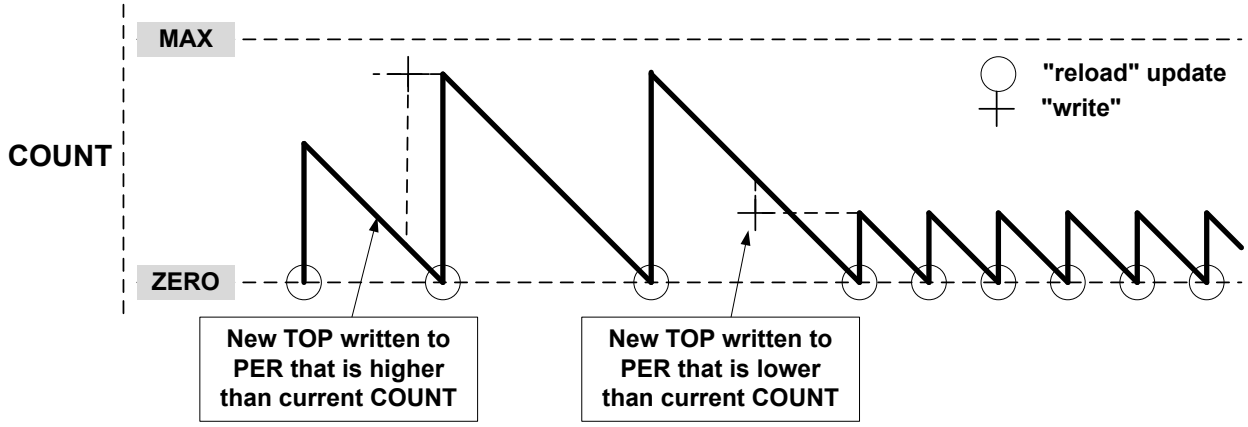
Figure 35-8. Unbuffered Single-Slope Up-Counting Operation



A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 35-8.

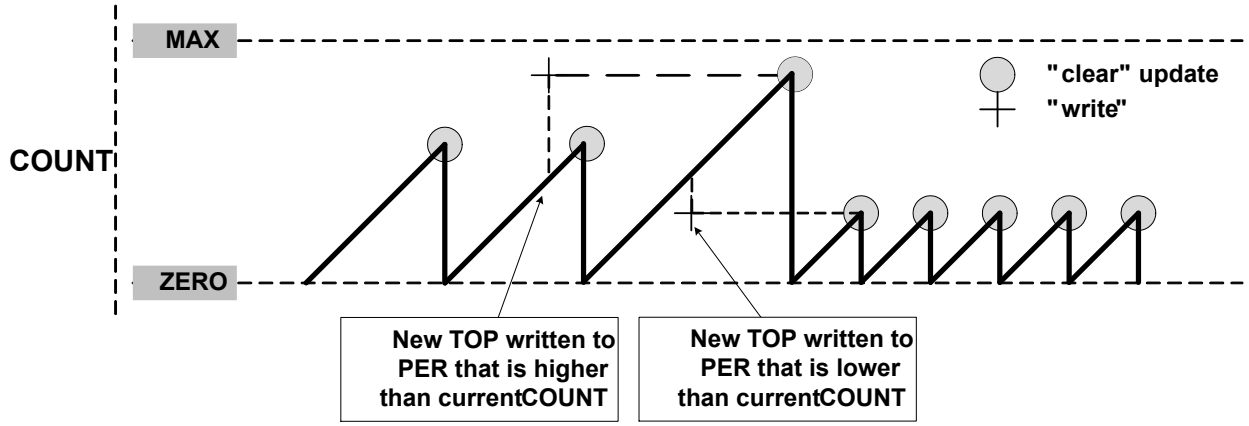
COUNT and TOP are continuously compared, so when a new TOP value that is lower than current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 35-9. Unbuffered Single-Slope Down-Counting Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 35-10. This prevents wraparound and the generation of odd waveforms.

Figure 35-10. Changing the Period Using Buffering



35.6.2.8 Capture Operations

To enable and use capture operations, the corresponding Capture Channel x Enable bit in the Control A register (CTRLA.CAPTENx) must be written to '1'.

A capture trigger can be provided by input event line TC_EV or by asynchronous IO pin WO[x] for each capture channel or by a TC event. To enable the capture from input event line, Event Input Enable bit in the Event Control register (EVCTRL.TCEI) must be written to '1'. To enable the capture from the IO pin, the Capture On Pin x Enable bit in CTRLA register (CTRLA.COPENx) must be written to '1'.

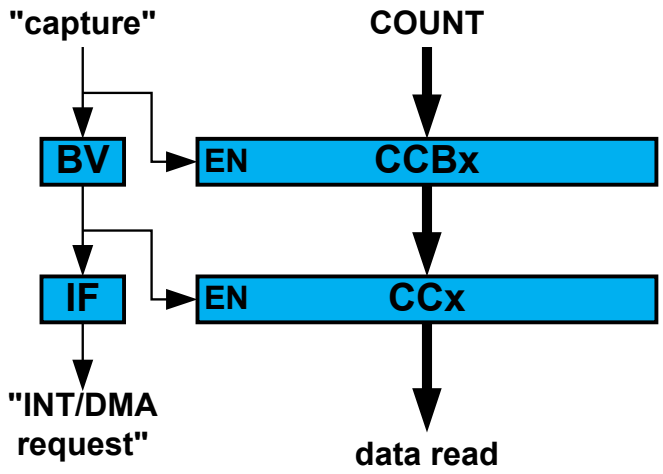
Note: The RETRIGGER, COUNT and START event actions are available only on an event from the Event System.

By default, a capture operation is done when a rising edge is detected on the input signal. Capture on falling edge is available, its activation is depending on the input source:

- When the channel is used with a IO pin, write a '1' to the corresponding Invert Enable bit in the Drive Control register (DRVCTRL.INVENx).
- When the channel is counting events from the Event System, write a '1' to the TC Event Input Invert Enable bit in Event Control register (EVCTRL.TCINV).

For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. CCBUFx register value can't be read, all captured data must be read from CCx register.

Figure 35-11. Capture Double Buffering

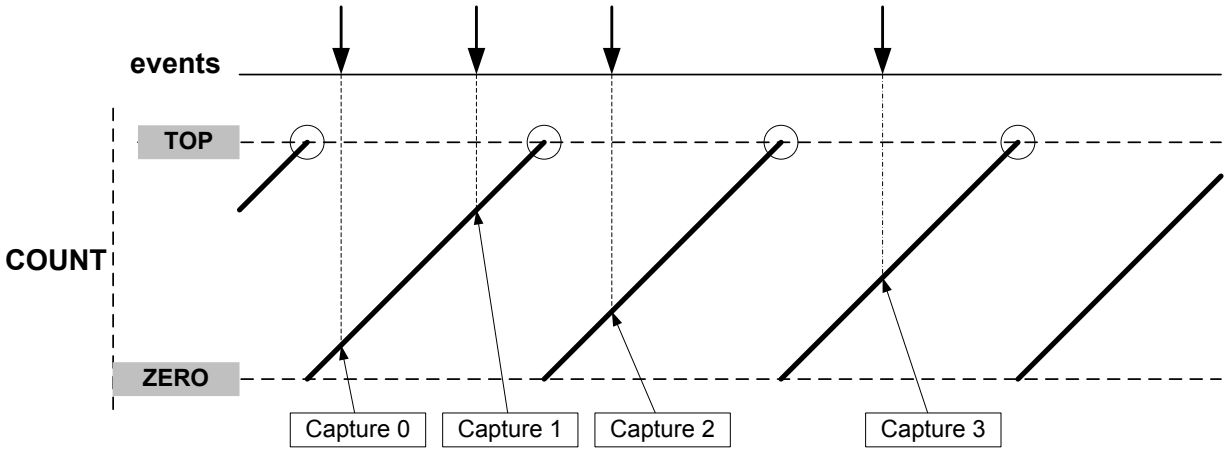


For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. CCBUFx register value can't be read, all captured data must be read from CCx register.

Event Capture Action

The compare/capture channels can be used as input capture channels to capture events from the Event System or from the corresponding IO pin, and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 35-12. Input Capture Timing



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

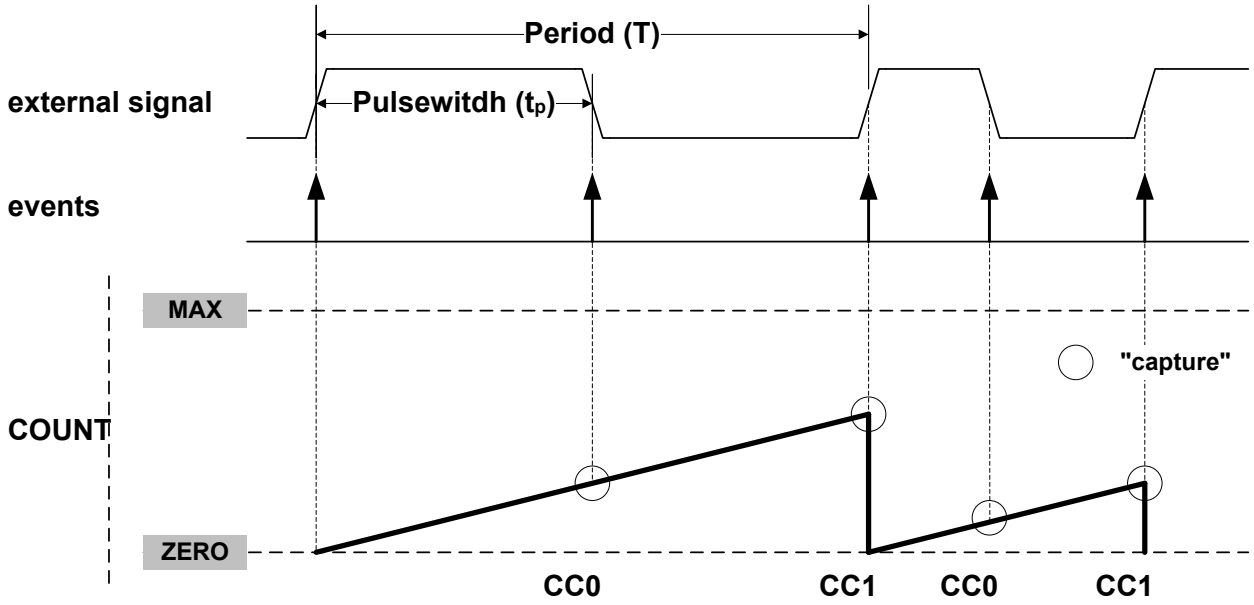
Period and Pulse-Width (PPW) Capture Action

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

$$f = \frac{1}{T}$$

$$\text{dutyCycle} = \frac{t_p}{T}$$

Figure 35-13. PWP Capture



Selecting PWP in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and the other one on the falling edge. The period *T* will be captured into CC1 and the pulse width t_p in CC0. EVCTRL.EVACT=PPW (period and pulse-width)offers identical functionality, but will capture *T* into CC0 and t_p into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCINV=1, the wraparound will happen on the falling edge. This also be for DRVCTRL.INVENx if pin capture is enabled.

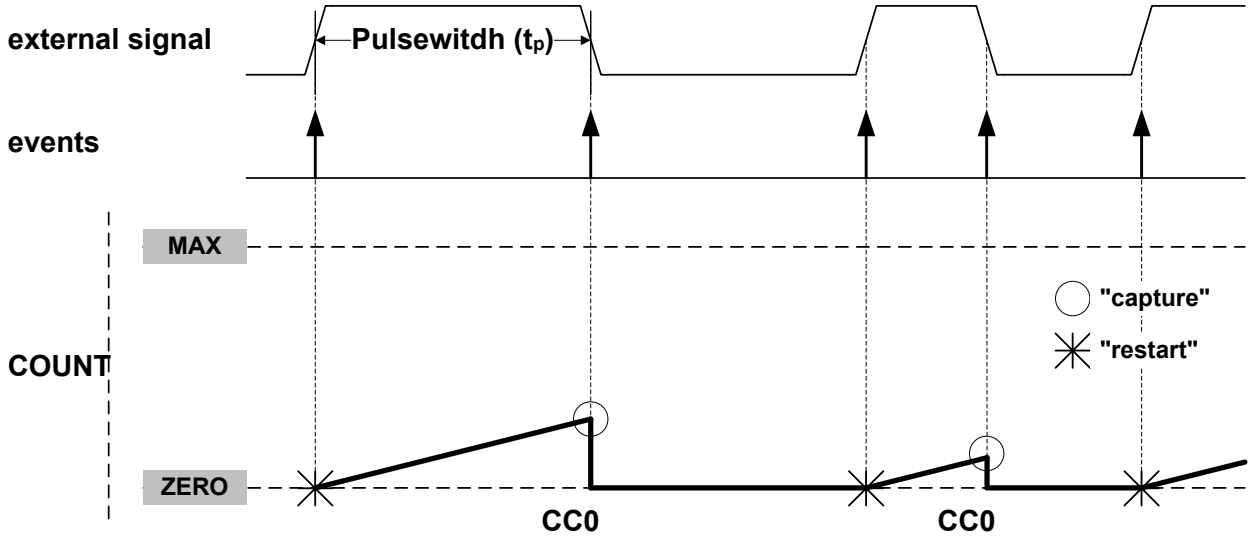
The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx=1). If not, the capture action is ignored and the channel is enabled in compare mode of operation. Consequently, both channels must be enabled in order to fully characterize the input.

Pulse-Width Capture Action

The TC performs the input capture on the falling edge of the input signal. When the edge is detected, the counter value is cleared and the TC stops counting. When a rising edge is detected on the input signal, the counter restarts the counting operation. To enable the operation on opposite edges, the input signal to capture must be inverted (refer to DRVCTRL.INVEN or EVCTRL.TCEINV).

Figure 35-14. Pulse-Width Capture on Channel 0



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

35.6.3 Additional Features

35.6.3.1 One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is automatically set and the waveform outputs are set to zero.

One-shot operation is enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT), and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TC will count until an overflow or underflow occurs and stops counting operation. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event, or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

35.6.3.2 Time-Stamp Capture

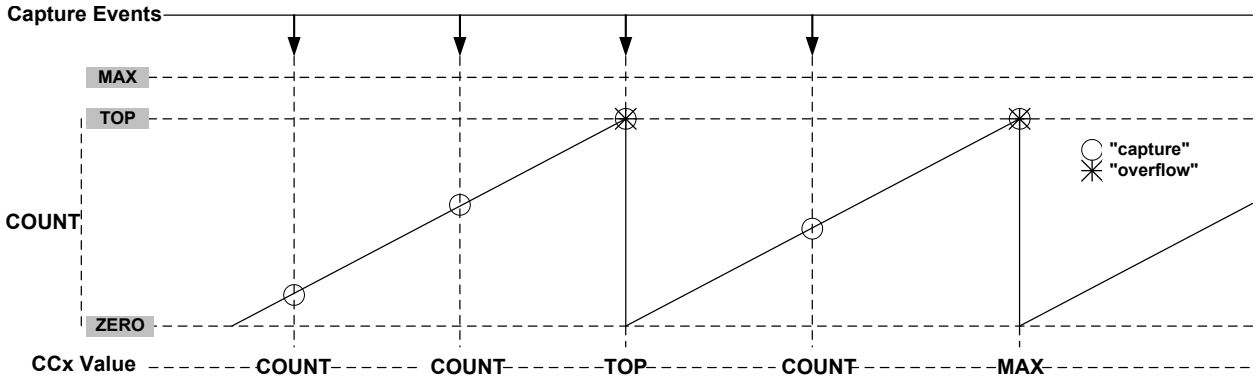
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 35-15. Time-Stamp



35.6.4 DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or re-trigger) is detected, the request is cleared by hardware on DMA acknowledge.
- Match or Capture Channel x (MCx): for a compare channel, the request is set on each compare match detection, the request is cleared by hardware on DMA acknowledge. For a capture channel, the request is set when valid data is present in the CCx register, and cleared when CCx register is read.

35.6.5 Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)
- Capture Overflow Error (ERR)

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the TC is reset. See INTFLAG for details on how to clear interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

35.6.6 Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Disable event action (OFF)
- Start TC (START)
- Re-trigger TC (RETRIGGER)
- Count on event (COUNT)
- Capture time stamp (STAMP)
- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. For further details on how configuring the asynchronous events, refer to *EVSYS - Event System*.

Related Links

EVSYS – Event System

35.6.7 Sleep Mode Operation

The TC can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

If the On Demand bit in the Control A register (CTRLA.ONDEMAND) is written to '1', the module stops requesting its peripheral clock when the STOP bit in STATUS register (STATUS.STOP) is set to '1'. When a re-trigger or start condition is detected, the TC requests the clock before the operation starts.

35.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Capture Channel Buffer Valid bit in STATUS register (STATUS.CCBUFVx)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Channel x Compare/Capture Value and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD).

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

35.7 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

35.7.1 Register Summary - 8-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x01		15:8					ALOCK	PRESCALER[2:0]		
0x02		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
0x03		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
0x07		15:8			MCEOx	MCEOx				OVFEO
0x08	INTENCLR	7:0				MCx			ERR	OVF
0x09	INTENSET	7:0				MCx			ERR	OVF
0x0A	INTFLAG	7:0				MCx			ERR	OVF
0x0B	STATUS	7:0				CCBUFVx	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0								INVENx
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
0x15 ... 0x1A	Reserved									
0x1B	PER	7:0	PER[7:0]							
0x1C	CC0	7:0	CC[7:0]							
0x1D	CC1	7:0	CC[7:0]							
0x1E ... 0x2E	Reserved									
0x2F	PERBUF	7:0	PERBUF[7:0]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31	CCBUF1	7:0	CCBUF[7:0]							

35.7.1.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00001cd8]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx: Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx: Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK: Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.

This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0]: Prescaler
These bits select the counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2

Value	Name	Description
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND: Clock On Demand
This bit selects the clock requirements when the TC is stopped.

In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.

This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY: Run in Standby
This bit is used to keep the TC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0]: Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Bits 3:2 – MODE[1:0]: Timer Counter Mode
These bits select the counter mode.

These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

35.7.1.2 Control B Clear

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Name: CTRLBCLR
Offset: 0x04 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.1.3 Control B Set

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Name: CTRLBSET
Offset: 0x05 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.1.4 Event Control

Name: EVCTRL
Offset: 0x06 [ID-00001cd8]
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEOx	MCEOx				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 13,12 – MCEOx: Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO: Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI: TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV: TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0]: Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

35.7.1.5 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x08 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR: Error Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

35.7.1.6 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x09 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

35.7.1.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A [ID-00001cd8]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR: Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

35.7.1.8 Status

Name: STATUS
Offset: 0x0B [ID-00001cd8]
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx: Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV: Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE: Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP: Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

35.7.1.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0]: Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in Waveform Output Operations.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

35.7.1.10 Driver Control

Name: DRVCTRL
Offset: 0x0D [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								INVENx
Access								R/W
Reset								0

Bit 0 – INVENx: Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

35.7.1.11 Debug Control

Name: DBGCTRL
Offset: 0x0F [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

35.7.1.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10 [ID-00001cd8]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – CCx: Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER: PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT: COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS: STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.

This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB: CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.

This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

35.7.1.13 Counter Value, 8-bit Mode

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Name: COUNT
Offset: 0x14 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – COUNT[7:0]: Counter Value
These bits contain the current counter value.

35.7.1.14 Period Value, 8-bit Mode

Name: PER
Offset: 0x1B [ID-00001cd8]
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PER[7:0]: Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

35.7.1.15 Channel x Compare/Capture Value, 8-bit Mode

Name: CCx
Offset: 0x1C + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CC[7:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 8-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

35.7.1.16 Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F [ID-00001cd8]
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PERBUF[7:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

35.7.1.17 Channel x Compare Buffer Value, 8-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CCBUF[7:0]: Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

35.7.2 Register Summary - 16-bit Mode

Offset	Name	Bit Pos.									
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST	
0x01		15:8					ALOCK	PRESCALER[2:0]			
0x02		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0	
0x03		31:24									
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR	
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR	
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]			
0x07		15:8			MCEOx	MCEOx				OVFEO	
0x08	INTENCLR	7:0				MCx			ERR	OVF	
0x09	INTENSET	7:0				MCx			ERR	OVF	
0x0A	INTFLAG	7:0				MCx			ERR	OVF	
0x0B	STATUS	7:0				CCBUFVx	PERBUFV		SLAVE	STOP	
0x0C	WAVE	7:0							WAVEGEN[1:0]		
0x0D	DRVCTRL	7:0								INVENx	
0x0E	Reserved										
0x0F	DBGCTRL	7:0								DBGRUN	
0x10	SYNCBUSY	7:0		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST	
0x11		15:8									
0x12		23:16									
0x13		31:24									
0x14	COUNT	7:0	COUNT[7:0]								
0x15		15:8	COUNT[15:8]								
0x16 ... 0x19	Reserved										
0x1A	PER	7:0	PER[7:0]								
0x1B		15:8	PER[15:8]								
0x1C	CC0	7:0	CC[7:0]								
0x1D		15:8	CC[15:8]								
0x1E	CC1	7:0	CC[7:0]								
0x1F		15:8	CC[15:8]								
0x20 ... 0x2D	Reserved										
0x2E	PERBUF	7:0	PERBUF[7:0]								
0x2F		15:8	PERBUF[15:8]								
0x30	CCBUF0	7:0	CCBUF[7:0]								
0x31		15:8	CCBUF[15:8]								
0x32	CCBUF1	7:0	CCBUF[7:0]								
0x33		15:8	CCBUF[15:8]								

35.7.2.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00001cd8]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx: Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx: Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK: Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.

This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0]: Prescaler
These bits select the counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND: Clock On Demand
This bit selects the clock requirements when the TC is stopped.

In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.

This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY: Run in Standby
This bit is used to keep the TC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0]: Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Bits 3:2 – MODE[1:0]: Timer Counter Mode
These bits select the counter mode.

These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode

Value	Name	Description
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

35.7.2.2 Control B Clear

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Name: CTRLBCLR
Offset: 0x04 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.2.3 Control B Set

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Name: CTRLBSET
Offset: 0x05 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.2.4 Event Control

Name: EVCTRL
Offset: 0x06 [ID-00001cd8]
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEOx	MCEOx				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 13,12 – MCEOx: Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO: Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI: TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV: TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0]: Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

35.7.2.5 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x08 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR: Error Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

35.7.2.6 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x09 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

35.7.2.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A [ID-00001cd8]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR: Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

35.7.2.8 Status

Name: STATUS
Offset: 0x0B [ID-00001cd8]
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx: Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV: Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE: Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP: Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

35.7.2.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0]: Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in Waveform Output Operations.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

35.7.2.10 Driver Control

Name: DRVCTRL
Offset: 0x0D [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								INVENx
Access								R/W
Reset								0

Bit 0 – INVENx: Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

35.7.2.11 Debug Control

Name: DBGCTRL
Offset: 0x0F [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

35.7.2.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10 [ID-00001cd8]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – CCx: Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER: PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT: COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS: STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.

This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB: CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.

This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

35.7.2.13 Counter Value, 16-bit Mode

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Name: COUNT
Offset: 0x14 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0]: Counter Value
These bits contain the current counter value.

35.7.2.14 Period Value, 16-bit Mode

Name: PER
Offset: 0x1A [ID-00001cd8]
Reset: 0xFFFF
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 15:0 – PER[15:0]: Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

35.7.2.15 Channel x Compare/Capture Value, 16-bit Mode

Name: CCx
Offset: 0x1C + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CC[15:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 16-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

35.7.2.16 Period Buffer Value, 16-bit Mode

Name: PERBUF
Offset: 0x2E [ID-00001cd8]
Reset: 0xFFFF
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 15:0 – PERBUF[15:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

35.7.2.17 Channel x Compare Buffer Value, 16-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CCBUF[15:0]: Channel x Compare Buffer Value

These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

35.7.3 Register Summary - 32-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x01		15:8					ALOCK	PRESCALER[2:0]		
0x02		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
0x03		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
0x07		15:8			MCEOx	MCEOx				OVFEO
0x08	INTENCLR	7:0				MCx			ERR	OVF
0x09	INTENSET	7:0				MCx			ERR	OVF
0x0A	INTFLAG	7:0				MCx			ERR	OVF
0x0B	STATUS	7:0				CCBUFVx	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0								INVENx
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
0x15		15:8	COUNT[15:8]							
0x16		23:16	COUNT[23:16]							
0x17		31:24	COUNT[31:24]							
0x18 ... 0x19	Reserved									
0x1A	PER	7:0	PER[7:0]							
0x1B		15:8	PER[15:8]							
0x1C		23:16	PER[23:16]							
0x1D		31:24	PER[31:24]							
0x1C	CC0	7:0	CC[7:0]							
0x1D		15:8	CC[15:8]							
0x1E		23:16	CC[23:16]							
0x1F		31:24	CC[31:24]							
0x20	CC1	7:0	CC[7:0]							
0x21		15:8	CC[15:8]							
0x22		23:16	CC[23:16]							
0x23		31:24	CC[31:24]							
0x26 ... 0x2B	Reserved									
0x2C	PERBUF	7:0	PERBUF[7:0]							
0x2D		15:8	PERBUF[15:8]							
0x2E		23:16	PERBUF[23:16]							

Offset	Name	Bit Pos.								
0x2F		31:24	PERBUF[31:24]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31		15:8	CCBUF[15:8]							
0x32		23:16	CCBUF[23:16]							
0x33		31:24	CCBUF[31:24]							
0x34	CCBUF1	7:0	CCBUF[7:0]							
0x35		15:8	CCBUF[15:8]							
0x36		23:16	CCBUF[23:16]							
0x37		31:24	CCBUF[31:24]							

35.7.3.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00001cd8]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx: Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx: Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK: Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.

This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0]: Prescaler
These bits select the counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND: Clock On Demand
This bit selects the clock requirements when the TC is stopped.

In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.

This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY: Run in Standby
This bit is used to keep the TC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0]: Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Bits 3:2 – MODE[1:0]: Timer Counter Mode
These bits select the counter mode.

These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

35.7.3.2 Control B Clear

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Name: CTRLBCLR
Offset: 0x04 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.3.3 Control B Set

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Name: CTRLBSET
Offset: 0x05 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.7.3.4 Event Control

Name: EVCTRL
Offset: 0x06 [ID-00001cd8]
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEOx	MCEOx				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 13,12 – MCEOx: Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO: Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/ underflow.

Bit 5 – TCEI: TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV: TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0]: Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

35.7.3.5 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x08 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR: Error Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

35.7.3.6 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x09 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

35.7.3.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A [ID-00001cd8]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
				MCx			ERR	OVF
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – MCx: Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR: Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

35.7.3.8 Status

Name: STATUS
Offset: 0x0B [ID-00001cd8]
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx: Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV: Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE: Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP: Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

35.7.3.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0]: Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in Waveform Output Operations.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

35.7.3.10 Driver Control

Name: DRVCTRL
Offset: 0x0D [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								INVENx
Access								R/W
Reset								0

Bit 0 – INVENx: Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

35.7.3.11 Debug Control

Name: DBGCTRL
Offset: 0x0F [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

35.7.3.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10 [ID-00001cd8]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – CCx: Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER: PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT: COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS: STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.

This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB: CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.

This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

35.7.3.13 Counter Value, 32-bit Mode

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Name: COUNT
Offset: 0x14 [ID-00001cd8]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0]: Counter Value
These bits contain the current counter value.

35.7.3.14 Period Value, 32-bit Mode

Name: PER
Offset: 0x1A [ID-00001cd8]
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PER[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 31:0 – PER[31:0]: Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

35.7.3.15 Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x1C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CC[31:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

35.7.3.16 Period Buffer Value, 32-bit Mode

Name: PERBUF
Offset: 0x2C [ID-00001cd8]
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	PERBUF[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PERBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 31:0 – PERBUF[31:0]: Period Buffer Value

These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

35.7.3.17 Channel x Compare Buffer Value, 32-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CCBUF[31:0]: Channel x Compare Buffer Value

These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

36. TCC – Timer/Counter for Control Applications

36.1 Overview

The device provides one instance of the Timer/Counter for Control applications (TCC) peripheral.

Each TCC instance consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter together with the compare/capture channels can be configured to time stamp input events, allowing capture of frequency and pulse-width. It can also perform waveform generation such as frequency generation and pulse-width modulation.

Waveform extensions are intended for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. They allow for low- and high-side output with optional dead-time insertion. Waveform extensions can also generate a synchronized bit pattern across the waveform output pins. The fault options enable fault protection for safe and deterministic handling, disabling and/or shut down of external drivers.

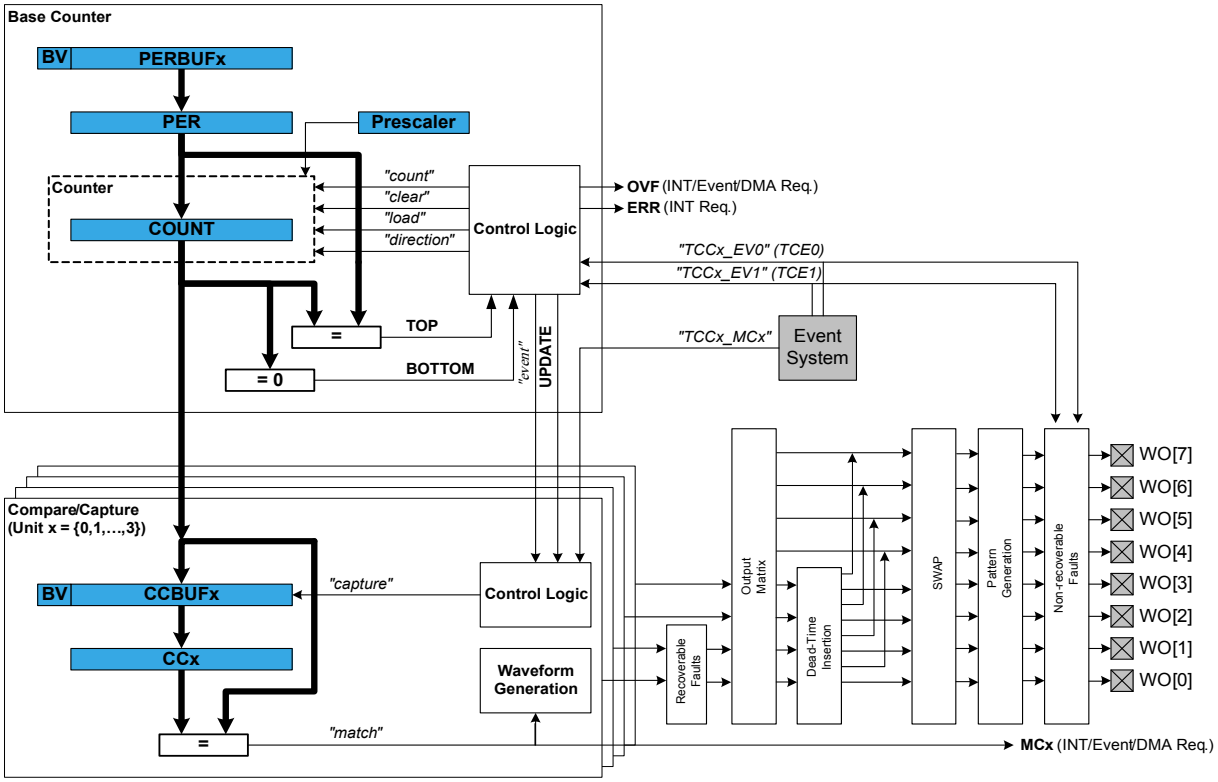
36.2 Features

- Up to four compare/capture channels (CC) with:
  - Double buffered period setting
  - Double buffered compare or capture channel
  - Circular buffer on period and compare channel registers
- Waveform generation:
  - Frequency generation
  - Single-slope pulse-width modulation (PWM)
  - Dual-slope pulse-width modulation with half-cycle reload capability
- Input capture:
  - Event capture
  - Frequency capture
  - Pulse-width capture
- Waveform extensions:
  - Configurable distribution of compare channels outputs across port pins
  - Low- and high-side output with programmable dead-time insertion
  - Waveform swap option with double buffer support
  - Pattern generation with double buffer support
  - Dithering support
- Fault protection for safe disabling of drivers:
  - Two recoverable fault sources
  - Two non-recoverable fault sources
  - Debugger can be source of non-recoverable fault
- Input events:
  - Two input events for counter
  - One input event for each channel
- Output events:

  - Three output events (Count, Re-Trigger and Overflow) available for counter
  - One Compare Match/Input Capture event output for each channel
- Interrupts:
  - Overflow and Re-Trigger interrupt
  - Compare Match/Input Capture interrupt
  - Interrupt on fault detection

36.3 Block Diagram

Figure 36-1. Timer/Counter for Control Applications - Block Diagram



36.4 Signal Description

Pin Name	Type	Description
TCC/WO[0]	Digital output	Compare channel 0 waveform output
TCC/WO[1]	Digital output	Compare channel 1 waveform output
…	...	...
TCC/WO[WO_NUM-1]	Digital output	Compare channel n waveform output

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

36.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

36.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

Related Links

PORT: IO Pin Controller

36.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

36.5.3 Clocks

The TCC bus clock (CLK_TCCx_APB, with x instance number of the TCCx) is enabled by default, and can be enabled and disabled in the Main Clock.

A generic clock (GCLK_TCCx) is required to clock the TCC. This clock must be configured and enabled in the generic clock controller before using the TCC.

The generic clocks (GCLK_TCCx) are asynchronous to the bus clock (CLK_TCCx_APB). Due to this asynchronicity, writing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

Peripheral Clock Masking

GCLK - Generic Clock Controller

36.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller

36.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

36.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System

36.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Refer to DBGCTRL register for details.

36.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag register (INTFLAG)
- Status register (STATUS)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture and Compare/Capture Buffer registers (CCx, CCBUFx)
- Control Waveform register (WAVE)
- Pattern Generation Value and Pattern Generation Value Buffer registers (PATT, PATTBUF)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

36.5.9 Analog Connections

Not applicable.

36.6 Functional Description

36.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 36-1. Timer/Counter for Control Applications - Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Generation Operations.
ZERO	The counter reaches ZERO when it contains all zeroes.
MAX	The counter reaches maximum when it contains all ones.
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g. counting external events).
CC	For compare operations, the CC are referred to as "compare channels." For capture operations, the CC are referred to as "capture channels."

Each TCC instance has up to four compare/capture channels (CCx).

The counter register (COUNT), period registers with buffer (PER and PERBUF), and compare and capture registers with buffers (CCx and CCBUFx) are 16- or 24-bit registers, depending on each TCC instance. Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached TOP or ZERO. In either case, the TCC can generate interrupt requests or generate events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

A prescaled generic clock (GCLK_TCCx) and events from the event system can be used to control the counter. The event system is also used as a source to the input capture.

The Recoverable Fault Unit enables event controlled waveforms by acting directly on the generated waveforms of the TCC compare channels output. These events can restart, halt the timer/counter period, shorten the output pulse active time, or disable waveform output as long as the fault condition is present. This can typically be used for current sensing regulation, and zero-crossing and demagnetization re-triggering.

The MCE0 and MCE1 event sources are shared with the Recoverable Fault Unit. Only asynchronous events are used internally when fault unit extension is enabled. For further details on how to configure asynchronous events routing, refer to *EVSYS – Event System*.

Recoverable fault sources can be filtered and/or windowed to avoid false triggering, for example from I/O pin glitches, by using digital filtering, input blanking, and qualification options. See also Recoverable Faults.

In addition, six optional independent and successive units primarily intended for use with different types of motor control, ballast, LED, H-bridge, power converter, and other types of power switching applications, are implemented in some of TCC instances. See also Figure 36-1.

The output matrix (OTMX) can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types. The Dead-Time Insertion (DTI) unit splits the four lower OTMX outputs into two non-overlapping signals: the non-inverted low side (LS) and inverted high side (HS) of the waveform output with optional dead-time insertion between LS and HS switching. The SWAP unit can swap the LS and HS pin outputs, and can be used for fast decay motor control.

The pattern generation unit can be used to generate synchronized waveforms with constant logic level on TCC UPDATE conditions. This is useful for easy stepper motor and full bridge control.

The non-recoverable fault module enables event controlled fault protection by acting directly on the generated waveforms of the timer/counter compare channel outputs. When a non-recoverable fault condition is detected, the output waveforms are forced to a safe and pre-configured value that is safe for the application. This is typically used for instant and predictable shut down and disabling high current or voltage drives.

The count event sources (TCE0 and TCE1) are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous action on waveform output, even for cases where the clock is not present. For further details on how to configure asynchronous events routing, refer to section *EVSYS – Event System*.

Related Links

EVSYS – Event System

36.6.2 Basic Operation

36.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled(CTRLA.ENABLE=0):

- Control A (CTRLA) register, except Run Standby (RUNSTDBY), Enable (ENABLE) and Software Reset (SWRST) bits
- Recoverable Fault n Control registers (FCTRLA and FCTRLB)
- Waveform Extension Control register (WEXCTRL)
- Drive Control register (DRVCTRL)
- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the “Enable-Protected” property in the register description.

Before the TCC is enabled, it must be configured as outlined by the following steps:

1. Enable the TCC bus clock (CLK_TCCx_APB).
2. If Capture mode is required, enable the channel in capture mode by writing a '1' to the Capture Enable bit in the Control A register (CTRLA.CPTEN).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESCSYNC).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR) to '1'.
4. Select the Waveform Generation operation in the WAVE register (WAVE.WAVEGEN).
5. Select the Waveform Output Polarity in the WAVE register (WAVE.POL).
6. The waveform output can be inverted for the individual channels using the Waveform Output Invert Enable bit group in the Driver register (DRVCTRL.INVEN).

36.6.2.2 Enabling, Disabling, and Resetting

The TCC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TCC is disabled by writing a zero to CTRLA.ENABLE.

The TCC is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TCC, except DBGCTRL, will be reset to their initial state, and the TCC will be disabled. Refer to Control A (CTRLA) register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

36.6.2.3 Prescaler Selection

The GCLK_TCCx clock is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

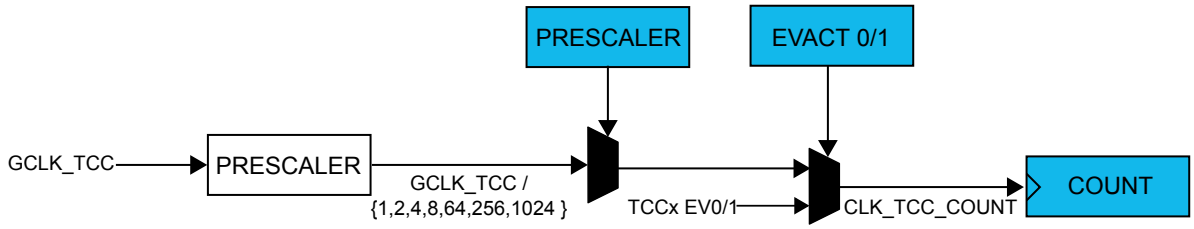
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCC clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TCC_COUNT.

Figure 36-2. Prescaler



36.6.2.4 Counter Operation

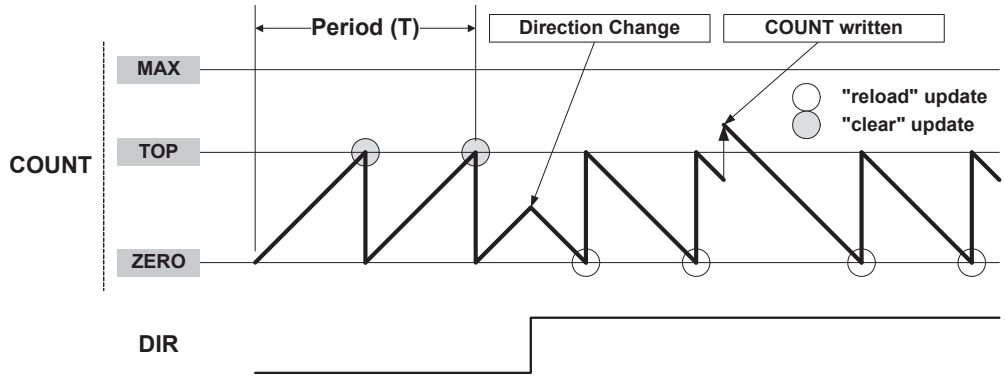
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCC_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If the bit is zero, it's counting up and one if counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it's counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When down-counting, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, or an event. An overflow/underflow occurrence (i.e. a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

Figure 36-3. Counter Operation



It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. The COUNT value will always be ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR, when starting the TCC, unless a different value has been written to it, or the TCC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See also Figure 36-3.

Stop Command

A stop command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x2, STOP).

When a stop is detected while the counter is running, the counter will maintain its current value. If the waveform generation (WG) is used, all waveforms are set to a state defined in Non-Recoverable State x

Output Enable bit and Non- Recoverable State x Output Value bit in the Driver Control register (DRVCTRL.NREx and DRVCTRL.NRVx), and the Stop bit in the Status register is set (STATUS.STOP).

Pause Event Action

A pause command can be issued when the stop event action is configured in the Input Event Action 1 bits in Event Control register (EVCTRL.EVACT1=0x3, STOP).

When a pause is detected, the counter will maintain its current value and all waveforms keep their current state, as long as a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0=0x3, START).

Re-Trigger Command and Event Action

A re-trigger command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x1, RETRIGGER), or from event when the re-trigger event action is configured in the Input Event 0/1 Action bits in Event Control register (EVCTRL.EVACTn=0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). The Re-Trigger bit in the Interrupt Flag Status and Clear register will be set (INTFLAG.TRG). It is also possible to generate an event by writing a '1' to the Re-Trigger Event Output Enable bit in the Event Control register (EVCTRL.TRGEO). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in COUNT.

Note:
When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACTn=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

Start Event Action

The start action can be selected in the Event Control register (EVCTRL.EVACT0=0x3, START) and can start the counting operation when previously stopped. The event has no effect if the counter is already counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

Note:
When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0=0x3, START), enabling the counter will not start the counter. The counter will start on the next incoming event, but it will not restart on subsequent events.

Count Event Action

The TCC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0=0x5, COUNT).

Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1=0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation, depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR or CTRLBCLR.DIR) and the direction bit value is updated accordingly.

Increment Event Action

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0=0x4, INC) and can change the counter state when an event is received. When the TCE0 event (TCCx_EV0) is received, the counter increments, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1=0x4, DEC) and can change the counter state when an event is received. When the TCE1 (TCCx_EV1) event is received, the counter decrements, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Non-recoverable Fault Event Action

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn=0x7, FAULT). When received, the counter will be stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREx and DRVCTRL.NRVx). TCE0 and TCE1 must be configured as asynchronous events.

Event Action Off

If the event action is disabled (EVCTRL.EVACTn=0x0, OFF), enabling the counter will also start the counter.

36.6.2.5 Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare/Capture Buffer Value (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD=0x3, UPDATE). For further details, refer to Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Waveform Output Generation Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Waveform Output x Inversion bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TCC_COUNT (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCx and/or EVCTRL.MCEOx is '1'. Both interrupt and event can be generated simultaneously.

There are seven waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal Frequency (NFRQ)
- Match Frequency (MFRQ)

- Normal Pulse-Width Modulation (NPWM)
- Dual-slope, interrupt/event at TOP (DSTOP)
- Dual-slope, interrupt/event at ZERO (DSBOTTOM)
- Dual-slope, interrupt/event at Top and ZERO (DSBOTH)
- Dual-slope, critical interrupt/event at ZERO (DSCRITICAL)

When using MFRQ configuration, the TOP value is defined by the CC0 register value. For the other waveform operations, the TOP value is defined by the Period (PER) register value.

For dual-slope waveform operations, the update time occurs when the counter reaches ZERO. For the other waveforms generation modes, the update time occurs on counter wraparound, on overflow, underflow, or re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 36-2. Counter Update and Overflow Event/interrupt Conditions

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTTOM	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTH	Dual-slope PWM	PER	TOP(1) & ZERO			TOP	ZERO
DSTOP	Dual-slope PWM	PER	ZERO			TOP	–

1. The UPDATE condition on TOP only will occur when circular buffer is enabled for the channel.

Related Links

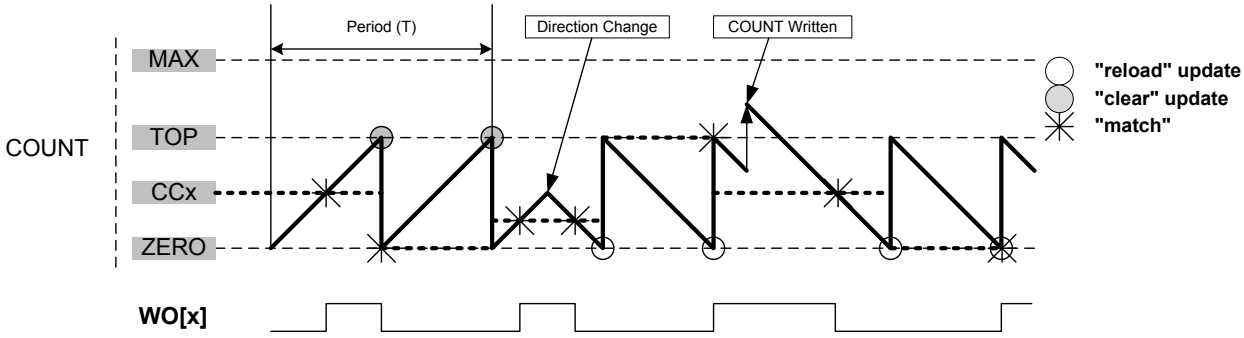
Circular Buffer

PORT: IO Pin Controller

Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (EVCTRL.MCEOx) will be set.

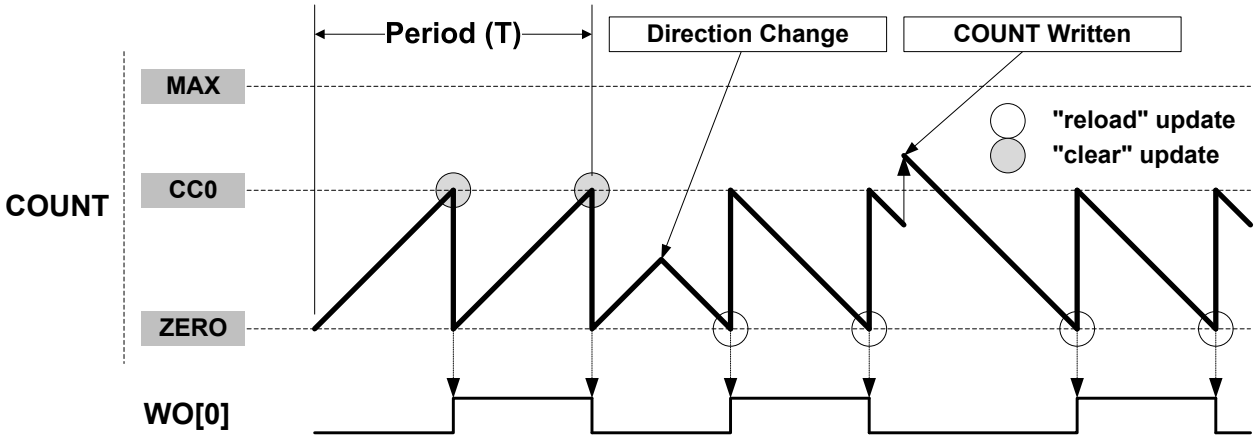
Figure 36-4. Normal Frequency Operation



Match Frequency (MFRQ)

For Match Frequency generation, the period time (T) is controlled by CC0 register instead of PER. WO[0] toggles on each update condition.

Figure 36-5. Match Frequency Operation



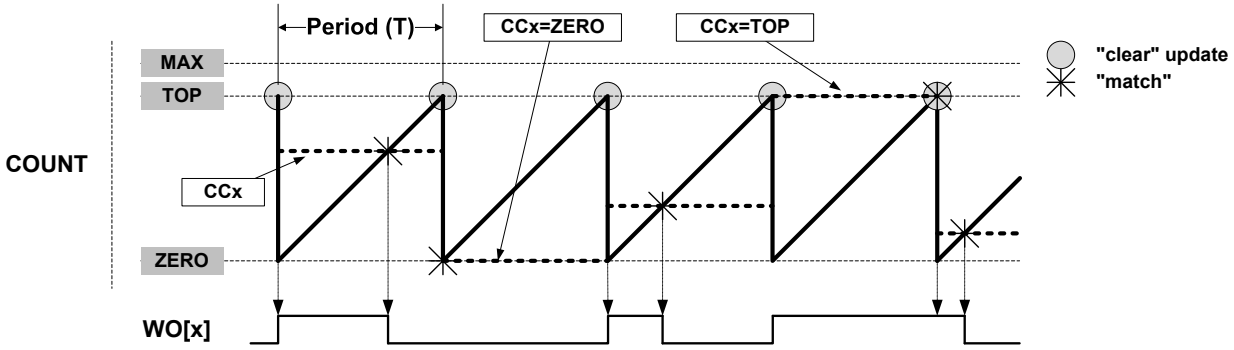
Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

Single-Slope PWM Operation

For single-slope PWM generation, the period time (T) is controlled by Top value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

Figure 36-6. Single-Slope PWM Operation



The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

$$f_{PWM_SS} = \frac{f_{GCLK_TCC}}{N(TOP+1)}$$

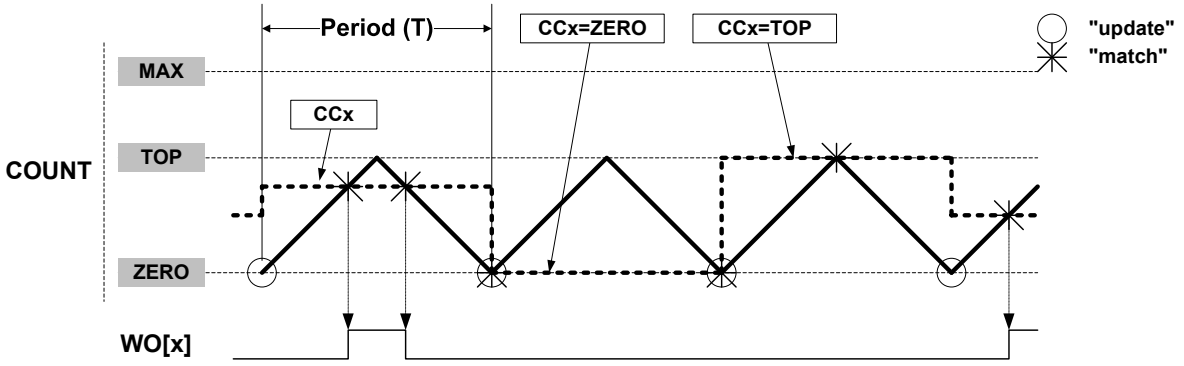
Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Dual-Slope PWM Generation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCx control the duty cycle of the generated waveform output. The figure below shows how the counter repeatedly counts from ZERO to PER and then from PER to ZERO. The waveform generator output is set on compare match when up-counting, and cleared on compare match when down-counting. An interrupt/event is generated on TOP and/or ZERO, depend of Dual slope.

In DSBOTH operation, a second update time occurs on TOP when circular buffer is enabled.

Figure 36-7. Dual-Slope Pulse Width Modulation



Using dual-slope PWM results in a lower maximum operation frequency compared to single-slope PWM generation. The period (TOP) defines the PWM resolution. The minimum resolution is 1 bit (TOP=0x00000001).

The following equation calculates the exact resolution for dual-slope PWM (R_{PWM_DS}):

$$R_{PWM_DS} = \frac{\log(PER+1)}{\log(2)}.$$

The PWM frequency f_{PWM_DS} depends on the period setting (TOP) and the peripheral clock frequency f_{GCLK_TCC}, and can be calculated by the following equation:

$$f_{PWM_DS} = \frac{f_{GCLK_TCC}}{2N \cdot PER}$$

N represents the prescaler divider used. The waveform generated will have a maximum frequency of half of the TCC clock frequency (f_{GCLK_TCC}) when TOP is set to 0x00000001 and no prescaling is used.

The pulse width (P_{PWM_DS}) depends on the compare channel (CCx) register value and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

$$P_{PWM_DS} = \frac{2N \cdot (TOP - CCx)}{f_{GCLK_TCC}}$$

N represents the prescaler divider used.

Note: In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the CCx MSB bit defines the ramp on which the CCx Match interrupt or event is generated. (Rising if CCx[MSB]=0, falling if CCx[MSB]=1.)

Related Links

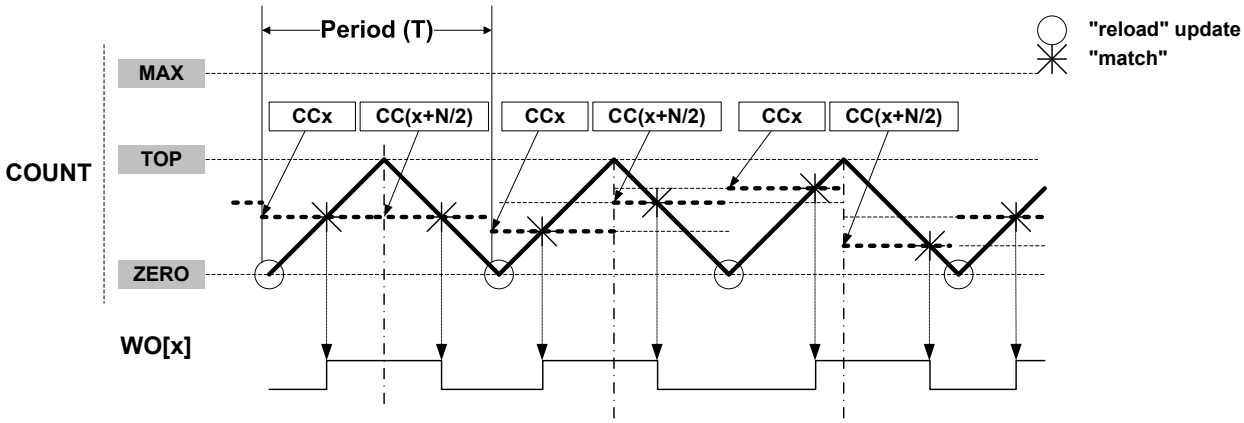
Circular Buffer

Dual-Slope Critical PWM Generation

Dual-Slope Critical PWM Generation

Critical mode generation allows generation of non-aligned centered pulses. In this mode, the period time is controlled by PER while CCx control the generated waveform output edge during up-counting and CC(x+CC_NUM/2) control the generated waveform output edge during down-counting.

Figure 36-8. Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)



Output Polarity

The polarity (WAVE.POLx) is available in all waveform output generation. In single-slope and dual-slope PWM operation, it is possible to invert the pulse edge alignment individually on start or end of a PWM cycle for each compare channels. The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.

Table 36-3. Waveform Generation Set/Clear Conditions

Waveform Generation operation	DIR	POLx	Waveform Generation Output Update	
			Set	Clear
Single-Slope PWM	0	0	Timer/counter matches TOP	Timer/counter matches CCx
		1	Timer/counter matches CC	Timer/counter matches TOP
	1	0	Timer/counter matches CC	Timer/counter matches ZERO
		1	Timer/counter matches ZERO	Timer/counter matches CC
Dual-Slope PWM	x	0	Timer/counter matches CC when counting up	Timer/counter matches CC when counting down
		1	Timer/counter matches CC when counting down	Timer/counter matches CC when counting up

In Normal and Match Frequency, the WAVE.POLx value represents the initial state of the waveform output.

36.6.2.6 Double Buffering

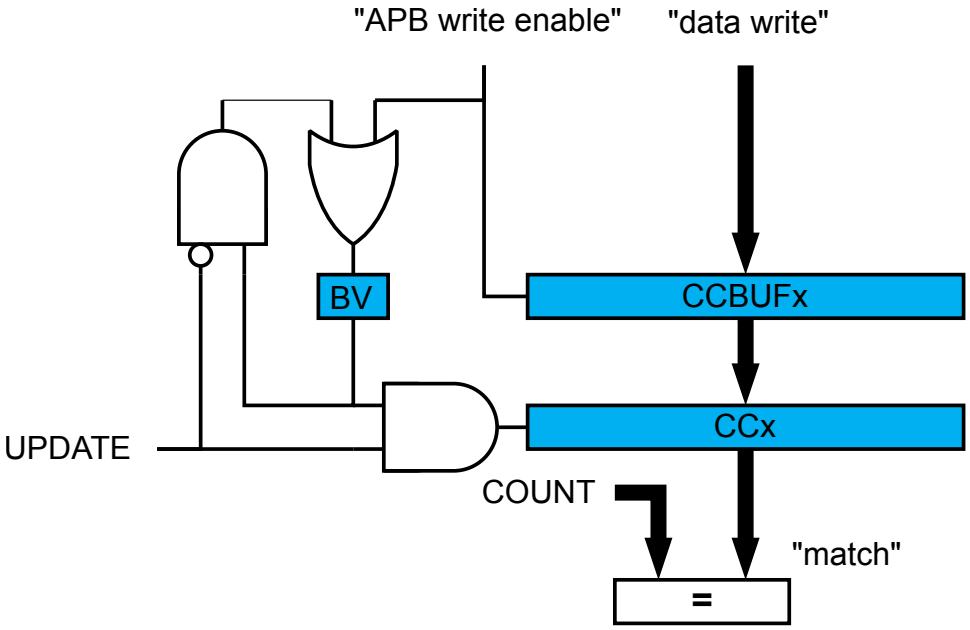
The Pattern (PATT), Period (PER) and Compare Channels (CCx) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV, PERBUFV or CCBUFVx) bit in the STATUS register, which indicates that the buffer register contains a valid value that can be copied into the corresponding register. .

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the buffer valid flags bit in the STATUS register are automatically cleared by hardware.

Note: Software update command (CTRLBSET.CMD=0x3) act independently of LUPD value.

A compare register is double buffered as in the following figure.

Figure 36-9. Compare Channel Double Buffering



Both the registers (PATT/PER/CCx) and corresponding buffer registers (PATTBUFPERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLSET.LUPD.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continuously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new Top value to the Period register (PER or CC0, depending on the waveform generation mode), any period update on registers (PER or CCx) is effective after the synchronization delay, whatever double buffering enabling is.

Figure 36-10. Unbuffered Single-Slope Up-Counting Operation

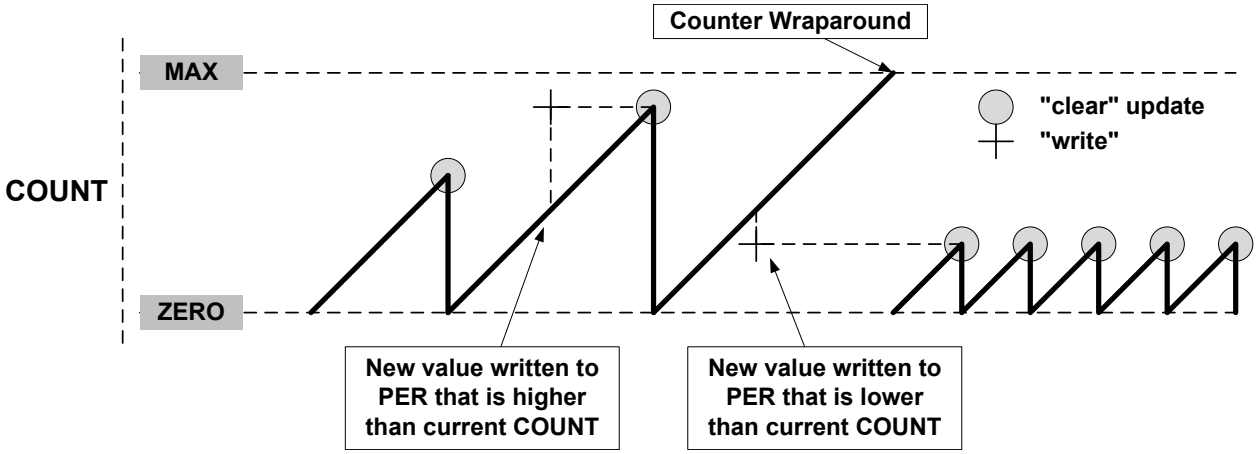
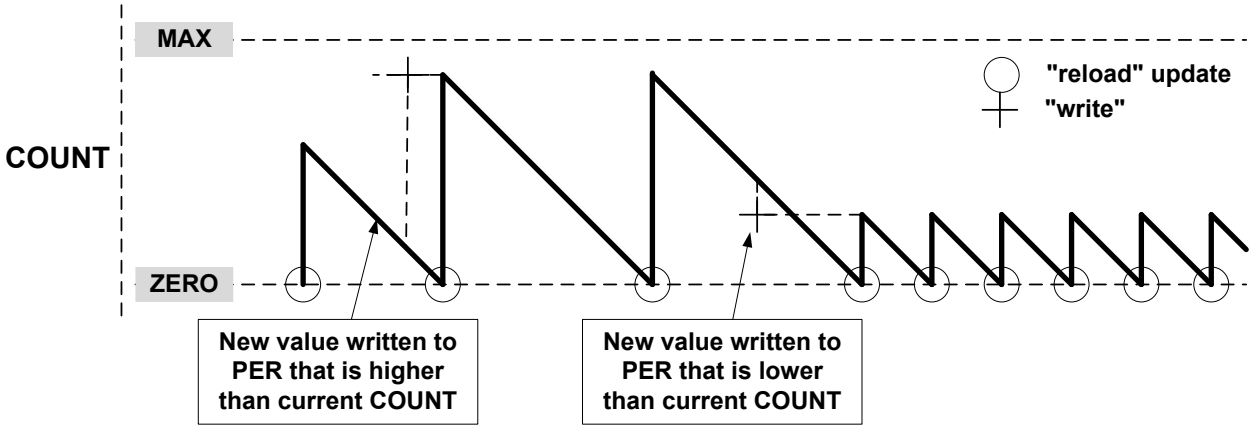
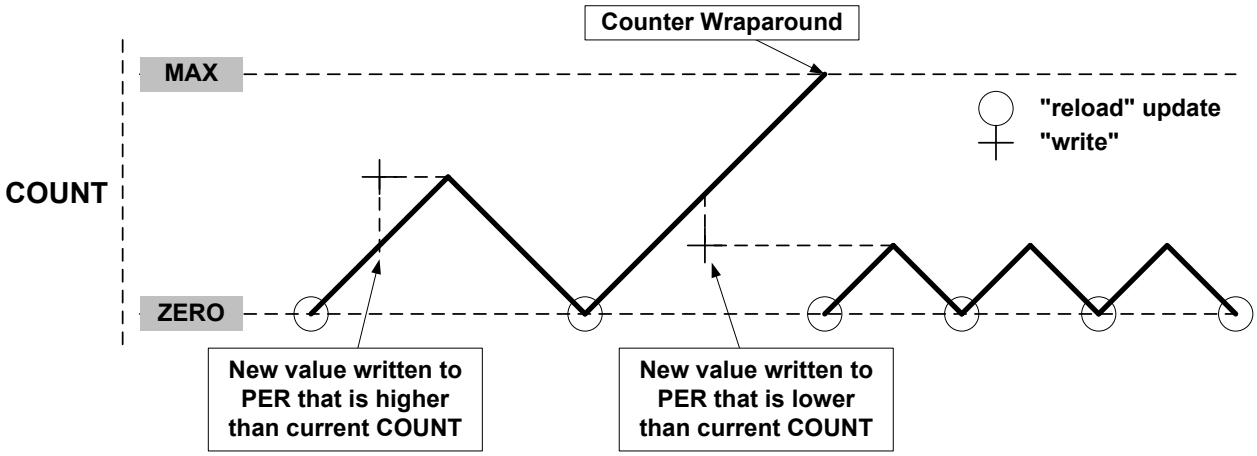


Figure 36-11. Unbuffered Single-Slope Down-Counting Operation



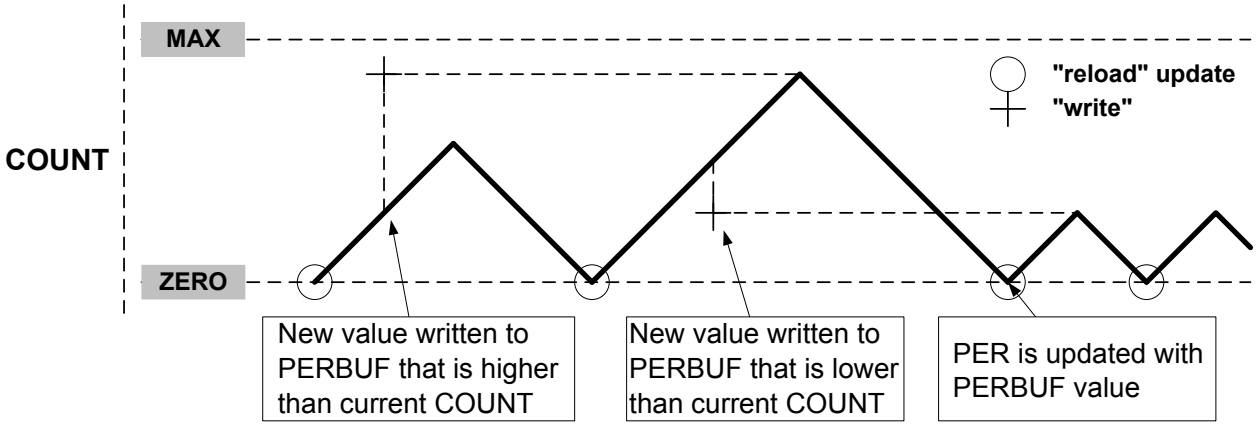
A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 36-10. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 36-12. Unbuffered Dual-Slope Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 36-13. This prevents wraparound and the generation of odd waveforms.

Figure 36-13. Changing the Period Using Buffering



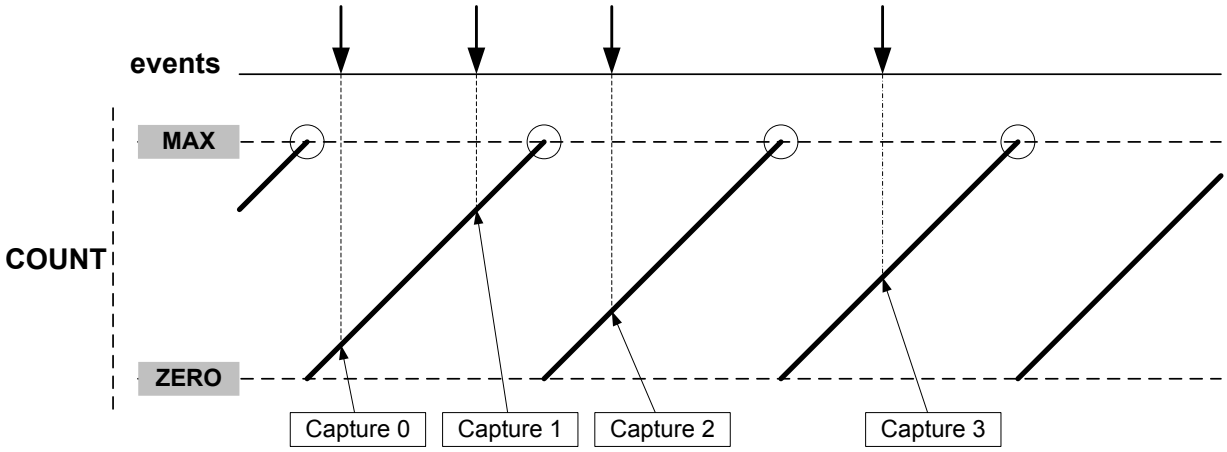
36.6.2.7 Capture Operations

To enable and use capture operations, the Match or Capture Channel x Event Input Enable bit in the Event Control register (EVCTRL.MCEIx) must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel x Enable bit in the Control A register (CTRLA.CPTENx) before capturing can be performed.

Event Capture Action

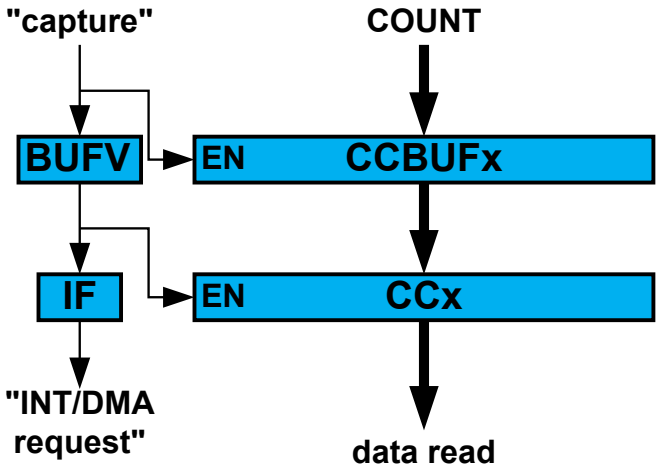
The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 36-14. Input Capture Timing



For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. CCBUFx register value can't be read, all captured data must be read from CCx register.

Figure 36-15. Capture Double Buffering



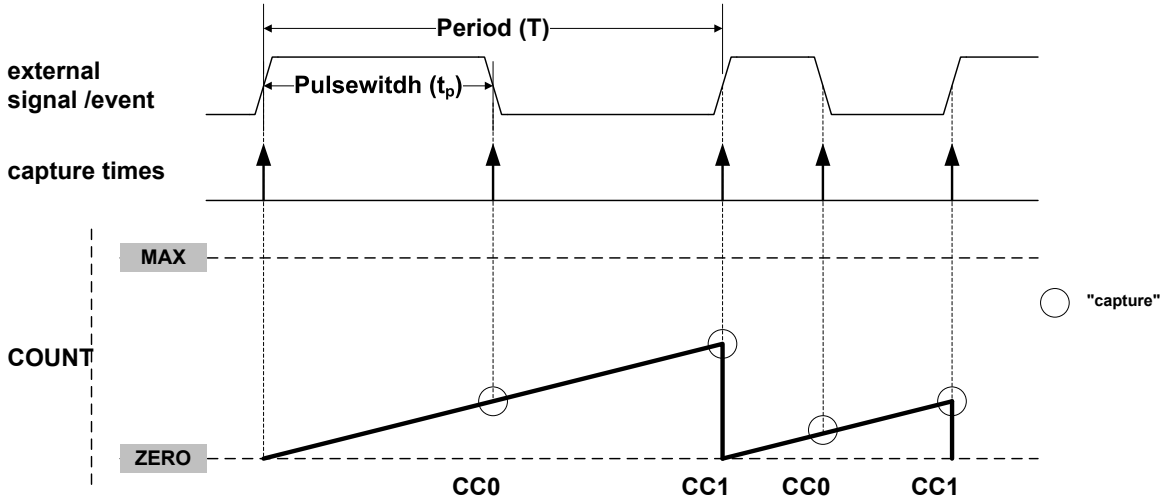
The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Buffer Valid flag (STATUS.CCBUFV) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Period and Pulse-Width (PPW) Capture Action

The TCC can perform two input captures and restart the counter on one of the edges. This enables the TCC to measure the pulse-width and period and to characterize the frequency *f* and *dutyCycle* of an input signal:

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$

Figure 36-16. PWP Capture



Selecting PWP or PPW in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. When using PPW (period and pulse-width) event action, period *T* will be captured into CC0 and the pulse-width t_p into CC1. The PWP (Pulse-width and Period) event action offers the same functionality, but *T* will be captured into CC1 and t_p into CC0.

The Timer/Counter Event x Invert Enable bit in Event Control register (EVCTRL.TCEINVx) is used for event source x to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCEINVx=1, the wraparound will happen on the falling edge.

The corresponding capture is done only if the channel is enabled in capture mode (CTRLA.CPTENx=1). If not, the capture action will be ignored and the channel will be enabled in compare mode of operation. When only one of these channel is required, the other channel can be used for other purposes.

The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the INTFLAG.MCx is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: When up-counting (CTRLBSET.DIR=0), counter values lower than 1 cannot be captured in Capture Minimum mode (FCTRLn.CAPTURE=CAPTMIN). To capture the full range including value 0, the TCC must be configured in down-counting mode (CTRLBSET.DIR=0).

Note: In dual-slope PWM operation, and when TOP is lower than MAX/2, the CCx MSB captures the CTRLB.DIR state to identify the ramp on which the capture has been done. For rising ramps CCx[MSB] is zero, for falling ramps CCx[MSB]=1.

36.6.3 Additional Features

36.6.3.1 One-Shot Operation

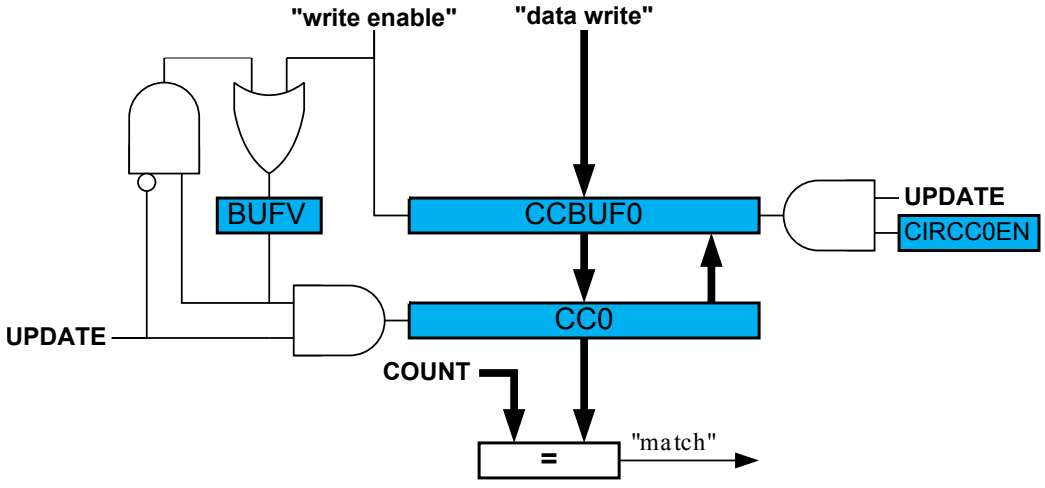
When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is set and the waveform outputs are set to the value defined by DRVCTRL.NREx and DRVCTRL.NRVx.

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

36.6.3.2 Circular Buffer

The Period register (PER) and the compare channels register (CC0 to CC3) support circular buffer operation. When circular buffer operation is enabled, the PER or CCx values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

Figure 36-17. Circular Buffer on Channel 0



36.6.3.3 Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 PWM cycles frame.

Dithering consists in adding some extra clocks cycles in a frame of several PWM cycles, and can improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Enhanced Resolution bits in CTRLA register (CTRLA.RESOLUTION):

- DITH4 enable dithering every 16 PWM frames
- DITH5 enable dithering every 32 PWM frames
- DITH6 enable dithering every 64 PWM frames

The DITHERCY bits of COUNT, PER and CCx define the number of extra cycles to add into the frame (DITHERCY bits from the respective COUNT, PER or CCx registers). The remaining bits of COUNT, PER, CCx define the compare value itself.

