MEC152x

Keyboard and Embedded Controller for Notebook PC

Operating Conditions

- Operating Voltages: 3.3 V and 1.8 V
- Operating Temperature Range: -40 °C to 85 °C

Low Power Modes

- Chip is designed to always operate in Lowest Power state during Normal Operation
- Supports all 5 ACPI Power States for PC platforms
- Supports 2 Chip-level Sleep Modes: Light Sleep and Heavy Sleep
  - Low Standby Current in Sleep Modes

ARM® Cortex-M4 Embedded Processor

- Programmable clock frequency up to 48 MHz
- Fixed point processor
- Single 4GByte Addressing Space
- Nested Vectored Interrupt Controller (NVIC)
  - Maskable Interrupt Controller
  - Maskable hardware wake up events
  - 8 Levels of priority, individually assignable by vector
- EC Interrupt Aggregator expands number of Interrupt sources supported or reduces number of vectors needed
- Complete ARM® Standard debug support
  - JTAG-Based DAP port, comprised of SWJ-DP and AHB-AP debugger access functions

Memory Components

- 256 KB Code/Data SRAM
  - 224 KB optimized for code performance
  - 32 KB optimized for data performance
- 64 Bytes Battery Powered Storage SRAM
- 4K bits OTP
  - In circuit programmable
- ROM
  - Contains Boot ROM
  - Contains Runtime APIs for built-in functions
- 2K byte Internal EEPROM (MEC1523 Only)
- 4Mbit in-chip SPI Serial Flash (MEC1527 only)
  - SST25PF040C
  - SPI Master controller
  - Supports Mode 0 and mode 3
  - 24MHz

Clocks

- 48 MHz Internal PLL
- 32 kHz Clock Sources
  - Internal 32 kHz silicon oscillator
  - External 32 kHz crystal (XTAL) source
  - External single-ended 32 kHz clock source

Package Options

- 144 pin WFBGA
- 128 pin VTQFP
- 128 pin WFBGA

Security Features

- Boot ROM Secure Boot Loader
  - Hardware Root of Trust (RoT) using Secure Boot and Immutable Code using ECDSA p-384 and SHA-384
  - Supports 2 Code Images in external SPI Flash (Primary and Fall back image)
  - Supports Full EC firmware redundancy over Two Flash components
  - Authenticates SPI Flash image before loading
  - Support AES-256 Encrypted SPI Flash images
  - Key Revocation
  - Roll back protection up to 128 Revisions
- Hardware Accelerators:
  - Multi purpose AES Crypto Engine:
    - Support for 128-bit - 256-bit key length
    - Supports Battery Authentication applications
  - Digital Signature Algorithm Support
    - Support for ECDSA and EC_KCDSA
  - Cryptographic Hash Engine
    - Support for SHA-1, SHA-256 to SHA-512
  - Public Key Crypto Engine
    - Hardware support for RSA and Elliptic Curve asymmetric public key algorithms
    - RSA keys length of 1024 to 4096 bits
    - ECC Prime Field keys up to 571 bits

© 2020 Microchip Technology Inc.

  - ECC Binary Field keys up to 571 bits
  - Microcoded support for standard public key algorithms
- OTP for storing Keys and IDs
  - Lockable on 32 B boundaries to prevent read access or write access
- True Random Number Generator
- 1 kbit FIFO
- JTAG Disabled by default

System Host interface

- Enhanced Serial Peripheral Interface (eSPI)
  - Intel eSPI Specification compliant
    - eSPI Interface Base Spec, Intel Doc. #327432-004, Rev. 1.0.MEC152x
    - eSPI Compatibility Spec, Intel Doc. #562633, Rev. 0.6
  - Support for Master Attached Flash Sharing (MAFS)
  - Support for Slave Attached Flash Sharing (SAFS)
  - Supports all four channels:
    - Peripheral Channel
    - Virtual Wires Channel
    - Out-of-Band (OOB) Tunneled Message Channel
    - Run-time Flash Access Channel
  - Supports EC Bus Master to Host Memory
  - Supports up to 66 MHz maximum operating frequency
- One Serial Peripheral Interface (SPI) Slave
  - Quad SPI (half-duplex) or Single wire (full duplex) support
  - Mode 0 and Mode3 operation
  - Programmable wait time for response delay
- System to EC Message Interface
  - Two Embedded Memory Interfaces
    - Provides Two Windows to On-Chip SRAM for Host Access
    - Two Register Mailbox Command Interface
  - Mailbox Registers Interface
    - Thirty-two 8-bit registers
    - Two Register Mailbox Command Interfaces
    - Two Register SMI Source Interfaces
  - Five ACPI Embedded Controller Interfaces
    - Four EC Interfaces
    - One Power Management Interface
- One Serial Peripheral Interface (SPI) Master Controller
  - Dual and Quad I/O Support
  - Flexible Clock Rates
  - Support for 1.8V and 3.3V slave devices
  - SPI Burst Capable
  - SPI Controller Operates with Internal DMA Controller with CRC Generation
  - Mappable to the following ports (only 1 port active at a time)
    - 1 shared SPI Interface
    - 1 Private SPI interface
    - 1 General Purpose SPI interface
    - 1 In-Chip SPI (MEC1527 Only)
- 8042 Emulated Keyboard Controller
  - 8042 Style Host Interface
  - Port 92 Legacy A20M Support
  - Fast GATEA20 & Fast CPU_RESET
- 18 x 8 Interrupt Capable Multiplexed Keyboard Scan Matrix
  - Optional Push-Pull Drive for Fast Signal Switching
- PECI Interface 3.1
  - Support Intel's low voltage PECI
- Port 80 BIOS Debug Port
  - Two Ports, Assignable to Any eSPI IO Address
  - 24-bit Timestamp with Adjustable Timebase
  - 16-Entry FIFO

Peripheral Features

- Internal DMA Controller
  - Hardware or Firmware Flow Control
  - Firmware Initiated Memory-to-Memory transfers
  - Hardware CRC-32 Generator on Channel 0
  - 12-Hardware DMA Channels support five SMBus Master/Slave Controllers and One SPI Controller
- I2C/SMBus Controllers
  - 5 I2C/SMBus controllers
  - 3 I2C only controllers without the Network layer
  - Up to 16 Configurable I2C ports
    - Full Crossbar switch allows any port to be connected to any controller
  - Supports Promiscuous mode of operation
  - Fully Operational on Standby Power
  - Multi-Master Capable
  - Supports Clock Stretching
  - Programmable Bus Speeds
  - 1 MHz Capable
  - Supports DMA Network Layer
- General Purpose I/O Pins
  - Inputs
    - Asynchronous rising and falling edge wakeup detection Interrupt High or Low Level

© 2020 Microchip Technology Inc.

  - Outputs:
    - Push Pull or Open Drain output
    - Programmable power well emulation
  - Pull up or pull down resistor control
    - Automatically disabling pull-up resistors when output driven low
    - Automatically disabling pull-down resistors when output driven high
  - Programmable drive strength
  - Two separate1.8V/3.3V configurable IO regions
  - Group or individual control of GPIO data
  - Glitch protection and Under-Voltage Protection on all GPIO pins
- Input Capture and Compare timer
  - Six 32-bit Capture Registers
  - 16Input Pins (ICTx)
    - Full Crossbar switch allows any port to be connected to any controller
  - 32-bit Free-running timer
  - Two 32-bit Compare Registers
  - Capture, Compare and Overflow Interrupts
- Universal Asynchronous Receiver Transmitter (UART)
  - Three High Speed NS16C550A Compatible UARTs with Send/Receive 16-Byte FIFOs
    - UART1 - Configurable 2-pin/4-pin/8-pin
    - UART2 - Configurable 2-pin/4-pin
    - UART3 - Configurable 2-pin/4-pin/8-pin
  - Programmable Main Power or Standby Power Functionality
  - Standard Baud Rates to 115.2 Kbps, Custom Baud Rates to 1.5 Mbps
- Programmable Timer Interface
  - Two16-bit Auto-reloading Timer Instances
    - 16 bit Pre-Scale divider
    - Halt and Reload control
    - Auto Reload
  - Two 32-bit Auto-reloading Timer Instances
    - 16 bit Pre-Scale divider
    - Halt and Reload control
    - Auto Reload
  - Three Operating Modes per Instance: Timer (Reload or Free-Running) or One-shot.
    - Event Mode is not supported
- 32-bit RTOS Timer
  - Runs Off 32kHz Clock Source
  - Continues Counting in all the Chip Sleep States regardless of Processor Sleep State
  - Counter is Halted when Embedded Controller is Halted (e.g., JTAG debugger active, break points)
  - Generates wake-capable interrupt event
- Watch Dog Timer (WDT)
- 9 Programmable Pulse Width Modulator (PWM) outputs
  - Multiple Clock Rates
  - 16-Bit ON & 16-Bit OFF Counters
- 4 Fan Tachometer Inputs
  - 16 Bit Resolution
- Breathing LED Interface
  - Three Blinking/Breathing LEDs
  - Programmable Blink Rates
  - Piecewise Linear Breathing LED Output Controller
    - Provides for programmable rise and fall waveforms
  - Operational in EC Sleep States
- PS2 Controller
  - Two PS2 controllers
  - Three PS2 ports
  - All three ports are 5 volt tolerant
- HDMI-CEC
  - One instance of HDMI Consumer Electronics Control (CEC) Bus Controller
  - Master/Slave
  - Handles signaling and encoding of the CEC data

Analog Features

- ADC Interface
  - 10-bit or 12-bit readings supported
  - ADC Conversion time 500nS/channel
  - 8 Channels
  - External voltage reference
  - Supports thermistor temperature readings
- Two Analog Comparators
  - May be used for Hardware Shutdown
    - Detection of voltage limit event
    - Detection of Thermistor Over-Temp Event

Battery Powered Peripherals

- Real Time Clock (RTC)
  - VBAT Powered
  - 32KHz Crystal Oscillator orExternal single-ended 32 kHz clock source
  - Time-of-Day and Calendar Registers
  - Programmable Alarms
  - Supports Leap Year and Daylight Savings Time
- Hibernation Timer Interface
  - Two 32.768 KHz Driven Timers
  - Programmable Wake-up from 0.5ms to 128 Minutes

© 2020 Microchip Technology Inc.

- Week Timer
  - System Power Present Input Pin
    - Week Alarm Event only generated when System Power is Available
  - Power-up Event
  - Week Alarm Interrupt with 1 Second to 8.5 Year Time-out
  - Sub-Week Alarm Interrupt with 0.50 Seconds - 72.67 hours time-out
  - 1 Second and Sub-second Interrupts
- VBAT-Powered Control Interface (VCI)
  - 4 Active-low VCI Inputs
  - 1 Active-high VCI Output Pin
  - System Power Present Detection for gating RTC wake eventsd
- Optional filter and latching
- Battery- powered General purpose Output (BGPO)

Debug Features

- 2-pin Serial Wire Debug (SWD) interface
- 4-Pin JTAG interface for Boundary Scan
- Trace FIFO Debug Port (TFDP)

© 2020 Microchip Technology Inc.

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

Table Of Contents

1.0 General Description 7
2.0 Pin Configuration 15
3.0 Device Inventory 54
4.0 Power, Clocks, and Resets 108
5.0 ARM M4 Based Embedded Controller 125
6.0 RAM and ROM 135
7.0 Internal DMA Controller 137
8.0 EC Interrupt Aggregator 153
9.0 Enhanced Serial Peripheral Interface (eSPI) 163
10.0 Chip Configuration 174
11.0 8042 Emulated Keyboard Controller 179
12.0 ACPI Embedded Controller Interface (ACPI-ECI) 194
13.0 ACPI PM1 Block 211
14.0 Embedded Memory Interface (EMI) 218
15.0 Mailbox Interface 232
16.0 UART 240
17.0 GPIO Interface 255
18.0 Watchdog Timer (WDT) 269
19.0 16/32 Bit Basic Timer 274
20.0 Input Capture and Compare Timer 281
21.0 Hibernation Timer 293
22.0 RTOS Timer 296
23.0 Real Time Clock 300
24.0 Week Timer 312
25.0 TACH 323
26.0 PWM 330
27.0 PECI Interface 335
28.0 HDMI-CEC Interface Controller 343
29.0 Analog to Digital Converter 353
30.0 Analog Comparator 365
31.0 Blinking/Breathing LED 368
32.0 Keyboard Scan Interface 385
33.0 I2C/SMBus Interface 392
34.0 I2C Controller 396
35.0 PROCHOT Monitor 413
36.0 Quad SPI Master Controller 423
37.0 Serial Peripheral Interface (SPI) Slave 442
38.0 EEPROM 466
39.0 PS/2 Interface 474
40.0 Trace FIFO Debug Port (TFDP) 482
41.0 Port 80 BIOS Debug Port 486
42.0 VBAT-Powered Control Interface 491
43.0 VBAT-Powered RAM 503
44.0 VBAT Register Bank 505
45.0 EC Subsystem Registers 509
46.0 Security Features 517
47.0 OTP Block 521
48.0 Test Mechanisms 524
49.0 Electrical Specifications 526
50.0 Timing Diagrams 536
51.0 Packaging Information 562
Appendix A: Data Sheet Revision History 566
The Microchip Web Site 567
Customer Change Notification Service 567
Customer Support 567
Product Identification System 568

© 2020 Microchip Technology Inc.

1.0 GENERAL DESCRIPTION

The MEC152x is a family of low power integrated embedded controllers designed for notebook applications. The MEC152x is a highly-configurable, mixed-signal, advanced I/O controller architecture. It contains a 32-bit ARM® Cortex-M4 processor core with closely-coupled memory for optimal code execution and data access. An internal ROM, embedded in the design, is used to store the power on/boot sequence and APIs available during run time. When VTR_CORE is applied to the device, the secure boot loader API is used to download the custom firmware image from the system's shared SPI Flash device, thereby allowing system designers to customize the device's behavior.

The MEC152x device is directly powered by a minimum of two separate suspend supply planes (VBAT and VTR) and senses a third runtime power plane (VCC) to provide "instant on" and system power management functions. The MEC152x has two banks of I/O pins that are able to operate at either 3.3 V or 1.8 V. Operating at 1.8V allows the MEC152x to interface with the latest platform controller hubs and will lower the overall power consumed by the device, Whereas 3.3V allows this device to be integrated into legacy platforms that require 3.3V operation.

The MEC152x host interface is the Intel® Enhanced Serial Peripheral Interface (eSPI). The eSPI Interface is a 1.8V interface that operates in single, double and quad I/O modes. The eSPI Interface supports all four eSPI channels: Peripheral Channel, Virtual Wires Channel, OOB Message Channel, and Run-time Flash Access Channel. The eSPI hardware Flash Access Channel is used by the Boot ROM to support Master Attached Flash Sharing (MAFS). In addition, the MEC152x has specially designed hardware to support Slave Attached Flash Sharing (SAFS). The eSPI SAFS Bridge imposes Region-Based Protection and Locking security feature, which limits access to certain regions of the flash to specific masters. There may be one or more masters (e.g., BIOS, ME, etc) that will access the SAF via the eSPI interface. The ARM® Cortex-M4 processor is also considered a master, which will also have its access limited to EC only regions of SPI Flash as determined by the customer firmware application.

The MEC152x secure boot loader authenticates and optionally decrypts the SPI Flash OEM boot image using the AES-256, ECDSA , SHA-384 cryptographic hardware accelerators. The MEC152x hardware accelerators support 128-bit and 256-bit AES encryption, ECDSA and EC_KCDSA signing algorithms, 1024-bits to 4096-bits RSA and Elliptic asymmetric public key algorithms, and a True Random Number Generator (TRNG). Runtime APIs are provided in the ROM for customer application code to use the cryptographic hardware. Additionally, the device offers lockable OTP storage for private keys and IDs.

The MEC152x is designed to be incorporated into low power PC architecture designs and supports ACPI sleep states (S0-S5). During normal operation, the hardware always operates in the lowest power state for a given configuration. The chip power management logic offers two low power states: light sleep and heavy sleep. These features can be used to support S0 Connected Standby state and the lower ACPI S3-S5 system sleep states. In connected standby, any eSPI command will wake the device and be processed. When the chip is sleeping, it has many wake events that can be configured to return the device to normal operation. Some examples of supported wake events are PS2 wake events, RTC, Week Alarm, Hibernation Timer, or any GPIO pin.

The MEC152x offers a software development system interface that includes a Trace FIFO Debug port, a host accessible serial debug port with a 16C550A register interface, a Port 80 BIOS Debug Port, and a 2-pin Serial Wire Debug (SWD) interface. Also included is a 4-wire JTAG interface used for Boundary Scan testing.

© 2020 Microchip Technology Inc.

1.1 Family Features

TABLE 1-1: MEC152X FEATURE LIST BY PACKAGE

Features	MEC1523 -144 WFBGA	MEC1521 144 WFBGA	MEC1527 144 WFBGA	MEC1521 128 VTQFP/ WFBGA	MEC1527 128 WFBGA
Device ID	0023_B4xxh	0023_34xxh	0023_74xxh	0023_33xxh	0023_73xxh
Boundary Scan JTAG ID	02232445h	02232445h	02232445h	02232445h	02232445h
Total SRAM Options	256KB	256KB	256KB	256KB	256KB
Code/Data Options (Primary Use)	224KB/32KB	224KB/32KB	224KB/32KB	224KB/32KB	224KB/32KB
Battery Backed SRAM	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes
Internal SPI Flash	0	0	512kB	0	512kB
EEPROM	2KB	No	No	No	No
eSPI Host Interface	Yes	Yes	Yes	Yes	Yes
eSPI SAF Interface	Yes	Yes	Yes	Yes	Yes
SPI Slave	Yes	Yes	Yes	Yes	Yes
8042 Emulated Keyboard Controller	Yes	Yes	Yes	Yes	Yes
Embedded Memory Interface (EMI)	2	2	2	2	2
Mailbox Register Interface	1	1	1	1	1
ACPI Embedded Memory Controller	4	4	4	4	4
ACPI PM1 Block Interface	1	1	1	1	1
Trace FIFO Debug Port	Yes	Yes	Yes	Yes	Yes
Internal DMA Channels	12	12	12	12	12
32-bit Basic Timer	2	2	2	2	2
16-bit Basic Timer	2	2	2	2	2
Capture Timer Reg	6	6	6	6	6
ICT Channels	16	16	16	13	13

© 2020 Microchip Technology Inc.

TABLE 1-1: MEC152X FEATURE LIST BY PACKAGE (CONTINUED)

Features	MEC1523 -144 WFBGA	MEC1521 144 WFBGA	MEC1527 144 WFBGA	MEC1521 128 VTQFP/ WFBGA	MEC1527 128 WFBGA
Compare Timer	2	2	2	2	2
Watchdog Timer (WDT)	1	1	1	1	1
Hibernation Timer	2	2	2	2	2
Week Timer	1	1	1	1	1
Sub Week Timer	1	1	1	1	1
RTC	1	1	1	1	1
RTOS Timer	1	1	1	1	1
Keyboard Matrix Scan Support	Yes	Yes	Yes	Yes	Yes
Port 80 BIOS Debug Port	2	2	2	2	2
SMBus 2.0 Host Controllers	5	5	5	5	5
I2C Host Controllers	3	3	3	3	3
SMBus/I2C Ports	16	16	16	15	15
QMSPI Controller	1 controller/ 3 ports	1 controller/ 3 ports	1 controller/ 3 ports	1 controller/ 2 ports	1 controller/ 2 ports
GP SPI port	Yes - 4 pin	Yes -4 pin	Yes -4 pin	No	No
PVT SPI port	Yes - 4 pin	Yes -4 pin	Yes -4 pin	Yes - 4 pin	Yes - 4 pin
SHD SPI port	Yes - 4 pin	Yes -4 pin	Yes -4 pin	Yes - 4 pin	Yes - 4 pin
ADC Channels	8	8	8	8	8
Vref 2 ADC	Yes	Yes	Yes	No	No
PWMs	9	9	9	9	9
TACHs	4	4	4	4	4
PECI 3.1 Interface	Yes	Yes	Yes	Yes	Yes
PS/2 Device Interface	2 controller/ 3 ports	2 controller/ 3 ports	2 controller/ 3 ports	2 controller/ 3 ports	2 controller/ 3 ports
Blinking/Breathing LED	3	3	3	3	3

TABLE 1-1: MEC152X FEATURE LIST BY PACKAGE (CONTINUED)

Features	MEC1523 -144 WFBGA	MEC1521 144 WFBGA	MEC1527 144 WFBGA	MEC1521 128 VTQFP/ WFBGA	MEC1527 128 WFBGA
UARTs	3 UART0: 2/4/8-pin UART1: 2/4/8-pin UART2: 1/2/4/8-pin	3 UART0: 2/4/8-pin UART1: 2/4/8-pin UART2: 1/2/4/8-pin	3 UART0: 2/4/8-pin UART1: 2/4/8-pin UART2: 1/2/4/8-pin	3 UART0: 2/4/8-pin UART1: 2/4-pin UART2: 2/4/8-pin	3 UART0: 2/4/8-pin UART1: 2/4-pin UART2: 2/4/8-pin
Customer OTP	4K bits	4K bits	4K bits	4K bits	4K bits
Analog Comparator	2	2	2	2	2
HDMI CEC	Yes	Yes	Yes	No	No
JTAG	4pin /2pin	4pin /2pin	4pin /2pin	4pin /2pin	4pin /2pin
GPIOs	128	128	128	108	108
Over voltage Protected Pads	17	17	17	15	15
GPTP	3	3	3	3	3
Battery-Powered General Purpose Output (BGPO)	3	3	3	1	1
Active Low VBAT-Powered Control Interface (VCI_IN#)	4	4	4	3	3
VBAT-Powered Control Interface (VCI_OUT)	1	1	1	1	1
Active High VBAT-Powered Control Interface (VCI_OVRD_IN)	0	1	1	1	1
2 pin parallel XTAL oscillator	No	Yes	Yes	Yes	Yes
Single ended external 32KHz clock input	Yes	Yes	Yes	Yes	Yes
PROCHOT_IN	Yes	Yes	Yes	No	No
AES Hardware Support	128-256 bit	128-256 bit	128-256 bit	128-256 bit	128-256 bit
SHA Hashing Support	SHA-1 to SHA-512	SHA-1 to SHA-512	SHA-1 to SHA-512	SHA-1 to SHA-512	SHA-1 to SHA-512

© 2020 Microchip Technology Inc.

TABLE 1-1: MEC152X FEATURE LIST BY PACKAGE (CONTINUED)

Features	MEC1523 -144 WFBGA	MEC1521 144 WFBGA	MEC1527 144 WFBGA	MEC1521 128 VTQFP/ WFBGA	MEC1527 128 WFBGA
Public Key Cryptography Support	RSA: 4K bit ECC: 571 bits	RSA: 4K bit ECC: 640 bit	RSA: 4K bit ECC: 640 bit	RSA: 4K bit ECC: 640 bit	RSA: 4K bit ECC: 640 bit
True Random Number Generator (1K bits)	Yes	Yes	Yes	Yes	Yes
Root of Trust	Yes	Yes	Yes	Yes	Yes
Secure Boot	Yes	Yes	Yes	Yes	Yes
Immutable Code	Yes	Yes	Yes	Yes	Yes
Optional OTP Selectable Features (Note 1)					
DSW Power OK	Yes	Yes	Yes	Yes	Yes
EEPROM Lock	Yes	No	No	No	No
Comparator Strap Option	Yes	Yes	Yes	Yes	Yes
QA Testing	Yes	Yes	Yes	Yes	Yes
JTAG Disable	Yes	Yes	Yes	Yes	Yes
Authentication	Yes	Yes	Yes	Yes	Yes
Encrypt ECDH Private Key (Bytes 0-31)	Yes	Yes	Yes	Yes	Yes
AES Encryption Mandatory	Yes	Yes	Yes	Yes	Yes
OTP Write Lock - [0] ECDH Private Key	Yes	Yes	Yes	Yes	Yes
OTP Write Lock - [4] Authentication Key - Public Qx	Yes	Yes	Yes	Yes	Yes
OTP Write Lock - [5] Authentication Key - Public Qy	Yes	Yes	Yes	Yes	Yes
OTP Write Lock - [6] ECDH Public Key 2, Public Rx	Yes	Yes	Yes	Yes	Yes

TABLE 1-1: MEC152X FEATURE LIST BY PACKAGE (CONTINUED)

Features	MEC1523 -144 WFBGA	MEC1521 144 WFBGA	MEC1527 144 WFBGA	MEC1521 128 VTQFP/ WFBGA	MEC1527 128 WFBGA
OTP Write Lock - [7] ECDH Public Key 2, Public Ry	Yes	Yes	Yes	Yes	Yes
TAG0 SPI Flash Base Address	Yes	Yes	Yes	Yes	Yes

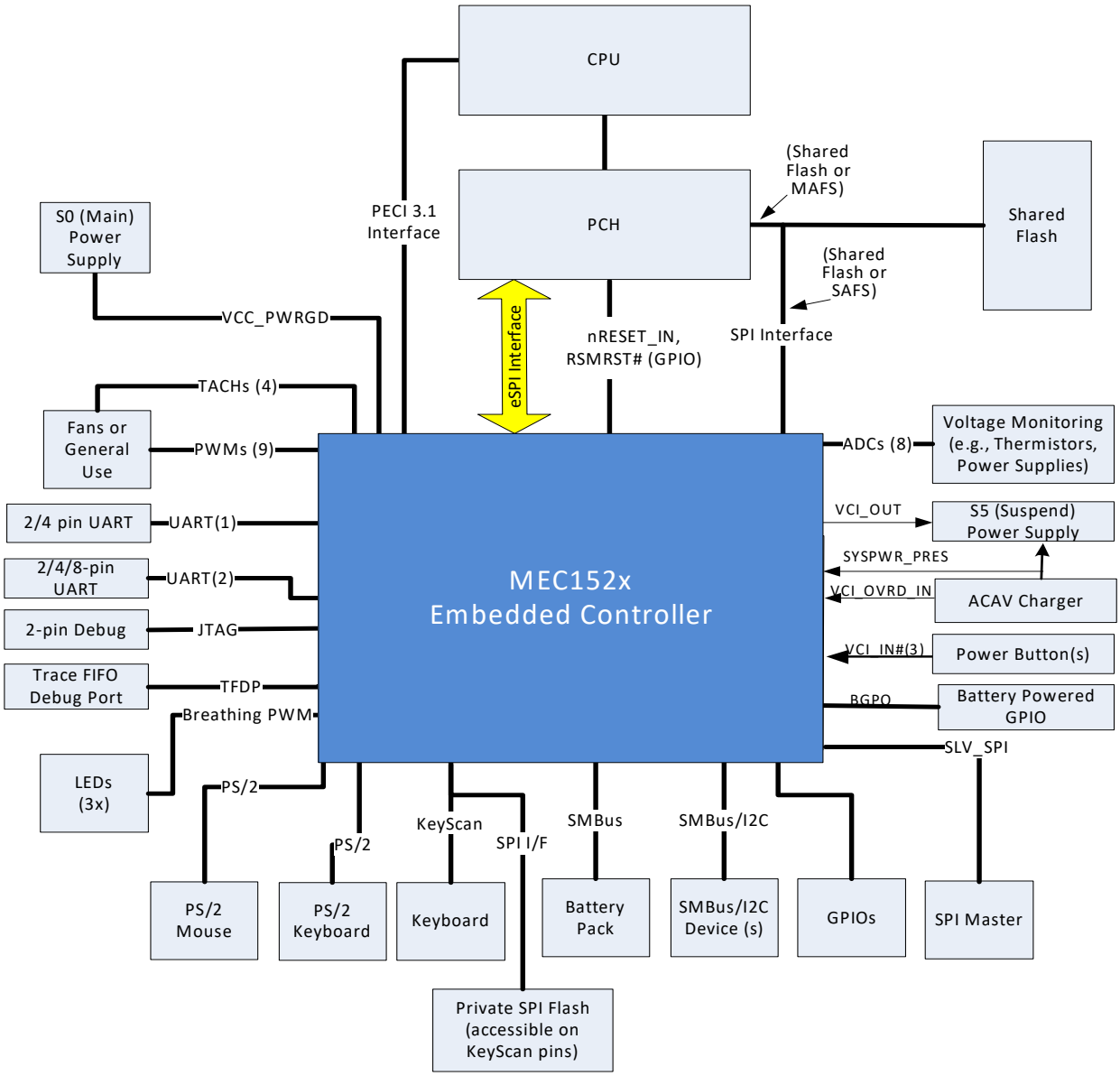
Note 1: Please refer to Boot ROM document for below set of optional OTP selectable feature.

1.2 Boot ROM

Following the release of the RESET_EC signal, the processor will start executing code in the Boot ROM. The Boot ROM executes the SPI Flash Loader, which downloads User Code from SPI Flash and stores it in the internal Code RAM. Refer to MEC152x Boot ROM document for further details.

© 2020 Microchip Technology Inc.

1.3 System Block Diagram

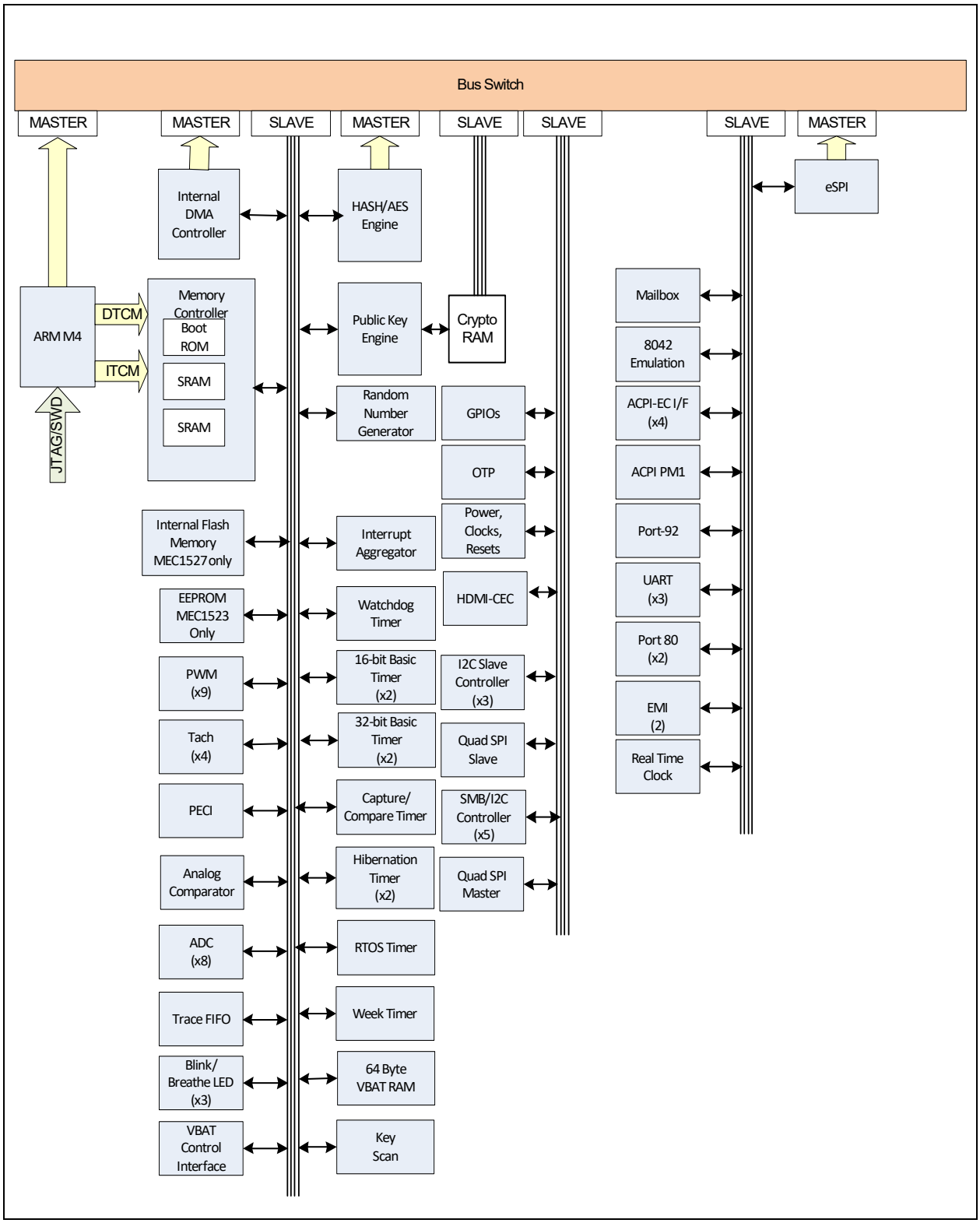


1.4 MEC152x Internal Address Spaces

The Internal Embedded Controller can access any register in the EC Address Space or Host Address Space. The eSPI Host Controllers can directly access peripheral registers in the Host Address Space. If the I^2C interface is used as the Host Interface, access to all the IP Peripherals is dependent on EC firmware.

Note:	The eSPI Host Controllers also have access to the SRAM data space via the SRAM Memory BAR, which is not illustrated below.

FIGURE 1-1: BLOCK DIAGRAM



© 2020 Microchip Technology Inc.

2.0 PIN CONFIGURATION

2.1 Description

The Pin Configuration chapter includes Pin List, Pin Multiplexing.

2.2 Terminology and Symbols for Pins/Buffers

2.2.1 BUFFER TERMINOLOGY

Term	Definition
#	The '#' sign at the end of a signal name indicates an active-low signal
n	The lowercase 'n' preceding a signal name indicates an active-low signal
PWR	Power
PIO	Programmable as Input, Output, Open Drain Output, Bi-directional or Bi-directional with Open Drain Output. Configurable drive strength from 2ma to12ma. **Note:** All GPIOs have programmable drive strength options of 2ma, 4ma, 8ma and 12ma. GPIO pin drive strength is determined by the Pin Control Register Defaults field in the Pin Control Register 2.
In	I Type Input Buffer.
O2	O-2 mA Type Buffer.
PECI	PECI Input/Output. These pins operate at the processor voltage level (VREF_VTT)
SB-TSI	SB-TSI Input/Output. These pins operate at the processor voltage level (VREF_VTT)

2.2.2 PIN NAMING CONVENTIONS

- Pin Name is composed of the multiplexed options separated by '/'. E.g., GPIOxxxx/SignalA/SignalB.
- The first signal shown in a pin name is the default signal. E.g., GPIOxxxx/SignalA/SignalB means the GPIO is the default signal.
- Parenthesis '()' are used to list aliases or alternate functionality for a single mux option. For example, GPIO062(RESETO#) has only a single mux option, GPIO062, but the signal GPIO062 can also be used or interpreted as RESETO#.
- Square brackets '[ ]' are used to indicate there is a Strap Option on a pin. This is always shown as the last signal on the Pin Name.
- Signal Names appended with a numeric value indicates the Instance Number. E.g., PWM0, PWM1, etc. indicates that PWM0 is the PWM output for PWM Instance 0, PWM1 is the PWM output for PWM Instance 1, etc. The instance number may be omitted if there in only one instance of the IP block implemented.

2.3 Pin List

TABLE 2-1: MEC152X PIN MAP

MEC1521H-B0-I/Z2	MEC1521H-B0-I/TF and MEC1527H-B0-I/TF	MEC1521H-B0-I/SZ and MEC1527H-B0-I/SZ	MEC1523H-B0-I/3Y	MEC1523H-B0-I/SZ	Pin Name
128	D4	C3	B2	C3	SYSPWR_PRES & VCI_IN3#/GPIO000/I2C11_SDA
119	F4	A7	B6	B6	GPIO062(RESETO#)/I2C11_SCL
1	B2	D3	D3	D3	GPIO033/TACH3
		A6	A5	A4	GPIO022/GPSPI_IO3
		C6	C6	C6	GPIO023/GPSPI_CLK

TABLE 2-1: MEC152X PIN MAP (CONTINUED)

MEC1521H-B0-I/Z2	MEC1521H-B0-I/TF and MEC1527H-B0-I/TF	MEC1521H-B0-I/SZ and MEC1527H-B0-I/SZ	MEC1523H-B0-I/3Y	MEC1523H-B0-I/SZ	Pin Name
		A5	B5	B5	GPIO024/GPSPI_CS#/I2C07_SCL_ALT
2	A1	B2	B1	B2	GPIO221/32KHz_OUT/SYS_SHDN#
3	C2	E3	C2	E3	nRESET_IN
4	B1	A1	C1	A1	GPIO057/VCC_PWRGD/GPSPI_CLK_ALT
5	D2	F3	E3	F3	GPIO106/PWROK
6	C1	C2	D1	C2	GPIO226
97	A10	C12	B11	A13	GPIO060/KBRST/TST_CLK_OUT/UART1_DCD#
102	B9	E9	D9	E9	GPIO051/ICT1_TACH1
103	A9	A13	C10	A12	GPIO050/ICT0_TACH0
20	J1	J1	H2	J1	GPIO200/ADC00/TRACEDAT0
21	J2	G3	H4	G3	GPIO201/ADC01/TRACEDAT1
22	K1	K1	J3	K1	GPIO202/ADC02/TRACEDAT2
23	K2	J2	J2	J2	GPIO203/ADC03/TRACEDAT3
24	J4	H3	H3	H3	GPIO204/ADC04
25	L1	K2	K1	K2	GPIO205/ADC05
26	L2	L2	K2	L2	GPIO206/ADC06
27	K4	M2	L4	M2	GPIO207/ADC07[CMP_STRAP]
30	K5	N1	L3	N1	GPIO064/PCI_RESET#
		H1	J1	H1	GPIO067/VREF2_ADC
31	M2	K3	L1	K3	GPIO066/ESPI_CS#/I2C13_SDA
29	M1	J3	K3	J3	GPIO061/ESPI_RESET#/PWM7_ALT/nEC_SCI_ALT
32	N1	L3	M1	L3	GPIO065/ESPI_CLKI2C13_SCL/ICT5_ALT
33	N2	L4	N1	L4	GPIO070/ESPI_IO0/I2C14_SDA
35	M3	N3	N3	N3	GPIO071/ESPI_IO1/I2C14_SCL
36	N3	M3	N2	M3	GPIO072/ESPI_IO2/I2C01_SDA_ALT
38	M4	L5	M2	L5	GPIO073/ESPI_IO3/I2C01_SCL_ALT
		M1	L2	M1	GPIO100/nEC_SCI_ALT2/ICT6
39	K6	J6	L5	J6	GPIO011/nSMI_ALT/PWM4/ICT7
40	N4	M4	M3	M4	GPIO063/ESPI_ALERT#/PWM6_ALT/ICT8
		J8	K8	J8	GPIO222/PROCHOT_IN#
41	N5	N4	N4	N4	GPIO224/GPTP_IN0/SHD_IO1
42	N6	M5	M4	M5	GPIO016/GPTP_IN1/SHD_IO3/ICT3(DSW_PWROK)
43	M5	L6	M5	L6	GPIO227/SHD_IO2[PWRGD_STRAP]

© 2020 Microchip Technology Inc.

TABLE 2-1: MEC152X PIN MAP (CONTINUED)

MEC1521H-B0-I/Z2	MEC1521H-B0-I/TF and MEC1527H-B0-I/TF	MEC1521H-B0-I/SZ and MEC1527H-B0-I/SZ	MEC1523H-B0-I/3Y	MEC1523H-B0-I/SZ	Pin Name
44	N7	N5	N5	N5	GPIO223/SHD_IO0
45	M6	N6	N7	N6	GPIO055(RSMRST#)/PWM2/SHD_CS0#[BSS_STRAP]
46	M7	M6	N6	M6	GPIO056/PWM3/SHD_CLK
47	K7	N2	M6	N2	GPIO012/I2C07_SDA/SLV_SPI_IO2
48	K8	J7	L6	J7	GPIO013/I2C07_SCL/SLV_SPI_IO3
50	K9	N7	K6	N7	GPIO130/I2C01_SDA/SLV_SPI_IO0
51	H10	M7	M7	M7	GPIO131/I2C01_SCL/SLV_SPI_CS#
57	N10	L9	M9	L9	GPIO020/KSI1
58	M10	J9	L8	J9	GPIO021/KSI2
52	K10	N8	K7	N8	GPIO052/ICT2_TACH2
53	N8	L7	N10	L7	GPIO002/PWM5/SHD_CS1#
54	M8	M8	M8	M8	GPIO014/PWM6/SLV_SPI_IO2/GPTP_IN2
		M9	N9	M9	GPIO015/PWM7/ICT10
59	M11	N10	M10	N10	GPIO151/ICT4/KSO15
16	G4	L1	G2	L1	GPIO152/KSO14/I2C07_SDA_ALT
60	N11	N11	M12	N11	GPIO017/KSI0/UART0_DCD#
61	M12	M10	M11	M10	GPIO040/GPTP_OUT2/KSO00/UART1_CTS#
62	N12	N12	N13	N12	GPIO032/KSI7/GPTP_OUT0/UART0_RI#
63	M13	L10	N11	L10	GPIO031/KSI6/GPTP_OUT1
98	D10	B12	B12	C11	GPIO132/I2C06_SDA/KSO16
99	B10	C13	C11	B13	GPIO140/I2C06_SCL/ICT5/KSO17
72	H12	L13	K11	J13	GPIO115/PS2_DAT0A
		D2	D4	D2	GPIO025/nEMI_INT/UART_CLK/UART1_RI#
64	N13	N13	L11	M12	GPIO026/KSI3/UART0_DTR#/I2C12_SDA
55	N9	L8	L7	L8	GPIO053/PWM0/SLV_SPI_MSTR_INT
56	M9	N9	N8	N9	GPIO054/PWM1/SLV_SPI_SCLK
65	L13	M11	N12	M13	GPIO027/KSI4/UART0_DSR#/I2C12_SCL
66	L12	L11	L9	L12	GPIO030/KSI5/I2C10_SDA
67	K12	K11	K9	K11	GPIO107/nSMI/KSO04/I2C10_SCL
68	K13	M12	L12	L13	GPIO120/KSO07/UART1_DTR#
69	H13	L12	K10	K13	GPIO112/KSO05
70	J13	M13	K12	K12	GPIO113/KSO06/ICT9
71	J12	J11	L10	J11	GPIO114/PS2_CLK0A/nEC_SCI

© 2020 Microchip Technology Inc.

TABLE 2-1: MEC152X PIN MAP (CONTINUED)

MEC1521H-B0-I/Z2	MEC1521H-B0-I/TF and MEC1527H-B0-I/TF	MEC1521H-B0-I/SZ and MEC1527H-B0-I/SZ	MEC1523H-B0-I/3Y	MEC1523H-B0-I/SZ	Pin Name
74	G13	K12	H13	H11	GPIO042/PECI_DAT/SB-TSI_DAT
75	F13	H9	H10	H9	GPIO043/SB-TSI_CLK
76	J10	H11	G13	G11	GPIO044/VREF_VTT
		H13	J13	H13	GPIO034/GPSPI_IO2
		B6	A6	A3	GPIO036
		E7	A4	E7	GPIO240
		E1	E2	E1	GPIO035/PWM8/CTOUT1/ICT15
78	G10	J12	H11	H12	GPIO170/UART1_TX/CEC_OUT[JTAG_STRAP]
		B11	B13	B10	GPIO171/UART1_RX/CEC_IN
79	E13	K13	J11	J12	JTAG_RST#
80	G12	J13	J12	G12	GPIO104/UART0_TX/TFDP_-CLK[VTR2_STRAP]
81	F12	H12	F13	G13	GPIO105/UART0_RX/TFDP_-DATA/TRACECLK
82	F9	G13	H12	F13	GPIO046/KSO02/ICT11
83	F10	G9	E11	G9	GPIO047/KSO03/PWM3_ALT/ICT13
84	C13	G12	G12	F12	GPIO121/PVT_IO0/KSO08
85	E12	G11	G10	F11	GPIO122/PVT_IO1/KSO09
86	D12	F12	F11	E12	GPIO123/PVT_IO2/KSO10
87	D13	F11	G11	E11	GPIO126/PVT_IO3/KSO13
17	H1	G2	H1	G2	GPIO124/PVT_CS#/KSO11/ICT12
88	C12	E11	F12	E13	GPIO125/PVT_CLK/KSO12
89	E10	F9	D10	F9	GPIO175/CMP_-VOUT1/PWM8_ALT
91	A12	F13	E13	D12	GPIO127/A20M/UART1_RTS#
92	A13	D11	E12	D11	GPIO156/LED0
100	D9	B13	A11	C10	GPIO157/LED1
101	E9	A12	C12	B11	GPIO153/LED2
93	A11	E12	D12	D13	GPIO007/I2C03_SDA/PS2_CLK0B
94	B13	E13	D11	C13	GPIO010/I2C03_SCL/PS2_DAT0B
95	B11	D13	D13	B12	GPIO154/I2C02_SDA/PS2_CLK1B
96	B12	D12	C13	C12	GPIO155/I2C02_SCL/PS2_DAT1B
10	E2	E2	F2	E2	GPIO246/CTOUT1_ALT/CMP_VREF0
		C9	C9	C9	GPIO245/Single-WireUART/GPSPI_IO0
13	G2	F1	F4	f1	GPIO244/UART_-CLK_ALT/nEMI_INT_ALT/CMP_VIN1
		C11	A10	A9	GPIO243/GPSPI_IO1
11	F2	D1	F1	D1	GPIO242/CMP_VIN0

© 2020 Microchip Technology Inc.

TABLE 2-1: MEC152X PIN MAP (CONTINUED)

MEC1521H-B0-I/Z2	MEC1521H-B0-I/TF and MEC1527H-B0-I/TF	MEC1521H-B0-I/SZ and MEC1527H-B0-I/SZ	MEC1523H-B0-I/3Y	MEC1523H-B0-I/SZ	Pin Name
106	D8	A11	A12	A11	GPIO241/PWM0_ALT/CMP_-VOUT0
12	F1	F2	F3	F2	GPIO254/PWM1_ALT/CMP_VREF1
107	E8	F8	A13	F8	GPIO045/KSO01/PWM2_ALT/ICT14[CR_STRAP]
108	A8	C10	B4	A10	GPIO165/32KHZ_IN/CTOUT0
109	B8	E8	B9	E8	GPIO145/I2C09_SDA/JTAG_TDI/UART2_RX
110	B7	C8	C8	C8	GPIO146/I2C09_SCL/JTAG_TDO/UART2_TX
111	A7	B10	A8	B9	GPIO147/I2C15_SDA/JTAG_-CLK/UART2_DSR#
112	B6	B9	B10	B8	GPIO150/I2C15_SCL/JTAG_TMS/UART2_DTR#
113	A6	A10	A9	A8	GPIO141/I2C05_SDA/UART2_RTS#
114	D7	A9	C7	A7	GPIO142/I2C05_SCL/UART2_CTS#
115	A5	B8	B8	B7	GPIO143/I2C04_SDA/UART0_CTS#
116	B5	C7	D8	C7	GPIO144/I2C04_SCL/UART0_RTS#
117	A3	A8	A7	A6	GPIO004/I2C00_SCL/UART2_DCD#
118	D6	B7	B7	A5	GPIO003/I2C00_SDA/UART2_RI#
125	B3	C5	C3	C5	VCI_IN1#/GPIO162
126	E4	F6	C4	F6	VCI_IN0#/GPIO163
120	D5	B5	A3	B4	BGPO0/GPIO253
		A3	A2	B3	BGPO1/GPIO101
		B3	A1	A1	BGPO2/GPIO102
121	B4	B4			VCI_OVRD_IN/GPIO172
122	F5	E6	C5	E6	VCI_OUT/GPIO250
		C4	B3	E5	VCI_IN2#/GPIO161
8	D1	B1	D2	C1	GPIO255/UART1_RX-_ALT/UART1_DSR#
124	A2	A2			XTAL1
123	A4	A4			XTAL2
			L13	N13	GPIO213
			M13	M11	GPIO211/I2C08_SDA
			K13	L11	GPIO212/I2C08_SCL
	E6	F5	D5	F5	VSS_ANALOG
7	E1	C1	E1	B1	VTR_PLL
127	E5	E5	D6	C4	VBAT
37	E7	G8	J10	G8	VSS1

TABLE 2-1: MEC152X PIN MAP (CONTINUED)

MEC1521H-B0-I/Z2	MEC1521H-B0-I/TF and MEC1527H-B0-I/TF	MEC1521H-B0-I/SZ and MEC1527H-B0-I/SZ	MEC1523H-B0-I/3Y	MEC1523H-B0-I/SZ	Pin Name
15	H2	G5	G4	G5	VTR_REG
19	H4	H2	G3	H2	VREF_ADC
77	G9	G6	D7	G6	VSS2
73	G5	F7	F10	F7	VTR1
	J7	J5	J4	J5	VTR_ANALOG
14	G1	G1	G1	G1	VR_CAP
105	H9	H7	E10	H7	VSS3
49	J8	H8	K5	H8	VTR2
34	J5	H6	K4	H6	VTR3
18	J6	H5	E4	H5	VSS_ADC
28					VTR1_ADC
90	H5				VTR1
9	J9				VTR1
104					VTR1

2.4 Pin Multiplexing

2.4.1 DEFAULT STATE

The default state for analog pins is Input. The default state for all pins that default to a GPIO function is input/output/interrupt disabled. The default state for pins that differ is shown in the Section 3.5, "GPIO Register Assignments". Entries for the Default State column are

- O2ma-Low: Push-Pull output, Slow slew rate, 2ma drive strength, grounded
- O2ma-High Push-Pull output, Slow slew rate, 2ma drive strength, high output
- PU Input, with pull-up resistor enabled

2.4.2 POWER RAIL

The Power Rail column defines the power pin that provides I/O power for the signal pin.