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
  int MASK;
  int value
  switch (resolution){
    DITH4: MASK = 0x0f;
    DITH5: MASK = 0x1f;
    DITH6: MASK = 0x3f;
  }
  value = cycle * dithercy;
  if (((MASK & value) + dithercy) > MASK)
    return 1;
 return 0;
}
```

Dithering on Period

Writing DITHERCY in PER will lead to an average PWM period configured by the following formulas.

DITH4 mode:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK_TCC}}}\right)$$

Note: If DITH4 mode is enabled, the last 4 significant bits from PER/CCx or COUNT register correspond to the DITHERCY value, rest of the bits corresponds to PER/CCx or COUNT value.

DITH5 mode:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{32} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK_TCC}}}\right)$$

DITH6 mode:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{64} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK_TCC}}}\right)$$

Dithering on Pulse Width

Writing DITHERCY in CCx will lead to an average PWM pulse width configured by the following formula.

DITH4 mode:

$$PwmPulseWidth = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{CCx}\right)\left(\frac{1}{f_{\mathrm{GCLK_TCC}}}\right)$$

DITH5 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{32} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK_TCC}}}\right)$$

DITH6 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{64} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK_TCC}}}\right)$$

Note: The PWM period will remain static in this case.

36.6.3.4 Ramp Operations

Three ramp operation modes are supported. All of them require the timer/counter running in single-slope PWM generation. The ramp mode is selected by writing to the Ramp Mode bits in the Waveform Control register (WAVE.RAMP).

RAMP1 Operation

This is the default PWM operation, described in Single-Slope PWM Generation.

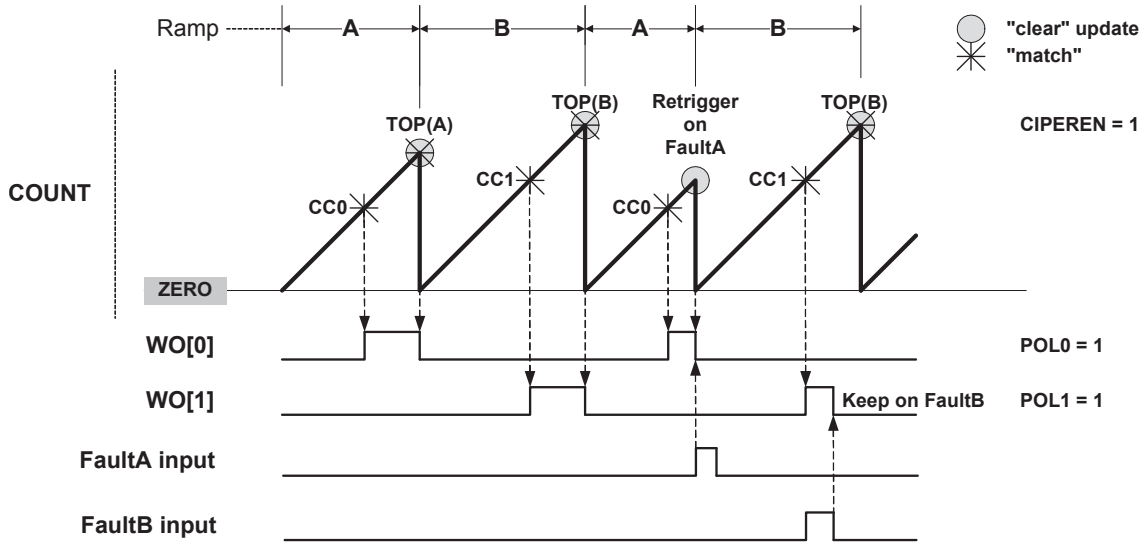
RAMP2 Operation

These operation modes are dedicated for power factor correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved, see Figure 36-18. In cycle A, odd channel output is disabled, and in cycle B, even channel output is disabled. The ramp index changes after each update, but can be software modified using the Ramp index command bits in Control B Set register (CTRLBSET.IDXCMD).

Standard RAMP2 (RAMP2) Operation

Ramp A and B periods are controlled by the PER register value. The PER value can be different on each ramp by the Circular Period buffer option in the Wave register (WAVE.CIPEREN=1). This mode uses a two-channel TCC to generate two output signals, or one output signal with another CC channel enabled in capture mode.

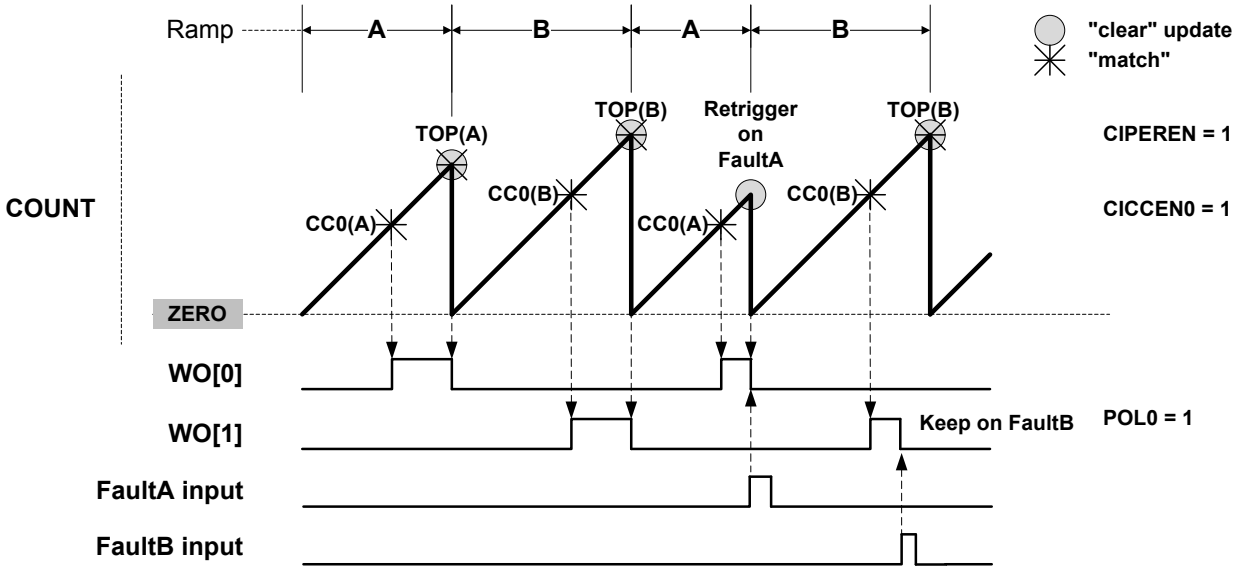
Figure 36-18. RAMP2 Standard Operation



Alternate RAMP2 (RAMP2A) Operation

Alternate RAMP2 operation is similar to RAMP2, but CC0 controls both WO[0] and WO[1] waveforms when the corresponding circular buffer option is enabled (CIPEREN=1). The waveform polarity is the same on both outputs. Channel 1 can be used in capture mode.

Figure 36-19. RAMP2 Alternate Operation



Critical RAMP2 (RAMP2C) Operation

Critical RAMP2 operation provides a way to cover RAMP2 operation requirements without the update constraint associated to the use of circular buffers. In this mode, CC0 is controlling the period of ramp A and PER is controlling the period of ramp B. When using more than two channels, WO[0] output is controlled by CC2 (HIGH) and CC0 (LOW). On TCC with 2 channels, a pulse on WO[0] will last the entire period of ramp A, if WAVE.POL0=0.

Figure 36-20. RAMP2 Critical Operation With More Than 2 Channels

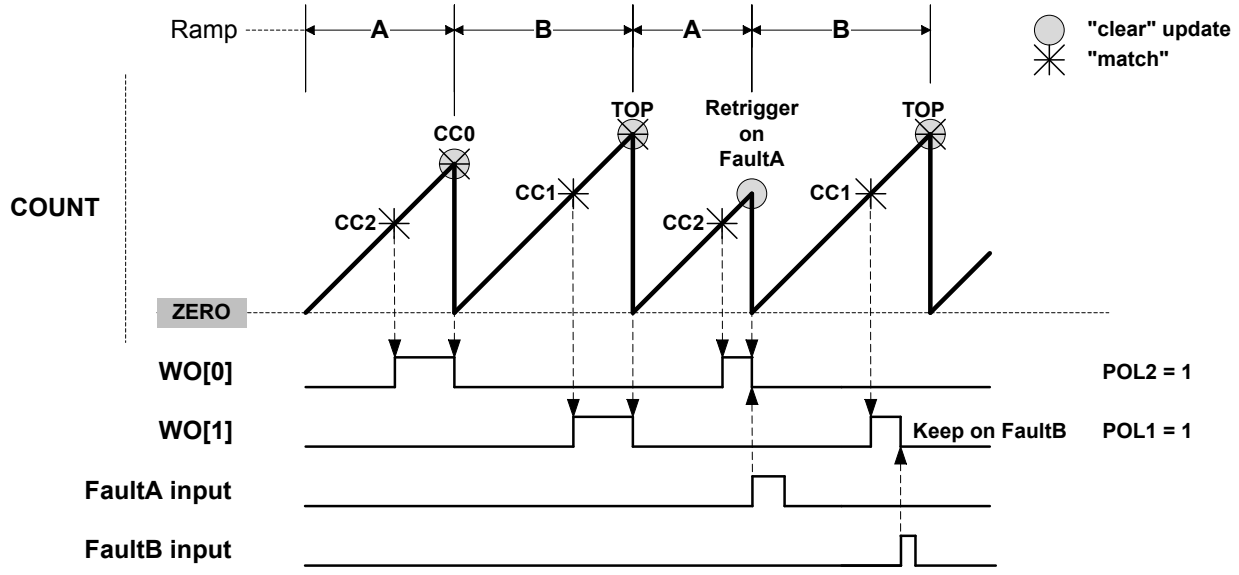
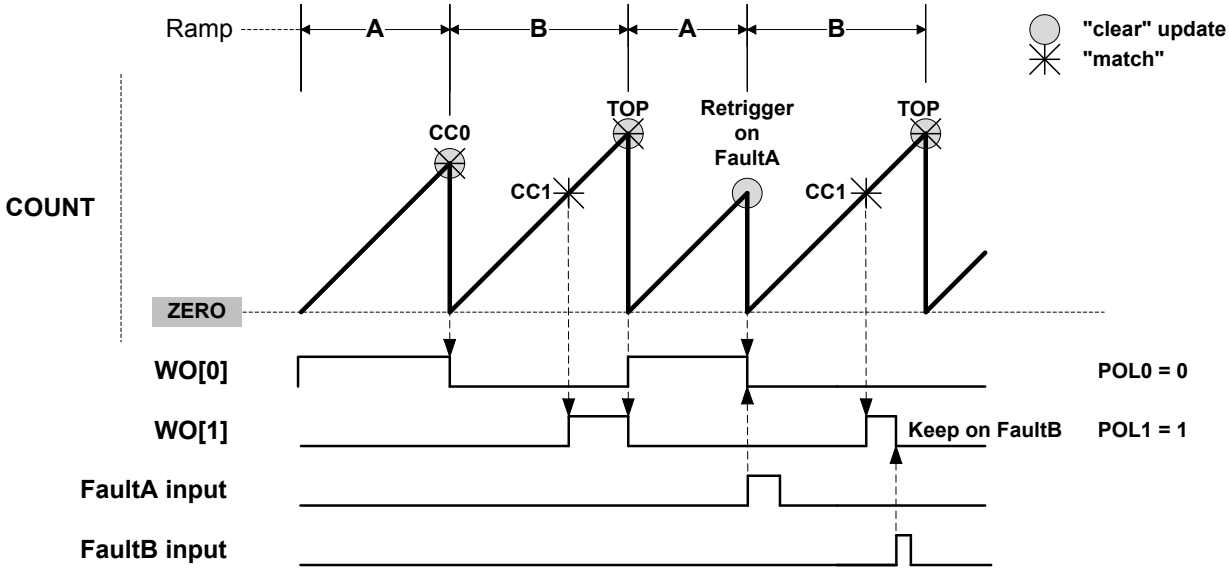


Figure 36-21. RAMP2 Critical Operation With 2 Channels



36.6.3.5 Recoverable Faults

Recoverable faults can restart or halt the timer/counter. Two faults, called Fault A and Fault B, can trigger recoverable fault actions on the compare channels CC0 and CC1 of the TCC. The compare channels' outputs can be clamped to inactive state either as long as the fault condition is present, or from the first valid fault condition detection on until the end of the timer/counter cycle.

Fault Inputs

The first two channel input events (TCCxMC0 and TCCxMC1) can be used as Fault A and Fault B inputs, respectively. Event system channels connected to these fault inputs must be configured as asynchronous. The TCC must work in a PWM mode.

Fault Filtering

There are three filters available for each input Fault A and Fault B. They are configured by the corresponding Recoverable Fault n Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

Input Filtering

By default, the event detection is asynchronous. When the event occurs, the fault system will immediately and asynchronously perform the selected fault action on the compare channel output, also in device power modes where the clock is not available. To avoid false fault detection on external events (e.g. due to a glitch on an I/O port) a digital filter can be enabled and configured by the Fault B Filter Value bits in the Fault n Configuration registers (FCTRLn.FILTERVAL). If the event width is less than FILTERVAL (in clock cycles), the event will be discarded. A valid event will be delayed by FILTERVAL clock cycles.

Fault Blanking

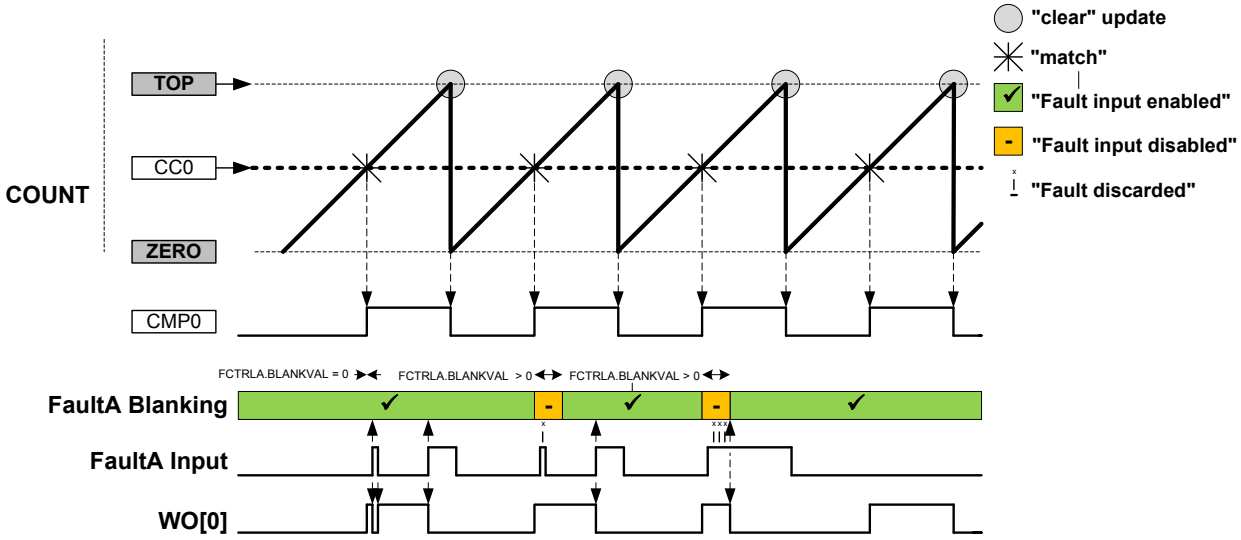
This ignores any fault input for a certain time just after a selected waveform output edge. This can be used to prevent false fault triggering due to signal bouncing, as shown in the figure below. Blanking can be enabled by writing an edge triggering configuration to the Fault n Blanking Mode bits in the Recoverable Fault n Configuration register (FCTRLn.BLANK). The desired duration of the blanking must be written to the Fault n Blanking Time bits (FCTRLn.BLANKVAL).
The blanking time $t_b$is calculated by

$$t_b = \frac{1 + \text{BLANKVAL}}{f_{\text{GCLK_TCCx_PRESC}}}$$

Here, $f_{\text{GCLK_TCCx_PRESC}}$ is the frequency of the prescaled peripheral clock frequency $f_{\text{GCLK_TCCx}}$.

The maximum blanking time (FCTRLn.BLANKVAL= 255) at $f_{\text{GCLK_TCCx}}$=96MHz is 2.67µs (no prescaler) or 170µs (prescaling). For $f_{\text{GCLK_TCCx}}$=1MHz, the maximum blanking time is either 170µs (no prescaling) or 10.9ms (prescaling enabled).

Figure 36-22. Fault Blanking in RAMP1 Operation with Inverted Polarity



Fault Qualification

This is enabled by writing a '1' to the Fault n Qualification bit in the Recoverable Fault n Configuration register (FCTRLn.QUAL). When the recoverable fault qualification is enabled (FCTRLn.QUAL=1), the fault input is disabled all the time the corresponding channel output has an inactive level, as shown in the figures below.

Figure 36-23. Fault Qualification in RAMP1 Operation

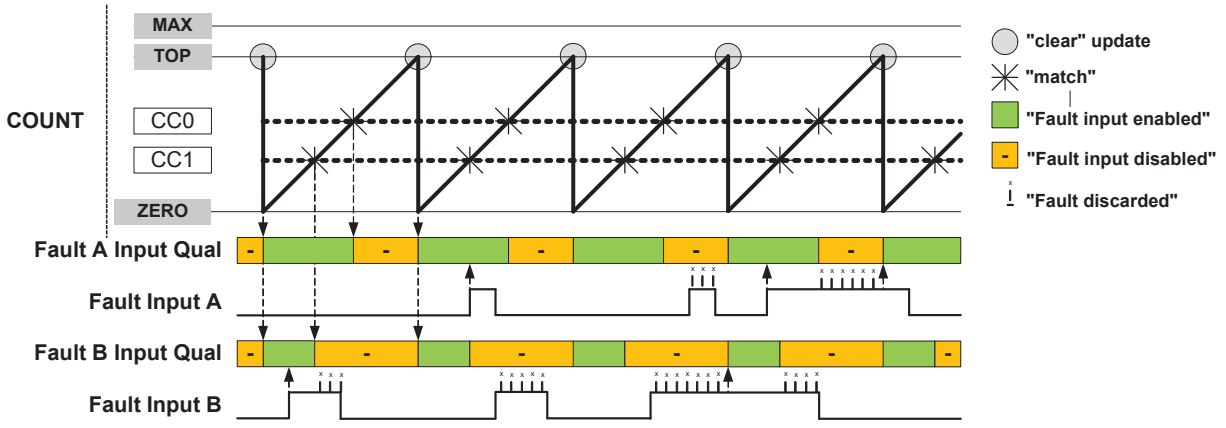
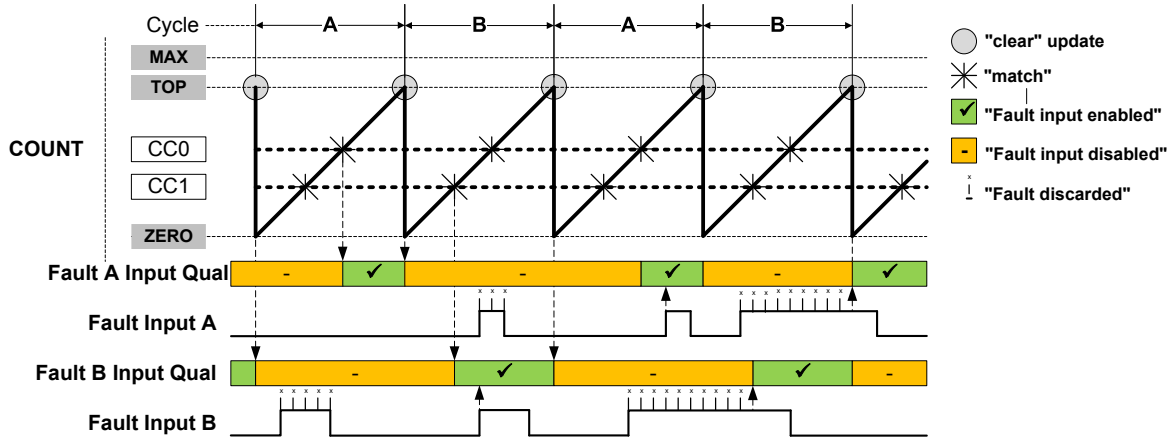


Figure 36-24. Fault Qualification in RAMP2 Operation with Inverted Polarity

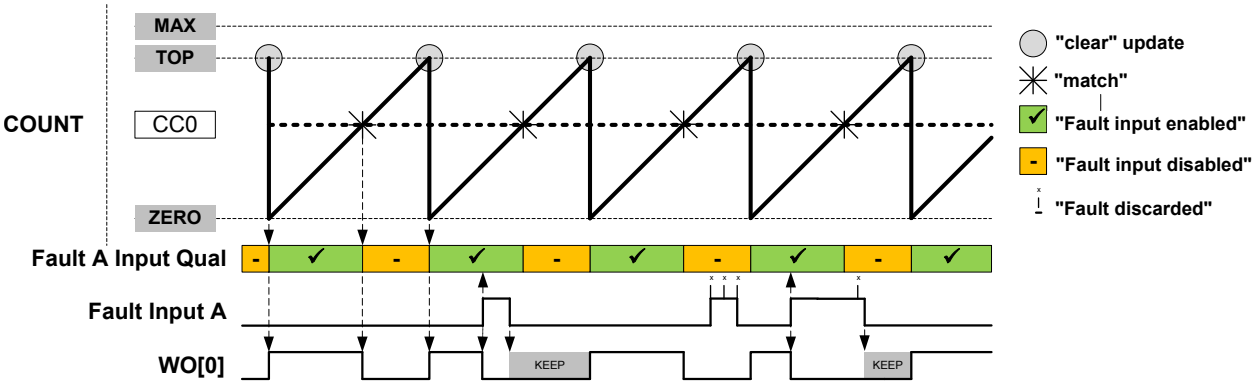


Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; hence two or more actions can be enabled at the same time to achieve a result that is a combination of fault actions.

Keep Action This is enabled by writing the Fault n Keeper bit in the Recoverable Fault n Configuration register (FCTRLn.KEEP) to '1'. When enabled, the corresponding channel output will be clamped to zero as long as the fault condition is present. The clamp will be released on the start of the first cycle after the fault condition is no longer present, see next Figure.

Figure 36-25. Waveform Generation with Fault Qualification and Keep Action



Restart Action This is enabled by writing the Fault n Restart bit in Recoverable Fault n Configuration register (FCTRLn.RESTART) to '1'. When enabled, the timer/counter will be restarted as soon as the corresponding fault condition is present. The ongoing cycle is stopped and the timer/counter starts a new cycle, see Figure 36-26. In Ramp 1 mode, when the new cycle starts, the compare outputs will be clamped to inactive level as long as the fault condition is present.

Note: For RAMP2 operation, when a new timer/counter cycle starts the cycle index will change automatically, see Figure 36-27. Fault A and Fault B are qualified only during the cycle A and cycle B respectively: Fault A is disabled during cycle B, and Fault B is disabled during cycle A.

Figure 36-26. Waveform Generation in RAMP1 mode with Restart Action

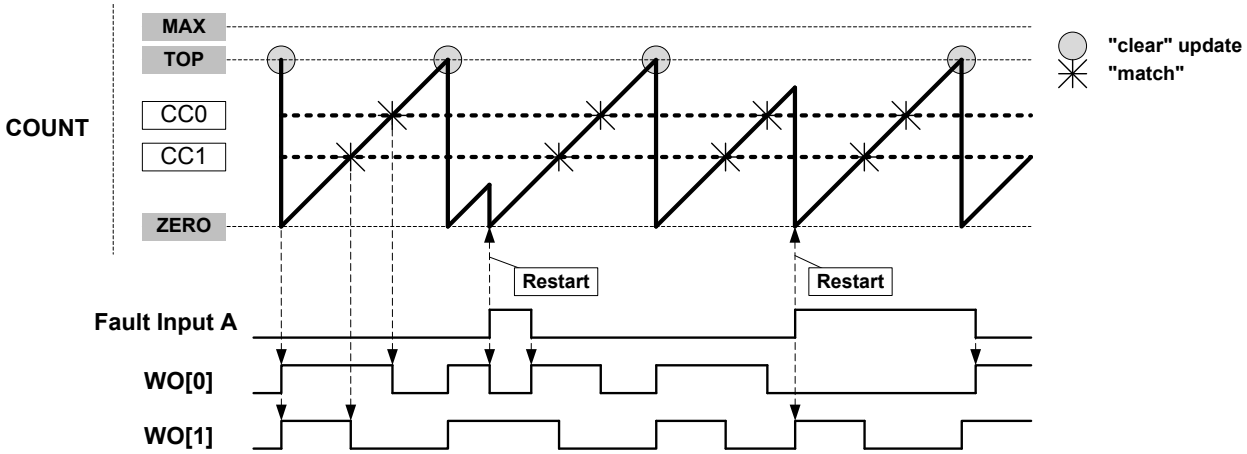
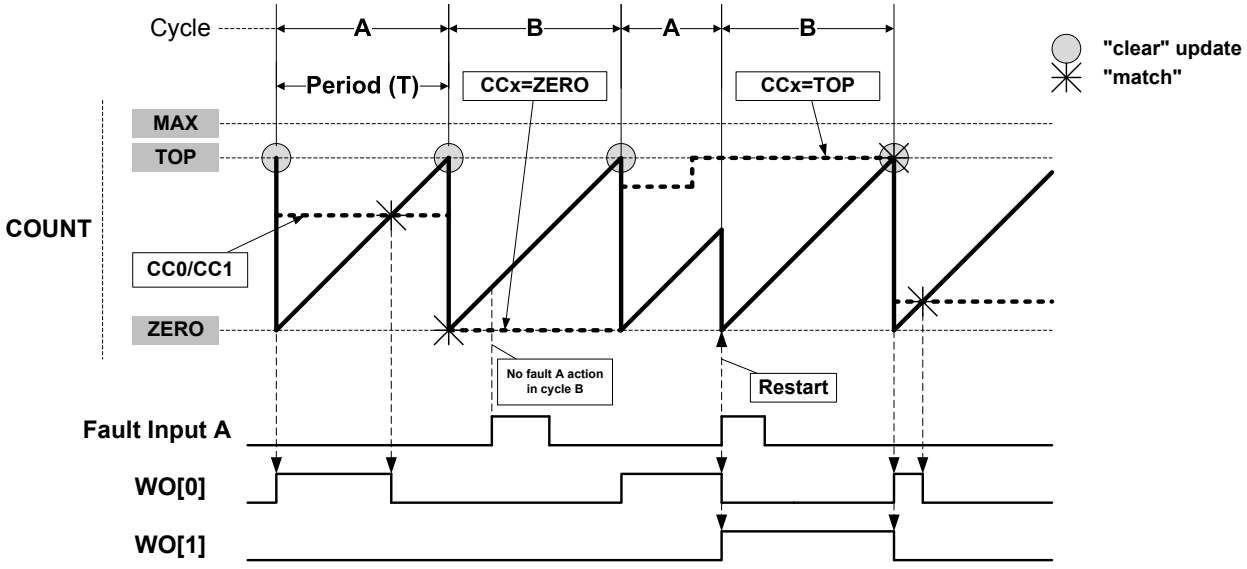


Figure 36-27. Waveform Generation in RAMP2 mode with Restart Action



Capture Action

Several capture actions can be selected by writing the Fault n Capture Action bits in the Fault n Control register (FCTRLn.CAPTURE). When one of the capture operations is selected, the counter value is captured when the fault occurs. These capture operations are available:

- CAPT - the equivalent to a standard capture operation, for further details refer to Capture Operations
- CAPTMIN - gets the minimum time stamped value: on each new local minimum captured value, an event or interrupt is issued.
- CAPTMAX - gets the maximum time stamped value: on each new local maximum captured value, an event or interrupt (IT) is issued, see Figure 36-28.
- LOCMIN - notifies by event or interrupt when a local minimum captured value is detected.
- LOCMAX - notifies by event or interrupt when a local maximum captured value is detected.
- DERIV0 - notifies by event or interrupt when a local extreme captured value is detected, see Figure 36-29.

CCx Content:

In CAPTMIN and CAPTMAX operations, CCx keeps the respective extremum captured values, see Figure 36-28. In LOCMIN, LOCMAX or DERIV0 operation, CCx follows the counter value at fault time, see Figure 36-29.

Before enabling CAPTMIN or CAPTMAX mode of capture, the user must initialize the corresponding CCx register value to a value different from zero (for CAPTMIN) top (for CAPTMAX). If the CCx register initial value is zero (for CAPTMIN) top (for CAPTMAX), no captures will be performed using the corresponding channel.

MCx Behaviour:

In LOCMIN and LOCMAX operation, capture is performed on each capture event. The MCx interrupt flag is set only when the captured value is upper or equal (for LOCMIN) or lower or equal (for LOCMAX) to the previous captured value. So interrupt flag is set when a new relative local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected. DERIV0 is equivalent to an OR function of (LOCMIN, LOCMAX).

In CAPT operation, capture is performed on each capture event. The MCx interrupt flag is set on each new capture.

In CAPTMIN and CAPTMAX operation, capture is performed only when on capture event time, the counter value is lower (for CAPTMIN) or upper (for CAPMAX) than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is upper or equal (for CAPTMIN) or lower or equal (for CAPTMAX) to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected.

Interrupt Generation

In CAPT mode, an interrupt is generated on each filtered Fault n and each dedicated CCx channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

Figure 36-28. Capture Action “CAPTMAX”

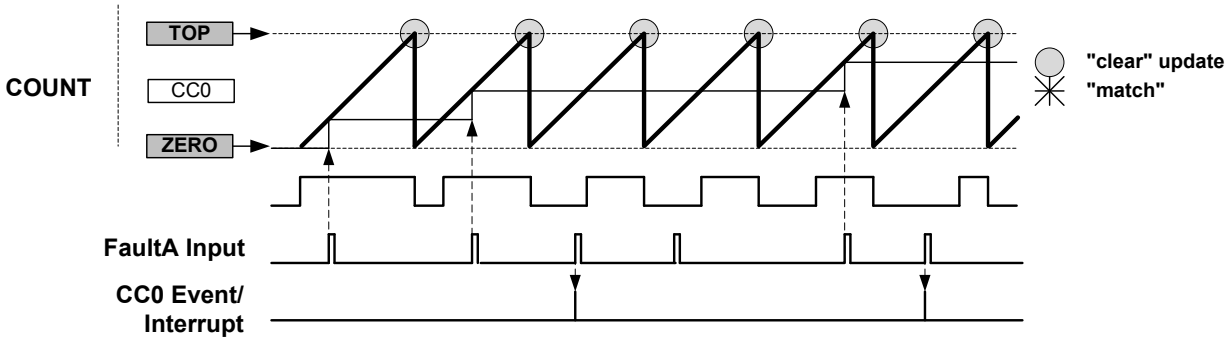
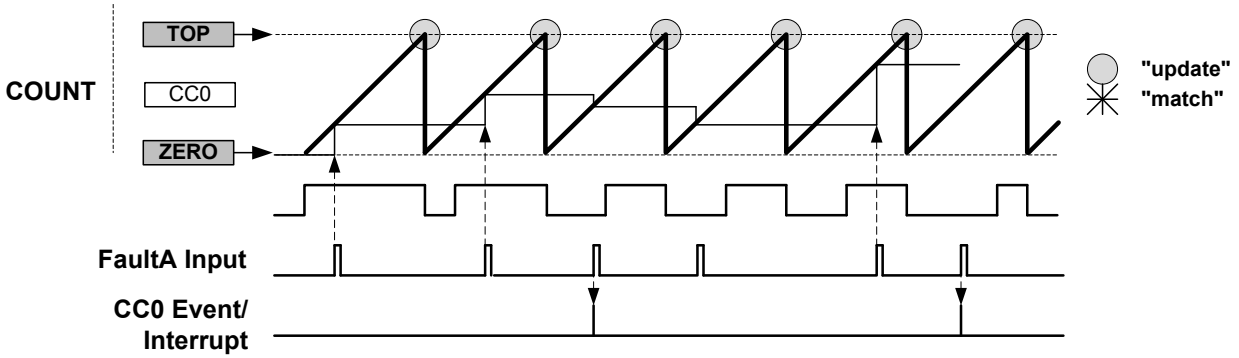


Figure 36-29. Capture Action “DERIV0”



Hardware Halt Action

This is configured by writing 0x1 to the Fault n Halt mode bits in the Recoverable Fault n Configuration register (FCTRLn.HALT). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

The next figure ('Waveform Generation with Halt and Restart Actions') shows an example where both restart action and hardware halt action are enabled for Fault A. The compare channel 0 output is clamped to inactive level as long as the timer/counter is halted. The timer/counter resumes the counting operation as soon as the fault condition is no longer present. As the restart action is enabled in this example, the timer/counter is restarted after the fault condition is no longer present.

The figure after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.

Figure 36-30. Waveform Generation with Halt and Restart Actions

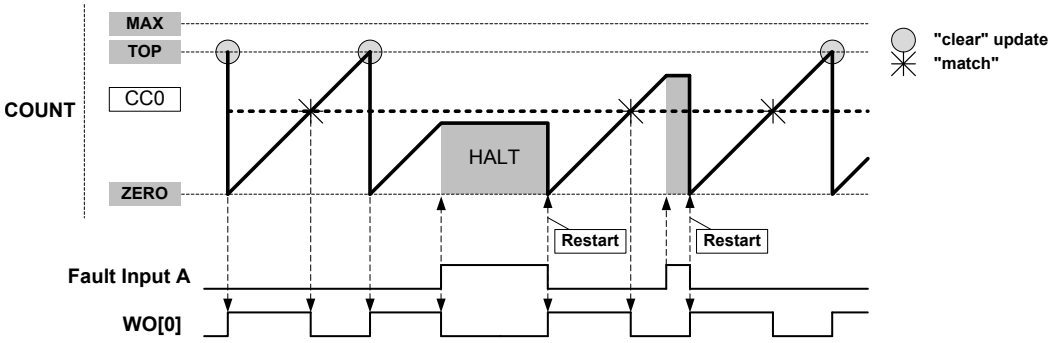
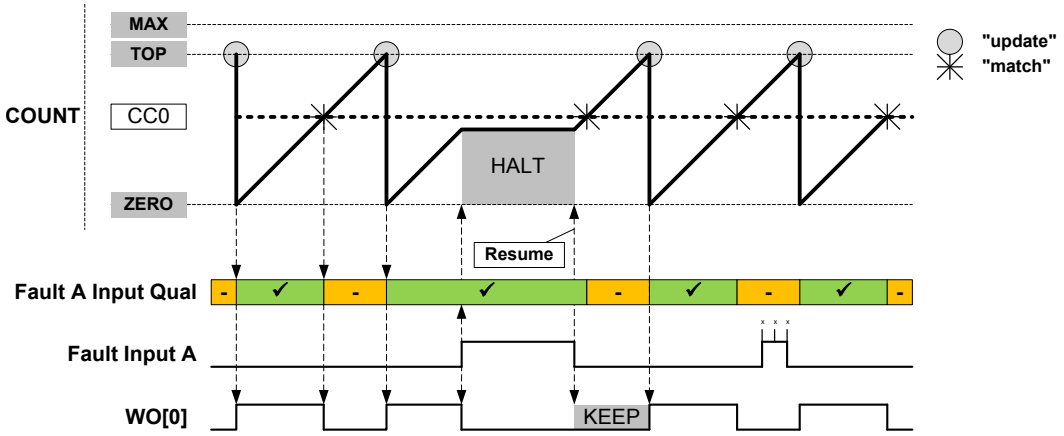


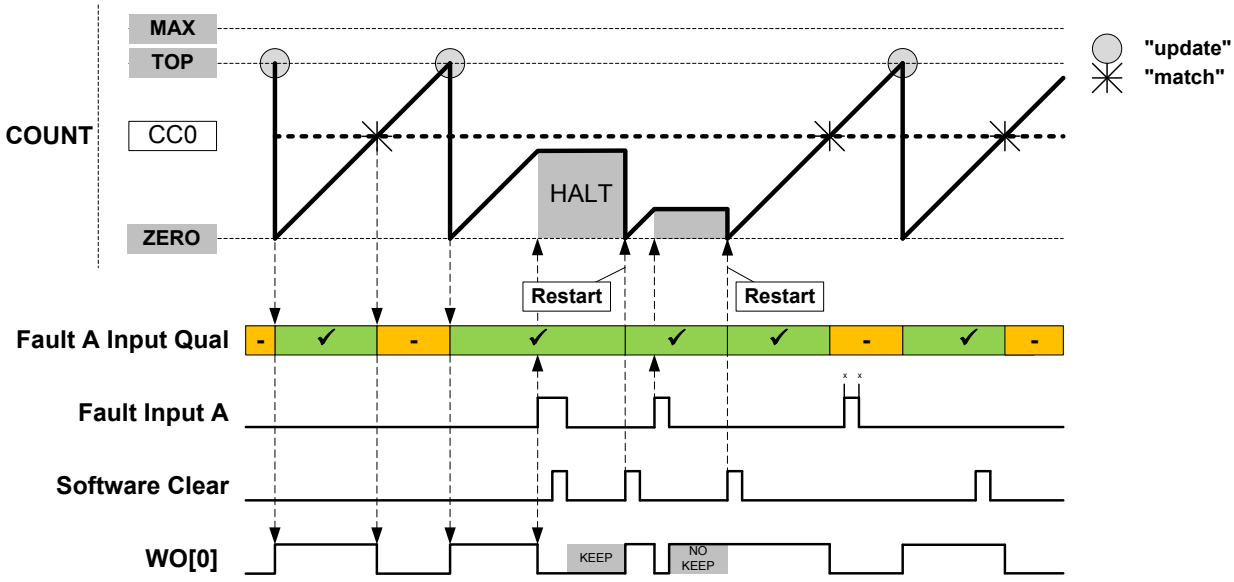
Figure 36-31. Waveform Generation with Fault Qualification, Halt, and Restart Actions



Software Halt Action

This is configured by writing 0x2 to the Fault n Halt mode bits in the Recoverable Fault n configuration register (FCTRLn.HALT). Software halt action is similar to hardware halt action, but in order to restart the timer/counter, the corresponding fault condition must not be present anymore, and the corresponding FAULT n bit in the STATUS register must be cleared by software.

Figure 36-32. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions



36.6.3.6 Non-Recoverable Faults

The non-recoverable fault action will force all the compare outputs to a pre-defined level programmed into the Driver Control register (DRVCTRL.NRE and DRVCTRL.NRV). The non-recoverable fault input (EV0 and EV1) actions are enabled in Event Control register (EVCTRL.EVACT0 and EVCTRL.EVACT1).

To avoid false fault detection on external events (e.g. a glitch on an I/O port) a digital filter can be enabled using Non-Recoverable Fault Input x Filter Value bits in the Driver Control register (DRVCTRL.FILTERVALn). Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD) is written to '1', a non-recoverable Debug Faults State and an interrupt (DFS) is generated when the system goes in debug operation.

In RAMP2, RAMP2A, or DSBOTH operation, when the Lock Update bit in the Control B register is set by writing CTRLBSET.LUPD=1 and the ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

36.6.3.7 Time-Stamp Capture

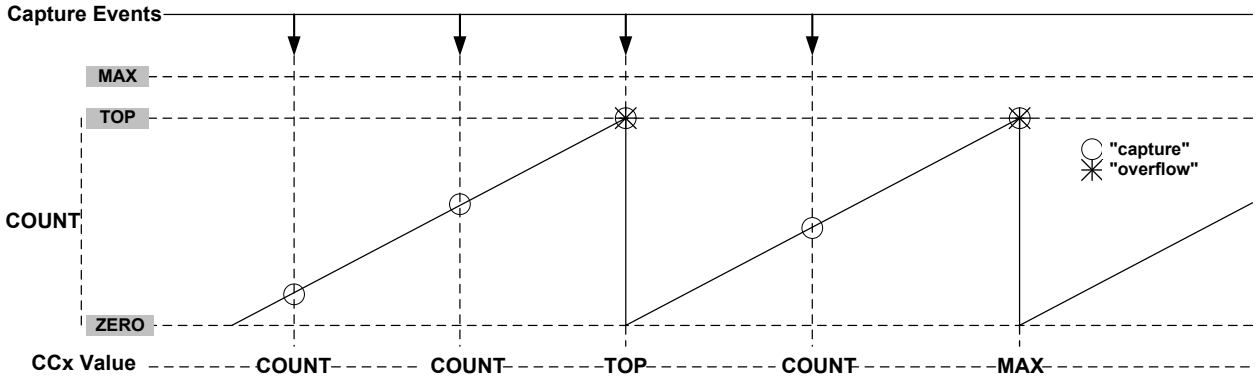
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 36-33. Time-Stamp



36.6.3.8 Waveform Extension

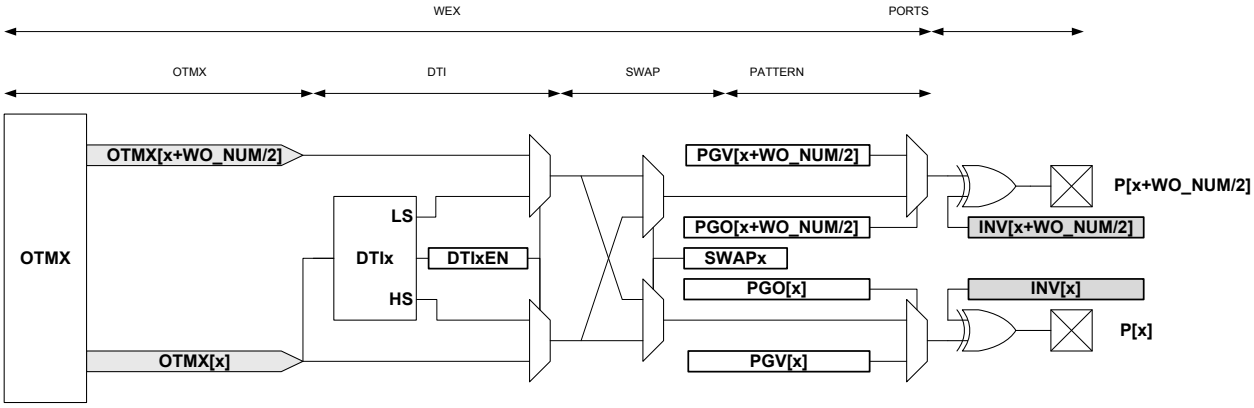
Figure 36-34 shows a schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as a four port pair slices:

- Slice 0 DTI0 / SWAP0 acting on port pins (WO[0], WO[WO_NUM/2 +0])
- Slice 1 DTI1 / SWAP1 acting on port pins (WO[1], WO[WO_NUM/2 +1])

And more generally:

- Slice n DTIx / SWAPx acting on port pins (WO[x], WO[WO_NUM/2 +x])

Figure 36-34. Waveform Extension Stage Details



The output matrix (OTMX) unit distributes compare channels, according to the selectable configurations in Table 36-4.

Table 36-4. Output Matrix Channel Pin Routing Configuration

Value	OTMX[x]							
0x0	CC3	CC2	CC1	CC0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC1	CC1	CC1	CC1	CC0

Notes on Table 36-4:

- Configuration 0x0 is the default configuration. The channel location is the default one, and channels are distributed on outputs modulo the number of channels. Channel 0 is routed to the Output matrix output OTMX[0], and Channel 1 to OTMX[1]. If there are more outputs than channels, then channel 0 is duplicated to the Output matrix output OTMX[CC_NUM], channel 1 to OTMX[CC_NUM+1] and so on.
- Configuration 0x1 distributes the channels on output modulo half the number of channels. This assigns twice the number of output locations to the lower channels than the default configuration. This can be used, for example, to control the four transistors of a full bridge using only two compare channels.
  Using pattern generation, some of these four outputs can be overwritten by a constant level, enabling flexible drive of a full bridge in all quadrant configurations.
- Configuration 0x2 distributes compare channel 0 (CC0) to all port pins. With pattern generation, this configuration can control a stepper motor.
- Configuration 0x3 distributes the compare channel CC0 to the first output, and the channel CC1 to all other outputs. Together with pattern generation and the fault extension, this configuration can control up to seven LED strings, with a boost stage.

Table 36-5. Example: four compare channels on four outputs

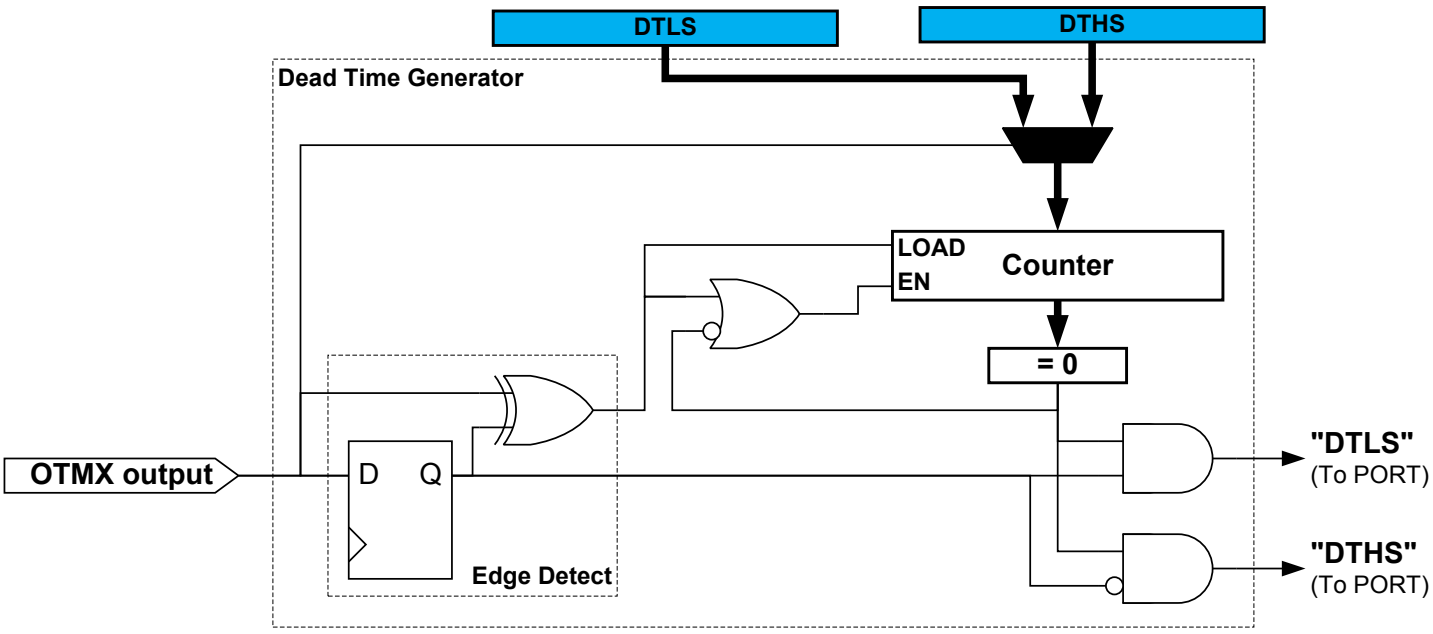
Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0

Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x2	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC0

The dead-time insertion (DTI) unit generates OFF time with the non-inverted low side (LS) and inverted high side (HS) of the wave generator output forced at low level. This OFF time is called dead time. Dead-time insertion ensures that the LS and HS will never switch simultaneously.

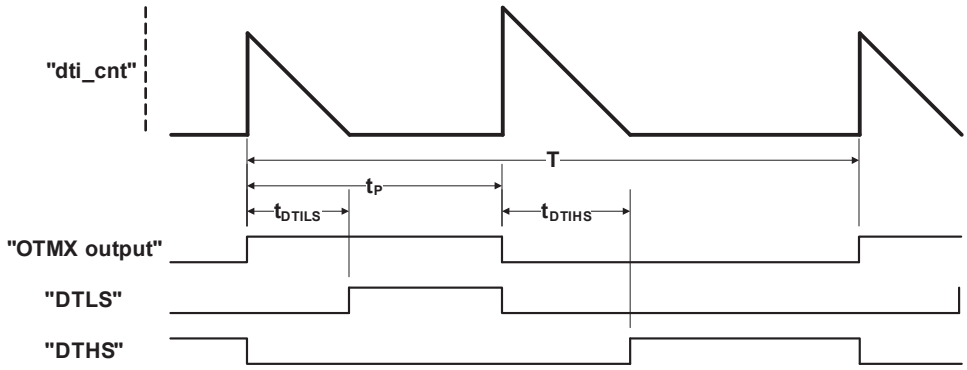
The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. Figure 36-35 shows the block diagram of one DTI generator. The four channels have a common register which controls the dead time, which is independent of high side and low side setting.

Figure 36-35. Dead-Time Generator Block Diagram



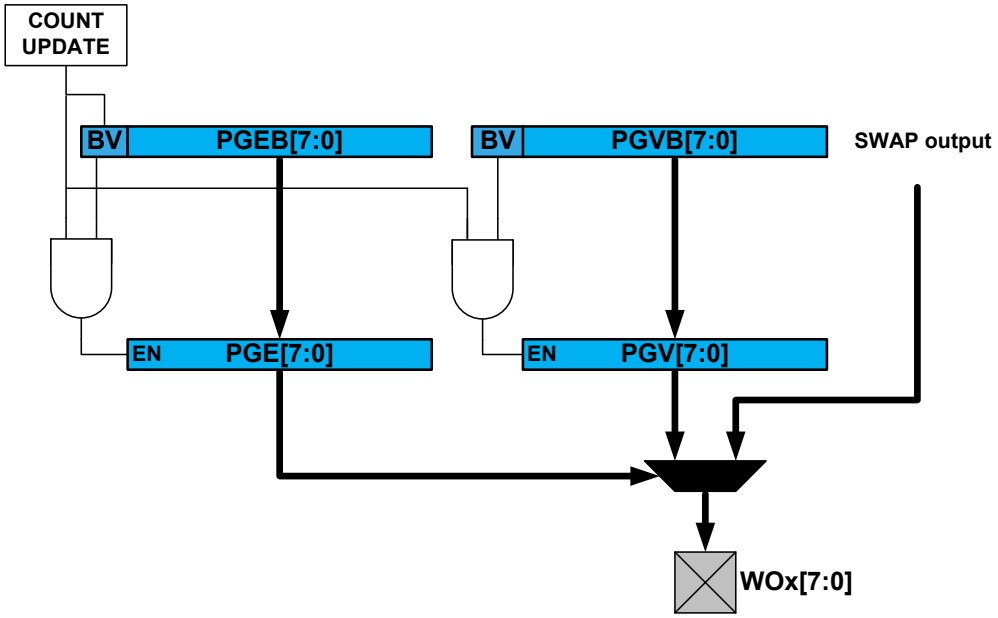
As shown in Figure 36-36, the 8-bit dead-time counter is decremented by one for each peripheral clock cycle until it reaches zero. A non-zero counter value will force both the low side and high side outputs into their OFF state. When the output matrix (OTMX) output changes, the dead-time counter is reloaded according to the edge of the input. When the output changes from low to high (positive edge) it initiates a counter reload of the DTLS register. When the output changes from high to low (negative edge) it reloads the DTHS register.

Figure 36-36. Dead-Time Generator Timing Diagram



The pattern generator unit produces a synchronized bit pattern across the port pins it is connected to. The pattern generation features are primarily intended for handling the commutation sequence in brushless DC motors (BLDC), stepper motors, and full bridge control. See also Figure 36-37.

Figure 36-37. Pattern Generator Block Diagram



As with other double-buffered timer/counter registers, the register update is synchronized to the UPDATE condition set by the timer/counter waveform generation operation. If synchronization is not required by the application, the software can simply access directly the PATT.PGE, PATT.PGV bits registers.

36.6.4 DMA, Interrupts, and Events

Table 36-6. Module Requests for TCC

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
Overflow / Underflow	Yes	Yes		Yes(1)	On DMA acknowledge
Channel Compare Match or Capture	Yes	Yes	Yes(2)	Yes(3)	For circular buffering: on DMA acknowledge For capture channel: when CCx register is read
Retrigger	Yes	Yes			
Count	Yes	Yes			
Capture Overflow Error	Yes				
Debug Fault State	Yes				
Recoverable Faults	Yes				
Non-Recoverable Faults	Yes				
TCCx Event 0 input			Yes(4)		
TCCx Event 1 input			Yes(5)		

Notes:

1. DMA request set on overflow, underflow or re-trigger conditions.
2. Can perform capture or generate recoverable fault on an event input.
3. In capture or circular modes.
4. On event input, either action can be executed:
   – re-trigger counter
   – control counter direction
   – stop the counter
   – decrement the counter
   – perform period and pulse width capture
   – generate non-recoverable fault
5. On event input, either action can be executed:
   – re-trigger counter
   – increment or decrement counter depending on direction
   – start the counter
   – increment or decrement counter based on direction
   – increment counter regardless of direction
   – generate non-recoverable fault

36.6.4.1 DMA Operation

The TCC can generate the following DMA requests:

Counter overflow (OVF)	If the Ones-shot Trigger mode in the control A register (CTRLA.DMAOS) is written to '0', the TCC generates a DMA request on each cycle when an update condition (overflow, underflow or re-trigger) is detected. When an update condition (overflow, underflow or re-trigger) is detected while CTRLA.DMAOS=1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD=DMAOS). In both cases, the request is cleared by hardware on DMA acknowledge.
Channel Match (MCx)	A DMA request is set only on a compare match if CTRLA.DMAOS=0. The request is cleared by hardware on DMA acknowledge. When CTRLA.DMAOS=1, the DMA requests are not generated.
Channel Capture (MCx)	For a capture channel, the request is set when valid data is present in the CCx register, and cleared once the CCx register is read. In this operation mode, the CTRLA.DMAOS bit value is ignored.

DMA Operation with Circular Buffer

When circular buffer operation is enabled, the buffer registers must be written in a correct order and synchronized to the update times of the timer. The DMA triggers of the TCC provide a way to ensure a safe and correct update of circular buffers.

Note: Circular buffer are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

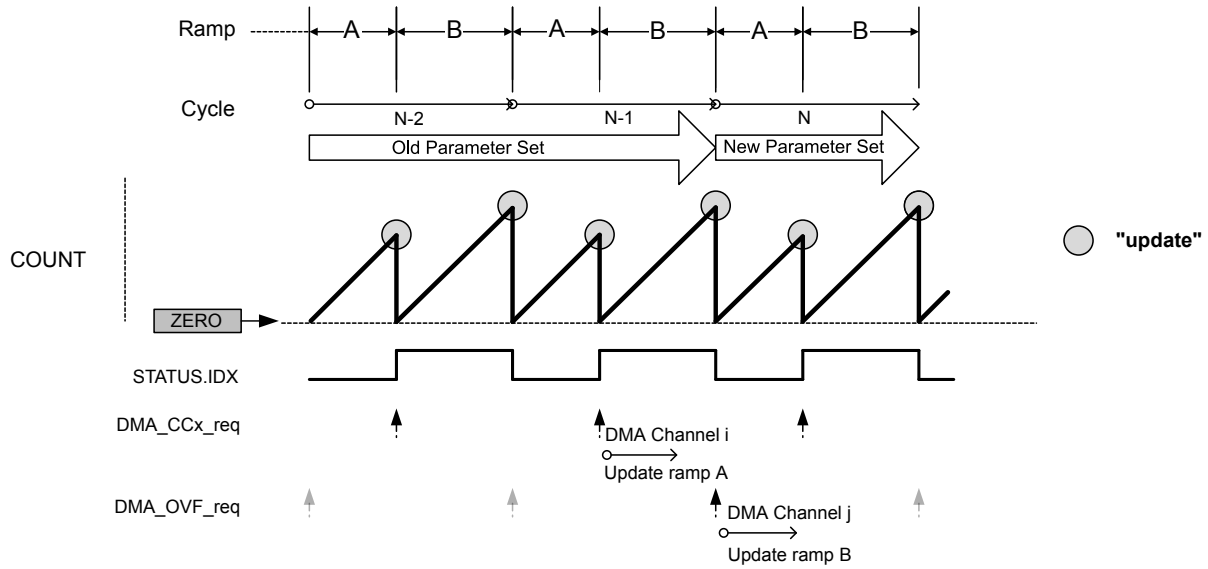
DMA Operation with Circular Buffer in RAMP and RAMP2A Mode

When a CCx channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of ramp B.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of ramp A with an effective DMA transfer on previous ramp B (DMA acknowledge).

The update of all circular buffer values for ramp A can be done through a DMA channel triggered on a MC trigger. The update of all circular buffer values for ramp B, can be done through a second DMA channel triggered by the overflow DMA request.

Figure 36-38. DMA Triggers in RAMP and RAMP2 Operation Mode and Circular Buffer Enabled

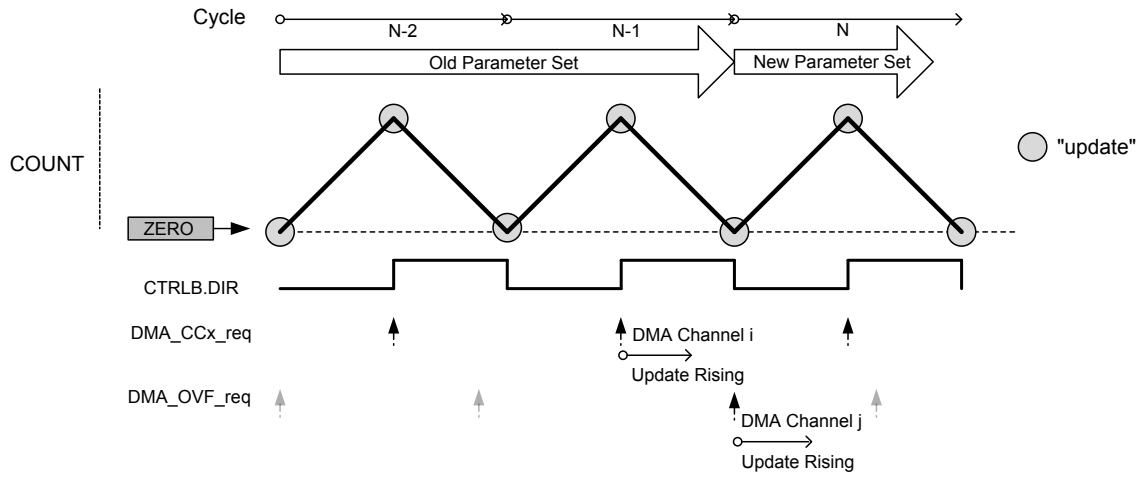


DMA Operation with Circular Buffer in DSBOTH Mode
When a CC channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of down-counting phase.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of up-counting phase with an effective DMA transfer on previous down-counting phase (DMA acknowledge).

When up-counting, all circular buffer values can be updated through a DMA channel triggered by MC trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.

Figure 36-39. DMA Triggers in DSBOTH Operation Mode and Circular Buffer Enabled



36.6.4.2 Interrupts

The TCC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) - refer also to description of EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Non-Recoverable Update Fault (UFS)
- Debug Fault State (DFS)
- Recoverable Faults (FAULTn)
- Non-recoverable Faults (FAULTx)
- Compare Match or Capture Channels (MCx)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/Sleep Mode Controller section for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the TCC is reset. See INTFLAG for details on how to clear interrupt flags. The TCC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

Sleep Mode Controller

36.6.4.3 Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)
- Counter (CNT) For further details, refer to EVCTRL.CNTSEL description.
- Compare Match or Capture on compare/capture channels: MCx

Writing a '1' ('0') to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables (disables) the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCx):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx EV1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event

- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx EV0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start - start counting on the event rising edge. Further events will not restart the counter; the counter will keep on counting using prescaled GCLK_TCCx, until it reaches TOP or ZERO, depending on the direction.
- Counter increment on event. This will increment the counter, irrespective of the counter direction.
- Count during active state of an asynchronous event (increment or decrement, depending on counter direction). In this case, the counter will be incremented or decremented on each cycle of the prescaled clock, as long as the event is active.
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0 and EVCTRL.EVACT1). For further details, refer to EVCTRL.

Writing a '1' ('0') to an Event Input bit in the Event Control register (EVCTRL.MCEIx or EVCTRL.TCEIx) enables (disables) the corresponding action on input event.

Note: When several events are connected to the TCC, the enabled action will apply for each of the incoming events. Refer to *EVSYS – Event System* for details on how to configure the event system.

Related Links

EVSYS – Event System

36.6.5 Sleep Mode Operation

The TCC can be configured to operate in any sleep mode. To be able to run in standby the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. The MODULE can in any sleep mode wake up the device using interrupts or perform actions through the Event System.

36.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)

- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

Register Synchronization

36.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RESOLUTION[1:0]					ENABLE	SWRST
0x01		15:8		ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
0x02		23:16								
0x03		31:24					CPTEN3	CPTEN2	CPTEN1	CPTEN0
0x04	CTRLBCLR	7:0	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x06 ... 0x07	Reserved									
0x08	SYNCBUSY	7:0	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x09		15:8					CC3	CC2	CC1	CC0
0x0A		23:16								
0x0B		31:24								
0x0C	FCTRLA	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x0D		15:8		CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
0x0E		23:16	BLANKVAL[7:0]							
0x0F		31:24					FILTERVAL[3:0]			
0x10	FCTRLB	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x11		15:8		CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
0x12		23:16	BLANKVAL[7:0]							
0x13		31:24					FILTERVAL[3:0]			
0x14	WEXCTRL	7:0							OTMX[1:0]	
0x15		15:8					DTIENx	DTIENx	DTIENx	DTIENx
0x16		23:16	DTLS[7:0]							
0x17		31:24	DTHS[7:0]							
0x18	DRVCTRL	7:0	NREx	NREx	NREx	NREx	NREx	NREx	NREx	NREx
0x19		15:8	NRVx	NRVx	NRVx	NRVx	NRVx	NRVx	NRVx	NRVx
0x1A		23:16	INVENx	INVENx	INVENx	INVENx	INVENx	INVENx	INVENx	INVENx
0x1B		31:24	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
0x1C ... 0x1D	Reserved									
0x1E	DBGCTRL	7:0						FDDBD		DBGRUN
0x1F	Reserved									
0x20	EVCTRL	7:0	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
0x21		15:8	TCEIx	TCEIx	TCINVx	TCINVx		CNTEO	TRGEO	OVFEO
0x22		23:16					MCEIx	MCEIx	MCEIx	MCEIx
0x23		31:24					MCEOx	MCEOx	MCEOx	MCEOx
0x24	INTENCLR	7:0					ERR	CNT	TRG	OVF
0x25		15:8	FAULTx	FAULTx	FAULTB	FAULTA	DFS	UFS		
0x26		23:16					MCx	MCx	MCx	MCx
0x27		31:24								
0x28	INTENSET	7:0					ERR	CNT	TRG	OVF
0x29		15:8	FAULTx	FAULTx	FAULTB	FAULTA	DFS	UFS		

Offset	Name	Bit Pos.								
0x2A		23:16					MCx	MCx	MCx	MCx
0x2B		31:24								
0x2C	INTFLAG	7:0					ERR	CNT	TRG	OVF
0x2D		15:8	FAULTx	FAULTx	FAULTB	FAULTA	DFS	UFS		
0x2E		23:16					MCx	MCx	MCx	MCx
0x2F		31:24								
0x30	STATUS	7:0	PERBUFV		PATTBUFV		DFS	UFS	IDX	STOP
0x31		15:8	FAULTx	FAULTx	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
0x32		23:16					CCBUFVx	CCBUFVx	CCBUFVx	CCBUFVx
0x33		31:24					CMPx	CMPx	CMPx	CMPx
0x34	COUNT	7:0	COUNT[7:0]							
0x35		15:8	COUNT[15:8]							
0x36		23:16	COUNT[23:16]							
0x37		31:24	COUNT[31:24]							
0x38	PATT	7:0	PGE0[7:0]							
0x39		15:8	PGV0[7:0]							
0x3A ... 0x3B	Reserved									
0x3C	WAVE	7:0	CIPEREN		RAMP[1:0]			WAVEGEN[2:0]		
0x3D		15:8					CICCEN3	CICCEN2	CICCEN1	CICCEN0
0x3E		23:16					POL3	POL2	POL1	POL0
0x3F		31:24					SWAP3	SWAP2	SWAP1	SWAP0
0x40	PER	7:0	PER[1:0]		DITHER[5:0]					
0x41		15:8	PER[9:2]							
0x42		23:16	PER[17:10]							
0x43		31:24	PER[25:18]							
0x44	CC0	7:0	CC[1:0]		DITHER[5:0]					
0x45		15:8	CC[9:2]							
0x46		23:16	CC[17:10]							
0x47		31:24	CC[25:18]							
0x48	CC1	7:0	CC[1:0]		DITHER[5:0]					
0x49		15:8	CC[9:2]							
0x4A		23:16	CC[17:10]							
0x4B		31:24	CC[25:18]							
0x4C	CC2	7:0	CC[1:0]		DITHER[5:0]					
0x4D		15:8	CC[9:2]							
0x4E		23:16	CC[17:10]							
0x4F		31:24	CC[25:18]							
0x50	CC3	7:0	CC[1:0]		DITHER[5:0]					
0x51		15:8	CC[9:2]							
0x52		23:16	CC[17:10]							
0x53		31:24	CC[25:18]							
0x54 ... 0x63	Reserved									
0x64	PATTBUF	7:0	PGEB0[7:0]							

Offset	Name	Bit Pos.								
0x65		15:8	PGVB0[7:0]							
0x66 ... 0x6B	Reserved									
0x6C	PERBUF	7:0	PERBUF[1:0]		DITHERBUF[5:0]					
0x6D		15:8	PERBUF[9:2]							
0x6E		23:16	PERBUF[17:10]							
0x6F		31:24	PERBUF[25:18]							
0x70	CCBUF0	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x71		15:8	CCBUF[9:2]							
0x72		23:16	CCBUF[17:10]							
0x73		31:24	CCBUF[25:18]							
0x74	CCBUF1	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x75		15:8	CCBUF[9:2]							
0x76		23:16	CCBUF[17:10]							
0x77		31:24	CCBUF[25:18]							
0x78	CCBUF2	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x79		15:8	CCBUF[9:2]							
0x7A		23:16	CCBUF[17:10]							
0x7B		31:24	CCBUF[25:18]							
0x7C	CCBUF3	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x7D		15:8	CCBUF[9:2]							
0x7E		23:16	CCBUF[17:10]							
0x7F		31:24	CCBUF[25:18]							

36.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

36.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00002e48]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized (ENABLE, SWRST)

Bit	31	30	29	28	27	26	25	24
					CPTEN3	CPTEN2	CPTEN1	CPTEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		RESOLUTION[1:0]					ENABLE	SWRST
Access		R/W	R/W				R/W	R/W
Reset		0	0				0	0

Bits 24, 25, 26, 27 – CPTEN0, CPTEN1, CPTEN2, CPTEN3: Capture Channel x Enable
These bits are used to select the capture or compare operation on channel x.

Writing a '1' to CPTENx enables capture on channel x.

Writing a '0' to CPTENx disables capture on channel x.

Bit 14 – ALOCK: Auto Lock
This bit is not synchronized.

Value	Description
0	The Lock Update bit in the Control B register (CTRLB.LUPD) is not affected by overflow/ underflow, and re-trigger events
1	CTRLB.LUPD is set to '1' on each overflow/underflow or re-trigger event.

Bits 13:12 – PRESCYNC[1:0]: Prescaler and Counter Synchronization
These bits select if on re-trigger event, the Counter is cleared or reloaded on either the next GCLK_TCCx clock, or on the next prescaled GCLK_TCCx clock. It is also possible to reset the prescaler on re-trigger event.

These bits are not synchronized.

Value	Name	Description	
		Counter Reloaded	Prescaler
0x0	GCLK	Reload or reset Counter on next GCLK	-
0x1	PRESC	Reload or reset Counter on next prescaler clock	-

Value	Name	Description	
		Counter Reloaded	Prescaler
0x2	RESYNC	Reload or reset Counter on next GCLK	Reset prescaler counter
0x3	Reserved		

Bit 11 – RUNSTDBY: Run in Standby
This bit is used to keep the TCC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TCC is halted in standby.
1	The TCC continues to run in standby.

Bits 10:8 – PRESCALER[2:0]: Prescaler
These bits select the Counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TCC
0x1	DIV2	Prescaler: GCLK_TCC/2
0x2	DIV4	Prescaler: GCLK_TCC/4
0x3	DIV8	Prescaler: GCLK_TCC/8
0x4	DIV16	Prescaler: GCLK_TCC/16
0x5	DIV64	Prescaler: GCLK_TCC/64
0x6	DIV256	Prescaler: GCLK_TCC/256
0x7	DIV1024	Prescaler: GCLK_TCC/1024

Bits 6:5 – RESOLUTION[1:0]: Dithering Resolution
These bits increase the TCC resolution by enabling the dithering options.

These bits are not synchronized.

Table 36-7. Dithering

Value	Name	Description
0x0	NONE	The dithering is disabled.
0x1	DITH4	Dithering is done every 16 PWM frames. PER[3:0] and CCx[3:0] contain dithering pattern selection.
0x2	DITH5	Dithering is done every 32 PWM frames. PER[4:0] and CCx[4:0] contain dithering pattern selection.
0x3	DITH6	Dithering is done every 64 PWM frames.

Value	Name	Description
		PER[5:0] and CCx[5:0] contain dithering pattern selection.

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TCC (except DBGCTRL) to their initial state, and the TCC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

36.8.2 Control B Clear

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBSET) register.

Name: CTRLBCLR
Offset: 0x04 [ID-00002e48]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0]: TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will read back zero. The commands are executed on the next prescaled GCLK_TCC clock cycle.

Writing zero to this bit group has no effect.

Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Clear start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force COUNT read synchronization

Bits 4:3 – IDXCMD[1:0]: Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.

Writing zero to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0x0	DISABLE	DISABLE Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT: One-Shot
This bit controls one-shot operation of the TCC. When one-shot operation is enabled, the TCC will stop counting on the next overflow/underflow condition or on a stop command.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable the one-shot operation.

Value	Description
0	The TCC will update the counter value on overflow/underflow condition and continue operation.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TCC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable updating.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

36.8.3 Control B Set

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBCLR) register.

Name: CTRLBSET
Offset: 0x05 [ID-00002e48]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0]: TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will be read back as zero. The commands are executed on the next prescaled GCLK_TCC clock cycle.

Writing zero to this bit group has no effect

Writing a valid value to this bit group will set the associated command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bits 4:3 – IDXCMD[1:0]: Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.

Writing a zero to these bits has no effect.

Writing a valid value to these bits will set a command.

Value	Name	Description
0x0	DISABLE	Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT: One-Shot
This bit controls one-shot operation of the TCC. When in one-shot operation, the TCC will stop counting on the next overflow/underflow condition or a stop command.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable the one-shot operation.

Value	Description
0	The TCC will count continuously.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TCC buffered registers.

When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will lock updating.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

36.8.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08 [ID-00002e48]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CC3	CC2	CC1	CC0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 8, 9, 10, 11 – CC: Compare/Capture Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare/Capture Channel x register between the clock domains is complete.

This bit is set when the synchronization of Compare/Capture Channel x register between clock domains is started.

CCx bit is available only for existing Compare/Capture Channels. For details on CC channels number, refer to each TCC feature list.

This bit is set when the synchronization of CCx register between clock domains is started.

Bit 7 – PER: PER Synchronization Busy
This bit is cleared when the synchronization of PER register between the clock domains is complete.

This bit is set when the synchronization of PER register between clock domains is started.

Bit 6 – WAVE: WAVE Synchronization Busy
This bit is cleared when the synchronization of WAVE register between the clock domains is complete.

This bit is set when the synchronization of WAVE register between clock domains is started.

Bit 5 – PATT: PATT Synchronization Busy
This bit is cleared when the synchronization of PATTERN register between the clock domains is complete.

This bit is set when the synchronization of PATTERN register between clock domains is started.

Bit 4 – COUNT: COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT register between the clock domains is complete.

This bit is set when the synchronization of COUNT register between clock domains is started.

Bit 3 – STATUS: STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS register between the clock domains is complete.

This bit is set when the synchronization of STATUS register between clock domains is started.

Bit 2 – CTRLB: CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB register between the clock domains is complete.

This bit is set when the synchronization of CTRLB register between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

36.8.5 Fault Control A and B

Name: FCTRLA, FCTRLB
Offset: 0x0C + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					FILTERVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BLANKVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 27:24 – FILTERVAL[3:0]: Recoverable Fault n Filter Value
These bits define the filter value applied on MCEx (x=0,1) event input line. The value must be set to zero when MCEx event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0]: Recoverable Fault n Blanking Value
These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLn.BLANK).

When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCC periods after the detection of the waveform edge.

Bits 14:12 – CAPTURE[2:0]: Recoverable Fault n Capture Action
These bits select the capture and Fault n interrupt/event conditions.

Table 36-8. Fault n Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault n is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local minimum detection.
0x3	CAPTMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC).

Value	Name	Description
		INTFLAG.FAULTn flag rises on each local maximun detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun or minimum detection.

Bits 11:10 – CHSEL[1:0]: Recoverable Fault n Capture Channel
These bits select the channel for capture operation triggered by recoverable Fault n.

Value	Name	Description
0x0	CC0	Capture value stored into CC0
0x1	CC1	Capture value stored into CC1
0x2	CC2	Capture value stored into CC2
0x3	CC3	Capture value stored into CC3

Bits 9:8 – HALT[1:0]: Recoverable Fault n Halt Operation
These bits select the halt action for recoverable Fault n.

Value	Name	Description
0x0	DISABLE	Halt action disabled
0x1	HW	Hardware halt action
0x2	SW	Software halt action
0x3	NR	Non-recoverable fault

Bit 7 – RESTART: Recoverable Fault n Restart
Setting this bit enables restart action for Fault n.

Value	Description
0	Fault n restart action is disabled.
1	Fault n restart action is enabled.

Bits 6:5 – BLANK[1:0]: Recoverable Fault n Blanking Operation
These bits, select the blanking start point for recoverable Fault n.

Value	Name	Description
0x0	START	Blanking applied from start of the Ramp period
0x1	RISE	Blanking applied from rising edge of the waveform output
0x2	FALL	Blanking applied from falling edge of the waveform output
0x3	BOTH	Blanking applied from each toggle of the waveform output

Bit 4 – QUAL: Recoverable Fault n Qualification
Setting this bit enables the recoverable Fault n input qualification.

Value	Description
0	The recoverable Fault n input is not disabled on CMPx value condition.
1	The recoverable Fault n input is disabled when output signal is at inactive level (CMPx == 0).

Bit 3 – KEEP: Recoverable Fault n Keep
Setting this bit enables the Fault n keep action.

Value	Description
0	The Fault n state is released as soon as the recoverable Fault n is released.
1	The Fault n state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0]: Recoverable Fault n Source
These bits select the TCC event input for recoverable Fault n.

Event system channel connected to MCEx event input, must be configured to route the event asynchronously, when used as a recoverable Fault n input.

Value	Name	Description
0x0	DISABLE	Fault input disabled
0x1	ENABLE	MCEx (x=0,1) event input
0x2	INVERT	Inverted MCEx (x=0,1) event input
0x3	ALTFAULT	Alternate fault (A or B) state at the end of the previous period.

36.8.6 Waveform Extension Control

Name: WEXCTRL
Offset: 0x14 [ID-00002e48]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DTHS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DTLS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					DTIENx	DTIENx	DTIENx	DTIENx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
							OTMX[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 31:24 – DTHS[7:0]: Dead-Time High Side Outputs Value
This register holds the number of GCLK_TCC clock cycles for the dead-time high side.

Bits 23:16 – DTLS[7:0]: Dead-time Low Side Outputs Value
This register holds the number of GCLK_TCC clock cycles for the dead-time low side.

Bits 11,10,9,8 – DTIENx : Dead-time Insertion Generator x Enable
Setting any of these bits enables the dead-time insertion generator for the corresponding output matrix. This will override the output matrix [x] and [x+WO_NUM/2], with the low side and high side waveform respectively.

Value	Description
0	No dead-time insertion override.
1	Dead time insertion override on signal outputs[x] and [x+WO_NUM/2], from matrix outputs[x] signal.

Bits 1:0 – OTMX[1:0]: Output Matrix
These bits define the matrix routing of the TCC waveform generation outputs to the port pins, according to Table 36-4.

36.8.7 Driver Control

Name: DRVCTRL
Offset: 0x18 [ID-00002e48]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	INVENx	INVENx	INVENx	INVENx	INVENx	INVENx	INVENx	INVENx
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NRVx	NRVx	NRVx	NRVx	NRVx	NRVx	NRVx	NRVx
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NREx	NREx	NREx	NREx	NREx	NREx	NREx	NREx
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – FILTERVAL1[3:0]: Non-Recoverable Fault Input 1 Filter Value
These bits define the filter value applied on TCE1 event input line. When the TCE1 event input line is configured as a synchronous event, this value must be 0x0.

Bits 27:24 – FILTERVAL0[3:0]: Non-Recoverable Fault Input 0 Filter Value
These bits define the filter value applied on TCE0 event input line. When the TCE0 event input line is configured as a synchronous event, this value must be 0x0.

Bits 23,22,21,20,19,18,17,16 – INVENx: Waveform Output x Inversion
These bits are used to select inversion on the output of channel x.

Writing a '1' to INVENx inverts output from WO[x].

Writing a '0' to INVENx disables inversion of output from WO[x].

Bits 15,14,13,12,11,10,9,8 – NRVx: NRVx Non-Recoverable State x Output Value
These bits define the value of the enabled override outputs, under non-recoverable fault condition.

Bits 7,6,5,4,3,2,1,0 – NREx: Non-Recoverable State x Output Enable
These bits enable the override of individual outputs by NRVx value, under non-recoverable fault condition.

Value	Description
0	Non-recoverable fault tri-state the output.
1	Non-recoverable faults set the output to NRVx level.

36.8.8 Debug control

Name: DBGCTRL
Offset: 0x1E [ID-00002e48]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						FDDBD		DBGRUN
Access						R/W		R/W
Reset						0		0

Bit 2 – FDDBD: Fault Detection on Debug Break Detection
This bit is not affected by software reset and should not be changed by software while the TCC is enabled.

By default this bit is zero, and the on-chip debug (OCD) fault protection is disabled. When this bit is written to '1', OCD break request from the OCD system will trigger non-recoverable fault. When this bit is set, OCD fault protection is enabled and OCD break request from the OCD system will trigger a non-recoverable fault.

Value	Description
0	No faults are generated when TCC is halted in debug mode.
1	A non recoverable fault is generated and FAULTD flag is set when TCC is halted in debug mode.

Bit 0 – DBGRUN: Debug Running State
This bit is not affected by software reset and should not be changed by software while the TCC is enabled.

Value	Description
0	The TCC is halted when the device is halted in debug mode.
1	The TCC continues normal operation when the device is halted in debug mode.

36.8.9 Event Control

Name: EVCTRL
Offset: 0x20 [ID-00002e48]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					MCEOx	MCEOx	MCEOx	MCEOx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					MCEIx	MCEIx	MCEIx	MCEIx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TCEIx	TCEIx	TCINVx	TCINVx		CNTEO	TRGEO	OVFEO
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27,26,25,24 – MCEOx: Match or Capture Channel x Event Output Enable
These bits control if the Match/capture event on channel x is enabled and will be generated for every match or capture.

Value	Description
0	Match/capture x event is disabled and will not be generated.
1	Match/capture x event is enabled and will be generated for every compare/capture on channel x.

Bits 19,18,17,16 – MCEIx: Match or Capture Channel x Event Input Enable
These bits indicate if the Match/capture x incoming event is enabled

These bits are used to enable match or capture input events to the CCx channel of TCC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 15,14 – TCEIx: Timer/Counter Event Input x Enable
This bit is used to enable input event x to the TCC.

Value	Description
0	Incoming event x is disabled.
1	Incoming event x is enabled.

Bits 13,12 – TCINVx: Timer/Counter Event x Invert Enable
This bit inverts the event x input.

Value	Description
0	Input event source x is not inverted.
1	Input event source x is inverted.

Bit 10 – CNTEO: Timer/Counter Event Output Enable
This bit is used to enable the counter cycle event. When enabled, an event will be generated on begin or end of counter cycle depending of CNTSEL[1:0] settings.

Value	Description
0	Counter cycle output event is disabled and will not be generated.
1	Counter cycle output event is enabled and will be generated depend of CNTSEL[1:0] value.

Bit 9 – TRGEO: Retrigger Event Output Enable
This bit is used to enable the counter retrigger event. When enabled, an event will be generated when the counter retriggers operation.

Value	Description
0	Counter retrigger event is disabled and will not be generated.
1	Counter retrigger event is enabled and will be generated for every counter retrigger.

Bit 8 – OVFEO: Overflow/Underflow Event Output Enable
This bit is used to enable the overflow/underflow event. When enabled an event will be generated when the counter reaches the TOP or the ZERO value.

Value	Description
0	Overflow/underflow counter event is disabled and will not be generated.
1	Overflow/underflow counter event is enabled and will be generated for every counter overflow/underflow.

Bits 7:6 – CNTSEL[1:0]: Timer/Counter Interrupt and Event Output Selection
These bits define on which part of the counter cycle the counter event output is generated.

Value	Name	Description
0x0	BEGIN	An interrupt/event is generated at begin of each counter cycle
0x1	END	An interrupt/event is generated at end of each counter cycle
0x2	BETWEEN	An interrupt/event is generated between each counter cycle.
0x3	BOUNDARY	An interrupt/event is generated at begin of first counter cycle, and end of last counter cycle.

Bits 5:3 – EVACT1[2:0]: Timer/Counter Event Input 1 Action
These bits define the action the TCC will perform on TCE1 event input.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start restart or re-trigger TC on event
0x2	DIR (asynch)	Direction control
0x3	STOP	Stop TC on event
0x4	DEC	Decrement TC on event
0x5	PPW	Period captured into CC0 Pulse Width on CC1
0x6	PWP	Period captured into CC1 Pulse Width on CC0
0x7	FAULT	Non-recoverable Fault

Bits 2:0 – EVACT0[2:0]: Timer/Counter Event Input 0 Action
These bits define the action the TCC will perform on TCE0 event input 0.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start restart or re-trigger TC on event
0x2	COUNTEV	Count on event.
0x3	START	Start TC on event
0x4	INC	Increment TC on EVENT
0x5	COUNT (async)	Count on active state of asynchronous event
0x6	STAMP	Capture overflow times (Max value)
0x7	FAULT	Non-recoverable Fault

36.8.10 Interrupt Enable Clear

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x24 [ID-00002e48]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MCx	MCx	MCx	MCx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULTx	FAULTx	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 19,18,17,16 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bits 15,14 – FAULTx: Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

Bit 13 – FAULTB: Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault B Interrupt Disable/Enable bit, which disables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault A Interrupt Disable/Enable bit, which disables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS: Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – UFS: Non-Recoverable Update Fault Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which disables the Non-Recoverable Update Fault interrupt.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Compare interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT: Counter Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Counter Interrupt Disable/Enable bit, which disables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG: Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Retrigger Interrupt Disable/Enable bit, which disables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

36.8.11 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x28 [ID-00002e48]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MCx	MCx	MCx	MCx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULTx	FAULTx	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 19,18,17,16 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bits 15,14 – FAULTx: Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Non-Recoverable Fault x Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

Bit 13 – FAULTB: Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault B Interrupt Disable/Enable bit, which enables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS: Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Debug Fault State Interrupt Disable/Enable bit, which enables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – UFS: Non-Recoverable Update Fault Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which enables the Non-Recoverable Update Fault interrupt.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Compare interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT: Counter Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG: Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Disable/Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

36.8.12 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x2C [ID-00002e48]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MCx	MCx	MCx	MCx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULTx	FAULTx	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 19,18,17,16 – MCx: Match or Capture Channel x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a match with the compare condition or once CCx register contain a valid capture value.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In Capture operation, this flag is automatically cleared when CCx register is read.

Bits 15,14 – FAULTx: Non-Recoverable Fault x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Non-Recoverable Fault x occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Non-Recoverable Fault x interrupt flag.

Bit 13 – FAULTB: Recoverable Fault B Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 11 – DFS: Non-Recoverable Debug Fault State Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an Debug Fault State occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Debug Fault State interrupt flag.

Bit 10 – UFS: Non-Recoverable Update Fault
This flag is set when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD).

Writing a zero to this bit has no effect.

Writing a one to this bit clears the Non-Recoverable Update Fault interrupt flag.

Bit 3 – ERR: Error Interrupt Flag
This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel x interrupt flag is one. In which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the error interrupt flag.

Bit 2 – CNT: Counter Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter event occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the CNT interrupt flag.

Bit 1 – TRG: Retrigger Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter retrigger occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the re-trigger interrupt flag.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an overflow condition occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

36.8.13 Status

Name: STATUS
Offset: 0x30 [ID-00002e48]
Reset: 0x00000001
Property: -

Bit	31	30	29	28	27	26	25	24
					CMPx	CMPx	CMPx	CMPx
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CCBUFVx	CCBUFVx	CCBUFVx	CCBUFVx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULTx	FAULTx	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
Access	R/W	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUFV		PATTBUFV		DFS	UFS	IDX	STOP
Access	R/W		R/W		R/W	R/W	R	R
Reset	0		0		0	0	0	1

Bits 27,26,25,24 – CMPx: Channel x Compare Value
This bit reflects the channel x output compare value.

Value	Description
0	Channel compare output value is 0.
1	Channel compare output value is 1.

Bits 19,18,17,16 – CCBUFVx: Channel x Compare or Capture Buffer Valid
For a compare channel, this bit is set when a new value is written to the corresponding CCBUFx register. The bit is cleared either by writing a '1' to the corresponding location when CTRLB.LUPD is set, or automatically on an UPDATE condition.

For a capture channel, the bit is set when a valid capture value is stored in the CCBUFx register. The bit is automatically cleared when the CCx register is read.

Bits 15,14 – FAULTx: Non-recoverable Fault x State
This bit is set by hardware as soon as non-recoverable Fault x condition occurs.

This bit is cleared by writing a one to this bit and when the corresponding FAULTxIN status bit is low.

Once this bit is clear, the timer/counter will restart from the last COUNT value. To restart the timer/counter from BOTTOM, the timer/counter restart command must be executed before clearing the corresponding STATEx bit. For further details on timer/counter commands, refer to available commands description (CTRLBSET.CMD).

Bit 13 – FAULTB: Recoverable Fault B State
This bit is set by hardware as soon as recoverable Fault B condition occurs.

This bit can be clear by hardware when Fault B action is resumed, or by writing a '1' to this bit when the corresponding FAULTBIN bit is low. If software halt command is enabled (FAULTB.HALT=SW), clearing this bit will release the timer/counter.

Bit 12 – FAULTA: Recoverable Fault A State
This bit is set by hardware as soon as recoverable Fault A condition occurs.

This bit can be clear by hardware when Fault A action is resumed, or by writing a '1' to this bit when the corresponding FAULTAIN bit is low. If software halt command is enabled (FAULTA.HALT=SW), clearing this bit will release the timer/counter.

Bit 11 – FAULT1IN: Non-Recoverable Fault 1 Input
This bit is set while an active Non-Recoverable Fault 1 input is present.

Bit 10 – FAULT0IN: Non-Recoverable Fault 0 Input
This bit is set while an active Non-Recoverable Fault 0 input is present.

Bit 9 – FAULTBIN: Recoverable Fault B Input
This bit is set while an active Recoverable Fault B input is present.

Bit 8 – FAULTAIN: Recoverable Fault A Input
This bit is set while an active Recoverable Fault A input is present.

Bit 7 – PERBUFV: Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. This bit is automatically cleared by hardware on UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit.

Bit 5 – PATTBUFV: Pattern Generator Value Buffer Valid
This bit is set when a new value is written to the PATTBUF register. This bit is automatically cleared by hardware on UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit.

Bit 3 – DFS: Debug Fault State
This bit is set by hardware in debug mode when DDBGCTRL.FDDBD bit is set. The bit is cleared by writing a '1' to this bit and when the TCC is not in debug mode.

When the bit is set, the counter is halted and the waveforms state depend on DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 2 – UFS: Non-recoverable Update Fault State
This bit is set by hardware when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD). The bit is cleared by writing a one to this bit.

When the bit is set, the waveforms state depend on DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 1 – IDX: Ramp Index
In RAMP2 and RAMP2A operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads zero. For details on ramp operations, refer to Ramp Operations.

Bit 0 – STOP: Stop
This bit is set when the TCC is disabled either on a STOP command or on an UPDATE condition when One-Shot operation mode is enabled (CTRLBSET.ONESHOT=1).

This bit is clear on the next incoming counter increment or decrement.

Value	Description
0	Counter is running.
1	Counter is stopped.

36.8.14 Counter Value

Note: Prior to any read access, this register must be synchronized by user by writing the according TCC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Name: COUNT
Offset: 0x34 [ID-00002e48]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0]: Counter Value
These bits hold the value of the counter register.

Note: When the TCC is configured as 24- or 16-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0 (depicted)
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6

36.8.15 Pattern

Name: PATT
Offset: 0x38 [ID-00002e48]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PGV0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PGE0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:15, 16:23, 24:31, 32:39, 40:47, 48:55, 56:63, 64:71 – PGV: Pattern Generation Output Value
This register holds the values of pattern for each waveform output.

Bits 0:7, 8:15, 16:23, 24:31, 32:39, 40:47, 48:55, 56:63 – PGE: Pattern Generation Output Enable
This register holds the enable status of pattern generation for each waveform output. A bit written to '1' will override the corresponding SWAP output with the corresponding PGVn value.

36.8.16 Waveform

Name: WAVE
Offset: 0x3C [ID-00002e48]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
					SWAP3	SWAP2	SWAP1	SWAP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					POL3	POL2	POL1	POL0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CICCEN3	CICCEN2	CICCEN1	CICCEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CIPEREN		RAMP[1:0]			WAVEGEN[2:0]		
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bits 24, 25, 26, 27 – SWAP: Swap DTI Output Pair x
Setting these bits enables output swap of DTI outputs [x] and [x+WO_NUM/2]. Note the DTIxEN settings will not affect the swap operation.

Bits 16, 17, 18, 19 – POL: Channel Polarity x
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.

Value	Name	Description
0	(single-slope PWM waveform generation)	Compare output is initialized to ~DIR and set to DIR when TCC counter matches CCx value
1	(single-slope PWM waveform generation)	Compare output is initialized to DIR and set to ~DIR when TCC counter matches CCx value.
0	(dual-slope PWM waveform generation)	Compare output is set to ~DIR when TCC counter matches CCx value
1	(dual-slope PWM waveform generation)	Compare output is set to DIR when TCC counter matches CCx value.

Bits 8, 9, 10, 11 – CICCEN: Circular CC Enable x
Setting this bits enables the compare circular buffer option on channel. When the bit is set, CCx register value is copied-back into the CCx register on UPDATE condition.

Bit 7 – CIPEREN: Circular Period Enable
Setting this bits enable the period circular buffer option. When the bit is set, the PER register value is copied-back into the PERB register on UPDATE condition.

Bits 5:4 – RAMP[1:0]: Ramp Operation
These bits select Ramp operation (RAMP). These bits are not synchronized.

Value	Name	Description
0x0	RAMP1	RAMP1 operation
0x1	RAMP2A	Alternative RAMP2 operation

Value	Name	Description
0x2	RAMP2	RAMP2 operation
0x3	RAMP2C	Critical RAMP2 operation

Bits 2:0 – WAVEGEN[2:0]: Waveform Generation Operation
These bits select the waveform generation operation. The settings impact the top value and control if frequency or PWM waveform generation should be used. These bits are not synchronized.

Value	Name	Description						
		Operation	Top	Update	Waveform Output On Match	Waveform Output On Update	OVFIF/Event Up	Down
0x0	NFRQ	Normal Frequency	PER	TOP/Zero	Toggle	Stable	TOP	Zero
0x1	MFRQ	Match Frequency	CC0	TOP/Zero	Toggle	Stable	TOP	Zero
0x2	NPWM	Normal PWM	PER	TOP/Zero	Set	Clear	TOP	Zero
0x3	Reserved	–	–	–	–	–	–	–
0x4	DSCRITICAL	Dual-slope PWM	PER	Zero	~DIR	Stable	–	Zero
0x5	DSBOTTOM	Dual-slope PWM	PER	Zero	~DIR	Stable	–	Zero
0x6	DSBOTH	Dual-slope PWM	PER	TOP & Zero	~DIR	Stable	TOP	Zero
0x7	DSTOP	Dual-slope PWM	PER	Zero	~DIR	Stable	TOP	–

36.8.17 Period Value

Name: PER
Offset: 0x40 [ID-00002e48]
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PER[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PER[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PER[1:0]		DITHER[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:6 – PER[25:0]: Period Value
These bits hold the value of the period register.

Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHER[5:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM pulse period every 64 PWM frames.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

36.8.18 Compare/Capture Channel x

The CCx register represents the 16-, 24- or 32-bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBUFx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Name: CC
Offset: 0x44 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[1:0]		DITHER[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:6 – CC[25:0]: Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.

Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4

CTRLA.RESOLUTION	Bits [31:m]
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHER[5:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM pulse width every 64 PWM frames.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

36.8.19 Pattern Buffer

Name: PATTBUF
Offset: 0x64
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	PGVB0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PGEB0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:15, 16:23, 24:31, 32:39, 40:47, 48:55, 56:63, 64:71 – PGVB: Pattern Generation Output Value Buffer
This register is the buffer for the PGV register. If double buffering is used, valid content in this register is copied to the PGV register on an UPDATE condition.

Bits 0:7, 8:15, 16:23, 24:31, 32:39, 40:47, 48:55, 56:63 – PGEB: Pattern Generation Output Enable Buffer
This register is the buffer of the PGE register. If double buffering is used, valid content in this register is copied into the PGE register at an UPDATE condition.

36.8.20 Period Buffer Value

Name: PERBUF
Offset: 0x6C [ID-00002e48]
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	PERBUF[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PERBUF[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PERBUF[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERBUF[1:0]		DITHERBUF[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:6 – PERBUF[25:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHERBUF[5:0]: Dithering Buffer Cycle Number