2.4.3 BUFFER TYPES

The Buffer Type column defines the type of Buffer associated with each signal. Some pins have signals with two different buffer types sharing the pin; in this case, table shows the buffer type for each of the signals that share the pin.

Input signals muxed with GPIOs are marked as “I”

Output signals muxed with GPIOs are marked as “PIO”, because the GPIO input path is always active even when the alternate function selected is “output only”. So the GPIO input can be read to see the level of the output signal.

Pad Types are defined in the Section 49.0, "Electrical Specifications," on page 526.

- I/O Pad Types are defined in Section 49.2.4, "DC Electrical Characteristics for I/O Buffers," on page 528.
- The abbreviation “PWR” is used to denote power pins. The power supplies are defined in Section 49.2.1, "Power Supply Operational Characteristics," on page 527.

© 2020 Microchip Technology Inc.

2.4.4 GLITCH PROTECTION

Pins with glitch protection are glitch-free tristate pins and will not drive out while their associated power rail is rising. These glitch-free tristate pins require either an external pull-up or pull-down to set the state of the pin high or low.

Note: If the pin needs to default low, a 1M ohm (max) external pull-down is required.

All pins, except the XTAL pins, are glitch protected.

Note: The power rail must rise monotonically in order for glitch protection to operate.

2.4.5 BGPO GLITCH PROTECTION

All BGPO pins are glitch protected while VBAT power is applied.

The BGPO outputs are glitch protected on VBAT power as well as VTR power. As VBAT rises from ground, the BGPOx output are not driven until the VBAT power rail reaches approximately 1V. Once the VBAT power rail reaches approximately 1V, the BGPO outputs drive low.

Note: It is recommended that a pull-down resistor be added to the BGPO pins.

2.4.6 OVER-VOLTAGE PROTECTION

If a pin is over-voltage protected (over-voltage protection = YES) then the following is true: If the pad is powered by 1.8V +/- 5% (operational) it can tolerate up to 3.63V on the pad. This allows for a pull-up to 3.3V power rail +/- 10%. If the pad is powered by 3.3V +/- 5% (operational) it can tolerate up to 5.5V on the pad. This allows for a pull-up to 5.0V power rail +/- 10%.

If a pin is not over-voltage protected (over-voltage protection = NO) then the following is true: If the pad is powered by 1.8V +/- 5% (operational), it can tolerate up to 1.8V +10% (i.e., +1.98V max). If the pad is powered by 3.3V +/- 5% (operational) it can tolerate up to 3.3V +10% (i.e., +3.63V max).

2.4.7 UNDER-VOLTAGE PROTECTION

Pins that are identified as having Under-voltage PROTECTION may be configured so they will not sink excess current if powered by 3.3V and externally pulled up to 1.8V. The following configuration requirements must be met.

- If the pad is an output only pad type and it is configured as either open drain or the output is disabled.
- If the pin is a GPIO pin with a PIO pad type then is must be configured as open drain output with the input disabled. The input is disabled by setting the GPIO Power Gating Signals (PGS) bits to 11b.

All pins, except the XTAL pins, are under voltage protected.

2.4.8 BACKDRIVE PROTECTION

Assuming that the external voltage on the pin is within the parameters defined for the specific pad type, the backdrive protected pin will not sink excess current when it is at a lower potential than the external circuit. There are two cases where this occurs:

- The pad power is off and the external circuit is powered
- The pad power is on and the external circuitry is pulled to a higher potential than the pad power. This may occur on 3.3V powered pads that are 5V tolerant or on 1.8V powered pads that are 3.6V tolerant.

2.4.9 EMULATED POWER WELL

Power well emulation for GPIOs and for signals that are multiplexed with GPIO signals is controlled by the Power Gating Signals (PGS) option in the GPIO Pin Control Register. The Emulated Power Well column in the Pin Multiplexing table defines the power gating programming options supported for each signal.

Note: VBAT powered signals do not support power emulation and must program the PGS bit field to 00b (VTR)

2.4.10 GATED STATE

This column defines the internal value of an input signal when either its emulated power well is inactive or it is not selected by the GPIO alternate function MUX. A value of “No Gate” means that the internal signal always follows the pin even when the emulated power well is inactive.

Note:	Gated state is only meaningful to the operation of input signals. A gated state on an output pin defines the internal behavior of the GPIO MUX and does not imply pin behavior.
Note:	Only the pins that are 5V tolerant have an entry in the 5VT column in the Pin Description Table.

2.4.11 NOTES

The below notes are for all tables in this chapter.

TABLE 2-2: NUMBERED NOTES

Note	Description
Note 1	An external cap must be connected as close to the VR_CAP pin/ball as possible with a routing resistance and CAP ESR of less than 100mohms. The capacitor value is 1uF and must be ceramic with X5R or X7R dielectric. The cap pin/ball should remain on the top layer of the PCB and traced to the CAP. Avoid adding vias to other layers to minimize inductance.
Note 2	This SMBus ports supports 1 Mbps operation as defined by I2C. For 1 Mbps I2C recommended capacitance/pull-up relationships from Intel, refer to the Shark Bay platform guide, Intel ref number 486714. Refer to the PCH - SMBus 2.0/SMLink Interface Design Guidelines, Bus Capacitance/Pull-Up Resistor Relationship.
Note 3	The GPIO062 pin defaults to output 'low' on Reset_SYS to support the firmware controlled RESETO# feature. RESETO# is not a GPIO alternate function; it is controlled by firmware as a GPIO function.
Note 4	In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_VTT Disable bit to 1.
Note 5	The voltage on the ADC pins must not exceed 3.6 V or damage to the device will occur.
Note 6	The XTAL1 pin should be left floating when using the XTAL2 pin for the single ended clock input.
Note 7	GPIO067/VREF2_ADC used as a GPIO can inject noise into the ADC. Hence care should be taken in system design to make sure this GPIO doesn't switch often or is used only when ADC is not active.
Note 8	This signal is a test signal used to detect when the internal 48MHz clock is toggling or stopped in heavy and deepest sleep modes.
Note 9	The VCI pins may be used as GPIOs. The VCI input signals are not gated by selecting the GPIO alternate function. Firmware must disable (i.e., gate) these inputs by writing the bits in the VCI Input Enable Register when the GPIO function is enabled.
Note 10	The KSI and KSO Key Scan pins require pull-up resistors. The system designer may opt to use either use the internal pull-up resistors or populate external pull-up resistors.
Note 11	The GPTP_OUT always drives at the level of the output buffer regardless of the voltage at the GPTP_IN pin. If the GPTP_IN pin is 1.8V the output essentially level-shifts the voltage up to 3.3V, as GPTP_OUT pins are powered by VTR1 (3.3V)
Note 12	The Over voltage protected GPIO pins will not support the Repeater mode mentioned in the GPIO pin configuration register
Note 13	SB-TSI port is mapped to I2C/SMB Port 10. (port_sel[3:0] = 1010)
Note 14	Refer Configurable Signal Routing section under Pin Configuration chapter for details on using the <signal> and <signal>_ALT. Both <signal> and <signal>_ALT cannot be enabled simultaneously.
Note 15	The nEC_SCI pin can be controlled by hardware and EC firmware. The nEC_SCI pin can drive either the ACPI Run-time GPE Chipset input or the Wake GPE Chipset input. Depending how the nEC_SCI pin is used, other ACPI-related SCI functions may be best supplied by other general purpose outputs that can be configured as open-drain drivers.
Note 16	SYSPWR_PRES&VCI_IN3# is named SYSPWR_PRES_VCI_IN3_n in MPLab Tools
Note 17	<Signal> with ‘#’ as suffix will be shown as <Signal>_n in MPLab Tools

© 2020 Microchip Technology Inc.

TABLE 2-2: NUMBERED NOTES

Note	Description
Note 18	32kHz_IN is named CLK32kHz_IN in MPLab Tools
Note 19	32kHz_OUT is named CLK32kHz_OUT in MPLab Tools
Note 20	Clock Enable Register Bits [3:2] should be configured to be driven by single ended 32Khz source. Connect the pin to SUSCLK from PCH.
Note 21	When the JTAG_RST# pin is not asserted (logic'1'), the JTAG or ARM SWJ signal functions in the JTAG interface are unconditionally routed to the GPIO interface; the Pin Control register for these GPIO pins has no effect.When the JTAG_RST# pin is asserted (logic'0'), the signal functions in the JTAG interface are not routed to the interface and the Pin Control Register for these GPIO pins controls the muxing. The pin control registers can not route the JTAG interface to the pins. System Board Designer should terminate this pin in all functional state using jumpers and pull-up or pull down resistors, etc.
Note 22	PS/2 ports ending with signal functions ending with "A" or "B" are muxed to a single controller. Only one set of clock and data are intended to used at a time (either "A" or "B" not both. The unused port segment should have its associated pin control register's, Mux Control Field programmed away from the PS2 controller.
Note 23	The JTAG signals TDI,TDO,TMS,TCK are muxed with GPIO pins. Routing of JTAG signals to these pins are dependent on DEBUG ENABLE REGISTER bits [2:0] and JTAG_RST# pin (Note 21. To configure these GPIO pins for non JTAG functions, pull JTAG_RST# low externally and select the appropriate alternate function in the Pin Control Register
Note 24	The BGPO pins may be used as GPIO. For this the BGPO power control register and GPIO pin control register needs to be configured
Note 25	GPIO000/VCI_IN3#, if not used must be connected to VBAT through a high impedance resistor of the order of 100k

2.4.12 MEC152X MULTIPLEXING

TABLE 2-3: MEC152X PIN MULTIPLEXING

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
0	GPIO000	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	SYS-PWR_PRES&VCI3_IN#	PIO			PGS=00 (only)	Low			Note 9, Note 16
2	Reserved								
3	I2C11_SDA	PIO			All PGS options	High			
Default: 0	GPIO062 (RESETO#)	PIO	O2ma-Low	VTR1	All PGS options	No Gate		No	Note 3
1	Reserved								
2	I2C11_SCL	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO033	PIO		VTR1	All PGS options	No Gate		Yes	

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
1	TACH3	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO022	PIO		VTR1	All PGS options	No Gate		Yes	
1	GPSPI_IO3	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO023	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	GPSPI_CLK	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO024	PIO		VTR1	All PGS options	No Gate		Yes	
1	GPSPI_CS#	PIO			All PGS options	High			
2	Reserved								
3	I2C07_SCL_ALT	PIO			All PGS options	High			
Default: 0	GPIO221	PIO		VTR1	All PGS options	No Gate		Yes	
1	32KHZ_OUT	PIO			All PGS options	NA			Note 19
2	Reserved								
3	SYS_SHDN#	PIO			PGS=00 (only)	High			
Default: 0	nRESET_IN	I		VTR1	PGS=00 (only)			Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO057	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	VCC_PWRGD	PIO			PGS=00 (only)	High			
2	Reserved								

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
3	GPSPI_CLK_ALT	PIO			All PGS options	NA			Note 14
Default: 0	GPIO106	PIO		VTR1	All PGS options	No Gate		Yes	
1	PWROK	PIO			PGS=00 (only)	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO226	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO060	PIO		VTR1	All PGS options	No Gate		Yes	
1	KBRST	PIO			All PGS options	NA			
2	TST_CLK_OUT	PIO			All PGS options	NA			Note 8
3	UART1_DCD#	PIO			All PGS options	High			Note 17
Default: 0	GPIO051	PIO		VTR1	All PGS options	No Gate		Yes	
1	ICT1_TACH1	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO050	PIO		VTR1	All PGS options	No Gate		Yes	
1	ICT0_TACH0	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO200	PIO		VTR1	All PGS options	No Gate		No	
1	ADC00	I_AN			PGS=00 (only)	Low			Note 5
2	Reserved								
3	Reserved								

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO201	PIO		VTR1	All PGS options	No Gate		No	
1	ADC01	I_AN			PGS=00 (only)	Low			Note 5
2	Reserved								
3	Reserved								
Default: 0	GPIO202	PIO		VTR1	All PGS options	No Gate		No	
1	ADC02	I_AN			PGS=00 (only)	Low			Note 5
2	Reserved								
3	Reserved								
Default: 0	GPIO203	PIO		VTR1	All PGS options	No Gate		No	
1	ADC03	I_AN			PGS=00 (only)	Low			Note 5
2	Reserved								
3	Reserved								
Default: 0	GPIO204	PIO		VTR1	All PGS options	No Gate		No	
1	ADC04	I_AN			PGS=00 (only)	Low			Note 5
2	Reserved								
3	Reserved								
Default: 0	GPIO205	PIO		VTR1	All PGS options	No Gate		No	
1	ADC05	I_AN			PGS=00 (only)	Low			Note 5
2	Reserved								
3	Reserved								
Default: 0	GPIO206	PIO		VTR1	All PGS options	No Gate		No	
1	ADC06	I_AN			PGS=00 (only)	Low			Note 5
2	Reserved								
3	Reserved								
Default: 0	GPIO207	PIO		VTR1	All PGS options	No Gate		No	
1	ADC07	I_AN			PGS=00 (only)	Low			Note 5

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
2	Reserved								
3	Reserved								
Strap	CMP_STRAP	PIO							
Default: 0	GPIO064	PIO		VTR3	All PGS options	No Gate		Yes	
1	PCI_RESET#	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO067	PIO		VTR1	All PGS options	No Gate		Yes	
1	VREF2_ADC	PIO			PGS=00 (only)	Low			Note 7
2	Reserved								
3	Reserved								
Default: 0	GPIO066	PIO		VTR3	All PGS options	No Gate		Yes	
1	ESPI_CS#	PIO			PGS=00 (only)	High			
2	I2C13_SDA	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO061	PIO		VTR3	All PGS options	No Gate		Yes	
1	ESPI_RESET#	PIO			PGS=00 (only)	High			
2	PWM7_ALT	PIO			All PGS options	NA			Note 14
3	nEC_SCI_ALT	PIO			All PGS options	NA			Note 14 Note 15
Default: 0	GPIO065	PIO		VTR3	All PGS options	No Gate		Yes	
1	ESPI_CLK	PIO			PGS=00 (only)	Low			
2	I2C13_SCL	PIO			All PGS options	High			
3	ICT5_ALT	I			All PGS options	Low			Note 14
Default: 0	GPIO070	PIO		VTR3	All PGS options	No Gate		Yes	

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
1	ESPI_IO0	PIO			PGS=00 (only)	Low			
2	I2C14_SDA	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO071	PIO		VTR3	All PGS options	No Gate		Yes	
1	ESPI_IO1	PIO			PGS=00 (only)	Low			
2	I2C14_SCL	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO072	PIO		VTR3	All PGS options	No Gate		Yes	
1	ESPI_IO2	PIO			PGS=00 (only)	Low			
2	I2C01_SDA_ALT	PIO			All PGS options	High			Note 14
3	Reserved								
Default: 0	GPIO073	PIO		VTR3	All PGS options	No Gate		Yes	
1	ESPI_IO3	PIO			PGS=00 (only)	Low			
2	I2C01_SDA_ALT	PIO			All PGS options	High			Note 14
3	Reserved								
Default: 0	GPIO100	PIO		VTR3	All PGS options	No Gate		Yes	
1	nEC_SCI_ALT2	PIO			All PGS options	NA			Note 14 Note 15
2	ICT6	I			All PGS options	Low			
3	Reserved								
Default: 0	GPIO011	PIO		VTR3	All PGS options	No Gate		Yes	
1	nSMI_ALT	PIO			All PGS options	NA			
2	PWM4	PIO			All PGS options	NA			

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
3	ICT7	I			All PGS options	Low			
Default: 0	GPIO063	PIO		VTR3	All PGS options	No Gate		Yes	
1	ESPI_ALERT#	PIO			PGS=00 (only)	NA			
2	PWM6_ALT	PIO			All PGS options	NA			Note 14
3	ICT8	I			All PGS options	Low			
Default: 0	GPIO222	PIO		VTR2	All PGS options	No Gate		Yes	
1	PROCHOT_IN#	I			All PGS options	High			Note 17
2	Reserved								
3	Reserved								
Default: 0	GPIO224	PIO		VTR2	All PGS options	No Gate		Yes	
1	GPTP_IN0	PIO			All PGS options	Low			Note 11
2	SHD_IO1	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO016	PIO		VTR2	All PGS options	No Gate		Yes	
1	GPTP_IN1	PIO			All PGS options	Low			Note 11
2	SHD_IO3	PIO			All PGS options	Low			
3	ICT3	PIO			All PGS options	Low			
Default: 0	GPIO227	PIO		VTR2	All PGS options	No Gate		Yes	
1	SHD_IO2	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Strap	PWRGD_STRAP	PIO							
Default: 0	GPIO223	PIO		VTR2	All PGS options	No Gate		Yes	
1	SHD_IO0	PIO			All PGS options	Low			

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVoltage Protect	Backdrive Protect	Notes
2	Reserved								
3	Reserved								
Default: 0	GPIO055	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM2	PIO			All PGS options	NA			
2	SHD_CS0#	PIO			All PGS options	NA			Note 17
3	Reserved								
Strap	BSS_STRAP	PIO							
Default: 0	GPIO056	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM3	PIO			All PGS options	NA			
2	SHD_CLK	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO012	PIO		VTR2	All PGS options	No Gate	Yes	Yes	Note 12
1	I2C07_SDA	PIO			All PGS options	High			
2	SLV_SPI_IO2	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO013	PIO		VTR2	All PGS options	No Gate	Yes	Yes	Note 12
1	I2C07_SCL	PIO			All PGS options	High			
2	SLV_SPI_IO3	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO130	PIO		VTR2	All PGS options	No Gate	Yes	Yes	Note 12
1	I2C01_SDA	PIO			All PGS options	High			
2	SLV_SPI_IO0	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO131	PIO		VTR2	All PGS options	No Gate	Yes	Yes	Note 12

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
1	I2C01_SCL	PIO			All PGS options	High			
2	SLV_SPI_CS#	I			All PGS options	High			Note 17
3	Reserved								
Default: 0	GPIO020	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI1	PIO			All PGS options	Low			Note 10
2	Reserved								
3	Reserved								
Default: 0	GPIO021	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI2	PIO			All PGS options	Low			Note 10
2	Reserved								
3	Reserved								
Default: 0	GPIO052	PIO		VTR2	All PGS options	No Gate		Yes	
1	ICT2_TACH2	PIO			All PGS options	Low			
2	Reserved								
3	Reserved								
Default: 0	GPIO002	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM5	PIO			All PGS options	NA			
2	SHD_CS1#	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO014	PIO		VTR2	All PGS options	No Gate	Yes	Yes	Note 12
1	PWM6	PIO			All PGS options	NA			
2	SLV_SPI_IO1	PIO			All PGS options	Low			
3	GPTP_IN2	PIO			All PGS options	Low			Note 11
Default: 0	GPIO015	PIO		VTR2	All PGS options	No Gate		Yes	

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
1	PWM7	PIO			All PGS options	NA			
2	ICT10	I			All PGS options	Low			
3	Reserved								
Default: 0	GPIO151	PIO		VTR1	All PGS options	No Gate		Yes	
1	ICT4	PIO			All PGS options	Low			
2	KSO15	PIO			All PGS options	NA			Note 10
3	Reserved								
Default: 0	GPIO152	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSO14	PIO			All PGS options	NA			Note 10
2	Reserved								
3	I2C07_SDA_ALT	PIO			All PGS options	High			
Default: 0	GPIO017	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI0	PIO			All PGS options	Low			Note 10
2	UART0_DCD#	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO040	PIO		VTR1	All PGS options	No Gate		Yes	
1	GPTP_OUT2	PIO			All PGS options	NA			Note 11
2	KSO0	PIO			All PGS options	NA			Note 10
3	UART1_CTS#	I			All PGS options	High			Note 17
Default: 0	GPIO032	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI7	PIO			All PGS options	Low			Note 10
2	GPTP_OUT0	PIO			All PGS options	NA			Note 11
3	UART0_RI#	PIO			All PGS options	High			

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO031	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI6	PIO			All PGS options	Low			Note 10
2	GPTP_OUT1	PIO			All PGS options	NA			Note 11
3	Reserved								
Default: 0	GPIO132	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C06_SDA	PIO			All PGS options	High			
2	KSO16	PIO			All PGS options	NA			Note 10
3	Reserved								
Default: 0	GPIO140	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C06_SCL	PIO			All PGS options	High			
2	ICT5	PIO			All PGS options	Low			
3	KSO17	PIO			All PGS options	NA			Note 10
Default: 0	GPIO115	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	PS2_DAT0A	PIO			All PGS options	Low			Note 22
2	Reserved								
3	Reserved								
Default: 0	GPIO025	PIO		VTR1	All PGS options	No Gate		Yes	
1	nEMI_INT	PIO			All PGS options	NA			
2	UART_CLK	PIO			All PGS options	Low			
3	UART1_RI#	I			All PGS options	High			Note 17
Default: 0	GPIO026	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI3	PIO			All PGS options	Low			Note 10

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
2	UART0_DTR#	PIO			All PGS options	NA			
3	I2C12_SDA	PIO			All PGS options	High			
Default: 0	GPIO053	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM0	PIO			All PGS options	NA			
2	SLV_SPI_M-STR_INT	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO054	PIO		VTR2	All PGS options	No Gate		Yes	
1	PWM1	PIO			All PGS options	NA			
2	SLV_SPI_SCLK	I			All PGS options	Low			
3	Reserved								
Default: 0	GPIO027	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI4	PIO			All PGS options	Low			Note 10
2	UART0_DSR#	PIO			All PGS options	High			
3	I2C12_SCL	PIO			All PGS options	High			
Default: 0	GPIO030	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSI5	PIO			All PGS options	Low			Note 10
2	I2C10_SDA	PIO			All PGS options	High			
3	Reserved								
Default: 0	GPIO107	PIO		VTR1	All PGS options	No Gate		Yes	
1	nSMI	PIO			All PGS options	NA			
2	KSO4	PIO			All PGS options	NA			Note 10
3	I2C10_SCL	PIO			All PGS options	High			

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO120	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSO7	PIO			All PGS options	NA			Note 10
2	UART1_DTR#	PIO			All PGS options	NA			Note 17
3	Reserved								
Default: 0	GPIO112	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSO5	PIO			All PGS options	NA			Note 10
2	Reserved								
3	Reserved								
Default: 0	GPIO113	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSO6	PIO			All PGS options	NA			Note 10
2	ICT9	I			All PGS options	Low			
3	Reserved								
Default: 0	GPIO114	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	PS2_CLK0A	PIO			All PGS options	Low			Note 22
2	nEC_SCI	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO042	PIO		VTR1	All PGS options	No Gate		No	
1	PECI_DAT	PECI_IO			All PGS options	Low			
2	SB-TSI_DAT	PECI_IO			All PGS options	High			Note 13
3	Reserved								
Default: 0	GPIO043	PIO		VTR1	All PGS options	No Gate		No	
1	SB-TSI_CLK	PECI_IO			All PGS options	High			Note 13
2	Reserved								
3	Reserved								

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO044	PIO		VTR1	All PGS options	No Gate		No	
1	VREF_VTT	I_AN			All PGS options	Low			Note 4
2	Reserved								
3	Reserved								
Default: 0	GPIO034	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	GPSPI_IO2	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO036	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO240	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO035	PIO		VTR1	All PGS options	No Gate		Yes	
1	PWM8	PIO			All PGS options	NA			
2	CTOUT1	PIO			All PGS options	NA			
3	ICT15	I			All PGS options	Low			
Default: 0	GPIO170	PIO	PU	VTR1	All PGS options	No Gate		Yes	
1	UART1_TX	PIO			All PGS options	NA			
2	CEC_OUT	PIO			All PGS options	NA			
3	Reserved								
Strap	JTAG_STRAP	PIO							
Default: 0	GPIO171	PIO		VTR1	All PGS options	No Gate		Yes	

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
1	UART1_RX	PIO			All PGS options	Low			
2	CEC_IN	PIO			All PGS options	Low			
3	Reserved								
Default: 0	JTAG_RST#	I		VTR1	N/A			Yes	Note 21
1	Reserved								
2	Reserved								
3	Reserved								
Default: 0	GPIO104	PIO		VTR1	All PGS options	No Gate		Yes	
1	UART0_TX	PIO			All PGS options	NA			
2	TFDP_CLK	PIO			All PGS options	NA			
3	Reserved								
Strap	VTR2_STRAP	PIO							
Default: 0	GPIO105	PIO		VTR1	All PGS options	No Gate		Yes	
1	UART0_RX	PIO			All PGS options	Low			
2	TFDP_DATA	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO046	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSO2	PIO			All PGS options	NA			Note 10
2	Reserved								
3	ICT11	I			All PGS options	Low			
Default: 0	GPIO047	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSO3	PIO			All PGS options	NA			Note 10
2	PWM3_ALT	PIO			All PGS options	NA			Note 14
3	ICT13	I			All PGS options	Low			

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO121	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_IO0	PIO			All PGS options	Low			
2	KSO8	PIO			All PGS options	NA			Note 10
3	Reserved								
Default: 0	GPIO122	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_IO1	PIO			All PGS options	Low			
2	KSO9	PIO			All PGS options	NA			Note 10
3	Reserved								
Default: 0	GPIO123	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	PVT_IO2	PIO			All PGS options	Low			
2	KSO10	PIO			All PGS options	NA			Note 10
3	Reserved								
Default: 0	GPIO126	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_IO3	PIO			All PGS options	Low			
2	KSO13	PIO			All PGS options	NA			Note 10
3	Reserved								
Default: 0	GPIO124	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_CS#	PIO			All PGS options	NA			
2	KSO11	PIO			All PGS options	NA			Note 10
3	ICT12	I			All PGS options	Low			
Default: 0	GPIO125	PIO		VTR1	All PGS options	No Gate		Yes	
1	PVT_CLK	PIO			All PGS options	NA			
2	KSO12	PIO			All PGS options	NA			Note 10

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
3	Reserved								
Default: 0	GPIO175	PIO		VTR1	All PGS options	No Gate		Yes	
1	CMP_VOUT1	PIO			All PGS options	NA			
2	Reserved								
3	PWM8_ALT	PIO			All PGS options	NA			Note 14
Default: 0	GPIO127	PIO		VTR1	All PGS options	No Gate		Yes	
1	A20M	PIO			All PGS options	NA			
2	UART1_RTS#	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO156	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	LED0	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO157	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	LED1	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO153	PIO		VTR1	All PGS options	No Gate		Yes	
1	LED2	PIO			All PGS options	NA			
2	Reserved								
3	Reserved								
Default: 0	GPIO007	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	I2C03_SDA	PIO			All PGS options	High			
2	PS2_CLK0B	PIO			All PGS options	Low			Note 22
3	Reserved								

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO010	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	I2C03_SCL	PIO			All PGS options	High			
2	PS2_DAT0B	PIO			All PGS options	Low			Note 22
3	Reserved								
Default: 0	GPIO154	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	I2C02_SDA	PIO			All PGS options	High			
2	PS2_CLK1B	PIO			All PGS options	Low			Note 22
3	Reserved								
Default: 0	GPIO155	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	I2C02_SCL	PIO			All PGS options	High			
2	PS2_DAT1B	PIO			All PGS options	Low			Note 22
3	Reserved								
Default: 0	GPIO246	PIO		VTR1	All PGS options	No Gate		No	
1	Reserved								
2	CTOUT1_ALT	PIO			All PGS options	NA			Note 14
3	CMP_VREF0	I_AN			PGS=00 (only)	NA			
Default: 0	GPIO245	PIO		VTR1	All PGS options	No Gate		Yes	
1	SingleWire UART	PIO			All PGS options	No Gate			
2	GPSPI_IO0	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO244	PIO		VTR1	All PGS options	No Gate		No	
1	UART_CLK_ALT	I			All PGS options	Low			
2	nEMI_INT_ALT	PIO			All PGS options	NA			Note 14

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
3	CMP_VIN1	I_AN			PGS=00 (only)	NA			
Default: 0	GPIO243	PIO		VTR1	All PGS options	No Gate	Yes	Yes	Note 12
1	Reserved								
2	GPSPI_IO1	PIO			All PGS options	Low			
3	Reserved								
Default: 0	GPIO242	PIO		VTR1	All PGS options	No Gate		No	
1	Reserved								
2	Reserved								
3	CMP_VIN0	I_AN			PGS=00 (only)	NA			
Default: 0	GPIO241	PIO		VTR1	All PGS options	No Gate		Yes	
1	PWM0_ALT	PIO			All PGS options	NA			Note 14
2	Reserved								
3	CMP_VOUT0	PIO			All PGS options	NA			
Default: 0	GPIO254	PIO		VTR1	All PGS options	No Gate		No	
1	PWM1_ALT	PIO			All PGS options	NA			Note 14
2	Reserved								
3	CMP_VREF1	I_AN			PGS=00 (only)	NA			
Default: 0	GPIO045	PIO		VTR1	All PGS options	No Gate		Yes	
1	KSO1	PIO			All PGS options	NA			Note 10
2	PWM2_ALT	PIO			All PGS options	NA			Note 14
3	Reserved								
Strap	CR_STRAP	PIO							
Default: 0	GPIO165	PIO		VTR1	All PGS options	No Gate		Yes	
1	32KHZ_IN	PIO			PGS=00 (only)	Low			Note 18,20
2	Reserved								

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
3	CTOUT0	PIO			All PGS options	NA			
Default: 0	GPIO145	PIO		VTR1	All PGS options	No Gate		Yes	Note 23
1	I2C09_SDA	PIO			All PGS options	High			
2	UART2_RX	I			All PGS options	Low			
3	Reserved								
Default: 0	GPIO146	PIO		VTR1	All PGS options	No Gate		Yes	Note 23
1	I2C09_SCL	PIO			All PGS options	High			
2	UART2_TX	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO147	PIO		VTR1	All PGS options	No Gate		Yes	Note 23
1	I2C15_SDA	PIO			All PGS options	High			
2	UART2_DSR#	I			All PGS options	High			Note 17
3	Reserved								
Default: 0	GPIO150	PIO		VTR1	All PGS options	No Gate		Yes	Note 23
1	I2C15_SCL	PIO			All PGS options	High			
2	UART2_DTR#	PIO			All PGS options	NA			Note 17
3	Reserved								
Default: 0	GPIO141	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C05_SDA	PIO			All PGS options	High			
2	UART2_RTS#	PIO			All PGS options	NA			Note 17
3	Reserved								
Default: 0	GPIO142	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C05_SCL	PIO			All PGS options	High			

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
2	UART2_CTS#	I			All PGS options	High			Note 17
3	Reserved								
Default: 0	GPIO143	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C04_SDA	PIO			All PGS options	High			
2	UART0_CTS#	I			All PGS options	High			Note 17
3	Reserved								
Default: 0	GPIO144	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C04_SCL	PIO			All PGS options	High			
2	UART0_RTS#	PIO			All PGS options	NA			
3	Reserved								
Default: 0	GPIO004	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C00_SCL	PIO			All PGS options	High			
2	Reserved								
3	UART2_DCD#	PIO			All PGS options	High			Note 17
Default: 0	GPIO003	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C00_SDA	PIO			All PGS options	High			
2	Reserved								
3	UART2_RI#	I			All PGS options	High			Note 17
0	GPIO162	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	VCI_IN1#	ILLK			PGS=00 (only)	No Gate			Note 9
2	Reserved								
3	Reserved								
0	GPIO163	PIO		VBAT	All PGS options	No Gate		Yes	

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 1	VCI_IN0#	ILLK			PGS=00 (only)	No Gate			Note 9
2	Reserved								
3	Reserved								
0	GPIO253	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	BGPO0	PIO	O2ma-Low		PGS=00 (only)	NA			Note 24
2	Reserved								
3	Reserved								
0	GPIO101	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	BGPO1	PIO	O2ma-Low		PGS=00 (only)	NA			Note 24
2	Reserved								
3	Reserved								
0	GPIO102	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	BGPO2	PIO	O2ma-Low		PGS=00 (only)	NA			Note 24
2	Reserved								
3	Reserved								
0	GPIO250	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	VCI_OUT	PIO	O2ma-High		PGS=00 (only)	NA			Note 9
2	Reserved								
3	Reserved								
0	GPIO172	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	VCI_OVRD_IN	ILLK			PGS=00 (only)	No Gate			Note 9
2	Reserved								
3	Reserved								
0	GPIO161	PIO		VBAT	All PGS options	No Gate		Yes	
Default: 1	VCI_IN2#	ILLK			PGS=00 (only)	No Gate			Note 9
2	Reserved								
3	Reserved								

© 2020 Microchip Technology Inc.

TABLE 2-3: MEC152X PIN MULTIPLEXING (CONTINUED)

Mux Value	Signal Name	Buffer Type	Drive Strength	PAD Power Well	Emulated Power Well	Gated State	OverVolt-age Protect	Back-drive Protect	Notes
Default: 0	GPIO255	PIO		VTR1	All PGS options	No Gate		Yes	
1	UART1_RX_ALT	I			All PGS options	Low			Note 14
2	UART1_DSR#	I			All PGS options	High			Note 17
3	Reserved								
Default: 0	GPIO211	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C08_SDA	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO212	PIO		VTR1	All PGS options	No Gate		Yes	
1	I2C08_SCL	PIO			All PGS options	High			
2	Reserved								
3	Reserved								
Default: 0	GPIO213	PIO		VTR1	All PGS options	No Gate		Yes	
1	Reserved								
2	Reserved								
3	Reserved								

Configurable Signal Routing

To accommodate the signal routing across packages, some Signals are routed to more than one GPIO. At any given time, only the <Signal> or <Signal>_ALT can be selected. Both cannot be selected at the same time.

TABLE 2-4: GPIO ALTERNATE FUNCTIONS

Function	GPIO <Signal>	Alternate GPIO <Signal>_ALT
I2C01_SCL	GPIO131	GPIO073
I2C01_SDA	GPIO130	GPIO072
nEC_SCI	GPIO114	GPIO061
nSMI	GPIO107	GPIO011
ICT5	GPIO140	GPIO065
PWM0	GPIO053	GPIO241
PWM1	GPIO054	GPIO254
PWM2	GPIO055	GPIO045
PWM3	GPIO056	GPIO047
PWM6	GPIO014	GPIO063

© 2020 Microchip Technology Inc.

2.4.13 SIGNAL DESCRIPTION BY INTERFACE

TABLE 2-5: SIGNAL DESCRIPTION BY INTERFACE

SIG_NAME	Description	Notes
ADC		
ADCxx	ADC channel input	Note 5 'xx' is the index of the ADC input. Refer Family features table to find the number of ADC inputs supported in the package
ESPI Host Interface		
ESPI_ALERT#	eSPI Alert	
ESPI_CLK	eSPI Clock	
ESPI_CS#	eSPI Chip Select	
ESPI_IO0	eSPI Data Pin 0	
ESPI_IO1	eSPI Data Pin 1	
ESPI_IO2	eSPI Data Pin 2	
ESPI_IO3	eSPI Data Pin 3	
ESPI_RESET#	eSPI Reset	
Host Interface		
nEC_SCI	Power management event	
nEMI_INT	EC to host Interrupt output	
nSMI	SMI output	
A20M	KBD GATEA20 Output	
KBRST	Keyboard CPU Reset	
Miscellaneous		
LEDx	LED (Blinking/Breathing PWM) Output 0	'x' is the index of the LED output. Refer Family features table to find the number of LED outputs supported in the package
Slave SPI		
SLV_SPI_SCLK	Slave SPI Clock	
SLV_SPI_IO3	Slave SPI Data 3	
SLV_SPI_IO2	Slave SPI Data 2	
SLV_SPI_IO1	Slave SPI Data 1	
SLV_SPI_IO0	Slave SPI Data 0	
SLV_SPI_CS#	Slave SPI Chip Select	
SLV_SPI_M-STR_INT	Slave SPI interrupt to Master	
Keyboard Scan		
KSI0	Keyboard Scan Matrix Input 0	
KSI1	Keyboard Scan Matrix Input 1	
KSI2	Keyboard Scan Matrix Input 2	
KSI3	Keyboard Scan Matrix Input 3	
KSI4	Keyboard Scan Matrix Input 4	
KSI5	Keyboard Scan Matrix Input 5	
KSI6	Keyboard Scan Matrix Input 6	
KSI7	Keyboard Scan Matrix Input 7	

© 2020 Microchip Technology Inc.

TABLE 2-5: SIGNAL DESCRIPTION BY INTERFACE (CONTINUED)

SIG_NAME	Description	Notes
KSO00	Keyboard Scan Matrix Output 0	
KSO01	Keyboard Scan Matrix Output 1	
KSO02	Keyboard Scan Matrix Output 2	
KSO03	Keyboard Scan Matrix Output 3	
KSO04	Keyboard Scan Matrix Output 4	
KSO05	Keyboard Scan Matrix Output 5	
KSO06	Keyboard Scan Matrix Output 6	
KSO07	Keyboard Scan Matrix Output 7	
KSO08	Keyboard Scan Matrix Output 8	
KSO09	Keyboard Scan Matrix Output 9	
KSO10	Keyboard Scan Matrix Output 10	
KSO11	Keyboard Scan Matrix Output 11	
KSO12	Keyboard Scan Matrix Output 12	
KSO13	Keyboard Scan Matrix Output 13	
KSO14	Keyboard Scan Matrix Output 14	
KSO15	Keyboard Scan Matrix Output 15	
KSO16	Keyboard Scan Matrix Output 16	
KSO17	Keyboard Scan Matrix Output 17	
I2C/SMBus Controller		
I2Cxx_SDA	I2C/SMBus Controller Port 0 Data	Note 2 'xx' is the index of the I2C port. Refer Family features table to find the number of I2C ports supported in the package
I2Cxx_SCL	I2C/SMBus Controller Port 0 Clock	Note 2
SB_TSI_DAT	I2C Controller AMD-TSI port data	
SB_TSI_CLK	I2C Controller AMD-TSI port clock	
Analog Comparator		
CMP_VIN0	Comparator 0 Positive Input	
CMP_VIN1	Comparator 1 Positive Input	
CMP_VOUT0	Comparator 0 Output	
CMP_VOUT1	Comparator 1 Output	
CMP_VREF0	Comparator 0 Negative Input	
CMP_VREF1	Comparator 1 Negative Input	
GPIO		
GPIOx	General Purpose Input Output Pins	
GPTP_IN0	General purpose pass through port input 0	
GPTP_IN1	General purpose pass through port input 1	
GPTP_IN2	General purpose pass through port input 2	
GPTP_OUT0	General purpose pass through port output 0	
GPTP_OUT1	General purpose pass through port output 1	
GPTP_OUT2	General purpose pass through port output 2	
PCR Interface		
XTAL1	32.768 KHz Crystal Output	

TABLE 2-5: SIGNAL DESCRIPTION BY INTERFACE (CONTINUED)

SIG_NAME	Description	Notes
XTAL2	32.768 KHz Crystal Input (single-ended 32.768 KHz clock input)	
32KHZ_OUT	32.768 KHz Digital Output	
32KHZ_IN	32.768 KHz Digital Input	
TST_CLK_OUT	48MHz System clock output	
nRESET_IN	External System Reset Input	
VCC_PWRGD	System Main Power Indication Input	
PWROK	System Main Power Indication Outpu	
PECI		
PECI_DAT	PECI Bus	
VREF_VTT	Processor Interface Voltage Reference	
Quad Mode SPI Controller Ports		
PVT_CS#	Private SPI Chip Select	SPI_CS0# of QMSPI Controller
PVT_IO0	Private SPI Data 0	SPI_IO0 of QMSPI Controller
PVT_IO1	Private SPI Data 1	SPI_IO1 of QMSPI Controller
PVT_IO2	Private SPI Data 2	SPI_IO2 of QMSPI Controller
PVT_IO3	Private SPI Data 3	SPI_IO3 of QMSPI Controller
PVT_CLK	Private SPI Clock	SPI_CLK of QMSPI Controller
SHD_CS1#	Shared SPI Chip Select1	SPI_CS1# of QMSPI Controlelr
SHD_CS0#	Shared SPI Chip Select	SPI_CS0# of QMSPI Controller
SHD_IO0	Shared SPI Data 0	SPI_IO0 of QMSPI Controller
SHD_IO1	Shared SPI Data 1	SPI_IO1 of QMSPI Controller
SHD_IO2	Shared SPI Data 2	SPI_IO2 of QMSPI Controller
SHD_IO3	Shared SPI Data 3	SPI_IO3 of QMSPI Controller
SHD_CLK	Shared SPI Clock	SPI_CLK of QMSPI Controller
GPSPI_CS#	General Purpose SPI Chip Select	SPI_CS0# of QMSPI Controller
GPSPI_IO0	General Purpose SPI Data 0	SPI_IO0 of QMSPI Controller
GPSPI_IO1	General Purpose SPI Data 1	SPI_IO1 of QMSPI Controller
GPSPI_IO2	General Purpose SPI Data 2	SPI_IO2 of QMSPI Controller
GPSPI_IO3	General Purpose SPI Data 3	SPI_IO3 of QMSPI Controller
GPSPI_CLK	General Purpose SPI Clock	SPI_CLK of QMSPI Controller
FAN PWM and Tachometer		
ICT0_TACH0	Fan Tachometer Input 0	
ICT1_TACH1	Fan Tachometer Input 1	
ICT2_TACH2	Fan Tachometer Input 2	
TACH3	Fan Tachometer Input 3	
PWMx	Pulse Width Modulator Output	‘x’ is the index of the PWM output. Refer Family features table to find the number of PWM outputs supported in the package
Input Capture/Compare Timer		
ICTx	Input capture timer input	‘x’ is the index of the PWM output. Refer Family features table to find the number of ICT inputs supported in the package
CTOUT0	Compare timer 0 toggle output	

© 2020 Microchip Technology Inc.

TABLE 2-5: SIGNAL DESCRIPTION BY INTERFACE (CONTINUED)

SIG_NAME	Description	Notes
CTOUT1	Compare timer 1 toggle output	
PS/2 Interface		
PS2_CLK0A	PS/2 clock 0 - Port A	Note 22
PS2_DAT0A	PS/2 data 0 - Port A	Note 22
PS2_CLK0B	PS/2 clock 0 - Port B	Note 22
PS2_DAT0B	PS/2 data 0 - Port B	Note 22
PS2_CLK1B	PS/2 clock 1 - Port B	Note 22
PS2_DAT1B	PS/2 data 1 - Port B	Note 22
Serial Ports		
UART_CLK	UART Baud Clock Input	
UART0_RX	UART Receive Data (RXD)	
UART0_TX	UART Transmit Data (TXD)	
UART0_CTS#	Clear to Send Input	
UART0_RTS#	Request to Send Output	
UART0_RI#	Ring Indicator Input	
UART0_DCD#	Data Carrier Detect Input	
UART0_DSR#	Data Set Ready Input	
UART0_DTR#	Data Terminal Ready Output	
UART1_RX	UART Receive Data (RXD)k	
UART1_TX	UART Transmit Data (TXD)	
UART1_CTS#	Clear to Send Input	
UART1_RTS#	Request to Send Output	
UART2_RX	UART Receive Data (RXD)	
UART2_TX	UART Transmit Data (TXD)	
UART2_CTS#	Clear to Send Input	
UART2_RTS#	Request to Send Output	
UART2_RI#	Ring Indicator Input	
UART2_DCD#	Data Carrier Detect Input	
UART2_DSR#	Data Set Ready Input	
UART2_DTR#	Data Terminal Ready Output	
Single-WireUART	Single wire UART Data	
VBAT Powered Control Interface		
BGPOx	Battery Powered General Purpose Output	‘x’ is the index of the BGPO output. Refer Family features table to find the number of BGPOs supported in the package
SYS-PWR_PRES&VCI_IN3#	Battery Powered System Power Present Input/ with VCI_IN3# input can cause wakeup or interrupt event	
VCI_INx#	Input can cause wakeup or interrupt event	‘x’ is the index of the VCI_IN# input. Refer Family features table to find the number of VCI_IN# pins supported in the package
VCI_OUT	Output from combinatorial logic and/or EC	
VCI_OVRD_IN	Input can cause wakeup or interrupt event	
SYS_SHDN#	System Main Power Shut Down Signal	

TABLE 2-5: SIGNAL DESCRIPTION BY INTERFACE (CONTINUED)

SIG_NAME	Description	Notes
HDMI -CEC		
CEC_OUT	HDMI CEC Control bus output	
CEC_IN	HDMI CEC Control bus input	
JTAG		
JTAG_RST#	JTAG test active low reset	Note 11,12
JTAG_TDI	JTAG test data in	Note 11,12
JTAG_TDO	JTAG test data out	Note 11,12
JTAG_CLK	JTAG test clk; SWDCLK	Note 11,12
JTAG_TMS	JTAG test mode select; SWDIO	Note 11,12
TFDP_DATA	Trace FIFO debug port - data	
TFDP_CLK	Trace FIFO debug port - clock	
TRACECLK	ARM Embedded Trace Macro Clock	Trace Port is enabled by setting TRACE_EN bit of ETM Trace enable register in EC Register Bank
TRACEDATA0	ARM Embedded Trace Macro Data 0	
TRACEDATA1	ARM Embedded Trace Macro Data 1	
TRACEDATA2	ARM Embedded Trace Macro Data 2	
TRACEDATA3	ARM Embedded Trace Macro Data 3	
Power Pins		
VREF_ADC	ADC Reference Voltage	
VSS_ADC	Analog ADC supply associated ground	
VBAT	VBAT supply	
VR_CAP	Internal Voltage Regulator Capacitor	Note 1
VSS	VTR associated ground	
VSS_VBAT	VBAT associated ground	
VTR1	VTR Suspend Power Supply	
VTR3	Host Interface Power Supply	
VTR2	Peripheral Power Supply	
VTR_PLL	PLL power supply	
VTR_REG	Main Regulator Power supply	
VTR_ANALOG	VTR power supply for Internal Analog Logic	
VTR1_ADC	VTR power supply for Band gap logic	

2.4.14 STRAPPING OPTIONS

GPIO170 is used for the TAP Controller select strap. If any of the JTAG TAP controllers are used, GPIO170 must only be configured as an output to a VTRx powered external function. GPIO170 may only be configured as an input when the JTAG TAP controllers are not needed or when an external driver does not violate the Slave Select Timing.See Section 48.2.1, "TAP Controller Select Strap Option".

© 2020 Microchip Technology Inc.

TABLE 2-6: STRAP PINS

Pin Name	Strap Name	Strap Define and Value	I/O Power Rail
GPIO170	JTAG_STRAP	1= Boundary Scan The JTAG Port is used to access the Boundary scan TAP controller 0= Normal Operation The JTAG port is used to access the ARM TAP Controller	VTR1
GPIO104	VTR2_STRAP	Voltage Level strap is used to determine if the Shared Flash interface must be configured for 3.3V or 1.8V operation 1= 3.3V Operation 0= 1.8V Operation	VTR1
GPIO045	CR_STRAP	Crisis Recovery Strap 1=Normal Boot Source 0=Use the Private SPI pins to boot from Crisis Recovery flash over Key scan connector Note: This pin requires an external pull-up for normal operation.	VTR1
GPIO207	CMP_STRAP	CMP_STRAP is the Comparator 0 Strap pin. This strap must be enabled in OTP.Note 1 1=Comparator 0 Enabled. 0=Hardware Default (GPIO input)	VTR1
GPIO055/SHD_CS0#	BSS_STRAP	Boot Source Select StrapNote 4 1=Use the Shared SPI pins for BootNote 2 0=Use the eSPI Flash Channel for Boot Note 1	VTR2

Note 1: If the eSPI Flash Channel is used for booting, i.e., eSPI Master Attached Flash Sharing (MAFS), the GPIO055/PWM2/SHD_CS0# pin must be used as RSMRST#. This pin will be driven high by the boot ROM code in order to activate the eSPI flash channel. In addition, the GPIO016/GPTP_IN1/SHD_IO3/ICT3 pin must be used as DSW_PWROK.This pin will also be driven high by the boot ROM code to support Deep Sleep Well timing requirements

2: If the Shared SPI port is used for booting, then any unused GPIO may be used for RSMRST#.

3: The comparator strap option is an optional feature that may be enabled in OTP to enable the Boot ROM to configure and lock the Comparator 0 pins. If the feature is enabled in OTP, and external pull-up/pull-down is required to determine the default comparator behavior. If the strap option is not enabled in OTP, the CMP_STRAP is not supported and no external pull-up or pull-down required. Application firmware may enable the comparator if supported by the specific package.

4: MEC1527 does not have the BSS_STRAP option, it always boots from Internal SPI

GPIO055/SHD_CS0# pin is used to determine the boot source (eSPI Flash channel or shared SPI). In addition, the GPIO055/SHD_CS0# pin is used as an indication that the Shared SPI is powered. This pin must be at a high level for the device to load code from the SPI Flash device.