These bits represent the PER.DITHER bits buffer. When the double buffering is enabled, the value of this bit field is copied to the PER.DITHER bits on an UPDATE condition.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

36.8.21 Channel x Compare/Capture Buffer Value

CCBUFx is copied into CCx at TCC update time

Name: CCBUF
Offset: 0x70 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[1:0]		DITHERBUF[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:6 – CCBUF[25:0]: Channel x Compare/Capture Buffer Value
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBUFVx status bit.

Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4

CTRLA.RESOLUTION	Bits [31:m]
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHERBUF[5:0]: Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

37. TRNG – True Random Number Generator

37.1 Overview

The True Random Number Generator (TRNG) generates unpredictable random numbers that are not generated by an algorithm. It passes the American NIST Special Publication 800-22 and Diehard Random Tests Suites.

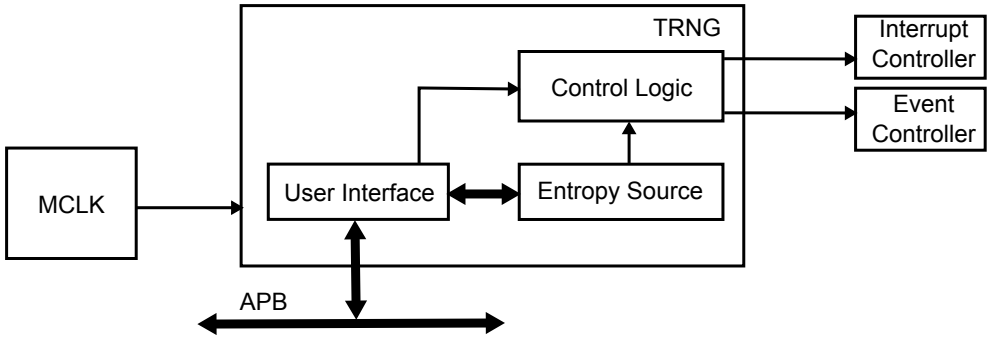
The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

37.2 Features

- Passed NIST Special Publication 800-22 Tests Suite
- Passed Diehard Random Tests Suite
- May be used as Entropy Source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit random number every 84 clock cycles

37.3 Block Diagram

Figure 37-1. TRNG Block Diagram.



37.4 Signal Description

Not applicable.

37.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

37.5.1 I/O Lines

Not applicable.

37.5.2 Power Management

The TRNG will continue to operate in any sleep mode, as long as its source clock is running. The TRNG interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

37.5.3 Clocks

The TRNG bus clock (CLK_TRNG_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TRNG_APB can be found in *Peripheral Clock Masking*.

Related Links

Peripheral Clock Masking

37.5.4 DMA

Not applicable.

37.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the TRNG interrupt(s) requires the interrupt controller to be configured first. Refer to NVIC - Nested Interrupt *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

37.5.6 Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

EVSYS – Event System

37.5.7 Debug Operation

When the CPU is halted in debug mode the TRNG continues normal operation. If the TRNG is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

37.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following register:

Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

37.5.9 Analog Connections

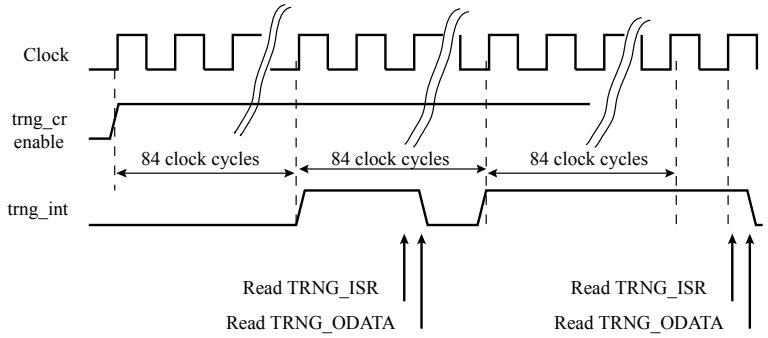
Not applicable.

37.6 Functional Description

37.6.1 Principle of Operation

As soon as the TRNG is enabled, the module automatically provides a new 32-bit random number every 84 CLK_TRNG_APB clock cycles. When new data is available, an optional interrupt or event can be generated.

Figure 37-2. TRNG Data Generation Sequence



37.6.2 Basic Operation

37.6.2.1 Initialization

The following register is enable-protected, meaning that it can only be written when the TRNG is disabled (CTRLA.ENABLE is zero):

Event Control register (EVCTRL)

Enable-protection is denoted by the Enable-Protected property in the register description.

37.6.2.2 Enabling, Disabling and Resetting

The TRNG is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TRNG is disabled by writing a zero to CTRLA.ENABLE.

37.6.3 Interrupts

The TRNG has the following interrupt source:

- Data Ready(DATARDY): Indicates that a new random data is available in DATA register and ready to be read.
  This interrupt is a synchronous wake-up source. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, or the interrupt is disabled. See INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Sleep Mode Controller

Nested Vector Interrupt Controller

37.6.4 Events

The TRNG can generate the following output event:

- Data Ready (DATARDY): Generated when a new random number is available in the DATA register.

Writing '1' to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuring the Event System.

Related Links

EVSYS – Event System

37.6.5 Sleep Mode Operation

The Run in Standby bit in Control A register (CTRLA.RUNSTDBY) controls the behavior of the TRNG during standby sleep mode:

When this bit is '0', the TRNG is disabled during sleep, but maintains its current configuration.

When this bit is '1', the TRNG continues to operate during sleep and any enabled TRNG interrupt source can wake up the CPU.

37.6.6 Synchronization

Not applicable.

37.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RUNSTDBY					ENABLE	
0x01 ... 0x03	Reserved									
0x04	EVCTRL	7:0								DATARDYEO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DATARDY
0x09	INTENSET	7:0								DATARDY
0x0A	INTFLAG	7:0								DATARDY
0x0B ... 0x1F	Reserved									
0x20	DATA	7:0	DATA[7:0]							
0x21		15:8	DATA[15:8]							
0x22		23:16	DATA[23:16]							
0x23		31:24	DATA[31:24]							

37.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to *PAC - Peripheral Access Controller* and Synchronizationfor details.

Related Links

PAC - Peripheral Access Controller

37.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000410]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	
Access		R/W					R/W	
Reset		0					0	

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the ADC behaves during standby sleep mode:

Value	Description
0	The TRNG is halted during standby sleep mode.
1	The TRNG is not stopped in standby sleep mode.

Bit 1 – ENABLE: Enable

Value	Description
0	The TRNG is disabled.
1	The TRNG is enabled.

37.8.2 Event Control

Name: EVCTRL
Offset: 0x04 [ID-00000410]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								DATARDYEO
Access								R/W
Reset								0

Bit 0 – DATARDYEO: Data Ready Event Output
This bit indicates whether the Data Ready event output is enabled or not and an output event will be generated when a new random value is completed.

Value	Description
0	Data Ready event output is disabled and an event will not be generated.
1	Data Ready event output is enabled and an event will be generated.

37.8.3 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08 [ID-00000410]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY: Data Ready Interrupt Enable
Writing a '1' to this bit will clear the Data Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

37.8.4 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x09 [ID-00000410]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY: Data Ready Interrupt Enable
Writing a '1' to this bit will set the Data Ready Interrupt Enable bit, which enables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

37.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A [ID-00000410]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY: Data Ready
This flag is set when a new random value is generated, and an interrupt will be generated if INTENCLR/SET.DATARDY=1.

This flag is cleared by writing a '1' to the flag or by reading the DATA register.

Writing a '0' to this bit has no effect.

37.8.6 Output Data

Name: DATA
Offset: 0x20 [ID-00000410]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Output Data
These bits hold the 32-bit randomly generated output data.

38. AES – Advanced Encryption Standard

38.1 Overview

The Advanced Encryption Standard peripheral (AES) provides a means for symmetric-key encryption of 128-bit blocks, in compliance to NIST specifications.

A symmetric-key algorithm requires the same key for both encryption and decryption.

Different key sizes are supported. The key size determines the number of repetitions of transformation rounds that convert the input (called the "plaintext") into the final output ("ciphertext"). The number of rounds of repetition is as follows:

- 10 rounds of repetition for 128-bit keys
- 12 rounds of repetition for 192-bit keys
- 14 rounds of repetition for 256-bit keys

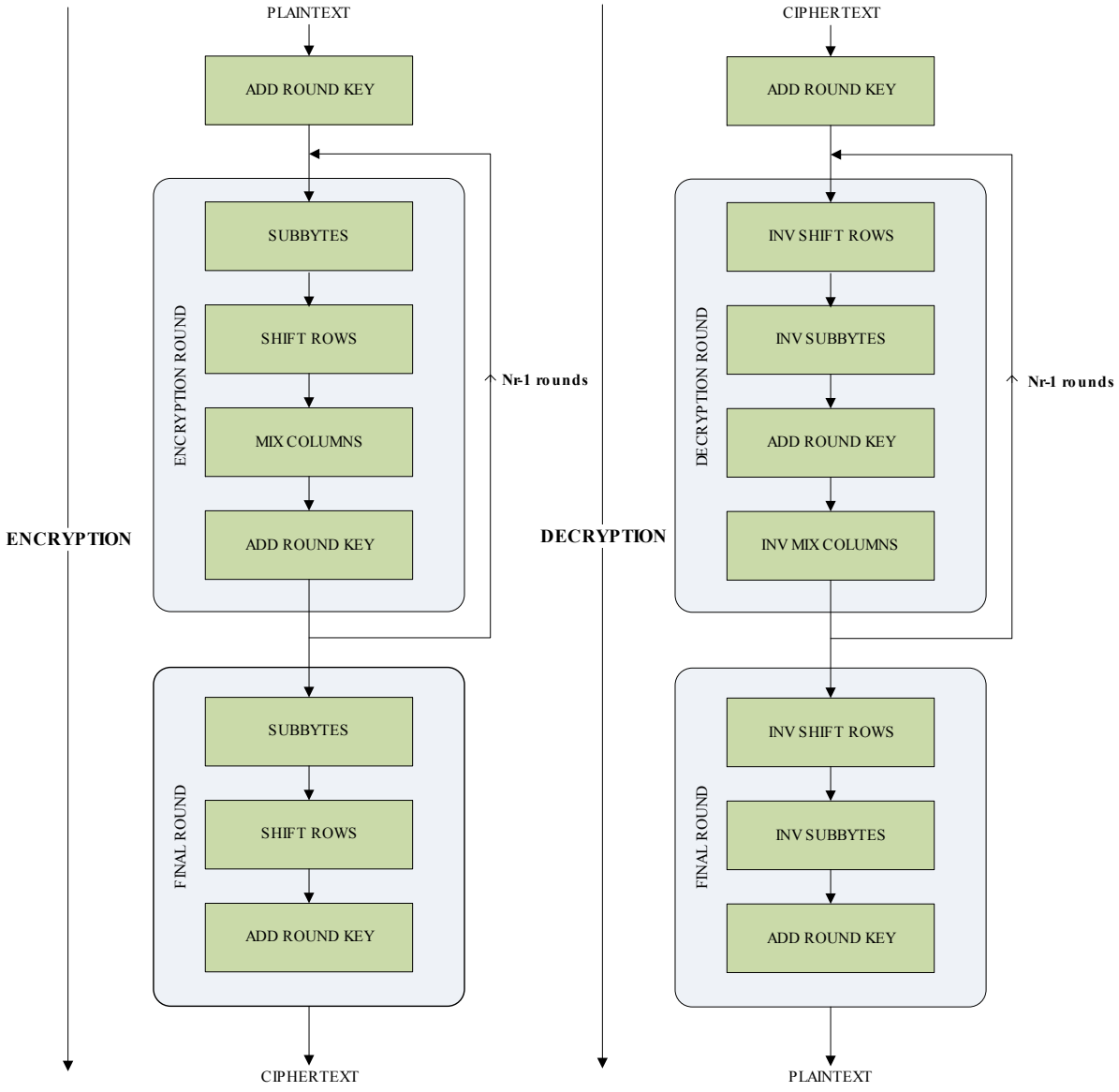
38.2 Features

- Compliant with FIPS Publication 197, Advanced Encryption Standard (AES)
- 128/192/256 bit cryptographic key supported
- Encryption time of 57/67/77 cycles with 128-bit/192-bit/256-bit cryptographic key
- Five confidentiality modes of operation as recommended in NIST Special Publication 800-38A
- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)
- Supports Counter with CBC-MAC (CCM/CCM*) mode for authenticated encryption
- 8, 16, 32, 64, 128-bit data sizes possible in CFB mode
- Optional (parameter) Galois Counter mode (GCM) encryption and authentication

© 2017 Microchip Technology Inc.

38.3 Block Diagram

Figure 38-1. AES Block Diagram



38.4 Signal Description

Not applicable.

38.5 Product Dependencies

In order to use this AES module, other parts of the system must be configured correctly, as described below.

38.5.1 I/O Lines

Not applicable.

38.5.2 Power Management

The AES will continue to operate in any sleep mode, if it's source clock is running. The AES interrupts can be used to wake up the device from sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

AES is clocked only on the following conditions:

- Whenever there is an APB access for any read and write operation to the AES registers.
- When the AES is enabled & encryption/decryption is on.

38.5.3 Clocks

The AES bus clock (CLK_AES_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_AES_APB can be found in *Peripheral Clock Masking*. The module is fully clocked by CLK_AES_APB.

Related Links

Peripheral Clock Masking

38.5.4 DMA

The AES has two DMA request lines; one for input data, and one for output data. They are both connected to the DMA Controller (DMAC). These DMA request triggers will be acknowledged by the DMAC ACK signals. Using the AES DMA requests requires the DMA Controller to be configured first. Refer to the device DMA documentation.

38.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the AES interrupt requires the interrupt controller to be configured first. Refer to the Processor and Architecture chapter for details.

All the AES interrupts are synchronous wake-up sources. See *Sleep Mode Controller* for details.

Related Links

Sleep Mode Controller

38.5.6 Events

Not applicable.

38.5.7 Debug Operation

When the CPU is halted in debug mode, the AES module continues normal operation. If the AES module is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging. The AES module can be forced to halt operation during debugging.

38.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the

following register:

- Interrupt Flag Register (INTFLAG)

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to *PAC - Peripheral Access Controller* chapter for details.

Related Links

PAC - Peripheral Access Controller

38.5.9 Analog Connections

Not applicable.

38.6 Functional Description

38.6.1 Principle of Operation

The following is a high level description of the algorithm. These are the steps:

- KeyExpansion: Round keys are derived from the cipher key using Rijndael's key schedule.
- InitialRound:
  - AddRoundKey: Each byte of the state is combined with the round key using bitwise XOR.
- Rounds:
  - SubBytes: A non-linear substitution step where each byte is replaced with another according to a lookup table.
  - ShiftRows: A transposition step where each row of the state is shifted cyclically a certain number of steps.
  - MixColumns: A mixing operation which operates on the columns of the state, combining the four bytes in each column.
  - AddRoundKey
- Final Round (no MixColumns):
  - SubBytes
  - ShiftRows
  - AddRoundKey

The relationship between the module's clock frequency and throughput (in bytes per second) is given by:

Clock Frequency = (Throughput/2) x (Nr+1) for 2 byte parallel processing

Clock Frequency = (Throughput/4) x (Nr+1) for 4 byte parallel processing

where Nr is the number of rounds, depending on the key length.

38.6.2 Basic Operation

38.6.2.1 Initialization

The following register is enable-protected:

- Control A (CTRLA)

Enable-protection is denoted by the Enable-Protected property in the register description.

38.6.2.2 Enabling, Disabling, and Resetting

The AES module is enabled by writing a one to the Enable bit in the Control A register (CTRLA.ENABLE). The module is disabled by writing a zero to CTRLA.ENABLE. The module is reset by writing a one to the Software Reset bit in the Control A register (CTRLA.SWRST).

38.6.2.3 Basic Programming

The CIPHER bit in the Control A Register (CTRLA.CIPHER) allows selection between the encryption and the decryption processes. The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. The Key Size (128/192/256) can be programmed in the KEYSIZE field in the Control A Register (CTRLA.KEYSIZE). This 128-bit/192-bit/256-bit key is defined in the Key Word Registers (KEYWORD). By setting the XORKEY bit of CTRLA register, keyword can be updated with the resulting XOR value of user keyword and previous keyword content.

The input data for processing is written to a data buffer consisting of four 32-bit registers through the Data register address. The data buffer register (note that input and output data shares the same data buffer register) that is written to when the next write is performed is indicated by the Data Pointer in the Data Buffer Pointer (DATABUFPTR) register. This field is incremented by one or wrapped by hardware when a write to the DATA register address is performed. This field can also be programmed, allowing the user direct control over which input buffer register to write to. Note that when AES module is in the CFB operation mode with the data segment size less than 128 bits, the input data must be written to the first (DATABUFPTR = 0) and/or second (DATABUFPTR = 1) input buffer registers (see Table 38-1).

The input to the encryption processes of the CBC, CFB and OFB modes includes, in addition to the plaintext, a 128-bit data block called the Initialization Vector (IV), which must be set in the Initialization Vector Registers (INTVECT). Additionally, the GCM mode 128-bit authentication data needs to be programmed. The Initialization Vector is used in the initial step in the encryption of a message and in the corresponding decryption of the message. The Initialization Vector Registers are also used by the Counter mode to set the counter value.

It is necessary to notify AES module whenever the next data block it is going to process is the beginning of a new message. This is done by writing a one to the New Message bit in the Control B register (CTRLB.NEWMSG).

The AES modes of operation are selected by setting the AESMODE field in the Control A Register (CTRLA.AESMODE). In Cipher Feedback Mode (CFB), five data sizes are possible (8, 16, 32, 64 or 128 bits), configurable by means of the CFBS field in the Control A Register (CTRLA.CFBS). In Counter mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 megabyte of data. The data pre-processing, post-processing and data chaining for the concerned modes are automatically performed by the module.

When data processing has completed, the Encryption Complete bit in the Interrupt Flag register (INTFLAG.ENCCMP) is set by hardware (which triggers an interrupt request if the corresponding interrupt is enabled). The processed output data is read out through the Output Data register (DATA) address from the data buffer consisting of four 32-bit registers. The data buffer register that is read from when the next read is performed is indicated by the Data Pointer field in the Data Buffer Pointer register (DATABUFPTR). This field is incremented by one or wrapped by hardware when a read from the DATA register address is performed. This field can also be programmed, giving the user direct control over which output buffer register to read from. Note that when AES module is in the CFB operation mode with the data segment size less than 128 bits, the output data must be read from the first (DATABUFPTR = 0) and/or second (DATABUFPTR = 1) output buffer registers (see Table 38-1). The Encryption Complete bit (INTFLAG.ENCCMP) is cleared by hardware after the processed data has been read from the relevant output buffer registers.

Table 38-1. Relevant Input/Output Data Registers for Different Confidentiality Modes

Confidentiality Mode	Relevant Input / Output Data Registers
ECB	All
CBC	All
OFB	All
128-bit CFB	All
64-bit CFB	First and Second
32-bit CFB	First
16-bit CFB	First
8-bit CFB	First
CTR	All

38.6.2.4 Start Modes

The Start mode field in the Control A Register (CTRLA.STARTMODE) allows the selection of encryption start mode.

1. Manual Start Mode
   In the Manual Start Mode the sequence is as follows:
   1.1. Write the 128/192/256 bit key in the Key Register (KEYWORDx)
   1.2. Write the initialization vector or counter in the Initialization Vector Register (INTVECTx). The initialization vector concerns all modes except ECB
   1.3. Enable interrupts in Interrupt Enable Set Register (INTENSET), depending on whether an interrupt is required or not at the end of processing.
   1.4. Write the data to be encrypted or decrypted in the Data Registers (DATA).
   1.5. Set the START bit in Control B Register (CTRLB.START) to begin the encryption or the decryption process.
   1.6. When the processing completes, the Encryption Complete bit in the Interrupt Flag Register (INTFLAG.ENCCMP) raises. If Encryption Complete interrupt has been enabled, the interrupt line of the AES is activated.
   1.7. When the software reads one of the Output Data Registers (DATA), INTFLAG.ENCCMP bit is automatically cleared.
2. Auto start Mode
   The Auto Start Mode is similar to the manual one, except in this mode, as soon as the correct number of input data registers is written, processing is automatically started without setting the START bit in the Control B Register. DMA operation uses this mode.
3. Last Output Data Mode (LOD)
   This mode is used to generate message authentication code (MAC) on data in CCM mode of operation. The CCM mode combines counter mode for encryption and CBC-MAC generation for authentication.

When LOD is disabled in CCM mode then counter mode of encryption is performed on the input data block.

When LOD is enabled in CCM mode then CBC-MAC generation is performed. Zero block is used as the initialization vector by the hardware. Also software read from the Output Data Register (DATA) is not required to clear the ENCCMP flag. The ENCCMP flag is automatically cleared by writing into the Input

Data Register (DATA). This allows retrieval of only the last data in several encryption/decryption processes. No output data register reads are necessary between each block of encryption/decryption process.

Note that assembling message depending on the security level identifier in CCM* has to be done in software.

38.6.2.5 Computation of last Nk words of expanded key

The AES algorithm takes the cryptographic key provided by the user and performs a Key Expansion routine to generate an expanded key. The expanded key contains a total of 4(Nr + 1) 32-bit words, where the first Nk (4/6/8 for a 128-/192-/256-bit key) words are the user-provided key. For data encryption, the expanded key is used in the forward direction, i.e., the first four words are used in the initial round of data processing, the second four words in the first round, the third four words in the second round, and so on. On the other hand, for data decryption, the expanded key is used in the reverse direction, i.e.,the last four words are used in the initial round of data processing, the last second four words in the first round, the last third four words in the second round, and so on.

To reduce gate count, the AES module does not generate and store the entire expanded key prior to data processing. Instead, it computes on-the-fly the round key (four 32-bit words) required for the current round of data processing. In general, the round key for the current round of data processing can be computed from the Nk words of the expanded key generated in the previous rounds. When AES module is operating in the encryption mode, the round key for the initial round of data processing is simply the user-provided key written to the KEY registers. On the other hand, when AES module is operating in the decryption mode, the round key for the initial round of data processing is the last four words of the expanded key, which is not available unless AES module has performed at least one encryption process prior to operating in the decryption mode.

In general, the last Nk words of the expanded key must be available before decryption can start. If desired, AES module can be instructed to compute the last Nk words of the expanded key in advance by writing a one to the Key Generate (KEYGEN) bit in the CTRLA register (CTRLA.KEYGEN). The computation takes Nr clock cycles. Alternatively, the last Nk words of the expanded key can be automatically computed by AES module when a decryption process is initiated if they have not been computed in advance or have become invalid. Note that this will introduce a latency of Nr clock cycles to the first decryption process.

38.6.2.6 Hardware Countermeasures against Differential Power Analysis Attacks

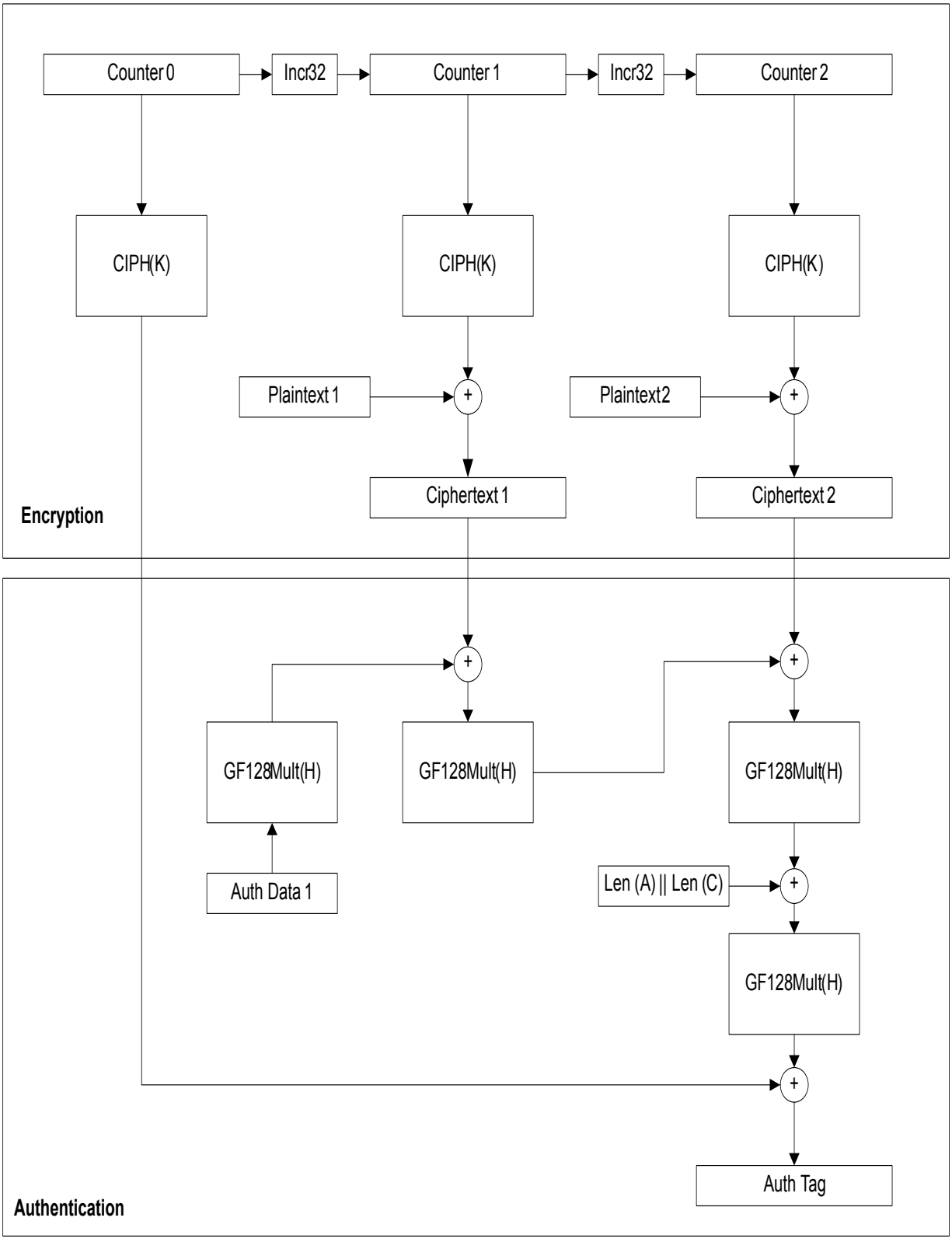
The AES module features four types of hardware countermeasures that are useful for protecting data against differential power analysis attacks:

- Type 1: Randomly add one cycle to data processing
- Type 2: Randomly add one cycle to data processing (other version)
- Type 3: Add a random number of clock cycles to data processing, subject to a maximum of 11/13/15 clock cycles for key sizes of 128/192/256 bits
- Type 4: Add random spurious power consumption during data processing

By default, all countermeasures are enabled. One or more of the countermeasures can be disabled by programming the Countermeasure Type field in the Control A (CTRLA.CTYPE) register. The countermeasures use random numbers generated by a deterministic random number generator embedded in AES module. The seed for the random number generator is written to the RANDSEED register. Note also that a new seed must be written after a change in the keysize. Note that enabling countermeasures reduces AES module's throughput. In short, the throughput is highest with all the countermeasures disabled. On the other hand, with all of the countermeasures enabled, the best protection is achieved but the throughput is worst.

38.6.3 Galois Counter Mode (GCM)

GCM is comprised of the AES engine in CTR mode along with a universal hash function (GHASH engine) that is defined over a binary Galois field to produce a message authentication tag. The GHASH engine processes data packets after the AES operation. GCM provides assurance of the confidentiality of data through the AES Counter mode of operation for encryption. Authenticity of the confidential data is assured through the GHASH engine. Refer to the NIST Special Publication 800-38D Recommendation for more complete information.


Counter 0
Incr32
Counter 1
Incr32
Counter 2
CIPH(K)
CIPH(K)
CIPH(K)
Plaintext 1
Plaintext2
Ciphertext 1
Ciphertext 2
Encryption
GF128Mult(H)
GF128Mult(H)
GF128Mult(H)
Auth Data 1
Len (A) || Len (C)
GF128Mult(H)
Auth Tag
Authentication

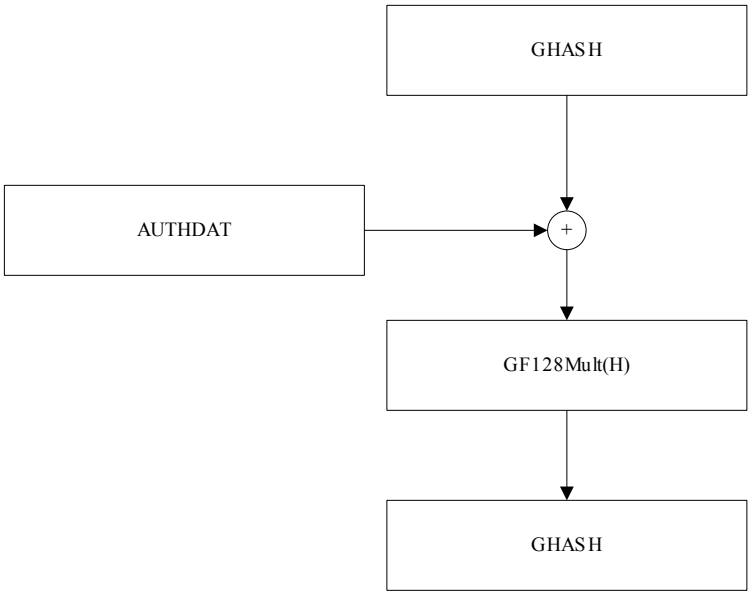
38.6.3.1 GCM Operation

Hashkey Generation

- Configure CTRLA register as follows:
  1.1. CTRLA.STARTMODE as Manual (Auto for DMAC)
  1.2. CTRLA.CIPHER as Encryption
  1.3. CTRLA.KEYSIZE as per the key used
  1.4. CTRLA.AESMODE as ECB
  1.5. CTRLA.CTYPE as per the countermeasures required.
- Set CTRLA.ENABLE
- Write zero to CIPLEN reg.
- Write the key in KEYWORDx register
- Write the zeros to DATA reg
- Set CTRLB.Start.
- Wait for INTFLAG.ENCCMP to be set
- AES Hardware generates Hash Subkey in HASHKEYx register.

Authentication Header Processing

- Configure CTRLA register as follows:
  1.1. CTRLA.STARTMODE as Manual
  1.2. CTRLA.CIPHER as Encryption
  1.3. CTRLA.KEYSIZE as per the key used
  1.4. CTRLA.AESMODE as GCM
  1.5. CTRLA.CTYPE as per the countermeasures required.
- Set CTRLA.ENABLE
- Write the key in KEYWORD register
- Set CTRLB.GFMUL
- Write the Authdata to DATA reg
- Set CTRLB.START as1
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates output in GHASH register
- Continue steps 4 to 7 for remaining Authentication Header.
  Note: If the Auth data is less than 128 bit, it has to be padded with zero to make it 128 bit aligned.



Plain text Processing

- Set CTRLB.NEWMSG for the new set of plain text processing.
- Load CIPLEN reg.
- Load (J0+1) in INTVECTx register.
- As described in NIST documentation J 0 = IV || 0 31 || 1 when len(IV)=96 and J0 =$GHASH_H$ (IV || 0 s+64 || [len(IV)] 64) (s is the minimum numer of zeroes that should be padded with the Initialization Vector to make it a multiple of 128) if len(IV) != 96.
- Load plain text in DATA register.
- Set CTRLB.START as 1.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register.
- Intermediate GHASH is stored in GHASHx register and Cipher Text available in DATA register.
- Continue 3 to 6 till the input of plain text to get the cipher text and the Hash keys.
- At the last input, set CTRLB.EOM.
- Write last indata to DATA reg.
- Set CTRLB.START as 1.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register and final Hash key in GHASHx register.
- Load [LEN(A)]64||[LEN(C)]64 in DATA register and set CTRLB.GFMUL and CTRLB.START as 1.
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates final GHASH value in GHASHx register.

Plain text processing with DMAC

- Set CTRLB.NEWMSG for the new set of plain text processing.
- Load CIPLEN reg.
- Load (J0+1) in INTVECTx register.
- Load plain text in DATA register.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register.

- Intermediate GHASH is stored in GHASHx register and Cipher Text available in DATA register.
- Continue 3 to 5 till the input of plain text to get the cipher text and the Hash keys.
- At the last input, set CTRLB.EOM.
- Write last indata to DATA reg.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register and final Hash key in GHASHx register.
- Load [LEN(A)]64||[LEN(C)]64 in DATA register and set CTRLB.GFMUL and CTRLB.START as 1.
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates final GHASH value in GHASHx register.

Tag Generation

- Configure CTRLA
  1.1. Set CTRLA.ENABLE to 0
  1.2. Set CTRLA.AESMODE as CTR
  1.3. Set CTRLA.ENABLE to 1
- Load J0 value to INITVECTV reg.
- Load GHASH value to DATA reg.
- Set CTRLB.NEWMSG and CTRLB.START to start the Counter mode operation.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates the GCM Tag output in DATA register.

38.6.4 Synchronization

Not applicable.

38.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	CFBS[2:0]			AESMODE[2:0]			ENABLE	SWRST
0x01		15:8		XORKEY	KEYGEN	LOD	STARTMODE	CIPHER	KEYSIZE[1:0]	
0x02		23:16					CTYPE[3:0]			
0x03		31:24								
0x04	CTRLB	7:0					GFMUL	EOM	NEWMSG	START
0x05	INTENCLR	7:0							GFMCMP	ENCCMP
0x06	INTENSET	7:0							GFMCMP	ENCCMP
0x07	INTFLAG	7:0							GFMCMP	ENCCMP
0x08	DATABUFPTR	7:0							INDATAPTR[1:0]	
0x09	DBGCTRL	7:0								DBGRUN
0x0A ... 0x0B	Reserved									
0C	KEYWORD0	7:0	KEYWORD[7:0]							
0D		15:8	KEYWORD[15:8]							
0E		23:16	KEYWORD[23:16]							
0F		31:24	KEYWORD[31:24]							
10	KEYWORD1	7:0	KEYWORD[7:0]							
11		15:8	KEYWORD[15:8]							
12		23:16	KEYWORD[23:16]							
13		31:24	KEYWORD[31:24]							
14	KEYWORD2	7:0	KEYWORD[7:0]							
15		15:8	KEYWORD[15:8]							
16		23:16	KEYWORD[23:16]							
17		31:24	KEYWORD[31:24]							
18	KEYWORD3	7:0	KEYWORD[7:0]							
19		15:8	KEYWORD[15:8]							
1A		23:16	KEYWORD[23:16]							
1B		31:24	KEYWORD[31:24]							
1C	KEYWORD4	7:0	KEYWORD[7:0]							
1D		15:8	KEYWORD[15:8]							
1E		23:16	KEYWORD[23:16]							
1F		31:24	KEYWORD[31:24]							
20	KEYWORD5	7:0	KEYWORD[7:0]							
21		15:8	KEYWORD[15:8]							
22		23:16	KEYWORD[23:16]							
23		31:24	KEYWORD[31:24]							
24	KEYWORD6	7:0	KEYWORD[7:0]							
25		15:8	KEYWORD[15:8]							
26		23:16	KEYWORD[23:16]							
27		31:24	KEYWORD[31:24]							
28	KEYWORD7	7:0	KEYWORD[7:0]							
29		15:8	KEYWORD[15:8]							
2A		23:16	KEYWORD[23:16]							

Offset	Name	Bit Pos.									
2B		31:24	KEYWORD[31:24]								
0x2C ... 0x37	Reserved										
0x38	DATA	7:0	DATA[7:0]								
0x39		15:8	DATA[15:8]								
0x3A		23:16	DATA[23:16]								
0x3B		31:24	DATA[31:24]								
3C	INTVECTx0	7:0	INTVECT[7:0]								
3D		15:8	INTVECT[15:8]								
3E		23:16	INTVECT[23:16]								
3F		31:24	INTVECT[31:24]								
40	INTVECTx1	7:0	INTVECT[7:0]								
41		15:8	INTVECT[15:8]								
42		23:16	INTVECT[23:16]								
43		31:24	INTVECT[31:24]								
44	INTVECTx2	7:0	INTVECT[7:0]								
45		15:8	INTVECT[15:8]								
46		23:16	INTVECT[23:16]								
47		31:24	INTVECT[31:24]								
48	INTVECTx3	7:0	INTVECT[7:0]								
49		15:8	INTVECT[15:8]								
4A		23:16	INTVECT[23:16]								
4B		31:24	INTVECT[31:24]								
0x4C ... 0x5B	Reserved										
0x5C	HASHKEYx0	7:0	HASHKEY[7:0]								
0x5D		15:8	HASHKEY[15:8]								
0x5E		23:16	HASHKEY[23:16]								
0x5F		31:24	HASHKEY[31:24]								
0x60	HASHKEYx1	7:0	HASHKEY[7:0]								
0x61		15:8	HASHKEY[15:8]								
0x62		23:16	HASHKEY[23:16]								
0x63		31:24	HASHKEY[31:24]								
0x64	HASHKEYx2	7:0	HASHKEY[7:0]								
0x65		15:8	HASHKEY[15:8]								
0x66		23:16	HASHKEY[23:16]								
0x67		31:24	HASHKEY[31:24]								
0x68	HASHKEYx3	7:0	HASHKEY[7:0]								
0x69		15:8	HASHKEY[15:8]								
0x6A		23:16	HASHKEY[23:16]								
0x6B		31:24	HASHKEY[31:24]								
0x6C	GHASHx0	7:0	GHASH[7:0]								
0x6D		15:8	GHASH[15:8]								
0x6E		23:16	GHASH[23:16]								
0x6F		31:24	GHASH[31:24]								

Offset	Name	Bit Pos.								
0x70	GHASHx1	7:0	GHASH[7:0]							
0x71		15:8	GHASH[15:8]							
0x72		23:16	GHASH[23:16]							
0x73		31:24	GHASH[31:24]							
0x74	GHASHx2	7:0	GHASH[7:0]							
0x75		15:8	GHASH[15:8]							
0x76		23:16	GHASH[23:16]							
0x77		31:24	GHASH[31:24]							
0x78	GHASHx3	7:0	GHASH[7:0]							
0x79		15:8	GHASH[15:8]							
0x7A		23:16	GHASH[23:16]							
0x7B		31:24	GHASH[31:24]							
0x7C ... 0x7F	Reserved									
80	CIPLEN	7:0	CIPLEN[7:0]							
81		15:8	CIPLEN[15:8]							
82		23:16	CIPLEN[23:16]							
83		31:24	CIPLEN[31:24]							
0x84	RANDSEED	7:0	RANDSEED[7:0]							
0x85		15:8	RANDSEED[15:8]							
0x86		23:16	RANDSEED[23:16]							
0x87		31:24	RANDSEED[31:24]							

38.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

38.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					CTYPE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		XORKEY	KEYGEN	LOD	STARTMODE	CIPHER	KEYSIZE[1:0]	
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CFBS[2:0]			AESMODE[2:0]			ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – CTYPE[3:0]: Counter Measure Type

Value	Name	Description
XXX0	CTYPE1 disabled	Countermeasure1 disabled
XXX1	CTYPE1 enabled	Countermeasure1 enabled
XX0X	CTYPE2 disabled	Countermeasure2 disabled
XX1X	CTYPE2 enabled	Countermeasure2 enabled
X0XX	CTYPE3 disabled	Countermeasure3 disabled
X1XX	CTYPE3 enabled	Countermeasure3 enabled
0XXX	CTYPE4 disabled	Countermeasure4 disabled
1XXX	CTYPE4 enabled	Countermeasure4 enabled

Bit 14 – XORKEY: XOR Key Operation

Value	Description
0	No effect
1	The user keyword gets XORed with the previous keyword register content.

Bit 13 – KEYGEN: Last Key Generation

Value	Description
0	No effect
1	Start Computation of the last NK words of the expanded key

Bit 12 – LOD: Last Output Data Mode

Value	Description
0	No effect
1	Start encryption in Last Output Data mode

Bit 11 – STARTMODE: Start Mode Select

Value	Name	Description
0	Manual Mode	Start Encryption / Decryption in Manual mode
1	Auto Mode	Start Encryption / Decryption in Auto mode

Bit 10 – CIPHER: Cipher Mode Select

Value	Description
0	Decryption
1	Encryption

Bits 9:8 – KEYSIZE[1:0]: Encryption Key Size

Value	Name	Description
0	128-bit Key	128-bit Key for Encryption / Decryption
1	192-bit Key	192-bit Key for Encryption / Decryption
2	256-bit Key	256-bit Key for Encryption / Decryption
3	Reserved	Reserved

Bits 7:5 – CFBS[2:0]: Cipher Feedback Block Size

Value	Name	Description
0	128-bit data block	128-bit Input data block for Encryption/Decryption in Cipher Feedback mode
1	64-bit data block	64-bit Input data block for Encryption/Decryption in Cipher Feedback mode
2	32-bit data block	32-bit Input data block for Encryption/Decryption in Cipher Feedback mode
3	16-bit data block	16-bit Input data block for Encryption/Decryption in Cipher Feedback mode
4	8-bit data block	8-bit Input data block for Encryption/Decryption in Cipher Feedback mode
5-7	Reserved	Reserved

Bits 4:2 – AESMODE[2:0]: AES Modes of Operation

Value	Name	Description
0	ECB	Electronic code book mode
1	CBC	Cipher block chaining mode
2	OFB	Output feedback mode
3	CFB	Cipher feedback mode
4	Counter	Counter mode
5	CCM	CCM mode
6	GCM	Galois counter mode
7	Reserved	Reserved

Bit 1 – ENABLE: Enable

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the AES module to their initial state, and the module will be disabled.

Writing a '1' to `SWRST` will always take precedence, meaning that all other writes in the same write operation will be discarded.

Value	Description
0	There is no reset operation ongoing
1	The reset operation is ongoing

38.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					GFMUL	EOM	NEWMSG	START
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – GFMUL: GF Multiplication
This bit is applicable only to GCM mode.

Value	Description
0	No action
1	Setting this bit calculates GF multiplication with data buffer content and hashkey register content.

Bit 2 – EOM: End of Message
This bit is applicable only to GCM mode.

Value	Description
0	No action
1	Setting this bit generates final GHASH value for the message.

Bit 1 – NEWMSG: New Message
This bit is used in cipher block chaining (CBC), cipher feedback (CFB) and output feedback (OFB), counter (CTR) modes to indicate the hardware to use Initialization vector for encrypting the first block of message.

Value	Description
0	No action
1	Setting this bit indicates start of new message to the module.

Bit 0 – START: Start Encryption/Decryption

Value	Description
0	No action
1	Start encryption / decryption in manual mode.

38.8.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							GFMCMP	ENCCMP
Access							R/W	R/W
Reset							0	0

Bit 1 – GFMCMP: GF Multiplication Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the GF Multiplication Complete Interrupt Enable bit, which disables the GF Multiplication Complete interrupt.

Value	Description
0	The GF Multiplication Complete interrupt is disabled.
1	The GF Multiplication Complete interrupt is enabled.

Bit 0 – ENCCMP: Encryption Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Encryption Complete Interrupt Enable bit, which disables the Encryption Complete interrupt.

Value	Description
0	The Encryption Complete interrupt is disabled.
1	The Encryption Complete interrupt is enabled.

38.8.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x06
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							GFMCMP	ENCCMP
Access							R/W	R/W
Reset							0	0

Bit 1 – GFMCMP: GF Multiplication Complete Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the GF Multiplication Complete Interrupt Enable bit, which enables the GF Multiplication Complete interrupt.

Value	Description
0	The GF Multiplication Complete interrupt is disabled.
1	The GF Multiplication Complete interrupt is enabled.

Bit 0 – ENCCMP: Encryption Complete Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Encryption Complete Interrupt Enable bit, which enables the Encryption Complete interrupt.

Value	Description
0	The Encryption Complete interrupt is disabled.
1	The Encryption Complete interrupt is enabled.

38.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x07
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
							GFMCMP	ENCCMP
Access							R/W	R/W
Reset							0	0

Bit 1 – GFMCMP: GF Multiplication Complete
This flag is cleared by writing a '1' to it.

This flag is set when GHASH value is available on the Galois Hash Registers (GHASHx) in GCM mode.

Writing a '0' to this bit has no effect.

This flag is also automatically cleared in the following cases.

1. Manual encryption/decryption occurs (START in CTRLB register).
2. Reading from the GHASHx register.

Bit 0 – ENCCMP: Encryption Complete
This flag is cleared by writing a '1' to it.

This flag is set when encryption/decryption is complete and valid data is available on the Data Register.

Writing a '0' to this bit has no effect.

This flag is also automatically cleared in the following cases:

1. Manual encryption/decryption occurs (START in CTRLA register). (This feature is needed only if we do not support double buffering of DATA registers).
2. Reading from the data register (DATAx) when LOD = 0.
3. Writing into the data register (DATAx) when LOD = 1.
4. Reading from the Hash Key register (HASHKEYx).

38.8.6 Data Buffer Pointer

Name: DATABUFPTR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							INDATAPTR[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – INDATAPTR[1:0]: Input Data Pointer
Writing to this field changes the value of the input data pointer, which determines which of the four data registers is written to/read from when the next write/read to the DATA register address is performed.

38.8.7 Debug

Name: DBGCTRL
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								W
Reset								0

Bit 0 – DBGRUN: Debug Run
Writing a '0' to this bit causes the AES to halt during debug mode.
Writing a '1' to this bit allows the AES to continue normal operation during debug mode. This bit can only be changed while the AES is disabled.

38.8.8 Keyword

Name: KEYWORD
Offset: 0x0C + n*0x04 [n=0..7]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	KEYWORD[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	KEYWORD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	KEYWORD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	KEYWORD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEYWORD[31:0]: Key Word Value

The four/six/eight 32-bit Key Word registers set the 128-bit/192-bit/256-bit cryptographic key used for encryption/decryption. `KEYWORD0.KEYWORD` corresponds to the first word of the key and `KEYWORD3/KEYWORD5/KEYWORD7.KEYWORD` to the last one.

Note: By setting the XORKEY bit of CTRLA register, keyword will update with the resulting XOR value of user keyword and previous keyword content.

38.8.9 Data

Name: DATA
Offset: 0x38
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data Value

A write to or read from this register corresponds to a write to or read from one of the four data registers. The four 32-bit Data registers set the 128-bit data block used for encryption/decryption. The data register that is written to or read from is given by the `DATABUFPTR.DATPTR` field.

Note: Both input and output shares the same data buffer. Reading DATA register will return 0's when AES is performing encryption or decryption operation.

38.8.10 Initialization Vector x Register

Name: INTVECTx
Offset: 0x3C + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	INTVECT[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	INTVECT[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INTVECT[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	INTVECT[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – INTVECT[31:0]: Initialization Vector Value

The four 32-bit Initialization Vector registers `INTVECTx` set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input. `INTVECT0.INTVECT` corresponds to the first word of the Initialization Vector, `INTVECT3.INTVECT` to the last one. These registers are write-only to prevent the Initialization Vector from being read by another application. For CBC, OFB, and CFB modes, the Initialization Vector corresponds to the initialization vector. For CTR mode, it corresponds to the counter value.

38.8.11 Hash Key x (GCM mode only)

Name: HASHKEYx
Offset: 0x5C + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-protection

Bit	31	30	29	28	27	26	25	24
	HASHKEY[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HASHKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HASHKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HASHKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – HASHKEY[31:0]: Hash Key Value

The four 32-bit `HASHKEYx` registers contain the 128-bit Hash Key value computed from the AES KEY. The Hash Key value can also be programmed offering single GF128 multiplication possibilities.

38.8.12 Galois Hash x (GCM mode only)

Name: GHASHx
Offset: 0x6C + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	GHASH[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GHASH[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GHASH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GHASH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GHASH[31:0]: Galois Hash Value

The four 32-bit Hash Word registers GHASHxcontain the GHASH value after GF128 multiplication in GCM mode. Writing a new key to KEYWORDx registers causes GHASHx to be initialized with zeroes. These registers can also be programmed.

38.8.13 Galois Hash x (GCM mode only)

Name: CIPLEN
Offset: 0X80
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CIPLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CIPLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CIPLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CIPLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CIPLEN[31:0]: Cipher Length
This register contains the length in bytes of the Cipher text that is to be processed. This is programmed by the user in GCM mode for Tag generation.

38.8.14 Random Seed

Name: RANDSEED
Offset: 0x84
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RANDSEED[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RANDSEED[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RANDSEED[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RANDSEED[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RANDSEED[31:0]: Random Seed

A write to this register corresponds to loading a new seed into the Random number generator.

39. USB – Universal Serial Bus

39.1 Overview

The Universal Serial Bus interface (USB) module complies with the Universal Serial Bus (USB) 2.1 specification supporting device modes.

The USB device mode supports 8 endpoint addresses. All endpoint addresses have one input and one output endpoint, for a total of 16 endpoints. Each endpoint is fully configurable in any of the four transfer types: control, interrupt, bulk or isochronous. The maximum data payload size is selectable up to 1023 bytes.

Internal SRAM is used to keep the configuration and data buffer for each endpoint. The memory locations used for the endpoint configurations and data buffers is fully configurable. The amount of memory allocated is dynamic according to the number of endpoints in use, and the configuration of these. The USB module has a built-in Direct Memory Access (DMA) and will read/write data from/to the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

To maximize throughput, an endpoint can be configured for ping-pong operation. When this is done the input and output endpoint with the same address are used in the same direction. The CPU or DMA Controller can then read/write one data buffer while the USB module writes/reads from the other buffer. This gives double buffered communication.

Multi-packet transfer enables a data payload exceeding the maximum packet size of an endpoint to be transferred as multiple packets without any software intervention. This reduces the number of interrupts and software intervention needed for USB transfers.

For low power operation the USB module can put the microcontroller in any sleep mode when the USB bus is idle and a suspend condition is given. Upon bus resume, the USB module can wake the microcontroller from any sleep mode.

39.2 Features

- Compatible with the USB 2.1 specification
- USB Device mode
- Supports full (12Mbit/s) and low (1.5Mbit/s) speed communication
- Supports Link Power Management (LPM-L1) protocol
- On-chip transceivers with built-in pull-ups and pull-downs
- On-Chip USB serial resistors
- 1kHz SOF clock available on external pin
- Device mode
  - Supports 8 IN endpoints and 8 OUT endpoints
  - No endpoint size limitations
  - Built-in DMA with multi-packet and dual bank for all endpoints
  - Supports feedback endpoint
  - Supports crystal less clock

39.3 USB Block Diagram

39.4 Signal Description

Pin Name	Pin Description	Type
DM	Data -: Differential Data Line - Port	Input/Output
DP	Data +: Differential Data Line + Port	Input/Output
SOF 1kHZ	SOF Output	Output

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

39.5 Product Dependencies

In order to use this peripheral module, other parts of the system must be configured correctly, as described below.

39.5.1 I/O Lines

The USB pins may be multiplexed with the I/O lines Controller. The user must first configure the I/O Controller to assign the USB pins to their peripheral functions.

A 1kHz SOF clock is available on an external pin. The user must first configure the I/O Controller to assign the 1kHz SOF clock to the peripheral function.

39.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

39.5.3 Clocks

The USB bus clock (CLK_USB_AHB) can be enabled and disabled in the Power Manager, and the default state of CLK_USB_AHB can be found in the *Peripheral Clock Masking*.

A generic clock (GCLK_USB) is required to clock the USB. This clock must be configured and enabled in the Generic Clock Controller before using the USB. Refer to *GCLK - Generic Clock Controller* for further details.

This generic clock is asynchronous to the bus clock (CLK_USB_AHB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to *GCLK Synchronization* for further details.

The USB module requires a GCLK_USB of 48 MHz ± 0.25% clock for low speed and full speed operation. To follow the USB data rate at 12Mbit/s in full-speed mode, the CLK_USB_AHB clock should be at minimum 8MHz.

Clock recovery is achieved by a digital phase-locked loop in the USB module, which complies with the USB jitter specifications. If crystal-less operation is used in USB device mode, refer to *USB Clock Recovery Module*.

Related Links

GCLK - Generic Clock Controller

Synchronization

USB Clock Recovery Module

39.5.4 DMA

The USB has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

39.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

39.5.6 Events

Not applicable.

39.5.7 Debug Operation

When the CPU is halted in debug mode the USB peripheral continues normal operation. If the USB peripheral is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

39.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Device Interrupt Flag (INTFLAG) register
- Endpoint Interrupt Flag (EPINTFLAG) register

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

39.5.9 Analog Connections

Not applicable.

39.5.10 Calibration

The output drivers for the DP/DM USB line interface can be fine tuned with calibration values from production tests. The calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register (PADCAL) by software, before enabling the USB, to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.

For details on Pad Calibration, refer to Pad Calibration (PADCAL) register.

Related Links

NVM Software Calibration Area Mapping

39.6 Functional Description

39.6.1 USB General Operation

39.6.1.1 Initialization

After a hardware reset, the USB is in the idle state. In this state:

- The module is disabled. The USB Enable bit in the Control A register (CTRLA.ENABLE) is reset.
- The module clock is stopped in order to minimize power consumption.
- The is in suspend mode.
- The internal states and registers of the deviceare reset.

Before using the USB, the Pad Calibration register (PADCAL) must be loaded with production calibration values from the NVM Software Calibration Area.

The USB is enabled by writing a '1' to CTRLA.ENABLE. The USB is disabled by writing a '0' to CTRLA.ENABLE.

The USB is reset by writing a '1' to the Software Reset bit in CTRLA (CTRLA.SWRST). All registers in the USB will be reset to their initial state, and the USB will be disabled. Refer to the CTRLA register for details.

The user can configure pads and speed before enabling the USB by writing to the Speed Configuration field in the Control B register (CTRLB.SPDCONF). These values are taken into account once the USB has been enabled by writing a '1' to CTRLA.ENABLE.

After writing a '1' to CTRLA.ENABLE, the USB enters device mode.

The USB can be disabled at any time by writing a '0' to CTRLA.ENABLE.

Refer to USB Device Operations for the basic operation of the device mode.

Related Links

NVM Software Calibration Area Mapping

39.6.2 USB Device Operations

This section gives an overview of the USB module device operation during normal transactions. For more details on general USB and USB protocol, refer to the Universal Serial Bus specification revision 2.1.

39.6.2.1 Initialization

To attach the USB device to start the USB communications from the USB host, a zero should be written to the Detach bit in the Device Control B register (CTRLB.DETACH). To detach the device from the USB host, a one must be written to the CTRLB.DETACH.

After the device is attached, the host will request the USB device descriptor using the default device address zero. On successful transmission, it will send a USB reset. After that, it sends an address to be configured for the device. All further transactions will be directed to this device address. This address should be configured in the Device Address field in the Device Address register (DADD.DADD) and the Address Enable bit in DADD (DADD.ADDEN) should be written to one to accept communications directed to this address. DADD.ADDEN is automatically cleared on receiving a USB reset.

39.6.2.2 Endpoint Configuration

Endpoint data can be placed anywhere in the device RAM. The USB controller accesses these endpoints directly through the AHB master (built-in DMA) with the help of the endpoint descriptors. The base

address of the endpoint descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer also to the Endpoint Descriptor structure in Endpoint Descriptor Structure.

Before using an endpoint, the user should configure the direction and type of the endpoint in Type of Endpoint field in the Device Endpoint Configuration register (EPCFG.EPTYPE0/1). The endpoint descriptor registers should be initialized to known values before using the endpoint, so that the USB controller does not read random values from the RAM.

The Endpoint Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported to the host for that endpoint. The Address of Data Buffer register (ADDR) should be set to the data buffer used for endpoint transfers.

The RAM Access Interrupt bit in Device Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during IN data stage.

When an endpoint is disabled, the following registers are cleared for that endpoint:

- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

39.6.2.3 Multi-Packet Transfers

Multi-packet transfer enables a data payload exceeding the endpoint maximum transfer size to be transferred as multiple packets without software intervention. This reduces the number of interrupts and software intervention required to manage higher level USB transfers. Multi-packet transfer is identical to the IN and OUT transactions described below unless otherwise noted in this section.

The application software provides the size and address of the RAM buffer to be proceeded by the USB module for a specific endpoint, and the USB module will split the buffer in the required USB data transfers without any software intervention.

39.6.2.4 USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- Device Frame Number (FNUM) register
- Device Address (DADD) register
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)
- Endpoint Interrupt Summary (EPINTSMRY) register
- Upstream resume bit in the Control B register (CTRLB.UPRSM)

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

39.6.2.5 Start-of-Frame

When a Start-of-Frame (SOF) token is detected, the frame number from the token is stored in the Frame Number field in the Device Frame Number register (FNUM.FNUM), and the Start-of-Frame interrupt bit in the Device Interrupt Flag register (INTFLAG.SOF) is set. If there is a CRC or bit-stuff error, the Frame Number Error status flag (FNUM.FNCERR) in the FNUM register is set.

39.6.2.6 Management of SETUP Transactions

When a SETUP token is detected and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint is enabled in EPCFG. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed endpoint. If the EPCFG.EPTYPE0 is not set to control, the USB module returns to idle and waits for the next token packet.

When the EPCFG.EPTYPE0 matches, the USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor and waits for a DATA0 packet. If a PID error or any other PID than DATA0 is detected, the USB module returns to idle and waits for the next token packet.

When the data PID matches and if the Received Setup Complete interrupt bit in the Device Endpoint Interrupt Flag register (EPINTFLAG.RXSTP) is equal to zero, ignoring the Bank 0 Ready bit in the Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the endpoint's maximum data payload size as specified by the PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. Software must never report a endpoint size to the host that is greater than the value configured in PCKSIZE.SIZE. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If data is successfully received, an ACK handshake is returned to the host, and the number of received data bytes, excluding the CRC, is written to the Byte Count (PCKSIZE.BYTE_COUNT). If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE, no CRC data is written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data is written to the data buffer. If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally the EPSTATUS is updated. Data Toggle OUT bit (EPSTATUS.DTGLOUT), the Data Toggle IN bit (EPSTATUS.DTGLIN), the current bank bit (EPSTATUS.CURRBK) and the Bank Ready 0 bit (EPSTATUS.BK0RDY) are set. Bank Ready 1 bit (EPSTATUS.BK1RDY) and the Stall Bank 0/1 bit (EPSTATUS.STALLQR0/1) are cleared on receiving the SETUP request. The RXSTP bit is set and triggers an interrupt if the Received Setup Interrupt Enable bit is set in Endpoint Interrupt Enable Set/Clear register (EPINTENSET/CLR.RXSTP).

39.6.2.7 Management of OUT Transactions

When an OUT token is detected, and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

If the address matches, the USB module checks if the endpoint number received is enabled in the EPCFG of the addressed endpoint. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks the Endpoint Configuration register (EPCFG) of the addressed output endpoint. If the type of the endpoint (EPCFG.EPTYPE0) is not set to OUT, the USB module returns to idle and waits for the next token packet.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor, and waits for a DATA0 or DATA1 packet. If a PID error or any other PID than DATA0 or DATA1 is detected, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ0 in EPSTATUS is set, the incoming data is discarded. If the endpoint is not isochronous, a STALL handshake is returned to the host and the Transmit Stall Bank 0 interrupt bit in EPINTFLAG (EPINTFLAG.STALL0) is set.

For isochronous endpoints, data from both a DATA0 and DATA1 packet will be accepted. For other endpoint types the PID is checked against EPSTATUS.DTGLOUT. If a PID mismatch occurs, the incoming data is discarded, and an ACK handshake is returned to the host.

If EPSTATUS.BK0RDY is set, the incoming data is discarded, the bit Transmit Fail 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRFAIL0) and the status bit STATUS_BK.ERRORFLOW are set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the maximum data payload specified as PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If the endpoint is isochronous and a bit-stuff or CRC error in the incoming data, the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. Finally the EPINTFLAG.TRFAIL0 and CRC Error bit in the Device Bank Status register (STATUS_BK.CRCERR) is set for the addressed endpoint.

If data was successfully received, an ACK handshake is returned to the host if the endpoint is not isochronous, and the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE no CRC data bytes are written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data byte is written to the data buffer If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally in EPSTATUS for the addressed output endpoint, EPSTATUS.BK0RDY is set and EPSTATUS.DTGLOUT is toggled if the endpoint is not isochronous. The flag Transmit Complete 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRCPT0) is set for the addressed endpoint.

39.6.2.8 Multi-Packet Transfers for OUT Endpoint

The number of data bytes received is stored in endpoint PCKSIZE.BYTE_COUNT as for normal operation. Since PCKSIZE.BYTE_COUNT is updated after each transaction, it must be set to zero when setting up a new transfer. The total number of bytes to be received must be written to PCKSIZE.MULTI_PACKET_SIZE. This value must be a multiple of PCKSIZE.SIZE, otherwise excess data may be written to SRAM locations used by other parts of the application.

EPSTATUS.DTGLOUT management for non-isochronous packets and EPINTFLAG.BK1RDY/BK0RDY management are as for normal operation.

If a maximum payload size packet is received, PCKSIZE.BYTE_COUNT will be incremented by PCKSIZE.SIZE after the transaction has completed, and EPSTATUS.DTGLOUT will be toggled if the endpoint is not isochronous. If the updated PCKSIZE.BYTE_COUNT is equal to PCKSIZE.MULTI_PACKET_SIZE (i.e. the last transaction), EPSTATUS.BK1RDY/BK0RDY, and EPINTFLAG.TRCPT0/TRCPT1 will be set.

39.6.2.9 Management of IN Transactions

When an IN token is detected, and if the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint received is enabled in the EPCFG of the addressed endpoint and if not, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed input endpoint. If the EPCFG.EPTYPE1 is not set to IN, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ1 in EPSTATUS is set, and the endpoint is not isochronous, a STALL handshake is returned to the host and EPINTFLAG.STALL1 is set.

If EPSTATUS.BK1RDY is cleared, the flag EPINTFLAG.TRFAIL1 is set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor. The data pointed to by the Data Buffer Address (ADDR) is sent to the host in a DATA0 packet if the endpoint is isochronous. For non-isochronous endpoints a DATA0 or DATA1 packet is sent depending on the state of EPSTATUS.DTGLIN. When the number of data bytes specified in endpoint PCKSIZE.BYTE_COUNT is sent, the CRC is appended and sent to the host.

For isochronous endpoints, EPSTATUS.BK1RDY is cleared and EPINTFLAG.TRCPT1 is set.

For all non-isochronous endpoints the USB module waits for an ACK handshake from the host. If an ACK handshake is not received within 16 bit times, the USB module returns to idle and waits for the next token packet. If an ACK handshake is successfully received EPSTATUS.BK1RDY is cleared, EPINTFLAG.TRCPT1 is set and EPSTATUS.DTGLIN is toggled.

39.6.2.10 Multi-Packet Transfers for IN Endpoint

The total number of data bytes to be sent is written to PCKSIZE.BYTE_COUNT as for normal operation. The Multi-packet size register (PCKSIZE.MULTI_PACKET_SIZE) is used to store the number of bytes that are sent, and must be written to zero when setting up a new transfer.

When an IN token is received, PCKSIZE.BYTE_COUNT and PCKSIZE.MULTI_PACKET_SIZE are fetched. If PCKSIZE.BYTE_COUNT minus PCKSIZE.MULTI_PACKET_SIZE is less than the endpoint PCKSIZE.SIZE, endpoint BYTE_COUNT minus endpoint PCKSIZE.MULTI_PACKET_SIZE bytes are transmitted, otherwise PCKSIZE.SIZE number of bytes are transmitted. If endpoint PCKSIZE.BYTE_COUNT is a multiple of PCKSIZE.SIZE, the last packet sent will be zero-length if the AUTOZLP bit is set.

If a maximum payload size packet was sent (i.e. not the last transaction), MULTI_PACKET_SIZE will be incremented by the PCKSIZE.SIZE. If the endpoint is not isochronous the EPSTATUS.DTLGIN bit will be toggled when the transaction has completed. If a short packet was sent (i.e. the last transaction), MULTI_PACKET_SIZE is incremented by the data payload. EPSTATUS.BK0/1RDY will be cleared and EPINTFLAG.TRCPT0/1 will be set.

39.6.2.11 Ping-Pong Operation

When an endpoint is configured for ping-pong operation, it uses both the input and output data buffers (banks) for a given endpoint in a single direction. The direction is selected by enabling one of the IN or OUT direction in EPCFG.EPTYPE0/1 and configuring the opposite direction in EPCFG.EPTYPE1/0 as Dual Bank.

When ping-pong operation is enabled for an endpoint, the endpoint in the opposite direction must be configured as dual bank. The data buffer, data address pointer and byte counter from the enabled endpoint are used as Bank 0, while the matching registers from the disabled endpoint are used as Bank 1.

The Bank Select flag in EPSTATUS.CURBK indicates which bank data will be used in the next transaction, and is updated after each transaction. According to EPSTATUS.CURBK, EPINTFLAG.TRCPT0 or EPINTFLAG.TRFAIL0 or EPINTFLAG.TRCPT1 or EPINTFLAG.TRFAIL1 in EPINTFLAG and Data Buffer 0/1 ready (EPSTATUS.BK0RDY and EPSTATUS.BK1RDY) are set. The EPSTATUS.DTGLOUT and EPSTATUS.DTGLIN are updated for the enabled endpoint direction only.

39.6.2.12 Feedback Operation

Feedback endpoints are endpoints with same the address but in different directions. This is usually used in explicit feedback mechanism in USB Audio, where a feedback endpoint is associated to one or more isochronous data endpoints to which it provides feedback service. The feedback endpoint always has the opposite direction from the data endpoint.

The feedback endpoint always has the opposite direction from the data endpoint(s). The feedback endpoint has the same endpoint number as the first (lower) data endpoint. A feedback endpoint can be created by configuring an endpoint with different endpoint size (PCKSIZE.SIZE) and different endpoint type (EPCFG.EPTYPE0/1) for the IN and OUT direction.

Example Configuration for Feedback Operation:

- Endpoint n / IN: EPCFG.EPTYPE1 = Interrupt IN, PCKSIZE.SIZE = 64.
- Endpoint n / OUT: EPCFG.EPTYPE0= Isochronous OUT, PCKSIZE.SIZE = 512.

39.6.2.13 Suspend State and Pad Behavior

The following figure, Pad Behavior, illustrates the behavior of the USB pad in device mode.

In Idle state, the pad is in low power consumption mode.

In Active state, the pad is active.

The following figure, Pad Events, illustrates the pad events leading to a PAD state change.

The Suspend Interrupt bit in the Device Interrupt Flag register (INTFLAG.SUSPEND) is set when a USB Suspend state has been detected on the USB bus. The USB pad is then automatically put in the Idle state. The detection of a non-idle state sets the Wake Up Interrupt bit in INTFLAG(INTFLAG.WAKEUP) and wakes the USB pad.