There is presently a requirement for a pull-up resistor on the SHD_CS0# pin on the board if the Shared SPI flash interface is used so that the SPI_CS0# is detected high while RSMRST# is low. The recommended value of the pull-up resistor on the SHD_CS0# pin may vary depending on the version of the Intel PCH that is used.

This is based on information in the current Intel PCH device specifications regarding the SPI0_CS0# pin. This information, as well as the information regarding other PCH devices, must be verified with Intel:

Skylake PCH LP: the signal is tri-stated with no pull-up or pull-down

- Use a resistor in the 4.7K-100K range (pulled-up to the 3.3V rail that powers the SPI device)

Skylake PCH H: the signal is tri-stated with a weak pull down (~ 20K)

© 2020 Microchip Technology Inc.

- Use a resistor in the 4.7K-8K range (pulled-up to the 3.3V rail that powers the SPI device).

These pull-up values must ensure the voltage on the pin is detected as a high (i.e., VTRx*0.7).

2.5 Pin Default State Through Power Transitions

The power state and power state transitions illustrated in the following tables are defined in Section 4.0, "Power, Clocks, and Resets". Pin behavior in this table assumes no specific programming to change the pin state. All GPIO default pins that have the same behavior are described in the table generically as GPIOXXX.

TABLE 2-7: PIN DEFAULT STATE THROUGH POWER TRANSITIONS

Signal	VBAT Applied	VBAT Stable	VTR Applied	RESET_SYS De-asserted	VCC_PWRGD Asserted	VCC_PWRGD De-asserted	RESET_SYS Asserted	VTR Un-powered	VBAT Un-powered	Note
GPIO062	un-powered	un-powered	Low	Out=0	Out	Out	Z	glitch	un-powered	
GPIO170	un-powered	un-powered	High	In	In	In	Z	glitch	un-powered	
GPIOXXX	un-powered	un-powered	Z	Z	Z	Z	Z	glitch	un-powered	Note D
nRESET_IN	un-powered	un-powered	Low	In	In	In	Z	glitch	un-powered	
BGPOx	Out=0	Out=0	Retain	Retain	Retain	Retain	Retain	Retain	un-powered	Note B
VCI_INx#	In	In	In	In	In	In	In	In	un-powered	
VCI_OUT	Out logic	Out logic	Out logic	Out logic	Out logic	Out logic	Out logic	Out logic	un-powered	Note C
XTAL1	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	
XTAL2	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	Crystal Out	

Legend
(P) = I/O state is driven by protocol while power is applied.

Z = Tristate
In = Input

Notes
Note D: Does not include GPIO062 and GPIO170

Note B: Pin is programmable by the EC and retains its value through a VTR power cycle.

Note C: Pin is programmable by the EC and affected by other VBAT inputs pins.

TABLE 2-8: PIN DEFAULT STATE THROUGH POWER TRANSITIONS

Signal	VBAT Applied	VBAT Stable	VTR Applied	RESET_SYS De-asserted	VCC_PWRGD Asserted	VCC_PWRGD De-asserted	RESET_SYS Asserted	VTR Un-powered	VBAT Un-powered	Note
nSMI	N/A	N/A	N/A	N/A	1> OD(P)> 1	OD(1)	In	glitch	N/A	
KBRST	N/A	N/A	N/A	N/A	1> OD(P)> 1	Z	Z>In	glitch	N/A	Note F

© 2020 Microchip Technology Inc.

TABLE 2-8: PIN DEFAULT STATE THROUGH POWER TRANSITIONS (CONTINUED)

A20M	N/A	N/A	N/A	N/A	1> OD(P)> 1	Z	Z	glitch	N/A	Note F

Legend
(P) = I/O state is driven by protocol while power is applied.

Z = Tristate
In = Input

OD = Open Drain Output
Undriven (1) or driven (0)

Notes

Note F: Pin is programmable by the EC and retains its value through a VTR power cycle

3.0 DEVICE INVENTORY

3.1 Conventions

Term	Definition
Block	Used to identify or describe the logic or IP Blocks implemented in the device.
Reserved	Reserved registers and bits defined in the following table are read only values that return 0 when read. Writes to these reserved registers have no effect.
TEST	Microchip Reserved locations which should not be modified from their default value. Changing a TEST register or a TEST field within a register may cause unwanted results.
b	The letter ‘b’ following a number denotes a binary number.
h	The letter ‘h’ following a number denotes a hexadecimal number.

Register access notation is in the form “Read / Write”. A Read term without a Write term means that the bit is read-only and writing has no effect. A Write term without a Read term means that the bit is write-only, and assumes that reading returns all zeros.

Register Field Type	Field Description
R	**Read:** A register or bit with this attribute can be read.
W	**Write:** A register or bit with this attribute can be written.
RS	**Read to Set:** This bit is set on read.
RC	**Read to Clear:** Content is cleared after the read. Writes have no effect.
WC or W1C	**Write One to Clear:** writing a one clears the value. Writing a zero has no effect.
WZC	**Write Zero to Clear:** writing a zero clears the value. Writing a one has no effect.
WS or W1S	**Write One to Set:** writing a one sets the value to 1. Writing a zero has no effect.
WZS	**Write Zero to Set:** writing a zero sets the value to 1. Writing a one has no effect.

© 2020 Microchip Technology Inc.

3.2 Block Overview and Base Addresses

Table 3-1, "Base address" lists all the IP components, referred to as Blocks, implemented in the design. The registers implemented in each block are accessible by the embedded controller (EC) at an offset from the Base Address shown in Table 3-1, "Base address". The registers can also be accessed by various hosts in the system as below

1. eSPI: Via a bank of Configuration and Runtime Registers as explained in Chapter 9.0, "Enhanced Serial Peripheral Interface (eSPI)".
2. SPI : By Configuring Memory Base and Read/Write limit registers as explained in Chapter 37.0, "Serial Peripheral Interface (SPI) Slave".
3. I2C : I2C host access is handled by firmware
4. JTAG : JTAG port has access to all the registers defined in Table 3-1, "Base address".

TABLE 3-1: BASE ADDRESS

Feature	Instance	Logical Device Number	Base Address
Watchdog Timer			4000_0400h
16-bit Basic Timer	0		4000_0C00h
16-bit Basic Timer	1		4000_0C20h
32-bit Basic Timer	0		4000_0C80h
32-bit Basic Timer	1		4000_0CA0h
Capture-Compare Timers			4000_1000h
DMA Controller			4000_2400h
EEPROM Controller			4000_2C00h
Prochot Monitor			4000_3400h
SMB-I2C Controller	0		4000_4000h
SMB-I2C Controller	1		4000_4400h
SMB-I2C Controller	2		4000_4800h
SMB-I2C Controller	3		4000_4C00h
SMB-I2C Controller	4		4000_5000h
I2C Controller	5		4000_5100h
I2C Controller	6		4000_5200h
I2C Controller	7		4000_5300h
Quad Master SPI			4007_0000h
16-bit PWM	0		4000_5800h
16-bit PWM	1		4000_5810h
16-bit PWM	2		4000_5820h
16-bit PWM	3		4000_5830h
16-bit PWM	4		4000_5840h
16-bit PWM	5		4000_5850h
16-bit PWM	6		4000_5860h
16-bit PWM	7		4000_5870h
16-bit PWM	8		4000_5880h
16-bit Tach	0		4000_6000h
16-bit Tach	1		4000_6010h
16-bit Tach	2		4000_6020h
16-bit Tach	3		4000_6030h
PECI			4000_6400h
RTOS Timer			4000_7400h

© 2020 Microchip Technology Inc.

TABLE 3-1: BASE ADDRESS (CONTINUED)

Feature	Instance	Logical Device Number	Base Address
ADC			4000_7C00h
Trace FIFO			4000_8C00h
PS-2	0		4000_9000h
PS-2	1		4000_9040h
Hibernation Timer	0		4000_9800h
Hibernation Timer	1		4000_9820h
Keyboard Matrix Scan			4000_9C00h
VBAT Register Bank			4000_A400h
VBAT Powered RAM			4000_A800h
Week Timer			4000_AC80h
VBAT-Powered Control Interface			4000_AE00h
Blinking-Breathing LED	0		4000_B800h
Blinking-Breathing LED	1		4000_B900h
Blinking-Breathing LED	2		4000_BA00h
Interrupt Aggregator			4000_E000h
EC Subsystem Registers			4000_FC00h
JTAG			4008_0000h
Power, Clocks and Resets			4008_0100h
GPIOs			4008_1000h
Mailbox		0h	400F_0000h
8042 Emulated Keyboard Controller		1h	400F_0400h
ACPI EC Channel	0	2h	400F_0800h
ACPI EC Channel	1	3h	400F_0C00h
ACPI EC Channel	2	4h	400F_1000h
ACPI EC Channel	3	5h	400F_1400h
ACPI PM1		7h	400F_1C00h
Port 92-Legacy		8h	400F_2000h
UART	0	9h	400F_2400h
UART	1	Ah	400F_2800h
UART	2	Bh	400F_2C00h
eSPI Interface IO Component		Dh	400F_3400h
eSPI Interface Memory Component		Eh	400F_3800h
eSPI SAF Bridge Component	0		4000_8000h
eSPI SAF Communication Registers	0		4007_1000h
Embedded Memory Interface (EMI)	0	10h	400F_4000h
Embedded Memory Interface (EMI)	1	11h	400F_4400h
Real Time Clock		14h	400F_5000h
BIOS Debug Port (Port 80)	0	20h	400F_8000h
BIOS Debug Port (Port 80)	1	21h	400F_8400h
eSPI Virtual Wires		27h	400F_9C00h
32Byte eSPI Test Block		2Fh	400F_BC00h
Global Configuration		3Fh	400F_FF00h
SPI Slave			4000_7000h

© 2020 Microchip Technology Inc.

3.3 Sleep Enable Register Assignments

TABLE 3-2: SLEEP ALLOCATION

Block	Instance	Bit Position	Sleep Enable Register	Clock Required Register	Reset Enable Register
JTAG STAP		0	NA	Clock Required 0	NA
Interrupt		0	Sleep Enable 1	Clock Required 1	Reset Enable 1
PECI		1	Sleep Enable 1	Clock Required 1	Reset Enable 1
Tach	0	2	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	0	4	Sleep Enable 1	Clock Required 1	Reset Enable 1
PMC/CPP reg Bank		5	Sleep Enable 1	Clock Required 1	NA
DMA		6	Sleep Enable 1	Clock Required 1	Reset Enable 1
TFDP		7	Sleep Enable 1	Clock Required 1	Reset Enable 1
PROCESSOR		8	Sleep Enable 1	Clock Required 1	NA
WDT		9	NA	Clock Required 1	Reset Enable 1
SMB	0	10	Sleep Enable 1	Clock Required 1	Reset Enable 1
Tach	1	11	Sleep Enable 1	Clock Required 1	Reset Enable 1
Tach	2	12	Sleep Enable 1	Clock Required 1	Reset Enable 1
Tach	3	13	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	1	20	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	2	21	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	3	22	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	4	23	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	5	24	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	6	25	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	7	26	Sleep Enable 1	Clock Required 1	Reset Enable 1
PWM	8	27	Sleep Enable 1	Clock Required 1	Reset Enable 1
EC Register Bank		29	Sleep Enable 1	Clock Required 1	NA
Basic Timer 16	0	30	Sleep Enable 1	Clock Required 1	Reset Enable 1
Basic Timer 16	1	31	Sleep Enable 1	Clock Required 1	Reset Enable 1
IMAP	0	0	NA	Clock Required 2	Reset Enable 2
UART	0	1	Sleep Enable 2	Clock Required 2	Reset Enable 2
UART	1	2	Sleep Enable 2	Clock Required 2	Reset Enable 2
Global Configuration		12	NA	Clock Required 2	NA
ACPI EC	0	13	NA	NA	Reset Enable 2
ACPI EC	1	14	NA	NA	Reset Enable 2
ACPI PM1		15	NA	NA	Reset Enable 2
8042 Emulation		16	NA	Clock Required 2	Reset Enable 2
Mailbox		17	NA	NA	Reset Enable 2
RTC		18	NA	Clock Required 2	NA
eSPI2AHB		19	Sleep Enable 2	Clock Required 2	NA
SCRATCH_32REGs		20	NA	NA	Reset Enable 2
ACPI EC	2	21	NA	NA	Reset Enable 2
ACPI EC	3	22	NA	NA	Reset Enable 2
Port 80	0	25	NA	NA	Reset Enable 2

© 2020 Microchip Technology Inc.

TABLE 3-2: SLEEP ALLOCATION (CONTINUED)

Block	Instance	Bit Position	Sleep Enable Register	Clock Required Register	Reset Enable Register
Port 80	1	26	NA	NA	Reset Enable 2
SAF_BRIDGE		27	Sleep Enable 2	Clock Required 2	NA
UART	2	28	Sleep Enable 2	Clock Required 2	Reset Enable 2
HDMI-CEC		1	Sleep Enable 3	Clock Required 3	Reset Enable 3
ADC		3	Sleep Enable 3	Clock Required 3	Reset Enable 3
PS2	0	5	Sleep Enable 3	Clock Required 3	Reset Enable 3
PS2	1	6	Sleep Enable 3	Clock Required 3	Reset Enable 3
Hibernation Timer	0	10	Sleep Enable 3	Clock Required 3	Reset Enable 3
Key scan		11	NA	Clock Required 3	Reset Enable 3
SMB	1	13	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	2	14	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	3	15	Sleep Enable 3	Clock Required 3	Reset Enable 3
LED	0	16	Sleep Enable 3	Clock Required 3	Reset Enable 3
LED	1	17	Sleep Enable 3	Clock Required 3	Reset Enable 3
LED	2	18	Sleep Enable 3	Clock Required 3	Reset Enable 3
SMB	4	20	Sleep Enable 3	Clock Required 3	Reset Enable 3
Basic Timer 32	0	23	Sleep Enable 3	Clock Required 3	Reset Enable 3
Basic Timer 32	1	24	Sleep Enable 3	Clock Required 3	Reset Enable 3
Hibernation Timer	1	29	Sleep Enable 3	Clock Required 3	Reset Enable 3
CCT	0	30	Sleep Enable 3	Clock Required 3	Reset Enable 3
RTOS Timer		6	NA	Clock Required 4	Reset Enable 4
Quad SPI Master		8	Sleep Enable 4	Clock Required 4	Reset Enable 4
I2C	5	10	Sleep Enable 4	Clock Required 4	Reset Enable 4
I2C	6	11	Sleep Enable 4	Clock Required 4	Reset Enable 4
I2C	7	12	Sleep Enable 4	Clock Required 4	Reset Enable 4
PROCHOT	0	13	Sleep Enable 4	Clock Required 4	Reset Enable 4
PSPI	0	14	Sleep Enable 4	Clock Required 4	Reset Enable 4
SLV_SPI	0	16	NA	Clock Required 4	Reset Enable 4

© 2020 Microchip Technology Inc.

3.4 Interrupt Aggregator Bit Assignments

TABLE 3-3: GIRQ_MAPPING

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
GIRQ8	0	GPIO140	GPIO Event	Yes	GPIO Interrupt Event	0	N/A
	1	GPIO141	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO142	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO143	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO144	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO145	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO146	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO147	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO150	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO151	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO152	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO153	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO154	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO155	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO156	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO157	GPIO Event	Yes	GPIO Interrupt Event		
	16	Reserved					
	17	GPIO161	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO162	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO163	GPIO Event	Yes	GPIO Interrupt Event		
	20	Reserved					
	21	GPIO165	GPIO Event	Yes	GPIO Interrupt Event		
	22-23	Reserved					
	24	GPIO170	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO171	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO172	GPIO Event	Yes	GPIO Interrupt Event		
	27-28	Reserved					
	29	GPIO175	GPIO Event	Yes	GPIO Interrupt Event		
	30-31	Reserved					
GIRQ9	0	GPIO100	GPIO Event	Yes	GPIO Interrupt Event	1	N/A
	1	GPIO101	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO102	GPIO Event	Yes	GPIO Interrupt Event		
	3	Reserved					
	4	GPIO104	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO105	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO106	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO107	GPIO Event	Yes	GPIO Interrupt Event		

© 2020 Microchip Technology Inc.

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	8-9	Reserved					
	10	GPIO112	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO113	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO114	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO115	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO116	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO117	GPIO Event	Yes	GPIO Interrupt Event		
	16	GPIO120	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO121	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO122	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO123	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO124	GPIO Event	Yes	GPIO Interrupt Event		
	21	GPIO125	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO126	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO127	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO130	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO131	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO132	GPIO Event	Yes	GPIO Interrupt Event		
	27-31	Reserved					
GIRQ10	0	GPIO040	GPIO Event	Yes	GPIO Interrupt Event	2	N/A
	1	Reserved					
	2	GPIO042	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO043	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO044	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO045	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO046	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO047	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO050	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO051	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO052	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO053	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO054	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO055	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO056	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO057	GPIO Event	Yes	GPIO Interrupt Event		
	16	GPIO060	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO061	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO062	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO063	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO064	GPIO Event	Yes	GPIO Interrupt Event		
	21	GPIO065	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO066	GPIO Event	Yes	GPIO Interrupt Event		

© 2020 Microchip Technology Inc.

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	23	GPIO067	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO070	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO071	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO072	GPIO Event	Yes	GPIO Interrupt Event		
	27	GPIO073	GPIO Event	Yes	GPIO Interrupt Event		
	28	GPIO074	GPIO Event	Yes	GPIO Interrupt Event		
	29	GPIO075	GPIO Event	Yes	GPIO Interrupt Event		
	31	Reserved					
GIRQ11	0	GPIO000	GPIO Event	Yes	GPIO Interrupt Event	3	N/A
	1	Reserved					
	2	GPIO002	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO003	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO004	GPIO Event	Yes	GPIO Interrupt Event		
	5-6	Reserved					
	7	GPIO007	GPIO Event	Yes	GPIO Interrupt Event		
	8	GPIO010	GPIO Event	Yes	GPIO Interrupt Event		
	9	GPIO011	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO012	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO013	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO014	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO015	GPIO Event	Yes	GPIO Interrupt Event		
	14	GPIO016	GPIO Event	Yes	GPIO Interrupt Event		
	15	GPIO017	GPIO Event	Yes	GPIO Interrupt Event		
	16	GPIO020	GPIO Event	Yes	GPIO Interrupt Event		
	17	GPIO021	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO022	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO023	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO024	GPIO Event	Yes	GPIO Interrupt Event		
	21	GPIO025	GPIO Event	Yes	GPIO Interrupt Event		
	22	GPIO026	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO027	GPIO Event	Yes	GPIO Interrupt Event		
	24	GPIO030	GPIO Event	Yes	GPIO Interrupt Event		
	25	GPIO031	GPIO Event	Yes	GPIO Interrupt Event		
	26	GPIO032	GPIO Event	Yes	GPIO Interrupt Event		
	27	GPIO033	GPIO Event	Yes	GPIO Interrupt Event		
	28	GPIO034	GPIO Event	Yes	GPIO Interrupt Event		
	29	GPIO035	GPIO Event	Yes	GPIO Interrupt Event		
	30	GPIO036	GPIO Event	Yes	GPIO Interrupt Event		
	31	Reserved					
GIRQ12	0	GPIO200	GPIO Event	Yes	GPIO Interrupt Event	4	N/A
	1	GPIO201	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO202	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO203	GPIO Event	Yes	GPIO Interrupt Event		

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	4	GPIO204	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO205	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO206	GPIO Event	Yes	GPIO Interrupt Event		
	7	GPIO207	GPIO Event	Yes	GPIO Interrupt Event		
	8	Reserved					
	9	GPIO211	GPIO Event	Yes	GPIO Interrupt Event		
	10	GPIO212	GPIO Event	Yes	GPIO Interrupt Event		
	11	GPIO213	GPIO Event	Yes	GPIO Interrupt Event		
	12-16	Reserved					
	17	GPIO221	GPIO Event	Yes	GPIO Interrupt Event		
	18	GPIO222	GPIO Event	Yes	GPIO Interrupt Event		
	19	GPIO223	GPIO Event	Yes	GPIO Interrupt Event		
	20	GPIO224	GPIO Event	Yes	GPIO Interrupt Event		
	21	Reserved					
	22	GPIO226	GPIO Event	Yes	GPIO Interrupt Event		
	23	GPIO227	GPIO Event	Yes	GPIO Interrupt Event		
	24-31	Reserved					
GIRQ13	0	SMB-I2C Controller0	SMB-I2C	No	SMB-I2C Controller 0 Interrupt Event	5	20
	1	SMB-I2C Controller1	SMB-I2C	No	SMB-I2C Controller 1 Interrupt Event		21
	2	SMB-I2C Controller2	SMB-I2C	No	SMB-I2C Controller 2 Interrupt Event		22
	3	SMB-I2C Controller3	SMB-I2C	No	SMB-I2C Controller 3 Interrupt Event		23
	4	SMB-I2C Controller4	SMB-I2C	No	SMB-I2C Controller 4 Interrupt Event		158
	5	I2C Controller5	I2C	No	Slave I2C Controller 5 Interrupt Event		168
	6	I2C Controller6	I2C	No	Slave I2C Controller 6 Interrupt Event		169
	7	I2C Controller7	I2C	No	Slave I2C Controller 7 Interrupt Event		170
	8-31	Reserved					
GIRQ14	0	DMA Controller	DMA0	No	DMA Controller - Channel 0 Interrupt Event	6	24
	1	DMA Controller	DMA1	No	DMA Controller - Channel 1 Interrupt Event		25
	2	DMA Controller	DMA2	No	DMA Controller - Channel 2 Interrupt Event		26
	3	DMA Controller	DMA3	No	DMA Controller - Channel 3 Interrupt Event		27
	4	DMA Controller	DMA4	No	DMA Controller - Channel 4 Interrupt Event		28

© 2020 Microchip Technology Inc.

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	5	DMA Controller	DMA5	No	DMA Controller - Channel 5 Interrupt Event		29
	6	DMA Controller	DMA6	No	DMA Controller - Channel 6 Interrupt Event		30
	7	DMA Controller	DMA7	No	DMA Controller - Channel 7 Interrupt Event		31
	8	DMA Controller	DMA8	No	DMA Controller - Channel 8 Interrupt Event		32
	9	DMA Controller	DMA9	No	DMA Controller - Channel 9 Interrupt Event		33
	10	DMA Controller	DMA10	No	DMA Controller - Channel 10 Interrupt Event		34
	11	DMA Controller	DMA11	No	DMA Controller - Channel 11 Interrupt Event		35
	12-31	Reserved					
GIRQ15	0	UART 0	UART	No	UART Interrupt Event	7	40
	1	UART 1	UART	No	UART Interrupt Event		41
	2	EMI 0	Host-to-EC	No	Embedded Memory Interface 0 - Host-to-EC Interrupt		42
	3	EMI 1	Host-to-EC	No	Embedded Memory Interface 1 - Host-to-EC Interrupt		43
	4	UART2	UART	No	UART Interrupt Event		44
	5	ACPI EC Interface 0	IBF	No	ACPI EC Interface 0 - Input Buffer Full Event		45
	6	ACPI EC Interface 0	OBE	No	ACPI EC Interface 0 - Output Buffer Empty Event, asserted when OBE flag goes to 1		46
	7	ACPI EC Interface 1	IBF	No	ACPI EC Interface 1 - Input Buffer Full Event		47
	8	ACPI EC Interface 1	OBE	No	ACPI EC Interface 1 - Output Buffer Empty Event, asserted when OBE flag goes to 1		48
	9	ACPI EC Interface 2	IBF	No	ACPI EC Interface 2 - Input Buffer Full Event		49
	10	ACPI EC Interface 2	OBE	No	ACPI EC Interface 2 - Output Buffer Empty Event, asserted when OBE flag goes to 1		50
	11	ACPI EC Interface 3	IBF	No	ACPI EC Interface 3 - Input Buffer Full Event		51
	12	ACPI EC Interface 3	OBE	No	ACPI EC Interface 3 - Output Buffer Empty Event, asserted when OBE flag goes to 1		52
	13-14	Reserved					
	15	ACPI_PM1	PM1_CTL	No	ACPI_PM1 Interface - PM1_CTL2 Interrupt Event		55

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	16	ACPI_PM1	PM1_EN	No	ACPI_PM1 Interface - PM1_EN2 Interrupt Event		56
	17	ACPI_PM1	PM1_STS	No	ACPI_PM1 Interface - PM1_STS2 Interrupt Event		57
	18	8042 Key-board	OBE	No	8042 Keyboard Controller - Output Buffer Empty Event, asserted when OBE flag goes to 1		58
	19	8042 Key-board	IBF	No	8042 Keyboard Controller - Input Buffer Full Event		59
	20	Mailbox	MBX	No	Mailbox Interface - Host-to-EC Interrupt Event		60
	21	Reserved					
	22	Port 80 Debug 0	BDP_INT	No	Port 80h BIOS Debug Port Event		62
	23	Port 80 Debug 1	BDP_INT	No	Port 80h BIOS Debug Port Event		63
	24-31	Reserved					
GIRQ16	0	Public Key Engine	PKE ERROR	No	PKE core error detected	8	65
	1	Public Key Engine	PKE END	No	PKE completed processing		66
	2	Random Number Gen-erator	RNG	No	RNG completed processing		67
	3	AES	AES	No	Interrupt from AES block		68
	4	Hash	HASH	No	Interrupt from SHA block		69
	5-31	Reserved					
GIRQ17	0	PECI	PECI_INT	No	PECI Host Event	9	70
	1	TACH 0	TACH	No	Tachometer 0 Interrupt Event		71
	2	TACH 1	TACH	No	Tachometer 1 Interrupt Event		72
	3	TACH 2	TACH	No	Tachometer 2 Interrupt Event		73
	4	TACH3	TACH	No	Tachometer 3 Interrupt Event		159
	5	CEC	CEC_INT	No	CEC Interrupt Event		160
	6-7	Reserved					
	8	ADC Control-ler	ADC_Single_Int	No	ADC Controller - Single-Sam-ple ADC Conversion Event		78
	9	ADC Control-ler	ADC_Repeat_Int	No	ADC Controller - Repeat-Sam-ple ADC Conversion Event		79
	10-12	Reserved					
	13	Breathing LED 0	PWM_WDT	No	Blinking LED 0 Watchdog Event		83
	14	Breathing LED 1	PWM_WDT	No	Blinking LED 1 Watchdog Event		84

© 2020 Microchip Technology Inc.

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	15	Breathing LED 2	PWM_WDT	No	Blinking LED 2 Watchdog Event		85
	16	Reserved					
	17	PROCHOT	PHOT	No	Prochot Monitor requires service		87
	18-31	Reserved					
GIRQ18	0	Slave SPI	SPI_EC_INTERRUPT	No	Slave SPI Interrupt	10	90
	1	Quad Master SPI Controller	QMSPI_INT	No	Master SPI Controller Requires Servicing		91
	2-9	Reserved					
	10	PS2 Interface 0	PS2_0_ACT	No	PS/2 Device Interface 0 - Activity Interrupt Event		100
	11	PS2 Interface 1	PS2_1_ACT	No	PS/2 Device Interface 1 - Activity Interrupt Event		101
	12	Reserved					
	13	EEPROM	EEPROM	No	EEPROM Transfer Complete		155
	14-19	Reserved					
	20	Capture Compare Timer	CAPTURE TIMER	No	CCT Counter Event		146
	21	Capture Compare Timer	CAPTURE 0	No	CCT Capture 0 Event		147
	22	Capture Compare Timer	CAPTURE 1	No	CCT Capture 1 Event		148
	23	Capture Compare Timer	CAPTURE 2	No	CCT Capture 2 Event		149
	24	Capture Compare Timer	CAPTURE 3	No	CCT Capture 3 Event		150
	25	Capture Compare Timer	CAPTURE 4	No	CCT Capture 4 Event		151
	26	Capture Compare Timer	CAPTURE 5	No	CCT Capture 5 Event		152
	27	Capture Compare Timer	COMPARE 0	No	CCT Compare 0 Event		153
	28	Capture Compare Timer	COMPARE 1	No	CCT Compare 1 Event		154
	29-31	Reserved					
GIRQ19	0	eSPI_Slave	INTR_PC	No	Peripheral Channel Interrupt	11	103
	1	eSPI_Slave	INTR_BM1	No	Bus Mastering Channel 1 Interrupt		104
	2	eSPI_Slave	INTR_BM2	No	Bus Mastering Channel 2 Interrupt		105
	3	eSPI_Slave	INTR_LTR	No	Peripheral Message (LTR) Interrupt		106

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	4	eSPI_Slave	INTR_OOB_UP	No	Out of Band Channel Up Interrupt		107
	5	eSPI_Slave	INTR_OOB_DOWN	No	Out of Band Channel Down Interrupt		108
	6	eSPI_Slave	INTR_FLASH	No	Flash Channel Interrupt		109
	7	eSPI_Slave	eSPI_RESET	No	eSPI_RESET		110
	8	eSPI_Slave	VWIRE_ENABLE	No	Virtual Wire Channel Enable Asserted		156
	9	SAF bidge EC	EC_CMPLTN	No	EC Completion Event-SAF mode		
	10	SAF ESPI Err	ESPI_ERROR	No	ESPI Error Event-SAF mode		
	11-31	Reserved					
GIRQ20	0-2	Reserved					
	3	OTP	READY_INTR	No	OTP ready interrupt		173
	4-31	Reserved					
GIRQ21	0-1	Reserved				13	
	2	WDT	WDT_INT	Yes	Watch Dog Timer Interupt		171
	3	Week Alarm	WEEK_ALARM_INT	Yes	Week Alarm Interrupt.		114
	4	Week Alarm	SUB-_WEEK_ALARM_INT	Yes	Sub-Week Alarm Interrupt		115
	5	Week Alarm	ONE_SECOND	Yes	Week Alarm - One Second Interrupt		116
	6	Week Alarm	SUB_SECOND	Yes	Week Alarm - Sub-second Interrupt		117
	7	Week Alarm	SYSPWR_PRES	Yes	System power present pin interrupt		118
	8	RTC	RTC	Yes	Real Time Clock Interrupt		119
	9	RTC	RTC ALARM	Yes	Real Time Clock Alarm Interrupt		120
	10	VBAT-Powered Control Interface	VCI_OVRD_IN	Yes	VCI_OVRD_IN active high input pin interrupt		121
	11	VBAT-Powered Control Interface	VCI_IN0	Yes	VCI_IN0 Active-low Input Pin Interrupt		122
	12	VBAT-Powered Control Interface	VCI_IN1	Yes	VCI_IN1 Active-low Input Pin Interrupt		123
	13	VBAT-Powered Control Interface	VCI_IN2	Yes	VCI_IN2 Active-low Input Pin Interrupt		124
	14	VBAT-Powered Control Interface	VCI_IN3	Yes	VCI_IN3 Active-low Input Pin Interrupt		125
	15-17	Reserved					

© 2020 Microchip Technology Inc.

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	18	PS2 Port	PS2_0A_WK	Yes	PS2 Wake Event. Start bit detect.		129
	19	PS2 Port	PS2_0B_WK	Yes	PS2 Wake Event. Start bit detect.		130
	20	Reserved					
	21	PS2 Port	PS2_1B_WK	Yes	PS2 Wake Event. Start bit detect.		132
	22-24	Reserved					
	25	Keyscan	KSC_INT	Yes	Keyboard Scan Interface Run-time Interrupt		135
	26-31	Reserved					
GIRQ22	0	Slave SPI	SPI_ASYNC_WAKE	Yes	Wake-Only Event	N/A	N/A
	1	SMB-I2C Controller0	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.0 START Detected		
	2	SMB-I2C Controller1	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.1 START Detected		
	3	SMB-I2C Controller2	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.2 START Detected		
	4	SMB-I2C Controller3	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.3 START Detected		
	5	SMB-I2C Controller4	SMB-I2C _WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - SMB-I2C.4 START Detected		
	6	I2C Controller5	I2C	Yes	Slave I2C Controller 5 Wake Event		
	7	I2C Controller6	I2C	Yes	Slave I2C Controller 6 Wake Event		
	8	I2C Controller7	I2C	Yes	Slave I2C Controller 7 Wake Event		
	9	ESPI Interface	ESPI_WAKE_ONLY	Yes	Wake-Only Event (No Interrupt Generated) - ESPI Traffic Detected		
	10-31	Reserved					
GIRQ23	0	16-Bit Basic Timer 0	Timer_16_0	No	Basic Timer Event	14	136
	1	16-Bit Basic Timer 1	Timer_16_1	No	Basic Timer Event		137
	2-3	Reserved					
	4	32-Bit Basic Timer 0	Timer_32_0	No	Basic Timer Event		140

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	5	32-Bit Basic Timer 1	Timer_32_1	No	Basic Timer Event		141
	6-9	Reserved					
	10	RTOS Timer	RTOS_TIMER	Yes	32-bit RTOS Timer Event		111
	11	RTOS Timer	SWI_0	No	Soft Interrupt request 0		
	12	RTOS Timer	SWI_1	No	Soft Interrupt request 1		
	13	RTOS Timer	SWI_2	No	Soft Interrupt request 2		
	14	RTOS Timer	SWI_3	No	Soft Interrupt request 3		
	15	Reserved					
	16	Hibernation Timer0	HTIMER	Yes	Hibernation Timer Event		112
	17	Hibernation Timer1	HTIMER	Yes	Hibernation Timer Event		113
	18-31	Reserved					
GIRQ24	0	eSPI_Slave	MSVW00_SRC0	Yes	M-to-S VW Interrupt Event	15	N/A
	1	eSPI_Slave	MSVW00_SRC1	Yes	M-to-S VW Interrupt Event		
	2	eSPI_Slave	MSVW00_SRC2	Yes	M-to-S VW Interrupt Event		
	3	eSPI_Slave	MSVW00_SRC3	Yes	M-to-S VW Interrupt Event		
	4	eSPI_Slave	MSVW01_SRC0	Yes	M-to-S VW Interrupt Event		
	5	eSPI_Slave	MSVW01_SRC1	Yes	M-to-S VW Interrupt Event		
	6	eSPI_Slave	MSVW01_SRC2	Yes	M-to-S VW Interrupt Event		
	7	eSPI_Slave	MSVW01_SRC3	Yes	M-to-S VW Interrupt Event		
	8	eSPI_Slave	MSVW02_SRC0	Yes	M-to-S VW Interrupt Event		
	9	eSPI_Slave	MSVW02_SRC1	Yes	M-to-S VW Interrupt Event		
	10	eSPI_Slave	MSVW02_SRC2	Yes	M-to-S VW Interrupt Event		
	11	eSPI_Slave	MSVW02_SRC3	Yes	M-to-S VW Interrupt Event		
	12	eSPI_Slave	MSVW03_SRC0	Yes	M-to-S VW Interrupt Event		
	13	eSPI_Slave	MSVW03_SRC1	Yes	M-to-S VW Interrupt Event		
	14	eSPI_Slave	MSVW03_SRC2	Yes	M-to-S VW Interrupt Event		
	15	eSPI_Slave	MSVW03_SRC3	Yes	M-to-S VW Interrupt Event		
	16	eSPI_Slave	MSVW04_SRC0	Yes	M-to-S VW Interrupt Event		
	17	eSPI_Slave	MSVW04_SRC1	Yes	M-to-S VW Interrupt Event		
	18	eSPI_Slave	MSVW04_SRC2	Yes	M-to-S VW Interrupt Event		
	19	eSPI_Slave	MSVW04_SRC3	Yes	M-to-S VW Interrupt Event		
	20	eSPI_Slave	MSVW05_SRC0	Yes	M-to-S VW Interrupt Event		
	21	eSPI_Slave	MSVW05_SRC1	Yes	M-to-S VW Interrupt Event		
	22	eSPI_Slave	MSVW05_SRC2	Yes	M-to-S VW Interrupt Event		
	23	eSPI_Slave	MSVW05_SRC3	Yes	M-to-S VW Interrupt Event		
	24	eSPI_Slave	MSVW06_SRC0	Yes	M-to-S VW Interrupt Event		
	25	eSPI_Slave	MSVW06_SRC1	Yes	M-to-S VW Interrupt Event		
	26	eSPI_Slave	MSVW06_SRC2	Yes	M-to-S VW Interrupt Event		
	27	eSPI_Slave	MSVW06_SRC3	Yes	M-to-S VW Interrupt Event		

© 2020 Microchip Technology Inc.

TABLE 3-3: GIRQ_MAPPING (CONTINUED)

Agg IRQ	Agg Bits	HWB Instance Name	Interrupt Event	Wake event	Source description	Agg NVIC	Direct NVIC
	28-31	Reserved					
GIRQ25	0	eSPI_Slave	MSVW07_SRC0	Yes	M-to-S VW Interrupt Event	16	N/A
	1	eSPI_Slave	MSVW07_SRC1	Yes	M-to-S VW Interrupt Event		
	2	eSPI_Slave	MSVW07_SRC2	Yes	M-to-S VW Interrupt Event		
	3	eSPI_Slave	MSVW07_SRC3	Yes	M-to-S VW Interrupt Event		
	4	eSPI_Slave	MSVW08_SRC0	Yes	M-to-S VW Interrupt Event		
	5	eSPI_Slave	MSVW08_SRC1	Yes	M-to-S VW Interrupt Event		
	6	eSPI_Slave	MSVW08_SRC2	Yes	M-to-S VW Interrupt Event		
	7	eSPI_Slave	MSVW08_SRC3	Yes	M-to-S VW Interrupt Event		
	8	eSPI_Slave	MSVW09_SRC0	Yes	M-to-S VW Interrupt Event		
	9	eSPI_Slave	MSVW09_SRC1	Yes	M-to-S VW Interrupt Event		
	10	eSPI_Slave	MSVW09_SRC2	Yes	M-to-S VW Interrupt Event		
	11	eSPI_Slave	MSVW09_SRC3	Yes	M-to-S VW Interrupt Event		
	12	eSPI_Slave	MSVW10_SRC0	Yes	M-to-S VW Interrupt Event		
	13	eSPI_Slave	MSVW10_SRC1	Yes	M-to-S VW Interrupt Event		
	14	eSPI_Slave	MSVW10_SRC2	Yes	M-to-S VW Interrupt Event		
	15	eSPI_Slave	MSVW10_SRC3	Yes	M-to-S VW Interrupt Event		
	16-31	Reserved					
GIRQ26	0	GPIO240	GPIO Event	Yes	GPIO Interrupt Event	17	N/A
	1	GPIO241	GPIO Event	Yes	GPIO Interrupt Event		
	2	GPIO242	GPIO Event	Yes	GPIO Interrupt Event		
	3	GPIO243	GPIO Event	Yes	GPIO Interrupt Event		
	4	GPIO244	GPIO Event	Yes	GPIO Interrupt Event		
	5	GPIO245	GPIO Event	Yes	GPIO Interrupt Event		
	6	GPIO246	GPIO Event	Yes	GPIO Interrupt Event		
	7	Reserved					
	8	GPIO250	GPIO Event	Yes	GPIO Interrupt Event		
	9-10	Reserved					
	11	GPIO253	GPIO Event	Yes	GPIO Interrupt Event		
	12	GPIO254	GPIO Event	Yes	GPIO Interrupt Event		
	13	GPIO255	GPIO Event	Yes	GPIO Interrupt Event		
	14-31	Reserved					

3.5 GPIO Register Assignments

All GPIOs except the below come up in default GPIO Input/output/interrupt disabled state. Pin control register defaults to 0x00008040.

TABLE 3-4: GPIO PIN CONTROL DEFAULT VALUES

GPIO	Pin Control Register Value	Default Function
GPIO000	0x00001040	VCI_IN, SYSPWR_PRES
GPIO161	0x00001040	VCI_IN
GPIO162	0x00001040	VCI_IN
GPIO163	0x00001040	VCI_IN
GPIO172	0x00001040	VCI_OVRD_IN
GPIO062	0x00008240	output
GPIO170	0x00000041	JTAG_STRAP BS (input, pull up)
GPIO116	0x00000041	input, pull up
GPIO250	0x00001240	VCI_OUT

© 2020 Microchip Technology Inc.

3.6 Register Map

TABLE 3-5: REGISTER MAP

Block	Instance	Register	Host Type	Register Address
Watchdog Timer	0	WDT Load Register		40000400
Watchdog Timer	0	WDT Control Register		40000404
Watchdog Timer	0	WDT Kick Register		40000408
Watchdog Timer	0	WDT Count Register		4000040C
Watchdog Timer	0	WDT Status Register		40000410
Watchdog Timer	0	WDT Interrupt Enable Register		40000414
16-bit Basic Timer	0	Timer Count Register		40000C00
16-bit Basic Timer	0	Timer Preload Register		40000C04
16-bit Basic Timer	0	Timer Status Register		40000C08
16-bit Basic Timer	0	Timer Int Enable Register		40000C0C
16-bit Basic Timer	0	Timer Control Register		40000C10
16-bit Basic Timer	1	Timer Count Register		40000C20
16-bit Basic Timer	1	Timer Preload Register		40000C24
16-bit Basic Timer	1	Timer Status Register		40000C28
16-bit Basic Timer	1	Timer Int Enable Register		40000C2C
16-bit Basic Timer	1	Timer Control Register		40000C30
32-bit Basic Timer	0	Timer Count Register		40000C80
32-bit Basic Timer	0	Timer Preload Register		40000C84
32-bit Basic Timer	0	Timer Status Register		40000C88
32-bit Basic Timer	0	Timer Int Enable Register		40000C8C
32-bit Basic Timer	0	Timer Control Register		40000C90
32-bit Basic Timer	1	Timer Count Register		40000CA0
32-bit Basic Timer	1	Timer Preload Register		40000CA4
32-bit Basic Timer	1	Timer Status Register		40000CA8
32-bit Basic Timer	1	Timer Int Enable Register		40000CAC
32-bit Basic Timer	1	Timer Control Register		40000CB0
Capture Compare Timer	0	Capture and Compare Timer Control Register		40001000
Capture Compare Timer	0	Capture Control 0 Register		40001004
Capture Compare Timer	0	Capture Control 1 Register		40001008
Capture Compare Timer	0	Free Running Timer Register		4000100C
Capture Compare Timer	0	Capture 0 Register		40001010
Capture Compare Timer	0	Capture 1 Register		40001014
Capture Compare Timer	0	Capture 2 Register		40001018
Capture Compare Timer	0	Capture 3 Register		4000101C
Capture Compare Timer	0	Capture 4 Register		40001020
Capture Compare Timer	0	Capture 5 Register		40001024
Capture Compare Timer	0	Compare 0 Register		40001028
Capture Compare Timer	0	Compare 1 Register		4000102C
Capture Compare Timer	0	ICT Mux Select Register		40001030
DMA Controller	0	DMA Main Control Register		40002400
DMA Controller	0	DMA Data Packet Register		40002404

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
DMA Controller	0	TEST		40002408
DMA Channel	0	DMA Channel N Activate Register		40002440
DMA Channel	0	DMA Channel N Memory Start Address Register		40002444
DMA Channel	0	DMA Channel N Memory End Address Register		40002448
DMA Channel	0	DMA Channel N Device Address		4000244C
DMA Channel	0	DMA Channel N Control Register		40002450
DMA Channel	0	DMA Channel N Interrupt Status Register		40002454
DMA Channel	0	DMA Channel N Interrupt Enable Register		40002458
DMA Channel	0	TEST		4000245C
DMA Channel	0	Channel N CRC Enable Register		40002460
DMA Channel	0	Channel N CRC Data Register		40002464
DMA Channel	0	Channel N CRC Post Status Register		40002468
DMA Channel	0	TEST		4000246C
DMA Channel	1	DMA Channel N Activate Register		40002480
DMA Channel	1	DMA Channel N Memory Start Address Register		40002484
DMA Channel	1	DMA Channel N Memory End Address Register		40002488
DMA Channel	1	DMA Channel N Device Address		4000248C
DMA Channel	1	DMA Channel N Control Register		40002490
DMA Channel	1	DMA Channel N Interrupt Status Register		40002494
DMA Channel	1	DMA Channel N Interrupt Enable Register		40002498
DMA Channel	1	TEST		4000249C
DMA Channel	1	Channel N Fill Enable Register		400024A0
DMA Channel	1	Channel N Fill Data Register		400024A4
DMA Channel	1	Channel N Fill Status Register		400024A8
DMA Channel	1	TEST		400024AC
DMA Channel	2	DMA Channel N Activate Register		400024C0
DMA Channel	2	DMA Channel N Memory Start Address Register		400024C4
DMA Channel	2	DMA Channel N Memory End Address Register		400024C8
DMA Channel	2	DMA Channel N Device Address		400024CC
DMA Channel	2	DMA Channel N Control Register		400024D0
DMA Channel	2	DMA Channel N Interrupt Status Register		400024D4
DMA Channel	2	DMA Channel N Interrupt Enable Register		400024D8
DMA Channel	2	TEST		400024DC
DMA Channel	3	DMA Channel N Activate Register		40002500
DMA Channel	3	DMA Channel N Memory Start Address Register		40002504
DMA Channel	3	DMA Channel N Memory End Address Register		40002508
DMA Channel	3	DMA Channel N Device Address		4000250C
DMA Channel	3	DMA Channel N Control Register		40002510
DMA Channel	3	DMA Channel N Interrupt Status Register		40002514
DMA Channel	3	DMA Channel N Interrupt Enable Register		40002518
DMA Channel	3	TEST		4000251C
DMA Channel	4	DMA Channel N Activate Register		40002540
DMA Channel	4	DMA Channel N Memory Start Address Register		40002544
DMA Channel	4	DMA Channel N Memory End Address Register		40002548

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
DMA Channel	4	DMA Channel N Device Address		4000254C
DMA Channel	4	DMA Channel N Control Register		40002550
DMA Channel	4	DMA Channel N Interrupt Status Register		40002554
DMA Channel	4	DMA Channel N Interrupt Enable Register		40002558
DMA Channel	4	TEST		4000255C
DMA Channel	5	DMA Channel N Activate Register		40002580
DMA Channel	5	DMA Channel N Memory Start Address Register		40002584
DMA Channel	5	DMA Channel N Memory End Address Register		40002588
DMA Channel	5	DMA Channel N Device Address		4000258C
DMA Channel	5	DMA Channel N Control Register		40002590
DMA Channel	5	DMA Channel N Interrupt Status Register		40002594
DMA Channel	5	DMA Channel N Interrupt Enable Register		40002598
DMA Channel	5	TEST		4000259C
DMA Channel	6	DMA Channel N Activate Register		400025C0
DMA Channel	6	DMA Channel N Memory Start Address Register		400025C4
DMA Channel	6	DMA Channel N Memory End Address Register		400025C8
DMA Channel	6	DMA Channel N Device Address		400025CC
DMA Channel	6	DMA Channel N Control Register		400025D0
DMA Channel	6	DMA Channel N Interrupt Status Register		400025D4
DMA Channel	6	DMA Channel N Interrupt Enable Register		400025D8
DMA Channel	6	TEST		400025DC
DMA Channel	7	DMA Channel N Activate Register		40002600
DMA Channel	7	DMA Channel N Memory Start Address Register		40002604
DMA Channel	7	DMA Channel N Memory End Address Register		40002608
DMA Channel	7	DMA Channel N Device Address		4000260C
DMA Channel	7	DMA Channel N Control Register		40002610
DMA Channel	7	DMA Channel N Interrupt Status Register		40002614
DMA Channel	7	DMA Channel N Interrupt Enable Register		40002618
DMA Channel	7	TEST		4000261C
DMA Channel	8	DMA Channel N Activate Register		40002640
DMA Channel	8	DMA Channel N Memory Start Address Register		40002644
DMA Channel	8	DMA Channel N Memory End Address Register		40002648
DMA Channel	8	DMA Channel N Device Address		4000264C
DMA Channel	8	DMA Channel N Control Register		40002650
DMA Channel	8	DMA Channel N Interrupt Status Register		40002654
DMA Channel	8	DMA Channel N Interrupt Enable Register		40002658
DMA Channel	8	TEST		4000265C
DMA Channel	9	DMA Channel N Activate Register		40002680
DMA Channel	9	DMA Channel N Memory Start Address Register		40002684
DMA Channel	9	DMA Channel N Memory End Address Register		40002688
DMA Channel	9	DMA Channel N Device Address		4000268C
DMA Channel	9	DMA Channel N Control Register		40002690
DMA Channel	9	DMA Channel N Interrupt Status Register		40002694
DMA Channel	9	DMA Channel N Interrupt Enable Register		40002698