The pad goes to the Idle state if the USB module is disabled or if CTRLB.DETACH is written to one. It returns to the Active state when CTRLA.ENABLE is written to one and CTRLB.DETACH is written to zero.

39.6.2.14 Remote Wakeup

The remote wakeup request (also known as upstream resume) is the only request the device may send on its own initiative. This should be preceded by a DEVICE_REMOTE_WAKEUP request from the host.

First, the USB must have detected a “Suspend” state on the bus, i.e. the remote wakeup request can only be sent after INTFLAG.SUSPEND has been set.

The user may then write a one to the Remote Wakeup bit in CTRLB(CTRLB.UPRSM) to send an Upstream Resume to the host initiating the wakeup. This will automatically be done by the controller after 5 ms of inactivity on the USB bus.

When the controller sends the Upstream Resume INTFLAG.WAKEUP is set and INTFLAG.SUSPEND is cleared.

The CTRLB.UPRSM is cleared at the end of the transmitting Upstream Resume.

In case of a rebroadcast resume initiated by the host, the End of Resume bit in INTFLAG(INTFLAG.EORSM) flag is set when the rebroadcast resume is completed.

In the case where the CTRLB.UPRSM bit is set while a host initiated downstream resume is already started, the CTRLB.UPRSM is cleared and the upstream resume request is ignored.

39.6.2.15 Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Device

The LPM Handshake bit in CTRLB.LPMHDSK should be configured to accept the LPM transaction.

When a LPM transaction is received on any enabled endpoint n and a handshake has been sent in response by the controller according to CTRLB.LPMHDSK, the Device Link Power Manager (EXTREG) register is updated in the bank 0 of the addressed endpoint's descriptor. It contains information such as the Best Effort Service Latency (BESL), the Remote Wake bit (bRemoteWake), and the Link State parameter (bLinkState). Usually, the LPM transaction uses only the endpoint number 0.

If the LPM transaction was positively acknowledged (ACK handshake), USB sets the Link Power Management Interrupt bit in INTFLAG(INTFLAG.LPMSUSP) bit which indicates that the USB transceiver is suspended, reducing power consumption. This suspend occurs 9 microseconds after the LPM transaction according to the specification.

To further reduce consumption, it is recommended to stop the USB clock while the device is suspended.

The MCU can also enter in one of the available sleep modes if the wakeup time latency of the selected sleep mode complies with the host latency constraint (see the BESL parameter in EXTREG register).

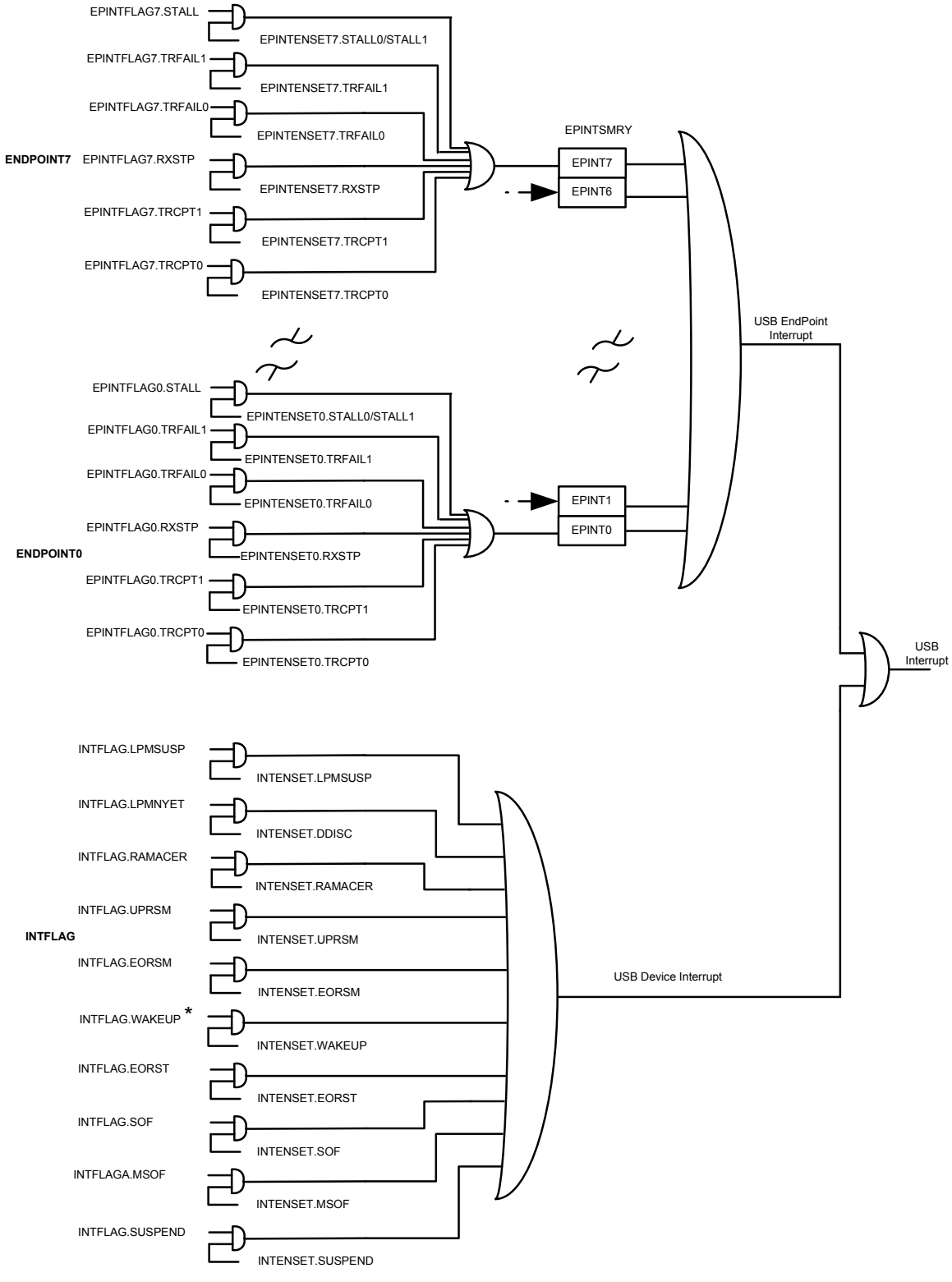
Recovering from this LPM-L1 suspend state is exactly the same as the Suspend state (see Section Suspend State and Pad Behavior) except that the remote wakeup duration initiated by USB is shorter to comply with the Link Power Management specification.

If the LPM transaction is responded with a NYET, the Link Power Management Not Yet Interrupt Flag INTFLAG(INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit in INTENCLR/SET (INTENCLR/SET.LPMNYET) is set.

If the LPM transaction is responded with a STALL or no handshake, no flag is set, and the transaction is ignored.

39.6.2.16 USB Device Interrupt

Figure 39-1. Device Interrupt



* Asynchronous interrupt

The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

39.7 Register Summary

The register mapping depends on the Operating Mode field in the Control A register (CTRLA.MODE). The register summary is detailed below.

39.7.1 Common Device Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MODE					RUNSTBY	ENABLE	SWRST
0x01	Reserved									
0x02	SYNCBUSY	7:0							ENABLE	SWRST
0x03	QOSCTRL	7:0					DQOS[1:0]		CQOS[1:0]	
0x0D	FSMSTATUS	7:0		FSMSTATE[6:0]						
0x24	DESCADD	7:0	DESCADD[7:0]							
0x25		15:8	DESCADD[15:8]							
0x26		23:16	DESCADD[23:16]							
0x27		31:24	DESCADD[31:24]							
0x28	PADCAL	7:0	TRANSN[1:0]			TRANSP[4:0]				
0x29		15:8		TRIM[2:0]				TRANSN[4:2]		

39.7.2 Device Summary

Table 39-1. General Device Registers

Offset	Name	Bit Pos.								
0x04	Reserved									
0x05	Reserved									
0x06	Reserved									
0x07	Reserved									
0x0A	DADD		ADDEN	DADD[6:0]						
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0E	Reserved									
0x0F	Reserved									
0x10	FNUM	7:0	FNUM[4:0]							
0x11		15:8	FNCERR		FNUM[10:5]					
0x12	Reserved									
0x16	Reserved									
0x17	Reserved									
0x1A	Reserved									
0x1B	Reserved									
0x1E	Reserved									
0x1F	Reserved									

Table 39-2. Device Endpoint Register n

Offset	Name	Bit Pos.								
0x1m1	Reserved									
0x1m2	Reserved									

Offset	Name	Bit Pos.								
0x1m3	Reserved									
0x1m4	EPSTATUSCLRn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m5	EPSTATUSSETn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m6	EPSTATUSn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m7	EPINTFLAGn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1m8	EPINTENCLRn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1m9	EPINTENSETn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1mA	Reserved									
0x1mB	Reserved									

Table 39-3. Device Endpoint n Descriptor Bank 0

Offset 0x n0 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	EXTREG	7:0	VARIABLE[3:0]				SUBPID[3:0]			
0x09		15:8		VARIABLE[10:4]						
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved	7:0								
0x0C	Reserved	7:0								
0x0D	Reserved	7:0								
0x0E	Reserved	7:0								
0x0F	Reserved	7:0								

Table 39-4. Device Endpoint n Descriptor Bank 1

Offset 0x n0 + 0x10 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	Reserved	7:0								
0x09	Reserved	15:8								

Offset 0x n0 + 0x10 + index	Name	Bit Pos.								
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved	7:0								
0x0C	Reserved	7:0								
0x0D	Reserved	7:0								
0x0E	Reserved	7:0								
0x0F	Reserved	7:0								

39.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to the Register Access Protection, *PAC - Peripheral Access Controller* and *GCLK Synchronization* for details.

Related Links

PAC - Peripheral Access Controller

39.8.1 Communication Device Host Registers

39.8.1.1 Control A

Name: CTRLA
Offset: 0x00 [ID-0000306e]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronised

Bit	7	6	5	4	3	2	1	0
						RUNSTDBY	ENABLE	SWRST
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – RUNSTDBY: Run in Standby Mode
This bit is Enable-Protected.

Value	Description
0	USB clock is stopped in standby mode.
1	USB clock is running in standby mode

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Synchronization status enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is Write-Synchronized.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST: Software Reset
Writing a zero to this bit has no effect.

Writing a '1' to this bit resets all registers in the USB, to their initial state, and the USB will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is Write-Synchronized.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

39.8.1.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x02 [ID-0000306e]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE: Synchronization Enable status bit
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.

This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST: Synchronization Software Reset status bit
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.

This bit is set when the synchronization of SWRST register between clock domains is started.

39.8.1.3 QOS Control

Name: QOSCTRL
Offset: 0x03 [ID-0000306e]
Reset: 0x0F
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					DQOS[1:0]		CQOS[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bits 3:2 – DQOS[1:0]: Data Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write data operation. Refer to *SRAM Quality of Service*.

Bits 1:0 – CQOS[1:0]: Configuration Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write configuration operation. Refer to *SRAM Quality of Service*.

39.8.1.4 Finite State Machine Status

Name: FSMSTATUS
Offset: 0x0D [ID-0000306e]
Reset: 0xXXXX
Property: Read only

Bit	7	6	5	4	3	2	1	0
		FSMSTATE[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	1

Bits 6:0 – FSMSTATE[6:0]: Fine State Machine Status
These bits indicate the state of the finite state machine of the USB controller.

Value	Name	Description
0x01	OFF (L3)	Corresponds to the powered-off, disconnected, and disabled state.
0x02	ON (L0)	Corresponds to the Idle and Active states.
0x04	SUSPEND (L2)	
0x08	SLEEP (L1)	
0x10	DNRESUME	Down Stream Resume.
0x20	UPRESUME	Up Stream Resume.
0x40	RESET	USB lines Reset.
Others		Reserved

39.8.1.5 Descriptor Address

Name: DESCADD
Offset: 0x24 [ID-0000306e]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DESCADD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DESCADD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DESCADD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DESCADD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DESCADD[31:0]: Descriptor Address Value
These bits define the base address of the main USB descriptor in RAM. The two least significant bits must be written to zero.

39.8.1.6 Pad Calibration

The Pad Calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register by software, before enabling the USB, to achieve the specified accuracy.
Refer to *NVM Software Calibration Area Mapping* for further details.

Refer to for further details.

Name: PADCAL
Offset: 0x28 [ID-0000306e]
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
		TRIM[2:0]				TRANSN[4:2]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	TRANSN[1:0]			TRANSP[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bits 14:12 – TRIM[2:0]: Trim bits for DP/DM
These bits calibrate the matching of rise/fall of DP/DM.

Bits 10:6 – TRANSN[4:0]: Trimmable Output Driver Impedance N
These bits calibrate the NMOS output impedance of DP/DM drivers.

Bits 4:0 – TRANSP[4:0]: Trimmable Output Driver Impedance P
These bits calibrate the PMOS output impedance of DP/DM drivers.

39.8.2 Device Registers - Common

39.8.2.1 Control B

Name: CTRLB
Offset: 0x08 [ID-0000306e]
Reset: 0x0001
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					LPMHDSK[1:0]		GNAK	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit	7	6	5	4	3	2	1	0
				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
Access				R	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 11:10 – LPMHDSK[1:0]: Link Power Management Handshake
These bits select the Link Power Management Handshake configuration.

Value	Description
0x0	No handshake. LPM is not supported.
0x1	ACK
0x2	NYET
0x3	Reserved

Bit 9 – GNAK: Global NAK
This bit configures the operating mode of the NAK.

This bit is not synchronized.

Value	Description
0	The handshake packet reports the status of the USB transaction
1	A NAK handshake is answered for each USB transaction regardless of the current endpoint memory bank status

Bit 4 – NREPLY: No reply excepted SETUP Token
This bit is cleared by hardware when receiving a SETUP packet.

This bit has no effect for any other endpoint but endpoint 0.

Value	Description
0	Disable the “NO_REPLY” feature: Any transaction to endpoint 0 will be handled according to the USB2.0 standard.
1	Enable the “NO_REPLY” feature: Any transaction to endpoint 0 will be ignored except SETUP.

Bits 3:2 – SPDCONF[1:0]: Speed Configuration
These bits select the speed configuration.

Value	Description
0x0	FS: Full-speed
0x1	LS: Low-speed
0x2	
0x3	

Bit 1 – UPRSM: Upstream Resume
This bit is cleared when the USB receives a USB reset or once the upstream resume has been sent.

Value	Description
0	Writing a zero to this bit has no effect.
1	Writing a one to this bit will generate an upstream resume to the host for a remote wakeup.

Bit 0 – DETACH: Detach

Value	Description
0	The device is attached to the USB bus so that communications may occur.
1	It is the default value at reset. The internal device pull-ups are disabled, removing the device from the USB bus.

39.8.2.2 Device Address

Name: DADD
Offset: 0x0A [ID-0000306e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ADDEN	DADD[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – ADDEN: Device Address Enable
This bit is cleared when a USB reset is received.

Value	Description
0	Writing a zero will deactivate the DADD field (USB device address) and return the device to default address 0.
1	Writing a one will activate the DADD field (USB device address).

Bits 6:0 – DADD[6:0]: Device Address
These bits define the device address. The DADD register is reset when a USB reset is received.

39.8.2.3 Status

Name: STATUS
Offset: 0x0C [ID-0000306e]
Reset: 0x40
Property: -

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R/W	R/W		
Reset	0	1			0	1		

Bits 7:6 – LINESTATE[1:0]: USB Line State Status
These bits define the current line state DP/DM.

LINESTATE[1:0]	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0]: Speed Status
These bits define the current speed used of the device

.

SPEED[1:0]	SPEED STATUS
0x0	Low-speed mode
0x1	Full-speed mode
0x2	
0x3	Reserved

39.8.2.4 Device Frame Number

Name: FNUM
Offset: 0x10 [ID-0000306e]
Reset: 0x0000
Property: Read only

Bit	15	14	13	12	11	10	9	8
	FNCERR		FNUM[10:5]					
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]					MFNUM[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FNCERR: Frame Number CRC Error
This bit is cleared upon receiving a USB reset.

This bit is set when a corrupted frame number (or micro-frame number) is received.

This bit and the SOF (or MSOF) interrupt bit are updated at the same time.

Bits 13:3 – FNUM[10:0]: Frame Number
These bits are cleared upon receiving a USB reset.

These bits are updated with the frame number information as provided from the last SOF packet even if a corrupted SOF is received.

Bits 2:0 – MFNUM[2:0]: Micro Frame Number
These bits are cleared upon receiving a USB reset or at the beginning of each Start-of-Frame (SOF interrupt).

These bits are updated with the micro-frame number information as provided from the last MSOF packet even if a corrupted MSOF is received.

39.8.2.5 Device Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x14 [ID-0000306e]
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP: Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Link Power Management Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled and an interrupt request will be generated when the Link Power Management Suspend interrupt Flag is set.

Bit 8 – LPMNYET: Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Link Power Management Not Yet interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled and an interrupt request will be generated when the Link Power Management Not Yet interrupt Flag is set.

Bit 7 – RAMACER: RAM Access Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM: Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – EORSM: End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the End Of Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled and an interrupt request will be generated when the End Of Resume interrupt Flag is set.

Bit 4 – WAKEUP: Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – EORST: End of Reset Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the End of Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled and an interrupt request will be generated when the End of Reset interrupt Flag is set.

Bit 2 – SOF: Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Start-of-Frame interrupt Flag is set.

Bit 0 – SUSPEND: Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled and an interrupt request will be generated when the Suspend interrupt Flag is set.

39.8.2.6 Device Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x18 [ID-0000306e]
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP: Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Link Power Management Suspend Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled.

Bit 8 – LPMNYET: Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Link Power Management Not Yet interrupt bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled.

Bit 7 – RAMACER: RAM Access Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the RAM Access Enable bit and enable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM: Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Upstream Resume Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – EORSM: End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the End Of Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled.

Bit 4 – WAKEUP: Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled.

Bit 3 – EORST: End of Reset Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the End of Reset interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled.

Bit 2 – SOF: Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled.

Bit 0 – SUSPEND: Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Suspend interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled.

39.8.2.7 Device Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x01C [ID-0000306e]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP: Link Power Management Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB module acknowledge a Link Power Management Transaction (ACK handshake) and has entered the Suspended state and will generate an interrupt if INTENCLR/SET.LPMSUSP is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the LPMSUSP Interrupt Flag.

Bit 8 – LPMNYET: Link Power Management Not Yet Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB module acknowledges a Link Power Management Transaction (handshake is NYET) and will generate an interrupt if INTENCLR/SET.LPMNYET is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the LPMNYET Interrupt Flag.

Bit 7 – RAMACER: RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a RAM access underflow error occurs during IN data stage. This bit will generate an interrupt if INTENCLR/SET.RAMACER is one.

Writing a zero to this bit has no effect.

Bit 6 – UPRSM: Upstream Resume Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB sends a resume signal called “Upstream Resume” and will generate an interrupt if INTENCLR/SET.UPRSM is one.

Writing a zero to this bit has no effect.

Bit 5 – EORSM: End Of Resume Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB detects a valid “End of Resume” signal initiated by the host and will generate an interrupt if INTENCLR/SET.EORSM is one.

Writing a zero to this bit has no effect.

Bit 4 – WAKEUP: Wake Up Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB is reactivated by a filtered non-idle signal from the lines and will generate an interrupt if INTENCLR/SET.WAKEUP is one.

Writing a zero to this bit has no effect.

Bit 3 – EORST: End of Reset Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a USB “End of Reset” has been detected and will generate an interrupt if INTENCLR/SET.EORST is one.

Writing a zero to this bit has no effect.

Bit 2 – SOF: Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a USB “Start-of-Frame” has been detected (every 1 ms) and will generate an interrupt if INTENCLR/SET.SOF is one.

The FNUM is updated. In High Speed mode, the MFNUM register is cleared.

Writing a zero to this bit has no effect.

Bit 0 – SUSPEND: Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a USB “Suspend” idle state has been detected for 3 frame periods (J state for 3 ms) and will generate an interrupt if INTENCLR/SET.SUSPEND is one.

Writing a zero to this bit has no effect.

39.8.2.8 Endpoint Interrupt Summary

Name: EPINTSMRY
Offset: 0x20 [ID-0000306e]
Reset: 0x00000000
Property: -

Bit	15	14	13	12	11	10	9	8
	EPINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EPINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EPINT[15:0]: EndPoint Interrupt
The flag EPINT[n] is set when an interrupt is triggered by the EndPoint n. See EPINTFLAGn register in the device EndPoint section.

This bit will be cleared when no interrupts are pending for EndPoint n.

39.8.3 Device Registers - Endpoint

39.8.3.1 Device Endpoint Configuration register n

Name: EPCFGn
Offset: 0x100 + (n x 0x20) [ID-0000306e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		EPTYPE1[2:0]				EPTYPE0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 6:4 – EPTYPE1[2:0]: Endpoint Type for IN direction
These bits contains the endpoint type for IN direction.

Upon receiving a USB reset EPCFGn.EPTYPE1 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank1 is disabled.
0x1	Bank1 is enabled and configured as Control IN.
0x2	Bank1 is enabled and configured as Isochronous IN.
0x3	Bank1 is enabled and configured as Bulk IN.
0x4	Bank1 is enabled and configured as Interrupt IN.
0x5	Bank1 is enabled and configured as Dual-Bank OUT (Endpoint type is the same as the one defined in EPTYPE0)
0x6-0x7	Reserved

Bits 2:0 – EPTYPE0[2:0]: Endpoint Type for OUT direction
These bits contains the endpoint type for OUT direction.

Upon receiving a USB reset EPCFGn.EPTYPE0 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank0 is disabled.
0x1	Bank0 is enabled and configured as Control SETUP / Control OUT.
0x2	Bank0 is enabled and configured as Isochronous OUT.
0x3	Bank0 is enabled and configured as Bulk OUT.
0x4	Bank0 is enabled and configured as Interrupt OUT.
0x5	Bank0 is enabled and configured as Dual Bank IN (Endpoint type is the same as the one defined in EPTYPE1)
0x6-0x7	Reserved

39.8.3.2 EndPoint Status Clear n

Name: EPSTATUSCLRn
Offset: 0x104 + (n * 0x20) [ID-0000306e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY: Bank 1 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY: Bank 0 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1: STALL bank 1 Request
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0: STALL bank 0 Request
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK: Current Bank
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.CURBK bit.

Bit 1 – DTGLIN: Data Toggle IN
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT: Data Toggle OUT
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the EPSTATUS.DTGLOUT bit.

39.8.3.3 EndPoint Status Set n

Name: EPSTATUSSETn
Offset: 0x105 + (n x 0x20) [ID-0000306e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY: Bank 1 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY: Bank 0 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1: STALL Request bank 1
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0: STALL Request bank 0
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK: Current Bank
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.CURBK bit.

Bit 1 – DTGLIN: Data Toggle IN
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT: Data Toggle OUT
Writing a zero to this bit has no effect.

Writing a one to this bit will set the EPSTATUS.DTGLOUT bit.

39.8.3.4 EndPoint Status n

Name: EPSTATUSn
Offset: 0x106 + (n x 0x20) [ID-0000306e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY: Bank 1 is ready
For Control/OUT direction Endpoints, the bank is empty.

Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.

Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.

Value	Description
0	The bank number 1 is not ready : For IN direction Endpoints, the bank is not yet filled in.
1	The bank number 1 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bit 6 – BK0RDY: Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.

Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.

Value	Description
0	The bank number 0 is not ready : For IN direction Endpoints, the bank is not yet filled in. For Control/OUT direction Endpoints, the bank is empty.
1	The bank number 0 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bit 5 – STALLRQ1: STALL bank 1 request
Writing a zero to the bit EPSTATUSCLR.STALLRQ1 will clear this bit.

Writing a one to the bit EPSTATUSSET.STALLRQ1 will set this bit.

This bit is cleared by hardware when receiving a SETUP packet.

Value	Description
0	Disable STALLRQ1 feature.
1	Enable STALLRQ1 feature: a STALL handshake will be sent to the host in regards to bank1.

Bit 4 – STALLRQ0: STALL bank 0 request
Writing a zero to the bit EPSTATUSCLR.STALLRQ0 will clear this bit.

Writing a one to the bit EPSTATUSSET.STALLRQ0 will set this bit.

This bit is cleared by hardware when receiving a SETUP packet.

Value	Description
0	Disable STALLRQ0 feature.
1	Enable STALLRQ0 feature: a STALL handshake will be sent to the host in regards to bank0.

Bit 2 – CURBK: Current Bank
Writing a zero to the bit EPSTATUSCLR.CURBK will clear this bit.

Writing a one to the bit EPSTATUSSET.CURBK will set this bit.

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 1 – DTGLIN: Data Toggle IN Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLINCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLINSET will set this bit.

Value	Description
0	The PID of the next expected IN transaction will be zero: data 0.
1	The PID of the next expected IN transaction will be one: data 1.

Bit 0 – DTGLOUT: Data Toggle OUT Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLOUTCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLOUTSET will set this bit.

Value	Description
0	The PID of the next expected OUT transaction will be zero: data 0.
1	The PID of the next expected OUR transaction will be one: data 1.

39.8.3.5 Device EndPoint Interrupt Flag n

Name: EPINTFLAGn
Offset: 0x107 + (n x 0x20) [ID-0000306e]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1: Transmit Stall 1 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transmit Stall occurs and will generate an interrupt if EPINTENCLR/SET.STALL1 is one.

EPINTFLAG.STALL1 is set for a single bank IN endpoint or double bank IN/OUT endpoint when current bank is "1".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the STALL1 Interrupt Flag.

Bit 5 – STALL0: Transmit Stall 0 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transmit Stall occurs and will generate an interrupt if EPINTENCLR/SET.STALL0 is one.

EPINTFLAG.STALL0 is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the STALL0 Interrupt Flag.

Bit 4 – RXSTP: Received Setup Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Received Setup occurs and will generate an interrupt if EPINTENCLR/SET.RXSTP is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the RXSTP Interrupt Flag.

Bit 3 – TRFAIL1: Transfer Fail 1 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a transfer fail occurs and will generate an interrupt if EPINTENCLR/SET.TRFAIL1 is one.

EPINTFLAG.TRFAIL1 is set for a single bank IN endpoint or double bank IN/OUT endpoint when current bank is "1".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRFAIL1 Interrupt Flag.

Bit 2 – TRFAIL0: Transfer Fail 0 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a transfer fail occurs and will generate an interrupt if EPINTENCLR/SET.TRFAIL0 is one.

EPINTFLAG.TRFAIL0 is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRFAIL0 Interrupt Flag.

Bit 1 – TRCPT1: Transfer Complete 1 interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer Complete occurs and will generate an interrupt if EPINTENCLR/SET.TRCPT1 is one. EPINTFLAG.TRCPT1 is set for a single bank IN endpoint or double bank IN/OUT endpoint when current bank is "1".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRCPT1 Interrupt Flag.

Bit 0 – TRCPT0: Transfer Complete 0 interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer complete occurs and will generate an interrupt if EPINTENCLR/SET.TRCPT0 is one. EPINTFLAG.TRCPT0 is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRCPT0 Interrupt Flag.

39.8.3.6 Device EndPoint Interrupt Enable n

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Set (EPINTENSET) register.

Name: EPINTENCLRn
Offset: 0x108 + (n x 0x20) [ID-0000306e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1: Transmit STALL 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transmit Stall 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmit Stall 1 interrupt is disabled.
1	The Transmit Stall 1 interrupt is enabled and an interrupt request will be generated when the Transmit Stall 1 Interrupt Flag is set.

Bit 5 – STALL0: Transmit STALL 0 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transmit Stall 0 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmit Stall 0 interrupt is disabled.
1	The Transmit Stall 0 interrupt is enabled and an interrupt request will be generated when the Transmit Stall 0 Interrupt Flag is set.

Bit 4 – RXSTP: Received Setup Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Received Setup Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled and an interrupt request will be generated when the Received Setup Interrupt Flag is set.

Bit 3 – TRFAIL1: Transfer Fail 1 Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Fail 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail 1 interrupt is disabled.
1	The Transfer Fail 1 interrupt is enabled and an interrupt request will be generated when the Transfer Fail 1 Interrupt Flag is set.

Bit 2 – TRFAIL0: Transfer Fail 0 Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Fail 0 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail bank 0 interrupt is disabled.
1	The Transfer Fail bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Fail 0 Interrupt Flag is set.

Bit 1 – TRCPT1: Transfer Complete 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Complete 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete 1 interrupt is disabled.
1	The Transfer Complete 1 interrupt is enabled and an interrupt request will be generated when the Transfer Complete 1 Interrupt Flag is set.

Bit 0 – TRCPT0: Transfer Complete 0 interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Complete 0 interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete bank 0 interrupt is disabled.
1	The Transfer Complete bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Complete 0 Interrupt Flag is set.

39.8.3.7 Device Interrupt EndPoint Set n

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Set (EPINTENCLR) register. This register is cleared by USB reset or when EPEN[n] is zero.

Name: EPINTENSETn
Offset: 0x109 + (n x 0x20) [ID-0000306e]
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1: Transmit Stall 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transmit bank 1 Stall interrupt.

Value	Description
0	The Transmit Stall 1 interrupt is disabled.
1	The Transmit Stall 1 interrupt is enabled.

Bit 5 – STALL0: Transmit Stall 0 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transmit bank 0 Stall interrupt.

Value	Description
0	The Transmit Stall 0 interrupt is disabled.
1	The Transmit Stall 0 interrupt is enabled.

Bit 4 – RXSTP: Received Setup Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Received Setup interrupt.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled.

Bit 3 – TRFAIL1: Transfer Fail bank 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 2 – TRFAIL0: Transfer Fail bank 0 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 1 – TRCPT1: Transfer Complete bank 1 interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Complete 0 interrupt.

Value	Description
0	The Transfer Complete bank 1 interrupt is disabled.
1	The Transfer Complete bank 1 interrupt is enabled.

Bit 0 – TRCPT0: Transfer Complete bank 0 interrupt Enable
Writing a zero to this bit has no effect.

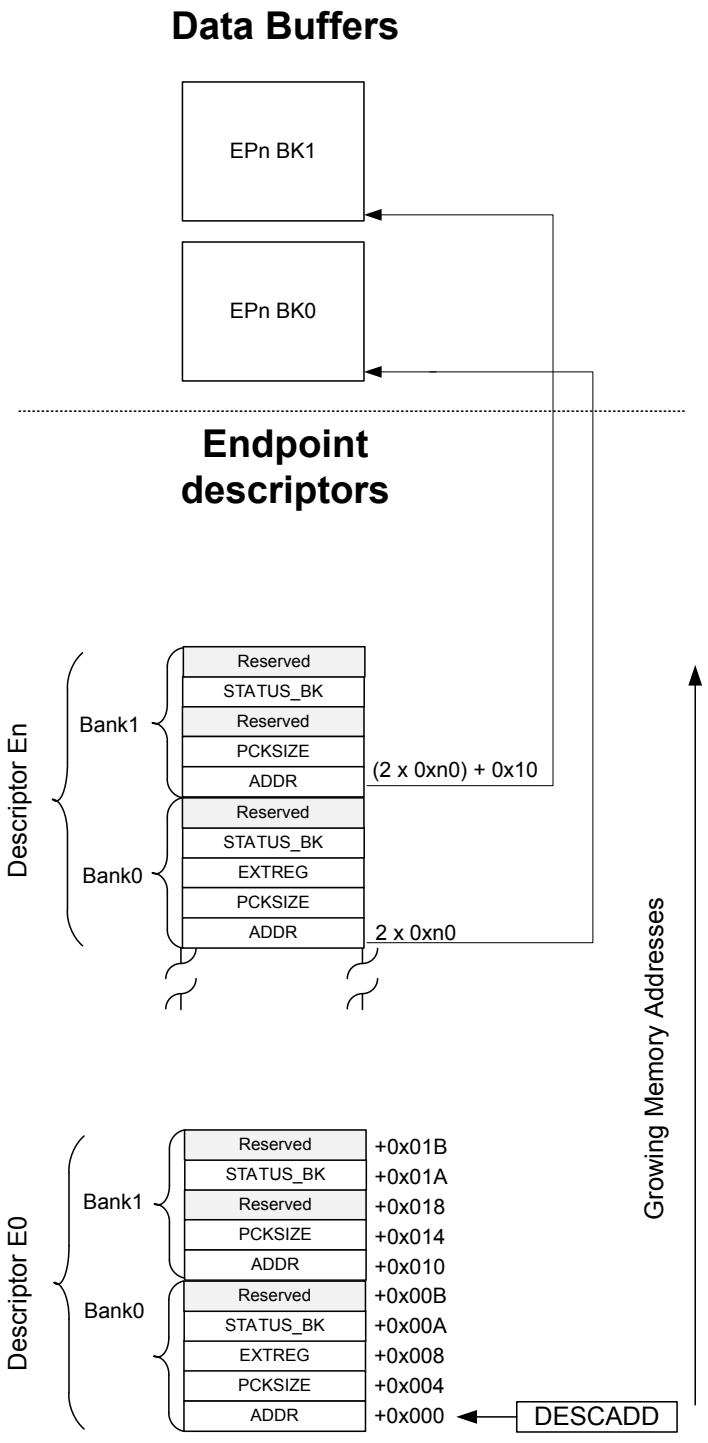
Writing a one to this bit will enable the Transfer Complete 1 interrupt.

0.2.4 Device Registers - Endpoint RAM

Value	Description
0	The Transfer Complete bank 0 interrupt is disabled.
1	The Transfer Complete bank 0 interrupt is enabled.

39.8.4 Device Registers - Endpoint RAM

39.8.4.1 Endpoint Descriptor Structure



39.8.4.2 Address of Data Buffer

Name: ADDR
Offset: 0x00 & 0x10 [ID-0000306e]
Reset: 0xxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 31:0 – ADDR[31:0]: Data Pointer Address Value
These bits define the data pointer address as an absolute word address in RAM. The two least significant bits must be zero to ensure the start address is 32-bit aligned.

39.8.4.3 Packet Size

Name: PCKSIZE
Offset: 0x04 & 0x14 [ID-0000306e]
Reset: 0xxxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BYTE_COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	x

Bit 31 – AUTO_ZLP: Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the endpoint.

When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for IN endpoints only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0]: Endpoint size
These bits contains the maximum packet size of the endpoint.

Value	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1023 Byte(1)

(1) for Isochronous endpoints only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0]: Multiple Packet Size
These bits define the 14-bit value that is used for multi-packet transfers.

For IN endpoints, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.

For OUT endpoints, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:0 – BYTE_COUNT[13:0]: Byte Count
These bits define the 14-bit value that is used for the byte count.

For IN endpoints, BYTE_COUNT holds the number of bytes to be sent in the next IN transaction.

For OUT endpoint or SETUP endpoints, BYTE_COUNT holds the number of bytes received upon the last OUT or SETUP transaction.

39.8.4.4 Extended Register

Name: EXTREG
Offset: 0x08 [ID-0000306e]
Reset: 0xxxxxxxx
Property: NA

Bit	15	14	13	12	11	10	9	8
		VARIABLE[10:4]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VARIABLE[3:0]				SUBPID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bits 14:4 – VARIABLE[10:0]: VARIABLE
These bits define the VARIABLE field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

To support the USB2.0 Link Power Management addition the VARIABLE field should be read as described below.

VARIABLES	Description
VARIABLE[3:0]	bLinkState (1)
VARIABLE[7:4]	BESL (2)
VARIABLE[8]	bRemoteWake (1)
VARIABLE[10:9]	Reserved

1. For a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum".

2. For a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

Bits 3:0 – SUBPID[3:0]: SUBPID
These bits define the SUBPID field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

39.8.4.5 Device Status Bank

Name: STATUS_BK
Offset: 0x0A & 0x1A [ID-0000306e]
Reset: 0xxxxxxxx
Property: NA

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW: Error Flow Status
This bit defines the Error Flow Status.

This bit is set when a Error Flow has been detected during transfer from/towards this bank.

For OUT transfer, a NAK handshake has been sent.

For Isochronous OUT transfer, an overrun condition has occurred.

For IN transfer, this bit is not valid. EPSTATUS.TRFAIL0 and EPSTATUS.TRFAIL1 should reflect the flow errors.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR: CRC Error
This bit defines the CRC Error Status.

This bit is set when a CRC error has been detected in an isochronous OUT endpoint bank.

0.2.5 Host Registers - Common

Value	Description
0	No CRC Error.
1	CRC Error detected.

40. CCL – Configurable Custom Logic

40.1 Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, to events, or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each LookUp Table (LUT) consists of three inputs, a truth table, and as options synchronizer, filter and edge detector. Each LUT can generate an output as a user programmable logic expression with three inputs. Inputs can be individually masked.

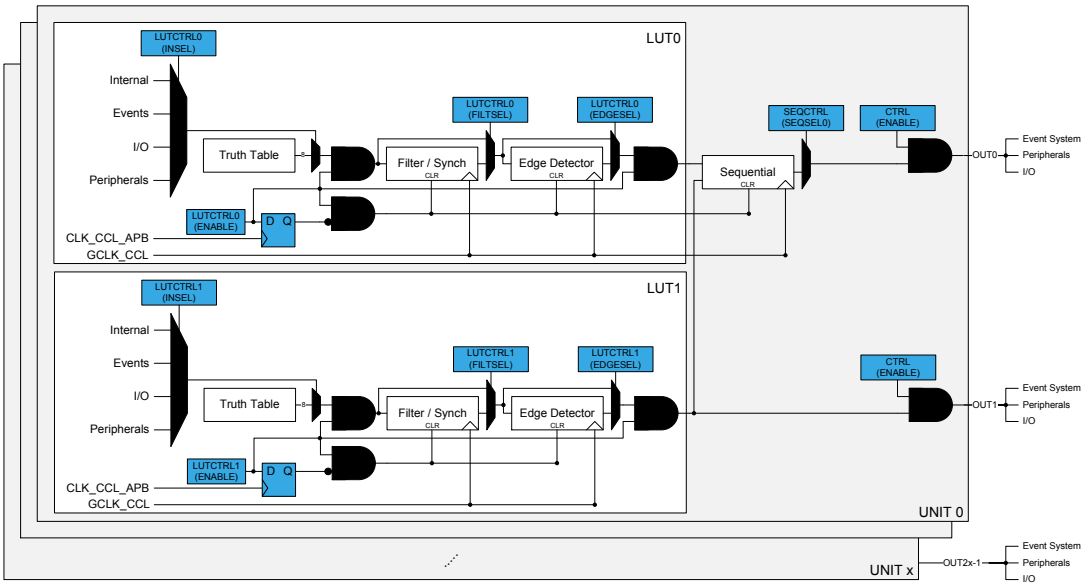
The output can be combinatorially generated from the inputs, and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1, LUT2/LUT3 etc.) outputs, enabling complex waveform generation.

40.2 Features

- Glue logic for general purpose PCB design
- Up to 4 programmable LookUp Tables (LUTs)
- Combinatorial logic functions:
  AND, NAND, OR, NOR, XOR, XNOR, NOT
- Sequential logic functions:
  Gated D Flip-Flop, JK Flip-Flop, gated D Latch, RS Latch
- Flexible LUT inputs selection:
  - I/Os
  - Events
  - Internal peripherals
  - Subsequent LUT output
- Output can be connected to the I/O pins or the Event System
- Optional synchronizer, filter, or edge detector available on each LUT output

40.3 Block Diagram

Figure 40-1. Configurable Custom Logic



40.4 Signal Description

Pin Name	Type	Description
OUT[n:0]	Digital output	Output from lookup table
IN[3n+2:0]	Digital input	Input to lookup table

1. n is the number of CCL groups.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

40.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

40.5.1 I/O Lines

Using the CCL I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

Related Links

PORT: IO Pin Controller

40.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

40.5.3 Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the power manager, and the default state of CLK_CCL_APB can be found in the *Peripheral Clock Masking*.

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using input events, filter, edge detection or sequential logic. GCLK_CCL is required when input events, a filter, an edge detector, or a sequential sub-module is enabled. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

Related Links

Peripheral Clock Masking

GCLK - Generic Clock Controller

40.5.4 DMA

Not applicable.

40.5.5 Interrupts

Not applicable.

40.5.6 Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

EVSYS – Event System

40.5.7 Debug Operation

When the CPU is halted in debug mode the CCL continues normal operation. If the CCL is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

40.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Refer to *PAC - Peripheral Access Controller* for details.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

40.5.9 Analog Connections

Not applicable.

40.6 Functional Description

40.6.1 Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins, internal peripherals, and the internal Event System as both input and output channels. The CCL can serve as glue logic between the device and external devices. The CCL can eliminate the need for external logic component and can also help the designer overcome challenging real-time constrains by combining core independent peripherals in clever ways to handle the most time critical parts of the application independent of the CPU.

40.6.2 Basic Operation

40.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control x (SEQCTRLx.SEQSEL) register

The following registers are enable-protected, meaning that they can only be written when the corresponding LUT is disabled (LUTCTRLx.ENABLE=0):

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1', but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

40.6.2.2 Enabling, Disabling, and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL will be reset to their initial state, and the CCL will be disabled. Refer to CTRL for details.

40.6.2.3 Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in Figure 40-2. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Figure 40-2. Truth Table Output Value Selection

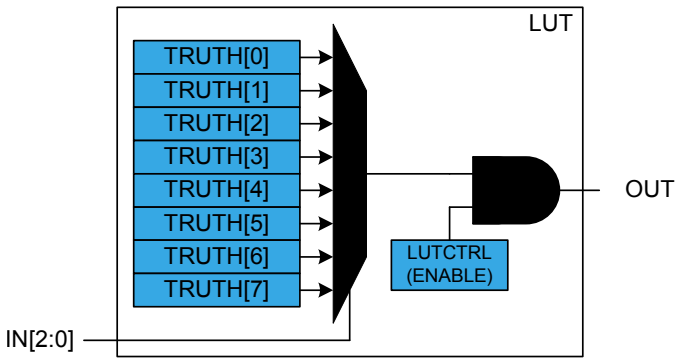


Table 40-1. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

40.6.2.4 Truth Table Inputs Selection

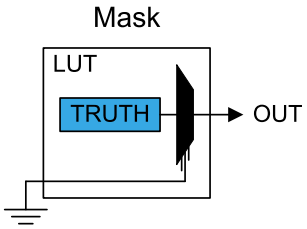
Input Overview
The inputs can be individually:

- Masked
- Driven by peripherals:
  - Analog comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

The Input Selection for each input y of LUT x is configured by writing the Input y Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

Masked Inputs (MASK)
When a LUT input is masked (LUTCTRLx.INSELy=MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

Figure 40-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)
When selected (LUTCTRLx.INSELy=FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

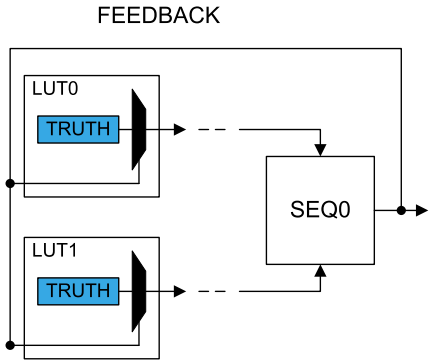
$\mathrm{IN}[2\mathrm{N}][i] = \mathrm{SEQ}[N]$

$\mathrm{IN}[2\mathrm{N}+1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i*=0,1,2 representing the LUT input index.

For details, refer to Sequential Logic.

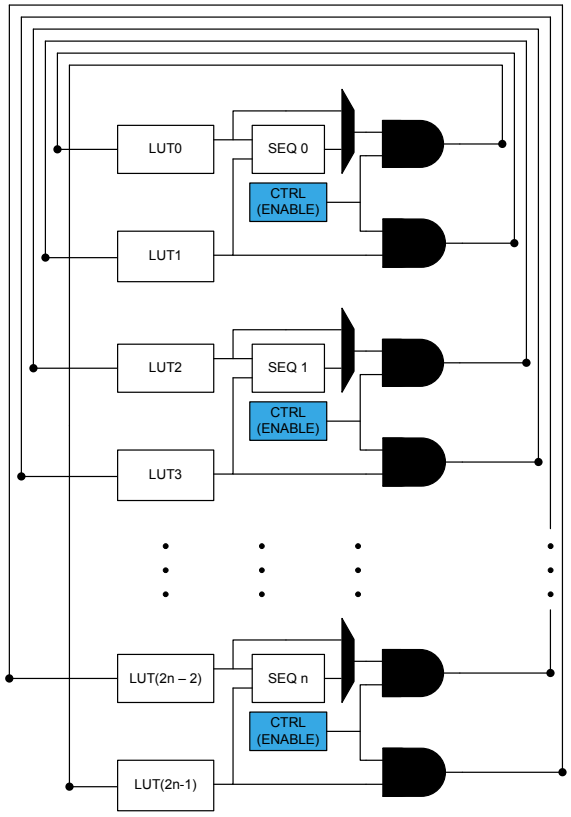
Figure 40-4. Feedback Input Selection



Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (e.g., LUT2 is the input for LUT1), as shown in this figure:

Figure 40-5. Linked LUT Input Selection



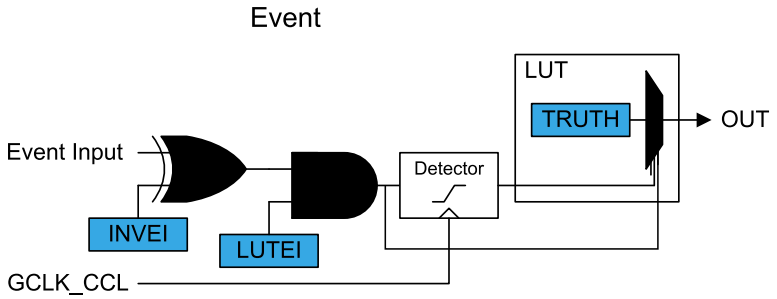
Internal Events Inputs Selection (EVENT)

Asynchronous events from the Event System can be used as input selection, as shown in Figure 40-6. For each LUT, one event input line is available and can be selected on each LUT input. Before enabling the event selection by writing LUTCTRLx.INSELy=EVENT, the Event System must be configured first.

By default CCL includes an edge detector. When the event is received, an internal strobe is generated when a rising edge is detected. The pulse duration is one GCLK_CCL clock cycle. Writing the LUTCTRLx.INSELy=ASYNCEVENT will disable the edge detector. In this case, it is possible to combine an asynchronous event input with any other input source. This is typically useful with event levels inputs (external IO pin events, as example). The following steps ensure proper operation:

1. Enable the GCLK_CCL clock.
2. Configure the Event System to route the event asynchronously.
3. Select the event input type (LUTCTRLx.INSEL).
4. If a strobe must be generated on the event input falling edge, write a '1' to the Inverted Event Input Enable bit in LUT Control register (LUTCTRLx.INVEI) .
5. Enable the event input by writing the Event Input Enable bit in LUT Control register (LUTCTRLx.LUTEI) to '1'.

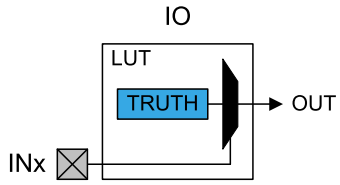
Figure 40-6. Event Input Selection



I/O Pin Inputs (IO)

When the IO pin is selected as LUT input (LUTCTRLx.INSELy=IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

Figure 40-7. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as input source for the LUT (LUTCTRLx.INSELy=AC).

The analog comparator outputs are distributed following the formula:

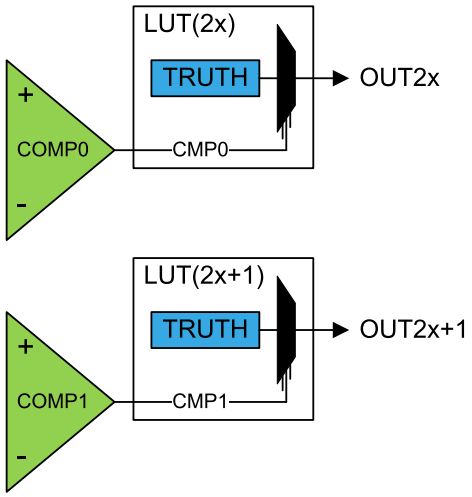
```
IN[N][i]=AC[N % ComparatorOutput_Number]
```

With *N* representing the LUT number and *i*=[0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0, LUT2) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1, LUT3), as shown in the figure below.

Figure 40-8. AC Input Selection



Timer/Counter Inputs (TC)

The TC waveform output WO[0] can be used as input source for the LUT (LUTCTRLx.INSELy=TC). Only consecutive instances of the TC, i.e. TCx and the subsequent TC(x+1), are available as default and alternative TC selections (e.g., TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1, etc). See the figure below for an example for LUT0. More general, the Timer/Counter selection for each LUT follows the formula:

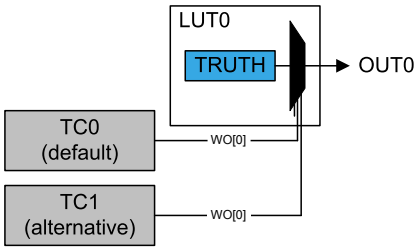
$$\mathrm{IN}[N][i] = DefaultTC[N \,\%\, \mathrm{TC_Instance_Number}]$$

$$\mathrm{IN}[N][i] = AlternativeTC[(N+1) \,\%\, \mathrm{TC_Instance_Number}]$$

Where N represents the LUT number and i represents the LUT input index (i=0,1,2).

Before selecting the waveform outputs, the TC must be configured first.

Figure 40-9. TC Input Selection

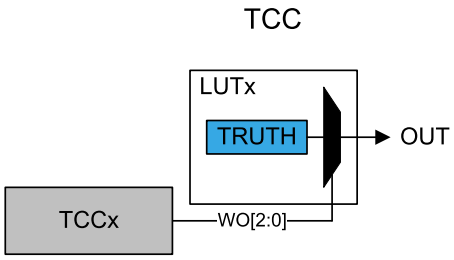


Timer/Counter for Control Application Inputs (TCC)

The TCC waveform outputs can be used as input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (i.e., IN0 is connected to WO0, IN1 to WO1, and IN2 to WO2), as shown in the figure below.

Before selecting the waveform outputs, the TCC must be configured first.

Figure 40-10. TCC Input Selection



Serial Communication Output Transmit Inputs (SERCOM)

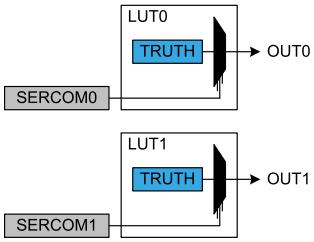
The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXd for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$$\mathrm{IN}[N][i] = SERCOM[N \,\%\, \mathrm{SERCOM_Instance_Number}]$$

With *N* representing the LUT number and *i*=0,1,2 representing the LUT input index.

Before selecting the SERCOM as input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as input pad to the CCL.

Figure 40-11. SERCOM Input Selection



Related Links

I/O Multiplexing and Considerations
PORT: IO Pin Controller
GCLK - Generic Clock Controller
AC – Analog Comparators
TC – Timer/Counter
TCC – Timer/Counter for Control Applications
SERCOM – Serial Communication Interface

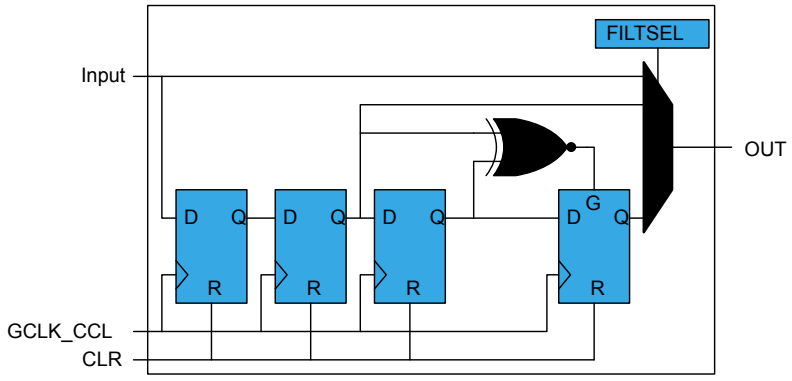
40.6.2.5 Filter

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.

Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 40-12. Filter



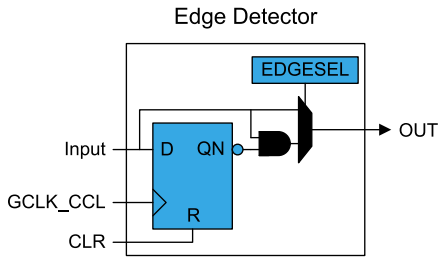
40.6.2.6 Edge Detector

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table should be inverted.

The edge detector is enabled by writing '1' to the Edge Selection bit in LUT Control register (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 40-13. Edge Detector



40.6.2.7 Sequential Logic

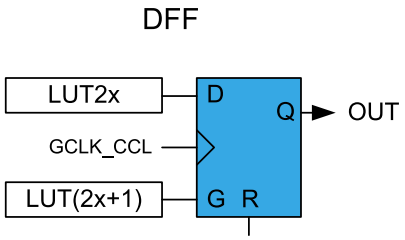
Each LUT pair can be connected to the internal sequential logic which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL clock and optionally each LUT filter or edge detector must be enabled.

Note: While configuring the sequential logic, the even LUT must be disabled. When configured the even LUT must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 40-14.

Figure 40-14. D Flip Flop



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 40-2.

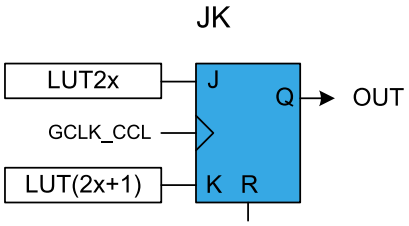
Table 40-2. DFF Characteristics

R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output (LUT0 and LUT2), and the K-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 40-15.

Figure 40-15. JK Flip Flop



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 40-3.

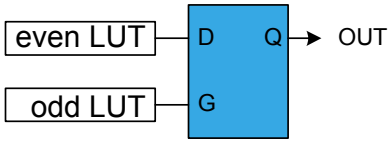
Table 40-3. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 40-14.

Figure 40-16. D-Latch



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 40-4.

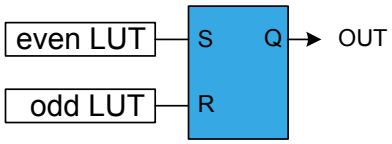
Table 40-4. D-Latch Characteristics

G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output (LUT0 and LUT2), and the R-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 40-17.

Figure 40-17. RS-Latch



When the even LUT is disabled LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 40-5.

Table 40-5. RS-latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

40.6.3 Events

The CCL can generate the following output events:

- LUTOUTx: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRL.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuration.

The CCL can take the following actions on an input event:

- INx: The event is used as input for the TRUTH table. For further details refer to Events.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event. Refer to *EVSYS – Event System* for details on configuration.

Related Links

EVSYS – Event System

40.6.4 Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in all sleep modes.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

40.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	SEQCTRL0	7:0					SEQSEL[3:0]			
0x05	SEQCTRL1	7:0					SEQSEL[3:0]			
0x06 ... 0x07	Reserved									
0x08	LUTCTRL0	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x09		15:8	INSELx[3:0]				INSELx[3:0]			
0x0A		23:16		LUTEO	LUTEI	INVEI	INSELx[3:0]			
0x0B		31:24	TRUTH[7:0]							
0x0C	LUTCTRL1	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x0D		15:8	INSELx[3:0]				INSELx[3:0]			
0x0E		23:16		LUTEO	LUTEI	INVEI	INSELx[3:0]			
0x0F		31:24	TRUTH[7:0]							
0x10	LUTCTRL2	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x11		15:8	INSELx[3:0]				INSELx[3:0]			
0x12		23:16		LUTEO	LUTEI	INVEI	INSELx[3:0]			
0x13		31:24	TRUTH[7:0]							
0x14	LUTCTRL3	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x15		15:8	INSELx[3:0]				INSELx[3:0]			
0x16		23:16		LUTEO	LUTEI	INVEI	INSELx[3:0]			
0x17		31:24	TRUTH[7:0]							

40.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

40.8.1 Control

Name: CTRL
Offset: 0x00 [ID-00000485]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	W
Reset		0					0	0

Bit 6 – RUNSTDBY: Run in Standby
This bit indicates if the GCLK_CCL clock must be kept running in standby mode. The setting is ignored for configurations where the generic clock is not required. For details refer to Sleep Mode Operation.

Value	Description
0	Generic clock is not required in standby sleep mode.
1	Generic clock is required in standby sleep mode.

Bit 1 – ENABLE: Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the CCL to their initial state.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

40.8.2 Sequential Control x

Name: SEQCTRL
Offset: 0x04 + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
					SEQSEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – SEQSEL[3:0]: Sequential Selection
These bits select the sequential configuration:

Sequential Selection

Value	Name	Description
0x0	DISABLE	Sequential logic is disabled
0x1	DFF	D flip flop
0x2	JK	JK flip flop
0x3	LATCH	D latch
0x4	RS	RS latch
0x5 - 0xF		Reserved

40.8.3 LUT Control x

Name: LUTCTRL
Offset: 0x08 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected (except LUTEN)

Bit	31	30	29	28	27	26	25	24
	TRUTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		LUTEO	LUTEI	INVEI	INSELx[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INSELx[3:0]				INSELx[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EDGESEL		FILTSEL[1:0]				ENABLE	
Access	R/W		R/W	R/W			R/W	
Reset	0		0	0			0	

Bits 31:24 – TRUTH[7:0]: Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

Bit 22 – LUTEO: LUT Event Output Enable

Value	Description
0	LUT event output is disabled.
1	LUT event output is enabled.

Bit 21 – LUTEI: LUT Event Input Enable

Value	Description
0	LUT incoming event is disabled.
1	LUT incoming event is enabled.

Bit 20 – INVEI: Inverted Event Input Enable

Value	Description
0	Incoming event is not inverted.
1	Incoming event is inverted.

Bit 7 – EDGESEL: Edge Selection

Value	Description
0	Edge detector is disabled.
1	Edge detector is enabled.

Bits 5:4 – FILTSEL[1:0]: Filter Selection
These bits select the LUT output filter options:

Filter Selection

Value	Name	Description
0x0	DISABLE	Filter disabled
0x1	SYNCH	Synchronizer enabled
0x2	FILTER	Filter enabled
0x3	-	Reserved

Bit 1 – ENABLE: LUT Enable

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

Bits 19:16,15:12,11:8 – INSELx: LUT Input x Source Selection
These bits select the LUT input x source:

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	AC	AC input source
0x6	TC	TC input source
0x7	ALTTC	Alternative TC input source
0x8	TCC	TCC input source
0x9	SERCOM	SERCOM input source
0xA - 0xF	-	Reserved

41. ADC – Analog-to-Digital Converter

41.1 Overview

The Analog-to-Digital Converter (ADC) converts analog signals to digital values. The ADC has up to 12-bit resolution, and is capable of a sampling rate of up to 1MSPS. The input selection is flexible, and both differential and single-ended measurements can be performed. In addition, several internal signal inputs are available. The ADC can provide both signed and unsigned results.

ADC measurements can be started by either application software or an incoming event from another peripheral in the device. ADC measurements can be started with predictable timing, and without software intervention.

Both internal and external reference voltages can be used.

The bandgap voltage as well as the scaled I/O and core voltages can also be measured by the ADC.

The ADC has a compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required.

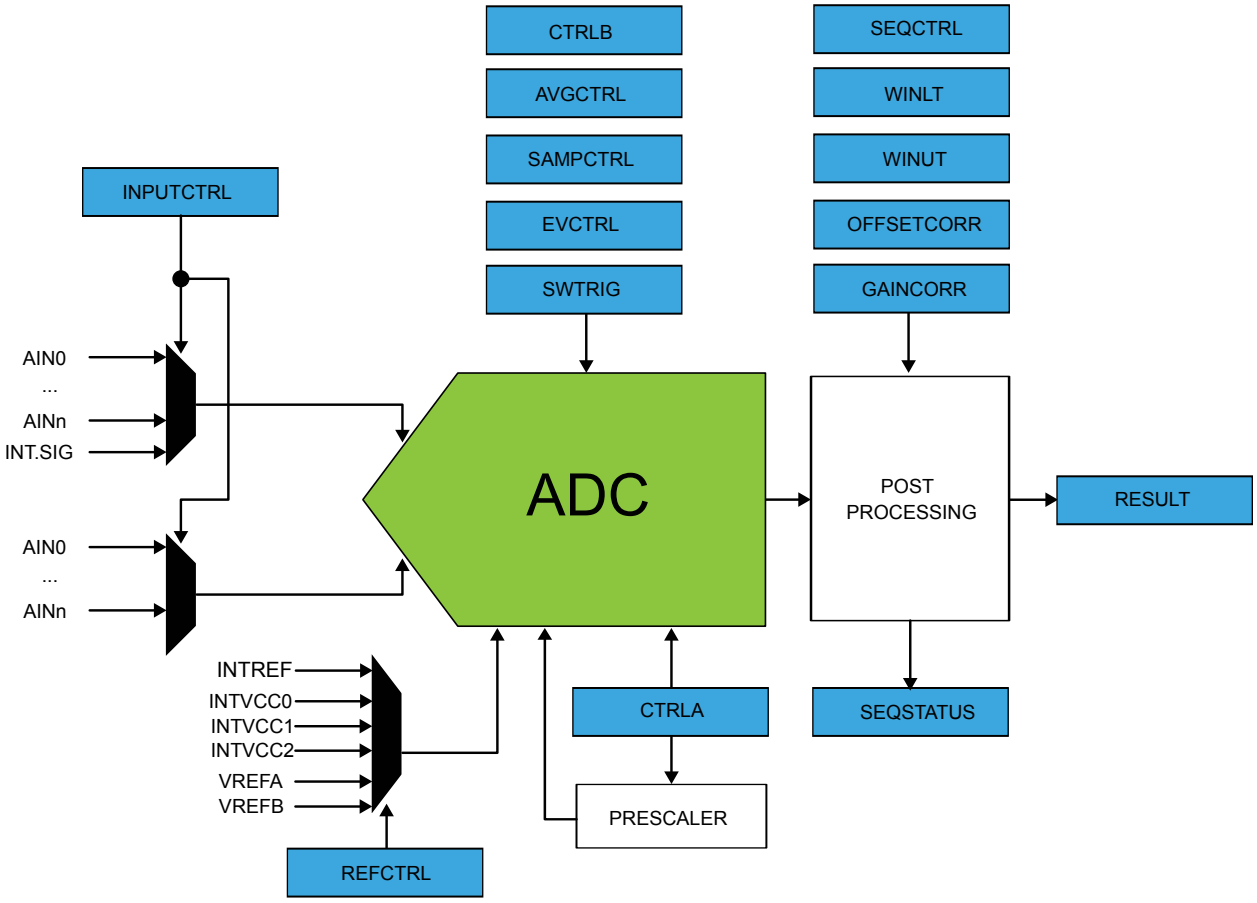
The ADC can be configured for 8-, 10- or 12-bit results. ADC conversion results are provided left- or right-adjusted, which eases calculation when the result is represented as a signed value. It is possible to use DMA to move ADC results directly to memory or peripherals when conversions are done.

41.2 Features

- 8-, 10- or 12-bit resolution
- Up to 1,000,000 samples per second (1MSPS)
- Differential and single-ended inputs
  - Up to 20 analog inputs
    26 positive and 9 negative, including internal and external
- Internal inputs:
  - Bandgap voltage
  - Scaled core supply
  - Scaled I/O supply
  - Scaled VBAT supply
- Single, continuous and sequencing options
- Windowing monitor with selectable channel
- Conversion range: V_{ref} = [1.0V to VDD_{ANA}]
- Built-in internal reference and external reference options
- Event-triggered conversion for accurate timing (one event input)
- Optional DMA transfer of conversion settings or result
- Hardware gain and offset compensation
- Averaging and oversampling with decimation to support up to 16-bit result
- Selectable sampling time
- Flexible Power / Throughput rate management

41.3 Block Diagram

Figure 41-1. ADC Block Diagram



41.4 Signal Description

Signal	Description	Type
VREF[A, B]	Analog input	External reference voltage
AIN[19..0]	Analog input	Analog input channels

Note: One signal can be mapped on several pins.

Related Links

Configuration Summary

I/O Multiplexing and Considerations

41.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

41.5.1 I/O Lines

Using the ADC's I/O lines requires the I/O pins to be configured using the port configuration (PORT).

Related Links

PORT: IO Pin Controller

41.5.2 Power Management

The ADC will continue to operate in any sleep mode where the selected source clock is running. The ADC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

41.5.3 Clocks

The ADC bus clock (CLK_APB_ADCx) can be enabled in the Main Clock, which also defines the default state.

The ADC requires a generic clock (GCLK_ADC). This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the ADC.

A generic clock is asynchronous to the bus clock. Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to *Synchronization* for further details.

Related Links

Synchronization

Peripheral Clock Masking

GCLK - Generic Clock Controller

41.5.4 DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the ADC DMA requests requires the DMA Controller to be configured first.

Related Links

DMAC – Direct Memory Access Controller

41.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the ADC interrupt requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller

41.5.6 Events

The events are connected to the Event System.

Related Links

EVSYS – Event System

41.5.7 Debug Operation

When the CPU is halted in debug mode the ADC will halt normal operation. The ADC can be forced to continue operation during debugging. Refer to DBGCTRL register for details.

41.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following register:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

41.5.9 Analog Connections

I/O-pins (AINx), as well as the VREF[A, B] reference voltage pins are analog inputs to the ADC.

41.5.10 Calibration

The BIAS and LINEARITY calibration values from the production test must be loaded from the NVM Software Calibration Area into the ADC Calibration register (CALIB) by software to achieve specified accuracy.

41.6 Functional Description

41.6.1 Principle of Operation

By default, the ADC provides results with 12-bit resolution. 8-bit or 10-bit results can be selected in order to reduce the conversion time, see Conversion Timing and Sampling Rate.

The ADC has an oversampling with decimation option that can extend the resolution to 16 bits. The input values can be either internal or external (connected I/O pins). The user can also configure whether the conversion should be single-ended or differential.

41.6.2 Basic Operation

41.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the ADC is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Reference Control register (REFCTRL)
- Event Control register (EVCTRL)
- Calibration register (CALIB)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

41.6.2.2 Enabling, Disabling and Resetting

The ADC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The ADC is disabled by writing CTRLA.ENABLE=0.

The ADC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the ADC, except DBGCTRL, will be reset to their initial state, and the ADC will be disabled. Refer to CTRLA for details.

41.6.2.3 Operation

In the most basic configuration, the ADC samples values from the configured internal or external sources (INPUTCTRL register). The rate of the conversion depends on the combination of the GCLK_ADC frequency and the clock prescaler.

To convert analog values to digital values, the ADC needs to be initialized first, as described in the Initialization section. Data conversion can be started either manually by setting the Start bit in the Software Trigger register (SWTRIG.START=1), or automatically by configuring an automatic trigger to initiate the conversions. A free-running mode can be used to continuously convert an input channel. When using free-running mode the first conversion must be started, while subsequent conversions will start automatically at the end of previous conversions.

The result of the conversion is stored in the Result register (RESULT) overwriting the result from the previous conversion.

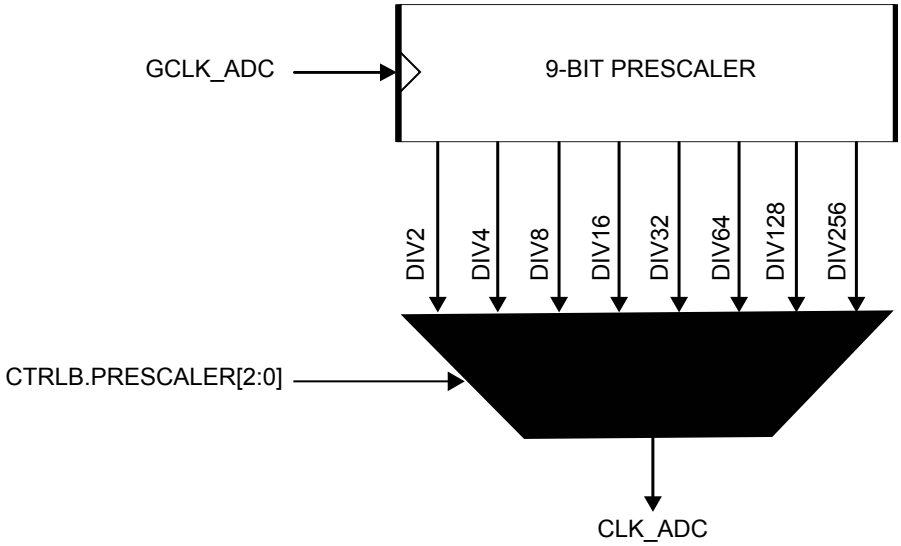
To avoid data loss if more than one channel is enabled, the conversion result must be read as soon as it is available (INTFLAG.RESRDY). Failing to do so will result in an overrun error condition, indicated by the OVERRUN bit in the Interrupt Flag Status and Clear register (INTFLAG.OVERRUN).

To enable one of the available interrupts sources, the corresponding bit in the Interrupt Enable Set register (INTENSET) must be written to '1'.

41.6.2.4 Prescaler Selection

The ADC is clocked by GCLK_ADC. There is also a prescaler in the ADC to enable conversion at lower clock rates. Refer to CTRLB for details on prescaler settings. Refer to Conversion Timing and Sampling Rate for details on timing and sampling rate.

Figure 41-2. ADC Prescaler



Note: The minimum prescaling factor is DIV2.

41.6.2.5 Reference Configuration

The ADC has various sources for its reference voltage V_{REF}. The Reference Voltage Selection bit field in the Reference Control register (REFCTRL.REFSEL) determines which reference is selected. By default, the internal voltage reference INTREF is selected. Based on customer application requirements, the external or internal reference can be selected. Refer to REFCTRL.REFSEL for further details on available selections.

Related Links

REFCTRL

41.6.2.6 ADC Resolution

The ADC supports 8-bit, 10-bit or 12-bit resolution. Resolution can be changed by writing the Resolution bit group in the Control C register (CTRLC.RESSEL). By default, the ADC resolution is set to 12 bits. The resolution affects the propagation delay, see also Conversion Timing and Sampling Rate.

41.6.2.7 Differential and Single-Ended Conversions

The ADC has two conversion options: differential and single-ended:

If the positive input is always positive, the single-ended conversion should be used in order to have full 12-bit resolution in the conversion.

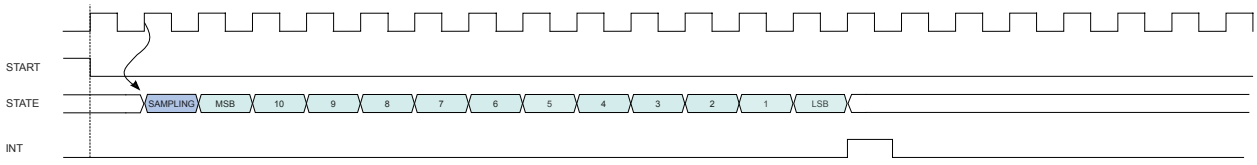
If the positive input may go below the negative input, the differential mode should be used in order to get correct results.

The differential mode is enabled by setting DIFFMODE bit in the Control C register (CTRLC.DIFFMODE). Both conversion types could be run in single mode or in free-running mode. When the free-running mode is selected, an ADC input will continuously sample the input and performs a new conversion. The INTFLAG.RESRDY bit will be set at the end of each conversion.

41.6.2.8 Conversion Timing and Sampling Rate

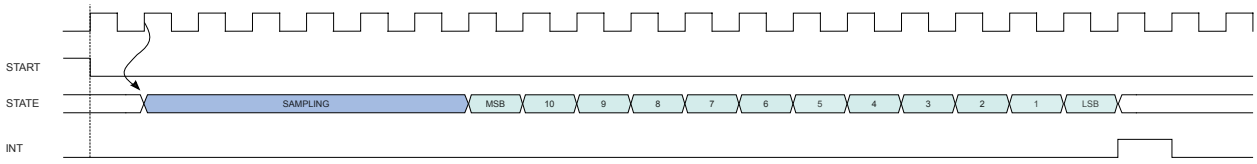
The following figure shows the ADC timing for one single conversion. A conversion starts after the software or event start are synchronized with the GCLK_ADC clock. The input channel is sampled in the first half CLK_ADC period.

Figure 41-3. ADC Timing for One Conversion in 12-bit Resolution



The sampling time can be increased by using the Sampling Time Length bit group in the Sampling Time Control register (SAMPCTRL.SAMPLEN). As example, the next figure is showing the timing conversion with sampling time increased to six CLK_ADC cycles.

Figure 41-4. ADC Timing for One Conversion with Increased Sampling Time, 12-bit



The ADC provides also offset compensation, see the following figure. The offset compensation is enabled by the Offset Compensation bit in the Sampling Control register (SAMPCTRL.OFFCOMP).

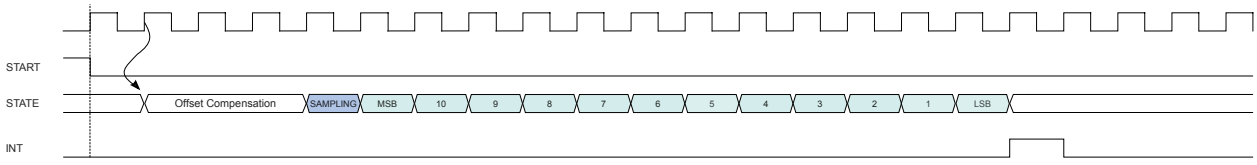
Note: If offset compensation is used, the sampling time must be set to one cycle of CLK_ADC.

In free running mode, the sampling rate R_S is calculated by

$R_S = f_{CLK_ADC} / (n_{SAMPLING} + n_{OFFCOMP} + n_{DATA})$

Here, $n_{SAMPLING}$ is the sampling duration in CLK_ADC cycles, $n_{OFFCOMP}$ is the offset compensation duration in clock cycles, and n_{DATA} is the bit resolution. f_{CLK_ADC} is the ADC clock frequency from the internal prescaler: $f_{CLK_ADC} = f_{GCLK_ADC} / 2^{(1 + CTRLB.PRESCALER)}$

Figure 41-5. ADC Timing for One Conversion with Offset Compensation, 12-bit



The impact of resolution on the sampling rate is seen in the next two figures, where free-running sampling in 12-bit and 8-bit resolution are compared.

Figure 41-6. ADC Timing for Free Running in 12-bit Resolution

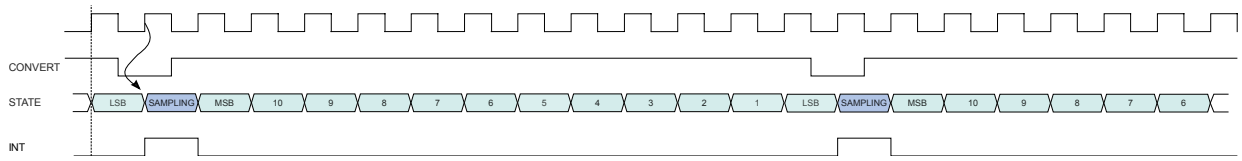
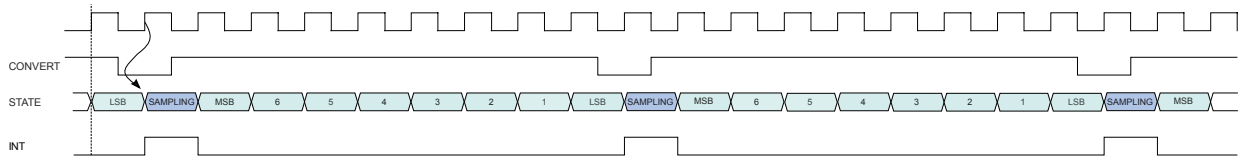


Figure 41-7. ADC Timing for Free Running in 8-bit Resolution



The propagation delay of an ADC measurement is given by:

$$\text{PropagationDelay} = \frac{1 + \text{Resolution}}{f_{\text{ADC}}}$$

> **Example.** In order to obtain 1MSPS in 12-bit resolution with a sampling time length of four CLK_ADC cycles, f_{CLK_ADC} must be 1MSPS * (4 + 12) = 16MHz. As the minimal division factor of the prescaler is 2, GCLK_ADC must be 32MHz.

41.6.2.9 Accumulation

The result from multiple consecutive conversions can be accumulated. The number of samples to be accumulated is specified by the Sample Number field in the Average Control register (AVGCTRL.SAMPLENUM). When accumulating more than 16 samples, the result will be too large to match the 16-bit RESULT register size. To avoid overflow, the result is right shifted automatically to fit within the available register size. The number of automatic right shifts is specified in the table below.

Note: To perform the accumulation of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Table 41-1. Accumulation

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	0	12 bits	0
2	0x1	0	13 bits	0

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
4	0x2	0	14 bits	0
8	0x3	0	15 bits	0
16	0x4	0	16 bits	0
32	0x5	1	16 bits	2
64	0x6	2	16 bits	4
128	0x7	3	16 bits	8
256	0x8	4	16 bits	16
512	0x9	5	16 bits	32
1024	0xA	6	16 bits	64
Reserved	0xB –0xF		12 bits	0

41.6.2.10 Averaging

Averaging is a feature that increases the sample accuracy, at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating m samples, as described in Accumulation, and dividing the result by m. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 41-2.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 41-2.