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
DMA Channel	9	TEST		4000269C
DMA Channel	10	DMA Channel N Activate Register		400026C0
DMA Channel	10	DMA Channel N Memory Start Address Register		400026C4
DMA Channel	10	DMA Channel N Memory End Address Register		400026C8
DMA Channel	10	DMA Channel N Device Address		400026CC
DMA Channel	10	DMA Channel N Control Register		400026D0
DMA Channel	10	DMA Channel N Interrupt Status Register		400026D4
DMA Channel	10	DMA Channel N Interrupt Enable Register		400026D8
DMA Channel	10	TEST		400026DC
DMA Channel	11	DMA Channel N Activate Register		40002700
DMA Channel	11	DMA Channel N Memory Start Address Register		40002704
DMA Channel	11	DMA Channel N Memory End Address Register		40002708
DMA Channel	11	DMA Channel N Device Address		4000270C
DMA Channel	11	DMA Channel N Control Register		40002710
DMA Channel	11	DMA Channel N Interrupt Status Register		40002714
DMA Channel	11	DMA Channel N Interrupt Enable Register		40002718
DMA Channel	11	TEST		4000271C
EEPROM Controller	0	EEPROM Mode Register		40002C00
EEPROM Controller	0	EEPROM Execute Register		40002C04
EEPROM Controller	0	EEPROM Status Register		40002C08
EEPROM Controller	0	EEPROM Interrupt Enable Register		40002C0C
EEPROM Controller	0	EEPROM Password Register		40002C10
EEPROM Controller	0	EEPROM Unlock Register		40002C14
EEPROM Controller	0	EEPROM lock Register		40002C18
EEPROM Controller	0	TEST		40002C1C
EEPROM Controller	0	EEPROM Buffer Register		40002C20
Prochot Monitor	0	PROCHOT Cumulative Count Register		40003400
Prochot Monitor	0	PROCHOT Duty Cycle Count Register		40003404
Prochot Monitor	0	PROCHOT Duty Cycle Period Register		40003408
Prochot Monitor	0	PROCHOT Status/Control Register		4000340C
Prochot Monitor	0	PROCHOT Assertion Counter Register		40003410
Prochot Monitor	0	PROCHOT Assertion Counter Limit Register		40003414
Prochot Monitor	0	TEST		40003418
I2C-SMB	0	Control Register		40004000
I2C-SMB	0	Status Register		40004000
I2C-SMB	0	Own Address Register		40004004
I2C-SMB	0	Data Register		40004008
I2C-SMB	0	Master Command Register		4000400C
I2C-SMB	0	Slave Command Register		40004010
I2C-SMB	0	PEC Register		40004014
I2C-SMB	0	Repeated START Hold Time Register		40004018
I2C-SMB	0	Completion Register		40004020
I2C-SMB	0	Idle Scaling Register		40004024
I2C-SMB	0	Configuration Register		40004028

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
I2C-SMB	0	Bus Clock Register		4000402C
I2C-SMB	0	Block ID Register		40004030
I2C-SMB	0	Revision Register		40004034
I2C-SMB	0	Bit-Bang Control Register		40004038
I2C-SMB	0	TEST		4000403C
I2C-SMB	0	Data Timing Register		40004040
I2C-SMB	0	Time-Out Scaling Register		40004044
I2C-SMB	0	Slave Transmit Buffer Register		40004048
I2C-SMB	0	Slave Receive Buffer Register		4000404C
I2C-SMB	0	Master Transmit Buffer Register		40004050
I2C-SMB	0	Master Receive Buffer Register		40004054
I2C-SMB	0	TEST		40004058
I2C-SMB	0	TEST		4000405C
I2C-SMB	0	Wake Status Register		40004060
I2C-SMB	0	Wake Enable Register		40004064
I2C-SMB	0	TEST		40004068
I2C-SMB	0	Slave address		4000406C
I2C-SMB	0	Promiscuous Interrupt		40004070
I2C-SMB	0	Promiscuous Interrupt Enable		40004074
I2C-SMB	0	Promiscuous Control		40004078
I2C-SMB	1	Control Register		40004400
I2C-SMB	1	Status Register		40004400
I2C-SMB	1	Own Address Register		40004404
I2C-SMB	1	Data Register		40004408
I2C-SMB	1	Master Command Register		4000440C
I2C-SMB	1	Slave Command Register		40004410
I2C-SMB	1	PEC Register		40004414
I2C-SMB	1	Repeated START Hold Time Register		40004418
I2C-SMB	1	Completion Register		40004420
I2C-SMB	1	Idle Scaling Register		40004424
I2C-SMB	1	Configuration Register		40004428
I2C-SMB	1	Bus Clock Register		4000442C
I2C-SMB	1	Block ID Register		40004430
I2C-SMB	1	Revision Register		40004434
I2C-SMB	1	Bit-Bang Control Register		40004438
I2C-SMB	1	TEST		4000443C
I2C-SMB	1	Data Timing Register		40004440
I2C-SMB	1	Time-Out Scaling Register		40004444
I2C-SMB	1	Slave Transmit Buffer Register		40004448
I2C-SMB	1	Slave Receive Buffer Register		4000444C
I2C-SMB	1	Master Transmit Buffer Register		40004450
I2C-SMB	1	Master Receive Buffer Register		40004454
I2C-SMB	1	TEST		40004458
I2C-SMB	1	TEST		4000445C

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
I2C-SMB	1	Wake Status Register		40004460
I2C-SMB	1	Wake Enable Register		40004464
I2C-SMB	1	TEST		40004468
I2C-SMB	1	Slave address		4000446C
I2C-SMB	1	Promiscuous Interrupt		40004470
I2C-SMB	1	Promiscuous Interrupt Enable		40004474
I2C-SMB	1	Promiscuous Control		40004478
I2C-SMB	2	Control Register		40004800
I2C-SMB	2	Status Register		40004800
I2C-SMB	2	Own Address Register		40004804
I2C-SMB	2	Data Register		40004808
I2C-SMB	2	Master Command Register		4000480C
I2C-SMB	2	Slave Command Register		40004810
I2C-SMB	2	PEC Register		40004814
I2C-SMB	2	Repeated START Hold Time Register		40004818
I2C-SMB	2	Completion Register		40004820
I2C-SMB	2	Idle Scaling Register		40004824
I2C-SMB	2	Configuration Register		40004828
I2C-SMB	2	Bus Clock Register		4000482C
I2C-SMB	2	Block ID Register		40004830
I2C-SMB	2	Revision Register		40004834
I2C-SMB	2	Bit-Bang Control Register		40004838
I2C-SMB	2	TEST		4000483C
I2C-SMB	2	Data Timing Register		40004840
I2C-SMB	2	Time-Out Scaling Register		40004844
I2C-SMB	2	Slave Transmit Buffer Register		40004848
I2C-SMB	2	Slave Receive Buffer Register		4000484C
I2C-SMB	2	Master Transmit Buffer Register		40004850
I2C-SMB	2	Master Receive Buffer Register		40004854
I2C-SMB	2	TEST		40004858
I2C-SMB	2	TEST		4000485C
I2C-SMB	2	Wake Status Register		40004860
I2C-SMB	2	Wake Enable Register		40004864
I2C-SMB	2	TEST		40004868
I2C-SMB	2	Slave address		4000486C
I2C-SMB	2	Promiscuous Interrupt		40004870
I2C-SMB	2	Promiscuous Interrupt Enable		40004874
I2C-SMB	2	Promiscuous Control		40004878
I2C-SMB	3	Control Register		40004C00
I2C-SMB	3	Status Register		40004C00
I2C-SMB	3	Own Address Register		40004C04
I2C-SMB	3	Data Register		40004C08
I2C-SMB	3	Master Command Register		40004C0C
I2C-SMB	3	Slave Command Register		40004C10

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
I2C-SMB	3	PEC Register		40004C14
I2C-SMB	3	Repeated START Hold Time Register		40004C18
I2C-SMB	3	Completion Register		40004C20
I2C-SMB	3	Idle Scaling Register		40004C24
I2C-SMB	3	Configuration Register		40004C28
I2C-SMB	3	Bus Clock Register		40004C2C
I2C-SMB	3	Block ID Register		40004C30
I2C-SMB	3	Revision Register		40004C34
I2C-SMB	3	Bit-Bang Control Register		40004C38
I2C-SMB	3	TEST		40004C3C
I2C-SMB	3	Data Timing Register		40004C40
I2C-SMB	3	Time-Out Scaling Register		40004C44
I2C-SMB	3	Slave Transmit Buffer Register		40004C48
I2C-SMB	3	Slave Receive Buffer Register		40004C4C
I2C-SMB	3	Master Transmit Buffer Register		40004C50
I2C-SMB	3	Master Receive Buffer Register		40004C54
I2C-SMB	3	TEST		40004C58
I2C-SMB	3	TEST		40004C5C
I2C-SMB	3	Wake Status Register		40004C60
I2C-SMB	3	Wake Enable Register		40004C64
I2C-SMB	3	TEST		40004C68
I2C-SMB	3	Slave address		40004C6C
I2C-SMB	3	Promiscuous Interrupt		40004C70
I2C-SMB	3	Promiscuous Interrupt Enable		40004C74
I2C-SMB	3	Promiscuous Control		40004C78
I2C-SMB	4	Control Register		40005000
I2C-SMB	4	Status Register		40005000
I2C-SMB	4	Own Address Register		40005004
I2C-SMB	4	Data Register		40005008
I2C-SMB	4	Master Command Register		4000500C
I2C-SMB	4	Slave Command Register		40005010
I2C-SMB	4	PEC Register		40005014
I2C-SMB	4	Repeated START Hold Time Register		40005018
I2C-SMB	4	Completion Register		40005020
I2C-SMB	4	Idle Scaling Register		40005024
I2C-SMB	4	Configuration Register		40005028
I2C-SMB	4	Bus Clock Register		4000502C
I2C-SMB	4	Block ID Register		40005030
I2C-SMB	4	Revision Register		40005034
I2C-SMB	4	Bit-Bang Control Register		40005038
I2C-SMB	4	TEST		4000503C
I2C-SMB	4	Data Timing Register		40005040
I2C-SMB	4	Time-Out Scaling Register		40005044
I2C-SMB	4	Slave Transmit Buffer Register		40005048

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
I2C-SMB	4	Slave Receive Buffer Register		4000504C
I2C-SMB	4	Master Transmit Buffer Register		40005050
I2C-SMB	4	Master Receive Buffer Register		40005054
I2C-SMB	4	TEST		40005058
I2C-SMB	4	TEST		4000505C
I2C-SMB	4	Wake Status Register		40005060
I2C-SMB	4	Wake Enable Register		40005064
I2C-SMB	4	TEST		40005068
I2C-SMB	4	Slave address		4000506C
I2C-SMB	4	Promiscuous Interrupt		40005070
I2C-SMB	4	Promiscuous Interrupt Enable		40005074
I2C-SMB	4	Promiscuous Control		40005078
I2C	0	Control Register		40005100
I2C	0	Status Register		40005100
I2C	0	Own Address Register		40005104
I2C	0	Data Register		40005108
I2C	0	Repeated START Hold Time Register		40005118
I2C	0	Completion Register		40005120
I2C	0	Configuration Register		40005128
I2C	0	Bus Clock Register		4000512C
I2C	0	Block ID Register		40005130
I2C	0	Revision Register		40005134
I2C	0	Bit-Bang Control Register		40005138
I2C	0	TEST		4000513C
I2C	0	Data Timing Register		40005140
I2C	0	Time-Out Scaling Register		40005144
I2C	0	TEST		40005158
I2C	0	TEST		4000515C
I2C	0	Wake Status Register		40005160
I2C	0	Wake Enable Register		40005164
I2C	0	TEST		40005168
I2C	0	Slave address		4000516C
I2C	0	Promiscuous Interrupt		40005170
I2C	0	Promiscuous Interrupt Enable		40005174
I2C	0	Promiscuous Control		40005178
I2C	1	Control Register		40005200
I2C	1	Status Register		40005200
I2C	1	Own Address Register		40005204
I2C	1	Data Register		40005208
I2C	1	Repeated START Hold Time Register		40005218
I2C	1	Completion Register		40005220
I2C	1	Configuration Register		40005228
I2C	1	Bus Clock Register		4000522C
I2C	1	Block ID Register		40005230

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
I2C	1	Revision Register		40005234
I2C	1	Bit-Bang Control Register		40005238
I2C	1	TEST		4000523C
I2C	1	Data Timing Register		40005240
I2C	1	Time-Out Scaling Register		40005244
I2C	1	TEST		40005258
I2C	1	TEST		4000525C
I2C	1	Wake Status Register		40005260
I2C	1	Wake Enable Register		40005264
I2C	1	TEST		40005268
I2C	1	Slave address		4000526C
I2C	1	Promiscuous Interrupt		40005270
I2C	1	Promiscuous Interrupt Enable		40005274
I2C	1	Promiscuous Control		40005278
I2C	2	Control Register		40005300
I2C	2	Status Register		40005300
I2C	2	Own Address Register		40005304
I2C	2	Data Register		40005308
I2C	2	Repeated START Hold Time Register		40005318
I2C	2	Completion Register		40005320
I2C	2	Configuration Register		40005328
I2C	2	Bus Clock Register		4000532C
I2C	2	Block ID Register		40005330
I2C	2	Revision Register		40005334
I2C	2	Bit-Bang Control Register		40005338
I2C	2	TEST		4000533C
I2C	2	Data Timing Register		40005340
I2C	2	Time-Out Scaling Register		40005344
I2C	2	TEST		40005358
I2C	2	TEST		4000535C
I2C	2	Wake Status Register		40005360
I2C	2	Wake Enable Register		40005364
I2C	2	TEST		40005368
I2C	2	Slave address		4000536C
I2C	2	Promiscuous Interrupt		40005370
I2C	2	Promiscuous Interrupt Enable		40005374
I2C	2	Promiscuous Control		40005378
QMSPI	0	QMSPI Mode Register		40070000
QMSPI	0	QMSPI Control Register		40070004
QMSPI	0	QMSPI Execute Register		40070008
QMSPI	0	QMSPI Interface Control Register		4007000C
QMSPI	0	QMSPI Status Register		40070010
QMSPI	0	QMSPI Buffer Count Status Register		40070014
QMSPI	0	QMSPI Interrupt Enable Register		40070018

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
QMSPI	0	QMSPI Buffer Count Trigger Register		4007001C
QMSPI	0	QMSPI Transmit Buffer Register		40070020
QMSPI	0	QMSPI Receive Buffer Register		40070024
QMSPI	0	QMSPI Chip Select Timing Register		40070028
QMSPI	0	QMSPI Description Buffer 0 Register		40070030
QMSPI	0	QMSPI Description Buffer 1 Register		40070034
QMSPI	0	QMSPI Description Buffer 2 Register		40070038
QMSPI	0	QMSPI Description Buffer 3 Register		4007003C
QMSPI	0	QMSPI Description Buffer 4 Register		40070040
QMSPI	0	QMSPI Description Buffer 5 Register		40070044
QMSPI	0	QMSPI Description Buffer 6 Register		40070048
QMSPI	0	QMSPI Description Buffer 7 Register		4007004C
QMSPI	0	QMSPI Description Buffer 8 Register		40070050
QMSPI	0	QMSPI Description Buffer 9 Register		40070054
QMSPI	0	QMSPI Description Buffer 10 Register		40070058
QMSPI	0	QMSPI Description Buffer 11 Register		4007005C
QMSPI	0	QMSPI Description Buffer 12 Register		40070060
QMSPI	0	QMSPI Description Buffer 13 Register		40070064
QMSPI	0	QMSPI Description Buffer 14 Register		40070068
QMSPI	0	QMSPI Description Buffer 15 Register		4007006C
16-bit PWM	0	PWMx Counter ON Time Register		40005800
16-bit PWM	0	PWMx Counter OFF Time Register		40005804
16-bit PWM	0	PWMx Configuration Register		40005808
16-bit PWM	0	TEST		4000580C
16-bit PWM	1	PWMx Counter ON Time Register		40005810
16-bit PWM	1	PWMx Counter OFF Time Register		40005814
16-bit PWM	1	PWMx Configuration Register		40005818
16-bit PWM	1	TEST		4000581C
16-bit PWM	2	PWMx Counter ON Time Register		40005820
16-bit PWM	2	PWMx Counter OFF Time Register		40005824
16-bit PWM	2	PWMx Configuration Register		40005828
16-bit PWM	2	TEST		4000582C
16-bit PWM	3	PWMx Counter ON Time Register		40005830
16-bit PWM	3	PWMx Counter OFF Time Register		40005834
16-bit PWM	3	PWMx Configuration Register		40005838
16-bit PWM	3	TEST		4000583C
16-bit PWM	4	PWMx Counter ON Time Register		40005840
16-bit PWM	4	PWMx Counter OFF Time Register		40005844
16-bit PWM	4	PWMx Configuration Register		40005848
16-bit PWM	4	TEST		4000584C
16-bit PWM	5	PWMx Counter ON Time Register		40005850
16-bit PWM	5	PWMx Counter OFF Time Register		40005854
16-bit PWM	5	PWMx Configuration Register		40005858
16-bit PWM	5	TEST		4000585C

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
16-bit PWM	6	PWMx Counter ON Time Register		40005860
16-bit PWM	6	PWMx Counter OFF Time Register		40005864
16-bit PWM	6	PWMx Configuration Register		40005868
16-bit PWM	6	TEST		4000586C
16-bit PWM	7	PWMx Counter ON Time Register		40005870
16-bit PWM	7	PWMx Counter OFF Time Register		40005874
16-bit PWM	7	PWMx Configuration Register		40005878
16-bit PWM	7	TEST		4000587C
16-bit PWM	8	PWMx Counter ON Time Register		40005880
16-bit PWM	8	PWMx Counter OFF Time Register		40005884
16-bit PWM	8	PWMx Configuration Register		40005888
16-bit PWM	8	TEST		4000588C
16-bit Tach	0	TACHx Control Register		40006000
16-bit Tach	0	TACHx Status Register		40006004
16-bit Tach	0	TACHx High Limit Register		40006008
16-bit Tach	0	TACHx Low Limit Register		4000600C
16-bit Tach	1	TACHx Control Register		40006010
16-bit Tach	1	TACHx Status Register		40006014
16-bit Tach	1	TACHx High Limit Register		40006018
16-bit Tach	1	TACHx Low Limit Register		4000601C
16-bit Tach	2	TACHx Control Register		40006020
16-bit Tach	2	TACHx Status Register		40006024
16-bit Tach	2	TACHx High Limit Register		40006028
16-bit Tach	2	TACHx Low Limit Register		4000602C
16-bit Tach	3	TACHx Control Register		40006030
16-bit Tach	3	TACHx Status Register		40006034
16-bit Tach	3	TACHx High Limit Register		40006038
16-bit Tach	3	TACHx Low Limit Register		4000603C
PECI	0	Write Data Register		40006400
PECI	0	Read Data Register		40006404
PECI	0	Control Register		40006408
PECI	0	Status Register 1		4000640C
PECI	0	Status Register 2		40006410
PECI	0	Error Register		40006414
PECI	0	Interrupt Enable 1 Register		40006418
PECI	0	Interrupt Enable 2 Register		4000641C
PECI	0	Optimal Bit Time Register (Low Byte)		40006420
PECI	0	Optimal Bit Time Register (High Byte)		40006424
PECI	0	TEST		40006428
PECI	0	TEST		4000642C
PECI	0	BAUD_CTRL		40006430
PECI	0	Block ID Register		40006440
PECI	0	Revision Register		40006444
RTOS Timer	0	RTOS Timer Count Register		40007400

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
RTOS Timer	0	RTOS Timer Preload Register		40007404
RTOS Timer	0	RTOS Timer Control Register		40007408
RTOS Timer	0	Soft Interrupt Register		4000740C
ADC	0	ADC Control Register		40007C00
ADC	0	ADC Delay Register		40007C04
ADC	0	ADC Status Register		40007C08
ADC	0	ADC Single Register		40007C0C
ADC	0	ADC Repeat Register		40007C10
ADC	0	ADC Channel 0 Reading Register		40007C14
ADC	0	ADC Channel 1 Reading Register		40007C18
ADC	0	ADC Channel 2 Reading Register		40007C1C
ADC	0	ADC Channel 3 Reading Register		40007C20
ADC	0	ADC Channel 4 Reading Register		40007C24
ADC	0	ADC Channel 5 Reading Register		40007C28
ADC	0	ADC Channel 6 Reading Register		40007C2C
ADC	0	ADC Channel 7 Reading Register		40007C30
ADC	0	ADC Channel 8 Reading Register		40007C34
ADC	0	ADC Channel 9 Reading Register		40007C38
ADC	0	ADC Channel 10 Reading Register		40007C3C
ADC	0	ADC Channel 11 Reading Register		40007C40
ADC	0	ADC Configuration Register		40007C7C
ADC	0	VREF Channel Register		40007C80
ADC	0	VREF Control Register		40007C84
ADC	0	SAR ADC Control Register		40007C88
ADC	0	SAR ADC Config Register		40007C8C
TFDP	0	Debug Data Register		40008C00
TFDP	0	Debug Control Register		40008C04
PS2	0	PS2 Transmit Buffer Register		40009000
PS2	0	PS2 Receive Buffer Register		40009000
PS2	0	PS2 Control Register		40009004
PS2	0	PS2 Status Register		40009008
PS2	1	PS2 Transmit Buffer Register		40009040
PS2	1	PS2 Receive Buffer Register		40009040
PS2	1	PS2 Control Register		40009044
PS2	1	PS2 Status Register		40009048
Hibernation Timer	0	HTimer Preload Register		40009800
Hibernation Timer	0	HTimer Control Register		40009804
Hibernation Timer	0	HTimer Count Register		40009808
Hibernation Timer	1	HTimer Preload Register		40009820
Hibernation Timer	1	HTimer Control Register		40009824
Hibernation Timer	1	HTimer Count Register		40009828
Keyscan	0	KSO Select Register		40009C04
Keyscan	0	KSI INPUT Register		40009C08
Keyscan	0	KSI STATUS Register		40009C0C

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
Keyscan	0	KSI INTERRUPT ENABLE Register		40009C10
Keyscan	0	Keyscan Extended Control Register		40009C14
VBAT Register Bank	0	Power-Fail and Reset Status Register		4000A400
VBAT Register Bank	0	TEST		4000A404
VBAT Register Bank	0	Clock Enable Register		4000A408
VBAT Register Bank	0	TEST		4000A40C
VBAT Register Bank	0	TEST		4000A410
VBAT Register Bank	0	TEST		4000A414
VBAT Register Bank	0	TEST		4000A41C
VBAT Register Bank	0	Monotonic Counter Register		4000A420
VBAT Register Bank	0	Counter HiWord Register		4000A424
VBAT Register Bank	0	TEST		4000A428
VBAT Register Bank	0	TEST		4000A42C
VBAT Powered RAM	0	Registers		4000A800
Week Timer	0	Control Register		4000AC80
Week Timer	0	Week Alarm Counter Register		4000AC84
Week Timer	0	Week Timer Compare Register		4000AC88
Week Timer	0	Clock Divider Register		4000AC8C
Week Timer	0	Sub-Second Programmable Interrupt Select Register		4000AC90
Week Timer	0	Sub-Week Control Register		4000AC94
Week Timer	0	Sub-Week Alarm Counter Register		4000AC98
Week Timer	0	BGPO Data Register		4000AC9C
Week Timer	0	BGPO Power Register		4000ACA0
Week Timer	0	BGPO Reset Register		4000ACA4
VBAT-Powered Control Interface	0	VCI Register		4000AE00
VBAT-Powered Control Interface	0	Latch Enable Register		4000AE04
VBAT-Powered Control Interface	0	Latch Resets Register		4000AE08
VBAT-Powered Control Interface	0	VCI Input Enable Register		4000AE0C
VBAT-Powered Control Interface	0	Holdoff Count Register		4000AE10
VBAT-Powered Control Interface	0	VCI Polarity Register		4000AE14
VBAT-Powered Control Interface	0	VCI Posedge Detect Register		4000AE18
VBAT-Powered Control Interface	0	VCI Negedge Detect Register		4000AE1C
VBAT-Powered Control Interface	0	VCI Buffer Enable Register		4000AE20
Blinking-Breathing PWM	0	LED Configuration Register		4000B800
Blinking-Breathing PWM	0	LED Limits Register		4000B804
Blinking-Breathing PWM	0	LED Delay Register		4000B808