Note: To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\text{AVGCTRL.SAMPLENUM}}$.

When the averaged result is available, the INTFLAG.RESRDY bit will be set.

Table 41-2. Averaging

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	1	0x0		12 bits	0
2	0x1	13	0	2	0x1	1	12 bits	0
4	0x2	14	0	4	0x2	2	12 bits	0
8	0x3	15	0	8	0x3	3	12 bits	0
16	0x4	16	0	16	0x4	4	12 bits	0
32	0x5	17	1	16	0x4	5	12 bits	2

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
64	0x6	18	2	16	0x4	6	12 bits	4
128	0x7	19	3	16	0x4	7	12 bits	8
256	0x8	20	4	16	0x4	8	12 bits	16
512	0x9	21	5	16	0x4	9	12 bits	32
1024	0xA	22	6	16	0x4	10	12 bits	64
Reserved	0xB –0xF				0x0		12 bits	0

41.6.2.11 Oversampling and Decimation

By using oversampling and decimation, the ADC resolution can be increased from 12 bits up to 16 bits, for the cost of reduced effective sampling rate.

To increase the resolution by n bits, 4^n samples must be accumulated. The result must then be right-shifted by n bits. This right-shift is a combination of the automatic right-shift and the value written to AVGCTRL.ADJRES. To obtain the correct resolution, the ADJRES must be configured as described in the table below. This method will result in n bit extra LSB resolution.

Table 41-3. Configuration Required for Oversampling and Decimation

Result Resolution	Number of Samples to Average	AVGCTRL.SAMPLENUM[3:0]	Number of Automatic Right Shifts	AVGCTRL.ADJRES[2:0]
13 bits	$4^1 = 4$	0x2	0	0x1
14 bits	$4^2 = 16$	0x4	0	0x2
15 bits	$4^3 = 64$	0x6	2	0x1
16 bits	$4^4 = 256$	0x8	4	0x0

41.6.2.12 Automatic Sequences

The ADC has the ability to automatically sequence a series of conversions. This means that each time the ADC receives a start-of-conversion request, it can perform multiple conversions automatically. All of the 32 positive inputs can be included in a sequence by writing to corresponding bits in the Sequence Control register (SEQCTRL). The order of the conversion in a sequence is the lower positive MUX selection to upper positive MUX (AIN0, AIN1, AIN2 ...). In differential mode, the negative inputs selected by MUXNEG field, will be used for the entire sequence.

When a sequence starts, the Sequence Busy status bit in Sequence Status register (SEQSTATUS.SEQBUSY) will be set. When the sequence is complete, the Sequence Busy status bit will be cleared.

Each time a conversion is completed, the Sequence State bit in Sequence Status register (SEQSTATUS.SEQSTATE) will store the input number from which the conversion is done. The result will be stored in the RESULT register, and the Result Ready Interrupt Flag (INTFLAG.RESRDY) is set.

If additional inputs must be scanned, the ADC will automatically start a new conversion on the next input present in the sequence list.

Note that if SEQCTRL register has no bits set to '1', the conversion is done with the selected MUXPOS input.

41.6.2.13 Window Monitor

The window monitor feature allows the conversion result in the RESULT register to be compared to predefined threshold values. The window mode is selected by setting the Window Monitor Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT) and Window Monitor Upper Threshold register (WINUT).

If differential input is selected, the WINLT and WINUT are evaluated as signed values. Otherwise they are evaluated as unsigned values. The significant WINLT and WINUT bits are given by the precision selected in the Conversion Result Resolution bit group in the Control C register (CTRLC.RESSEL). This means that for example in 8-bit mode, only the eight lower bits will be considered. In addition, in differential mode, the eighth bit will be considered as the sign bit, even if the ninth bit is zero.

The INTFLAG.WINMON interrupt flag will be set if the conversion result matches the window monitor condition.

41.6.2.14 Offset and Gain Correction

Inherent gain and offset errors affect the absolute accuracy of the ADC.

The offset error is defined as the deviation of the actual ADC transfer function from an ideal straight line at zero input voltage. The offset error cancellation is handled by the Offset Correction register (OFFSETCORR). The offset correction value is subtracted from the converted data before writing the Result register (RESULT).

The gain error is defined as the deviation of the last output step's midpoint from the ideal straight line, after compensating for offset error. The gain error cancellation is handled by the Gain Correction register (GAINCORR).

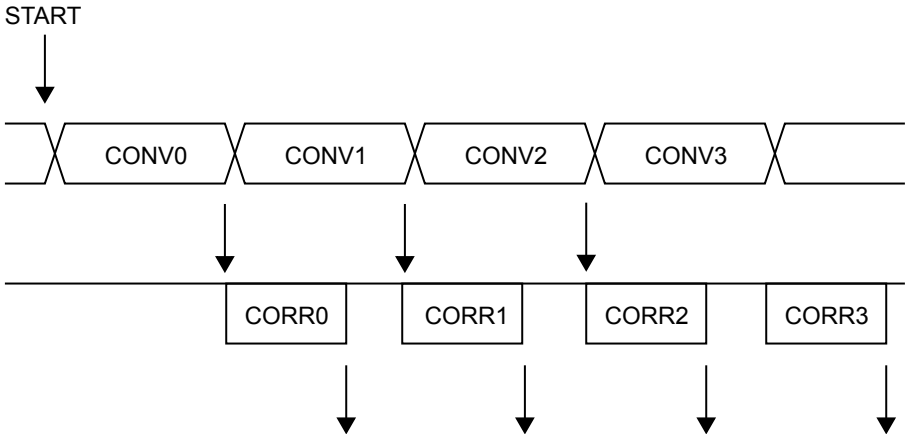
To correct these two errors, the Digital Correction Logic Enabled bit in the Control C register (CTRLC.CORREN) must be set.

Offset and gain error compensation results are both calculated according to:

$$\text{Result} = (\text{Conversion value} + -\text{OFFSETCORR}) \cdot \text{GAINCORR}$$

The correction will introduce a latency of 13 CLK_ADC clock cycles. In free running mode this latency is introduced on the first conversion only, since its duration is always less than the propagation delay. In single conversion mode this latency is introduced for each conversion.

Figure 41-8. ADC Timing Correction Enabled



41.6.3 DMA Operation

The ADC generates the following DMA request:

- Result Conversion Ready (RESRDY): the request is set when a conversion result is available and cleared when the RESULT register is read. When the averaging operation is enabled, the DMA request is set when the averaging is completed and result is available.

41.6.4 Interrupts

The ADC has the following interrupt sources:

- Result Conversion Ready: RESRDY
- Window Monitor: WINMON
- Overrun: OVERRUN

These interrupts are asynchronous wake-up sources. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the ADC is reset. See INTFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

Sleep Mode Controller

INTENCLR

INTENSET

INTFLAG

41.6.5 Events

The ADC can generate the following output events:

- Result Ready (RESRDY): Generated when the conversion is complete and the result is available. Refer to EVCTRL for details.
- Window Monitor (WINMON): Generated when the window monitor condition match. Refer to CTRLC for details.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The ADC can take the following actions on an input event:

- Start conversion (START): Start a conversion. Refer to SWTRIG for details.
- Conversion flush (FLUSH): Flush the conversion. Refer to SWTRIG for details.

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

The ADC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the ADC will detect a rising edge on the incoming event. If the ADC action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.xINV=1).

Note: If several events are connected to the ADC, the enabled action will be taken on any of the incoming events. If FLUSH and START events are available at the same time, the FLUSH event has priority.

Related Links

EVSYS – Event System

41.6.6 Sleep Mode Operation

The ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the ADC during standby sleep mode, in cases where the ADC is enabled (CTRLA.ENABLE = 1). For further details on available options, refer to Table 41-4.

Note: When CTRLA.ONDEMAND=1, the analog block is powered-off when the conversion is complete. When a start request is detected, the system returns from sleep and starts a new conversion after the start-up time delay.

Table 41-4. ADC Sleep Behavior

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
x	x	0	Disabled
0	0	1	Run in all sleep modes except STANDBY.
0	1	1	Run in all sleep modes on request, except STANDBY.
1	0	1	Run in all sleep modes.
1	1	1	Run in all sleep modes on request.

41.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

The following registers are synchronized when written:

- Input Control register (INPUTCTRL)
- Control C register (CTRLC)
- Average control register (AVGCTRL)
- Sampling time control register (SAMPCTRL)
- Window Monitor Lower Threshold register (WINLT)
- Window Monitor Upper Threshold register (WINUT)
- Gain correction register (GAINCORR)
- Offset Correction register (OFFSETCORR)

- Software Trigger register (SWTRIG)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization

41.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	SWRST
0x01	CTRLB	7:0						PRESCALER[2:0]		
0x02	REFCTRL	7:0	REFCOMP				REFSEL[3:0]			
0x03	EVCTRL	7:0			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
0x04	INTENCLR	7:0						WINMON	OVERRUN	RESRDY
0x05	INTENSET	7:0						WINMON	OVERRUN	RESRDY
0x06	INTFLAG	7:0						WINMON	OVERRUN	RESRDY
0x07	SEQSTATUS	7:0	SEQBUSY			SEQSTATE[4:0]				
0x08	INPUTCTRL	7:0				MUXPOS[4:0]				
0x09		15:8				MUXNEG[4:0]				
0x0A	CTRLC	7:0			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
0x0B		15:8						WINMODE[2:0]		
0x0C	AVGCTRL	7:0		ADJRES[2:0]			SAMPLENUM[3:0]			
0x0D	SAMPCTRL	7:0	OFFCOMP		SAMPLEN[5:0]					
0x0E	WINLT	7:0	WINLT[7:0]							
0x0F		15:8	WINLT[15:8]							
0x10	WINUT	7:0	WINUT[7:0]							
0x11		15:8	WINUT[15:8]							
0x12	GAINCORR	7:0	GAINCORR[7:0]							
0x13		15:8					GAINCORR[11:8]			
0x14	OFFSETCORR	7:0	OFFSETCORR[7:0]							
0x15		15:8					OFFSETCORR[11:8]			
0x16 ... 0x17	Reserved									
0x18	SWTRIG	7:0							START	FLUSH
0x19 ... 0x1B	Reserved									
0x1C	DBGCTRL	7:0								DBGRUN
0x1D ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
0x21		15:8						SWTRIG	OFFSETCOR R	GAINCORR
0x22 ... 0x23	Reserved									
0x24	RESULT	7:0	RESULT[7:0]							
0x25		15:8	RESULT[15:8]							
0x26 ... 0x27	Reserved									
0x28	SEQCTRL	7:0	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn

Offset	Name	Bit Pos.								
0x29		15:8	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn
0x2A		23:16	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn
0x2B		31:24	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn
0x2C	CALIB	7:0						BIASCOMP[2:0]		
0x2D		15:8						BIASREFBUF[2:0]		

41.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the section on Synchronization.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization section.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

41.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	SWRST
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the ADC to be enabled or disabled, depending on other peripheral requests.

In On Demand operation mode, i.e., if the ONDEMAND bit has been previously set, the ADC will only be running when requested by a peripheral. If there is no peripheral requesting the ADC will be in a disable state.

If On Demand is disabled the ADC will always be running when enabled.

In standby sleep mode, the On Demand operation is still active if the CTRLA.RUNSTDBY bit is '1'. If CTRLA.RUNSTDBY is '0', the ADC is disabled.

This bit is not synchronized.

Value	Description
0	The ADC is always on , if enabled.
1	The ADC is enabled, when a peripheral is requesting the ADC conversion. The ADC is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the ADC behaves during standby sleep mode.

This bit is not synchronized.

Value	Description
0	The ADC is halted during standby sleep mode.
1	The ADC is not stopped in standby sleep mode. If CTRLA.ONDEMAND=1, the ADC will be running when a peripheral is requesting it. If CTRLA.ONDEMAND=0, the ADC will always be running in standby sleep mode.

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The ADC is disabled.
1	The ADC is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the ADC, except DBGCTRL, to their initial state, and the ADC will be disabled.

Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

41.8.2 Control B

Name: CTRLB
Offset: 0x01 [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
						PRESCALER[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – PRESCALER[2:0]: Prescaler Configuration
This field defines the ADC clock relative to the peripheral clock.

Value	Name	Description
0x0	DIV2	Peripheral clock divided by 2
0x1	DIV4	Peripheral clock divided by 4
0x2	DIV8	Peripheral clock divided by 8
0x3	DIV16	Peripheral clock divided by 16
0x4	DIV32	Peripheral clock divided by 32
0x5	DIV64	Peripheral clock divided by 64
0x6	DIV128	Peripheral clock divided by 128
0x7	DIV256	Peripheral clock divided by 256

41.8.3 Reference Control

Name: REFCTRL
Offset: 0x02 [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	REFCOMP				REFSEL[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – REFCOMP: Reference Buffer Offset Compensation Enable
The gain error can be reduced by enabling the reference buffer offset compensation. This will decrease the input impedance and thus increase the start-up time of the reference.

Value	Description
0	Reference buffer offset compensation is disabled.
1	Reference buffer offset compensation is enabled.

Bits 3:0 – REFSEL[3:0]: Reference Selection
These bits select the reference for the ADC.

Value	Name	Description
0x0	INTREF	internal variable reference voltage
x01	INTVCC0	1/1.6 VDDANA
0x2	INTVCC1	1/2 VDDANA (only for VDDANA > 2.0V)
0x3	VREFA	External reference
0x4	VREFB	External reference
0x5	INTVCC2	VDDANA
0x6 - 0xF		Reserved

41.8.4 Event Control

Name: EVCTRL
Offset: 0x03 [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – WINMONEO: Window Monitor Event Out
This bit indicates whether the Window Monitor event output is enabled or not and an output event will be generated when the window monitor detects something.

Value	Description
0	Window Monitor event output is disabled and an event will not be generated.
1	Window Monitor event output is enabled and an event will be generated.

Bit 4 – RESRDYEO: Result Ready Event Out
This bit indicates whether the Result Ready event output is enabled or not and an output event will be generated when the conversion result is available.

Value	Description
0	Result Ready event output is disabled and an event will not be generated.
1	Result Ready event output is enabled and an event will be generated.

Bit 3 – STARTINV: Start Conversion Event Invert Enable

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 2 – FLUSHINV: Flush Event Invert Enable

Value	Description
0	Flush event input source is not inverted.
1	Flush event input source is inverted.

Bit 1 – STARTEI: Start Conversion Event Input Enable

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

Bit 0 – FLUSHEI: Flush Event Input Enable

Value	Description
0	A flush and new conversion will not be triggered on any incoming event.
1	A flush and new conversion will be triggered on any incoming event.

41.8.5 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x04 [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON: Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Window Monitor Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The window monitor interrupt is disabled.
1	The window monitor interrupt is enabled, and an interrupt request will be generated when the Window Monitor interrupt flag is set.

Bit 1 – OVERRUN: Overrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled, and an interrupt request will be generated when the Overrun interrupt flag is set.

Bit 0 – RESRDY: Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Result Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled, and an interrupt request will be generated when the Result Ready interrupt flag is set.

41.8.6 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x05 [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON: Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Window Monitor Interrupt bit, which enables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 1 – OVERRUN: Overrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overrun Interrupt bit, which enables the Overrun interrupt.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled.

Bit 0 – RESRDY: Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Result Ready Interrupt bit, which enables the Result Ready interrupt.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled.

41.8.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06 [ID-0000120e]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON: Window Monitor
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.

This flag is set on the next GCLK_ADC cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Window Monitor interrupt flag.

Bit 1 – OVERRUN: Overrun
This flag is cleared by writing a '1' to the flag.

This flag is set if RESULT is written before the previous value has been read by CPU, and an interrupt request will be generated if INTENCLR/SET.OVERRUN=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overrun interrupt flag.

Bit 0 – RESRDY: Result Ready
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.

This flag is set when the conversion result is available, and an interrupt will be generated if INTENCLR/SET.RESRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Result Ready interrupt flag.

41.8.8 Sequence Status

Name: SEQSTATUS
Offset: 0x07 [ID-0000120e]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	SEQBUSY			SEQSTATE[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – SEQBUSY: Sequence busy
This bit is set when the sequence start.

This bit is clear when the last conversion in a sequence is done.

Bits 4:0 – SEQSTATE[4:0]: Sequence State
These bit fields are the pointer of sequence. This value identifies the last conversion done in the sequence.

41.8.9 Input Control

Name: INPUTCTRL
Offset: 0x08 [ID-0000120e]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
				MUXNEG[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				MUXPOS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 12:8 – MUXNEG[4:0]: Negative MUX Input Selection
These bits define the MUX selection for the negative ADC input.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin
0x08 - 0x17	-	Reserved
0x18	GND	Internal ground
0x19 - 0x1F	-	Reserved

Bits 4:0 – MUXPOS[4:0]: Positive MUX Input Selection
These bits define the MUX selection for the positive ADC input. If the internal bandgap voltage input channel is selected, then the Sampling Time Length bit group in the Sampling Control register must be written with a corresponding value.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin

Value	Name	Description
0x08	AIN8	ADC AIN8 pin
0x09	AIN9	ADC AIN9 pin
0x0A	AIN10	ADC AIN10 pin
0x0B	AIN11	ADC AIN11 pin
0x0C	AIN12	ADC AIN12 pin
0x0D	AIN13	ADC AIN13 pin
0x0E	AIN14	ADC AIN14 pin
0x0F	AIN15	ADC AIN15 pin
0x10	AIN16	ADC AIN16 pin
0x11	AIN17	ADC AIN17 pin
0x12	AIN18	ADC AIN18 pin
0x13	AIN19	ADC AIN19 pin
0x14 - 0x17	-	Reserved
0x18	-	Reserved
0x19	BANDGAP	Bandgap Voltage
0x1A	SCALEDCOREVCC	1/4 Scaled Core Supply
0x1B	SCALEDIOVCC	1/4 Scaled I/O Supply
0x1C	-	Reserved
0x1D	SCALEDVBAT	1/4 Scaled VBAT Supply
0x1E	-	Reserved
0x1F	-	Reserved

41.8.10 Control C

Name: CTRLC
Offset: 0x0A [ID-0000120e]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						WINMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 10:8 – WINMODE[2:0]: Window Monitor Mode
These bits enable and define the window monitor mode.

Value	Name	Description
0x0	DISABLE	No window mode (default)
0x1	MODE1	RESULT > WINLT
0x2	MODE2	RESULT < WINUT
0x3	MODE3	WINLT < RESULT < WINUT

Value	Name	Description
0x4	MODE4	WINUT < RESULT < WINLT
0x5 - 0x7		Reserved

Bits 5:4 – RESSEL[1:0]: Conversion Result Resolution
These bits define whether the ADC completes the conversion 12-, 10- or 8-bit result resolution.

Value	Name	Description
0x0	12BIT	12-bit result
0x1	16BIT	For averaging mode output
0x2	10BIT	10-bit result
0x3	8BIT	8-bit result

Bit 3 – CORREN: Digital Correction Logic Enabled

Value	Description
0	Disable the digital result correction.
1	Enable the digital result correction. The ADC conversion result in the RESULT register is then corrected for gain and offset based on the values in the GAINCORR and OFFSETCORR registers. Conversion time will be increased by 13 cycles according to the value in the Offset Correction Value bit group in the Offset Correction register.

Bit 2 – FREERUN: Free Running Mode

Value	Description
0	The ADC run in single conversion mode.
1	The ADC is in free running mode and a new conversion will be initiated when a previous conversion completes.

Bit 1 – LEFTADJ: Left-Adjusted Result

Value	Description
0	The ADC conversion result is right-adjusted in the RESULT register.
1	The ADC conversion result is left-adjusted in the RESULT register. The high byte of the 12-bit result will be present in the upper part of the result register. Writing this bit to zero (default) will right-adjust the value in the RESULT register.

Bit 0 – DIFFMODE: Differential Mode

Value	Description
0	The ADC is running in singled-ended mode.
1	The ADC is running in differential mode. In this mode, the voltage difference between the MUXPOS and MUXNEG inputs will be converted by the ADC.

41.8.11 Average Control

Name: AVGCTRL
Offset: 0x0C [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
		ADJRES[2:0]			SAMPLENUM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:4 – ADJRES[2:0]: Adjusting Result / Division Coefficient
These bits define the division coefficient in 2n steps.

Bits 3:0 – SAMPLENUM[3:0]: Number of Samples to be Collected
These bits define how many samples are added together. The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLC.RESSEL must be changed.

Value	Description
0x0	1 sample
0x1	2 samples
0x2	4 samples
0x3	8 samples
0x4	16 samples
0x5	32 samples
0x6	64 samples
0x7	128 samples
0x8	256 samples
0x9	512 samples
0xA	1024 samples
0xB - 0xF	Reserved

41.8.12 Sampling Time Control

Name: SAMPCTRL
Offset: 0x0D [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	OFFCOMP		SAMPLEN[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – OFFCOMP: Comparator Offset Compensation Enable
Setting this bit enables the offset compensation for each sampling period to ensure low offset and immunity to temperature or voltage drift. This compensation increases the sampling time by three clock cycles.

This bit must be set to zero to validate the SAMPLEN value. It's not possible to use OFFCOMP=1 and SAMPLEN>0.

Bits 5:0 – SAMPLEN[5:0]: Sampling Time Length
These bits control the ADC sampling time in number of CLK_ADC cycles, depending of the prescaler value, thus controlling the ADC input impedance. Sampling time is set according to the equation:

$$\text{Sampling time} = (\text{SAMPLEN}+1) \cdot \left(\text{CLK}_{\text{ADC}}\right)$$

41.8.13 Window Monitor Lower Threshold

Name: WINLT
Offset: 0x0E [ID-0000120e]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINLT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINLT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINLT[15:0]: Window Lower Threshold
If the window monitor is enabled, these bits define the lower threshold value.

41.8.14 Window Monitor Upper Threshold

Name: WINUT
Offset: 0x10 [ID-0000120e]
Reset: 0x0000
Property: PAV Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINUT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINUT[15:0]: Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.

41.8.15 Gain Correction

Name: GAINCORR
Offset: 0x12 [ID-0000120e]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					GAINCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – GAINCORR[11:0]: Gain Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for gain error before being written to the result register. The gain correction is a fractional value, a 1-bit integer plus an 11-bit fraction, and therefore ½ <= GAINCORR < 2. GAINCORR values range from 0.10000000000 to 1.11111111111.

41.8.16 Offset Correction

Name: OFFSETCORR
Offset: 0x14 [ID-0000120e]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					OFFSETCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – OFFSETCORR[11:0]: Offset Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for offset error before being written to the Result register. This OFFSETCORR value is in two's complement format.

41.8.17 Software Trigger

Name: SWTRIG
Offset: 0x18 [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							START	FLUSH
Access							W	W
Reset							0	0

Bit 1 – START: ADC Start Conversion
Writing a '1' to this bit will start a conversion or sequence. The bit is cleared by hardware when the conversion has started. Writing a '1' to this bit when it is already set has no effect.

Writing a '0' to this bit will have no effect.

Bit 0 – FLUSH: ADC Conversion Flush
Writing a '1' to this bit will flush the ADC pipeline. A flush will restart the ADC clock on the next peripheral clock edge, and all conversions in progress will be aborted and lost. This bit is cleared until the ADC has been flushed.

After the flush, the ADC will resume where it left off; i.e., if a conversion was pending, the ADC will start a new conversion.

Writing this bit to '0' will have no effect.

41.8.18 Debug Control

Name: DBGCTRL
Offset: 0x1C [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

This bit should be written only while a conversion is not ongoing.

Value	Description
0	The ADC is halted when the CPU is halted by an external debugger.
1	The ADC continues normal operation when the CPU is halted by an external debugger.

41.8.19 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20 [ID-0000120e]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						SWTRIG	OFFSETCORR	GAINCORR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SWTRIG: Software Trigger Synchronization Busy
This bit is cleared when the synchronization of SWTRIG register between the clock domains is complete.

This bit is set when the synchronization of SWTRIG register between clock domains is started.

Bit 9 – OFFSETCORR: Offset Correction Synchronization Busy
This bit is cleared when the synchronization of OFFSETCORR register between the clock domains is complete.

This bit is set when the synchronization of OFFSETCORR register between clock domains is started.

Bit 8 – GAINCORR: Gain Correction Synchronization Busy
This bit is cleared when the synchronization of GAINCORR register between the clock domains is complete.

This bit is set when the synchronization of GAINCORR register between clock domains is started.

Bit 7 – WINUT: Window Monitor Lower Threshold Synchronization Busy
This bit is cleared when the synchronization of WINUT register between the clock domains is complete.

This bit is set when the synchronization of WINUT register between clock domains is started.

Bit 6 – WINLT: Window Monitor Upper Threshold Synchronization Busy
This bit is cleared when the synchronization of WINLT register between the clock domains is complete.

This bit is set when the synchronization of WINLT register between clock domains is started.

Bit 5 – SAMPCTRL: Sampling Time Control Synchronization Busy
This bit is cleared when the synchronization of SAMPCTRL register between the clock domains is complete.

This bit is set when the synchronization of SAMPCTRL register between clock domains is started.

Bit 4 – AVGCTRL: Average Control Synchronization Busy
This bit is cleared when the synchronization of AVGCTRL register between the clock domains is complete.

This bit is set when the synchronization of AVGCTRL register between clock domains is started.

Bit 3 – CTRLC: Control C Synchronization Busy
This bit is cleared when the synchronization of CTRLC register between the clock domains is complete.

This bit is set when the synchronization of CTRLC register between clock domains is started.

Bit 2 – INPUTCTRL: Input Control Synchronization Busy
This bit is cleared when the synchronization of INPUTCTRL register between the clock domains is complete.

This bit is set when the synchronization of INPUTCTRL register between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.

This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.

This bit is set when the synchronization of SWRST register between clock domains is started

41.8.20 Result

Name: RESULT
Offset: 0x24 [ID-0000120e]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	RESULT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RESULT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RESULT[15:0]: Result Conversion Value
These bits will hold up to a 16-bit ADC conversion result, depending on the configuration.

In single conversion mode without averaging, the ADC conversion will produce a 12-bit result, which can be left- or right-shifted, depending on the setting of CTRLC.LEFTADJ.

If the result is left-adjusted (CTRLC.LEFTADJ), the high byte of the result will be in bit position [15:8], while the remaining 4 bits of the result will be placed in bit locations [7:4]. This can be used only if an 8-bit result is needed; i.e., one can read only the high byte of the entire 16-bit register.

If the result is not left-adjusted (CTRLC.LEFTADJ) and no oversampling is used, the result will be available in bit locations [11:0], and the result is then 12 bits long. If oversampling is used, the result will be located in bit locations [15:0], depending on the settings of the Average Control register.

41.8.21 Sequence Control

Name: SEQCTRL
Offset: 0x28 [ID-0000120e]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn	SEQENn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEQENn: Enable Positive Input in the Sequence
For details on available positive mux selection, refer to INPUTCTRL.MUXENG.

The sequence start from the lowest input, and go to the next enabled input automatically when the conversion is done. If no bits are set the sequence is disabled.

Value	Description
0	Disable the positive input mux n selection from the sequence.
1	Enable the positive input mux n selection to the sequence.

41.8.22 Calibration

Name: CALIB
Offset: 0x2C [ID-0000120e]
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
						BIASREFBUF[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						BIASCOMP[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 10:8 – BIASREFBUF[2:0]: Bias Reference Buffer Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.

The value must be copied only, and must not be changed.

Bits 2:0 – BIASCOMP[2:0]: Bias Comparator Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.

The value must be copied only, and must not be changed

42. AC – Analog Comparators

42.1 Overview

The Analog Comparator (AC) supports two individual comparators. Each comparator (COMP) compares the voltage levels on two inputs, and provides a digital output based on this comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change.

Hysteresis and propagation delay can be adjusted to achieve the optimal operation for each application.

The input selection includes four shared analog port pins and several internal signals. Each comparator output state can also be output on a pin for use by external devices.

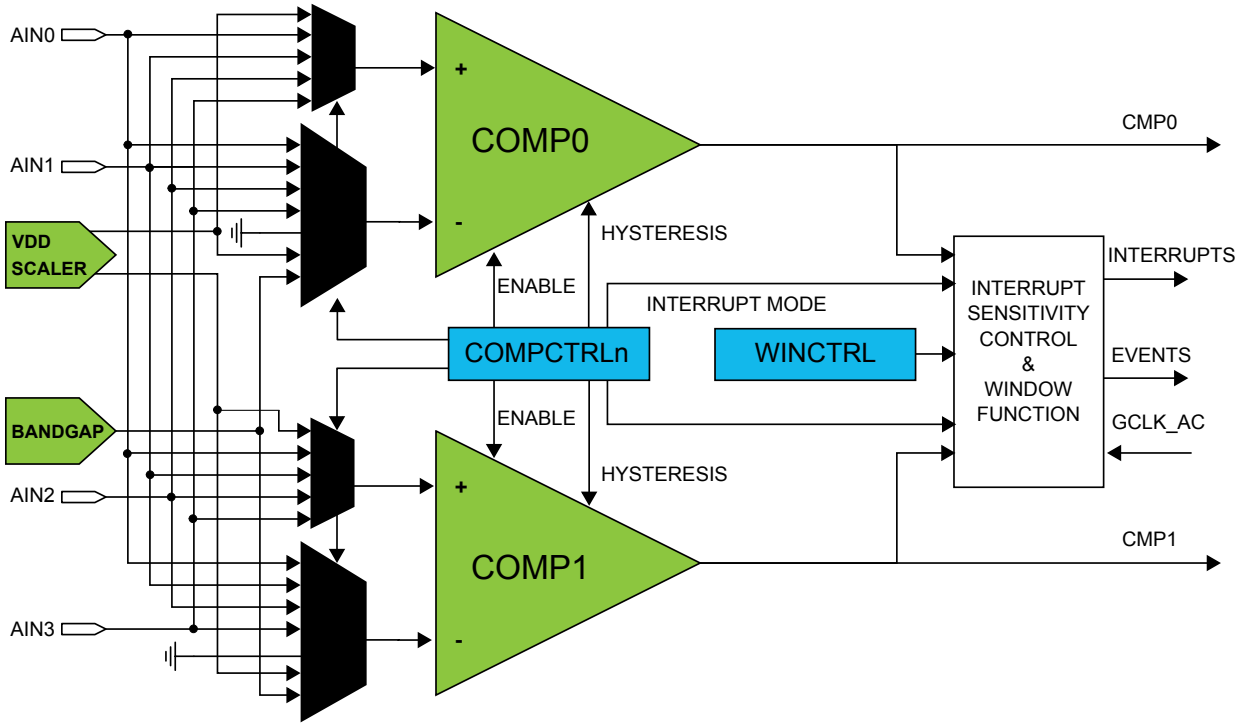
The comparators are always grouped in pairs on each port. The AC peripheral implements one pair of comparators. These are called Comparator 0 (COMP0) and Comparator 1 (COMP1) They have identical behaviors, but separate control registers. The pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.

42.2 Features

- Two individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis: 4-level On, or Off
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Flexible input selection:
  - Four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - 64-level programmable VDD scaler per comparator
- Interrupt generation on:
  - Rising or falling edge
  - Toggle
  - End of comparison
- Window function interrupt generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
  - Signal outside window
- Event generation on:
  - Comparator output
  - Window function inside/outside window
- Optional digital filter on comparator output

42.3 Block Diagram

Figure 42-1. Analog Comparator Block Diagram



42.4 Signal Description

Signal	Description	Type
AIN[3..0]	Analog input	Comparator inputs
CMP[1..0]	Digital output	Comparator outputs

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

42.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

42.5.1 I/O Lines

Using the AC's I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

Related Links

PORT: IO Pin Controller

42.5.2 Power Management

The AC will continue to operate in any sleep mode where the selected source clock is running. The AC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

42.5.3 Clocks

The AC bus clock (CLK_AC_APB) can be enabled and disabled in the Power Manager, and the default state of CLK_AC_APB can be found in the Peripheral Clock Masking section in the Power Manager description.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC. Refer to the Generic Clock Controller chapter for details.

This generic clock is asynchronous to the bus clock (CLK_AC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

PM – Power Manager

42.5.4 DMA

Not applicable.

42.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

42.5.6 Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

EVSYS – Event System

42.5.7 Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any on-going comparison is completed. The AC can be forced to continue normal operation during debugging. Refer to DBGCTRL for details. If the AC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

42.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Control B register (CTRLB)
- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller

42.5.9 Analog Connections

Each comparator has up to four I/O pins that can be used as analog inputs. Each pair of comparators shares the same four pins. These pins must be configured for analog operation before using them as comparator inputs.

Any internal reference source, such as a bandgap voltage reference, must be configured and enabled prior to its use as a comparator input.

42.6 Functional Description

42.6.1 Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as a bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive, and '0' otherwise.

The individual comparators can be used independently (normal mode) or paired to form a window comparison (window mode).

42.6.2 Basic Operation

42.6.2.1 Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected:

- Event Control register (EVCTRL)

Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

42.6.2.2 Enabling, Disabling and Resetting

The AC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The AC is disabled writing a '0' to CTRLA.ENABLE.

The AC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC will be reset to their initial state, and the AC will be disabled. Refer to *CTRLA* for details.

42.6.2.3 Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See Starting a Comparison for more details.
- Select the desired hysteresis with COMPCTRLx.HYSTEN and COMPCTRLx.HYST. See Input Hysteresis for more details.
- Select the comparator speed versus power with COMPCTRLx.SPEED. See Propagation Delay vs. Power Consumption for more details.

- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See Selecting Comparator Inputs for more details.
- Select the filtering option with COMPCTRLx.FLEN.
- Select standby operation with Run in Standby bit (COMPCTRLx.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLx.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLx.ENABLE bits.

42.6.2.4 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLx.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in *Electrical Characteristics*. During the start-up time, the COMP output is not available.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the single-shot mode to chain further events in the system, regardless of the state of the comparator outputs. The interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state, regardless of whether the interrupt is enabled or not.

Related Links

Electrical Characteristics

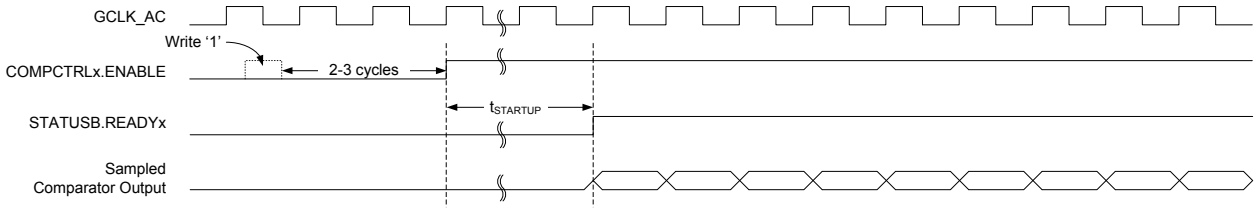
Continuous Measurement

Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the Figure 42-2.

Figure 42-2. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep modes without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the Power Manager will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is then disabled again automatically, unless configured to wake up the system from sleep.

Related Links

Electrical Characteristics

Single-Shot

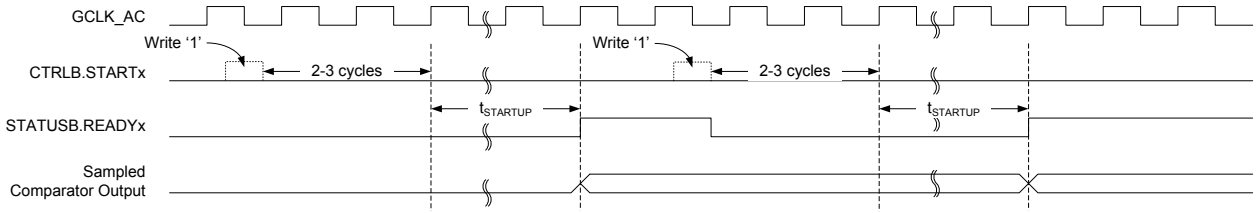
Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, and after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in Figure 42-3.

Figure 42-3. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the startup time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake up the system from sleep.

Related Links

Electrical Characteristics

42.6.3 Selecting Comparator Inputs

Each comparator has one positive and one negative input. The positive input is one of the external input pins (AINx). The negative input can be fed either from an external input pin (AINx) or from one of the several internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS)

- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG)

In the case of using an external I/O pin, the selected pin must be configured for analog use in the PORT Controller by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

42.6.4 Window Operation

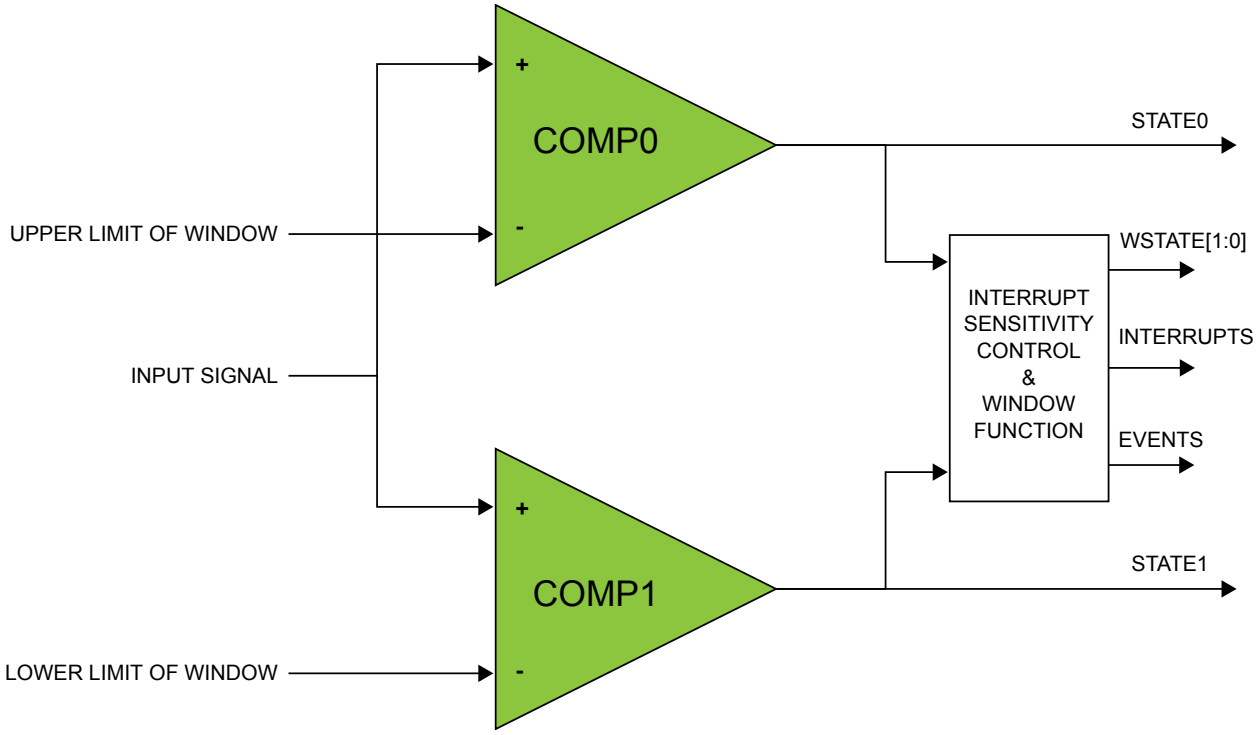
Each comparator pair can be configured to work together in window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators in a pair must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

To physically configure the pair of comparators for window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In Figure 42-4, COMP0 defines the upper limit and COMP1 defines the lower limit of the window, as shown but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during window mode.

When the comparators are configured for window mode and single-shot mode, measurements are performed simultaneously on both comparators. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise either peripheral event can start a measurement.

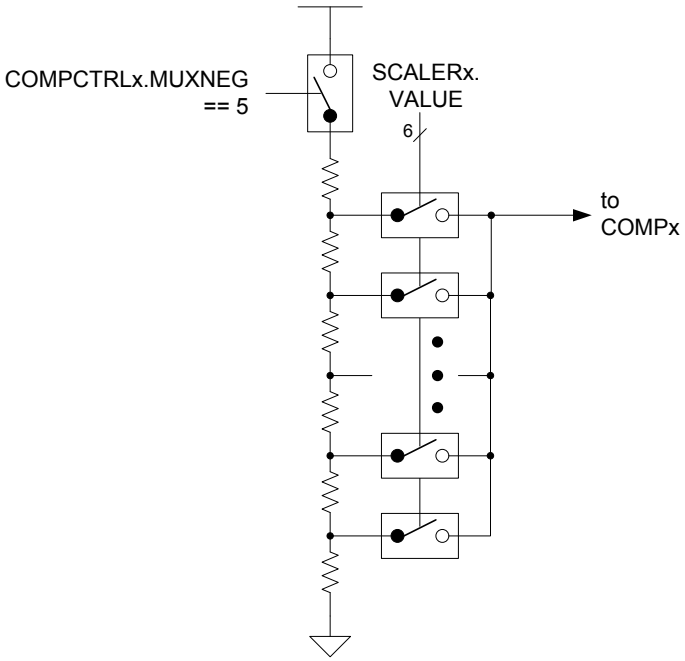
Figure 42-4. Comparators in Window Mode



42.6.5 VDD Scaler

The VDD scaler generates a reference voltage that is a fraction of the device's supply voltage, with 64 levels. One independent voltage channel is dedicated for each comparator. The scaler of a comparator is enabled when the Negative Input Mux bit field in the respective Comparator Control register (COMPCTRLx.MUXNEG) is set to 0x5 and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the Scaler x registers (SCALERx.VALUE).

Figure 42-5. VDD Scaler



42.6.6 Input Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other.

Hysteresis is enabled for each comparator individually by the Hysteresis Enable bit in the Comparator x Control register (COMPCTRLx.HYSTEN). Furthermore, when enabled, the level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLx.HYST). Hysteresis is available only in continuous mode (COMPCTRLx.SINGLE=0).

42.6.7 Propagation Delay vs. Power Consumption

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit group in the Comparator x Control register (COMPCTRLx.SPEED). The Speed bits select the amount of bias current provided to the comparator, and as such will also affect the start-up time.

42.6.8 Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 42-6. For single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 42-7.

Figure 42-6. Continuous Mode Filtering

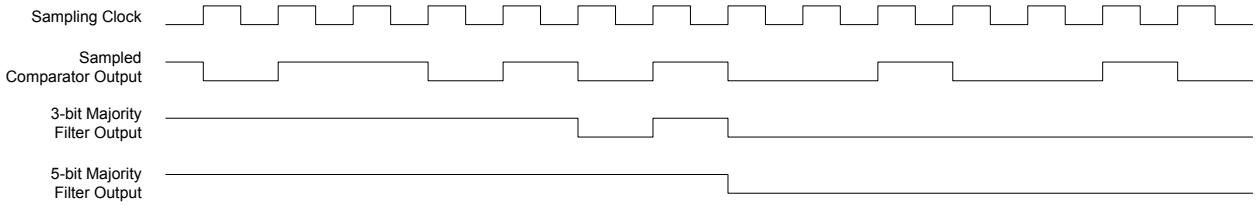
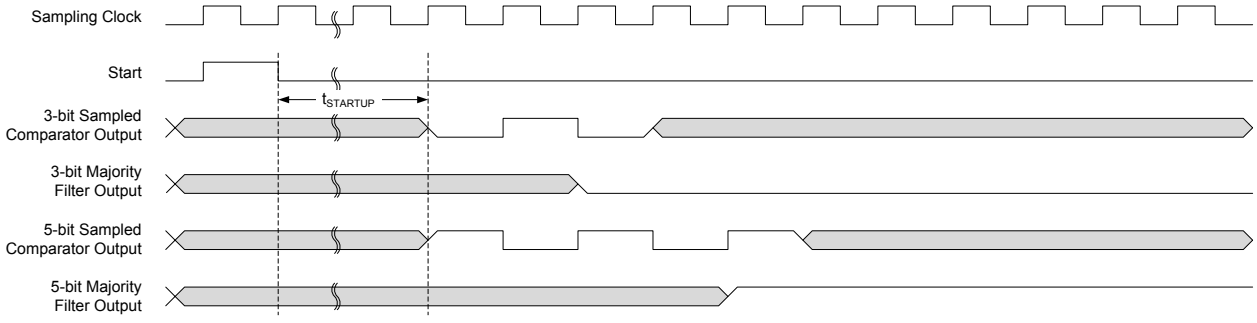


Figure 42-7. Single-Shot Filtering



During sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during sleep modes, or the resulting interrupt/event may be generated incorrectly.

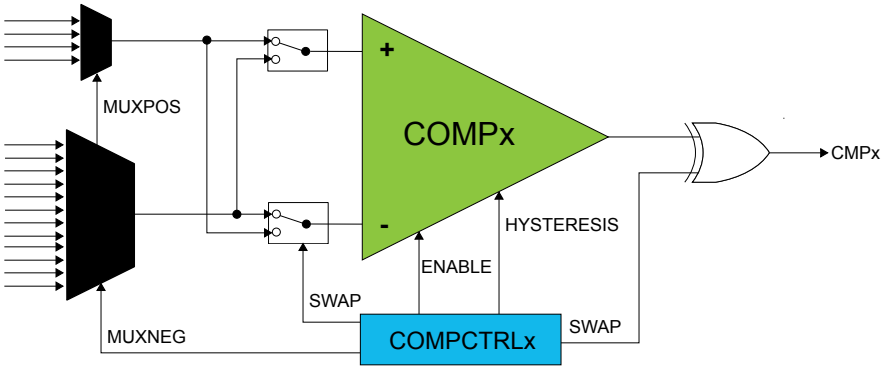
42.6.9 Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

42.6.10 Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLx.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in Figure 42-8. This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLx.SWAP can be changed only while the comparator is disabled.

Figure 42-8. Input Swapping for Offset Compensation



42.6.11 DMA Operation

Not applicable.

42.6.12 Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1): Indicates a change in comparator status.
- Window (WIN0): Indicates a change in the window status.

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRLx.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSEL[1:0]).

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the AC is reset. See INFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller

42.6.13 Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event:

- Start comparison (START0, START1): Start a comparison.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. Refer to the Event System chapter for details on configuring the event system.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

42.6.14 Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLx.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in Table 42-1.

Table 42-1. Sleep Mode Operation

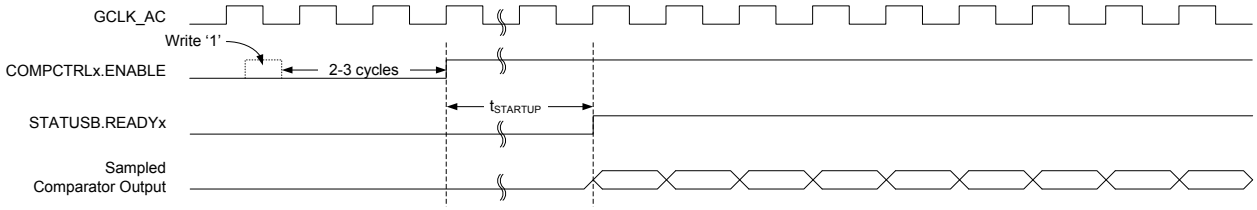
COMPCTRLx.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	COMPx disabled	GCLK_AC stopped, COMPx enabled
1 (Single-shot)	COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

42.6.14.1 Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and GCLK_AC is disabled during sleep, the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device;

otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

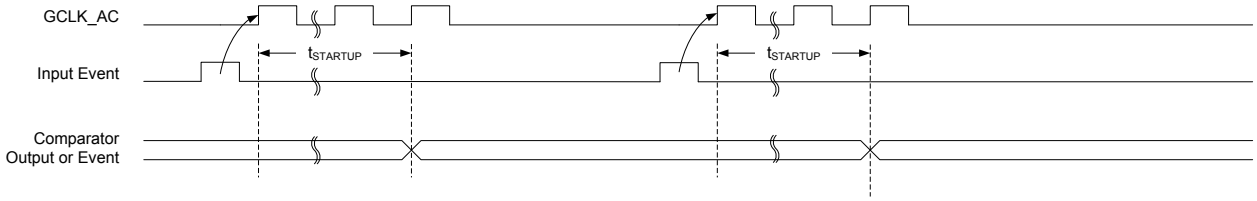
Figure 42-9. Continuous Mode SleepWalking



42.6.14.2 Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in Figure 42-10. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 42-10. Single-Shot SleepWalking



42.6.15 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)
- Enable bit in Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- Window Control register (WINCTRL)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization

42.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0							STARTx	STARTx
0x02	EVCTRL	7:0				WINEO0			COMPEOx	COMPEOx
0x03		15:8			INVEIx	INVEIx			COMPEIx	COMPEIx
0x04	INTENCLR	7:0				WIN0			COMPx	COMPx
0x05	INTENSET	7:0				WIN0			COMPx	COMPx
0x06	INTFLAG	7:0				WIN0			COMPx	COMPx
0x07	STATUSA	7:0			WSTATE0[1:0]				STATEx	STATEx
0x08	STATUSB	7:0							READYx	READYx
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0						WINTSEL0[1:0]		WEN0
0x0B	Reserved									
0x0C	SCALER0	7:0			VALUE[5:0]					
0x0D	SCALER1	7:0			VALUE[5:0]					
0x0E ... 0x0F	Reserved									
0x10	COMPCTRL0	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x11		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x12		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
0x13		31:24			OUT[1:0]			FLEN[2:0]		
0x14	COMPCTRL1	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x15		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x16		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
0x17		31:24			OUT[1:0]			FLEN[2:0]		
0x18 ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0				COMPCTRLx	COMPCTRLx	WINCTRL	ENABLE	SWRST
0x21		15:8								
0x22		23:16								
0x23		31:24								

42.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

42.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000fbb]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	W
Reset							0	0

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

Value	Description
0	The AC is disabled.
1	The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE).

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the AC to their initial state, and the AC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

42.8.2 Control B

Name: CTRLB
Offset: 0x01 [ID-00000fbb]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							STARTx	STARTx
Access							R/W	R/W
Reset							0	0

Bits 1,0 – STARTx: Comparator x Start Comparison
Writing a '0' to this field has no effect.

Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If comparator x is not implemented, or if it is not enabled in single-shot mode, Writing a '1' has no effect.

This bit always reads as zero.

42.8.3 Event Control

Name: EVCTRL
Offset: 0x02 [ID-00000fbb]
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			INVEIx	INVEIx			COMPEIx	COMPEIx
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
				WINEO0			COMPEOx	COMPEOx
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 13,12 – INVEIx: Inverted Event Input Enable x

Value	Description
0	Incoming event is not inverted for comparator x.
1	Incoming event is inverted for comparator x.

Bits 9,8 – COMPEIx: Comparator x Event Input
Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.

These bits indicate whether a comparison will start or not on any incoming event.

Value	Description
0	Comparison will not start on any incoming event.
1	Comparison will start on any incoming event.

Bit 4 – WINEO0: Window 0 Event Output Enable
These bits indicate whether the window 0 function can generate a peripheral event or not.

Value	Description
0	Window 0 Event is disabled.
1	Window 0 Event is enabled.

Bits 1,0 – COMPEOx: Comparator x Event Output Enable
These bits indicate whether the comparator x output can generate a peripheral event or not.

Value	Description
0	COMPx event generation is disabled.
1	COMPx event generation is enabled.

42.8.4 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x04 [ID-00000fbb]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				WIN0			COMPx	COMPx
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0: Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit disables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 1,0 – COMPx: Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit disables the Comparator x interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

42.8.5 Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x05 [ID-00000fbb]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				WIN0			COMPx	COMPx
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0: Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit enables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 1,0 – COMPx: Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Ready interrupt bit and enable the Ready interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

42.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06 [ID-00000fbb]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
				WIN0			COMPx	COMPx
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0: Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and will generate an interrupt if INTENCLR/SET.WINx is also one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Window 0 interrupt flag.

Bits 1,0 – COMPx: Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as zero.

This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLx.INTSEL) and will generate an interrupt if INTENCLR/SET.COMPx is also one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Comparator x interrupt flag.

42.8.7 Status A

Name: STATUSA
Offset: 0x07 [ID-00000fbb]
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
			WSTATE0[1:0]				STATEx	STATEx
Access			R	R			R	R
Reset			0	0			0	0

Bits 5:4 – WSTATE0[1:0]: Window 0 Current State
These bits show the current state of the signal if the window 0 mode is enabled.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3		Reserved

Bits 1,0 – STATEx: Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when STATUSB.READYx is one.

42.8.8 Status B

Name: STATUSB
Offset: 0x08 [ID-00000fbb]
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
							READYx	READYx
Access							R	R
Reset							0	0

Bits 1,0 – READYx: Comparator x Ready
This bit is cleared when the comparator x output is not ready.
This bit is set when the comparator x output is ready.

42.8.9 Debug Control

Name: DBGCTRL
Offset: 0x09 [ID-00000fbb]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The AC is halted when the CPU is halted by an external debugger. Any on-going comparison will complete.
1	The AC continues normal operation when the CPU is halted by an external debugger.

42.8.10 Window Control

Name: WINCTRL
Offset: 0x0A [ID-00000fbb]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						WINTSEL0[1:0]		WEN0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – WINTSEL0[1:0]: Window 0 Interrupt Selection
These bits configure the interrupt mode for the comparator window 0 mode.

Value	Name	Description
0x0	ABOVE	Interrupt on signal above window
0x1	INSIDE	Interrupt on signal inside window
0x2	BELOW	Interrupt on signal below window
0x3	OUTSIDE	Interrupt on signal outside window

Bit 0 – WEN0: Window 0 Mode Enable

Value	Description
0	Window mode is disabled for comparators 0 and 1.
1	Window mode is enabled for comparators 0 and 1.

42.8.11 Scaler n

Name: SCALER
Offset: 0x0C + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			VALUE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – VALUE[5:0]: Scaler Value
These bits define the scaling factor for channel n of the V_{DD} voltage scaler. The output voltage, V_{SCALE}, is:

$$V_{SCALE} = \frac{V_{DD} \cdot (VALUE+1)}{64}$$

42.8.12 Comparator Control n

Name: COMPCTRL
Offset: 0x10 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
			OUT[1:0]			FLEN[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
			HYST[1:0]		HYSTEN		SPEED[1:0]	
Access			R/W	R/W	R/W		R/W	R/W
Reset			0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
Access		R/W		R/W	R/W	R/W	R/W	
Reset		0		0	0	0	0	

Bits 29:28 – OUT[1:0]: Output
These bits configure the output selection for comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is zero.

Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	The output of COMPn is not routed to the COMPn I/O port
0x1	ASYNC	The asynchronous output of COMPn is routed to the COMPn I/O port
0x2	SYNC	The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port
0x3	N/A	Reserved

Bits 26:24 – FLEN[2:0]: Filter Length
These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	No filtering
0x1	MAJ3	3-bit majority function (2 of 3)
0x2	MAJ5	5-bit majority function (3 of 5)
0x3-0x7	N/A	Reserved

Bits 21:20 – HYST[1:0]: Hysteresis Level
These bits indicate the hysteresis level of comparator n when hysteresis is enabled (COMPCTRLn.HYSTEN=1). Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	HYST50	50mV
0x1	HYST70	70mV
0x2	HYST90	90mV
0x3	HYST110	110mV

Bit 19 – HYSTEN: Hysteresis Enable
This bit indicates the hysteresis mode of comparator n. Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.

This bit is not synchronized.

Value	Description
0	Hysteresis is disabled.
1	Hysteresis is enabled.

Bits 17:16 – SPEED[1:0]: Speed Selection
This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	LOW	Low speed
0x1	MEDLOW	Medium low speed
0x2	MEDHIGH	Medium high speed
0x3	HIGH	High speed

Bit 15 – SWAP: Swap Inputs and Invert
This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Description
0	The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input.
1	The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input.

Bits 14:12 – MUXPOS[2:0]: Positive Input Mux Selection
These bits select which input will be connected to the positive input of comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	VSCALE	VDD scaler
0x5–0x7	-	Reserved

Bits 10:8 – MUXNEG[2:0]: Negative Input Mux Selection
These bits select which input will be connected to the negative input of comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	GND	Ground
0x5	VSCALE	VDD scaler
0x6	BANDGAP	Internal bandgap voltage
0x7	-	Reserved

Bit 6 – RUNSTDBY: Run in Standby
This bit controls the behavior of the comparator during standby sleep mode.

This bit is not synchronized

Value	Description
0	The comparator is disabled during sleep.
1	The comparator continues to operate during sleep.

Bits 4:3 – INTSEL[1:0]: Interrupt Selection
These bits select the condition for comparator n to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	TOGGLE	Interrupt on comparator output toggle
0x1	RISING	Interrupt on comparator output rising
0x2	FALLING	Interrupt on comparator output falling
0x3	EOC	Interrupt on end of comparison (single-shot mode only)

Bit 2 – SINGLE: Single-Shot Mode
This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Description
0	Comparator n operates in continuous measurement mode.
1	Comparator n operates in single-shot mode.

Bit 1 – ENABLE: Enable
Writing a zero to this bit disables comparator n.
Writing a one to this bit enables comparator n.

Due to synchronization, there is delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE will read back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.

Writing a one to COMPCTRLn.ENABLE will prevent further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to zero and the write is synchronized.

42.8.13 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20 [ID-00000fbb]
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				COMPCTRLx	COMPCTRLx	WINCTRL	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 4,3 – COMPCTRLx: COMPCTRLx Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLx register between the clock domains is complete.

This bit is set when the synchronization of the COMPCTRLx register between clock domains is started.

Bit 2 – WINCTRL: WINCTRL Synchronization Busy
This bit is cleared when the synchronization of the WINCTRL register between the clock domains is complete.

This bit is set when the synchronization of the WINCTRL register between clock domains is started.

Bit 1 – ENABLE: Enable Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

Bit 0 – SWRST: Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.

43. SLCD - Segment Liquid Crystal Display Controller

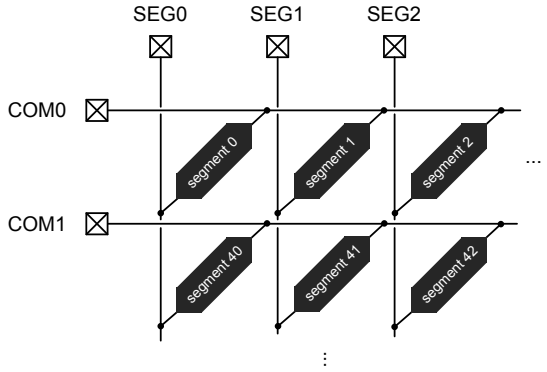
43.1 Overview

An LCD display is made of several segments (pixels or complete symbols) which can be visible or invisible. A segment has two electrodes with liquid crystal between them. These electrodes are the common terminal (COM pin) and the segment terminal (SEG pin). When a voltage above a threshold voltage is applied across the liquid crystal, the segment becomes visible.

The LCD controller is intended for monochrome passive liquid crystal display (LCD) with up to 8 common terminals and up to 44 segment terminals. A charge pump provides LCD display supply which can be higher than supply voltage of the device. Each LCD pin, segment or common terminals, can be configured as general purpose I/O pins if not driven by LCD controller.

Several features such as character mapping, automated characters string display, autonomous animation are implemented to reduce CPU load and power consumption.

Figure 43-1. LCD Panel - Segment/Common Terminals Connections



Note: In order to avoid degradation due to electrophoresis in the liquid crystal, the waveform of the voltage across a segment must not have a DC component.

43.2 Features

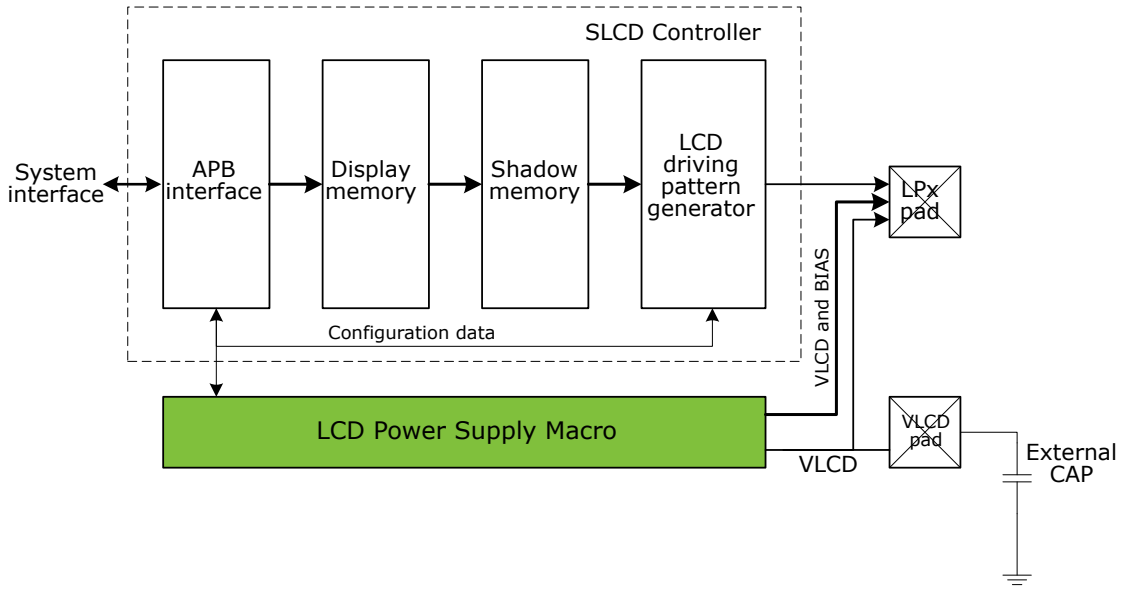
- Display capacity up to 44 segments lines and up to 8 common lines (up to 320 segments)
- Selection of maximum 48 segment/common lines from 52 LCD pins
- Support from static up to 1/8 duty
- Support static, 1/2, 1/3 and 1/4 bias
- Shadow display memory for full freedom in segment update
- LCD pins as segment terminal, common terminal or general purpose I/O
- Blink mode and frequency configuration (up to 16 segments)
- Regular and low power waveform
- Software contrast adjustment control
- On-chip LCD power supply
- Character mapping
- Automated characters string scrolling
- Automated characters string display
- Automated segments display

© 2017 Microchip Technology Inc.

- Autonomous animation (up to 16 segments)
- Three frame counters for timing schedule
- Interrupt request and event output for display update
- DMA Support
- SleepWalking Support

43.3 Block Diagram

Figure 43-2. SLCD Block Diagram



43.4 Signal Description

Signal	Description	Type
LPx	LCD Pin x (COM or SEG terminal)	Analog output
VLCD	LCD Voltage	Analog input or output

One signal can be mapped to one of several pins.

Related Links

I/O Multiplexing and Considerations

43.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

43.5.1 I/O Lines

Using the SLCD I/O lines requires the I/O pins to be configured.

Related Links

PORT: IO Pin Controller

43.5.2 Power Management

The SLCD will continue to operate in any sleep mode where the selected source clock is running. The SLCD interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

43.5.3 Clocks

The SLCD bus clock (CLK_SLCD_APB) can be enabled and disabled in the Main Clock module MCLK, and the default state of CLK_SLCD_APB can be found in Peripheral Clock Masking section.

A 32.768kHz oscillator clock (CLK_SLCD_OSC) is required to clock the SLCD. This clock must be configured and enabled in the 32KHz oscillator controller (OSC32KCTRL) before using the SLCD.

This oscillator clock is asynchronous to the bus clock (CLK_SLCD_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

Related Links

Peripheral Clock Masking

OSC32KCTRL – 32KHz Oscillators Controller

SLCD Clock Selection

Synchronous and Asynchronous Clocks

43.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). Using the SLCD DMA requests requires the DMA Controller to be configured first.

Related Links

DMAC – Direct Memory Access Controller

43.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the SLCD interrupt(s) requires the interrupt controller to be configured first.

43.5.6 Events

The events are connected to the Event System.

Related Links

EVSYS – Event System

43.5.7 Debug Operation

When the CPU is halted in debug mode the SLCD continues normal operation. If the SLCD is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

43.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Segments Data Low and High for COMx Line (SDATAL/Hx)
- Indirect Segments Data Access (ISDATA)

- Character Mapping Data (CMDATA)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

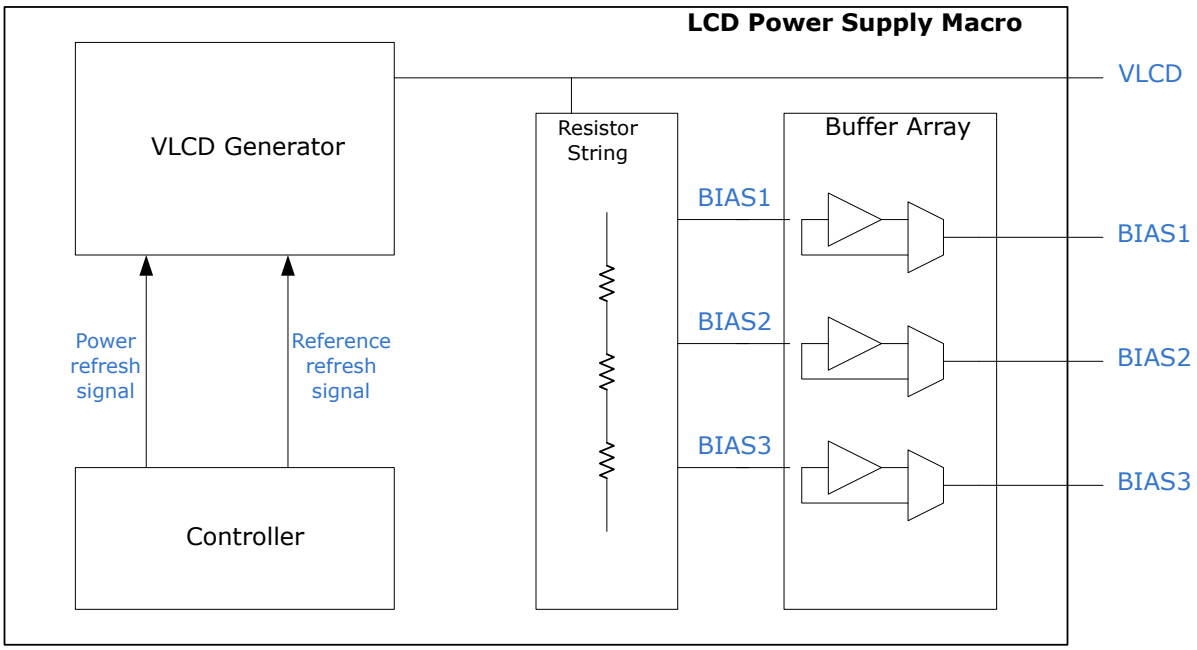
Related Links

PAC - Peripheral Access Controller

43.5.9 Analog Connections

The LCD Power Supply Macro provides voltages (VDDLCD, BIAS1, BIAS2, BIAS3) for the LCD pads. Before enabling the peripheral, it must be configured correctly.

Figure 43-3. LCD Power Supply Macro Functional Block Diagram



43.6 Functional Description

43.6.1 Basic Operation

43.6.1.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the SLCD is disabled (CTRLA.ENABLE is zero):

- All bits in CTRLA register except SWRST and ENABLE

The following registers are enable-protected:

- Event Control register (EVCTRL)
- LCD Pins Enable registers (LPENL and LPENH)
- Blink Configuration register (BCFG)
- Circular Shift Register Configuration register (CSRCFG)
- Automated Character Mapping Configuration register (ACMCFG)
- Automated Bit Mapping Configuration register (ABMCFG)

The following registers are also enable-protected according their corresponding enable bit:

- Frame Counter 0 register (FC0) enabled-protected by Frame Counter 0 Enable (CTRLD.FC0EN)
- Frame Counter 1 register (FC1) enabled-protected by Frame Counter 1 Enable (CTRLD.FC1EN)
- Frame Counter 2 register (FC2) enabled-protected by Frame Counter 2 Enable (CTRLD.FC2EN)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

43.6.1.2 Enabling, Disabling, and Resetting

The SLCD is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The SLCD is disabled by writing a '0' to CTRLA.ENABLE. To stop driving the LCD panel properly, SLCD will be disabled after the current frame is completed. The Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will thus be cleared at the end of the frame.

The SLCD is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the SLCD will be reset to their initial state, and the SLCD will be disabled.

Software reset will stop waveforms generation of the current frame. For this reason it is recommended to disable SLCD first.

43.6.1.3 LCD Display

The display memory stores the values of all segments to display. The display memory is accessible through APB, and should be filled before the next frame starts. A start of a new frame triggers copying the display memory into the shadow display memory. A display memory refresh is thus possible without affecting data already sent to the panel.
Note: The display memory is not initialized at startup.

When a bit in the display memory is written to '1', the corresponding segment will be energized (ON / opaque), and de-energized (OFF / transparent) when this bit is written to '0'.

Each COM signal has identical waveforms but different phases. The maximum amplitude (VLCD) occurs during the corresponding phase of the frame (phase 0 for COM0, phase 1 for COM1 etc.). Otherwise, the signal amplitude is one of the bias voltages (depending on the bias setting).

The SEG lines are controlled according to the corresponding value in shadow display memory. For each phase of the frame, SEG lines are driven to VLCD and GND when the pixel is ON, or to one of the bias voltages when the pixel is OFF.

43.6.1.4 Operating Modes

SLCD supports up to eight COM and up to three bias voltages. The multiplexing and bias configurations are set independently by writing to the Duty Ratio bits and to the Bias Setting bits in Control A register (CTRLA.DUTY and CTRLA.BIAS, respectively). CTRLA.DUTY defines the number of COM lines used (NB_COM), as shown in the table below, and thus the number of phases.

Table 43-1. Duty Selection (CTRLA.DUTY)

DUTY[2:0]	Duty (NB_COM)	Recommended Bias	COM Pins
0x0	1	Static	COM0
0x1	2	1/2	COM0..1
0x2	3	1/3	COM0..2
0x3	4	1/3	COM0..3

DUTY[2:0]	Duty (NB_COM)	Recommended Bias	COM Pins
0x4	6	1/3	COM0..5
0x5	8	1/4	COM0..7

The CTRLA.BIAS[1:0] bit field defines the number of bias voltage:

Table 43-2. Bias Selection (CTRLA.BIAS)

BIAS[1:0]	Description
00	Static
01	1/2
10	1/3
11	1/4

The LCD controller will drive different waveforms according to the different bias configurations. The LCD controller supports two types of driving waveform, bit-inversion (type A, standard) or frame-inversion (type B, low-power). The frame-inversion mode has a lower switching frequency than the bit-inversion mode, and thus reduced power consumption comparing to standard waveform. Both waveform modes have the same period, and their DC-component is null.

By default the low-power waveforms mode is enabled. To select frame-inversion mode, write a zero in the Waveform Mode bit in Control A register (CTRLA.WMOD). To select bit-inversion mode, write a '1' to CTRLA.WMOD.

The following section illustrates how the SLCD controller drives different LCD panels.

For static bias, standard and low-power waveform are identical. However, for 1/2, 1/3 and 1/4 bias, low-power waveform provides a lower toggle rate comparing to the standard waveform.

Figure 43-4. Static Bias Waveform template

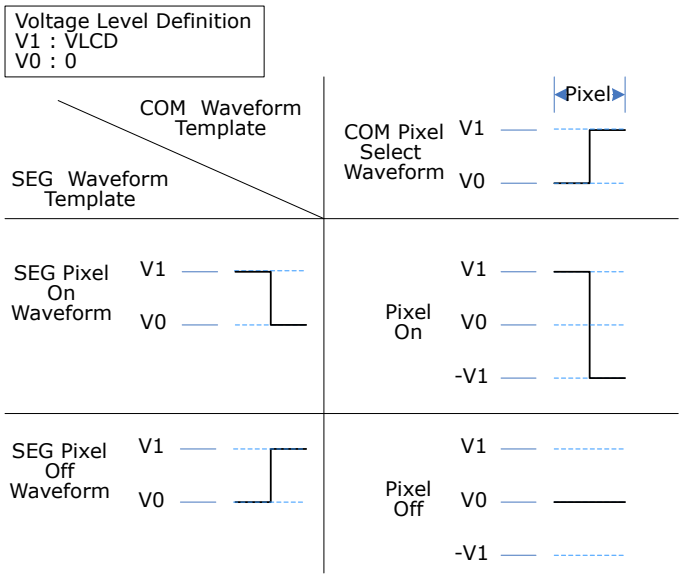


Figure 43-5. Static Bias and Duty Waveform

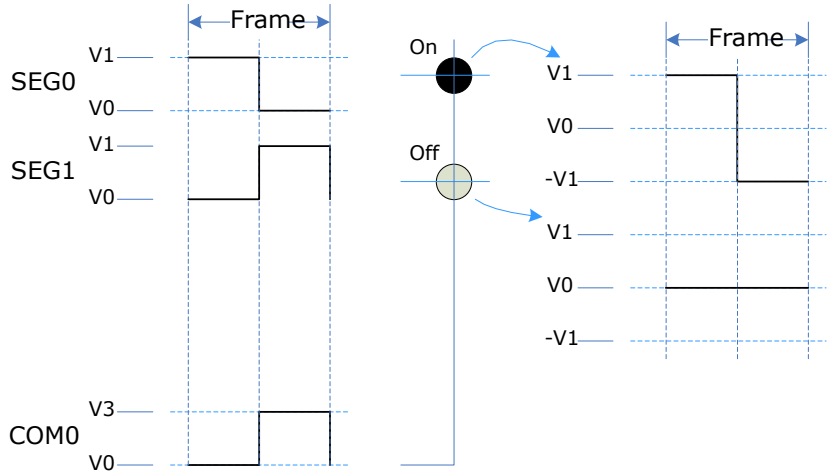


Figure 43-6. 1/2 Bias Waveform Template

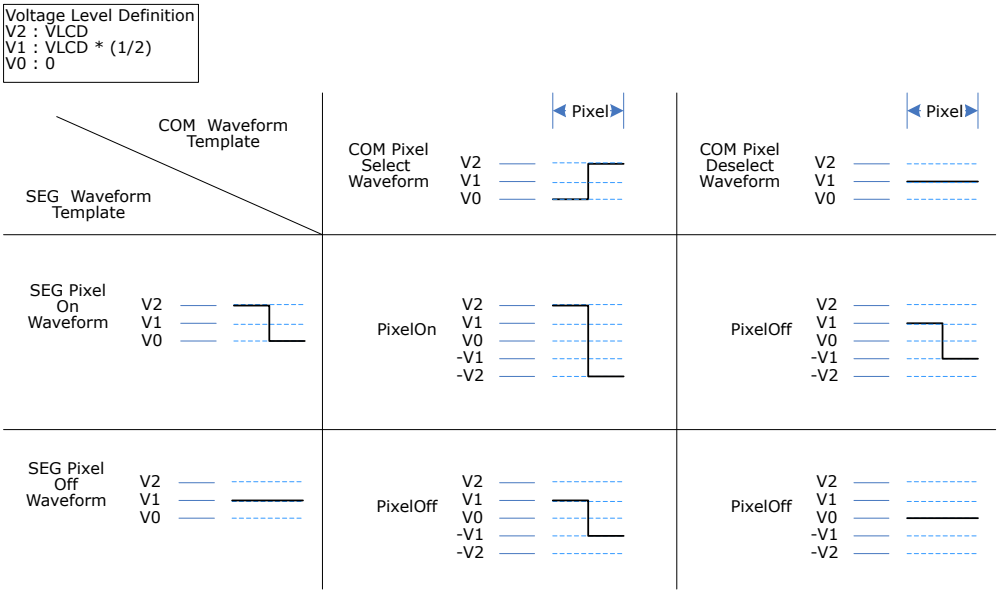
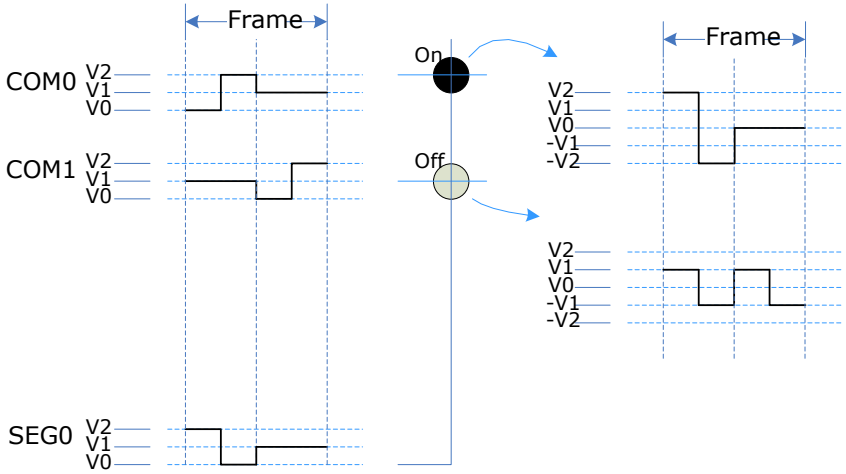


Figure 43-7. 1/2 bias, 1/2 Duty Standard Waveform



The low-power waveform template can be derived from the standard one by separating the standard waveform template into two subframe templates, as shown in the following figure:

Figure 43-8. Derive Low-Power Waveform Template from Standard Waveform Template

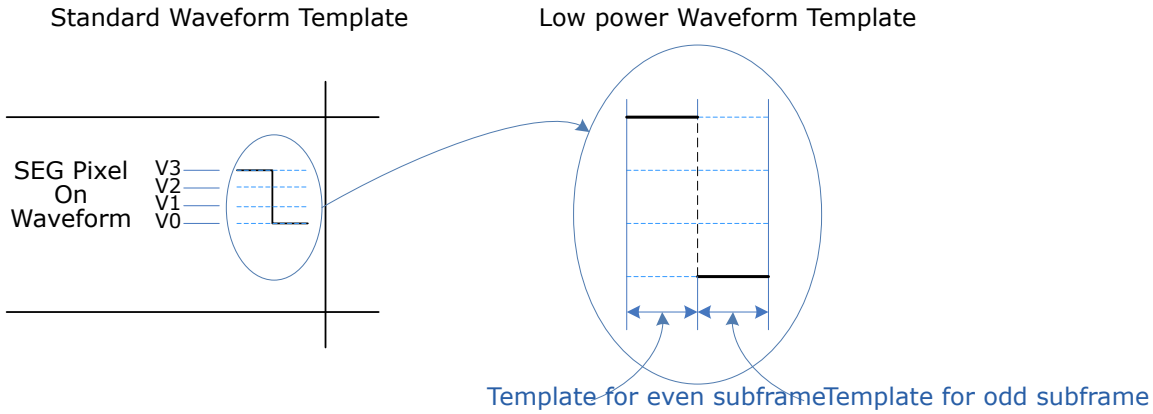


Figure 43-9. 1/2 Bias, 1/2 Duty Low-Power Waveform

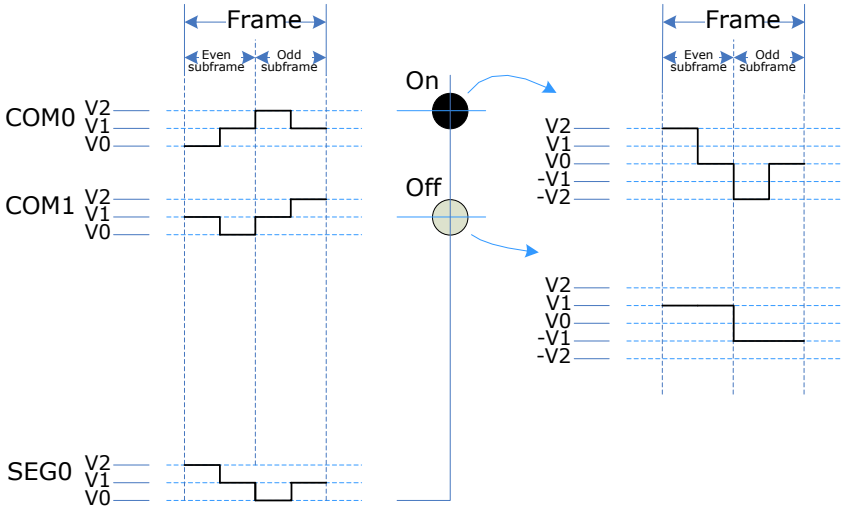


Figure 43-10. 1/3 Bias Standard Driving Waveform Template and Combination

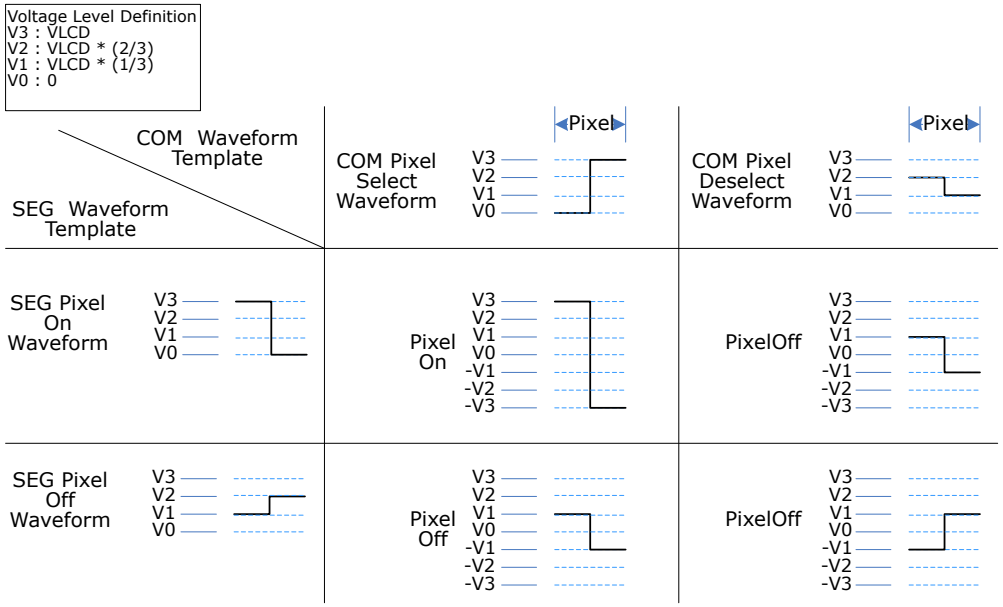


Figure 43-11. 1/3 Bias, 1/4 Duty, Standard Waveform

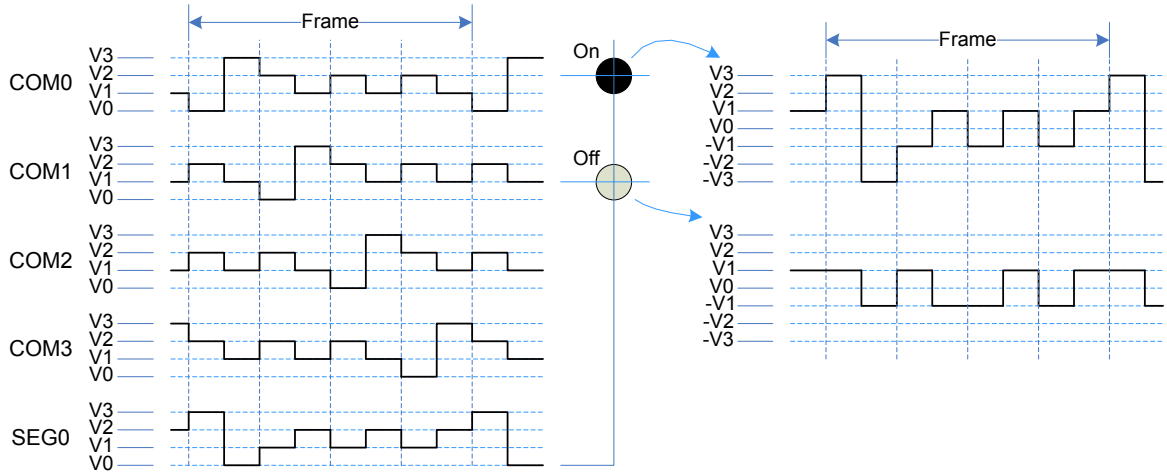


Figure 43-12. 1/3 Bias, 1/4 Duty, Low-Power Waveform

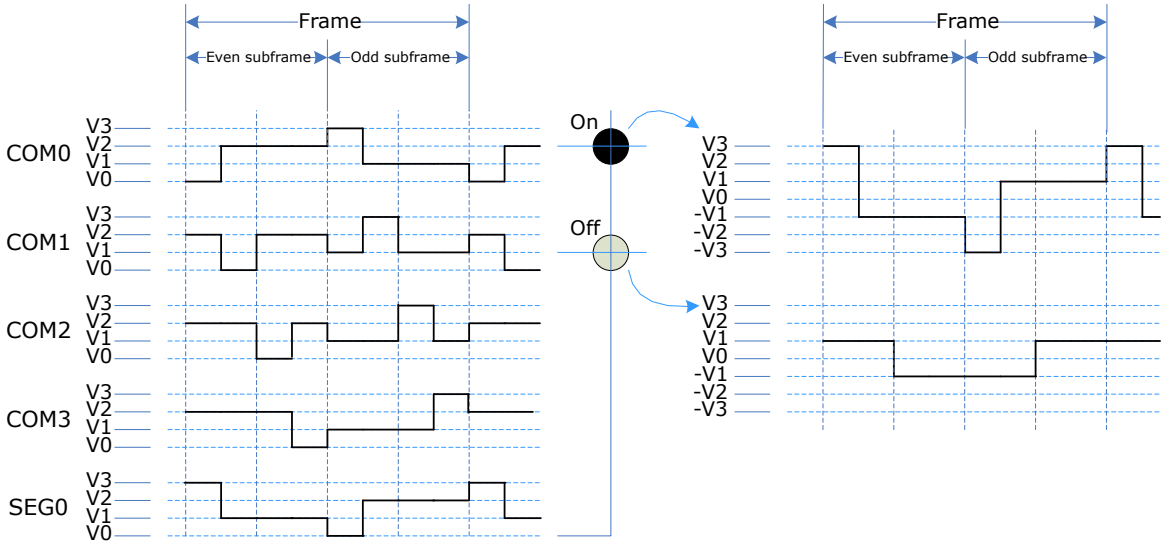


Figure 43-13. 1/4 Bias Standard Driving Waveform Template and Combination

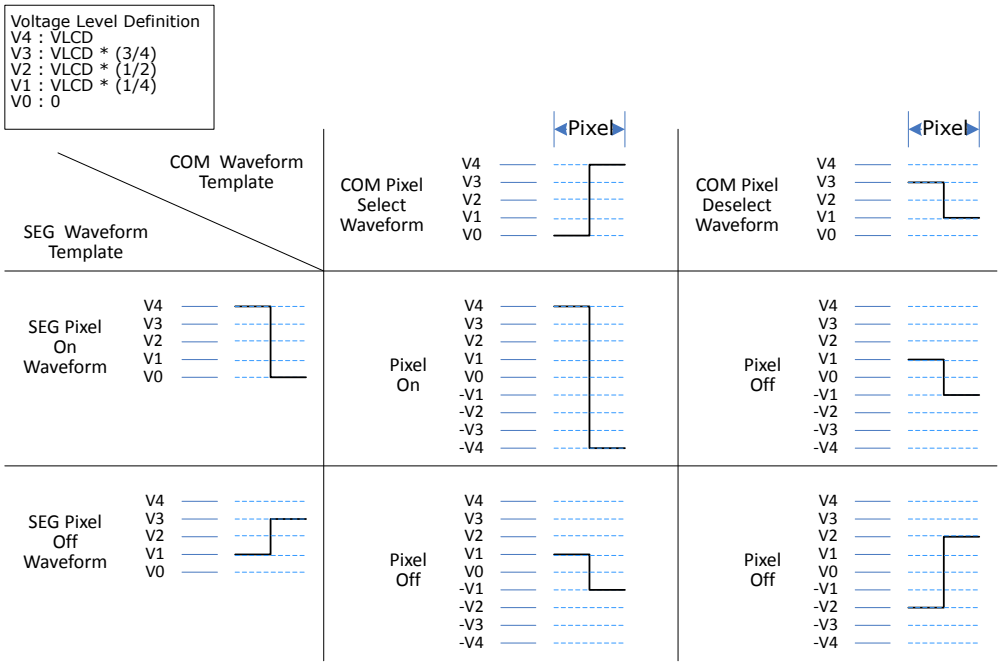


Figure 43-14. 1/4 Bias, 1/8 Duty, Standard Waveform

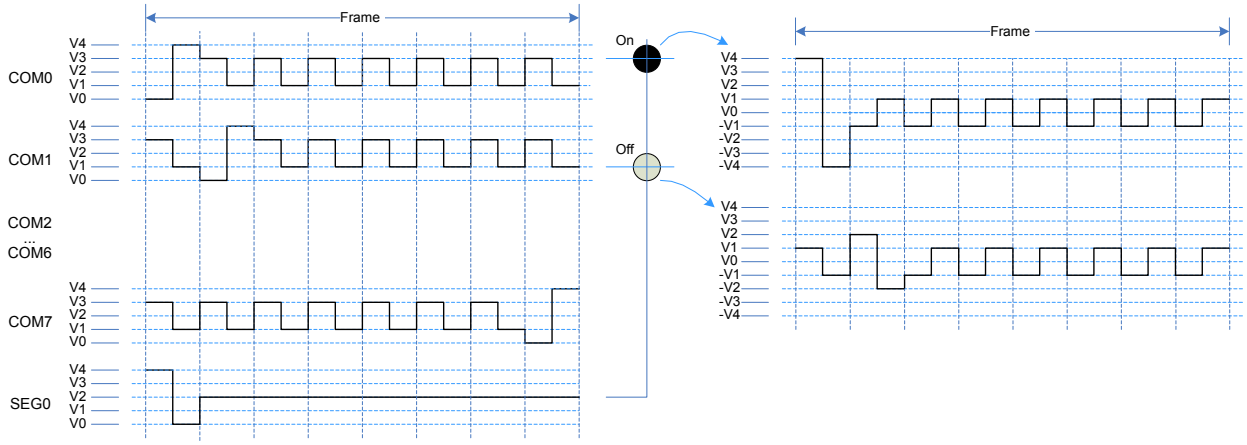
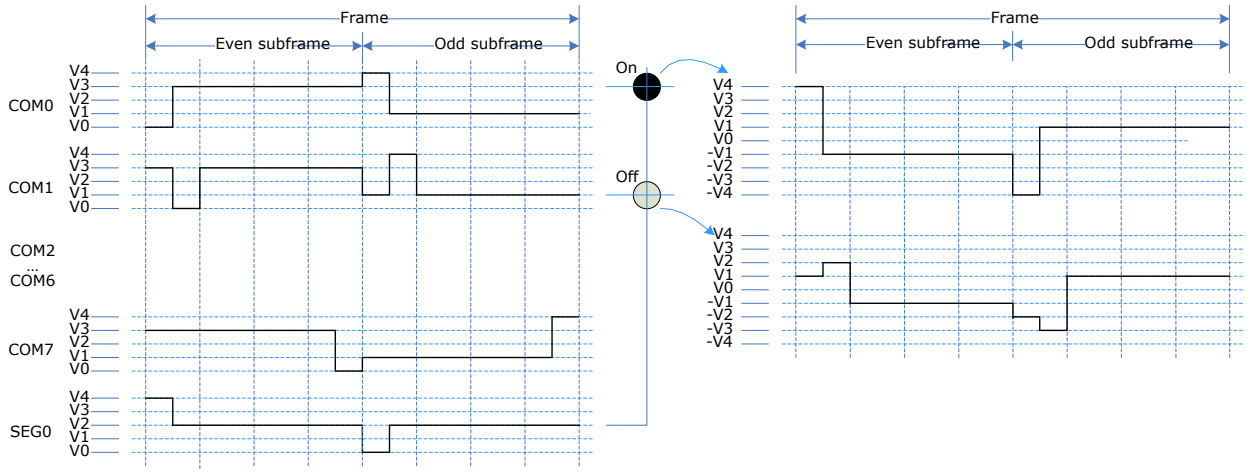


Figure 43-15. 1/4 Bias, 1/8 Duty, Low-Power Waveform



43.6.1.5 LCD Frame Frequency

The LCD frame frequency is defined as the number of times the segments are energized per second. The optimal frame frequency should be in range from 30Hz up to 100Hz to avoid flickering and ghosting effect.

The 32KHz oscillator clock (CLK_SLCD_OSC) is the base clock to define the LCD frame frequency (or frame rate) which in turn depends on the Duty Ratio bits in the Control A register (CTRLA.DUTY[2:0]). CLK_SLCD_OSC is used to generate the LCD waveform data for the enabled phases.

CLK_SLCD_OSC is first divided by a prescaler PVAL=16..128, then divided by DIV=1..8: The prescaler value PVAL is selected by writing the Prescaler bits in the Control A register (CTRLA.PRESC[1:0]), see table below.

Table 43-3. Prescaler Selection

PRESC[1:0]	Prescaler Value (PVAL)
0x0	16
0x1	32
0x2	64
0x3	128

The clock division factor DIV is selected by the Clock Divider bits CTRLA.CKDIV[2:0]. The division factor is DIV=CKDIV[2:0]+1. The duty ratio NB_COM is selected by writing to the Duty Ratio bits in the Control A register (CTRLA.DUTY).

The resulting frame rate is calculated according to this formula:

$$\text{FrameRate} = \frac{f(\text{CLK_SLCD_OSC})}{\text{PVAL} \times \text{DIV} \times \text{NB_COM}}$$

Table 43-4. Examples of Frame Rates for f(CLK_SLCD_OSC) = 32768Hz

Prescaler Value (PVAL)	CKDIV[2:0]	DIV	NB_COM	Frame Rate
128	0x7	8	1	32 Hz
128	0x2	3	1	85.3 Hz
64	0x7	8	2	32 Hz
64	0x2	3	2	85.3 Hz
64	0x4	5	3	34.1 Hz
32	0x3	4	3	85.3 Hz
32	0x7	8	4	32 Hz
32	0x2	3	4	85.3 Hz
32	0x4	5	6	34.1 Hz
16	0x3	4	6	85.3 Hz
16	0x7	8	8	32 Hz
16	0x2	3	8	85.3 Hz

43.6.1.6 LCD Pins Selection

Selection of maximum 48 segment/common lines from 52 LCD pins

There are 52 LCD pins (LPx), of which up to 48 LCD pins can be enabled or disabled individually according to the LCD glass. Each LCD pin can be configured as frontplane (SEG) or backplane (COM), offering various configurations. The maximum number of SEG lines and COM lines are defined in the table below.

Table 43-5. Maximum Number of COM and SEG Lines

Duty	#COM Lines	Max #SEG Lines
Static	1	44
1/2	2	44
1/3	3	44
1/4	4	44
1/6	6	42
1/8	8	40

In order to enable an LCD pin, Write a '1' to the corresponding bit in the corresponding register:

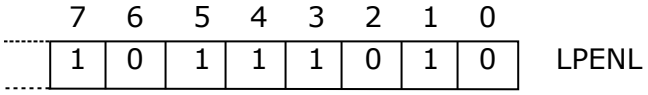
- For LP[31:0], write to LCD Pin Enable Low register bits LPENL[31:0]
- For LP[51:32], writ to LCD Pin Enable High register bits LPENH[18:0]

Writing a '0' to a bit in LPENL or LPENH will disable the corresponding LCD pin. The number of LCD pins enabled should not be higher than the maximum of COM and SEG lines supported (see the table above). Any LCD pin can be enabled individually, LCD pins do not have to be enabled in contiguous manner.

A disabled LCD pin can thus be used as GPIO or alternate function.

According to their duty configuration, COM lines are assigned first to LCD pins enabled. The number of SEG lines enabled is thus the number of LCD pins enabled minus the number of COM lines assigned (limited to the maximum SEG lines supported according duty selection). COM and SEG lines are always assigned in ascending order, as shown in the figure here.

Figure 43-16. LCD Pins Configuration Example

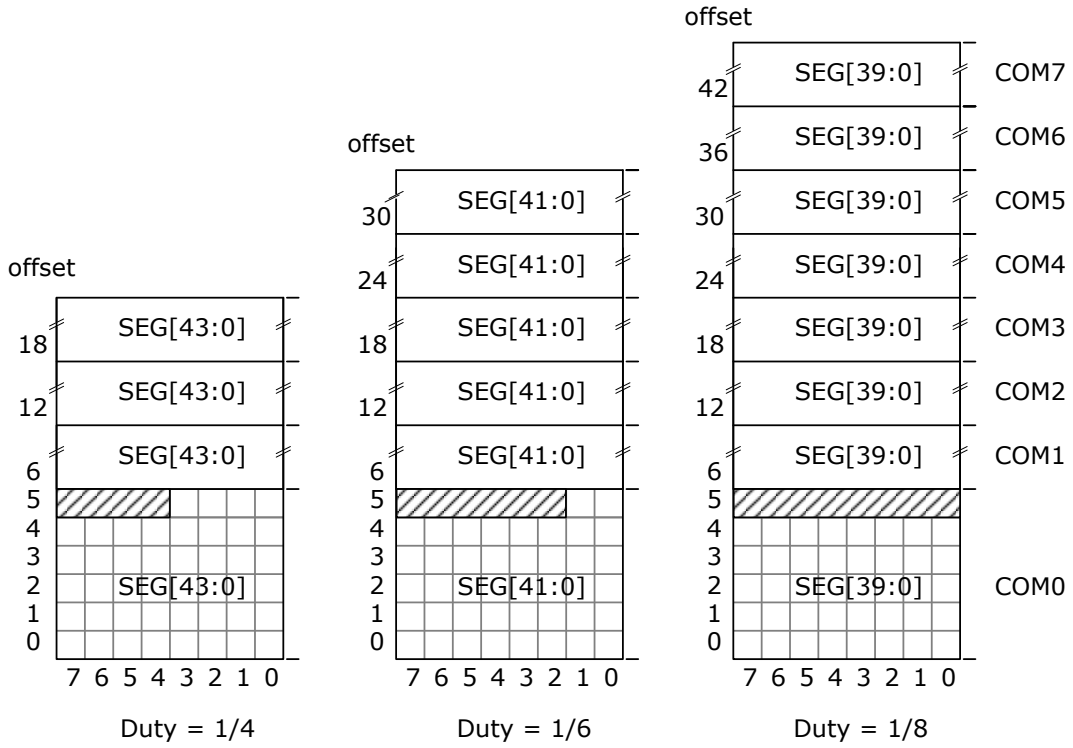


LP0 = _
LP1 = COM0 duty = 1/2
LP2 = _
LP3 = COM1
LP4 = SEG0
LP5 = SEG1
LP6 = _
LP7 = SEG2
...

43.6.1.7 Display Memory Mapping

The display memory size depends on the configured duty ratio. For duty ratios above 1/4, the display memory is 44 bits wide per COM line. For 1/6 duty, the display memory is 42 bits wide per COM line, and for 1/8 duty, the display memory is 40 bits wide per COM line.

Figure 43-17. Display Memory Mapping



Direct Access

CPU can access display memory in direct access by writing to the corresponding Segments Data Low/ High for COMx Line register (SDATAL/Hx). For each bit y of the SDATAL/Hx register there is a corresponding segment connected to SEG y line and COMx line. For example, to update the segment connected to SEG4/COM2, write to bit 4 of the SDATAL2 register.

Indirect Access

The CPU can also update the display memory in indirect access mode, i.e. by writing to the Indirect Segments Data Access register (ISDATA). This register allows to write up to 8 contiguous bits in a single write operation to the display memory:

- SDATA[7:0], segments data value (see the figure above)
- SDMASK[7:0], mask for SDATA. When SDMASK[y]=1, SDATA[y] is not written to display memory
- OFF[5:0], byte offset in display memory (see the figure above)

Locking Shadow Display Memory

Shadow display memory can be locked so that it is not updated when new frames start.

Writing a '1' to the Lock bit in the Control B register (CTRLB.LOCK) will lock the shadow display memory. If the display memory is modified, the display remains unchanged.

Writing a '0' to CTRLB.LOCK will to unlock the shadow display memory. The shadow display memory will be updated when a new frame starts.

43.6.1.8 LCD Power Supply

The LCD voltage levels are generated either from the VLCD pin or by the internal power supply block, through an internal resistor divider network. The internal power supply is selected by writing a '0' to the

External VLCD bit in the Control A register (CTRLA.XVLCD); the external power supply is selected by writing a '1' to CTRLA.XVLCD.

The LCD power supply block generates up to three intermediate voltage levels (in internal or external supply mode), depending on the bias configuration bits in the CTRLA register (CTRLA.BIAS[1:0]).

Table 43-6. VLCD Voltages

CTRLA.BIAS[1:0]	Configuration	Voltages
00	Static	VLCD
01	1/2	VLCD, 1/2 VLCD
10	1/3	VLCD, 2/3 VLCD, 1/3 VLCD
11	1/4	VLCD, 3/4 VLCD, 1/2 VLCD, 1/4 VLCD

The VLCD Status bit in the STATUS register (STATUS.VLCDS) indicates the current relation of VLCD and VDD. When VDD33 > Target VLCD, STATUS.VLCDS is set to '1'. Otherwise, STATUS.VLCDS is cleared to '0'.

The voltage status of VLCD itself is indicated by STATUS.VLCDR: when the VLCD voltage is not well regulated to the target voltage, indicated by STATUS.VLCDR=0, the display quality may be impaired during the ongoing settling period. If the display quality is critical for the application, transient phenomena can be avoided by first disabling the COM/SEG output by writing CTRLD.DISPEN=0, then enable COM/SEG output again once the STATUS.VLCDR bit is set.

43.6.1.9 Contrast Adjustment

The contrast of the LCD is determined by the value of VLCD voltage. The higher the VLCD voltage, the higher is the contrast. The software contrast adjustment is only possible in internal supply mode.

In internal supply mode, VLCD is in the range of 2.5V to 3.5V. VLCD can be adjusted in 16 steps of 60mV by writing a value to the Contrast bits in the Control B register (CTRLB.CTST), see also the according Electrical Characteristics section.

The contrast value can be written at any time, even if SLCD is enabled and running.

Related Links

LCD Power Supply

SLCD Characteristics

43.6.1.10 Saving Power with LowR and Buffer

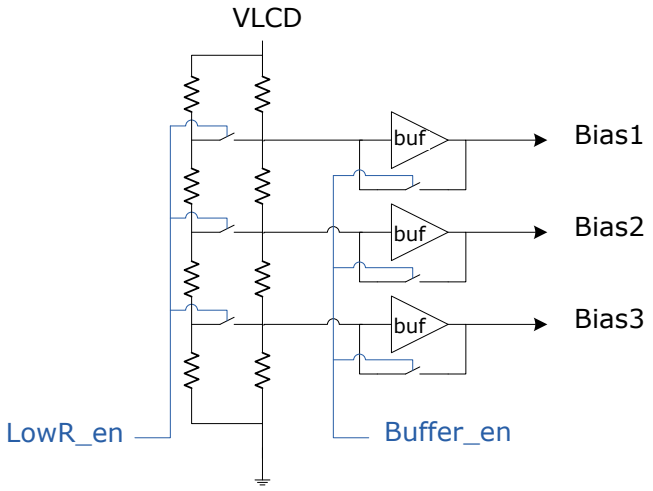
The bias generation block generates the bias voltage that the LCD waveform needs. For intermediate bias levels between ground and VLCD, they are generated by an on-chip resistive voltage divider.

The voltage divider is made of two strings, one high resistance string and one low resistance string. The high resistance string is always on when the LCD is enabled. The low resistance string can be turned on for a configurable amount of time (defined in the SLCD controller) to increase the drive capability of the bias, at the price of increased power consumption. The on-time is aligned at waveform transition so that the best trade off can be found between power and waveform quality by tuning the ratio of the LowR on-time versus the waveform period to find the best trade-off. The LowR on-time is selected by writing the Low Resistance Enable Duration bits in the Control B register (CTRLB.LRD).

A buffer is also provided for each of the intermediate bias levels. The buffers have a fixed current consumption, but provide dynamic current drive capability. Similar to the LowR driver, the buffer can be turned on for only a portion of the waveform period after transition. This is tuned by writing to the Bias Buffer Enable bits in the Control B register (CTRLB.BBD).

Compared to LowR, buffers offer a higher drive capability with relatively fixed operating current, which achieve a better performance (waveform quality versus power consumption) when the panel represents a big capacitive load (i.e. 10nF).

Figure 43-18. LowR and Buffer Illustration



43.6.1.11 Reference Refresh and Power Refresh

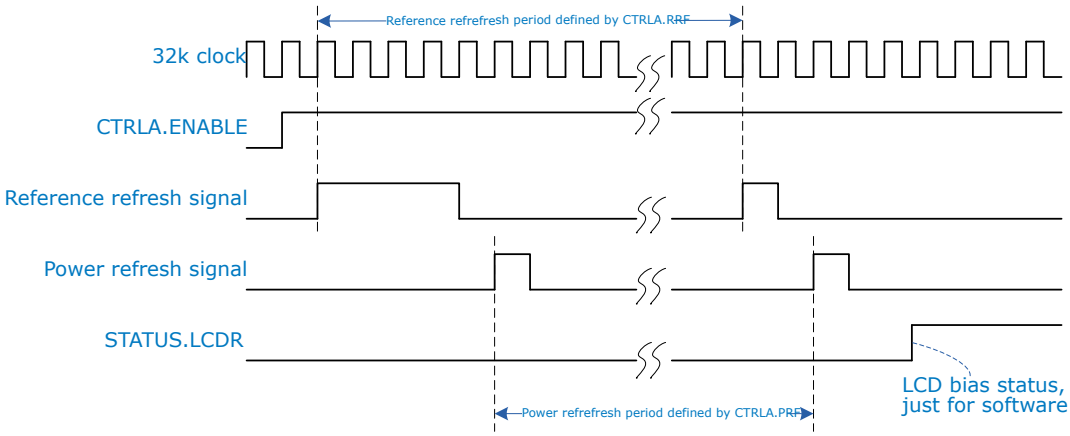
There are two refresh signals for SLCD:

- Reference refresh, which is used to periodically sample the bias current and target VLCD voltage.
- Power refresh, which is used to charge the external VLCD capacitor to the target VLCD level. This is usually due to the fact that the LCD panel driving waveform will need to charge the panel segment capacitors.

The reference refreshment rate is defined by the leakage of the circuit. The bias current and reference voltage start to drift slowly due to parasitic effects. A high frequency refresh rate can tolerate higher leakage at cost of higher power consumption. User can configure the reference refresh rate by writing the Reference Refresh Frequency bits in the Control A register (CTRLA.RRF).

The power refreshment rate is defined by the leakage of the circuit. High frequency could result in bit higher power but can tolerate higher load current on VLCD, which means driving a bigger panel at higher frequency. User can configure the power refresh rate by writing the Power Refresh Frequency bits in the Control A register (CTRLA.PRF)

Figure 43-19. Reference and Power Refresh System Behavior



43.6.2 Additional Features

43.6.2.1 Frame Counters

For several functions (e.g., blinking, or automated modes) a frame counter is required as time base. There are three independent frame counters FC0, FC1 and FC2, which can be associated with any function.

Each frame counter x is enabled by writing a '1' to the Frame Counter x Enable bit in the Control D register (CTRLD.FCxEN), and disabled by writing a '0' to it. When disabling a frame counter, it will be disabled once the CTRLD register is synchronized.

The frame counter is synchronized to the LCD frame start and generates an internal event each time the counter overflows. The overflow value is selected by writing the Write Overflow bits in the Frame Counter x register (FCx.OVF). The FCx register can only be written when the frame counter x is disabled (CTRLD.FCxEN=0).

The frequency of the internal event is defined by the following formula:

$$f_{FCx} = \frac{\text{FrameRate}}{(\text{FCx.OVF} \times 8) + 1}$$

The prescaler of 8 can be bypassed by writing a '1' to the Prescaler Bypass bit in the FCx register (FCx.PB):

$$f_{FCx} = \frac{\text{FrameRate}}{(\text{FCx.OVF}) + 1}$$

43.6.2.2 Blank

To blank the LCD panel, write a '1' to the Blank bit in the Control D register (CTRLD.BLANK). All LCD segments will be OFF at the beginning of next frame. The content of the display memory is not modified.

To output segment data normally, write a '0' to CTRLD.BLANK.

43.6.2.3 Blink

SLCD can be configured to blink all or selected LCD segments. Segments will alternate between on and off state at the frequency given by the selected frame counter.

The blinking feature is configured in the Blink Configuration register (BCFG):

- To blink all segments, write a '0' to the Mode bit in the BCFG register (BCFG.MODE). To blink only selected segments, write a '1' to BCFG.MODE. Then, up to sixteen segments can be enabled individually to blink, they are connected to SEG0, SEG1, and COM[0..7]:
  - To enable blinking for the segment connected to SEG0 and COMx, write a '1' to bit x of the Blink Segment Selection 0 bit field in the BCFG register (BCFG.BSS0). Write a '0' to this bit to disable blinking.
  - To enable blinking for the segment connected to SEG1 and COMy, write a '1' to bit y of the Blink Segment Selection 1 bit field in the BCFG register (BCFG.BSS1). Write a '0' to this bit to disable blinking.

  A segment will blink only if it is already on - otherwise it will remain off.
- The blink frequency is defined by the number of frames between each on/off state, as given by a frame counter. Which frame counter is associated to the blink feature is selected by writing the corresponding frame counter x index to the Frame Counter Selection bits in the BCFG register (BCFG.FCS).

Once the desired blink configuration is written to the BCFG register, blinking is enabled by writing a '1' the to Blink bit in Control C register (CTRLC.BLINK). Blinking is disabled by writing a '0' to CTRLC.BLINK.

The BCFG register cannot be written when blink is enabled. The blink frequency can be modified, provided the selected frame counter has been disabled first.

Table 43-7. Blink Modes

CTRLC.BLANK	CTRLC.BLINK	BSS1[7:0] BSS0[7:0]	Mode
1	x	-	All segments are off
0	0	-	All segments are driven according display memory
0	1	00000000 00000000	All segments are blinking
		xxxxxxxx xxxxxxxx	Selected segments (x=1) are blinking

43.6.2.4 Autonomous Segment Animation

Up to sixteen segments can be animated by an internal circular shift register. Up to sixteen states are then defined, e.g. to make a running wheel. The size of the animation can be configured by writing the number of segments to use to the Size bits in the Circular Shift Register Configuration register (CSRCFG.SIZE).

The circular shift register bits are associated to segments connected to SEG2, SEG3 and COM[0..7] lines, refer to the table below.

Table 43-8. Circular Shift Register Bit Mapping

Circular Shift Register Bit	COM	SEG
0	0	2
1	0	3
2	1	2
3	1	3
4	2	2
5	2	3
6	3	2
7	3	3
8	4	2
9	4	3
10	5	2
11	5	3
12	6	2
13	6	3
14	7	2
15	7	3

The shifting frequency is configured in a frame counter as a number of frames. A frame counter is associated to this feature by writing the corresponding index to the Frame Counter Selection bits in the CSRCFG register (CSRCFG.FCS).

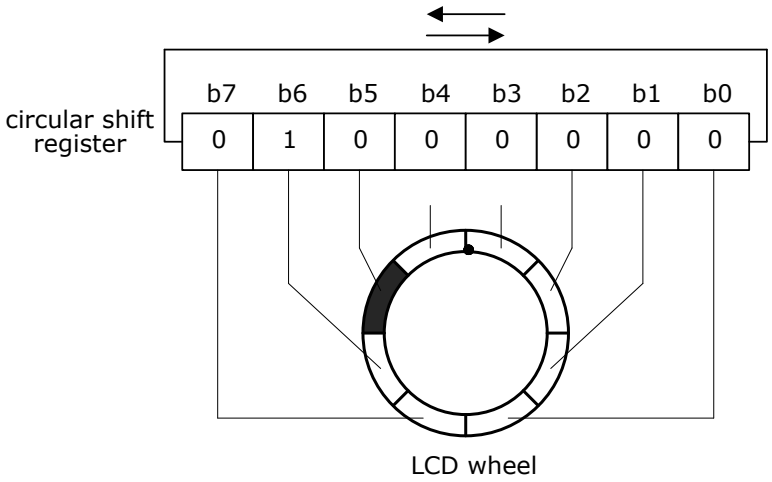
The shift direction is selected by writing to the Direction bit in CSRCFG register (CSRCFG.DIR): DIR=0 is shifting to the left; DIR=1 will shift to the right.

The initial value of the circular shift register must be written to the Data bits in the CSRCFG register (CSRCFG.DATA).

Once the circular shift register is configured, autonomous segment animation is enabled by writing a '1' to the Circular Shift Register Enable bit in the Control C register (CTRLC.CSREN). Animation is disabled by writing a '0' to CTRLC.CSREN. The initial value is shifted each time the selected frame counter overflows.

The CSRCFG register cannot be written when circular shift register is enabled. The shifting frequency can be modified, provided selected frame counter has been disabled first.

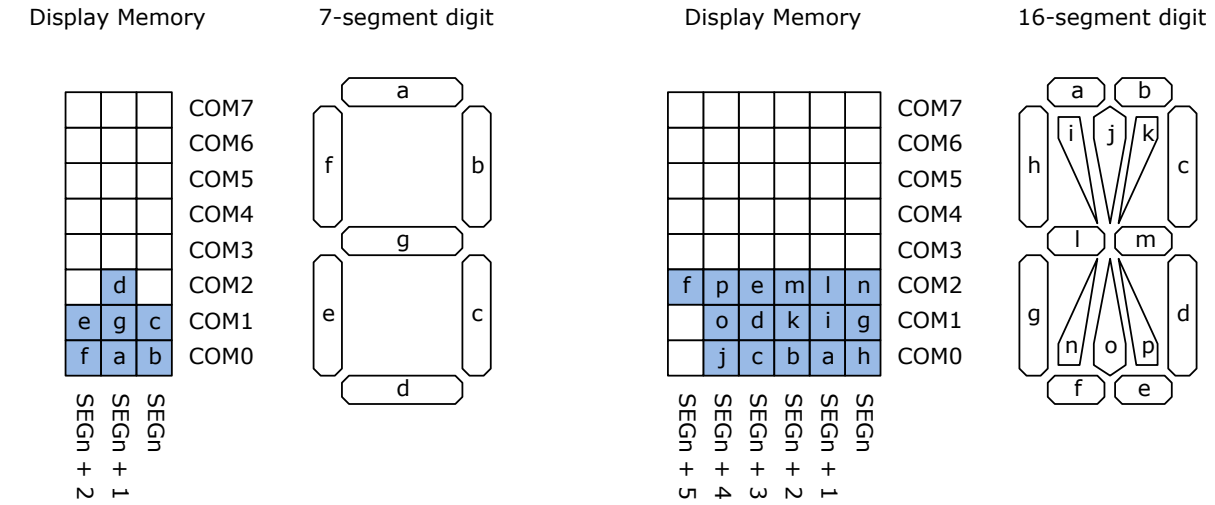
Figure 43-20. LCD Wheel Animation Example (8 Segments)



43.6.2.5 Character Mapping

Multiple segments of the LCD panel can be gathered to digits in order to display characters. Digits can be of various type (e.g., 7-segment, 16-segment) and mapped in the display memory differently, refer to the figure below for examples. The SLCD supports displaying up to 24-segment characters on any existing LCD panel.

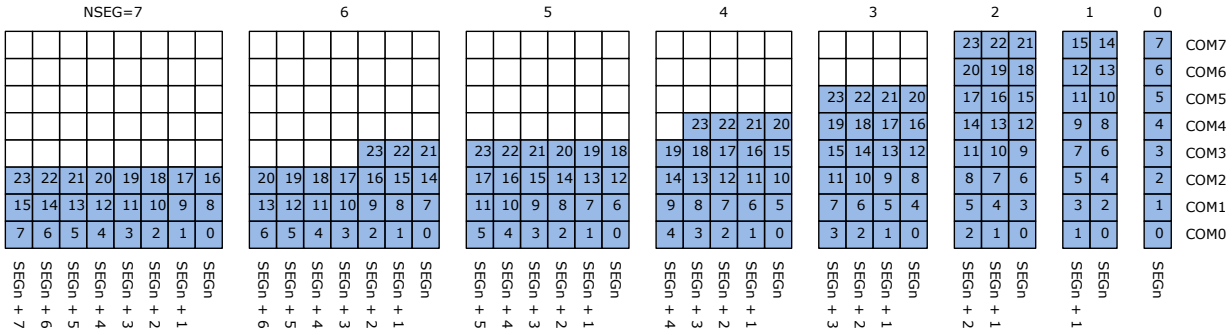
Figure 43-21. Examples of Digit Mapped in Display Memory



Several digits can be mapped in display memory to form a digits chain, which is used to display a characters string. Once configured, the user updates a digit by providing up to 24 bits with a single write access. ASCII tables provide standard symbols which should be handled by software before being displayed.

Each character is a continuous display area with its SEG axis defined by the Character Mapping Configuration register (CMCFG.NSEG). The maximum size is 24. Refer also to the example figure below.

Figure 43-22. Character Mapping Example



The 24 segments value is updated by a write access to the Character Mapping Segments Data register (CMDATA).

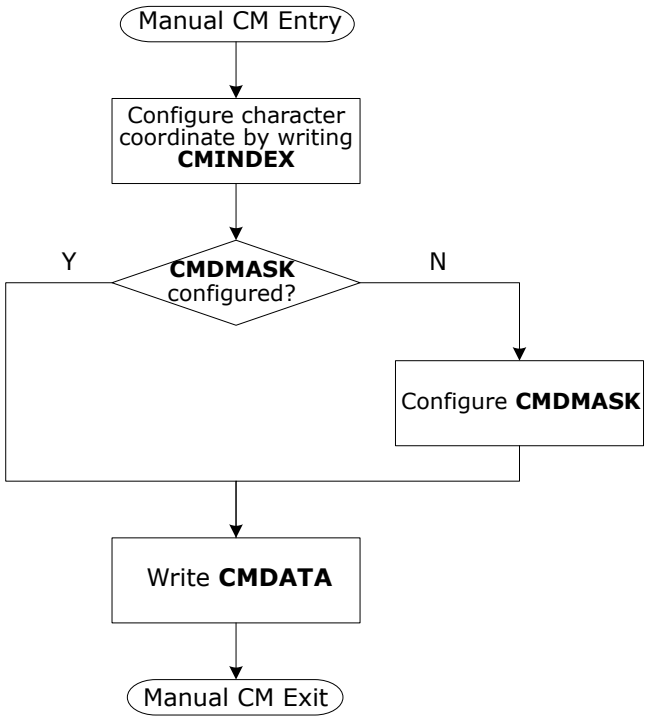
Since the CMDATA register can be up to 24 bits whereas a digit can consist of less than 24 segments, a mask can be configured to write only selected bits: Writing a '1' to a bit in the Character Mapping Data Mask register (CMDMASK) will mask the corresponding bit in the CMDATA register. This bit will not be written in display memory when CMDATA register is written. Write a '0' to a bit of CMDMASK to disable the mask for that bit. The corresponding bit in CMDATA will be written to display memory.

CMDATA is not written to the display memory in one shot, but in a serial way. Serialization takes 24 APB cycles. The user can check the status of the serial writing by reading the Character Mapping Write Busy bit in the Status register (STATUS.CMWRBUSY). Any write access to the display memory through CMDATA/SDATA/ISDATA will be ignored when STATUS.CMWRBUSY is high. Therefore, user must not write to CMDATA while STATUS.CMWRBUSY is asserted.

The coordinate of a character is defined by Character Mapping Index (CMINDEX) register.

The expected usage model for character mapping is shown below:

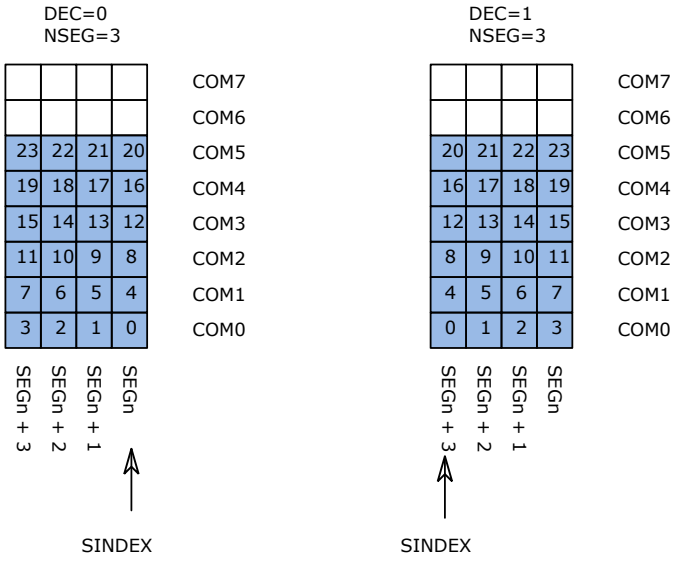
Figure 43-23. Usage Model for Character Mapping Function



By this character mapping feature, user could update any particular rectangular window defined by CMCFG and CMINDEX in the display memory just by a single APB write access.

The Decrement SEG Index bit in Character Mapping Configuration (CMCFG.DEC) defines the mapping order. And CMINDEX.SINDEX also have different coordinate under different DEC configuration. Refer to figure below for illustration.

Figure 43-24. Character Mapping Order



Note: The user must make sure all character bits could be mapped to some bit in display memory, not somewhere out side display memory.

43.6.2.6 Automated Character Mapping

Displaying predefined character strings can be automated by using the Direct Memory Access (DMA). This is the Automated Character Mapping (ACM) feature. Compared to manual character mapping, there is one restriction for ACM: the first character is always mapped to the COM0 line.

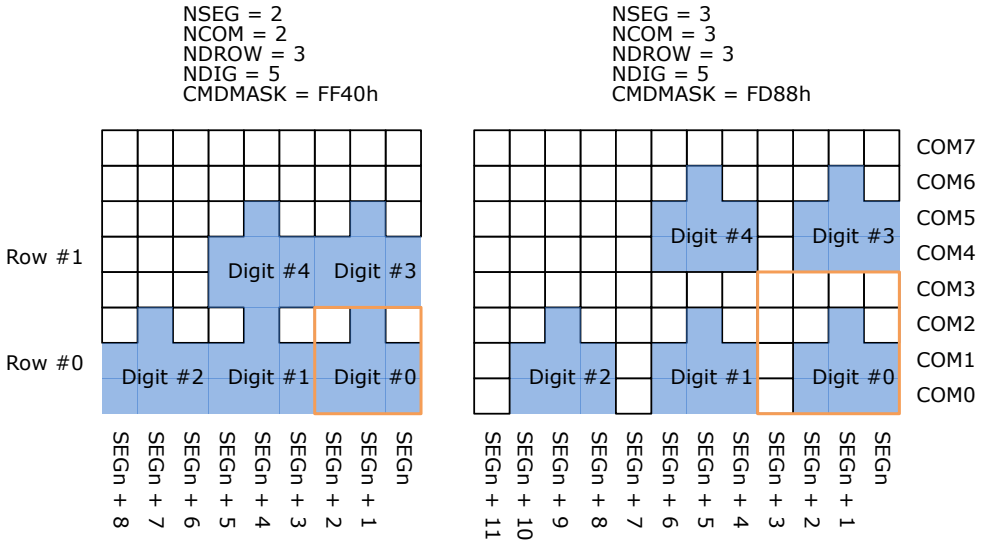
The size of a character is defined by the CMCFG register; the mask of a character is defined by the CMDMASK register.

Several digits can be mapped on different rows and form a string (chain), which could be configured by the Auto Character Mapping Configuration (ACMCFG) register:

- ACMCFG.NDROW defines number of digit per row.
- ACMCFG.STSEG defines the index of the first SEG line of the first digit.
- ACMCFG.NDIG defines the number character in the whole string.

If the three fields are configured correctly, ACM logic can infer each character in the string automatically. Each row should contain NDROW digits except the last row. Digits of next row should be aligned with the digits of the previous row (same SEG lines). The figure below gives an example for how to configure a string in ACM mode.

Figure 43-25. Examples of Digits Chains Mapped in Display Memory

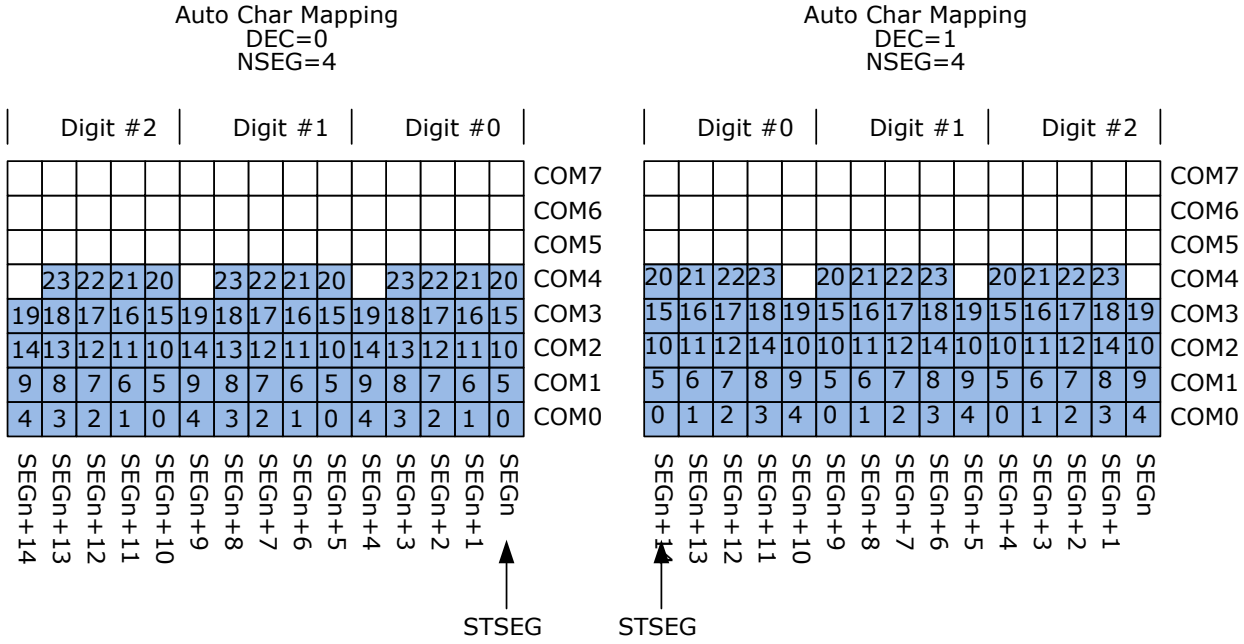


After the user has enabled ACM by writing a '1' to the ACM Enable bit in the Control C register (CTRLC.ACMEN), the ACM logic (state machine) will start according to the configuration by user. Whenever the user writes a '0' to CTRLC.ACMEN, the ACM logic will return to its initial state.

The mapping method can be altered: The line index direction is selected by writing to the Decrement SEG Line Index bit in the Character Mapping Configuration register (CMCFG.DEC). Either the segment is starting from the bottom right (DEC=0) or from the bottom left (DEC=1).
Note: The Start SEG bits in the ACM Configuration register (ACMCFG.STSEG) have different coordinates under different CMCFG.DEC configurations.

Refer to the figure below for illustration.

Figure 43-26. DEC and Auto Character Mapping Order



There are two usage models available under ACM, defined by the Mode bit in the Automated Character Mapping Configuration register (ACMCFG.MODE): when MODE=0, the *Sequential Characters String Display* mode is selected. When MODE=1, the *Scrolling of Characters String Display* mode is selected.

Both modes use the character mapping configuration (see Character Mapping) and require the digit to be mapped with contiguous SEG/COM lines in order to increment the index for the next digit automatically. For example when the first digit is mapped on SEG2 and SEG3 lines, the second digit must be mapped on SEG4 and SEG5 lines.

Sequential Characters String Display

This mode displays characters of a string on a digits chain on the LCD panel. The characters string is split in several sub strings, which are to be displayed sequentially at a selected frequency (frame counter basis). It requires an internal frame counter (time base) and the DMA controller to transfer the sub strings.

To use this mode, character mapping must be configured first (see Character Mapping), then additional configuration must be done to specify the sequential display. In the Automated Character Mapping Configuration register (ACMCFG):

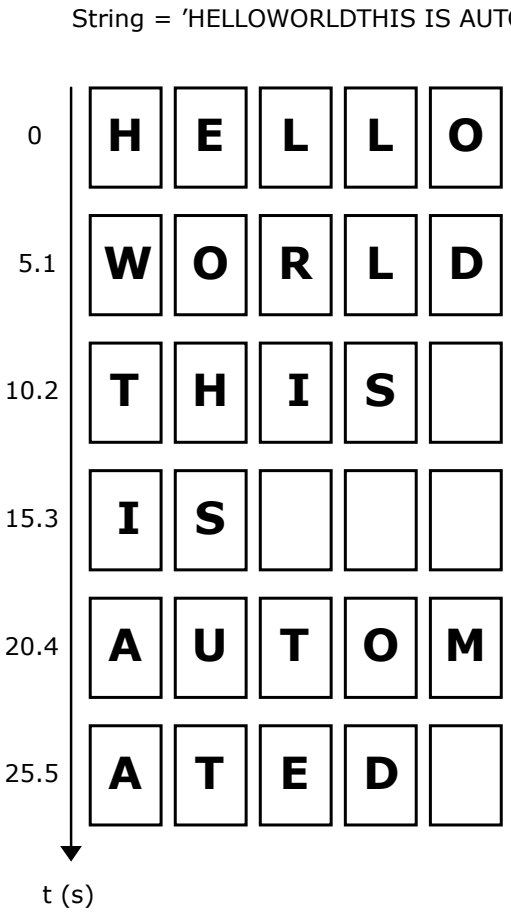
- Write the size of the digits chain on LCD panel to the Number of Digit bits (ACMCFG.NDIG).
- Write a '0' to the Mode bit in order to select the sequential mode (ACMCFG.MODE).
- Configure a frame counter to set the display period (period between two sub strings display, see Frame Counters), and write the frame counter index to the Frame Counter Selection bits (ACMCFG.FCS).

The DMA controller should be configured to transfer several blocks corresponding to the sub strings. Each block size is equal to the number of digit on the LCD panel (sub strings may require some blank characters in order to be displayed correctly). The number of blocks is the number of sub strings required to display the complete characters string. To repeat the complete character string display, the DMA controller should repeat the total blocks transfer. The DMA must write the blocks to the Character Mapping Segments Data register (CMDATA).

To enable the sequential characters string display, write a '1' to the Automated Character Mapping Enable bit in the Control B register (CTRLB.ACMEN); write a '0' to CTRLB.ACMEN to disable it.

The selected frame counter defines the number of frames before the SLCD triggers a new DMA block transfer. For example, if frame rate is 50Hz (20ms) in standard waveform mode, the characters string can be displayed with a period from 160ms up to 5.1s.

Figure 43-27. Sequential Character String Example (NDIG=5, frame rate=50Hz, FC0.OVF=31)



Scrolling of Characters String

This mode displays characters of a string on a digits chain on the LCD panel with a scrolling effect. For each step, a part of the characters string is displayed by shifting the string by one character in left direction. The period between each step can be configured on frame counter basis. This mode requires an internal frame counter (time base) and the DMA controller to transfer the character string.

To use this mode, character mapping must be configured first (see Character Mapping), then additional configuration must be done to specify the scrolling display. In the Automated Character Mapping Configuration register (ACMCFG):

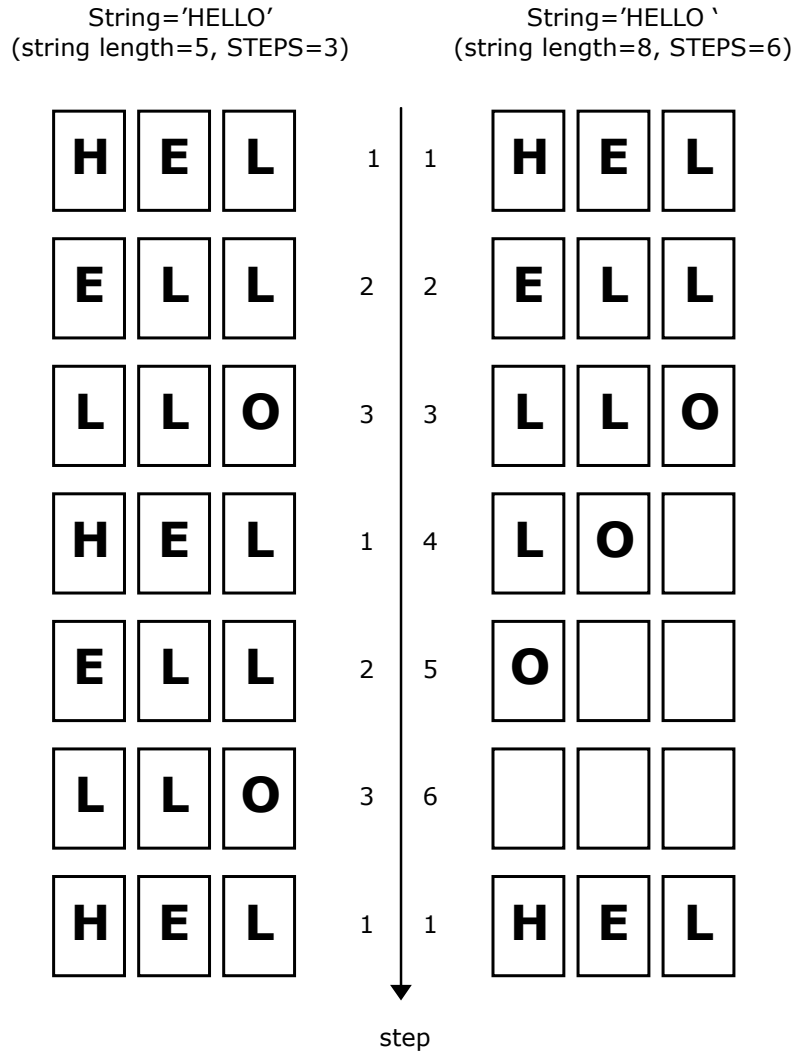
- Write the size of the digits chain on the LCD panel to the Number of Digit bits (ACMCFG.NDIG).
- Write the number of scrolling steps to the Steps bits (ACMCFG.STEPS).
  The number of steps is equal to string length - NDIG + 1.
- Write a '1' to the Mode bit in order to to select the scrolling mode (ACMCFG.MODE).
- Configure a frame counter to set the display period (period between two steps, see Frame Counters) and write the frame counter index to the Frame Counter Selection bits (ACMCFG.FCS).

The DMA controller should be configured to transfer several times the same block corresponding to the complete characters string (the string may require some blank characters at the end in order to be displayed correctly). For a complete scrolling of the string, the number of DMA block transfers should be at least the number of steps. To repeat the complete character string scrolling, the DMA controller should repeat the blocks transfer. The DMA must write the blocks to the Character Mapping Segments Data register (CMDATA).

To enable the scrolling of characters string, write a '1' to the Automated Character Mapping Enable bit in the Control B register (CTRLB.ACMEN); write a '0' to CTRLB.ACMEN and to disable it.

The selected frame counter defines the number of frames before the SLCD triggers a new DMA block transfer, after each scrolling steps.

Figure 43-28. Characters String Scrolling Examples (NDIG=3)

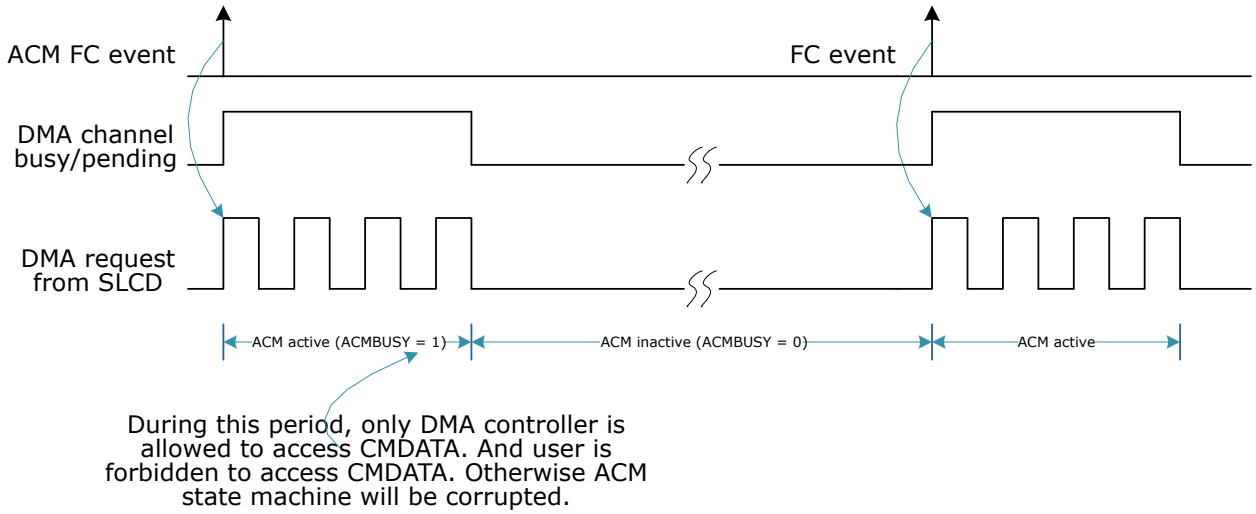


Limitations of Auto Character Mapping Mode

If ACM is enabled and the Auto Character Mapping Busy bit in the Status register (STATUS.ACMBUSY) is asserted, any write access to the Character Mapping Data register (CMDATA) will be considered as coming from the DMA controller, and therefore handled by the ACM state machine. So if the user accidentally writes CMDATA in this period, the ACM state machine will be corrupted. Consequently, the user should not write to the CMDATA register when ACM is active, and make sure that all CMDATA access is in the period of ACM inactivity.

For this reason, the recommended usage is to first disable ACM, and then update CMINDEX and CMDATA. After that, ACM can be re-enabled safely.

Figure 43-29. Limitations of Auto Character Mapping Mode



43.6.2.7 Automated Bit Mapping

Several segments on the LCD panel can be gathered to make a *symbol*, which can be animated (i.e., have several states). Data corresponding to each state of the animation can be stored in system memory, and is transferred periodically to the display memory using the DMA controller.

The DMA controller can update up to eight contiguous bits in display memory by writing to the ISDATA register, refer to Display Memory Mapping. To update more than eight bits, the DMA controller must be configured to transfer multiple words before the shadow memory is updated. This number of words must be written to the Size bits in the Automated Bit Mapping Configuration register (ABMCFG.SIZE); it indicates the number of DMA writes to the display memory to form an animation frame.

To make an automated animation of N states with M contiguous segment values in display memory, the DMA controller must be configured to transfer N x M/8 words (8 contiguous segments are updated per write access). If segment values are not contiguous, the DMA size can be up to N x M (1 segment is updated per write access).

The display period (time between each frame) is defined by a frame counter, which will trigger a new DMA block transfer for the next frame (refer to Frame Counters). The frame counter index must be written to the Frame Counter Selection bits in the ABMCFG register (ABMCFG.FCS).

To enable the automated bit mapping, write a '1' to the Automated Bit Mapping Enable bit in the Control B register (CTRLB.ABMEN); to disable it, write a '0' to CTRLB.ABMEN.

An animation can be repeated if the DMA controller is configured to repeat the whole transfer.

If ABM is enabled, the user can check whether ABM is busy or idle by reading the Auto Bit Mapping Busy bit in the Status register (STATUS.ABMBUSY).

43.6.3 DMA Operation

The SLCD generates the following DMA requests:

- Display Memory Update (DMU): the request is set when the selected frame counter overflows. To select the frame counter for DMA to update the display memory, write the frame counter index to Display Memory Frame Counter Selection bits in the Control A register (CTRLA.DMFCS).

The request is cleared when any SDATAL/Hx register is written to. Refer to Frame Counters for details.
The frame counter associated to the DMA request is the same as the interrupt source, refer also to Interrupts.

- Automated Character Mapping Data Ready (ACMDRDY): the request is set when the frame counter associated to automated character mapping function overflows. The request is cleared when CMDATA register is written. Refer to Automated Character Mapping for details.
- Automated Bit Mapping Data Ready (ABMDRDY): the request is set when the frame counter associated to automated bit mapping function overflows. The request is cleared when ISDATA register is written. Refer to Automated Bit Mapping for details.

Note: If the CPU accesses the registers which are source of DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

43.6.4 Interrupts

The SLCD has the following interrupt sources:

- Frame Counter 0 Overflows (FC0O): Indicates that the frame counter 0 has overflowed, it has reached its top value and wrapped to zero. Refer to Frame Counters for details.
- Frame Counter 1 Overflows (FC1O): Indicates that the frame counter 1 has overflowed, it has reached its top value and wrapped to zero. Refer to Frame Counters for details.
- Frame Counter 2 Overflows (FC2O): Indicates that the frame counter 2 has overflowed, it has reached its top value and wrapped to zero. Refer to Frame Counters for details.
- VLCD Ready Toggle (VLCDRT): Indicates that status of LCD Ready has changed.
- VLCD Status Toggle (VLCDST): Indicates that relation between VLCD and chip VDD has changed.
- Pump Run Status Toggle (PRST): Indicates that Pump Run Status has changed.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the SLCD is reset. See INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the Nested Vector Interrupt Controller for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

43.6.5 Events

The SLCD can generate the following output events:

- Frame Counter 0 Overflow (FC0O): Indicates that the frame counter 0 has overflowed, it has reached its top value and wrapped to zero. Refer to Frame Counters for details.
- Frame Counter 1 Overflow (FC1O): Indicates that the frame counter 1 has overflowed, it has reached its top value and wrapped to zero. Refer to Frame Counters for details.
- Frame Counter 0 Overflow (FC2O): Indicates that the frame counter 2 has overflowed, it has reached its top value and wrapped to zero. Refer to Frame Counters for details.

Writing a '1' to an Event Output bit in the Event Control Register enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

43.6.6 Sleep Mode Operation

The SLCD can be configured to operate in any sleep mode. To be able to run in standby mode, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be written to '1'. The SLCD will continue to refresh the LCD panel from the content of the memory display. If the RUNSTDBY bit is written to '0' when device enters into standby mode, the SLCD will stop driving the LCD panel at the end of the current frame, and the VLCD and bias generators will be stopped.

The SLCD can wake up the device from any sleep mode using interrupts.

43.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

The following registers are synchronized when written:

- Control D register (CTRLD)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

43.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RUNSTDBY	WMOD	DUTY[2:0]			ENABLE	SWRST
0x01		15:8		CKDIV[2:0]					PRESC[1:0]	
0x02		23:16	DMFCS[1:0]		PRF[1:0]		XVLCD		BIAS[1:0]	
0x03		31:24						RRF[2:0]		
0x04	CTRLB	7:0	BBEN				BBD[3:0]			
0x05		15:8	LREN				LRD[3:0]			
0x06	CTRLC	7:0	CTST[3:0]				ACMEN	ABMEN	LOCK	CLEAR
0x07		15:8								
0x08	CTRLD	7:0	DISPEN	FC2EN	FC1EN	FC0EN		CSREN	BLINK	BLANK
0x09 ... 0x0B	Reserved									
0x0C	EVCTRL	7:0						FC2OEO	FC1OEO	FC0OEO
0x0D	INTENCLR	7:0			PRST	VLCDST	VLCDRT	FC2O	FC1O	FC0O
0x0E	INTENSET	7:0			PRST	VLCDST	VLCDRT	FC2O	FC1O	FC0O
0x0F	INTFLAG	7:0			PRST	VLCDST	VLCDRT	FC2O	FC1O	FC0O
0x10	STATUS	7:0			ABMBUSY	ACMBUSY	CMWRBUSY	VLCDS	PRUN	VLCDR
0x11 ... 0x13	Reserved									
0x14	SYNCBUSY	7:0						CTRLD	ENABLE	SWRST
0x15		15:8								
0x16		23:16								
0x17		31:24								
0x18	FC0	7:0	PB			OVF[4:0]				
0x19	FC1	7:0	PB			OVF[4:0]				
0x1A	FC2	7:0	PB			OVF[4:0]				
0x1B	Reserved									
0x1C	LPENL	7:0	LPEN[7:0]							
0x1D		15:8	LPEN[15:8]							
0x1E		23:16	LPEN[23:16]							
0x1F		31:24	LPEN[31:24]							
0x20	LPENH	7:0	LPEN[7:0]							
0x21		15:8	LPEN[15:8]							
0x22		23:16	LPEN[23:16]							
0x23		31:24	LPEN[31:24]							
0x24	SDATAL0	7:0	SDATA[7:0]							
0x25		15:8	SDATA[15:8]							
0x26		23:16	SDATA[23:16]							
0x27		31:24	SDATA[31:24]							
0x28	SDATAH0	7:0	SDATA[7:0]							
0x29		15:8					SDATA[11:8]			
0x2A		23:16								
0x2B		31:24								

Offset	Name	Bit Pos.								
0x2C	SDATAL1	7:0	SDATA[7:0]							
0x2D		15:8	SDATA[15:8]							
0x2E		23:16	SDATA[23:16]							
0x2F		31:24	SDATA[31:24]							
0x30	SDATAH1	7:0	SDATA[7:0]							
0x31		15:8					SDATA[11:8]			
0x32		23:16								
0x33		31:24								
0x34	SDATAL2	7:0	SDATA[7:0]							
0x35		15:8	SDATA[15:8]							
0x36		23:16	SDATA[23:16]							
0x37		31:24	SDATA[31:24]							
0x38	SDATAH2	7:0	SDATA[7:0]							
0x39		15:8					SDATA[11:8]			
0x3A		23:16								
0x3B		31:24								
0x3C	SDATAL3	7:0	SDATA[7:0]							
0x3D		15:8	SDATA[15:8]							
0x3E		23:16	SDATA[23:16]							
0x3F		31:24	SDATA[31:24]							
0x40	SDATAH3	7:0	SDATA[7:0]							
0x41		15:8					SDATA[11:8]			
0x42		23:16								
0x43		31:24								
0x44	SDATAL4	7:0	SDATA[7:0]							
0x45		15:8	SDATA[15:8]							
0x46		23:16	SDATA[23:16]							
0x47		31:24	SDATA[31:24]							
0x48	SDATAH4	7:0	SDATA[7:0]							
0x49		15:8						SDATA[9:8]		
0x4A		23:16								
0x4B		31:24								
0x4C	SDATAL5	7:0	SDATA[7:0]							
0x4D		15:8	SDATA[15:8]							
0x4E		23:16	SDATA[23:16]							
0x4F		31:24	SDATA[31:24]							
0x50	SDATAH5	7:0	SDATA[7:0]							
0x51		15:8						SDATA[9:8]		
0x52		23:16								
0x53		31:24								
0x54	SDATAL6	7:0	SDATA[7:0]							
0x55		15:8	SDATA[15:8]							
0x56		23:16	SDATA[23:16]							
0x57		31:24	SDATA[31:24]							
0x58	SDATAH6	7:0	SDATA[7:0]							
0x59		15:8								
0x5A		23:16								

Offset	Name	Bit Pos.								
0x5B		31:24								
0x5C	SDATAL7	7:0	SDATA[7:0]							
0x5D		15:8	SDATA[15:8]							
0x5E		23:16	SDATA[23:16]							
0x5F		31:24	SDATA[31:24]							
0x60	SDATAH7	7:0	SDATA[7:0]							
0x61		15:8								
0x62		23:16								
0x63		31:24								
0x64	ISDATA	7:0	SDATA[7:0]							
0x65		15:8	SDMASK[7:0]							
0x66		23:16			OFF[5:0]					
0x67		31:24								
0x68	BCFG	7:0						FCS[1:0]		MODE
0x69		15:8	BSS0[7:0]							
0x6A		23:16	BSS1[7:0]							
0x6B		31:24								
0x6C	CSRCFG	7:0	SIZE[3:0]					FCS[1:0]		DIR
0x6D		15:8	DATA[7:0]							
0x6E		23:16	DATA[15:8]							
0x6F		31:24								
0x70	CMCFG	7:0					DEC	NSEG[2:0]		
0x71 ... 0x73	Reserved									
0x74	ACMCFG	7:0	NDIG[3:0]					NCOM[2:0]		
0x75		15:8	STEPS[7:0]							
0x76		23:16	MODE		NDROW[5:0]					
0x77		31:24	FCS[1:0]		STSEG[5:0]					
0x78	ABMCFG	7:0	SIZE[5:0]						FCS[1:0]	
0x79 ... 0x7B	Reserved									
0x7C	CMDATA	7:0	SDATA[7:0]							
0x7D		15:8	SDATA[15:8]							
0x7E		23:16	SDATA[23:16]							
0x7F		31:24								
0x80	CMDMASK	7:0	SDMASK[7:0]							
0x81		15:8	SDMASK[15:8]							
0x82		23:16	SDMASK[23:16]							
0x83		31:24								
0x84	CMINDEX	7:0			SINDEX[5:0]					
0x85		15:8						CINDEX[2:0]		

43.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

43.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x03D80000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
						RRF[2:0]		
Access						RW	RW	RW
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DMFCS[1:0]		PRF[1:0]		XVLCD		BIAS[1:0]	
Access	RW	RW	RW	RW	RW		RW	RW
Reset	0	0	0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
		CKDIV[2:0]					PRESC[1:0]	
Access		RW	RW	RW			RW	RW
Reset		0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	WMOD	DUTY[2:0]			ENABLE	SWRST
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bits 26:24 – RRF[2:0]: Reference Refresh Frequency
These bits define the bias reference refresh frequency

These bits are not synchronized.

Value	Name	Description
0	RR2000	2kHz
1	RR1000	1kHz
2	RR500	500Hz
3	RR250	250Hz

Value	Name	Description
4	RR125	125Hz
5	RR62	62.5Hz

Bits 23:22 – DMFCS[1:0]: Display Memory Update Frame Counter Selection
These bits select the frame counter to use to update display memory.

These bits are not synchronized.

Value	Name	Description
0	FC0	Frame Counter 0
1	FC1	Frame Counter 1
2	FC2	Frame Counter 2
3	NFC	Frame Counter event to DMU is forced to 0

Bits 21:20 – PRF[1:0]: Power Refresh Frequency
These bits define the charge pump refresh frequency

These bits are not synchronized.

Value	Name	Description
0	PR2000	2kHz
1	PR1000	1kHz
2	PR500	500Hz
3	PR250	250Hz

Bit 19 – XVLCD: External VLCD
This bit configures how VLCD is generated.

This bit is not synchronized.

Value	Description
0	Internal VLCD generation.
1	External VLCD generation.

Bits 17:16 – BIAS[1:0]: Bias Setting
These bits configure the bias setting.

These bits are not synchronized.

Value	Name	Description
0	STATIC	Static
1	HALF	1/2 bias
2	THIRD	1/3 bias
3	FOURTH	1/4 bias

Bits 14:12 – CKDIV[2:0]: Clock Divider
These bits configure the clock divider, refer to LCD Frame Frequency.

Clock division value after prescaler DIV = CKDIV + 1.

These bits are not synchronized.

Bits 9:8 – PRESC[1:0]: Clock Prescaler
These bits configure the clock prescaler, refer to LCD Frame Frequency.

These bits are not synchronized.

Value	Name	Description
0	PRESC16	16
1	PRESC32	32
2	PRESC64	64
3	PRESC128	128

Bit 6 – RUNSTDBY: Run in Standby
This bit controls the behavior of SLCD during standby sleep mode.

0: SLCD stops driving LCD panel.

1: SLCD continues to operate during standby sleep mode.

Bit 5 – WMOD: Waveform Mode
This bit configures the waveform mode.

This bit is not synchronized.

Value	Name	Description
0	LP	Low Power Waveform Mode (type B, frame-inversion)
1	STD	Standard Waveform Mode (type A, bit-inversion).

Bits 4:2 – DUTY[2:0]: Duty Ratio
These bits configure the duty ratio NB_COM for the LCD frame rate.

These bits are not synchronized.

Value	Name	Description
0x0	STATIC	NB_COM=1
0x1	HALF	NB_COM=2
0x2	THIRD	NB_COM=3
0x3	FOURTH	NB_COM=4
0x4	SIXTH	NB_COM=6
0x5	EIGHT	NB_COM=8

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the SLCD, except display/shadow memory to their initial state, and the SLCD will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Due to synchronization, there is delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	A reset operation is ongoing.

43.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	LREN				LRD[3:0]			
Access	RW				RW	RW	RW	RW
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BBEN				BBD[3:0]			
Access	RW				RW	RW	RW	RW
Reset	0				0	0	0	0

Bit 15 – LREN: Low Resistance Enable
This bit enables/disables the low resistance network.

This bit is not synchronized.

Value	Description
0	Low resistance network is disabled.
1	Low resistance network is enabled.

Bits 11:8 – LRD[3:0]: Low Resistance Enable Duration
These bits configure the enable duration of the low resistance network.

Enable duration = (LRD + 1) x period of CLK_SLCD_OSC.

These bits are not synchronized.

Bit 7 – BBEN: Bias Buffer Enable
This bit enables/disables the bias buffer.

This bit is not synchronized.

Value	Description
0	Bias buffer is disabled.
1	Bias buffer is enabled.

Bits 3:0 – BBD[3:0]: Bias Buffer Enable Duration
These bits configure the enable duration of the bias buffer.

Enable duration = (BBD + 1) x period of CLK_SLCD_OSC.

These bits are not synchronized.

43.8.3 Control C

Name: CTRLC
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CTST[3:0]				ACMEN	ABMEN	LOCK	CLEAR
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – CTST[3:0]: Contrast Adjustment
These bits adjust the contrast of the LCD panel. There are 16 steps, each step is 60mV. Refer to Contrast Adjustment.

Bit 3 – ACMEN: Automated Character Mapping Enable
This bit enables the automated character mapping mode.

Value	Description
0	Automated character mapping is disabled.
1	Automated character mapping is enabled.

Bit 2 – ABMEN: Automated Bit Mapping Enable
This bit enables the automated bit mapping mode.

Value	Description
0	Automated bit mapping is disabled.
1	Automated bit mapping is enabled.

Bit 1 – LOCK: Lock Shadow Memory
This bit allows update of shadow display memory.

Value	Description
0	Shadow display memory is unlocked.
1	Shadow display memory is locked.

Bit 0 – CLEAR: Clear Display Memory
Writing a '1' to this bit clears immediately the display memory.

Writing a '0' to this bit has no effect.

43.8.4 Control D

Name: CTRLD
Offset: 0x08
Reset: 0x80
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	DISPEN	FC2EN	FC1EN	FC0EN		CSREN	BLINK	BLANK
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bit 7 – DISPEN: Display enable
These bit enable COM/SEG signal output.

These bits are not synchronized.

Bit 6 – FC2EN: Frame Counter 2 Enable
This bit enables the frame counter 2.

Value	Description
0	Frame counter 2 is disabled.
1	Frame counter 2 is enabled.

Bit 5 – FC1EN: Frame Counter 1 Enable
This bit enables the frame counter 1.

Value	Description
0	Frame counter 1 is disabled.
1	Frame counter 1 is enabled.

Bit 4 – FC0EN: Frame Counter 0 Enable
This bit enables the frame counter 0.

Value	Description
0	Frame counter 0 is disabled.
1	Frame counter 0 is enabled.

Bit 2 – CSREN: Circular Shift Register Enable
This bit enables the circular shift register.

Value	Description
0	Circular shift register is disabled.
1	Circular shift register is ensabled.

Bit 1 – BLINK: Blinking Enable
This bit enables the blink mode.

Value	Description
0	Blink mode is disabled.
1	Blink mode is ensabled.

Bit 0 – BLANK: Blank LCD
This bit allows user to blank all LCD segments (transparent).

Value	Description
0	The state of the LCD segments is defined by shadow display memory.
1	Blank all LCD segments.

43.8.5 Event Control

Name: EVCTRL
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
						FC2OEO	FC1OEO	FC0OEO
Access						RW	RW	RW
Reset						0	0	0

Bit 2 – FC2OEO: Frame Counter 2 Overflow Event Output Enable
This bit enables the Frame Counter 2 Overflow event.

Value	Description
0	Frame Counter 2 Overflow event is disabled and no event is generated.
1	Frame Counter 2 Overflow event is enabled and will be generated when the frame counter overflows.

Bit 1 – FC1OEO: Frame Counter 1 Overflow Event Output Enable
This bit enables the Frame Counter 1 Overflow event.

Value	Description
0	Frame Counter 1 Overflow event is disabled and no event is generated.
1	Frame Counter 1 Overflow event is enabled and will be generated when the frame counter overflows.

Bit 0 – FC0OEO: Frame Counter 0 Overflow Event Output Enable
This bit enables the Frame Counter 0 Overflow event.

Value	Description
0	Frame Counter 0 Overflow event is disabled and no event is generated.
1	Frame Counter 0 Overflow event is enabled and will be generated when the frame counter overflows.

43.8.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			PRST	VLCDST	VLCDRT	FC2O	FC1O	FC0O
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit 5 – PRST: Pump Run Status Toggle Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding Pump Run Status Toggle Interrupt Disable/Enable bit, which disables the Pump Run Status Toggle interrupt.

Value	Description
0	The Pump Run Status Toggle interrupt is disabled.
1	The Pump Run Status Toggle interrupt is enabled.

Bit 4 – VLCDST: VLCD Status Toggle Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding VLCD Status Toggle Interrupt Disable/Enable bit, which disables the VLCD Status Toggle interrupt.

Value	Description
0	The VLCD Status Toggle interrupt is disabled.
1	The VLCD Status Toggle interrupt is enabled.

Bit 3 – VLCDRT: VLCD Ready Toggle Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding VLCD Ready Toggle Interrupt Disable/Enable bit, which disables the VLCD Ready Toggle interrupt.

Value	Description
0	The VLCD Ready Toggle interrupt is disabled.
1	The VLCD Ready Toggle interrupt is enabled.

Bit 2 – FC2O: Frame Counter 2 Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding Frame Counter 2 Overflow Interrupt Disable/Enable bit, which disables the Frame Counter 2 Overflow interrupt.

Value	Description
0	The Frame Counter 2 Overflow interrupt is disabled.
1	The Frame Counter 2 Overflow interrupt is enabled.

Bit 1 – FC1O: Frame Counter 1 Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding Frame Counter 1 Overflow Interrupt Disable/Enable bit, which disables the Frame Counter 1 Overflow interrupt.

Value	Description
0	The Frame Counter 1 Overflow interrupt is disabled.
1	The Frame Counter 1 Overflow interrupt is enabled.

Bit 0 – FC0O: Frame Counter 0 Overflow Interrupt Disable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding Frame Counter 0 Overflow Interrupt Disable/Enable bit, which disables the Frame Counter 0 Overflow interrupt.

Value	Description
0	The Frame Counter 0 Overflow interrupt is disabled.
1	The Frame Counter 0 Overflow interrupt is enabled.

43.8.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			PRST	VLCDST	VLCDRT	FC2O	FC1O	FC0O
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit 5 – PRST: Pump Run Status Toggle Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding Pump Run Status Toggle Interrupt Disable/Enable bit, which enables the Pump Run Status Toggle interrupt.

Value	Description
0	The Pump Run Status Toggle interrupt is disabled.
1	The Pump Run Status Toggle interrupt is enabled.

Bit 4 – VLCDST: VLCD Status Toggle Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding VLCD Status Toggle Interrupt Disable/Enable bit, which enables the VLCD Status Toggle interrupt.

Value	Description
0	The VLCD Status Toggle interrupt is disabled.
1	The VLCD Status Toggle interrupt is enabled.

Bit 3 – VLCDRT: VLCD Ready Toggle Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding VLCD Ready Toggle Interrupt Disable/Enable bit, which enables the VLCD Ready Toggle interrupt.

Value	Description
0	The VLCD Ready Toggle interrupt is disabled.
1	The VLCD Ready Toggle interrupt is enabled.

Bit 2 – FC2O: Frame Counter 2 Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding Frame Counter 2 Overflow Interrupt Disable/Enable bit, which enables the Frame Counter 2 Overflow interrupt.

Value	Description
0	The Frame Counter 2 Overflow interrupt is disabled.
1	The Frame Counter 2 Overflow interrupt is enabled.

Bit 1 – FC1O: Frame Counter 1 Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding Frame Counter 1 Overflow Interrupt Disable/Enable bit, which enables the Frame Counter 1 Overflow interrupt.

Value	Description
0	The Frame Counter 1 Overflow interrupt is disabled.
1	The Frame Counter 1 Overflow interrupt is enabled.

Bit 0 – FC0O: Frame Counter 0 Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding Frame Counter 0 Overflow Interrupt Disable/Enable bit, which enables the Frame Counter 0 Overflow interrupt.

Value	Description
0	The Frame Counter 0 Overflow interrupt is disabled.
1	The Frame Counter 0 Overflow interrupt is enabled.

43.8.8 Interrupt Flag

Name: INTFLAG
Offset: 0x0F
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			PRST	VLCDST	VLCDRT	FC2O	FC1O	FC0O
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit 5 – PRST: Pump Run Status Toggle
The status of the pump inside the LCD power macro is changed. Either pump start to run or stop.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Pump Run Status Toggle flag.

Bit 4 – VLCDST: VLCD Status Toggle
The status of VLCD is changed, which indicates that the relation of target VLCD and VDD33 has changed.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VLCD Status Toggle flag.

Bit 3 – VLCDRT: VLCD Ready Toggle
The status of the VLCD Ready is changed.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VLCD Ready Toggle flag.

Bit 2 – FC2O: Frame Counter 2 Overflow
This flag is set when the frame counter 2 overflows and will generate an interrupt request if the Frame Counter 2 Overflow Enable bit in Interrupt Enable Set register (INTENSET.FC2O) is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Frame Counter 2 Overflow interrupt flag.

Bit 1 – FC1O: Frame Counter 1 Overflow
This flag is set when the frame counter 1 overflows and will generate an interrupt request if the Frame Counter 1 Overflow Enable bit in Interrupt Enable Set register (INTENSET.FC1O) is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Frame Counter 1 Overflow interrupt flag.

Bit 0 – FC0O: Frame Counter 0 Overflow
This flag is set when the frame counter 0 overflows, and will generate an interrupt request if the Frame Counter 0 Overflow Enable bit in Interrupt Enable Set register (INTENSET.FC0O) is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Frame Counter 0 Overflow interrupt flag.

43.8.9 Status

Name: STATUS
Offset: 0x10
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			ABMBUSY	ACMBUSY	CMWRBUSY	VLCDS	PRUN	VLCDR
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – ABMBUSY: ABM state machine busy
This bit indicates the status of auto bit mapping.

Value	Description
0	Auto bit mapping state machine is idle.
1	Auto bit mapping state machine is active.

Bit 4 – ACMBUSY: ACM state machine busy
This bit indicates the status of auto character mapping.

Value	Description
0	Auto character mapping state machine is idle.
1	Auto character mapping state machine is active.

Bit 3 – CMWRBUSY: Character mapping write busy
This bit indicates the status of character writing function. This flag will be set to one after user write CMDATA register, and clear automatically when character data has been updated in the display memory.

Value	Description
0	Character writing function is ready for use.
1	Character writing function is busy.

Bit 2 – VLCDS: VLCD Status
This bit indicates the status of VLCD and VDD33.

Value	Description
0	Target VLCD is greater than VDD33.
1	VDD33 is greater than TARGET VLCD.

Bit 1 – PRUN: LCD Charge Pump is Running
This bit indicates the status of LCD power charge pump.

Value	Description
0	Charge pump is stopped.
1	Charge pump is running.

Bit 0 – VLCDR: VLCD Ready
This bit indicates the regulated status of VLCD.

Value	Description
0	VLCD is not well regulated to the target value.
1	VLCD is well regulated to the target value.

43.8.10 Synchronization Busy

Name: SYNCBUSY
Offset: 0x14
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLD	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLD: Control D
This bit is cleared when the synchronization of Control D register between the clock domains is complete.

This bit is set when the synchronization of Control D register between the clock domains is started.

Bit 1 – ENABLE: Enable
This bit is cleared when the synchronization of Enable bit between the clock domains is complete.

This bit is set when the synchronization of Enable bit between the clock domains is started.

Bit 0 – SWRST: Software Reset
This bit is cleared when the synchronization of Software Reset bit between the clock domains is complete.

This bit is set when the synchronization of Software Reset bit between the clock domains is started.

43.8.11 Frame Counter 0

Name: FC0
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	PB			OVF[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – PB: Prescaler Bypass
This bit enables the bypass of the frame counter 0 prescaler.

0: Prescaler is not bypassed.

1: Prescaler is bypassed.

Bits 4:0 – OVF[4:0]: Frame Counter Overflow Value
These bits define the overflow value.

The number of frame before overflow is ((OVF+1)*8) when PB=0 else (OVF+1).

43.8.12 Frame Counter 1

Name: FC1
Offset: 0x19
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	PB			OVF[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – PB: Prescaler Bypass
This bit enables the bypass of the frame counter 1 prescaler.

0: Prescaler is not bypassed.

1: Prescaler is bypassed.

Bits 4:0 – OVF[4:0]: Frame Counter Overflow Value
These bits define the overflow value.

The number of frame before overflow is ((OVF+1)*8) when PB=0 else (OVF+1).

43.8.13 Frame Counter 2

Name: FC2
Offset: 0x1A
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	PB			OVF[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – PB: Prescaler Bypass
This bit enables the bypass of the frame counter 2 prescaler.

0: Prescaler is not bypassed.

1: Prescaler is bypassed.

Bits 4:0 – OVF[4:0]: Frame Counter Overflow Value
These bits define the overflow value.

The number of frame before overflow is ((OVF+1)*8) when PB=0 else (OVF+1).

43.8.14 LCD Pin Enable Low

Name: LPENL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	LPEN[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LPEN[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LPEN[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LPEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – LPEN[31:0]: LCD Pin Enable
Each bit enables the corresponding LCD pin, from LP0 to LP31.

Value	Description
0	LCD pin is disabled.
1	LCD pin is enabled.

43.8.15 LCD Pin Enable High

Name: LPENH
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	LPEN[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LPEN[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LPEN[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LPEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – LPEN[31:0]: LCD Pin Enable
Each bit enables the corresponding LCD pin, from LP32 to LP51.

Value	Description
0	LCD pin is disabled.
1	LCD pin is enabled.

43.8.16 Segments Data Low 0

Name: SDATAL0
Offset: 0x24
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM0 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.17 Segments Data High 0

Name: SDATAH0
Offset: 0x28
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					SDATA[11:8]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – SDATA[11:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG43 to write in display memory for COM0 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.18 Segments Data Low 1

Name: SDATAL1
Offset: 0x2C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM1 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.19 Segments Data High 1

Name: SDATAH1
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					SDATA[11:8]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – SDATA[11:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG43 to write in display memory for COM1 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.20 Segments Data Low 2

Name: SDATAL2
Offset: 0x34
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM2 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.21 Segments Data High 2

Name: SDATAH2
Offset: 0x38
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					SDATA[11:8]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – SDATA[11:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG43 to write in display memory for COM2 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.22 Segments Data Low 3

Name: SDATAL3
Offset: 0x3C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM3 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.23 Segments Data High 3

Name: SDATAH3
Offset: 0x40
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					SDATA[11:8]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – SDATA[11:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG43 to write in display memory for COM3 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.24 Segments Data Low 4

Name: SDATAL4
Offset: 0x44
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM4 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.25 Segments Data High 4

Name: SDATAH4
Offset: 0x48
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							SDATA[9:8]	
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – SDATA[9:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG41 to write in display memory for COM4 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.26 Segments Data Low 5

Name: SDATAL5
Offset: 0x4C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM5 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.27 Segments Data High 5

Name: SDATAH5
Offset: 0x50
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							SDATA[9:8]	
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – SDATA[9:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG41 to write in display memory for COM5 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.28 Segments Data Low 6

Name: SDATAL6
Offset: 0x54
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM6 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.29 Segments Data High 6

Name: SDATAH6
Offset: 0x58
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – SDATA[7:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG39 to write in display memory for COM6 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.30 Segments Data Low 7

Name: SDATAL7
Offset: 0x5C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SDATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SDATA[31:0]: Segments Data
Each bit defines the segment value from SEG0 to SEG31 to write in display memory for COM7 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.31 Segments Data High 7

Name: SDATAH7
Offset: 0x60
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – SDATA[7:0]: Segments Data
Each bit defines the segment value from SEG32 to SEG39 to write in display memory for COM7 line.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.32 Indirect Segment Data

Name: ISDATA
Offset: 0x64
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			OFF[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDMASK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 21:16 – OFF[5:0]: Byte Offset
These bits defines a byte offset in display memory, refer to Display Memory Mapping.

Bits 15:8 – SDMASK[7:0]: Segments Data Mask
Each bit defines the mask for corresponding SDATA bit.

Value	Description
0	The corresponding bit is not masked and will be written in display memory.
1	The corresponding bit is masked and will not be written in display memory.

Bits 7:0 – SDATA[7:0]: Segments Data
Each bit defines the segment value to write in display memory.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.33 Blink Configuration

Name: BCFG
Offset: 0x68
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	BSS1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BSS0[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						FCS[1:0]		MODE
Access						RW	RW	RW
Reset						0	0	0

Bits 23:16 – BSS1[7:0]: Blink Segment Selection 1
Each bit enables the segment 1 (SEG1), connected to COM0 up to COM7, to blink.

Value	Description
0	Segment 1 is not enabled to blink.
1	Segment 1 is enabled to blink.

Bits 15:8 – BSS0[7:0]: Blink Segment Selection 0
Each bit enables the segment 0 (SEG0), connected to COM0 up to COM7, to blink.

Value	Description
0	Segment 0 is not enabled to blink.
1	Segment 0 is enabled to blink.

Bits 2:1 – FCS[1:0]: Frame Counter Selection
These bits select the frame counter to use for blink operation.

Value	Name	Description
0	FC0	Frame Counter 0
1	FC1	Frame Counter 1
2	FC2	Frame Counter 2

Bit 0 – MODE: Blinking Mode
This bit determines if all or a sub-set of segments are selected to blink.

Value	Name	Description
0	BLINKALL	Blink all segments
1	BLINKSEL	Blink segments selected by BSS0, BSS1

43.8.34 Circular Shift Register Configuration

Name: CSRCFG
Offset: 0x6C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIZE[3:0]					FCS[1:0]		DIR
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bits 23:8 – DATA[15:0]: Circular Shift Register Value
These bits defines the initial value of circular shift register.

Bits 7:4 – SIZE[3:0]: Circular Shift Register Size
These bits defines the size of the circular shift register which is (SIZE+1) bits long.

Bits 2:1 – FCS[1:0]: Frame Counter Selection
These bits select the frame counter to use for the circular shift register.

Value	Name	Description
0	FC0	Frame Counter 0
1	FC1	Frame Counter 1
2	FC2	Frame Counter 2

Bit 0 – DIR: Direction
This bit select the shift direction.

Value	Description
0	Shifting to the left
1	Shifitng to the right

43.8.35 Character Mapping Configuration

Name: CMCFG
Offset: 0x70
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					DEC	NSEG[2:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit 3 – DEC: Decrement SEG Line Index

Bits 2:0 – NSEG[2:0]: Number of SEG lines
These bits define the number of SEG line per digit.

NSEG = number of SEG line - 1.

43.8.36 Automated Character Mapping Configuration

Name: ACMCFG
Offset: 0x74
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FCS[1:0]		STSEG[5:0]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MODE		NDROW[5:0]					
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	STEPS[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NDIG[3:0]					NCOM[2:0]		
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bits 31:30 – FCS[1:0]: Frame Counter Selection
These bits select the frame counter to use for automated character mapping.

Value	Name	Description
0	FC0	Frame Counter 0
1	FC1	Frame Counter 1
2	FC2	Frame Counter 2

Bits 29:24 – STSEG[5:0]: Start SEG Line
These bits define the index of the first segment terminal of the digit to display.

Bit 23 – MODE: Mode
This bit selects the display mode.

Value	Name	Description
0	SEQ	Sequential Display Mode
1	SCROLL	Scrolling Display Mode

Bits 21:16 – NDROW[5:0]: Number of Digit per Row
These bits define the number of digit per row.

Bits 15:8 – STEPS[7:0]: Scrolling Steps
These bits define the number of steps in scrolling mode.

STEPS = character string length - NDIG + 1

Bits 7:4 – NDIG[3:0]: Number of Digit
These bits define the number of digit used (must be greater than 1).

Bits 2:0 – NCOM[2:0]: COM Lines per Row
These bits define the number of COM line per row.

NCOM = number of COM line - 1.

43.8.37 Automated Bit Mapping Configuration

Name: ABMCFG
Offset: 0x78
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	SIZE[5:0]						FCS[1:0]	
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:2 – SIZE[5:0]: Size
These bits define the number of DMA writes to ISDATA register to transfer the state of a symbol (must be greater than 1).

Bits 1:0 – FCS[1:0]: Frame Counter Selection
These bits select the frame counter to use for automated bit mapping.

Value	Name	Description
0	FC0	Frame Counter 0
1	FC1	Frame Counter 1
2	FC2	Frame Counter 2

43.8.38 Character Mapping Segments Data

Name: CMDATA
Offset: 0x7C
Reset: 0x00000000
Property: W

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – SDATA[23:0]: Segments Data
Each bit defines the segment value to write in display memory.

Value	Description
0	Segment is not energized (transparent).
1	Segment is energized (opaque).

43.8.39 Character Mapping Data Mask

Name: CMDMASK
Offset: 0x80
Reset: 0x0000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SDMASK[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDMASK[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDMASK[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – SDMASK[23:0]: Segments Data Mask
Each bit defines the mask for corresponding bit in CMDATA register.

Value	Description
0	The corresponding bit is not masked and will be written in display memory.
1	The corresponding bit is masked and will not be written in display memory.

43.8.40 Character Mapping Index

Name: CMINDEX
Offset: 0x84
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
						CINDEX[2:0]		
Access						RW	RW	RW
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			SINDEX[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 10:8 – CINDEX[2:0]: COM Line Index
These bits define COM line index of the character to be updated.

Bits 5:0 – SINDEX[5:0]: SEG Line Index
These bits define SEG line index of the character to be updated.

44. PTC - Peripheral Touch Controller

44.1 Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self- and mutual-capacitance sensors.

In mutual-capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

The number of available pins and the assignment of X- and Y-lines is depending on both package type and device configuration. Refer to the Configuration Summary and I/O Multiplexing table for details.

Related Links

I/O Multiplexing and Considerations

44.2 Features

- Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, wheels
- Supports wake-up on touch from standby sleep mode
- Supports mutual capacitance and self-capacitance sensing
  - Mix-and-match mutual-and self-capacitance sensors
- One pin per electrode – no external components
- Load compensating charge sensing
  - Parasitic capacitance compensation and adjustable gain for superior sensitivity
- Zero drift over the temperature and V_{DD} range
  - Auto calibration and re-calibration of sensors
- Single-shot and free-running charge measurement
- Hardware noise filtering and noise signal de-synchronization for high conducted immunity
- Selectable channel change delay allows choosing the settling time on a new channel, as required
- Acquisition-start triggered by command or through auto-triggering feature
- Low CPU utilization through interrupt on acquisition-complete
- Supported by the QTouch® Composer development tools. See also Atmel|START and Atmel Studio documentation.

Related Links

I/O Multiplexing and Considerations

44.3 Block Diagram

Figure 44-1. PTC Block Diagram Mutual-Capacitance

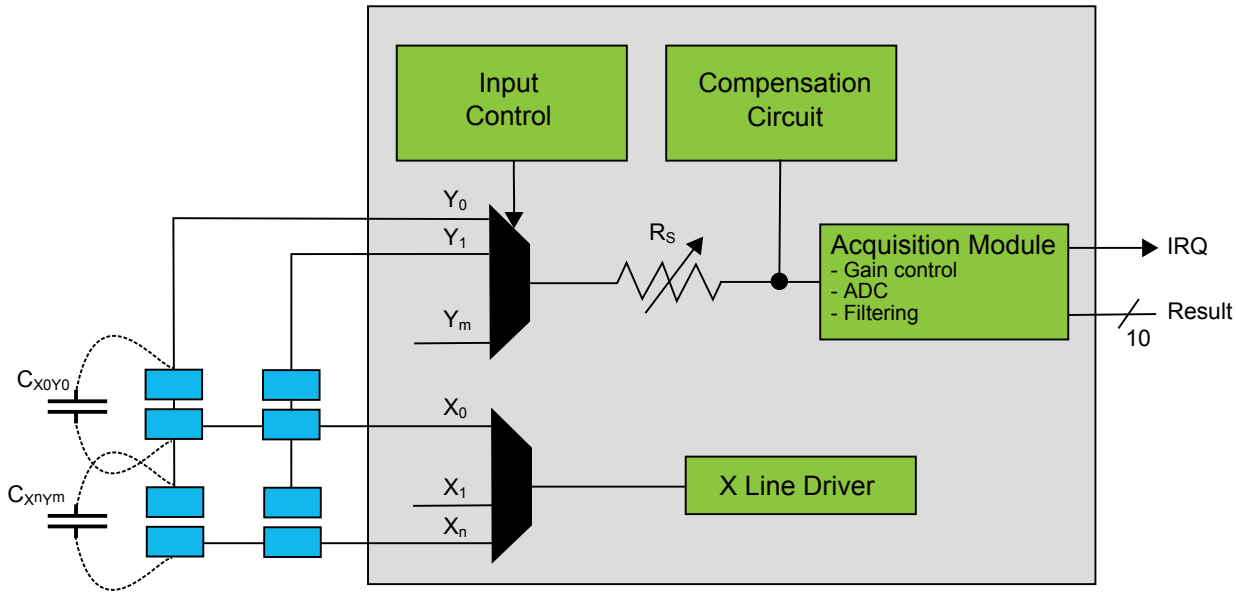
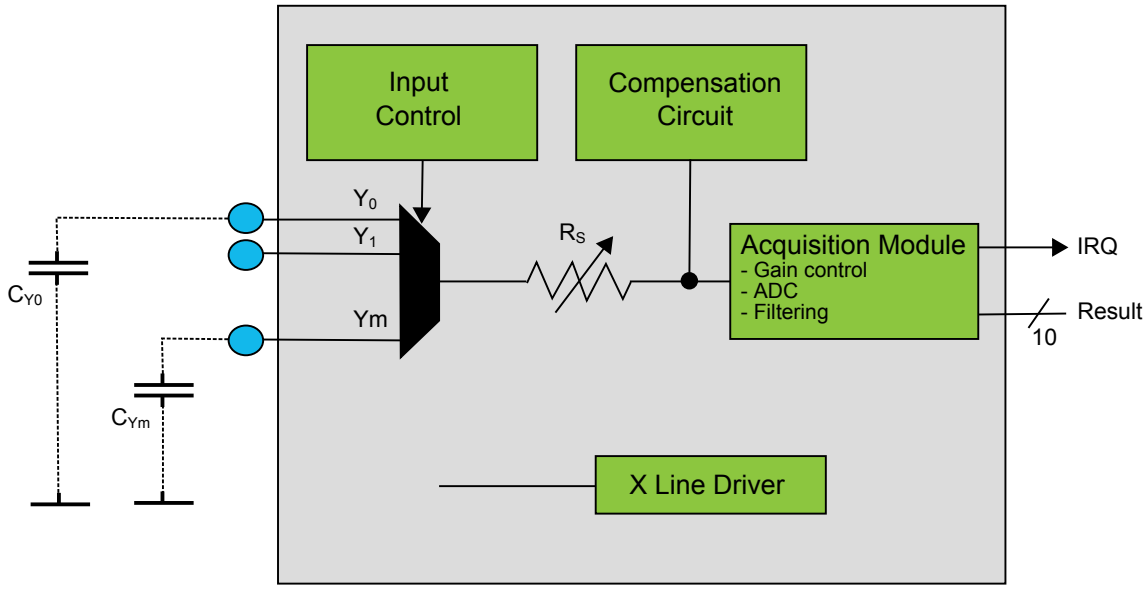


Figure 44-2. PTC Block Diagram Self-Capacitance



44.4 Signal Description

Table 44-1. Signal Description for PTC

Name	Type	Description
Y[m:0]	Analog	Y-line (Input/Output)
X[n:0]	Digital	X-line (Output)

Note: The number of X and Y lines are device dependent. Refer to *Configuration Summary* for details.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations

44.5 Product Dependencies

In order to use this Peripheral, configure the other components of the system as described in the following sections.

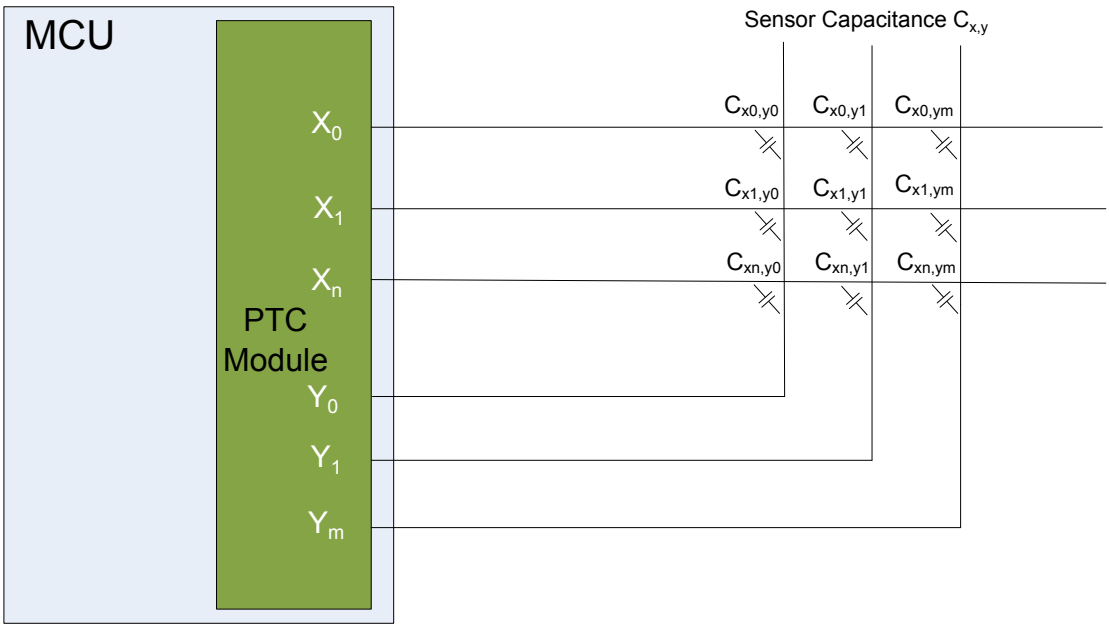
44.5.1 I/O Lines

The I/O lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1kΩ or more can be used on X-lines and Y-lines.

44.5.1.1 Mutual-Capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for sensing. The mutual capacitance between the X and Y electrode is measured by the Peripheral Touch Controller.

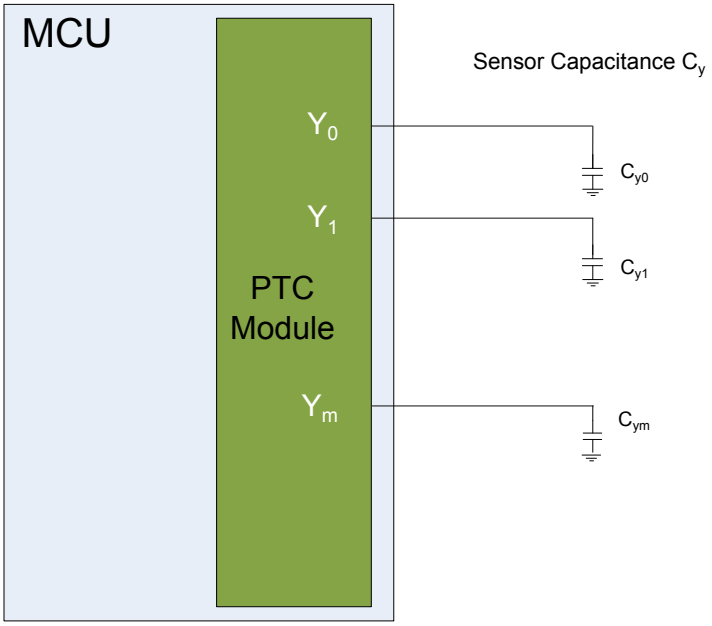
Figure 44-3. Mutual Capacitance Sensor Arrangement



44.5.1.2 Self-Capacitance Sensor Arrangement

A self-capacitance sensor is connected to a single pin on the Peripheral Touch Controller through the Y electrode for sensing the signal. The sense electrode capacitance is measured by the Peripheral Touch Controller.

Figure 44-4. Self-capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide on http://www.atmel.com.

44.5.2 Clocks

The PTC is clocked by the GCLK_PTC clock. The PTC operates from an asynchronous clock source and the operation is independent of the main system clock and its derivative clocks, such as the peripheral bus clock (CLK_APB). A number of clock sources can be selected as the source for the asynchronous GCLK_PTC. The clock source is selected by configuring the Generic Clock Selection ID in the Generic Clock Control register. For more information about selecting the clock sources, refer to *GCLK - Generic Clock Controller*.

The selected clock must be enabled in the Power Manager, before it can be used by the PTC. By default these clocks are disabled. The frequency range of GCLK_PTC is 400kHz to 4MHz.

Related Links

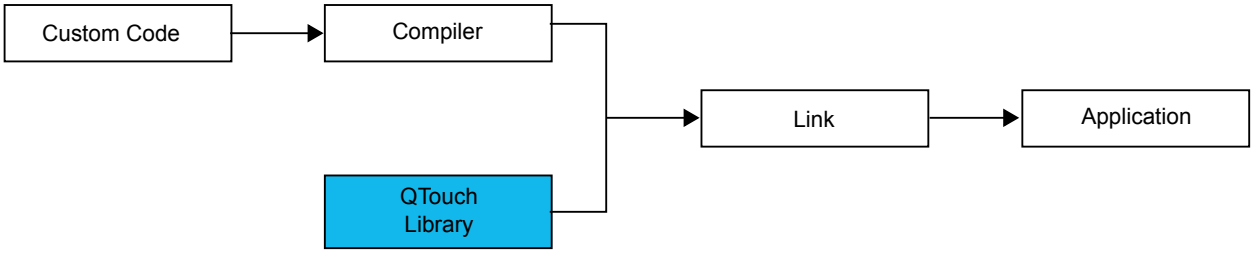
GCLK - Generic Clock Controller

PM – Power Manager

44.6 Functional Description

In order to access the PTC, the user must use the QTouch Composer tool to configure and link the QTouch Library firmware with the application software. QTouch Library can be used to implement buttons, sliders, wheels in a variety of combinations on a single interface.

Figure 44-5. QTouch Library Usage



For more information about QTouch Library, refer to the QTouch Library Peripheral Touch Controller User Guide.

45. Electrical Characteristics

45.1 Disclaimer

All typical values are measured at T = 25°C unless otherwise specified. All minimum and maximum values are valid across operating temperature and voltage unless otherwise specified.

45.2 Absolute Maximum Ratings

Stresses beyond those listed in this section may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 45-1. Absolute Maximum Ratings

Symbol	Description	Min.	Max.	Units
V_{DD}	Power supply voltage	0	3.8	V
I_{VDD}	Current into a V_{DD} pin	-	92	mA
I_{GND}	Current out of a GND pin	-	130	mA
V_{PIN}	Pin voltage with respect to GND and V_{DD}	GND-0.6	V_{DD}+0.6V	V
$T_{storage}$	Storage temperature	-60	150	°C

Note:
Maximum source current is 46mA and maximum sink current is 65mA per cluster. A cluster is a group of GPIOs as shown in section GPIO Pin Clusters.

Also note that each VDD/GND pair is connected to two clusters, so current consumption through the pair will be a sum of the clusters' source/sink currents.

Caution: This device is sensitive to electrostatic discharges (ESD). Improper handling may lead to permanent performance degradation or malfunctioning.
Handle the device following best practice ESD protection rules: Be aware that the human body can accumulate charges large enough to impair functionality or destroy the device.

45.3 General Operating Ratings

The device must operate within the ratings listed in this section in order for all other electrical characteristics and typical characteristics of the device to be valid.

Table 45-2. General Operating Conditions

Symbol	Description	Min.	Typ.	Max.	Units
V_{DDIO}	IO Supply Voltage	1.62	3.3	3.63	V
V_{DDANA}	Analog supply voltage	1.62	3.3	3.63	V
T_A	Temperature range	-40	25	85	°C
T_J	Junction temperature	-	-	100	°C

Note: The same voltage must be applied to both V_{DDIO} and V_{DDANA}. This common voltage is referred to as V_{DD} in the datasheet.

Caution: In debugger cold-plugging mode, NVM erase operations are not protected by the BOD33 and BOD12. NVM erase operation at supply voltages below specified minimum can cause corruption of NVM areas that are mandatory for correct device behavior.

Related Links

Junction Temperature

45.4 Supply Characteristics

Table 45-3. Supply Characteristics

Symbol	Voltage		
	Min.	Max.	Units
V_{DDIO}	1.62	3.63	V
V_{DDANA}	1.62	3.63	V
V_{BAT}	1.62	3.63	V

Table 45-4. Supply Slew Rates(1)

Symbol	Fall Rate	Rise Rate	Units
	Max.	Max.	
V_{DDIO}	0.05	0.1	V/µs
V_{DDANA}	0.05	0.1	V/µs
V_{BAT}	0.05	0.1	V/µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

45.5 Maximum Clock Frequencies

Table 45-5. Maximum GCLK Generator Output Frequencies

Symbol	Description	F_{max} [MHz]		Units
		PL0	PL2	
$F_{gclkgen}$[2:0]	GCLK Generator output Frequency	24	96	MHz
$F_{gclkgen}$[4:3]		16	66	MHz

Table 45-6. Maximum Peripheral Clock Frequencies

Symbol	Description	Max.		Units
		PL0	PL2	
f_{CPU}	CPU clock frequency	8	32	MHz
f_{AHB}	AHB clock frequency	8	32	MHz
f_{APBA}	APBA clock frequency / Bus clock domain = BACKUP	8	8	MHz
f_{APBA}	APBA clock frequency / Bus clock domain = Low Power	8	32	MHz
f_{APBB}	APBB clock frequency	8	32	MHz
f_{APBC}	APBC clock frequency	8	32	MHz
$f_{GCLK_DFLL48M_REF}$	DFLL48M Reference clock frequency	NA	33	KHz
f_{GCLK_DPLL}	FDPLL96M Reference clock frequency	2	2	MHz
$f_{GCLK_DPLL_32K}$	FDPLL96M 32k Reference clock frequency	32	32	KHz
f_{GCLK_EIC}	EIC input clock frequency	12	48	MHz
$f_{GCLK_FREQM_MSR}$	FREQM Measure	12	48	MHz
$f_{GCLK_FREQM_REF}$	FREQM Reference	12	48	MHz
f_{GCLK_USB}	USB input clock frequency	NA	48	MHz
$f_{GCLK_EVSYS_CHANNEL_0}$	EVSYS channel 0 input clock frequency	12	48	MHz
$f_{GCLK_EVSYS_CHANNEL_1}$	EVSYS channel 1 input clock frequency			
$f_{GCLK_EVSYS_CHANNEL_2}$	EVSYS channel 2 input clock frequency			
$f_{GCLK_EVSYS_CHANNEL_3}$	EVSYS channel 3 input clock frequency			
$f_{GCLK_EVSYS_CHANNEL_4}$	EVSYS channel 4 input clock frequency			
$f_{GCLK_EVSYS_CHANNEL_5}$	EVSYS channel 5 input clock frequency			
$f_{GCLK_EVSYS_CHANNEL_6}$	EVSYS channel 6 input clock frequency			
$f_{GCLK_EVSYS_CHANNEL_7}$	EVSYS channel 7 input clock frequency			
$f_{GCLK_SERCOMx_SLOW}$	Common SERCOM slow input clock frequency	1	5	MHz
$f_{GCLK_SERCOM0_CORE}$	SERCOM0 input clock frequency	12	48	MHz

Symbol	Description	Max.		Units
		PL0	PL2	
$f_{GCLK_SERCOM1_CORE}$	SERCOM1 input clock frequency			
$f_{GCLK_SERCOM2_CORE}$	SERCOM2 input clock frequency			
$f_{GCLK_SERCOM3_CORE}$	SERCOM3 input clock frequency			
$f_{GCLK_SERCOM4_CORE}$	SERCOM4 input clock frequency			
$f_{GCLK_SERCOM5_CORE}$	SERCOM5 input clock frequency			
f_{GCLK_TCC0}	TCC0 input clock frequency	24	96	MHz
$f_{GCLK_TC0, GCLK_TC1}$	TC0,TC1 input clock frequency	12	48	MHz
$f_{GCLK_TC2, GCLK_TC3}$	TC2,TC3 input clock frequency			
f_{GCLK_ADC}	ADC input clock frequency	12	48	MHz
f_{GCLK_AC}	AC digital input clock frequency			
f_{GCLK_PTC}	PTC input clock frequency			
f_{GCLK_CCL}	CCL input clock frequency			
f_{GCLKin}	External GCLK clock input			

45.6 Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating conditions**
  - CPU is running on flash with 1 wait states for V_{DD}=3.3V and 3 wait states for V_{DD}=1.8V
  - Low power cache is enabled.
  - BOD33 disabled
  - I/Os are inactive input mode with Input trigger disable
- **Oscillators**
  - XOSC (crystal oscillator) stopped
  - XOSC32K (32kHz crystal oscillator) running with external 32kHz crystal
  - When in active PL2 mode FDPLL using XOSC32K as reference and running at 32MHz
  - When in active PL0 internal Multi RC Oscillator running at 8MHz or 4MHz

Table 45-7. Active Current Consumption

Mode	Conditions	Regulator	PL	Clock	V_{DD}	Typ.	Max.	Units
ACTIVE	Coremark	LDO Mode	PL0	OSC 8MHz	1.8V	85	126	µA/MHz
					3.3V	90	131	
				OSC 4MHz	1.8V	94	181	
					3.3V	100	188	
			PL2	OSC 12MHz	1.8V	101	149	

Mode	Conditions	Regulator	PL	Clock	V_{DD}	Typ.	Max.	Units
					3.3V	106	151	
				FDPLL 32MHz	1.8V	87	100	
					3.3V	103	116	
		BUCK Mode	PL0	OSC 8MHz	1.8V	55	81	
					3.3V	41	57	
				OSC 4MHz	1.8V	62	116	
					3.3V	49	84	
			PL2	OSC 12MHz	1.8V	79	110	
					3.3V	52	76	
				FDPLL 32MHz	1.8V	63	75	
					3.3V	46	53	
ACTIVE	FIBO	LDO Mode	PL0	OSC 8MHz	1.8V	69	111	µA/MHz
					3.3V	71	113	
				OSC 4MHz	1.8V	78	166	
					3.3V	81	170	
			PL2	OSC 12MHz	1.8V	85	130	
					3.3V	87	132	
				FDPLL 32MHz	1.8V	83	96	
					3.3V	83	96	
		BUCK Mode	PL0	OSC 8MHz	1.8V	44	71	
					3.3V	31	49	
				OSC 4MHz	1.8V	52	107	
					3.3V	39	76	
			PL2	OSC 12MHz	1.8V	66	103	
					3.3V	42	63	
				FDPLL 32MHz	1.8V	60	71	
					3.3V	35	42	
ACTIVE	While 1	LDO Mode	PL0	OSC 8MHz	1.8V	51	95	µA/MHz
					3.3V	53	97	
				OSC 4MHz	1.8V	60	151	
					3.3V	62	154	
			PL2	OSC 12MHz	1.8V	60	109	

Mode	Conditions	Regulator	PL	Clock	V_{DD}	Typ.	Max.	Units
					3.3V	62	111	
				FDPLL 32MHz	1.8V	56	75	
					3.3V	59	75	
		BUCK Mode	PL0	OSC 8MHz	1.8V	33	61	
					3.3V	25	43	
				OSC 4MHz	1.8V	41	97	
					3.3V	33	70	
			PL2	OSC 12MHz	1.8V	47	84	
					3.3V	31	54	
				FDPLL 32MHz	1.8V	43	55	
					3.3V	25	34	
IDLE2		BUCK	PL0	OSC 8MHz	1.8V	14	31	µA/MHz
					3.3V	12	23	

Table 45-8. Current Consumption - Standby Mode(1)

Mode	conditions	Regulator Mode	V_{DD}	Ta	Typ.	Max.	Units
STANDBY	No interface running	LPEFF Disable	1.8V	25°C	1.8	7.6	µA
				85°C	43.2	170.4	
		LPEFF Enable	3.3V	25°C	1.6	6.9	
				85°C	31.5	103.4	
	RTC running on XOSC32K	LPEFF Disable	1.8V	25°C	2.2	8.1	
				85°C	43.7	171.4	
		LPEFF Enable	3.3V	25°C	2.0	7.3	
				85°C	32.0	104.1	
	RTC running on ULP32K	LPEFF Disable	1.8V	25°C	1.9	8.0	
				85°C	43.4	170.9	
		LPEFF Enable	3.3V	25°C	1.7	7.0	
				85°C	31.6	103.6	

Table 45-9. Current Consumption - BACKUP and OFF Mode

Mode	conditions	V_{DD}	Ta	Typ.	Max.	Units
BACKUP	powered by V_{DDIO}, V_{DDANA}+V_{DDIO} consumption	1.8V	25°C	0.47	0.79	µA
			85°C	4.4	8.6	
		3.3V	25°C	0.55	0.9	

Mode	conditions	V_{DD}	Ta	Typ.	Max.	Units
			85°C	5.7	9.9	
	powered by V_{DDIO}, V_{BAT} consumption	1.8V	25°C	0.000	0.00	
			85°C	0.013	0.009	
		3.3V	25°C	0.000	0.00	
			85°C	0.026	0.018	
	powered by V_{DDIO} with RTC running on OSCULP32K, V_{DDANA} +V_{DDIO} consumption	1.8V	25°C	0.55	0.88	
			85°C	4.4	8.7	
		3.3V	25°C	0.63	1.0	
			85°C	5.7	10.0	
	powered by V_{DDIO} with RTC running on OSCULP32K, V_{BAT} consumption	1.8V	25°C	0.000	0.00	
			85°C	0.008	0.009	
		3.3V	25°C	0.000	0.00	
			85°C	0.015	0.018	
BACKUP	powered by V_{BAT}, V_{DDANA}+V_{DDIO} consumption	1.8V	25°C	0.08	0.09	µA
			85°C	2.0	2.2	
		3.3V	25°C	0.14	0.2	
			85°C	3.2	3.5	
	powered by V_{BAT}, V_{BAT} consumption	1.8V	25°C	0.38	0.69	
			85°C	2.3	6.3	
		3.3V	25°C	0.41	0.73	
			85°C	2.4	6.4	
	powered by V_{BAT} with RTC running on OSCULP32K, V_{DDANA} +V_{DDIO} consumption	1.8V	25°C	0.08	0.09	
			85°C	2.1	2.3	
		3.3V	25°C	0.14	0.1	
			85°C	3.2	3.5	
	powered by V_{BAT} with RTC running on OSCULP32K, V_{BAT} consumption	1.8V	25°C	0.46	0.77	
			85°C	2.4	6.4	
		3.3V	25°C	0.49	0.81	
			85°C	2.5	6.6	
OFF		1.8V	25°C	0.10	0.18	µA
			85°C	2.3	4.5	

Mode	conditions	V_{DD}	Ta	Typ.	Max.	Units
		3.3V	25°C	0.18	0.39	
			85°C	3.6	7.5	

Related Links

Junction Temperature

45.7 Wake-up Timing

Conditions:

- V_{DD} = 3.3V
- LDO Regulation mode
- CPU clock = OSC16M @12MHz
- 1 Wait-state
- Cache enabled
- Flash Fast Wake up enabled (NVMCTRL->CTRLB.FWUP=1)
- Flash in WAKEUPINSTANT mode (NVMCTRL->CTRLB.SLEEPPRM=1)

Measure method:

- For IDLE and STANDBY, CPU sets an IO by writing PORT->IOBUS without jumping in an interrupt handler (Cortex M0+ register PRIMASK=1). The wake-up time is measured between the falling edge of the input signal and to the rising edge of the GPIO pin.
- For Backup, the exit of mode is done through reset, the set of the IO is done by the first executed instructions after reset. For OFF mode, the exit of mode is done through reset pin, the time is measured between the rising edge of the RESETN signal and the set of the IO done by the first executed instructions after reset.
- For OFF mode, the exit of mode is done through reset pin, the time is measured between the rising edge of the RESETN signal and the set of the IO done by the first executed instructions after reset.

Table 45-10. Wake-up Timings

Sleep Mode	Condition	Typ.	Unit
IDLE	PL2 or PL0	1	µs
STANDBY	PL0 and PM.PLSEL.PLDIS=1	2.5	µs
STANDBY	PL2 and Voltage scaling at default values: SUPC->VREG.VSVSTEP=0 SUPC->VREG.VSPER=0	70	µs
STANDBY	PL2 and Voltage scaling at fastest setting: SUPC->VREG.VSVSTEP=15 SUPC->VREG.VSPER=0	15	µs
BACKUP		90	µs
OFF		2500	µs

45.8 IO Pin Characteristics

Three are three different pin types with three different speeds: Backup, Normal, and High Sink (3,4,5). The Drive Strength bit is located in the Pin Configuration register PORT (PORT.PINCFG.DRVSTR).

Table 45-11. I/O Pins Common Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
V_{IL}	Input low-level voltage	V_{DD}=1.62V-2.7V	-	-	0.25*V_{DD}	V
		V_{DD}=2.7V-3.63V	-	-	0.3*V_{DD}	
V_{IH}	Input high-level voltage	V_{DD}=1.62V-2.7V	0.7*V_{DD}	-	-	
		V_{DD}=2.7V-3.63V	0.55*V_{DD}	-	-	
V_{OL}	Output low-level voltage	V_{DD}>1.6V, I_{OL} max	-	0.1*V_{DD}	0.2*V_{DD}	
V_{OH}	Output high-level voltage	V_{DD}>1.6V, I_{OH} max	0.8*V_{DD}	0.9*V_{DD}	-	
R_{PULL}	Pull-up - Pull-down resistance	All pins excepted PA24, PA25	20	40	60	kΩ
		PA24, PA25 (1)	50	100	150	
I_{LEAK}	Input leakage current	Pull-up resistors disabled	-1	+/-0.015	1	µA

Table 45-12. I/O Pins Maximum Output Current

Symbol	Parameter	Conditions	Backup pins in Backup mode.	Backup and Normal pins	High Sink pins	Backup and Normal pins	High Sink pins	Units
				DRVSTR=0		DRVSTR=1		
I_{OL}	Maximum Output low-level current	V_{DD}=1.62V-3V	0.005	1	2	2	4	mA
		V_{DD}=3V-3.63V	0.008	2.5	6	6	12	
I_{OH}	Maximum Output high-level current	V_{DD}=1.62V-3V	0.005	0.7	1.5	1.5	3	
		V_{DD}=3V-3.63V	0.008	2	5	5	10	

Table 45-13. I/O Pins Dynamic Characteristics(1)

Symbol	Parameter	Conditions	Backup pins in Backup mode.	Backup and Normal pins	High Sink pins	Backup and Normal pins	High Sink pins	Units
				DRVSTR=0		DRVSTR=1		
t_{RISE}	Maximum Rise time	Load = 20pF, V_{DD} = 3.3V	2000	13	6	6	4.5	ns
t_{FALL}	Maximum Fall time	Load = 20pF, V_{DD} = 3.3V	2000	12	7	7	4.5	

The pins with I^2C alternative mode available are compliance(2) with I^2C norms. All I^2C pins support Standard (Sm), Fast (Fm) and Fast plus mode (Fm+). The PB30 and PB31 pins support the High speed mode (Hs, up to 3.4MHz). The available I^2C pins are listed in the I/O Multiplexing section.

Note:

1. These values are based on simulation. These values are not covered by test limits in production or characterization
2. The pins PB12, PB13 are limited on output low-level current in I2C standard mode (Sm) and Fast mode (Fm). The limitation is 2.5mA instead of 3mA for a VOL = 0.4V, and 3mA instead of 6mA for a VOL = 0.6V.
3. The following pins are High Sink pins and have different properties than normal pins: PA12, PA13, PA22, PA23, PA27, PA31, PB30, PB31.
4. The following pins are Backups pins and have different properties than normal pins: PA00, PA01, PB00, PB01, PB02, PB03, PC00, PC01.
5. The following digital pins are only a input pins: PC08, PC09, PC10, PC11, PC18, PC19, PC20, PC21.

45.9 Injection Current

Stresses beyond those listed in the table below may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 45-14. Injection Current(1,2)

Symbol	Description	min	max	Unit
I_{inj1} (3)	IO pin injection current	-1	+1	mA
I_{inj2} (4)	IO pin injection current	-15	+15	mA
$I_{injtotal}$	Sum of IO pins injection current	-45	+45	mA

Note:

1. Injecting current may have an effect on the accuracy of Analog blocks
2. Injecting current on Backup IOs is not allowed
3. Conditions for V_{pin}: $V_{pin} < GND-0.6V$ or $3.6V<V_{pin}\leq 4.2V$.

   Conditions for V_{DD}: $3V<V_{DD}\leq 3.6V$.

   If V_{pin} is lower than GND-0.6V, a current limiting resistor is required. The negative DC injection current limiting resistor R is calculated as $R = |(GND-0.6V - V_{pin})/I_{inj1}|$.

   If V_{pin} is greater than V_{DD}+0.6V, a current limiting resistor is required. The positive DC injection current limiting resistor R is calculated as $R = (V_{pin}-(V_{DD}+0.6))/I_{inj1}$.
4. Conditions for V_{pin}: $V_{pin} < GND-0.6V$ or $V_{pin}\leq 3.6V$.

   Conditions for V_{DD}: $V_{DD}\leq 3V$.

   If V_{pin} is lower than GND-0.6V, a current limiting resistor is required. The negative DC injection current limiting resistor R is calculated as $R = |(GND-0.6V - V_{pin})/I_{inj2}|$.

   If V_{pin} is greater than V_{DD}+0.6V, a current limiting resistor is required. The positive DC injection current limiting resistor R is calculated as $R = (V_{pin}-(V_{DD}+0.6))/I_{inj2}$.

45.10 Analog Characteristics

45.10.1 Voltage Regulator Characteristics

45.10.1.1 Buck Converter

Table 45-15. Buck converter Electrical Characteristics

Symbol	Parameter	Conditions	Typ.	Units
P_{EFF}(2)	Power Efficiency	I_{OUT} = 5mA	86	%
		I_{OUT} = 50mA	85	%
VREGSCAL(1)	Voltage scaling	min step size for PLx to PLy transition	5	mV
		Voltage Scaling Period	1	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization

Table 45-16. External Components requirements in switching mode(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
C_{IN}	Input regulator capacitor		-	4.7	-	µF
		Ceramic dielectric	-	100	-	nF
C_{OUT}	Output regulator capacitor		-	1	-	µF
		Ceramic dielectric	-	100	-	nF
L_{EXT}	External inductance	Murata LQH3NPN100MJ0	-	10	-	µH
$R_{SERIE_L_{EXT}}$	Serial resistance of L_{ext}	-	-	-	0.7	Ω
$I_{SAT_L_{EXT}}$	Saturation current	-	275	-	-	mA

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

45.10.1.2 LDO Regulator

Table 45-17. LDO Regulator Electrical Characteristics

Symbol	Parameter	Conditions	Typ.	Units
VREGSCAL	Voltage scaling	min step size for PLx to Ply transistion	5	mV
		Voltage Scaling Period	1	µs

Table 45-18. External Components requirements in linear mode

Symbol	Parameter	Conditions	Typ.	Units
C_{IN}	Input regulator capacitor		4.7	µF
		Ceramic dielectric X7R	100	nF
C_{OUT}	Output regulator capacitor		1	µF
		Ceramic dielectric X7R	100	nF

45.10.2 APWS

Table 45-19. Automatic Power Switch Characteristics

Symbol	Parameters	Typ.	Unit
CD	Decoupling capacitor on VDDIN	4.7	µF
T_{HUP}	V_{DD} threshold	1.84	V
T_{HDWN}		1.75	V
T_{HHYS}	V_{DD} hysteresis	90	mV

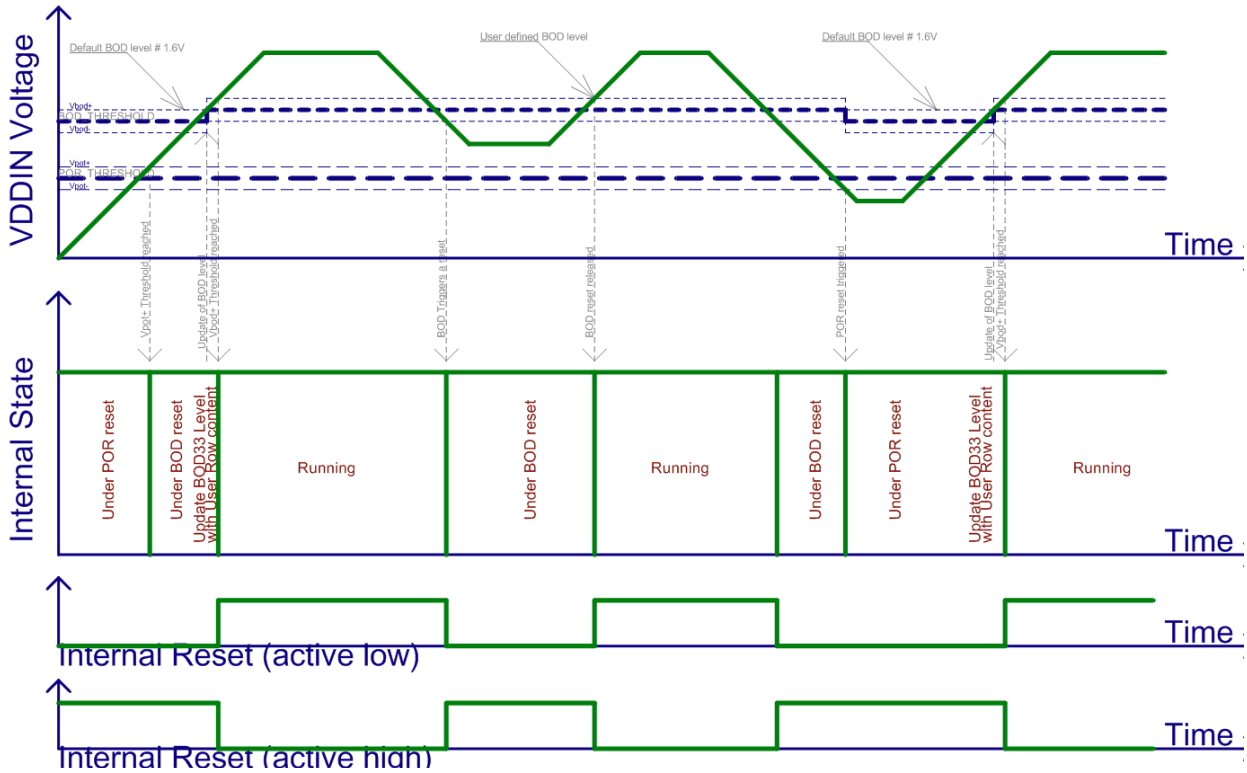
45.10.3 Power-On Reset (POR) Characteristics

Table 45-20. POR33 Characteristics

Symbol	Parameters	Min.	Typ.	Max.	Unit
V_{POT+}	Voltage threshold Level on V_{DD} rising	1.53	1.58	1.62	V
V_{POT-}	Voltage threshold Level on V_{DD} falling	0.6	1.04	1.39	V

Note: These values are based on simulation. They are not covered by production test limits or characterization.

Figure 45-1. BOD Reset Behavior at Startup and Default Levels



45.10.4 BOD33

Table 45-21. BOD33 Characteristics(1)

Symbol	Parameters	Conditions	Min	Typ.	Max	Unit
V_{BOD+}	BOD33 high threshold Level	V_{BAT} level, Bod setting = 15	1.67	1.74	1.81	V
		V_{DD} level, Bod setting = 7	1.75	1.75	1.80	
		V_{DD} level, Bod setting = 6 (default)	1.66	1.72	1.75	
		V_{BAT} level, Bod setting = 55	2.80	2.90	3.01	
		V_{DD} level, Bod setting = 39	2.65	2.87	2.95	
		V_{BAT} level, Bod setting = 63	3.02	3.14	3.26	
		V_{DD} level, Bod setting = 48	3.12	3.20	3.29	
V_{BOD-} / V_{BOD}	BOD33 low threshold Level	V_{BAT} level, Bod setting = 15	1.60	1.66	1.72	V
		V_{DD} level, Bod setting = 7	1.63	1.67	1.71	
		V_{DD} level, Bod setting = 6	1.60	1.65	1.68	
		V_{BAT} level, Bod setting = 55	2.70	2.81	2.92	
		V_{DD} level, Bod setting = 39	2.70	2.77	2.84	
		V_{BAT} level, Bod setting = 63	2.92	3.04	3.16	
		V_{DD} level, Bod setting = 48	3.00	3.08	3.16	
	Step size	-	-	34	-	mV

Symbol	Parameters	Conditions	Min	Typ.	Max	Unit
V_{HYS}	Hysteresis (V_{BOD+} - V_{BOD-}) BOD33.LEVEL = 0x0 to 0x3F	V_{BAT}	38	-	135	mV
		V_{DD}	47	-	155	mV
T_{START}	Start-up time	Timing from enable to RDY	-	3.2	-	µs

Note:

1. These values are based on characterization.

Table 45-22. Power Consumption(1)

Symbol	Parameters	V_{DD} (V)	Ta	Typ.	Max.	Units
I_{DD}	IDLE, Mode CONT	1.8	Max. 85°C Typ. 25°C	17.9	21.5	µA
		3.3		28.8	33.1	
	IDLE, Mode SAMPL	1.8		0.020	0.306	µA
		3.3		0.033	0.197	
	STDBY, Mode SAMPL	1.8		0.087	0.231	µA
		3.3		0.114	0.347	

Note:

1. These values are based on characterization.

45.10.5 Analog-to-Digital (ADC) Characteristics

Table 45-23. Operating Conditions

Symbol	Parameters	Conditions		Min.	Typ.	Max.	Unit
RES	Resolution			-	-	12	bits
R_S	Sampling rate			10	-	1000	kSPS
fs	Sampling clock			10	-	1000	kHz
	Differential mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=1	resolution 12bit (RESEL=0)			16		cycles
		resolution 10bit (RESEL=2)			14		
		resolution 8bit (RESEL=3)			12		
	Differential mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=0	SAMLPLEN corresponds to the decimal value of the SAMPLEN[5:0] register	resolution 12bit (RESEL=0)		SAMPLEN + 13		cycles
			resolution 10bit (RESEL=2)		SAMPLEN + 11		
			resolution 8bit (RESEL=3)		SAMPLEN + 9		
	Single-ended mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=1	resolution 12bit (RESEL=0)			16		cycles
		resolution 10bit (RESEL=2)			15		

Symbol	Parameters	Conditions		Min.	Typ.	Max.	Unit
		resolution 8bit (RESEL=3)			13		
	Single-ended mode Number of ADC clock cycles SAMPCTRL.OFFCOMP=0	SAMLPLEN corresponds to the decimal value of the SAMPLEN[5:0] register	resolution 12bit (RESEL=0)		SAMPLEN + 13		cycles
			resolution 10bit (RESEL=2)		SAMPLEN + 12		
			resolution 8bit (RESEL=3)		SAMPLEN + 10		
fadc	ADC Clock frequency			160	-	16000	kHz
T_S	Sampling time			250	(SAMPLEN +1)/fadc	25000	ns
	Conversion range	Differential mode		$-V_{REF}$	-	V_{REF}	V
		Single-Ended mode		0	-	V_{REF}	
V_{REF}	Reference input	REFCOMP=1		1	-	V_{DDANA}-0.6	V
		REFCOMP=0		V_{DDANA}	-	V_{DDANA}	
V_{IN}	Input channel range	-		0	-	V_{DDANA}	V
V_{CMIN}	Input common mode voltage	For V_{REF} > 1.0V		0.7	-	V_{REF}-0.7	V
		For V_{REF}=1.0V		0.3	-	V_{REF}-0.3	
CSAMPLE(1)	Input sampling capacitance			-	2.8	3.2	pF
RSAMPLE(1)	Input channel source resistance			-	-	1715	Ω
R_{ref}(1)	Reference input source resistance	REFCOMP=1		-	-	5	kΩ

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 45-24. Power Consumption(1)

Symbol	Parameters	Conditions		Ta	Min.	Typ.	Max.	Unit
$I_{DD_{VDDANA}}$(1)	Differential Mode	fs = 1 MSPS / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111'	V_{DDANA}=V_{REF}=1.6V	Max. 85°C Typ.25°C	-	105	128	µA
			V_{DDANA}=V_{REF}=3.6V		-	279	307	
		fs = 1 MSPS / Reference buffer enabled /	V_{DDANA}=1.6V, V_{REF}=1.0V		-	175	231	µA

Symbol	Parameters	Conditions		Ta	Min.	Typ.	Max.	Unit
		BIASREFBUF = '111', BIASREFCOMP = '111'	V_{DDANA}=3.0V, V_{REF}=2.0V		-	300	374	
			V_{DDANA}=3.6V, V_{REF}=3.0V		-	356	438	
		fs = 10 kSPS / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111'	V_{DDANA}=V_{REF}=1.6V		-	30	41	µA
			V_{DDANA}=V_{REF}=3.6V		-	53	71	
		fs = 10 kSPS / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111'	V_{DDANA}=1.6V, V_{REF}=1.0V		-	95	139	µA
			V_{DDANA}=3.0V, V_{REF}=2.0V		-	115	178	
			V_{DDANA}=3.6V, V_{REF}=3.0V		-	122	187	
$I_{DD_{VDDANA}}$ (1)	Single-Ended Mode	fs = 1 MSPS / Reference buffer disabled	V_{DDANA}=V_{REF}=1.6V	Max. 85°C Typ.25°C	-	138	158	µA
			V_{DDANA}=V_{REF}=3.6V		-	321	359	
		fs = 1 MSPS / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111'	V_{DDANA}=1.6V, V_{REF}=1.0V		-	203	257	µA
			V_{DDANA}=3.0V, V_{REF}=2.0V		-	331	413	
			V_{DDANA}=3.6V, V_{REF}=3.0V		-	388	482	
		fs = 10 kSPS / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111'	V_{DDANA}=V_{REF}=1.6V		-	46	62	µA
			V_{DDANA}=V_{REF}=3.6V		-	89	120	
		fs = 10 kSPS / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111'	V_{DDANA}=1.6V, V_{REF}=1.0V		-	109	157	µA
			V_{DDANA}=3.0V, V_{REF}=2.0V		-	138	211	
			V_{DDANA}=3.6V, V_{REF}=3.0V		-	148	228	

Note:

1. These are based on characterization.

Table 45-25. Differential Mode(1)

Symbol	Parameters	Conditions	Min	Typ.	Max	Unit
ENOB	Effective Number of bits (With gain compensation)	V_{DDANA}=3.0V / V_{ref} =2.0V	9.6	10.5	10.6	bits
		V_{DDANA}=1.6V/3.6V, V_{ref}=1.0V	8.9	9.7	9.9	
		V_{DDANA}=V_{ref}=1.6V	10	10.5	11.1	
		V_{DDANA}=V_{ref}=3.6V	10.5	10.9	11.0	
TUE	Total Unadjusted Error	V_{DDANA}=3.0V, V_{ref}=2.0V	-	7.5	11	LSB
INL	Integral Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-1.5	+/-2.1	LSB
DNL	Differential Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-0.8	+1.1/-1.0	LSB
	Gain Error	External Reference voltage 1.0V	-	+/-0.7	+/-1.5	%
		External Reference voltage 3.0V		+/-0.2	0.5	
		Reference bandgap voltage	-	+/-0.4	+/-4.4	
		V_{DDANA}		+/-0.1	0.4	
		V_{DDANA}/2	-	+/-0.4	+/-1.3	
		V_{DDANA}/1.6	-	+/-0.3	+/-0.9	
	Offset Error	External Reference voltage 1.0V	-	+/-1.1	+/-2.4	mV
		External Reference voltage 3.0V		+/-1.1	3	
		Reference bandgap voltage	-	+/-2.3	+/-7.5	
		V_{DDANA}		+/-0.9	2.9	
		V_{DDANA}/2	-	+/-1	+/-2.6	
		V_{DDANA}/1.6	-	+/-1	+/-2.9	
SFDR	Spurious Free Dynamic Range	F_s=1MHz / F_{in}=13 kHz / Full range Input signal V_{DDANA}=3.0V, V_{ref}=2.0V	68	75	77	dB
SINAD	Signal to Noise and Distortion ratio		60	65	66	
SNR	Signal to Noise ratio		61	66	67	
THD	Total Harmonic Distortion		-74	-73	-67	
	Noise RMS	External Reference voltage	-	1.0	2.5	mV

Note:

1. These are based on characterization.

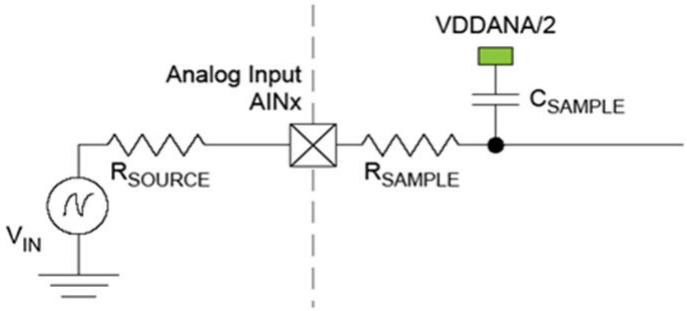
Table 45-26. Single-Ended Mode(1)

Symbol	Parameters	Conditions	Min	Typ.	Max	Unit
ENOB	Effective Number of bits (With gain compensation)	V_{DDANA}=3.0V / V_{ref} =2.0V	8.5	9.5	9.8	bits
		V_{DDANA}=1.6V/3.6V, V_{ref}=1.0V	7.5	8.7	8.9	
		V_{DDANA}=V_{ref}=1.6V	9.0	9.5	9.8	

Symbol	Parameters	Conditions	Min	Typ.	Max	Unit
		V_{DDANA}=V_{ref}=3.6V	9.2	9.8	9.9	
TUE	Total Unadjusted Error	V_{DDANA}=3.0V, V_{ref}=2.0V	-	17.4	31	LSB
INL	Integral Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-2.2	+/-10.1	LSB
DNL	Differential Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-0.8	+/-0.9	LSB
	Gain Error	External Reference voltage 1.0V	-	+/-1	+/-1.3	%
		External Reference voltage 3.0V		+/-0.3	+/-0.6	
		Reference bandgap voltage	-	+/-0.4	+/-3.2	
		V_{DDANA}		+/-0.1	+/-0.3	
		V_{DDANA}/2	-	+/-0.6	+/-1.4	
		V_{DDANA}/1.6	-	+/-0.4	+/-1	
	Offset Error	External Reference voltage 1.0V	-	+/-3.35	+/-13	mV
		External Reference voltage 3.0V		+/-3.6	+/-23.7	
		Reference bandgap voltage	-	+/-1	+/-14.4	
		V_{DDANA}		+/-4.2	+/-24.8	
		V_{DDANA}/2	-	+/-5.7	+/-10.1	
		V_{DDANA}/1.6	-	+/-6.3	+/-13	
SFDR	Spurious Free Dynamic Range	F_s=1MHz / F_{in}=13 kHz / Full range Input signal V_{DDANA}=3.0V, V_{ref}=2.0V	65	71	78	dB
SINAD	Signal to Noise and Distortion ratio		53	59	61	
SNR	Signal to Noise ratio		53	59	61	
THD	Total Harmonic Distortion		-76	-70	64	
	Noise RMS	External Reference voltage	-	2.0	7.0	mV

Note:

1. These are based on characterization.



The minimum sampling time $t_{SAMPLEHOLD}$ for a given R_{SOURCE} can be found using this formula:

$t_{SAMPLEHOLD} \geq (R_{SAMPLE} + R_{SOURCE}) \times (C_{SAMPLE}) \times (n+2) \times \ln(2)$

For a 12 bits accuracy: $t_{SAMPLEHOLD} \geq (R_{SAMPLE} + R_{SOURCE}) \times (C_{SAMPLE}) \times 9.7$

where $t_{SAMPLEHOLD} = \frac{1}{2 \times f_{ADC}}$

45.10.6 Analog Comparator (AC) Characteristics

Table 45-27. Electrical and Timing

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
PNIVR	Positive and Negative input range voltage	-	0	-	V_{DDANA}	V
ICMR	Input common mode range	-	0	-	V_{DDANA}-0.1	V
Off(2)	Offset	COMPCTRLn.SPEED=0x0	-70	-4.5/+1.5	70	mV
		COMPCTRLn.SPEED=0x1	-55	-4.5/+1.5	55	
		COMPCTRLn.SPEED=0x2	-48	-4.5/+1.5	48	
		COMPCTRLn.SPEED=0x3	-42	-4.5/+1.5	42	
V_{Hys}(2)	Hysteresis	COMPCTRLn.HYST=0x0	10	45	74	mV
		COMPCTRLn.HYST=0x1	22	70	106	
		COMPCTRLn.HYST=0x2	37	90	116	
		COMPCTRLn.HYST=0x3	49	105	131	
T_{pd}(2)	Propagation Delay $V_{cm}=V_{ddana}/2$, V_{in} = +-100mV overdrive from V_{cm}	COMPCTRLn.SPEED=0x0	-	4.0	12.3	µs
		COMPCTRLn.SPEED=0x1	-	0.97	2.59	
		COMPCTRLn.SPEED=0x2	-	0.56	1.41	
		COMPCTRLn.SPEED=0x3	-	0.33	0.77	
T_{start}(1)	Start-up time(1)	COMPCTRLn.SPEED=0x0	-	17	56	µs
		COMPCTRLn.SPEED=0x1	-	0.85	4.5	
		COMPCTRLn.SPEED=0x2	-	0.55	3.2	
		COMPCTRLn.SPEED=0x3	-	0.45	2.7	
V_{scale}(2)	INL		-	0.34	-	LSB
	DNL		-	0.06	-	
	Offset Error		-	0.1	-	
	Gain Error		-	1.22	-	

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization

Table 45-28. Power Consumption(1)

Symbol	Parameters	Conditions	Ta.	Min.	Typ	Max.	Unit
I_{DDANA}	Current consumption - $V_{cm}=V_{DDANA}/2$, +-100 mV overdrive from V_{cm}, Voltage scaler disabled	COMPCTRLn.SPEED=0x0, V_{DDANA}=3.3V	Max. 85°C Typ. 25°C	-	50	1973	nA
		COMPCTRLn.SPEED=0x1, V_{DDANA}=3.3V		-	156	2082	
		COMPCTRLn.SPEED=0x2, V_{DDANA}=3.3V		-	289	2223	
		COMPCTRLn.SPEED=0x3, V_{DDANA}=3.3V		-	549	2495	
	Current consumption Voltage scaler only	V_{DDANA}=3.3V		-	13	17	µA

Note:

1. These are based on characterization.

45.10.7 Voltage Reference

Table 45-29. Reference Voltage Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
ADC Ref	ADC internal reference	nom. 1.0V, V_{DDANA}=3.0V, T= 25°C	0.967	1.0	1.017	V
		nom. 1.1V, V_{DDANA}=3.0V, T= 25°C	1.069	1.1	1.120	V
		nom. 1.2V, V_{DDANA}=3.0V, T= 25°C	1.167	1.2	1.227	V
		nom. 1.25V, V_{DDANA}=3.0V, T= 25°C	1.214	1.3	1.280	V
		nom. 2.0V, V_{DDANA}=3.0V, T= 25°C	1.935	2.0	2.032	V
		nom. 2.2V, V_{DDANA}=3.0V, T= 25°C	2.134	2.2	2.242	V
		nom. 2.4V, V_{DDANA}=3.0V, T= 25°C	2.328	2.4	2.458	V
		nom. 2.5V, V_{DDANA}=3.0V, T= 25°C	2.420	2.5	2.565	V
	Ref Temperature coefficient	drift over [-40, +85]°C	-	[-0.01:+0.015]	-	%/°C
		drift over [25, +85]°C	-	[-0.01:+0.005]	-	%/°C
	Ref Supply coefficient	drift over [1.6, 3.6]V	-	[-0.35:+0.35]	-	%/V
AC Ref	AC Ref Accuracy	V_{DDANA}=3.0V, T= 25°C	1.073	1.1	1.123	V
	Ref Temperature coefficient	drift over [-40, +85]°C	-	[-0.01:+0.01]	-	%/°C
		drift over [25, +85]°C	-	[-0.005:+0.001]	-	
	Ref Supply coefficient	drift over[1.6, 3.6]V	-	[-0.35:+0.35]	-	%/V

Note:

1. These values are based on characterization.

45.11 NVM Characteristics

Table 45-30. NVM Max Speed Characteristics

	Conditions	CPU F_{max} (MHz)		
		0WS	1WS	2WS
PL0 (-40/85°C)	V_{DD}>1.6 V	6	8	8
	V_{DD}>2.7 V	7.5	8	8
PL2 (-40/85°C)	V_{DD}>1.6 V	14	28	32
	V_{DD}>2.7 V	24	32	32

Table 45-31. NVM Timing Characteristics(1)

Symbol	Parameter	Max	Units
t_{FPP}	Page programming time	2.5	ms
t_{FRE}	Row erase time	6	

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

For this Flash technology, a maximum number of 8 consecutive writes is allowed per row. Once this number is reached, a row erase is mandatory.

Table 45-32. Flash Endurance and Data Retention

Symbol	Parameter	Conditions	Min.	Typ.	Units
Ret_{NVM25k}	Retention after up to 25k	Average ambient 55°C	10	50	Years
$Ret_{NVM2.5k}$	Retention after up to 2.5k	Average ambient 55°C	20	100	Years
Ret_{NVM100}	Retention after up to 100	Average ambient 55°C	25	>100	Years
Cyc_{NVM}	Cycling Endurance(1)	-40°C < Ta < 85°C	25K	100K	Cycles

Note:

1. An endurance cycle is a write and an erase operation.

Table 45-33. EEPROM Emulation(1) Reliability Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Units
Ret_{EE100k}	Retention after up to 100k	Average ambient 55°C	10	50	Years
Ret_{EE10k}	Retention after up to 10k	Average ambient 55°C	20	100	Years
Cyc_{EE}	Cycling Endurance(2)	-40°C < Ta < 85°C	100K	400K	Cycles

Note:

1. The EEPROM emulation is a software emulation described in the App note AT03265.
2. An endurance cycle is a write and an erase operation.

Table 45-34. Flash Erase and Programming Current

Symbol	Parameter	Typ.	Units
IDD_{NVM}	Maximum current (peak) during whole programming or erase operation	10	mA

45.12 Oscillators Characteristics

45.12.1 XOSC

45.12.1.1 Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on Xin.

Table 45-35. Digital Clock Characteristics

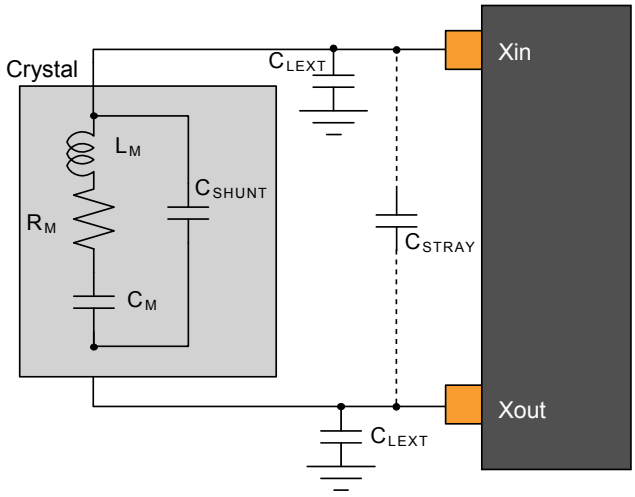
Symbol	Parameter	Min.	Typ.	Max	Units
F_{Xin}	Xin clock frequency	-	-	24	MHz
DC_{Xin}(1)	Xin clock duty cycle	40	50	60	%

Note: These values are based on simulation. They are not covered by production test limits or characterization.

45.12.1.2 Crystal Oscillator Characteristics

The following table describes the characteristics for the oscillator when a crystal is connected between XIN and XOUT .

Figure 45-2. Oscillator Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$$C_{LEXT} = 2\left(C_L - C_{STRAY} - C_{SHUNT}\right)$$

where C_{STRAY} is the capacitance of the pins and PCB, C_{SHUNT} is the shunt capacitance of the crystal.

Table 45-36. Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
f_{OUT}	Crystal oscillator frequency		0.4	-	32	MHz
ESR	Crystal Equivalent Series Resistance Safety Factor=3	f=0.455MHz, C_L=100pF XOSC.GAIN=0	-	-	5.6K	Ω
		f=2MHz, C_L=20pF XOSC.GAIN=0	-	-	416	
		f=4MHz, C_L=20pF XOSC.GAIN=1	-	-	243	
		f=8MHz, C_L=20pF XOSC.GAIN=2	-	-	138	
		f=16MHz, C_L=20pF XOSC.GAIN=3	-	-	66	
		f=32MHz, C_L=20pF XOSC.GAIN=4	-	-	56	
C_{XIN}	Parasitic capacitor load		-	5.8	-	pF
C_{XOUT}	Parasitic capacitor load		-	3.2	-	pF
$t_{STARTUP}$	Start-up time	F=2MHz - CL=20pF XOSC,GAIN=0	-	14k	48k	cycles
		F=4MHz - CL=20pF XOSC,GAIN=1	-	6.8k	19.5k	
		F=8MHz - CL=20pF XOSC,GAIN=2	-	5.6k	13k	
		F=16MHz - CL=20 pF XOSC,GAIN=3	-	6.8k	14.5k	
		F=32MHz - CL=20pF XOSC,GAIN=4	-	5.3K	9.6k	

Note: 1. These values are based on characterization.

Table 45-37. Power consumption

Symbol	Parameter	Conditions		Ta.	Typ.	Max.	Units
I_{DD}	Current Consumption	F=2MHz - CL=20pF XOSC, GAIN=0, V_{DD} =3.3V	AGC=OFF	Max. 85°C Typ. 25°C	89	133	µA
			AGC=ON		82	130	
		F=4MHz - CL=20pF XOSC, GAIN=1, V_{DD} =3.3V	AGC=OFF		140	194	
			AGC=ON		102	156	
		F=8MHz - CL=20pF XOSC, GAIN=2, V_{DD} =3.3V	AGC=OFF		243	313	
			AGC=ON		166	232	
		F=16MHz - CL=20pF XOSC, GAIN=3, V_{DD} =3.3V	AGC=OFF		493	605	
			AGC=ON		293	393	
		F=32MHz - CL=20pF XOSC, GAIN=4, V_{DD} =3.3V	AGC=OFF		1467	1777	
			AGC=ON		651	1045	

Note: 1. These values are based on characterization.

45.12.2 XOSC32K

45.12.2.1 Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on Xin32 pin.

Table 45-38. Digital Clock Characteristics(1)

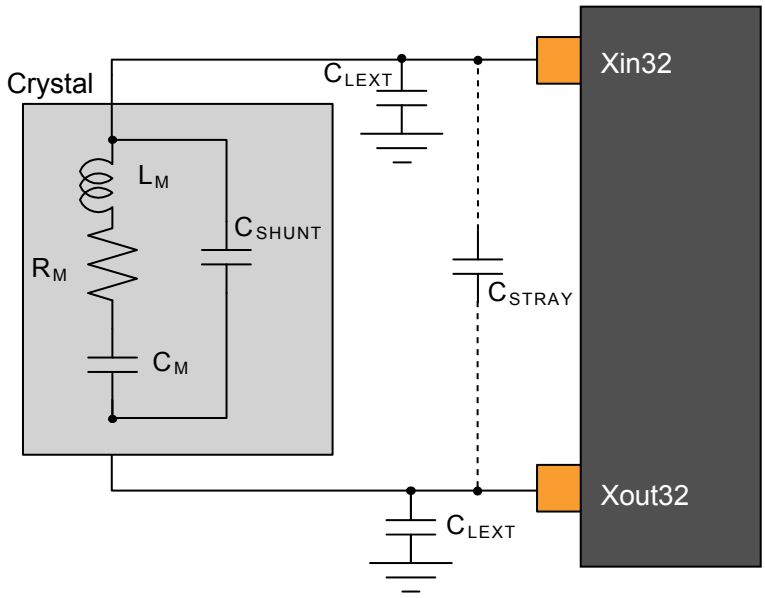
Symbol	Parameter	Min.	Typ.	Max.	Units
$f_{CPXIN32}$	Xin32 clock frequency	-	32.768	1000	kHz
DC_{xin}	Xin32 clock duty cycle	40	50	60	%

Note: These values are based on simulation. They are not covered by production test limits or characterization.

45.12.2.2 External 32KHz Crystal Oscillator (XOSC32K) Characteristics

The following table describes the characteristics for the oscillator when a crystal is connected between XIN32 and XOUT32.

Figure 45-3. Oscillator Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$$C_{LEXT} = 2\left(C_L - C_{STRAY} - C_{SHUNT}\right)$$

where C_{STRAY} is the capacitance of the pins and PCB, C_{SHUNT} is the shunt capacitance of the crystal.

Table 45-39. 32KHz Crystal Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Crystal oscillator frequency	-	-	32.768	-	kHz
CL	Crystal load capacitance	-	7	9	12.5	pF

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
C_{SHUNT}	Crystal shunt capacitance	-	0.6	-	2	pF
C_M	Motional capacitance	-	0.6	-	3	fF
ESR(2)	Crystal Equivalent Series Resistance - SF = 3	f=32.768kHz, C_L=12.5pF	-	50	70	kΩ
C_{XIN32k}	Parasitic load capacitor	-	-	2.31	-	pF
$C_{XOUT32k}$		-	-	2.53	-	
$t_{STARTUP}$	Startup time	f=32.768kHz, C_L=12.5pF	-	25k	82k	cycles
P_{on}	Drive Level (1)	-	-	-	0.1	µW

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization

Table 45-40. Power Consumption

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	V_{DD}=3.3V	Max. 85°C Typ. 25°C	-	311	723	nA

Note: These are based on characterization

45.12.3 OSCULP32K

Table 45-41. Ultra Low Power Internal 32KHz RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max	Units
F_{OUT}	Output frequency	at 25 °C, at V_{DDIO} = 3.3V	31.557	32.768	34.313	kHz
		at 25 °C, over [1.62, 3.63]V	31.670	32.768	34.420	kHz
		over[-40,+85]°C, over [1.62, 3.63]V	25.806	32.768	39.014	kHz
Duty	Duty Cycle		-	50	-	%

45.12.4 16MHz RC Oscillator (OSC16M) Characteristics

Table 45-42. Multi RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	V_{DD}=3.3V, T=25°C	3.953	4	4.062	MHz
			7.877	8	8.112	
			11.857	12	12.139	
			15.754	16	16.235	
TempDrift	Freq vs. temperature drift	-	-4	-	4	%

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
SupplyDrift	Freq vs. supply drift	-	-2	-	2	
T_{WUP}(2)	Wake up time - 1st clock edge after enable	F_{OUT} = 4MHz	-	0.12	0.27	µs
		F_{OUT} = 8MHz	-	0.12	0.25	
		F_{OUT} = 12MHz	-	0.12	0.27	
		F_{OUT} = 16MHz	-	0.12	0.25	
$T_{STARTUP}$(2)	Startup time	F_{OUT} = 4MHz	-	1.2	2.9	µs
		F_{OUT} = 8MHz	-	1.3	2.6	
		F_{OUT} = 12MHz	-	1.3	2.8	
		F_{OUT} = 16MHz	-	1.4	3.1	
Duty(1)	Duty Cycle	-	45	50	55	%

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization

Table 45-43. Power Consumption

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	F_{OUT} = 4MHz, V_{DD} =3.3V	Max. 85°C Typ. 25°C	-	64	96	µA
		F_{OUT} = 8MHz, V_{DD} =3.3V		-	91	122	
		F_{OUT} = 12MHz, V_{DD} =3.3V		-	114	144	
		F_{OUT} = 16MHz, V_{DD} =3.3V		-	141	169	

Note: These are based on characterization

45.12.5 Digital Frequency Locked Loop (DFLL48M) Characteristics

Table 45-44. DFLL48M Characteristics - Open Loop Mode(1)(2)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
f_{OUT}	Output frequency	IDFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512 LDO Regulator mode	46.6	47.8	49	MHz
$t_{STARTUP}$	Start-up time	DFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512 f_{OUT} within 90 % of final value	-	8.3	9.1	µs

Note:

1. DFLL48 in open loop can be used only with LDO regulator
2. These are based on characterization

Table 45-45. DFLL48M Characteristics - Closed Loop Mode

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$f_{CloseOUT}$	Average Output frequency	f_{REF} = XTAL, 32.768kHz, 100ppm DFLLMUL=1464	47.963	47.972	47.981	MHz
f_{REF}(2)(3)	Reference frequency		732	32768	33000	Hz
$f_{CloseJitter}$(1)	Cycle to Cycle jitter	f_{REF} = XTAL, 32.768kHz, 100ppm DFLLMUL=1464	-	-	0.51	ns
t_{LOCK}(1)	Lock time	f_{REF} = XTAL, 32.768kHz, 100ppm DFLLMUL=1464 DFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512 DFLLCTRL.BPLCKC = 1 DFLLCTRL.QLDIS = 0 DFLLCTRL.DDDIS = 1 DFLLMUL.FSTEP = 10	-	200	700	µs

Note:

1. These are based on characterization
2. To ensure that the device stays within the maximum allowed clock frequency, any reference clock for DFLL in close loop must be within a 2% error accuracy
3. These values are based on simulation. They are not covered by production test limits or characterization.

Table 45-46. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Power consumption,Open loop	DFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512, V_{DD}=3.3V	Max. 85°C Typ. 25°C	-	286	-	µA
	Power consumption, Close loop	f_{REF} = 32.768kHz, V_{DD}=3.3V		-	362	-	

Note:

1. These values are based on characterization.

45.12.6 Digital Phase Lock Loop Characteristics

Table 45-47. Fractional Digital Phase Lock Loop (FDPLL96M) Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{IN}	Input Clock Frequency	-	32	-	2000	kHz
F_{OUT}	Output frequency	PL0	48	-	48	MHz
		PL2	48	-	96	MHz

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
J_P(2)	Period Jitter	PL0, F_{IN}=32kHz @ F_{OUT}=48MHz	-	1.9	5.0	%
		PL2, F_{IN}=32kHz @ F_{OUT}=48MHz	-	1.9	4.0	
		PL2, F_{IN}=32kHz @ F_{OUT}=96MHz	-	3.3	7.0	
		PL0, F_{IN}=2MHz @ F_{OUT}=48MHz	-	2.0	8.0	
		PL2, F_{IN}=2MHz @ F_{OUT}=48MHz	-	2.0	4.0	
		PL2, F_{IN}=2MHz @ F_{OUT}=96MHz	-	4.2	7.0	
T_{LOCK}(1)	Lock Time	After startup, time to get lock signal, F_{IN} = 32kHz @ F_{OUT} = 96MHz	-	1	2	ms
		After startup, time to get lock signal, F_{IN} = 2MHz @ F_{OUT} = 96MHz	-	25	35	µs
Duty	Duty Cycle(1)	-	40	50	60	%

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These are based on characterization.

Table 45-48. Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	Ck=48MHz (PL0), V_{DD} =3.3V	Max. 85°C Typ. 25°C	-	454	548	µA
		Ck=96MHz (PL2), V_{DD} =3.3V		-	934	1052	

Note:

1. These are based on characterization.

45.13 USB Characteristics

The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

The USB interface is USB-IF certified :

- TID 40001708 - Peripheral Silicon > Low/Full Speed > Silicon Building Blocks

Electrical configuration required to be USB-compliant:

- the performance level must be PL2 only
- the CPU fequency must be higher 8MHz when USB is active (No contrain for USB suspend mode)
- the operating voltages must be 3.3V (Min. 3.0V, Max. 3.6V).
- the GCLK_USB frequency accuracy source must be less than:
  - in USB device mode, 48MHz +/-0.25%

Table 45-49. GCLK_USB Clock Setup Recommendations

Clock setup		USB Device
DFLL48M	Open loop	No
	Close loop, Ref. internal OSC source	No
	Close loop, Ref. external XOSC source	Yes
	Close loop, Ref. SOF (USB recovery mode)(1)	Yes(2)
FDPLL	internal OSC (32K, 8M…)	No
	external OSC (<1MHz)	Yes
	external OSC (>1MHz)	Yes(3)

Note:

1. When using DFLL48M in USB recovery mode, the Fine Step value must be Ah to guarantee a USB clock at +/-0.25% before 11ms after a resume. Only usable in LDO regulator mode.
2. Very high signal quality and crystal less. It is the best setup for USB Device mode.
3. FDPLL lock time is short when the clock frequency source is high (> 1 MHz). Thus, FDPLL and external OSC can be stopped during USB suspend mode to reduce consumption and guarantee a USB wakeup time (See TDRSMDN in USB specification).

45.14 SLCD Characteristics

The values in the table below are measured values of power consumption under the following conditions, except where noted:

- T=25°C. Standby Mode. Low Power waveform Mode. Frame Rate = 32Hz from ULP32K Oscillator.
- No ACM, ABM or Segment Animation Features activated
- All Segments on, No Glass Load = 0pF and With Glass Load = 22pF on each COM and SEG line.
- Contrast Adjustment CTST control set to 0x7 (~2.97V) when using internal VLCD generation
- No Bias Buffer. No External Bias Output. No Low Resistance Network Enable
- Reference Refresh Frequency = 500Hz. Power Refresh Frequency = 1kHz
- I_{LCD} current based on $I_{LCD} = I_{stdby(LCD\ On)} - I_{stdby\ (LCD\ Off)}$. with $I_{stdby} = I_{VDD} + I_{VDDANA}$
- To minimize power consumption of the SLCD module use $V_{LCD} = V_{DD}$ or $V_{LCD} > V_{DD} + 0.4V$

Table 45-50. Liquid Crystal Display Controller Characteristics

Symbol	Parameter	Conditions		Ta	Min.	Typ.	Max.	Units
V_{DD}	Power Supply for SLCD operation				1.62	3.3	3.63	V
V_{LCD}	VLCD maximum value	V_{DD}=1.62V, Contrast Adjustment: CTST = 0xF	I_{load} = 10µA	Max 85°C	2.94	3.05	3.15	V
			I_{load} = 20µA	Typ 25°C	2.78	2.97	3.13	
		V_{DD}=1.8V, Contrast Adjustment: CTST = 0xF	I_{load} = 10µA		3.32	3.44	3.55	

Symbol	Parameter	Conditions			Ta	Min.	Typ.	Max.	Units
				I_{load} = 20µA.		3.17	3.36	3.53	
		V_{DD}>2V, Contrast Adjustment: CTST = 0xF		I_{load} = 10µA		3.32	3.48	3.71	
				I_{load} = 20µA.		3.28	3.45	3.69	
$C_{V_{LCD}}$	External V_{LCD} capacitor					-	1	-	µF
F_{frame}	LCD Frame Frequency					-	32	120	Hz
T_{start}	V_{LCD} Start-up Time	V_{DD}=2.4V, CTST=0x7				-	25	-	ms
V_{LCD}	LCD Regulated Voltages	Contrast Adjustment: CTST = 0x7				-	3	-	V
BIAS3/4						-	3*V_{LCD}/4	-	
BIAS1/2						-	V_{LCD}/2	-	
BIAS1/4						-	V_{LCD}/4	-	
BIAS2/3						-	2*V_{LCD}/3	-	
BIAS1/3						-	V_{LCD}/3	-	
I_{LCD}	LCD current supply With no Glass (Load = 0pF)	Internal V_{LCD} generation	Nb COM = 8 Nb SEG = 40 (1/4 Bias)	V_{DD} = 3.6V	Max 85°C Typ 25°C		1.68	5.97	µA
				V_{DD} = 2.4V			2.41	15.34	
			Nb COM = 4 Nb SEG = 44 (1/3 Bias)	V_{DD} = 3.6V			1.63	6.61	
				V_{DD} = 2.4V			2.28	16.86	
			Nb COM = 1 Nb SEG = 44 (Static Bias)	V_{DD} = 3.6V			1.20	5.97	
				V_{DD} = 2.4V			1.25	15.24	
		External V_{LCD} generation (V_{LCD} = 3.0V)	Nb COM = 8 Nb SEG = 40 (1/4 Bias)	V_{DD} = 3.6V			1.03	5.76	
				V_{DD} = 2.4V			1.25	7.30	
			Nb COM = 4 Nb SEG = 44 (1/3 Bias)	V_{DD} = 3.6V			1.06	5.66	
				V_{DD} = 2.4V			1.26	7.39	

Symbol	Parameter	Conditions			Ta	Min.	Typ.	Max.	Units
			Nb COM = 1 Nb SEG = 44 (Static Bias)	V_{DD} = 3.6V			1.07	5.81	
				V_{DD} = 2.4V			1.22	7.04	
I_{LCD}	LCD current supply With Glass (Load = 22pF)	Internal V_{LCD} generation	Nb COM = 8 Nb SEG = 40 (1/4 Bias)	V_{DD} = 3.6V	Max 85°C Typ 25°C		3.02	8.06	µA
				V_{DD} = 2.4V			4.84	18.44	
			Nb COM = 4 Nb SEG = 44 (1/3 Bias)	V_{DD} = 3.6V			2.44	8.22	
				V_{DD} = 2.4V			3.84	17.64	
			Nb COM = 1 Nb SEG = 44 (Static Bias)	V_{DD} = 3.6V			1.78	6.86	
				V_{DD} = 2.4V			1.99	15.23	
		External V_{LCD} generation (V_{LCD} = 3.0V)	Nb COM = 8 Nb SEG = 40 (1/4 Bias)	V_{DD} = 3.6V	Max 85°C Typ 25°C		1.09	5.96	
				V_{DD} = 2.4V			1.25	6.62	
			Nb COM = 4 Nb SEG = 44 (1/3 Bias)	V_{DD} = 3.6V			1.06	6.34	
				V_{DD} = 2.4V			1.27	8.41	
			Nb COM = 1 Nb SEG = 44 (Static Bias)	V_{DD} = 3.6V			1.23	6.38	
				V_{DD} = 2.4V			1.22	8.00	

Table 45-51. Liquid Crystal Display Contrast Values (VLCD)

Symbol	Parameter	Conditions	CTRLC.CTST value	Typ.	Units
V_{LCD}	Contrast V_{LCD} Values	I_{load} = 20µA V_{DD}>2V V_{LCD} generated internally	0x0	2.45	V
			0x1	2.51	
			0x2	2.57	
			0x3	2.64	
			0x4	2.71	
			0x5	2.78	

Symbol	Parameter	Conditions	CTRLC.CTST value	Typ.	Units
			0x6	2.84	
			0x7	2.91	
			0x8	2.98	
			0x9	3.04	
			0xA	3.11	
			0xB	3.18	
			0xC	3.25	
			0xD	3.31	
			0xE	3.38	
			0xF	3.45	

45.15 External Reset Pin

Table 45-52. External Reset Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
t_{EXT}	Minimum reset pulse width duration		1	-	-	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

46. Typical Characteristics

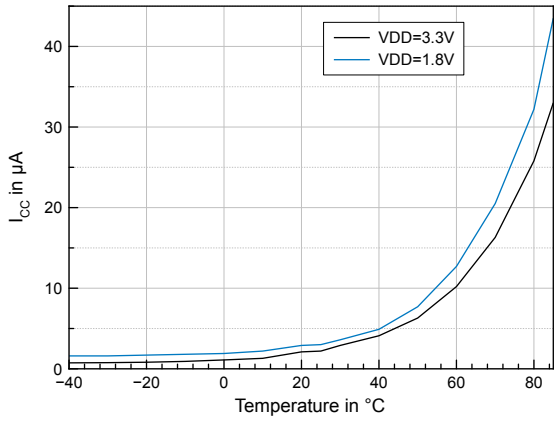
46.1 Power Consumption over Temperature in Sleep Modes

Power Consumption in Standby Sleep Mode with RTC

Operating conditions:

- VDDIN = 3.3V or 1.8V
- ULPVERG LPEFF Enable
- RTC running on external 32KHz crystal
- BOD33 is disabled

Figure 46-1. Power Consumption over Temperature in STANDBY Sleep Mode with RTC

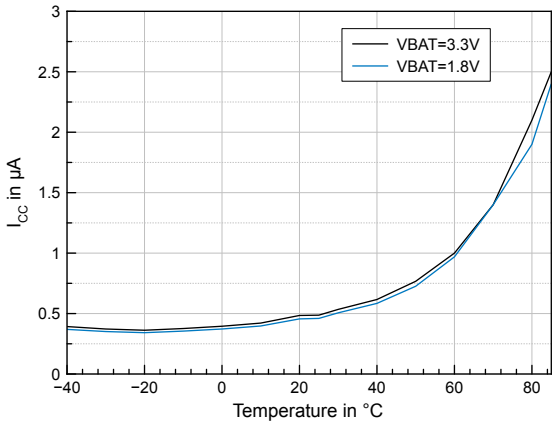


Power Consumption in BACKUP Sleep Mode with RTC

Operating conditions:

- VDDIN =0V
- VBAT = 3.3V or 1.8V
- RTC running on external 32KHz crystal
- BOD33 is disabled

Figure 46-2. Power Consumption over Temperature in Standby Sleep Mode with RTC



47. Packaging Information

47.1 Thermal Considerations

47.1.1 Thermal Resistance Data

The following table summarizes the thermal resistance data depending on the package.

Table 47-1. Thermal Resistance Data

Package Type	θ_{JA}	θ_{JC}
48-pin TQFP	64.2°C/W	12.3°C/W
64-pin TQFP	60.8°C/W	12.0°C/W
100-pin TQFP	58.5°C/W	12.7°C/W
48-pin QFN	32.4°C/W	11.2°C/W
64-pin QFN	32.7°C/W	10.8°C/W
49-pin WLCSP	37.3°C/W	5.8°C/W
100-pin UFBGA	54.7°C/W	9.0°C/W

Related Links

Junction Temperature

47.1.2 Junction Temperature

The average chip-junction temperature, T_J, in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

where:

- θ_{JA} = Package thermal resistance, Junction-to-ambient (°C/W), see Thermal Resistance Data
- θ_{JC} = Package thermal resistance, Junction-to-case thermal resistance (°C/W), see Thermal Resistance Data
- $\theta_{HEATSINK}$ = Thermal resistance (°C/W) specification of the external cooling device
- P_D = Device power consumption (W)
- T_A = Ambient temperature (°C)

From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device is to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

Related Links

General Operating Ratings

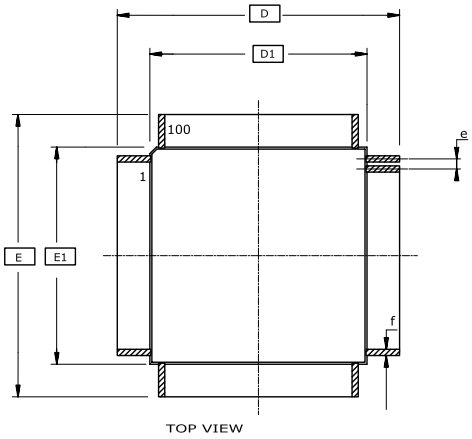
Power Consumption

Thermal Resistance Data

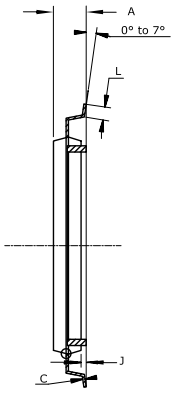
47.2 Package Drawings

47.2.1 100 pin TQFP

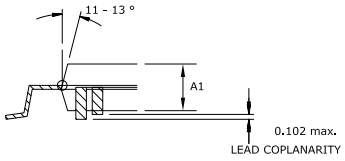
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



DETAIL VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.20	
A1	0.95	-----	1.05	
C	0.09	-----	0.20	
D/E		16.00		
D1/E1		14.00		2
J	0.05	-----	0.15	
L	0.45	-----	0.75	
e		0.50 BSC		
f	0.17	-----	0.27	
n		100		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation AED.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side.
Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

Table 47-2. Device and Package Maximum Weight

520	mg

Table 47-3. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 47-4. Package Reference

JEDEC Drawing Reference	MS-026, variant AED
JESD97 Classification	e3

47.2.2 100-Pin UFBGA

Figure 47-1. 100-Pin UFBGA Outline Drawing

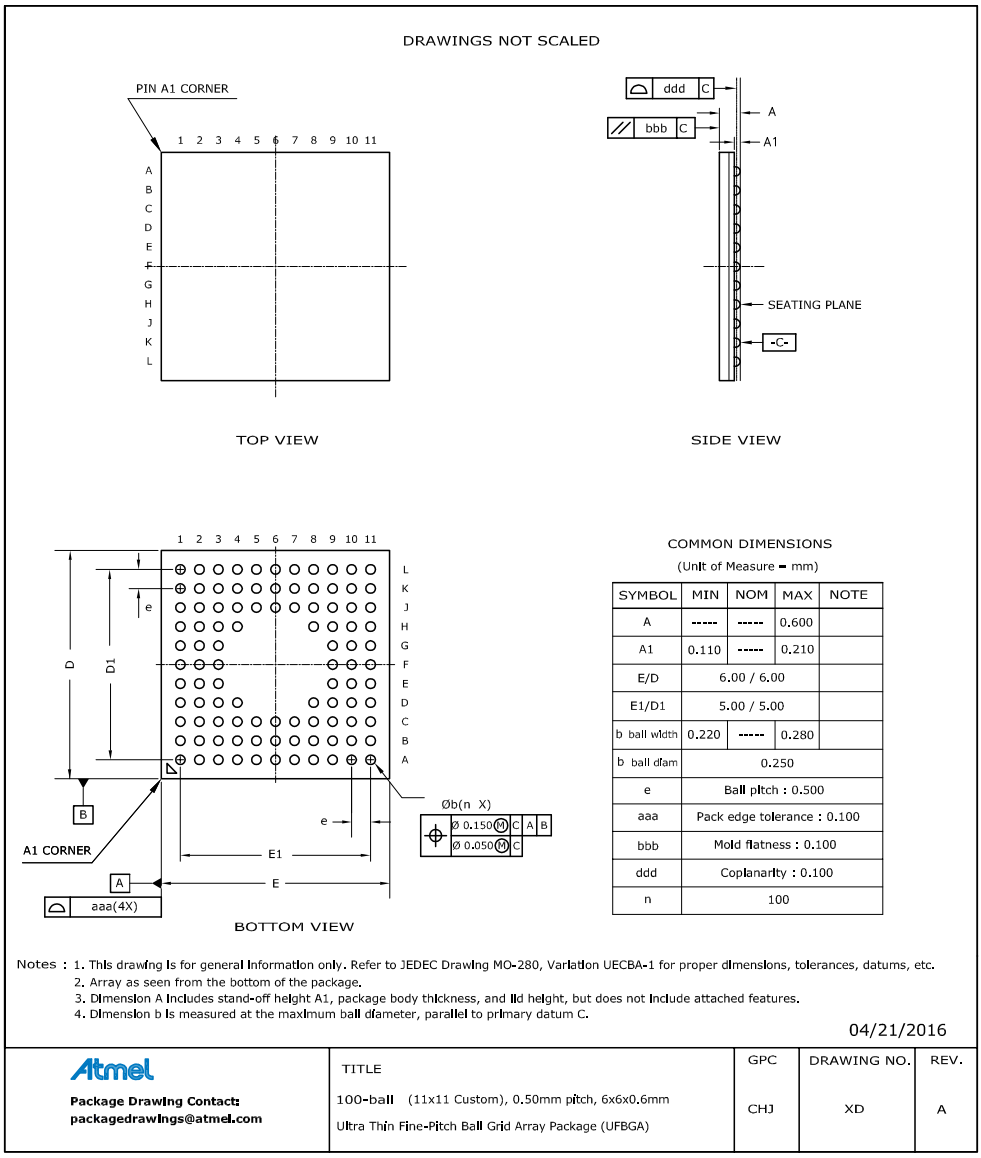


Table 47-5. Device and Package Maximum Weight

42	mg

Table 47-6. Package Characteristics

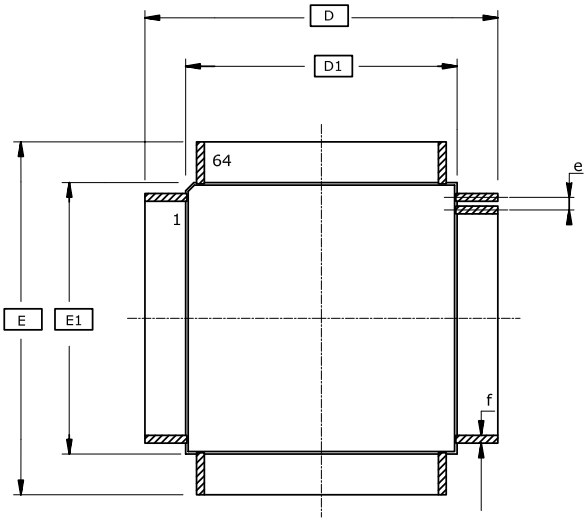
Moisture Sensitivity Level	MSL3

Table 47-7. Package Reference

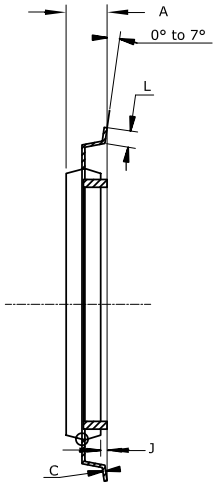
JEDEC Drawing Reference	N/A (custom)
JESD97 Classification	e8

47.2.3 64 pin TQFP

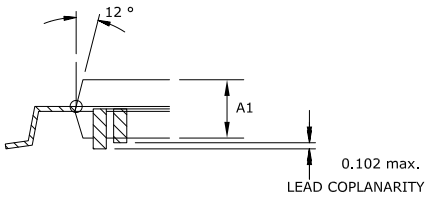
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



DETAIL VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.20	
A1	0.95	-----	1.05	
C	0.09	-----	0.20	
D/E		12.00		
D1/E1		10.00		2
J	0.05	-----	0.15	
L	0.45	-----	0.75	
e		0.50 BSC		
f	0.17	-----	0.27	
n		64		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ACD.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

05/03/2013

Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	TITLE MF, 64 Lds - 0.50mm Pitch, 10x10x1.0mm Body size Thin Profile Plastic Quad Flat Package (TQFP)	GPC AIN	DRAWING NO. MF	REV. D

Table 47-8. Device and Package Maximum Weight

300	mg

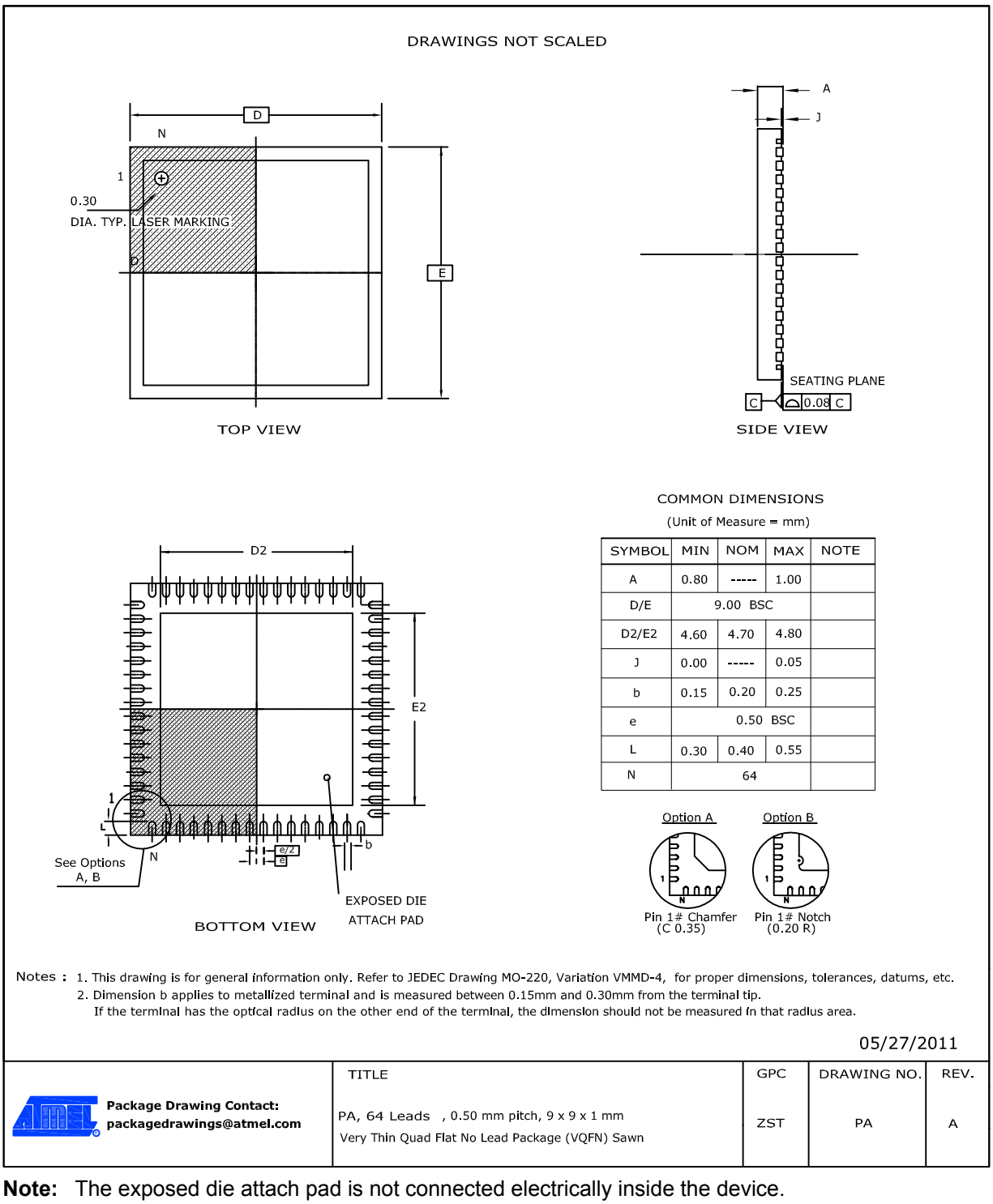
Table 47-9. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 47-10. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

47.2.4 64 pin QFN



Note: The exposed die attach pad is not connected electrically inside the device.

Table 47-11. Device and Package Maximum Weight

200	mg

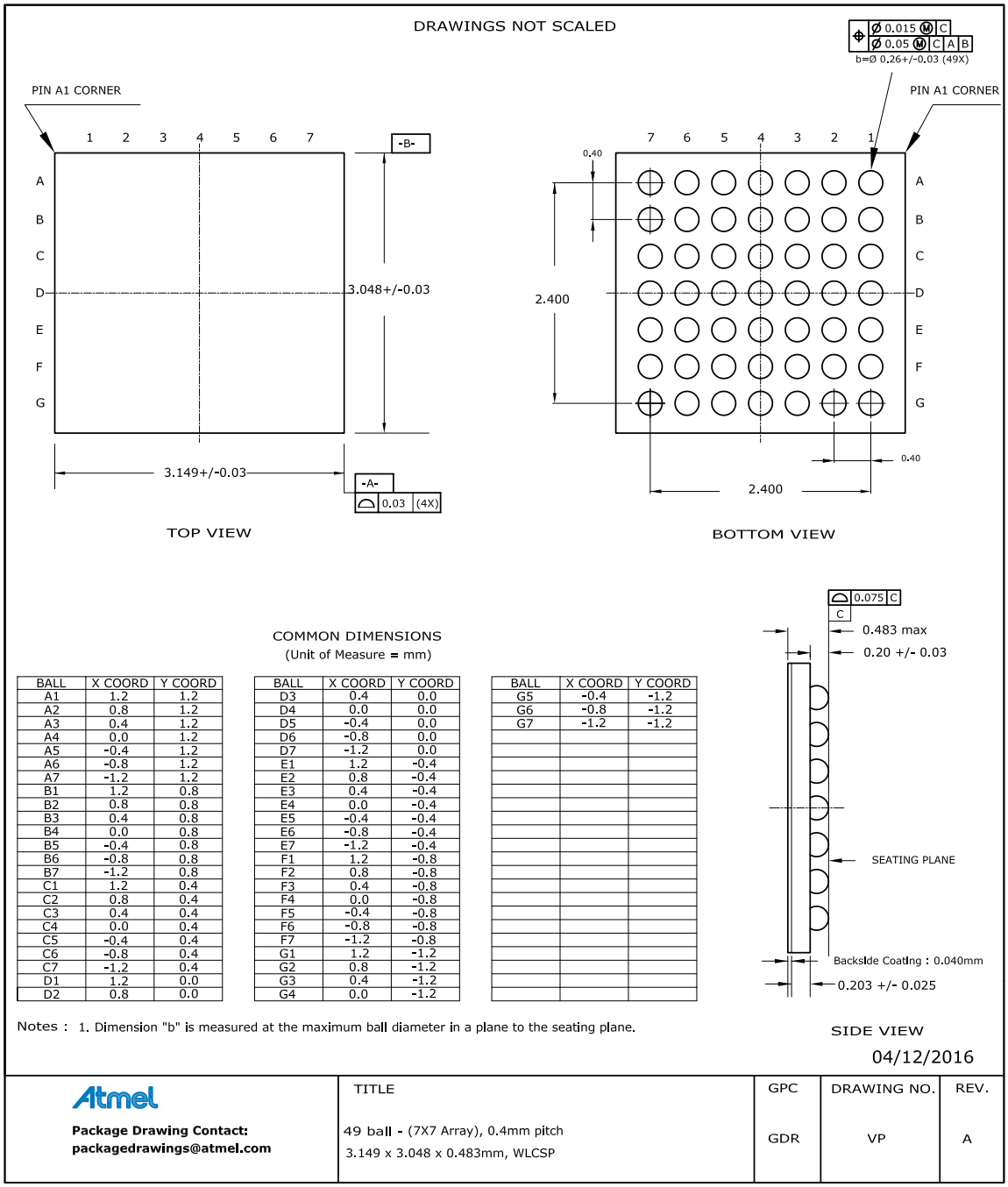
Table 47-12. Package Charateristics

Moisture Sensitivity Level	MSL3

Table 47-13. Package Reference

JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

47.2.5 49-Ball WLCSP



COMMON DIMENSIONS
(Unit of Measure = mm)

BALL	X COORD	Y COORD
A1	1.2	1.2
A2	0.8	1.2
A3	0.4	1.2
A4	0.0	1.2
A5	-0.4	1.2
A6	-0.8	1.2
A7	-1.2	1.2
B1	1.2	0.8
B2	0.8	0.8
B3	0.4	0.8
B4	0.0	0.8
B5	-0.4	0.8
B6	-0.8	0.8
B7	-1.2	0.8
C1	1.2	0.4
C2	0.8	0.4
C3	0.4	0.4
C4	0.0	0.4
C5	-0.4	0.4
C6	-0.8	0.4
C7	-1.2	0.4
D1	1.2	0.0
D2	0.8	0.0
D3	0.4	0.0
D4	0.0	0.0
D5	-0.4	0.0
D6	-0.8	0.0
D7	-1.2	0.0
E1	1.2	-0.4
E2	0.8	-0.4
E3	0.4	-0.4
E4	0.0	-0.4
E5	-0.4	-0.4
E6	-0.8	-0.4
E7	-1.2	-0.4
F1	1.2	-0.8
F2	0.8	-0.8
F3	0.4	-0.8
F4	0.0	-0.8
F5	-0.4	-0.8
F6	-0.8	-0.8
F7	-1.2	-0.8
G1	1.2	-1.2
G2	0.8	-1.2
G3	0.4	-1.2
G4	0.0	-1.2
G5	-0.4	-1.2
G6	-0.8	-1.2
G7	-1.2	-1.2

Notes : 1. Dimension "b" is measured at the maximum ball diameter in a plane to the seating plane.

Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	TITLE 49 ball - (7X7 Array), 0.4mm pitch 3.149 x 3.048 x 0.483mm, WLCSP	GPC GDR	DRAWING NO. VP	REV. A

Table 47-14. Device and Package Maximum Weight

8.45	mg

Table 47-15. Package Characteristics

Moisture Sensitivity Level	MSL1

Table 47-16. Package Reference

JEDEC Drawing Reference	N/A
JESD97 Classification	E1

47.2.6 48 pin TQFP

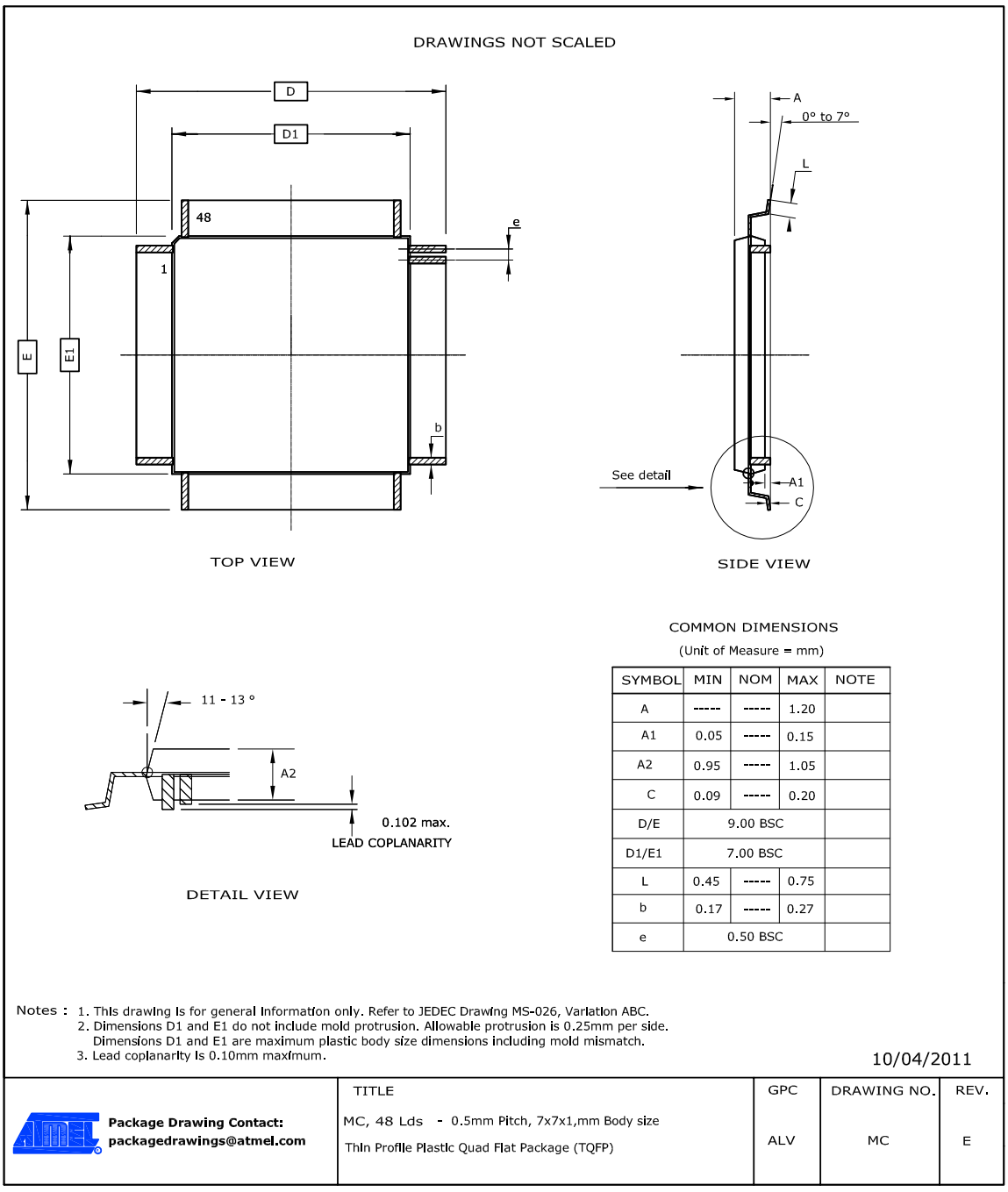


Table 47-17. Device and Package Maximum Weight

140	mg

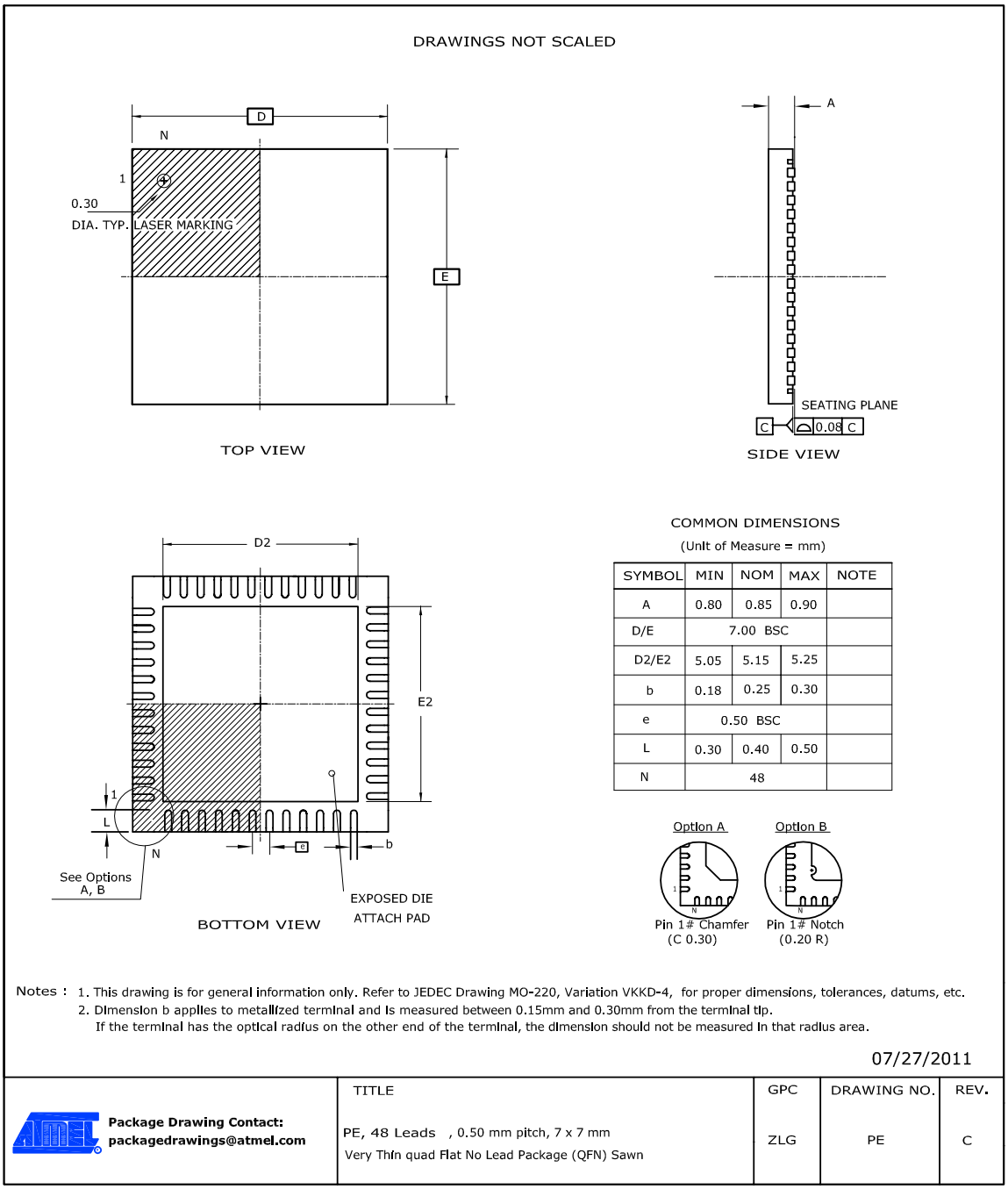
Table 47-18. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 47-19. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

47.2.7 48 pin QFN



Note: The exposed die attach pad is not connected electrically inside the device.

Table 47-20. Device and Package Maximum Weight

140	mg

Table 47-21. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 47-22. Package Reference

JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

47.3 Soldering Profile

The following table gives the recommended soldering profile from J-STD-20.

Table 47-23.

Profile Feature	Green Package
Average Ramp-up Rate (217°C to peak)	3°C/s max.
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150s
Time within 5°C of Actual Peak Temperature	30s
Peak Temperature Range	260°C
Ramp-down Rate	6°C/s max.
Time 25°C to Peak Temperature	8 minutes max.

A maximum of three reflow passes is allowed per component.

48. Schematic Checklist

48.1 Introduction

This chapter describes a common checklist which should be used when starting and reviewing the schematics for a SAM L22 design. This chapter illustrates recommended power supply connections, how to connect external analog references, programmer, debugger, oscillator and crystal.

48.1.1 Operation in Noisy Environment

If the device is operating in an environment with much electromagnetic noise it must be protected from this noise to ensure reliable operation. In addition to following best practice EMC design guidelines, the recommendations listed in the schematic checklist sections must be followed. In particular placing decoupling capacitors very close to the power pins, a RC-filter on the $\overline{\text{RESET}}$ pin, and a pull-up resistor on the SWCLK pin is critical for reliable operations. It is also relevant to eliminate or attenuate noise in order to avoid that it reaches supply pins, I/O pins and crystals.

48.2 Power Supply

The SAM L22 supports a single or dual power supply from 1.62V to 3.63V. The same voltage must be applied to both VDDIN and VDDANA.

The internal voltage regulator has four different modes:

- Linear mode: this mode does not require any external inductor. This is the default mode when CPU and peripherals are running
- Switching mode (Buck): the most efficient mode when the CPU and peripherals are running.
- Low Power (LP) mode: This is the default mode used when the chip is in standby mode
- Shutdown mode: When the chip is in backup mode, the internal regulator is turned off

Selecting between switching mode and linear mode can be done by software on the fly, but the power supply must be designed according to which mode is to be used.

48.2.1 Power Supply Connections

The following figures show the recommended power supply connections for switched/linear mode, linear mode only and with battery backup.

Figure 48-1. Power Supply Connection for Switching/Linear Mode

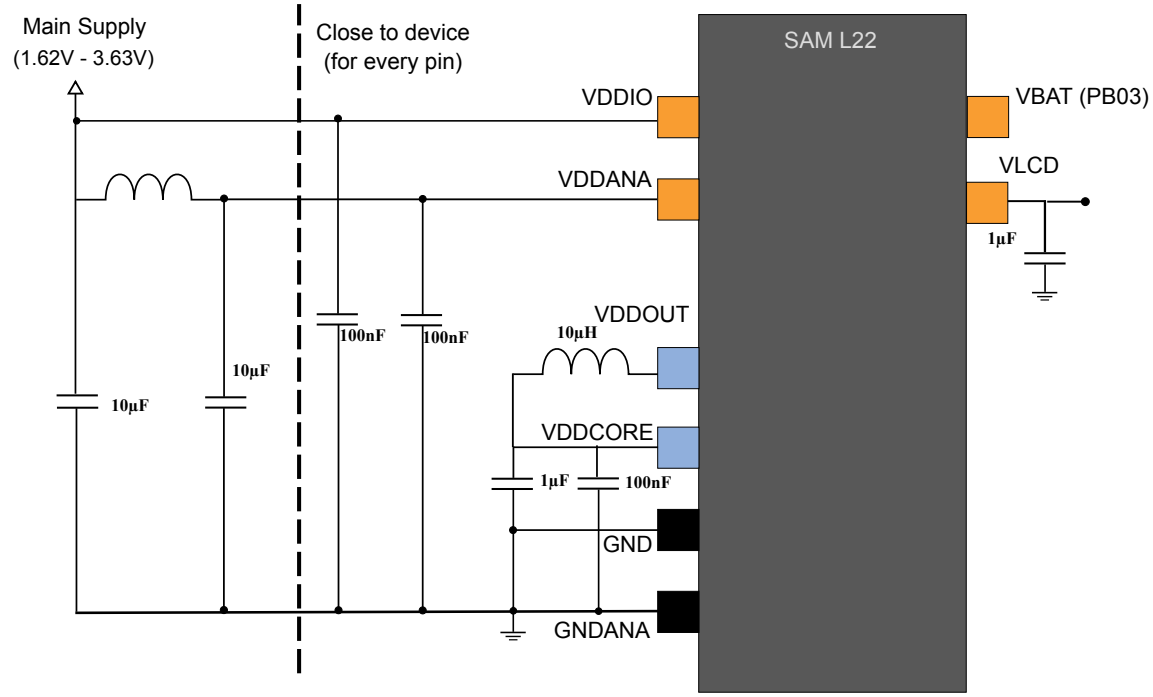


Figure 48-2. Power Supply Connection for Linear Mode Only

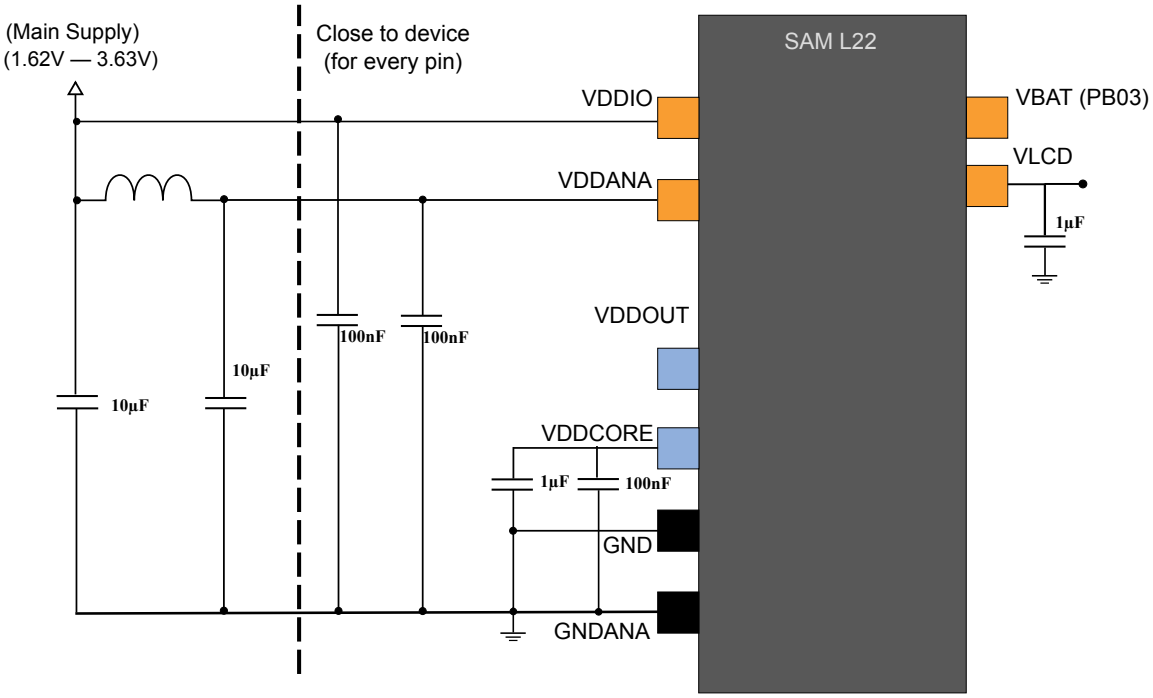


Figure 48-3. Power Supply Connection for Battery Backup

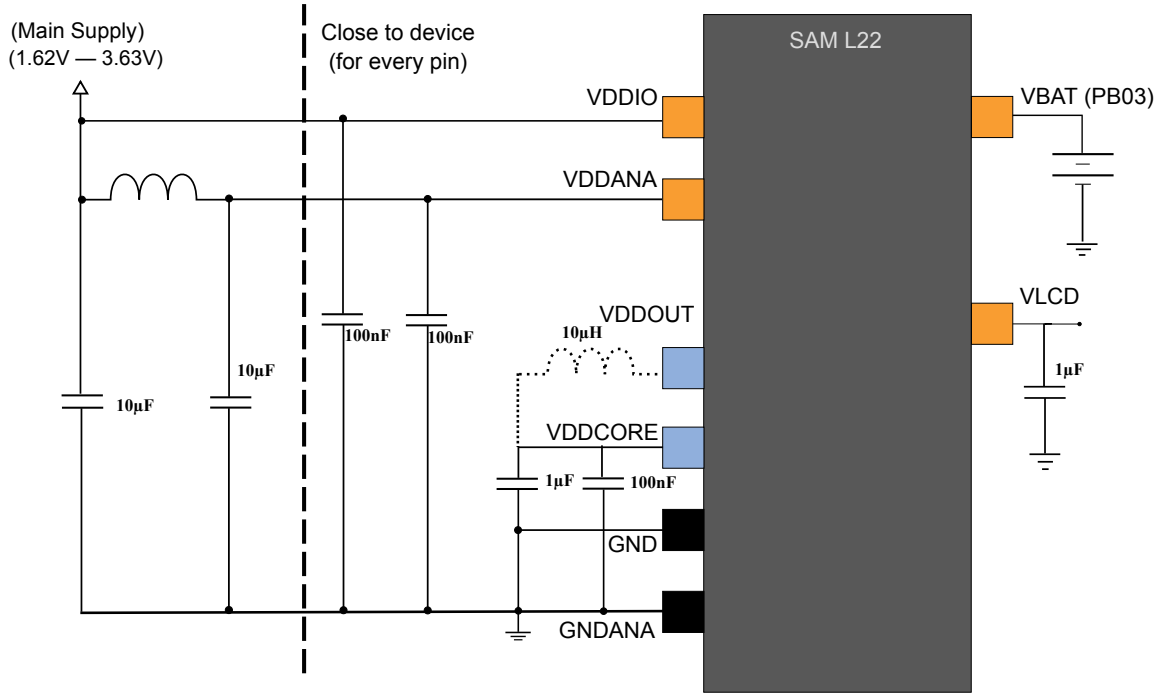


Table 48-1. Power Supply Connections, V_{DDCORE} or V_{DDOUT} From Internal Regulator

Signal Name	Recommended Pin Connection	Description
V_{DDIO}	1.62V to 3.63V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Decoupling/filtering inductor 10µH(1)(3)	Digital supply voltage
V_{DDANA}	1.62V to 3.63V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Ferrite bead(4) prevents the V_{DD} noise interfering with V_{DDANA}	Analog supply voltage
V_{VDDOUT}	Switching regulator mode: 10µH inductor with saturation current above 150mA and DCR<1Ω Linear regulator mode: Not connected	On-chip switching mode regulator output
V_{DDCORE}	0.9V to 1.2V typical Decoupling/filtering capacitors 100nF(1)(2) and 1µF(1)	Linear regulator mode: Core supply voltage output/ external decoupling pin Switched regulator mode: Core supply voltage input, must be connected to V_{DDOUT} via inductor
V_{BAT}	1.62V to 3.63V when connected(6)	External battery supply input
V_{LCD}	LCD Bias Voltage Analog input/output	Powers the LCD voltage pump

Signal Name	Recommended Pin Connection	Description
	When V_{LCD} is generated internally, the output voltage range is 2.5V to 3.5V. When V_{LCD} is provided externally, the input voltage range is 2.4V to 3.6V. External VLCD capacitor 1µF(1)(5) **Note:** Leave VLCD pin unconnected when the LCD controller is not used.	
GND	Common ground	Ground
GND_{ANA}	Ground for VDDANA	Ground for the analog power domain

1. These values are only given as a typical example.
2. Decoupling capacitors should be placed close to the device for each supply pin pair in the signal group, low ESR capacitors should be used for better decoupling.
3. An inductor should be added between the external power and the V_{DD} for power filtering.
4. A ferrite bead has better filtering performance compared to standard inductor at high frequencies. A ferrite bead can be added between the main power supply (V_{DDIO}) and V_{DDANA} to prevent digital noise from entering the analog power domain. The bead should provide enough impedance (e.g. 50Ω at 20MHz and 220Ω at 100MHz) to separate the digital and analog power domains. Make sure to select a ferrite bead designed for filtering applications with a low DC resistance to avoid a large voltage drop across the ferrite bead.
5. The external capacitor 1µF on the V_{LCD} pin is only required if V_{LCD} is generated internally (SLCD.CTRLA.XVLCD=0). When the SLCD controller is not used at all, the VLCD pin can remain unconnected - in this case, no external capacitor required, too.
6. When the V_{BAT} pin is used for battery supply, PB03 (V_{BAT}) cannot be used as PORT I/O pin or for any other multiplexed peripheral signal.

48.2.2 Special Considerations for QFN Packages

The QFN package has an exposed paddle that must be connected to GND.

48.3 External Analog Reference Connections

The following schematic checklist is only necessary if the application is using one or more of the external analog references. If the internal references are used instead, the following two circuits are not necessary.

Figure 48-4. External Analog Reference Schematic With Two References

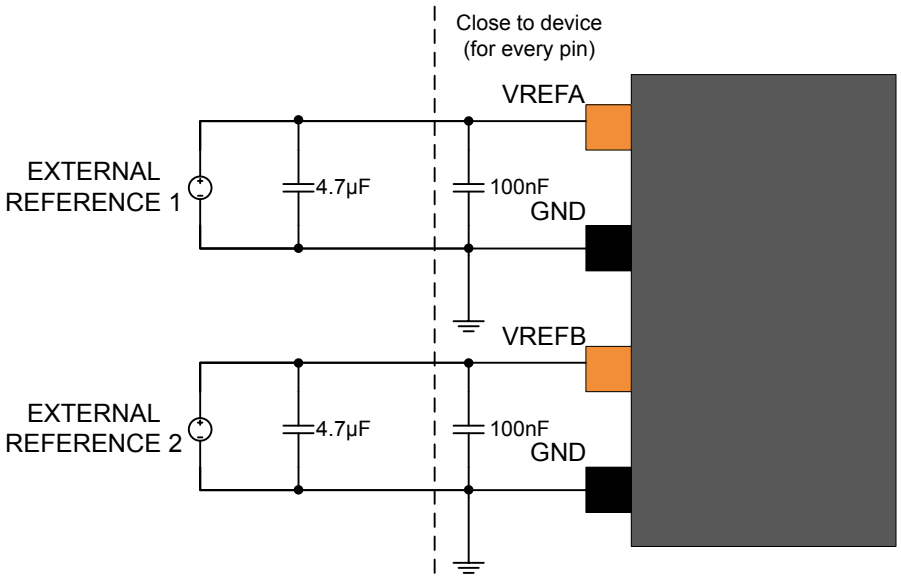


Figure 48-5. External Analog Reference Schematic With One Reference

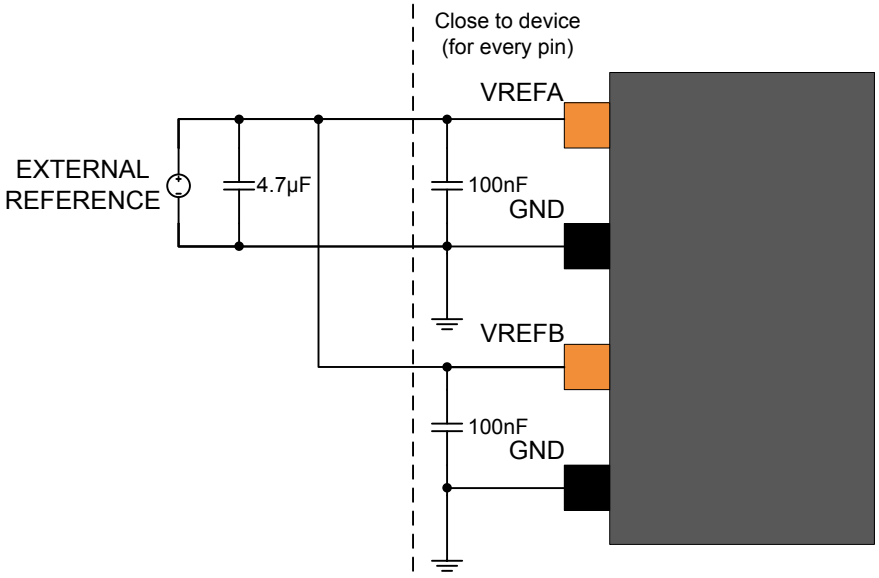


Table 48-2. External Analog Reference Connections

Signal Name	Recommended Pin Connection	Description
V_{REFA}/V_{REFB}	1.0V to (V_{DDANA} - 0.6V) for ADC Decoupling capacitors 100nF(1)(2) and 4.7µF(1)	External reference $V_{REFA}/V_{REFB}$for the analog port
GND		Ground

1. These values are only given as a typical example.

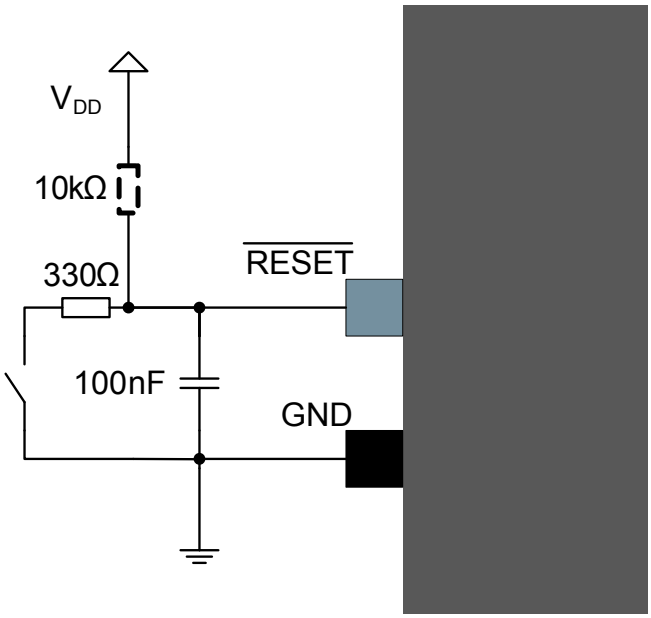
2. Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

48.4 External Reset Circuit

The external Reset circuit is connected to the $\overline{\text{RESET}}$ pin when the external Reset function is used. The circuit is not necessary when the $\overline{\text{RESET}}$ pin is not driven LOW externally by the application circuitry.

The reset switch can also be removed, if a manual reset is not desired. The $\overline{\text{RESET}}$ pin itself has an internal pull-up resistor, hence it is optional to add any external pull-up resistor.

Figure 48-6. External Reset Circuit Schematic



A pull-up resistor makes sure that the reset does not go low and unintentionally causing a device reset. An additional resistor has been added in series with the switch to safely discharge the filtering capacitor, i.e. preventing a current surge when shorting the filtering capacitor which again can cause a noise spike that can have a negative effect on the system.

Table 48-3. Reset Circuit Connections

Signal Name	Recommended Pin Connection	Description
$\overline{\text{RESET}}$	Reset low level threshold voltage V_{DDIO} = 1.6V - 2.0V: Below 0.33 * V_{DDIO} V_{DDIO} = 2.7V - 3.6V: Below 0.36 * V_{DDIO} Decoupling/filter capacitor 100nF(1) Pull-up resistor 10kΩ(1)(2) Resistor in series with the switch 330Ω(1)	Reset pin

1. These values are only given as a typical example.

2. The SAM L22 features an internal pull-up resistor on the $\overline{\text{RESET}}$ pin, hence an external pull-up is optional.

48.5 Unused or Unconnected Pins

For unused pins the default state of the pins will give the lowest current leakage. Thus there is no need to do any configuration of the unused pins in order to lower the power consumption.

48.6 Clocks and Crystal Oscillators

The SAM L22 can be run from internal or external clock sources, or a mix of internal and external sources. An example of usage can be to use the internal 16MHz oscillator as source for the system clock and an external 32.768kHz watch crystal as clock source for the Real-Time counter (RTC).

48.6.1 External Clock Source

Figure 48-7. External Clock Source Schematic

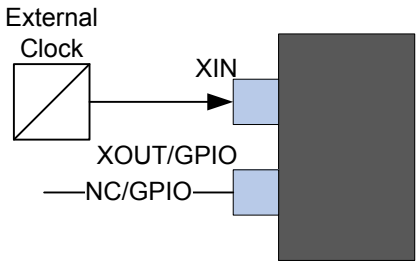
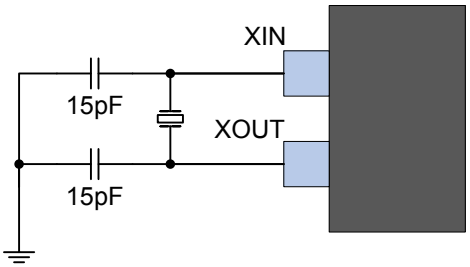


Table 48-4. External Clock Source Connections

Signal Name	Recommended Pin Connection	Description
XIN	XIN is used as input for an external clock signal	Input for inverting oscillator pin
XOUT/GPIO	Can be left unconnected or used as normal GPIO	NC/GPIO

48.6.2 Crystal Oscillator

Figure 48-8. Crystal Oscillator Schematic



The crystal should be located as close to the device as possible. Long signal lines may cause too high load to operate the crystal, and cause crosstalk to other parts of the system.

Table 48-5. Crystal Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN	Load capacitor 15pF(1)(2)	External crystal between 0.4 to 32MHz
XOUT	Load capacitor 15pF(1)(2)	

1. These values are only given as a typical example.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

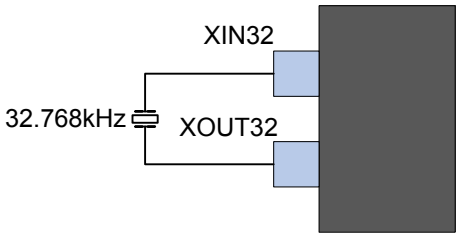
48.6.3 External Real Time Oscillator

The low frequency crystal oscillator is optimized for use with a 32.768kHz watch crystal. When selecting crystals, load capacitance and the crystal's Equivalent Series Resistance (ESR) must be taken into consideration. Both values are specified by the crystal vendor.

SAM L22 oscillator is optimized for very low power consumption, hence close attention should be made when selecting crystals.

The typical parasitic load capacitance values are available in the Electrical Characteristics section. This capacitance and PCB capacitance can allow using a crystal inferior to 12.5pF load capacitance without external capacitors as shown in Figure 48-9.

Figure 48-9. External Real Time Oscillator without Load Capacitor



To improve accuracy and Safety Factor, the crystal datasheet can recommend adding external capacitors as shown in Figure 48-10.

To find suitable load capacitance for a 32.768kHz crystal, consult the crystal datasheet.

Figure 48-10. External Real Time Oscillator with Load Capacitor

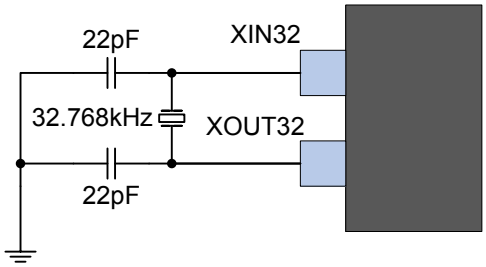


Table 48-6. External Real Time Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN32	Load capacitor 22pF(1)(2)	Timer oscillator input
XOUT32	Load capacitor 22pF(1)(2)	Timer oscillator output

1. These values are only given as typical examples.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

Note: In order to minimize the cycle-to-cycle jitter of the external oscillator, keep the neighboring pins as steady as possible. For neighboring pin details, refer to the Oscillator Pinout section.

Related Links

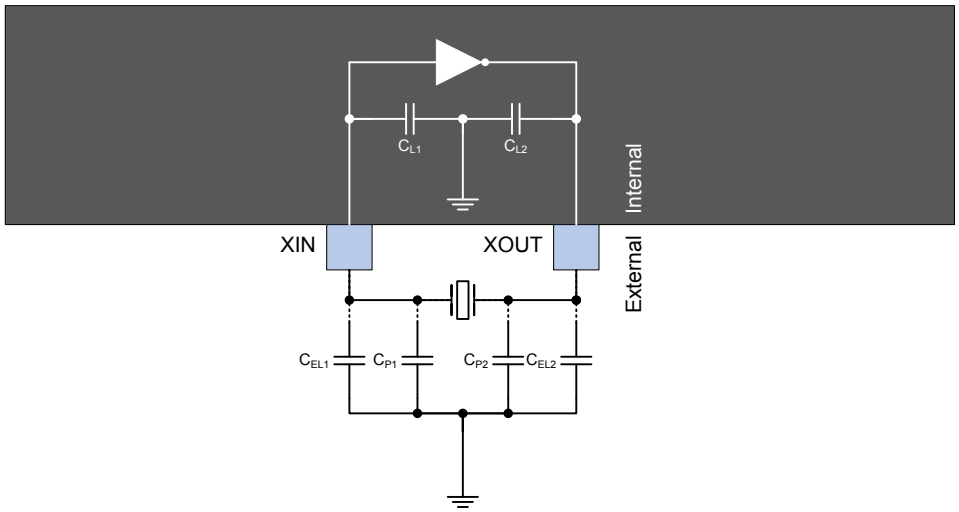
Oscillator Pinout

Oscillators Characteristics

48.6.4 Calculating the Correct Crystal Decoupling Capacitor

The model shown in Figure 48-11 can be used to calculate correct load capacitor for a given crystal. This model includes internal capacitors C_{Ln}, external parasitic capacitance C_{ELn} and external load capacitance C_{Pn}.

Figure 48-11. Crystal Circuit With Internal, External and Parasitic Capacitance



Using this model the total capacitive load for the crystal can be calculated as shown in the equation below:

$$\sum C_{\text{tot}} = \frac{(C_{L1} + C_{P1} + C_{\text{EL1}})(C_{L2} + C_{P2} + C_{\text{EL2}})}{C_{L1} + C_{P1} + C_{\text{EL1}} + C_{L2} + C_{P2} + C_{\text{EL2}}}$$

where C_{tot} is the total load capacitance seen by the crystal. This value should be equal to the load capacitance value found in the crystal manufacturer datasheet.

The parasitic capacitance C_{ELn} can in most applications be disregarded as these are usually very small. If accounted for, these values are dependent on the PCB material and PCB layout.

For some crystal the internal capacitive load provided by the device itself can be enough. To calculate the total load capacitance in this case. C_{ELn} and C_{Pn} are both zero, $C_{L1} = C_{L2} = C_L$, and the equation reduces to the following:

$$\sum C_{\text{tot}} = \frac{C_L}{2}$$

See the related links for equivalent internal pin capacitance values.

Related Links

External 32KHz Crystal Oscillator (XOSC32K) Characteristics

48.7 Programming and Debug Ports

For programming and/or debugging the SAM L22, the device should be connected using the Serial Wire Debug, SWD, interface. Currently the SWD interface is supported by several Microchip and third party programmers and debuggers, like the Atmel-ICE, SAM-ICE or SAM L22 Xplained Pro (SAM L22 evaluation kit) Embedded Debugger.

Refer to the Atmel-ICE, SAM-ICE or SAM L22 Xplained Pro user guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

The SAM L22 Xplained Pro evaluation board supports programming and debugging through the onboard embedded debugger so no external programmer or debugger is needed.

Note: A pull-up resistor on the SWCLK pin is critical for reliable operation. Refer to related link for more information.

Figure 48-12. SWCLK Circuit Connections

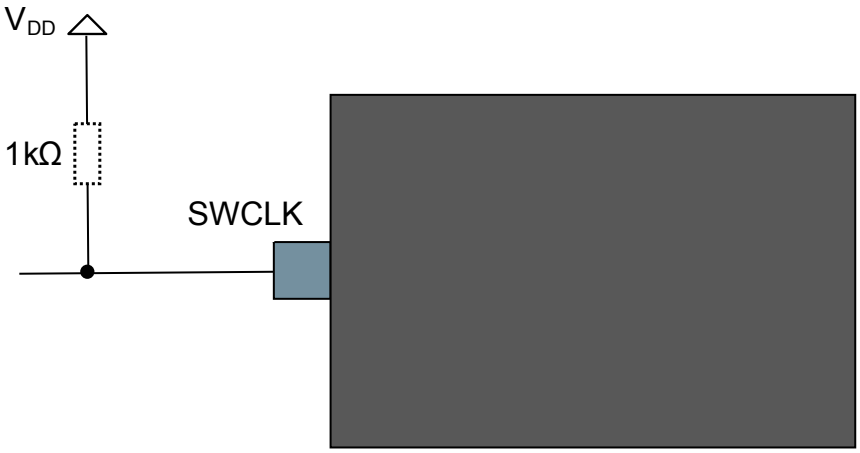


Table 48-7. SWCLK Circuit Connections

Pin Name	Description	Recommended Pin Connection
SWCLK	Serial wire clock pin	Pull-up resistor 1kΩ

Related Links

Operation in Noisy Environment

48.7.1 Cortex Debug Connector (10-pin)

For debuggers and/or programmers that support the Cortex Debug Connector (10-pin) interface the signals should be connected as shown in Figure 48-13 with details described in Table 48-8.

Figure 48-13. Cortex Debug Connector (10-pin)

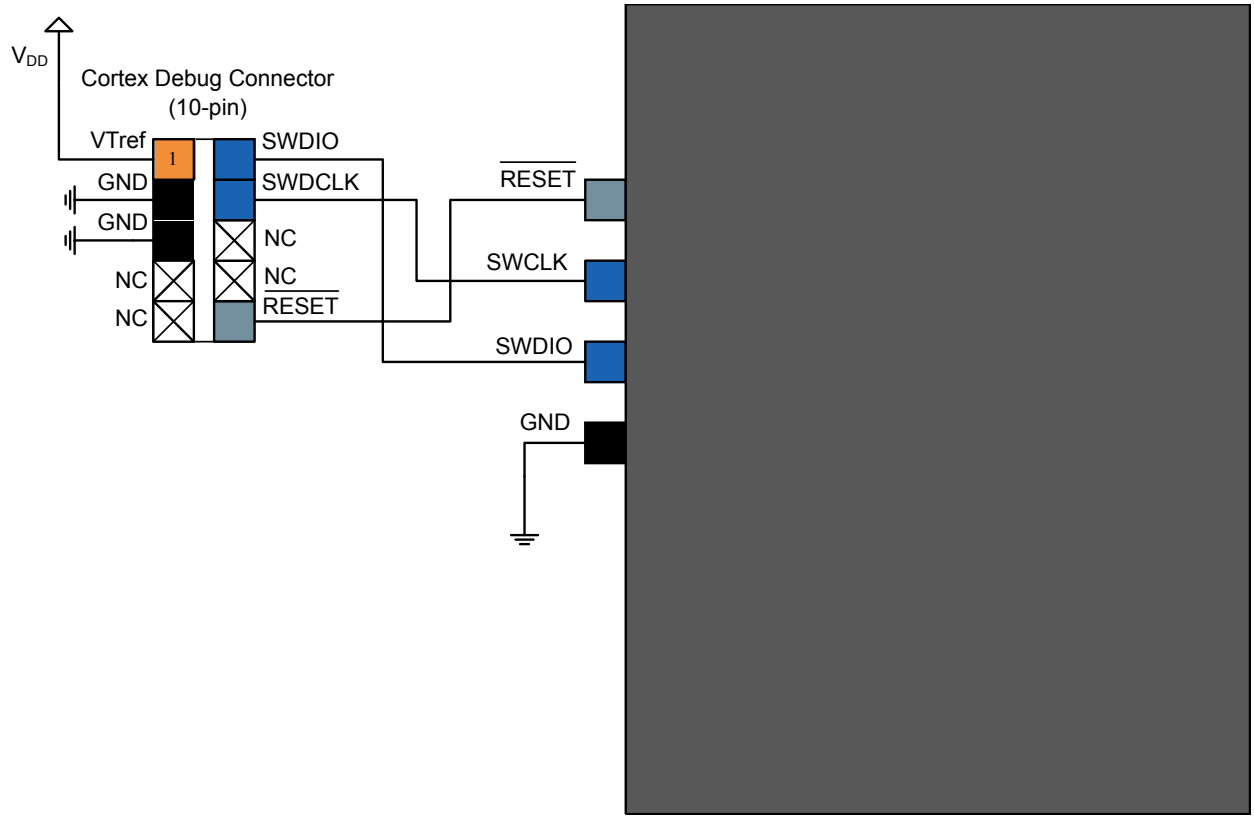


Table 48-8. Cortex Debug Connector (10-pin)

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTref	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

48.7.2 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

The JTAGICE3 debugger and programmer does not support the Cortex Debug Connector (10-pin) directly, hence a special pinout is needed to directly connect the SAM L22 to the JTAGICE3, alternatively one can use the JTAGICE3 squid cable and manually match the signals between the JTAGICE3 and SAM L22. Figure 48-14 describes how to connect a 10-pin header that support connecting the JTAGICE3 directly to the SAM L22 without the need for a squid cable. This can also be used for the Atmel-ICE AVR connector port.

The JTAGICE3 squid cable or the JTACICE3 50mil cable can be used to connect the JTAGICE3 programmer and debugger to the SAM L22. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface illustrates the correct pinout for the JTAGICE3 50 mil, and details are given in Table 48-9.

Figure 48-14. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

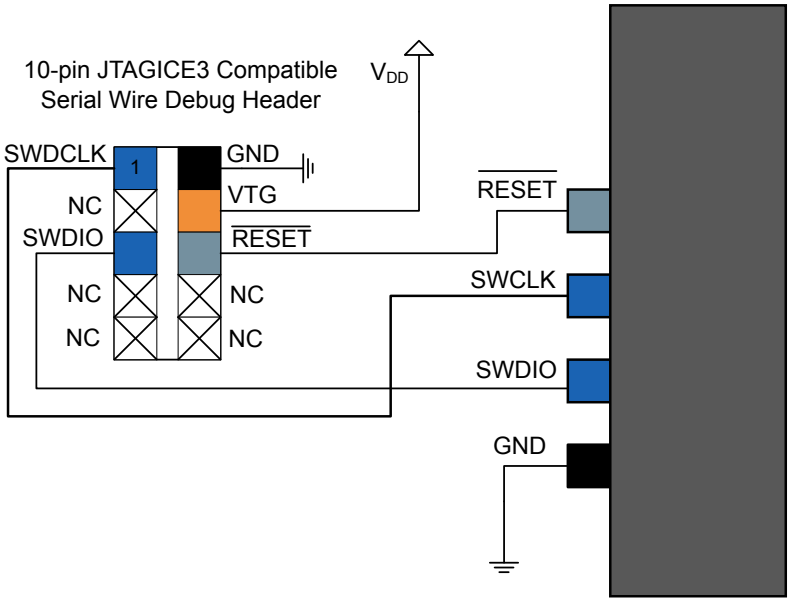


Table 48-9. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTG	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

48.7.3 20-pin IDC JTAG Connector

For debuggers and/or programmers that support the 20-pin IDC JTAG Connector, e.g. the SAM-ICE, the signals should be connected as shown in Figure 48-15 with details described in Table 48-10.

Figure 48-15. 20-pin IDC JTAG Connector

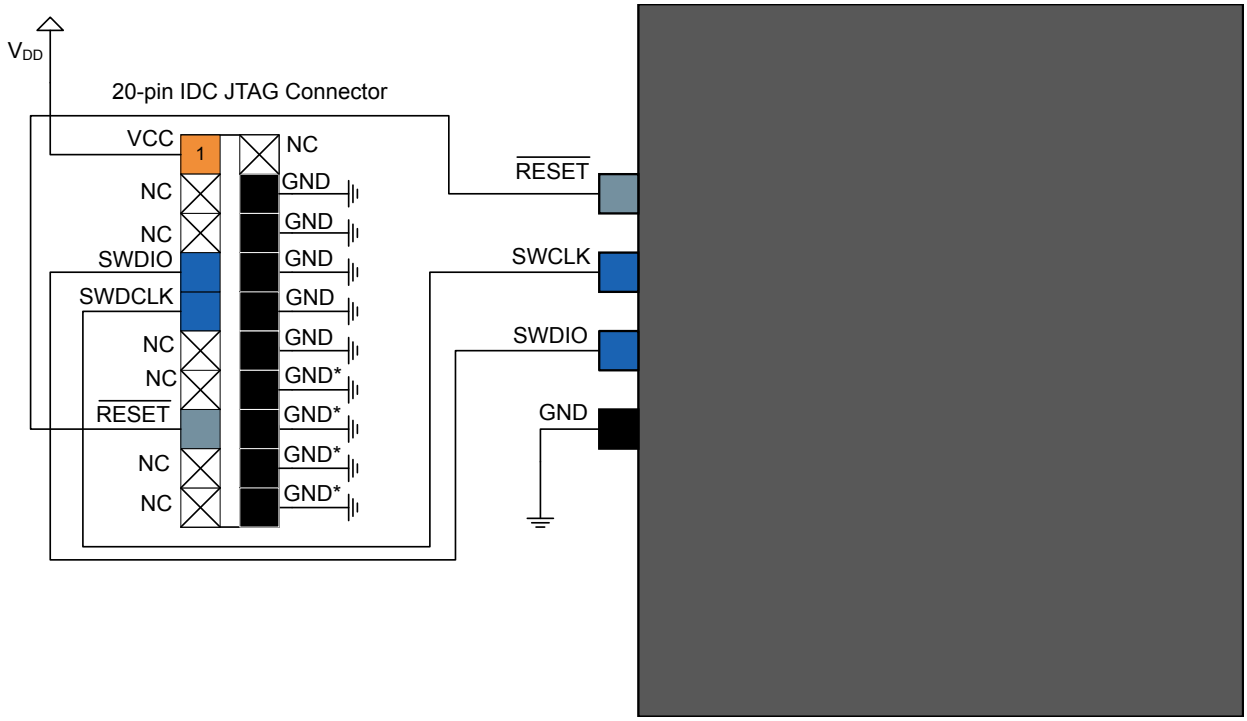


Table 48-10. 20-pin IDC JTAG Connector

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VCC	Target voltage sense, should be connected to the device V_{DD}
GND	Ground
GND*	These pins are reserved for firmware extension purposes. They can be left unconnected or connected to GND in normal debug environment. They are not essential for SWD in general.

48.8 USB Interface

The USB interface consists of a differential data pair (D+/D-) and a power supply (VBUS, GND).

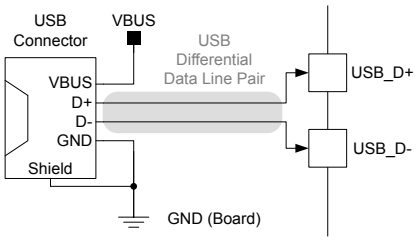
Refer to the Electrical Characteristics section for operating voltages which will allow USB operation.

Table 48-11. USB Interface Checklist

Signal Name	Recommended Pin Connection	Description
D+	• The impedance of the pair should be matched on the PCB to minimize reflections.	USB full speed / low speed positive data upstream pin

Signal Name	Recommended Pin Connection	Description
D-	• USB differential tracks should be routed with the same characteristics (length, width, number of vias, etc.) • For a tightly coupled differential pair,the signal routing should be as parallel as possible, with a minimum number of angles and vias.	USB full speed / low speed negative data upstream pin

Figure 48-16. Low Cost USB Interface Example Schematic

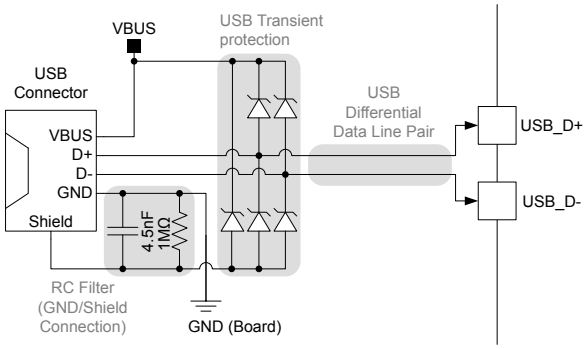


It is recommended to increase ESD protection on the USB D+, D-, and VBUS lines using dedicated transient suppressors. These protections should be located as close as possible to the USB connector to reduce the potential discharge path and reduce discharge propagation within the entire system.

The USB FS cable includes a dedicated shield wire that should be connected to the board with caution. Special attention should be paid to the connection between the board ground plane and the shield from the USB connector and the cable.

Tying the shield directly to ground would create a direct path from the ground plane to the shield, turning the USB cable into an antenna. To limit the USB cable antenna effect, it is recommended to connect the shield and ground through an RC filter.

Figure 48-17. Protected USB Interface Example Schematic



Related Links

Electrical Characteristics

48.9 LCD

Refer to the Electrical Characteristics for LCD interface related parameters like V_{LCD} ,I_{LCD} ,C_{VLCD} etc.

The maximum number of segment LCD pins is based on the number of pins in the package (100/64/48 pin package). SLCD Pins can be configured also as GPIOs when SLCD controller is not used: for the

100-pin package, 44 SLCD pins can be configured as GPIO pins, 8 SLCD pins can be used only as GP input pins.

Related Links

Configuration Summary

I/O Multiplexing and Considerations

Electrical Characteristics

48.10 SERCOM I²C Pins

Not all the pins/instances of SERCOM instance can be used for I²C operation.

Related Links

I/O Multiplexing and Considerations

SERCOM USART and I2C Configurations

48.11 Pin Characteristics

Note: The electrical characteristics differ, based on the type of the IO pin, such as standard (normal) I/O pins, high sink current I/O pins, LCD pins in I/O configuration etc.

Related Links

I/O Multiplexing and Considerations

48.12 Reference Schematic

The SAM L22 Xplained Pro Evaluation Kit schematic can be used as a starting point for reference when designing with the SAM L22 devices.

The SAM L22 Xplained Pro Evaluation Kit user guide contains details about pin assignment of the extension interfaces on the board and other hardware information.

This user guide is available as pdf file, schematic is available as zip file (in folder icon) under in the ‘user guide’ section at the following link:

http://www.atmel.com/tools/ATSAML22-XPRO.aspx?tab=documents

49. Errata

The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

49.1 Die Revision A

49.1.1 AC

1 – After a Power-on Reset, the capacitance on PA02 is offset by an amount that slowly decays during 5 seconds, making any touch-related measurements on this pin unreliable.
Errata reference: 15745
Fix/Workaround:
To get rid of this offset, reconfigure the AC0 input muxes to use internal inputs instead of AC0 pin 0(default reset value) before starting any touch-related measurements
(COMPCTRL0.MUXPOS=4,COMPCTRL0.MUXNEG=5).

49.1.2 PM

1 – If the PM.STDBYCFG.VREGSMOD field is set to 2 (Low Power configuration), the oscillator source driving the GCLK_MAIN clock will still be running in standby mode causing extra consumption.
Errata reference: 14539
Fix/Workaround:
Before entering in standby mode, switch the GCLK_MAIN to the OSCULP32K clock. After wakeup, switch back to the GCLK_MAIN clock.

49.1.3 DFLL48M

1 – The DFLL clock must be requested before being configured otherwise a write access to a DFLL register can freeze the device.
Errata reference: 9905
Fix/Workaround:
Write a zero to the DFLL ONDEMAND bit in the DFLLCTRL register before configuring the DFLL module.

2 – The DFLL status bits in the STATUS register during the USB clock recovery mode can be wrong after a USB suspend state.
Errata reference: 16193
Fix/Workaround:
Do not monitor the DFLL status bits in the STATUS register during the USB clock recovery mode.

3 – If the DFLL48M reaches the maximum or minimum COARSE or FINE calibration values during the locking sequence, an out of bounds interrupt will be generated. These interrupts will be generated even if the final calibration values at DFLL48M lock are not at maximum or minimum, and might therefore be false out of bounds interrupts.
Errata reference: 16192
Fix/Workaround:

Check that the lockbits: DFLLLCKC and DFLLLCKF in the OSCCTRL Interrupt Flag Status and Clear register (INTFLAG) are both set before enabling the DFLLOOB interrupt.

49.1.4 FDPLL

1 – When changing on-the-fly the FDPLL ratio in DPLLnRATIO register, STATUS.DPLLnLDRTO will not be set when the ratio update will be completed.
Errata reference: 15753
Fix/Workaround:
Wait for the interruption flag INTFLAG.DPLLnLDRTO instead.

49.1.5 PORT

1 – PORT read/write attempts on non-implemented registers, including addresses beyond the last implemented register group (PA, PB,...) do not generate a PAC protection error.
Errata reference: 15611
Fix/Workaround:
None

49.1.6 SUPC

1 – When Buck converter is set as main voltage regulator (SUPC.VREG.SEL=1), the microcontroller can freeze when leaving standby mode.
Errata reference: 15264
Fix/Workaround:
Enable the main voltage regulator in standby mode (SUPC.VREG.RUNSTDBY=1) and set the Standby in PL0 bit to one (SUPC.VREG.STDBYPL0=1).

49.1.7 Device

1 – Writing PM.STDBYCFG.VREGSMOD to one does not set the main voltage regulator in standby mode, the low power regulator is still used in standby mode.
Errata reference: 14543
Fix/Workaround :
Set SUPC.VREG.RUNSTDBY to one.

2 – Five PTC lines are mapped on PC00, PC01, PB00, PB01, PB02 instead of PC05, PC06, PA11, PA10, PA09
Errata reference: 14792
Fix/Workaround:
None.

3 – When LCD feature is enable and (VLCD - VDD) > 0.7V, an extra consumption occurs. In case of VLCD internally generated, the VLCD voltage will be out of specification.
Errata reference: 14696
Fix/Workaround :
LCD feature must be used only when (VLCD - VDD) < 0.7V

4 – When ABS(VLCD - VDD) < 50mV, an extra consumption can occur on VLCD (if VLCD generated externally) or on VDD (if VLCD generated internally).
Errata reference: 14742
Fix/Workaround :
ABS(VLCD - VDD) should be greater than 50mV

5 – When VBAT>VDDANA, in battery backup mode or in battery forced mode (SUPC.BBPS.CONF=FORCED) an overconsumption appears due to too high voltage on PC00, PC01, PB00, PB01, PB02 pins.
Errata reference: 14643
Fix/Workaround:
PC00, PC01, PB00, PB01, PB02 should be tied to GND.

6 – When configured in HS or FastMode+, SDA and SCL fall times are shorter than I2C specification requirement and can lead to reflection.
Errata reference: 16225
Fix/Workaround:
When reflection is observed a 100 ohms serial resistor can be added on the impacted line.

7 – SAML22G devices delivered before date code 1716 only have 3 sercoms available (0,1,2) instead of 4 (0,1,2,3)
Errata reference: 15511
Fix/Workaround:
None.

49.1.8 RTC

1 – When the tamper controller is configured for asynchronous detection, an RTC tamper interrupt can occur while the RTC is disabled.
Errata reference: 14653
Fix/Workaround:
Set the tamper interrupt enable only when the RTC is enabled.
- Program INTEN.TAMPER=1 after setting CTRLA.ENABLE register and clearing INTFLAG.TAMPER register.
- Program INTEN.TAMPER=0 before clearing CTRLA.ENABLE register.

2 – When the tamper controller is configured for CAPTURE while the RTC is disabled, a noisy pin can trigger the following once the RTC is enabled:
- the timestamp capture
- the tamper interrupt if enabled
- the DMA trigger if enabled.
Errata reference: 15092
Fix/Workaround:
(1) Set the tamper interrupt enable only when the RTC is enabled:
- Clear the tamper interrupt flags & ID registers (INTFLAG.TAMPER & TAMPID.TAMPIDx registers).
- Enable RTC (CTRLA.ENABLE = 1).
- Enable the tamper interrupt (INTEN.TAMPER = 1)
To disable the RTC, first disable the Tamper interrupts before disabling the RTC.

- Disable Tamper interrupts (INTEN.TAMPER=0)
- Disable the RTC (CTRLA.ENABLE=0).
(2) Issue a CPU read of the TIMESTAMP register immediately after the RTC is enabled. This releases the register lock allowing the capture of the next (and valid) tamper. This releases the DMA trigger of the erroneous capture tamper.

3 – When the tamper controller is configured for ACTL, the mismatch signal used to qualify the DMA and interrupt triggers produces different results. The DMA implements a level-detection whereas the interrupt implements an edge-detection. The result is that the DMA may trigger frequently from the same mismatch compared to the interrupt which will only trigger once.
Errata reference: 14692
Fix/Workaround:
If no other tamper configurations are implemented (i.e. other TAMPCTRL.INxACT != WAKE/CAPTURE and EVCTRL.EVEI=0), do not enable the DMA if possible to prevent performance degradation.

4 – When the tamper controller is configured for timestamp capture, the RTC tamper interrupt occurs before the TIMESTAMP register is updated.
Errata reference: 14654
Fix/Workaround:
Two workarounds are available.
- Use the DMA trigger to determine when the TIMESTAMP value is registered. The DMA trigger sets after the TIMESTAMP register update.
- Implement a wait loop to create a delay when the tamper interrupt handler routine begins to when the TIMESTAMP register is read. The delay must be long enough to wait for 3x CLK_RTC period. Example:
* If CLK_RTC frequency is 1KHz, the delay must be at least 3ms.
* If CLK_RTC frequency is 32KHz, the delay must be at least 92us.

5 – If PB01 is multiplexed to RTC peripheral (RTC/IN2), the system will always see this input pin as logic ‘0’ when backup mode is entered. If the detection transition (TAMPCTRL.TAMLVL2 = 0), it might falsely wake up the system.
Errata reference: 15010
Fix/Workaround:
If the system is expected to enter backup mode, use other tamper pins (IN0/IN1/IN3/IN4) for tamper detection.

6 – When the tamper controller is configured for active layer mode, the RTC tamper interrupt occurs before the TIMESTAMP register is updated.
Errata reference: 14810
Fix/Workaround:
Two workarounds are available.
- Use the DMA trigger to determine when the TIMESTAMP value is registered. The DMA trigger sets after the TIMESTAMP register update.
- Implement a wait loop to create a delay when the tamper interrupt handler routine begins to when the TIMESTAMP register is read. The delay must be long enough to wait for 3x CLK_RTC period. Example:

* If CLK_RTC frequency is 1KHz, the delay must be at least 3ms.
* If CLK_RTC frequency is 32KHz, the delay must be at least 92us.

49.1.9 ADC

1 – Overconsumption for up to 1.6 seconds on VDDANA when the ADC is disabled(manually or automatically)
Errata reference: 14349
Fix/Workaround :
None

2 – The ADC Effective number of Bits (ENOB) is 9.2 in this revision

49.1.10 TRNG

1 – When TRNG is enabled with configuration CTRL.RUNSTDBY = 0, (disabled during sleep) it could still continue to operate resulting in over-consumption (~50uA) in standby mode.
Errata reference: 14827
Fix/Workaround:
Disable the TRNG before entering standby mode.

49.1.11 EVSYS

1 – The acknowledge between an event user and the EVSYS clears the CHSTATUS.CHBUSYn bit before this information is fully propagated in the EVSYS one GCLK_EVSYS_CHANNEL_n clock cycle later. As a consequence, any generator event occurring on that channel before that extra GCLK_EVSYS_CHANNEL_n clock cycle will trigger the overrun flag.
Errata reference: 14835
Fix/Workaround:
For applications using event generators other than the software event, monitor the OVR flag.
For applications using the software event generator, wait one GCLK_EVSYS_CHANNEL_n clock cycle after the CHSTATUS.CHBUSYn bit is cleared before issuing a software event.

2 – Using synchronous, spurious overrun can appear with generic clock for the channel always on.
Errata reference: 14532
Fix/Workaround:
- Request the generic clock on demand by setting the CHANNEL.ONDEMAND bit to one.
- No penalty is introduced.

49.1.12 DMAC

1 – When using more than one DMA channel and if one of these DMA channels has a linked descriptor, a fetch error can appear on this channel.
Errata reference: 15670
Fix/Workaround:
Do not use linked descriptors, make a software link instead:

Replace the channel which used linked descriptor by two channels DMA (with linked descriptor disabled) handled by two channels event system:
- DMA channel 0 transfer completion is able to send a conditional event for DMA channel 1 (via event system with configuration of BTCTRL.EVOSEL=BLOCK for channel 0 and configuration CHCTRLB.EVACT=CBLOCK for channel 1).
- On the transfer complete reception of the DMA channel 0, immediately re-enable the channel 0.
- Then DMA channel 1 transfer completion is able to send a conditional event for DMA channel 0 (via event system with configuration of BTCTRL.EVOSEL=BLOCK for channel 1 and configuration CHCTRLB.EVACT=CBLOCK for channel 0).
- On the transfer complete reception of the DMA channel 1, immediately re-enable the channel 1.
- The mechanism can be launched by sending a software event on the DMA channel 0.

2 – A write from DMAC to a register in a module to disable a trigger from the module to DMAC, does not work in standby mode.
(For example DAC, SLCD, SERCOM in transmission mode)
Errata reference: 14648
Fix/Workaround:
If the module generating the trigger also generates event, use event interface instead of triggers with DMAC (for example SLCD)

3 – When at least one channel using linked descriptors is already active, enabling another DMA channel (with or without linked descriptors) can result in a channel Fetch Error (FERR) or an incorrect descriptor fetch.
This happens if the channel number of the channel being enabled is lower than the channel already active.
Errata reference: 15683
Fix/Workaround:
When enabling a DMA channel while other channels using linked descriptors are already active, the channel number of the new channel enabled must be greater than the other channel numbers.

49.1.13 EIC

1 – Changing the NMI configuration(CONFIGn.SENSEx)on the fly may lead to a false NMI interrupt.
Errata reference: 15279
Fix/Workaround:
Clear the NMIFLAG bit once the NMI has been modified.

2 – When the asynchronous edge detection is enabled, and the system is in standby mode, only the first edge will generate an event. The following edges won't generate events until the system wakes up.
Errata reference: 16103
Fix/Workaround:
Asynchronous edge detection doesn't work, instead use the synchronous edge detection (ASYNCH.ASYNCH[x]=0). In order to reduce power consumption when using synchronous edge detection, either set the

GCLK_EIC frequency as low as possible or select the ULP32K clock (EIC CTRLA.CKSEL=1).

3 – Access to EIC_ASYNCH register in 8/16-bit mode is not functional.
*** Writing in 8-bit mode will also write this byte in all bytes of the 32-bit word.**
*** Writing higher 16-bits will also write the lower 16-bits.**
*** Writing lower 16-bits will also write the higher 16-bits.**
Errata reference: 14417
Fix/Workaround:
Two workarounds are available.
- Use 32-bit write mode.
- Write only lower 16-bits (This will write upper 16-bits also, but does not impact the application).

4 – When the EIC is configured to generate an interrupt on a low level or rising edge or both edges (CONFIGn.SENSEx) with the filter enabled (CONFIGn.FILTENx), a spurious flag might appear for the dedicated pin on the INTFLAG.EXTINT[x] register as soon as the EIC is enabled using CTRLA ENABLE bit.
Errata reference: 15278
Fix/Workaround:
Clear the INTFLAG bit once the EIC enabled and before enabling the interrupts.

49.1.14 SERCOM

1 – In USART autobaud mode, missing stop bits are not recognized as inconsistent sync (ISF) or framing (FERR) errors.
Errata reference: 13852
Fix/Workaround:
None

2 – In ISO7816 T0 mode when start of frame detect is enabled(CTRLB.SFDE=1), if there is a parity error, receive start (INTFLAG.RXS) can be erroneously set. This is because the transmitted parity low is also seen by the receiver and looks like a start of frame.
Errata reference: 14694
Fix/Workaround:
Clear INTFLAG.RXS when received on parity error.

49.1.15 TC

1 – When clearing STATUS.xxBUFV flag, SYNCBUSY is released before the register is restored to its appropriate value.
Errata reference: 15056
Fix/Workaround:
To ensure that the register value is properly restored before updating this same register through xx or xxBuf with a new value, the STATUS.xxBUFV flag must be cleared successively two times.

49.1.16 TCC

1 – Using TCC in dithering mode with external retrigger events can lead to unexpected stretch of right aligned pulses, or shrink of left aligned pulses.
Errata reference: 15625
Fix/Workaround:
Do not use retrigger events/actions when TCC is configured in dithering mode.

2 – When clearing STATUS.xxBUFV flag, SYNCBUSY is released before the register is restored to its appropriate value.
Errata reference: 15057
Fix/Workaround:
To ensure that the register value is properly restored before updating this same register through xx or xxBUF with a new value, the STATUS.xxBUFV flag must be cleared successively two times.

3 – Advance capture mode (CAPTMIN CAPTMAX LOCMIN LOCMAX DERIV0) doesn't work if an upper channel is not in one of these mode. Example: when CC[0]=CAPTMIN, CC[1]=CAPTMAX, CC[2]=CAPTEN, and CC[3]=CAPTEN, CAPTMIN and CAPTMAX won't work.
Errata reference: 14817
Fix/Workaround:
Basic capture mode must be set in lower channel and advance capture mode in upper channel.
Example: CC[0]=CAPTEN , CC[1]=CAPTEN , CC[2]=CAPTMIN, CC[3]=CAPTMAX
All capture will be done as expected.

49.1.17 ADC

1 – In standby sleep mode when the ADC is in free-running mode (CTRLC.FREERUN=1) and the RUNSTDBY bit is set to 0 (CTRLA.RUNSTDBY=0), the ADC keeps requesting its generic clock.
Errata reference: 15463
Fix/Workaround:
Stop the free-running mode (CTRLC.FREERUN=0) before entering standby sleep mode

2 – ADC SYNCBUSY.SWTRIG get stuck to one after wakeup from standby sleep mode.
Errata reference: 16027
Fix/Workaround:
Ignore ADC SYNCBUSY.SWTRIG status when waking up from standby sleep mode. ADC result can be read after INTFLAG.RESRDY is set. To start the next conversion, write a '1' to SWTRIG.START.

3 – The LSB of ADC result is stuck at zero, in unipolar mode for 8-bit and 10-bit resolution.
Errata reference: 14431
Fix/Workaround:
Use 12-bit resolution and take only least 8 bits or 10 bits, if necessary.

4 – If a synchronized event is received during an ADC conversion, the ADC will not acknowledge the event, causing a stall of the event channel.
Errata reference: 14795
Fix/Workaround:
When using events with the ADC, only the asynchronous path from the Event System must be used.

49.2 Die Revision B

49.2.1 AC

1 – After a Power-on Reset, the capacitance on PA02 is offset by an amount that slowly decays during 5 seconds, making any touch-related measurements on this pin unreliable.
Errata reference: 15745
Fix/Workaround:
To get rid of this offset, reconfigure the AC0 input muxes to use internal inputs instead of AC0 pin 0(default reset value) before starting any touch-related measurements
(COMPCTRL0.MUXPOS=4,COMPCTRL0.MUXNEG=5).

49.2.2 DFLL48M

1 – The DFLL clock must be requested before being configured otherwise a write access to a DFLL register can freeze the device.
Errata reference: 9905
Fix/Workaround:
Write a zero to the DFLL ONDEMAND bit in the DFLLCTRL register before configuring the DFLL module.

2 – The DFLL status bits in the STATUS register during the USB clock recovery mode can be wrong after a USB suspend state.
Errata reference: 16193
Fix/Workaround:
Do not monitor the DFLL status bits in the STATUS register during the USB clock recovery mode.

3 – If the DFLL48M reaches the maximum or minimum COARSE or FINE calibration values during the locking sequence, an out of bounds interrupt will be generated. These interrupts will be generated even if the final calibration values at DFLL48M lock are not at maximum or minimum, and might therefore be false out of bounds interrupts.
Errata reference: 16192
Fix/Workaround:
Check that the lockbits: DFLLLCKC and DFLLLCKF in the OSCCTRL Interrupt Flag Status and Clear register (INTFLAG) are both set before enabling the DFLLOOB interrupt.

49.2.3 FDPLL

1 – When changing on-the-fly the FDPLL ratio in DPLLnRATIO register, STATUS.DPLLnLDRTO will not be set when the ratio update will be completed.
Errata reference: 15753
Fix/Workaround:
Wait for the interruption flag INTFLAG.DPLLnLDRTO instead.

49.2.4 PORT

1 – PORT read/write attempts on non-implemented registers, including addresses beyond the last implemented register group (PA, PB,...) do not generate a PAC protection error.
Errata reference: 15611
Fix/Workaround:
None

49.2.5 SUPC

1 – When Buck converter is set as main voltage regulator (SUPC.VREG.SEL=1), the microcontroller can freeze when leaving standby mode.
Errata reference: 15264
Fix/Workaround:
Enable the main voltage regulator in standby mode (SUPC.VREG.RUNSTDBY=1) and set the Standby in PL0 bit to one (SUPC.VREG.STDBYPL0=1).

49.2.6 Device

1 – Writing PM.STDBYCFG.VREGSMOD to one does not set the main voltage regulator in standby mode, the low power regulator is still used in standby mode.
Errata reference: 14543
Fix/Workaround :
Set SUPC.VREG.RUNSTDBY to one.

2 – When configured in HS or FastMode+, SDA and SCL fall times are shorter than I2C specification requirement and can lead to reflection.
Errata reference: 16225
Fix/Workaround:
When reflection is observed a 100 ohms serial resistor can be added on the impacted line.

3 – SAML22G devices delivered before date code 1716 only have 3 sercoms available (0,1,2) instead of 4 (0,1,2,3)
Errata reference: 15511
Fix/Workaround:
None.

49.2.7 RTC

1 – When the tamper controller is configured for CAPTURE while the RTC is disabled, a noisy pin can trigger the following once the RTC is enabled:

- the timestamp capture
- the tamper interrupt if enabled
- the DMA trigger if enabled.
Errata reference: 15092
Fix/Workaround:
(1) Set the tamper interrupt enable only when the RTC is enabled:
- Clear the tamper interrupt flags & ID registers (INTFLAG.TAMPER & TAMPID.TAMPIDx registers).
- Enable RTC (CTRLA.ENABLE = 1).
- Enable the tamper interrupt (INTEN.TAMPER = 1)
To disable the RTC, first disable the Tamper interrupts before disabling the RTC.
- Disable Tamper interrupts (INTEN.TAMPER=0)
- Disable the RTC (CTRLA.ENABLE=0).
(2) Issue a CPU read of the TIMESTAMP register immediately after the RTC is enabled. This releases the register lock allowing the capture of the next (and valid) tamper. This releases the DMA trigger of the erroneous capture tamper.

2 – When the tamper input 4 action is configured for active layer mode (TAMPCTRL.IN4ACT=3), the RTC tamper interrupt occurs before the TIMESTAMP register is updated.
Errata reference: 14819
Fix/Workaround:
Two workarounds are available.
- Use the DMA trigger to determine when the TIMESTAMP value is registered. The DMA trigger sets after the TIMESTAMP register update. Refer Errata reference 14820.
- Implement a wait loop to create a delay, when the tamper interrupt handler routine begins, to when the TIMESTAMP register is read. The delay must be long enough to wait for 3x CLK_RTC period.
Example:
* If CLK_RTC frequency is 1KHz, the delay must be at least 3ms.
* If CLK_RTC frequency is 32KHz, the delay must be at least 92us

3 – If PB01 is multiplexed to RTC peripheral (RTC/IN2), the system will always see this input pin as logic ‘0’ when backup mode is entered. If the detection transition (TAMPCTRL.TAMLVL2 = 0), it might falsely wake up the system.
Errata reference: 15010
Fix/Workaround:
If the system is expected to enter backup mode, use other tamper pins (IN0/IN1/IN3/IN4) for tamper detection.

4 – When the tamper input 4 action is configured for active layer mode (TAMPCTRL.IN4ACT=3), the mismatch signal used to qualify the DMA and interrupt triggers produces different results. The DMA implements a level-detection whereas the interrupt implements an edge-detection. The result is that the DMA may trigger frequently from the same mismatch compared to the interrupt which will only trigger once.
Errata reference: 14820
Fix/Workaround:

Following three workarounds are available
1) Tamper inputs 0, 1, 2 and 3 can be configured for active layer with DMA.
2) Tamper input 4 can be configured with DMA for any mode other than active layer.
3) If Tamper input 4 is to be used in active layer, do not enable the DMA, to prevent performance degradation.

49.2.8 TRNG

1 – When TRNG is enabled with configuration CTRL.RUNSTDBY = 0, (disabled during sleep) it could still continue to operate resulting in over-consumption (~50uA) in standby mode.
Errata reference: 14827
Fix/Workaround:
Disable the TRNG before entering standby mode.

49.2.9 EVSYS

1 – The acknowledge between an event user and the EVSYS clears the CHSTATUS.CHBUSYn bit before this information is fully propagated in the EVSYS one GCLK_EVSYS_CHANNEL_n clock cycle later. As a consequence, any generator event occurring on that channel before that extra GCLK_EVSYS_CHANNEL_n clock cycle will trigger the overrun flag.
Errata reference: 14835
Fix/Workaround:
For applications using event generators other than the software event, monitor the OVR flag.
For applications using the software event generator, wait one GCLK_EVSYS_CHANNEL_n clock cycle after the CHSTATUS.CHBUSYn bit is cleared before issuing a software event.

2 – Using synchronous, spurious overrun can appear with generic clock for the channel always on.
Errata reference: 14532
Fix/Workaround:
- Request the generic clock on demand by setting the CHANNEL.ONDEMAND bit to one.
- No penalty is introduced.

49.2.10 DMAC

1 – When using more than one DMA channel and if one of these DMA channels has a linked descriptor, a fetch error can appear on this channel.
Errata reference: 15670
Fix/Workaround:
Do not use linked descriptors, make a software link instead:
Replace the channel which used linked descriptor by two channels DMA (with linked descriptor disabled) handled by two channels event system:
- DMA channel 0 transfer completion is able to send a conditional event for DMA channel 1 (via event system with configuration of BTCTRL.EVOSEL=BLOCK for channel 0 and configuration CHCTRLB.EVACT=CBLOCK for channel 1).

- On the transfer complete reception of the DMA channel 0, immediately re-enable the channel 0.
- Then DMA channel 1 transfer completion is able to send a conditional event for DMA channel 0 (via event system with configuration of BTCTRL.EVOSEL=BLOCK for channel 1 and configuration CHCTRLB.EVACT=CBLOCK for channel 0).
- On the transfer complete reception of the DMA channel 1, immediately re-enable the channel 1.
- The mechanism can be launched by sending a software event on the DMA channel 0.

2 – When at least one channel using linked descriptors is already active, enabling another DMA channel (with or without linked descriptors) can result in a channel Fetch Error (FERR) or an incorrect descriptor fetch.
This happens if the channel number of the channel being enabled is lower than the channel already active.
Errata reference: 15683
Fix/Workaround:
When enabling a DMA channel while other channels using linked descriptors are already active, the channel number of the new channel enabled must be greater than the other channel numbers.

49.2.11 EIC

1 – Changing the NMI configuration(CONFIGn.SENSEx)on the fly may lead to a false NMI interrupt.
Errata reference: 15279
Fix/Workaround:
Clear the NMIFLAG bit once the NMI has been modified.

2 – When the asynchronous edge detection is enabled, and the system is in standby mode, only the first edge will generate an event. The following edges won't generate events until the system wakes up.
Errata reference: 16103
Fix/Workaround:
Asynchronous edge detection doesn't work, instead use the synchronous edge detection (ASYNCH.ASYNCH[x]=0). In order to reduce power consumption when using synchronous edge detection, either set the GCLK_EIC frequency as low as possible or select the ULP32K clock (EIC CTRLA.CKSEL=1).

3 – Access to EIC_ASYNCH register in 8/16-bit mode is not functional.
*** Writing in 8-bit mode will also write this byte in all bytes of the 32-bit word.**
*** Writing higher 16-bits will also write the lower 16-bits.**
*** Writing lower 16-bits will also write the higher 16-bits.**
Errata reference: 14417
Fix/Workaround:
Two workarounds are available.
- Use 32-bit write mode.
- Write only lower 16-bits (This will write upper 16-bits also, but does not impact the application).

4 – When the EIC is configured to generate an interrupt on a low level or rising edge or both edges (CONFIGn.SENSEx) with the filter enabled (CONFIGn.FILTENx), a spurious flag might appear for the dedicated pin on the INTFLAG.EXTINT[x] register as soon as the EIC is enabled using CTRLA ENABLE bit.
Errata reference: 15278
Fix/Workaround:
Clear the INTFLAG bit once the EIC enabled and before enabling the interrupts.

49.2.12 SERCOM

1 – In USART autobaud mode, missing stop bits are not recognized as inconsistent sync (ISF) or framing (FERR) errors.
Errata reference: 13852
Fix/Workaround:
None

2 – In ISO7816 T0 mode when start of frame detect is enabled(CTRLB.SFDE=1), if there is a parity error, receive start (INTFLAG.RXS) can be erroneously set. This is because the transmitted parity low is also seen by the receiver and looks like a start of frame.
Errata reference: 14694
Fix/Workaround:
Clear INTFLAG.RXS when received on parity error.

49.2.13 TC

1 – When clearing STATUS.xxBUFV flag, SYNCBUSY is released before the register is restored to its appropriate value.
Errata reference: 15056
Fix/Workaround:
To ensure that the register value is properly restored before updating this same register through xx or xxBuf with a new value, the STATUS.xxBUFV flag must be cleared successively two times.

49.2.14 TCC

1 – Using TCC in dithering mode with external retrigger events can lead to unexpected stretch of right aligned pulses, or shrink of left aligned pulses.
Errata reference: 15625
Fix/Workaround:
Do not use retrigger events/actions when TCC is configured in dithering mode.

2 – When clearing STATUS.xxBUFV flag, SYNCBUSY is released before the register is restored to its appropriate value.
Errata reference: 15057
Fix/Workaround:
To ensure that the register value is properly restored before updating this same register through xx or xxBUF with a new value, the STATUS.xxBUFV flag must be cleared successively two times.

3 – Advance capture mode (CAPTMIN CAPTMAX LOCMIN LOCMAX DERIV0) doesn't work if an upper channel is not in one of these mode. Example: when CC[0]=CAPTMIN, CC[1]=CAPTMAX, CC[2]=CAPTEN, and CC[3]=CAPTEN, CAPTMIN and CAPTMAX won't work.
Errata reference: 14817
Fix/Workaround:
Basic capture mode must be set in lower channel and advance capture mode in upper channel.
Example: CC[0]=CAPTEN , CC[1]=CAPTEN , CC[2]=CAPTMIN, CC[3]=CAPTMAX
All capture will be done as expected.

49.2.15 ADC

1 – In standby sleep mode when the ADC is in free-running mode (CTRLC.FREERUN=1) and the RUNSTDBY bit is set to 0 (CTRLA.RUNSTDBY=0), the ADC keeps requesting its generic clock.
Errata reference: 15463
Fix/Workaround:
Stop the free-running mode (CTRLC.FREERUN=0) before entering standby sleep mode

2 – ADC SYNCBUSY.SWTRIG get stuck to one after wakeup from standby sleep mode.
Errata reference: 16027
Fix/Workaround:
Ignore ADC SYNCBUSY.SWTRIG status when waking up from standby sleep mode. ADC result can be read after INTFLAG.RESRDY is set. To start the next conversion, write a '1' to SWTRIG.START.

3 – The LSB of ADC result is stuck at zero, in unipolar mode for 8-bit and 10-bit resolution.
Errata reference: 14431
Fix/Workaround:
Use 12-bit resolution and take only least 8 bits or 10 bits, if necessary.

4 – If a synchronized event is received during an ADC conversion, the ADC will not acknowledge the event, causing a stall of the event channel.
Errata reference: 14795
Fix/Workaround:
When using events with the ADC, only the asynchronous path from the Event System must be used.

50. Conventions

50.1 Numerical Notation

Table 50-1. Numerical Notation

Symbol	Description
165	Decimal number
0b0101	Binary number (example 0b0101 = 5 decimal)
'0101'	Binary numbers are given without prefix if unambiguous.
0x3B24	Hexadecimal number
X	Represents an unknown or don't care value
Z	Represents a high-impedance (floating) state for either a signal or a bus

50.2 Memory Size and Type

Table 50-2. Memory Size and Bit Rate

Symbol	Description
KB (kbyte)	kilobyte (2^{10} = 1024)
MB (Mbyte)	megabyte (2^{20} = 1024*1024)
GB (Gbyte)	gigabyte (2^{30} = 1024*1024*1024)
b	bit (binary '0' or '1')
B	byte (8 bits)
1kbit/s	1,000 bit/s rate (not 1,024 bit/s)
1Mbit/s	1,000,000 bit/s rate
1Gbit/s	1,000,000,000 bit/s rate
word	32 bit
half-word	16 bit

50.3 Frequency and Time

Symbol	Description
kHz	1kHz = 10^3Hz = 1,000Hz
KHz	1KHz = 1,024Hz, 32KHz = 32,768Hz
MHz	10^6 = 1,000,000Hz

Symbol	Description
GHz	10^9 = 1,000,000,000Hz
s	second
ms	millisecond
µs	microsecond
ns	nanosecond

50.4 Registers and Bits

Table 50-3. Register and Bit Mnemonics

Symbol	Description
R/W	Read/Write accessible register bit. The user can read from and write to this bit.
R	Read-only accessible register bit. The user can only read this bit. Writes will be ignored.
W	Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value.
BIT	Bit names are shown in uppercase. (Example ENABLE)
FIELD[n:m]	A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0}
Reserved	Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read. Reserved bit field values must not be written to a bit field. A reserved value won't be read from a read-only bit field.
PERIPHERAL*i*	If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance.
Reset	Value of a register after a power Reset. This is also the value of registers in a peripheral after performing a software Reset of the peripheral, except for the Debug Control registers.
SET/CLR	Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a one to a bit in the CLR register will clear the corresponding bit in both registers, while writing a one to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence.

51. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

Table 51-1. Acronyms and Abbreviations

Abbreviation	Description
AC	Analog Comparator
ADC	Analog-to-Digital Converter
ADDR	Address
AES	Advanced Encryption Standard
AHB	AMBA Advanced High-performance Bus
AMBA®	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus
AREF	Analog reference voltage
BLB	Boot Lock Bit
BOD	Brown-out detector
CAL	Calibration
CC	Compare/Capture
CCL	Configurable Custom Logic
CLK	Clock
CRC	Cyclic Redundancy Check
CTRL	Control
DAP	Debug Access Port
DFLL	Digital Frequency Locked Loop
DMAC	DMA (Direct Memory Access) Controller
DSU	Device Service Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EIC	External Interrupt Controller
EVSYS	Event System
FREQM	Frequency Meter
GCLK	Generic Clock Controller
GND	Ground
GPIO	General Purpose Input/Output
I^2C	Inter-Integrated Circuit
IF	Interrupt flag

Abbreviation	Description
INT	Interrupt
MBIST	Memory built-in self-test
MEM-AP	Memory Access Port
MTB	Micro Trace Buffer
NMI	Non-maskable interrupt
NVIC	Nested Vector Interrupt Controller
NVM	Non-Volatile Memory
NVMCTRL	Non-Volatile Memory Controller
OSC	Oscillator
PAC	Peripheral Access Controller
PC	Program Counter
PER	Period
PM	Power Manager
POR	Power-on reset
PORT	I/O Pin Controller
PTC	Peripheral Touch Controller
PWM	Pulse Width Modulation
RAM	Random-Access Memory
REF	Reference
RTC	Real-Time Counter
RX	Receiver/Receive
SERCOM	Serial Communication Interface
SLCD	Segmented Liquid Crystal Display
SMBus™	System Management Bus
SP	Stack Pointer
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SUPC	Supply Controller
SWD	Serial Wire Debug
TC	Timer/Counter
TCC	Timer/Counter for Control Applications
TRNG	True Random Number Generator

Abbreviation	Description
TX	Transmitter/Transmit
ULP	Ultra-low power
USART	Universal Synchronous and Asynchronous Serial Receiver and Transmitter
USB	Universal Serial Bus
V_{DD}	Common voltage to be applied to VDDIO and VDDANA
V_{DDIO}	Digital supply voltage
V_{DDANA}	Analog supply voltage
VREF	Voltage reference
WDT	Watchdog Timer
XOSC	Crystal Oscillator

52. Datasheet Revision History

Note: The datasheet revision is independent of the die revision (Revision bit in the Device Identification register of the Device Service Unit, DSU.DID.REVISION) and the device variant (last letter of the ordering number).

52.1 Rev.A - 03/2017

Section	Changes
Document	• Change of document style. • New Microchip document number. Previous version was Atmel document 42402 rev.E.
Device	• Package UFBGA100 added. • Colors and legends in pinout diagrams updated. • Editorial updates.
PAC - Peripheral Access Controller	• Editorial updates.
FREQM – Frequency Meter	• Editorial updates.
SUPC – Supply Controller	• Register BOD12 is PAC Write-Protected and Write-Synchronized. • Editorial updates.
WDT – Watchdog Timer	• Register CONFIG is not Enable-protected.
RTC – Real-Time Counter	• Functional description of General Purpose registers added. • Bit field position updated: CTRLB.ACTF, CTRLB.DEBF, SYNCBUSY.GPn. • Editorial updates.
DMAC – Direct Memory Access Controller	• STEPSIZE factor in SRAM register BTCTRL is based on beat size in bytes. • Editorial updates.
NVMCTRL – Non-Volatile Memory Controller	• 16 region locks available.
PORT - I/O Pin Controller	• Notes added: PINCFG and PMUX registers are repeated per PORT group with offset 0x80. • Bit PORT.CTRL.SAMPLING is write-only.
SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter	• Editorial updates.
TC – Timer/Counter	• Register representation re-organized.

Section	Changes
	• CTRLA.SWRST is write-only • STATUS register is Read-Synchronized. • Editorial updates.
TCC – Timer/Counter for Control Applications	• Registers PER, PATT are Write-Synchronized. • Bit fields in PATBYF register are 8 bit in size. • Bit field TCC.DBGCTRL.FDDBD: Default '0' is OCD fault protection disabled.
TRNG – True Random Number Generator	• Bit field DATA.DATA is R/W.
AES – Advanced Encryption Standard	• Register DBGCTRL is not enable-protected. • Editorial updates.
USB – Universal Serial Bus	• Device STATUS register has reset value 0x40. • Editorial updates.
CCL – Configurable Custom Logic	• Editorial updates.
PTC - Peripheral Touch Controller	• Editorial updates.
Electrical Characteristics	• FDPLL96M: Maximum F_{OUT} 96MHz in PL2, and 48MHz in PL0.
Schematic Checklist	• Leave VLCD pin unconnected when the LCD controller is not used. • Editorial updates.
Errata	New Errata: • **SERCOM**: In USART autobaud mode, missing stop bits are not recognized as inconsistent sync (ISF) or framing (FERR) errors. *Errata reference 13852* • **EIC**: Changing the NMI configuration (CONFIGn.SENSEx) on the fly may lead to a false NMI interrupt. *Errata reference 15279* • **ADC**: Clock request not going low in standby mode. *Errata reference 15463* • **PORT**: PORT read/write attempts on non-implemented registers, including addresses beyond the last implemented register group (PA, PB, …) do not generate a PAC protection error. *Errata reference 15611* • **TCC**: Using dithering mode with external retrigger events can lead to unexpected stretch of right aligned pulses, or shrink of left aligned pulses. *Errata reference 15625*

Section	Changes
	• **DMAC**: When using more than one DMA channel and one of them is w/linked descriptor, a fetch error can appear on this channel. *Errata reference 15670* • **DMAC**: When at least one channel w/linked descriptors is already active, and the channel number of the channel being enabled is lower than the channel already active, enabling another DMA channel can result in a channel Fetch Error (FERR) or an incorrect descriptor fetch. *Errata reference 15683* • **FDPLL**: When changing on-the-fly the FDPLL ratio in DPLLnRATIO register, STATUS.DPLLnLDRTO will not be set when the ratio update will be completed. *Errata reference 15753*

52.2 Rev.E - 07/2016

Section	Changes
PM – Power Manager	• Editorial updates.
RTC – Real-Time Counter	• Editorial updates.
DMAC – Direct Memory Access Controller	• Editorial updates.
EIC – External Interrupt Controller	• EIC.CTRLA.SWRST is write-only. • Editorial updates.
ADC – Analog-to-Digital Converter	• PRESCALER bit field is in CTRLB register. • Editorial updates.
AC – Analog Comparators	• Editorial updates.
Electrical Characteristics	• Absolute Maximum Ratings: V_{pin} increased. • Section 'Injection Current' added.
Schematic Checklist	Editorial updates.

52.3 Rev.D - 05/2016

Section	Changes
Device	WLCSP49 Package added.
Device, SUPC – Supply Controller, ADC – Analog-to-Digital Converter	SUPC sub-unit DETREF generates reference voltages INTREF and BANDGAP.

Section	Changes
NVM User Row Mapping	• Updated. • Factory settings added.
DSU - Device Service Unit	• MBIST not available when device is operated from external address range and device is protected • MBIST run time depending on frequency and number of tested bytes. • Bit field description ADDR.ADDR updated. • Bit field name DID.DEVSEL updated.
GCLK - Generic Clock Controller	Available signals are GCLK_IO[7:0].
FREQM – Frequency Meter	Wakeup from Idle sleep mode.
SUPC – Supply Controller	Editorial updates.
RTC – Real-Time Counter	• Two General Purpose (GPn) registers and eight Backup (BKUPn) registers available. • ALARM[3:1] registers at different offsets. • MASK[3:0] registers added.
DMAC – Direct Memory Access Controller	Editorial updates.
NVMCTRL – Non-Volatile Memory Controller	Editorial updates.
SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter	Editorial updates.
TC – Timer/Counter	• Registers PERBUF, CCBUFx are write-synchronized. • Editorial updates.
TCC – Timer/Counter for Control Applications	• In CAPTMIN mode, value 0 can be captured only in down-counting mode. • In Counter Operation: Section 'Stop Command and Event Action' split into 'Stop Command' and 'Pause Event Action'. • RAMP2C Operation: figures added. • COUNT register to be read-synchronized by user. • Register presentation updated. • Editorial updates.
TRNG – True Random Number Generator	Block diagram updated.
AES – Advanced Encryption Standard	Register address offsets corrected.
USB – Universal Serial Bus	CTRLB.SPDCONF values updated.
ADC – Analog-to-Digital Converter	Editorial updates.

Section	Changes
SLCD - Segment Liquid Crystal Display Controller	Editorial updates.
Electrical Characteristics	• Absolute Maximum Ratings: ESD caution note added. • General Operating Ratings: Voltage drop caution note added. • Power Consumption: RTC clock source in BACKUP is OSCULP32K. • APWS: Editorial updates. • ADC table 'Operating Conditions': Typical value for T_S added. • ADC table 'Differential Mode': Max. values for INL, DNL improved. • ADC table 'Single-Ended Mode': Max. values for DNL improved. • Analog Comparator Characteristics: change 'Conditions' to explicit register values. • DFLL48M, Table 'Closed Loop Characteristics': – Conditions for $F_{CloseOUT}$, $F_{CloseJitter}$, and T_{Lock} specified further. – Values for $F_{CloseOUT}$ updated. • DPLL Characteristics: Period Jitter values added for different conditions. • OSC16M Characteristics, Table 'Multi RC Oscillator Electrical Characteristics': symbols TempCo, SupplyCo replaced by TempDrift, SupplyDrift. • NVM Characteristics – Table added: "Flash Erase and Programming Current" – t_{FSE} removed
Typical Characteristics	Section added.
Schematic Checklist	• Power Supply Connections: Recommendation for capacitance on VBAT removed - depends on customer application. • Editorial updates.
Errata	New Errata: • **SUPC**: Buck supply in Standby mode. *Errata reference 15264* • **EIC**: Spurious flag at enable possible. *Errata reference 15278*

Section	Changes
	• **Device**: Number of SERCOM instances increased for SAM L22G. *Errata reference 15511*

52.4 Rev.C - 01/2016

Section	Changes
Device, SUPC – Supply Controller, ADC – Analog-to-Digital Converter	Temperature sensor functionality removed.
DSU - Device Service Unit	Bit CTRL.CRC is write-only.
DMAC – Direct Memory Access Controller	• Bit field descriptions for PRICTRL0.LVLPRIn updated. • Editorial updates.
PORT - I/O Pin Controller	Editorial updates.
OSCCTRL – Oscillators Controller	• Register DFLLCTRL: bit field descriptions added for BPLCKC and WAITLOCK.
EVSYS – Event System	Editorial updates.
TC – Timer/Counter	Editorial updates.
AES – Advanced Encryption Standard	• Correct name of register INTENFLAG is INTFLAG.
SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter	• Start-of-Frame support added. • Recommended max. Rx Error explanation added to 'Asynchronous Operational Range'.
SLCD - Segment Liquid Crystal Display Controller	Editorial updates.
PTC - Peripheral Touch Controller	Editorial updates.
Electrical Characteristics: General	• V_{DDIO} and V_{DDANA} are referred to as V_{DD}. • GPIO Clusters moved to GPIO Pin Clusters. • Power Consumption values moved to individual sections, tables "Power Consumption." • Differentiation between values covered by production tests, characterization, and simulation introduced. • Editorial updates. • Section structure flattened.
Electrical Characteristics: General Operating Ratings	• T_J is 100°C max. • Note added on V_{DD}.

Section	Changes
Electrical Characteristics: Supply Characteristics	• Supply Rise Rates replaced and updated by Supply Slew Rates. • Current Requirements removed.
Electrical Characteristics: Power Consumption	• Active Current Consumption re-organized. • Standby and Backup values added.
Electrical Characteristics: Wake-up Timing	• Values updated. • Standby and Backup values added.
Electrical Characteristics: IO Pin Characteristics	• Tables re-structured. • Values updated and added.
Electrical Characteristics, Analog Characteristics: Buck Converter, LDO Regulator, AWPS, POR, BOD33, ADC, AC, Voltage Reference	• Values updated and added.
Electrical Characteristics: NVM Characteristics	• T_{FCE} removed.
Electrical Characteristics, Oscillators Characteristics: XOSC, XOSC32K,OSCULP32K, OSC16M, DFLL48M, DPLL96M	• Values added.
Electrical Characteristics: SLCD Characteristics	• Condition for minimizing power consumption added. • Values added.
Schematic Checklist	• 1kΩ pull-up resistor for SWCLK pin recommended. • Editorial updates.
Errata	Changed: • Erratum 15010 moved from Device to RTC category. • Errata 14417, 14431, 14819, 14827, 15010: editorial updates. New: • **TCC**: Advance capture mode requires upper channels to be configured. *Errata reference 14817* • **TC**: SYNCBUSY.CC and .PER can be released too early. *Errata reference 15056* • **TCC**: SYNCBUSY.CCB and .PERB can be released too early. *Errata reference 15057* • **RTC**: Tamper pin activity while RTC is disabled can erroneously set tamper detection when enabled. *Errata reference 15092*

52.5 Rev.B - 11/2015

Section	Changes
Description	Updated to 2.46 Coremark/MHz
NVM Software Calibration Area Mapping	Temperature calibration content removed.
Processor and Architecture	Editorial update.
Oscillator Pinout	Recommendation for XOSC32 jitter optimization added.
Power Supply and Start-Up Considerations	• VLCD has two alternative functions. • Section "Performance Level Overview" added.
Processor and Architecture	QoS levels can be written/read using 32-bit accesss only.
DSU - Device Service Unit	• Register PID1.JEPIDCL reset value 0xF. • Editorial updates.
GCLK - Generic Clock Controller	Editorial updates.
RSTC – Reset Controller	Editorial updates.
PM – Power Manager	• STDBYCFG.BBIASHS=0x0 turns off Back Biasing in Standby mode. • Added Power Reset as option for exiting OFF mode.
OSCCTRL – Oscillators Controller	DFLL48M: • Only reference clock drift will make run the compensation out of bounds - voltage and temperature swings are compensated for. • USB Clock recovery mode enabled by writing to DFLLCTRL.USBCRM and DFLLCTRL.MODE. • STATUS.DFLLRDY indicates readiness of the DFLL48M registers for read/write access. • Editorial updates.
OSC32KCTRL – 32KHz Oscillators Controller	XOSC32K.STARTUP times corrected.
SUPC – Supply Controller	Register bit VREF.TSSEL removed.
RTC – Real-Time Counter	• 16-bit counter: TIMESTAMP.COUT has reset value 0x0000. • Frequency correction sign affecting periods, not frequencies. • Editorial updates.
DMAC – Direct Memory Access Controller	• CHCTRLA.RUNSTDBY supported

Section	Changes
	• Editorial updates.
NVMCTRL – Non-Volatile Memory Controller	Editorial updates.
SERCOM – Serial Communication Interface	Editorial updates.
SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter	• Register RXPL.RXPL: formula corrected. • Register CTRLA.TXPO: pin configuration updated.
SERCOM I2C – SERCOM Inter-Integrated Circuit	Master Operation: Startup timing explained.
TCC – Timer/Counter for Control Applications	Register DRVCTRL.NRVx size corrected.
USB – Universal Serial Bus	• Register PCKSIZE.BYTE_COUNT size corrected. • Register HSOFC.FLENCE description updated.
CCL – Configurable Custom Logic	Editorial updates.
ADC – Analog-to-Digital Converter	• Features: Number of analog inputs updated. • Section on Device Temperature Measurement removed: Only one temperature sensor available. • Signal Description: – Updated signal name from ADC to AIN. – AREFA/B naming updated to VREFA/B. • Editorial updates.
SLCD - Segment Liquid Crystal Display Controller	• Contrast/VLCD values moved to Electrical Characteristics. • Frame rate calculation using NB_COM. • No Power Mode of LCD Power Supply. • Dead Time Insertion removed. • Editorial updates.
Electrical Characteristics	LCD parameters expanded.
Errata	New Errata: • **TRNG**: Power consumption in standby mode. *Errata reference 14827* • **EVSYS**: Under certain conditions, CHSTATUS.CHBUSYn is cleared too early causing an overrun flag. *Errata reference 14835* • **Device**: Some tampering pins may not work in backup mode. *Errata reference 15010* Removed Erratum 14815.

52.6 Rev A - 08/2015

Initial revision.

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC16F18313, PIC16LF18313, PIC16F18323, PIC16LF18323	
Tape and Reel Option:	Blank	= Standard packaging (tube or tray)
	T	= Tape and Reel(1)
Temperature Range:	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package:(2)	JQ	= UQFN
	P	= PDIP
	ST	= TSSOP
	SL	= SOIC-14
	SN	= SOIC-8
	RF	= UDFN
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

- PIC16LF18313- I/P Industrial temperature, PDIP package
- PIC16F18313- E/SS Extended temperature, SSOP package

Note:

1. Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
2. Small form-factor packaging options may be available. Please check http://www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

© 2017 Microchip Technology Inc.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN:

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	**Asia Pacific Office**	**China - Xiamen**	**Austria - Wels**
2355 West Chandler Blvd.	Suites 3707-14, 37th Floor	Tel: 86-592-2388138	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	Tower 6, The Gateway	Fax: 86-592-2388130	Fax: 43-7242-2244-393
Tel: 480-792-7200	Harbour City, Kowloon	**China - Zhuhai**	**Denmark - Copenhagen**
Fax: 480-792-7277	**Hong Kong**	Tel: 86-756-3210040	Tel: 45-4450-2828
Technical Support:	Tel: 852-2943-5100	Fax: 86-756-3210049	Fax: 45-4485-2829
http://www.microchip.com/	Fax: 852-2401-3431	**India - Bangalore**	**Finland - Espoo**
support	**Australia - Sydney**	Tel: 91-80-3090-4444	Tel: 358-9-4520-820
Web Address:	Tel: 61-2-9868-6733	Fax: 91-80-3090-4123	**France - Paris**
www.microchip.com	Fax: 61-2-9868-6755	**India - New Delhi**	Tel: 33-1-69-53-63-20
Atlanta	**China - Beijing**	Tel: 91-11-4160-8631	Fax: 33-1-69-30-90-79
Duluth, GA	Tel: 86-10-8569-7000	Fax: 91-11-4160-8632	**France - Saint Cloud**
Tel: 678-957-9614	Fax: 86-10-8528-2104	**India - Pune**	Tel: 33-1-30-60-70-00
Fax: 678-957-1455	**China - Chengdu**	Tel: 91-20-3019-1500	**Germany - Garching**
Austin, TX	Tel: 86-28-8665-5511	**Japan - Osaka**	Tel: 49-8931-9700
Tel: 512-257-3370	Fax: 86-28-8665-7889	Tel: 81-6-6152-7160	**Germany - Haan**
Boston	**China - Chongqing**	Fax: 81-6-6152-9310	Tel: 49-2129-3766400
Westborough, MA	Tel: 86-23-8980-9588	**Japan - Tokyo**	**Germany - Heilbronn**
Tel: 774-760-0087	Fax: 86-23-8980-9500	Tel: 81-3-6880- 3770	Tel: 49-7131-67-3636
Fax: 774-760-0088	**China - Dongguan**	Fax: 81-3-6880-3771	**Germany - Karlsruhe**
Chicago	Tel: 86-769-8702-9880	**Korea - Daegu**	Tel: 49-721-625370
Itasca, IL	**China - Guangzhou**	Tel: 82-53-744-4301	**Germany - Munich**
Tel: 630-285-0071	Tel: 86-20-8755-8029	Fax: 82-53-744-4302	Tel: 49-89-627-144-0
Fax: 630-285-0075	**China - Hangzhou**	**Korea - Seoul**	Fax: 49-89-627-144-44
Dallas	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	**Germany - Rosenheim**
Addison, TX	Fax: 86-571-8792-8116	Fax: 82-2-558-5932 or	Tel: 49-8031-354-560
Tel: 972-818-7423	**China - Hong Kong SAR**	82-2-558-5934	**Israel - Ra'anana**
Fax: 972-818-2924	Tel: 852-2943-5100	**Malaysia - Kuala Lumpur**	Tel: 972-9-744-7705
Detroit	Fax: 852-2401-3431	Tel: 60-3-6201-9857	**Italy - Milan**
Novi, MI	**China - Nanjing**	Fax: 60-3-6201-9859	Tel: 39-0331-742611
Tel: 248-848-4000	Tel: 86-25-8473-2460	**Malaysia - Penang**	Fax: 39-0331-466781
Houston, TX	Fax: 86-25-8473-2470	Tel: 60-4-227-8870	**Italy - Padova**
Tel: 281-894-5983	**China - Qingdao**	Fax: 60-4-227-4068	Tel: 39-049-7625286
Indianapolis	Tel: 86-532-8502-7355	**Philippines - Manila**	**Netherlands - Drunen**
Noblesville, IN	Fax: 86-532-8502-7205	Tel: 63-2-634-9065	Tel: 31-416-690399
Tel: 317-773-8323	**China - Shanghai**	Fax: 63-2-634-9069	Fax: 31-416-690340
Fax: 317-773-5453	Tel: 86-21-3326-8000	**Singapore**	**Norway - Trondheim**
Tel: 317-536-2380	Fax: 86-21-3326-8021	Tel: 65-6334-8870	Tel: 47-7289-7561
Los Angeles	**China - Shenyang**	Fax: 65-6334-8850	**Poland - Warsaw**
Mission Viejo, CA	Tel: 86-24-2334-2829	**Taiwan - Hsin Chu**	Tel: 48-22-3325737
Tel: 949-462-9523	Fax: 86-24-2334-2393	Tel: 886-3-5778-366	**Romania - Bucharest**
Fax: 949-462-9608	**China - Shenzhen**	Fax: 886-3-5770-955	Tel: 40-21-407-87-50
Tel: 951-273-7800	Tel: 86-755-8864-2200	**Taiwan - Kaohsiung**	**Spain - Madrid**
Raleigh, NC	Fax: 86-755-8203-1760	Tel: 886-7-213-7830	Tel: 34-91-708-08-90
Tel: 919-844-7510	**China - Wuhan**	**Taiwan - Taipei**	Fax: 34-91-708-08-91
New York, NY	Tel: 86-27-5980-5300	Tel: 886-2-2508-8600	**Sweden - Gothenberg**
Tel: 631-435-6000	Fax: 86-27-5980-5118	Fax: 886-2-2508-0102	Tel: 46-31-704-60-40
San Jose, CA	**China - Xian**	**Thailand - Bangkok**	**Sweden - Stockholm**
Tel: 408-735-9110	Tel: 86-29-8833-7252	Tel: 66-2-694-1351	Tel: 46-8-5090-4654
Tel: 408-436-4270	Fax: 86-29-8833-7256	Fax: 66-2-694-1350	**UK - Wokingham**
Canada - Toronto			Tel: 44-118-921-5800
Tel: 905-695-1980			Fax: 44-118-921-5820
Fax: 905-695-2078			