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
Blinking-Breathing PWM	0	LED Update Stepsize Register		4000B80C
Blinking-Breathing PWM	0	LED Update Interval Register		4000B810
Blinking-Breathing PWM	0	LED Output Delay		4000B814
Blinking-Breathing PWM	1	LED Configuration Register		4000B900
Blinking-Breathing PWM	1	LED Limits Register		4000B904
Blinking-Breathing PWM	1	LED Delay Register		4000B908
Blinking-Breathing PWM	1	LED Update Stepsize Register		4000B90C
Blinking-Breathing PWM	1	LED Update Interval Register		4000B910
Blinking-Breathing PWM	1	LED Output Delay		4000B914
Blinking-Breathing PWM	2	LED Configuration Register		4000BA00
Blinking-Breathing PWM	2	LED Limits Register		4000BA04
Blinking-Breathing PWM	2	LED Delay Register		4000BA08
Blinking-Breathing PWM	2	LED Update Stepsize Register		4000BA0C
Blinking-Breathing PWM	2	LED Update Interval Register		4000BA10
Blinking-Breathing PWM	2	LED Output Delay		4000BA14
Interrupt Aggregator	0	GIRQ8 Source Register		4000E000
Interrupt Aggregator	0	GIRQ8 Enable Set Register		4000E004
Interrupt Aggregator	0	GIRQ8 Result Register		4000E008
Interrupt Aggregator	0	GIRQ8 Enable Clear Register		4000E00C
Interrupt Aggregator	0	GIRQ9 Source Register		4000E014
Interrupt Aggregator	0	GIRQ9 Enable Set Register		4000E018
Interrupt Aggregator	0	GIRQ9 Result Register		4000E01C
Interrupt Aggregator	0	GIRQ9 Enable Clear Register		4000E020
Interrupt Aggregator	0	GIRQ10 Source Register		4000E028
Interrupt Aggregator	0	GIRQ10 Enable Set Register		4000E02C
Interrupt Aggregator	0	GIRQ10 Result Register		4000E030
Interrupt Aggregator	0	GIRQ10 Enable Clear Register		4000E034
Interrupt Aggregator	0	GIRQ11 Source Register		4000E03C
Interrupt Aggregator	0	GIRQ11 Enable Set Register		4000E040
Interrupt Aggregator	0	GIRQ11 Result Register		4000E044
Interrupt Aggregator	0	GIRQ11 Enable Clear Register		4000E048
Interrupt Aggregator	0	GIRQ12 Source Register		4000E050
Interrupt Aggregator	0	GIRQ12 Enable Set Register		4000E054
Interrupt Aggregator	0	GIRQ12 Result Register		4000E058
Interrupt Aggregator	0	GIRQ12 Enable Clear Register		4000E05C
Interrupt Aggregator	0	GIRQ13 Source Register		4000E064
Interrupt Aggregator	0	GIRQ13 Enable Set Register		4000E068
Interrupt Aggregator	0	GIRQ13 Result Register		4000E06C
Interrupt Aggregator	0	GIRQ13 Enable Clear Register		4000E070
Interrupt Aggregator	0	GIRQ14 Source Register		4000E078
Interrupt Aggregator	0	GIRQ14 Enable Set Register		4000E07C
Interrupt Aggregator	0	GIRQ14 Result Register		4000E080
Interrupt Aggregator	0	GIRQ14 Enable Clear Register		4000E084
Interrupt Aggregator	0	GIRQ15 Source Register		4000E08C

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
Interrupt Aggregator	0	GIRQ15 Enable Set Register		4000E090
Interrupt Aggregator	0	GIRQ15 Result Register		4000E094
Interrupt Aggregator	0	GIRQ15 Enable Clear Register		4000E098
Interrupt Aggregator	0	GIRQ16 Source Register		4000E0A0
Interrupt Aggregator	0	GIRQ16 Enable Set Register		4000E0A4
Interrupt Aggregator	0	GIRQ16 Result Register		4000E0A8
Interrupt Aggregator	0	GIRQ16 Enable Clear Register		4000E0AC
Interrupt Aggregator	0	GIRQ17 Source Register		4000E0B4
Interrupt Aggregator	0	GIRQ17 Enable Set Register		4000E0B8
Interrupt Aggregator	0	GIRQ17 Result Register		4000E0BC
Interrupt Aggregator	0	GIRQ17 Enable Clear Register		4000E0C0
Interrupt Aggregator	0	GIRQ18 Source Register		4000E0C8
Interrupt Aggregator	0	GIRQ18 Enable Set Register		4000E0CC
Interrupt Aggregator	0	GIRQ18 Result Register		4000E0D0
Interrupt Aggregator	0	GIRQ18 Enable Clear Register		4000E0D4
Interrupt Aggregator	0	GIRQ19 Source Register		4000E0DC
Interrupt Aggregator	0	GIRQ19 Enable Set Register		4000E0E0
Interrupt Aggregator	0	GIRQ19 Result Register		4000E0E4
Interrupt Aggregator	0	GIRQ19 Enable Clear Register		4000E0E8
Interrupt Aggregator	0	GIRQ20 Source Register		4000E0F0
Interrupt Aggregator	0	GIRQ20 Enable Set Register		4000E0F4
Interrupt Aggregator	0	GIRQ20 Result Register		4000E0F8
Interrupt Aggregator	0	GIRQ20 Enable Clear Register		4000E0FC
Interrupt Aggregator	0	GIRQ21 Source Register		4000E104
Interrupt Aggregator	0	GIRQ21 Enable Set Register		4000E108
Interrupt Aggregator	0	GIRQ21 Result Register		4000E10C
Interrupt Aggregator	0	GIRQ21 Enable Clear Register		4000E110
Interrupt Aggregator	0	GIRQ22 Source Register		4000E118
Interrupt Aggregator	0	GIRQ22 Enable Set Register		4000E11C
Interrupt Aggregator	0	GIRQ22 Result Register		4000E120
Interrupt Aggregator	0	GIRQ22 Enable Clear Register		4000E124
Interrupt Aggregator	0	GIRQ23 Source Register		4000E12C
Interrupt Aggregator	0	GIRQ23 Enable Set Register		4000E130
Interrupt Aggregator	0	GIRQ23 Result Register		4000E134
Interrupt Aggregator	0	GIRQ23 Enable Clear Register		4000E138
Interrupt Aggregator	0	GIRQ24 Source Register		4000E140
Interrupt Aggregator	0	GIRQ24 Enable Set Register		4000E144
Interrupt Aggregator	0	GIRQ24 Result Register		4000E148
Interrupt Aggregator	0	GIRQ24 Enable Clear Register		4000E14C
Interrupt Aggregator	0	GIRQ25 Source Register		4000E154
Interrupt Aggregator	0	GIRQ25 Enable Set Register		4000E158
Interrupt Aggregator	0	GIRQ25 Result Register		4000E15C
Interrupt Aggregator	0	GIRQ25 Enable Clear Register		4000E160
Interrupt Aggregator	0	GIRQ26 Source Register		4000E168

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
Interrupt Aggregator	0	GIRQ26 Enable Set Register		4000E16C
Interrupt Aggregator	0	GIRQ26 Result Register		4000E170
Interrupt Aggregator	0	GIRQ26 Enable Clear Register		4000E174
Interrupt Aggregator	0	Block Enable Set Register		4000E200
Interrupt Aggregator	0	Block Enable Clear Register		4000E204
Interrupt Aggregator	0	Block IRQ Vector Register		4000E208
EC Register Bank	0	TEST		4000FC00
EC Register Bank	0	AHB Error Address Register		4000FC04
EC Register Bank	0	TEST		4000FC08
EC Register Bank	0	TEST		4000FC0C
EC Register Bank	0	TEST		4000FC10
EC Register Bank	0	AHB Error Control Register		4000FC14
EC Register Bank	0	Interrupt Control Register		4000FC18
EC Register Bank	0	ETM TRACE Enable Register		4000FC1C
EC Register Bank	0	Debug Enable Register		4000FC20
EC Register Bank	0	TEST		4000FC24
EC Register Bank	0	WDT Event Count Register		4000FC28
EC Register Bank	0	PECI DISABLE Register		4000FC40
EC Register Bank	0	TEST		4000FC44
EC Register Bank	0	TEST		4000FC48
EC Register Bank	0	TEST		4000FC4C
EC Register Bank	0	VCI FW Override		4000FC50
EC Register Bank	0	TEST		4000FC60
EC Register Bank	0	GPIO Bank Power Register		4000FC64
EC Register Bank	0	TEST		4000FC68
EC Register Bank	0	TEST		4000FC6C
EC Register Bank	0	Vwire FW Override Register		4000FC90
EC Register Bank	0	Analog Comparator Control		4000FC94
EC Register Bank	0	Comparator Sleep Control		4000FC98
EC Register Bank	0	Other IP trim Register		4000FCF0
EC Register Bank	0	TEST		4000FD00
EC Register Bank	0	FW Scratch Register0		4000FD80
EC Register Bank	0	FW Scratch Register1		4000FD84
EC Register Bank	0	FW Scratch Register2		4000FD88
EC Register Bank	0	FW Scratch Register3		4000FD8C
Power Clocks and Resets	0	System Sleep Control Register		40080100
Power Clocks and Resets	0	Processor Clock Control Register		40080104
Power Clocks and Resets	0	Slow Clock Control Register		40080108
Power Clocks and Resets	0	Oscillator ID Register		4008010C
Power Clocks and Resets	0	PCR Power Reset Status Register		40080110
Power Clocks and Resets	0	Power Reset Control Register		40080114
Power Clocks and Resets	0	System Reset Register		40080118
Power Clocks and Resets	0	TEST		4008011C
Power Clocks and Resets	0	TEST		40080120

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
Power Clocks and Resets	0	Sleep Enable 0 Register		40080130
Power Clocks and Resets	0	Sleep Enable 1 Register		40080134
Power Clocks and Resets	0	Sleep Enable 2 Register		40080138
Power Clocks and Resets	0	Sleep Enable 3 Register		4008013C
Power Clocks and Resets	0	Sleep Enable 4 Register		40080140
Power Clocks and Resets	0	Clock Required 0 Register		40080150
Power Clocks and Resets	0	Clock Required 1 Register		40080154
Power Clocks and Resets	0	Clock Required 2 Register		40080158
Power Clocks and Resets	0	Clock Required 3 Register		4008015C
Power Clocks and Resets	0	Clock Required 4 Register		40080160
Power Clocks and Resets	0	Reset Enable 0 Register		40080170
Power Clocks and Resets	0	Reset Enable 1 Register		40080174
Power Clocks and Resets	0	Reset Enable 2 Register		40080178
Power Clocks and Resets	0	Reset Enable 3 Register		4008017C
Power Clocks and Resets	0	Reset Enable 4 Register		40080180
Power Clocks and Resets	0	Peripheral Reset Lock Register		40080184
GPIO	0	GPIO000 Pin Control Register		40081000
GPIO	0	GPIO002 Pin Control Register		40081008
GPIO	0	GPIO003 Pin Control Register		4008100C
GPIO	0	GPIO004 Pin Control Register		40081010
GPIO	0	GPIO007 Pin Control Register		4008101C
GPIO	0	GPIO010 Pin Control Register		40081020
GPIO	0	GPIO011 Pin Control Register		40081024
GPIO	0	GPIO012 Pin Control Register		40081028
GPIO	0	GPIO013 Pin Control Register		4008102C
GPIO	0	GPIO014 Pin Control Register		40081030
GPIO	0	GPIO015 Pin Control Register		40081034
GPIO	0	GPIO016 Pin Control Register		40081038
GPIO	0	GPIO017 Pin Control Register		4008103C
GPIO	0	GPIO020 Pin Control Register		40081040
GPIO	0	GPIO021 Pin Control Register		40081044
GPIO	0	GPIO022 Pin Control Register		40081048
GPIO	0	GPIO023 Pin Control Register		4008104C
GPIO	0	GPIO024 Pin Control Register		40081050
GPIO	0	GPIO025 Pin Control Register		40081054
GPIO	0	GPIO026 Pin Control Register		40081058
GPIO	0	GPIO027 Pin Control Register		4008105C
GPIO	0	GPIO030 Pin Control Register		40081060
GPIO	0	GPIO031 Pin Control Register		40081064
GPIO	0	GPIO032 Pin Control Register		40081068
GPIO	0	GPIO033 Pin Control Register		4008106C
GPIO	0	GPIO034 Pin Control Register		40081070
GPIO	0	GPIO035 Pin Control Register		40081074
GPIO	0	GPIO036 Pin Control Register		40081078

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO040 Pin Control Register		40081080
GPIO	0	GPIO042 Pin Control Register		40081088
GPIO	0	GPIO043 Pin Control Register		4008108C
GPIO	0	GPIO044 Pin Control Register		40081090
GPIO	0	GPIO045 Pin Control Register		40081094
GPIO	0	GPIO046 Pin Control Register		40081098
GPIO	0	GPIO047 Pin Control Register		4008109C
GPIO	0	GPIO050 Pin Control Register		400810A0
GPIO	0	GPIO051 Pin Control Register		400810A4
GPIO	0	GPIO052 Pin Control Register		400810A8
GPIO	0	GPIO053 Pin Control Register		400810AC
GPIO	0	GPIO054 Pin Control Register		400810B0
GPIO	0	GPIO055 Pin Control Register		400810B4
GPIO	0	GPIO056 Pin Control Register		400810B8
GPIO	0	GPIO057 Pin Control Register		400810BC
GPIO	0	GPIO060 Pin Control Register		400810C0
GPIO	0	GPIO061 Pin Control Register		400810C4
GPIO	0	GPIO062 Pin Control Register		400810C8
GPIO	0	GPIO063 Pin Control Register		400810CC
GPIO	0	GPIO064 Pin Control Register		400810D0
GPIO	0	GPIO065 Pin Control Register		400810D4
GPIO	0	GPIO066 Pin Control Register		400810D8
GPIO	0	GPIO067 Pin Control Register		400810DC
GPIO	0	GPIO070 Pin Control Register		400810E0
GPIO	0	GPIO071 Pin Control Register		400810E4
GPIO	0	GPIO072 Pin Control Register		400810E8
GPIO	0	GPIO073 Pin Control Register		400810EC
GPIO	0	GPIO074 Pin Control Register		400810F0
GPIO	0	GPIO075 Pin Control Register		400810F4
GPIO	0	GPIO100 Pin Control Register		40081100
GPIO	0	GPIO101 Pin Control Register		40081104
GPIO	0	GPIO102 Pin Control Register		40081108
GPIO	0	GPIO104 Pin Control Register		40081110
GPIO	0	GPIO105 Pin Control Register		40081114
GPIO	0	GPIO106 Pin Control Register		40081118
GPIO	0	GPIO107 Pin Control Register		4008111C
GPIO	0	GPIO112 Pin Control Register		40081128
GPIO	0	GPIO113 Pin Control Register		4008112C
GPIO	0	GPIO114 Pin Control Register		40081130
GPIO	0	GPIO115 Pin Control Register		40081134
GPIO	0	GPIO116 Pin Control Register		40081138
GPIO	0	GPIO117 Pin Control Register		4008113C
GPIO	0	GPIO120 Pin Control Register		40081140
GPIO	0	GPIO121 Pin Control Register		40081144

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO122 Pin Control Register		40081148
GPIO	0	GPIO123 Pin Control Register		4008114C
GPIO	0	GPIO124 Pin Control Register		40081150
GPIO	0	GPIO125 Pin Control Register		40081154
GPIO	0	GPIO126 Pin Control Register		40081158
GPIO	0	GPIO127 Pin Control Register		4008115C
GPIO	0	GPIO130 Pin Control Register		40081160
GPIO	0	GPIO131 Pin Control Register		40081164
GPIO	0	GPIO132 Pin Control Register		40081168
GPIO	0	GPIO140 Pin Control Register		40081180
GPIO	0	GPIO141 Pin Control Register		40081184
GPIO	0	GPIO142 Pin Control Register		40081188
GPIO	0	GPIO143 Pin Control Register		4008118C
GPIO	0	GPIO144 Pin Control Register		40081190
GPIO	0	GPIO145 Pin Control Register		40081194
GPIO	0	GPIO146 Pin Control Register		40081198
GPIO	0	GPIO147 Pin Control Register		4008119C
GPIO	0	GPIO150 Pin Control Register		400811A0
GPIO	0	GPIO151 Pin Control Register		400811A4
GPIO	0	GPIO152 Pin Control Register		400811A8
GPIO	0	GPIO153 Pin Control Register		400811AC
GPIO	0	GPIO154 Pin Control Register		400811B0
GPIO	0	GPIO155 Pin Control Register		400811B4
GPIO	0	GPIO156 Pin Control Register		400811B8
GPIO	0	GPIO157 Pin Control Register		400811BC
GPIO	0	GPIO161 Pin Control Register		400811C4
GPIO	0	GPIO162 Pin Control Register		400811C8
GPIO	0	GPIO163 Pin Control Register		400811CC
GPIO	0	GPIO165 Pin Control Register		400811D4
GPIO	0	GPIO170 Pin Control Register		400811E0
GPIO	0	GPIO171 Pin Control Register		400811E4
GPIO	0	GPIO172 Pin Control Register		400811E8
GPIO	0	GPIO175 Pin Control Register		400811F4
GPIO	0	GPIO200 Pin Control Register		40081200
GPIO	0	GPIO201 Pin Control Register		40081204
GPIO	0	GPIO202 Pin Control Register		40081208
GPIO	0	GPIO203 Pin Control Register		4008120C
GPIO	0	GPIO204 Pin Control Register		40081210
GPIO	0	GPIO205 Pin Control Register		40081214
GPIO	0	GPIO206 Pin Control Register		40081218
GPIO	0	GPIO207 Pin Control Register		4008121C
GPIO	0	GPIO211 Pin Control Register		40081224
GPIO	0	GPIO212 Pin Control Register		40081228
GPIO	0	GPIO213 Pin Control Register		4008122C

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO221 Pin Control Register		40081244
GPIO	0	GPIO222 Pin Control Register		40081248
GPIO	0	GPIO223 Pin Control Register		4008124C
GPIO	0	GPIO224 Pin Control Register		40081250
GPIO	0	GPIO226 Pin Control Register		40081258
GPIO	0	GPIO227 Pin Control Register		4008125C
GPIO	0	GPIO240 Pin Control Register		40081280
GPIO	0	GPIO241 Pin Control Register		40081284
GPIO	0	GPIO242 Pin Control Register		40081288
GPIO	0	GPIO243 Pin Control Register		4008128C
GPIO	0	GPIO244 Pin Control Register		40081290
GPIO	0	GPIO245 Pin Control Register		40081294
GPIO	0	GPIO246 Pin Control Register		40081298
GPIO	0	GPIO250 Pin Control Register		400812A0
GPIO	0	GPIO253 Pin Control Register		400812AC
GPIO	0	GPIO254 Pin Control Register		400812B0
GPIO	0	GPIO255 Pin Control Register		400812B4
GPIO	0	Input GPIO[000:036]		40081300
GPIO	0	Input GPIO[040:076]		40081304
GPIO	0	Input GPIO[100:127]		40081308
GPIO	0	Input GPIO[140:176]		4008130C
GPIO	0	Input GPIO[200:236]		40081310
GPIO	0	Input GPIO[240:276]		40081314
GPIO	0	Output GPIO[000:036]		40081380
GPIO	0	Output GPIO[040:076]		40081384
GPIO	0	Output GPIO[100:127]		40081388
GPIO	0	Output GPIO[140:176]		4008138C
GPIO	0	Output GPIO[200:236]		40081390
GPIO	0	Output GPIO[240:276]		40081394
GPIO	0	GPIO000 Pin Control2 Register		40081500
GPIO	0	GPIO002 Pin Control2 Register		40081508
GPIO	0	GPIO003 Pin Control2 Register		4008150C
GPIO	0	GPIO004 Pin Control2 Register		40081510
GPIO	0	GPIO007 Pin Control2 Register		4008151C
GPIO	0	GPIO010 Pin Control2 Register		40081520
GPIO	0	GPIO011 Pin Control2 Register		40081524
GPIO	0	GPIO012 Pin Control2 Register		40081528
GPIO	0	GPIO013 Pin Control2 Register		4008152C
GPIO	0	GPIO014 Pin Control2 Register		40081530
GPIO	0	GPIO015 Pin Control2 Register		40081534
GPIO	0	GPIO016 Pin Control2 Register		40081538
GPIO	0	GPIO017 Pin Control2 Register		4008153C
GPIO	0	GPIO020 Pin Control2 Register		40081540
GPIO	0	GPIO021 Pin Control2 Register		40081544

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO022 Pin Control2 Register		40081548
GPIO	0	GPIO025 Pin Control2 Register		40081554
GPIO	0	GPIO026 Pin Control2 Register		40081558
GPIO	0	GPIO027 Pin Control2 Register		4008155C
GPIO	0	GPIO030 Pin Control2 Register		40081560
GPIO	0	GPIO031 Pin Control2 Register		40081564
GPIO	0	GPIO032 Pin Control2 Register		40081568
GPIO	0	GPIO033 Pin Control2 Register		4008156C
GPIO	0	GPIO034 Pin Control2 Register		40081570
GPIO	0	GPIO035 Pin Control2 Register		40081574
GPIO	0	GPIO036 Pin Control2 Register		40081578
GPIO	0	GPIO040 Pin Control2 Register		40081580
GPIO	0	GPIO042 Pin Control2 Register		40081588
GPIO	0	GPIO043 Pin Control2 Register		4008158C
GPIO	0	GPIO044 Pin Control2 Register		40081590
GPIO	0	GPIO045 Pin Control2 Register		40081594
GPIO	0	GPIO046 Pin Control2 Register		40081598
GPIO	0	GPIO047 Pin Control2 Register		4008159C
GPIO	0	GPIO050 Pin Control2 Register		400815A0
GPIO	0	GPIO051 Pin Control2 Register		400815A4
GPIO	0	GPIO052 Pin Control2 Register		400815A8
GPIO	0	GPIO053 Pin Control2 Register		400815AC
GPIO	0	GPIO054 Pin Control2 Register		400815B0
GPIO	0	GPIO055 Pin Control2 Register		400815B4
GPIO	0	GPIO056 Pin Control2 Register		400815B8
GPIO	0	GPIO057 Pin Control2 Register		400815BC
GPIO	0	GPIO060 Pin Control2 Register		400815C0
GPIO	0	GPIO061 Pin Control2 Register		400815C4
GPIO	0	GPIO062 Pin Control2 Register		400815C8
GPIO	0	GPIO063 Pin Control2 Register		400815CC
GPIO	0	GPIO064 Pin Control2 Register		400815D0
GPIO	0	GPIO065 Pin Control2 Register		400815D4
GPIO	0	GPIO066 Pin Control2 Register		400815D8
GPIO	0	GPIO067 Pin Control2 Register		400815DC
GPIO	0	GPIO070 Pin Control2 Register		400815E0
GPIO	0	GPIO071 Pin Control2 Register		400815E4
GPIO	0	GPIO072 Pin Control2 Register		400815E8
GPIO	0	GPIO073 Pin Control2 Register		400815EC
GPIO	0	GPIO074 Pin Control2 Register		400815F0
GPIO	0	GPIO075 Pin Control2 Register		400815F4
GPIO	0	GPIO100 Pin Control2 Register		40081600
GPIO	0	GPIO101 Pin Control2 Register		40081604
GPIO	0	GPIO102 Pin Control2 Register		40081608
GPIO	0	GPIO104 Pin Control2 Register		40081610

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO105 Pin Control2 Register		40081614
GPIO	0	GPIO106 Pin Control2 Register		40081618
GPIO	0	GPIO107 Pin Control2 Register		4008161C
GPIO	0	GPIO112 Pin Control2 Register		40081628
GPIO	0	GPIO113 Pin Control2 Register		4008162C
GPIO	0	GPIO114 Pin Control2 Register		40081630
GPIO	0	GPIO115 Pin Control2 Register		40081634
GPIO	0	GPIO116 Pin Control2 Register		40081638
GPIO	0	GPIO117 Pin Control2 Register		4008163C
GPIO	0	GPIO120 Pin Control2 Register		40081640
GPIO	0	GPIO121 Pin Control2 Register		40081644
GPIO	0	GPIO122 Pin Control2 Register		40081648
GPIO	0	GPIO123 Pin Control2 Register		4008164C
GPIO	0	GPIO124 Pin Control2 Register		40081650
GPIO	0	GPIO125 Pin Control2 Register		40081654
GPIO	0	GPIO126 Pin Control2 Register		40081658
GPIO	0	GPIO127 Pin Control2 Register		4008165C
GPIO	0	GPIO130 Pin Control2 Register		40081660
GPIO	0	GPIO131 Pin Control2 Register		40081664
GPIO	0	GPIO132 Pin Control2 Register		40081668
GPIO	0	GPIO140 Pin Control2 Register		40081680
GPIO	0	GPIO141 Pin Control2 Register		40081684
GPIO	0	GPIO142 Pin Control2 Register		40081688
GPIO	0	GPIO143 Pin Control2 Register		4008168C
GPIO	0	GPIO144 Pin Control2 Register		40081690
GPIO	0	GPIO145 Pin Control2 Register		40081694
GPIO	0	GPIO146 Pin Control2 Register		40081698
GPIO	0	GPIO147 Pin Control2 Register		4008169C
GPIO	0	GPIO150 Pin Control2 Register		400816A0
GPIO	0	GPIO151 Pin Control2 Register		400816A4
GPIO	0	GPIO152 Pin Control2 Register		400816A8
GPIO	0	GPIO153 Pin Control2 Register		400816AC
GPIO	0	GPIO154 Pin Control2 Register		400816B0
GPIO	0	GPIO155 Pin Control2 Register		400816B4
GPIO	0	GPIO156 Pin Control2 Register		400816B8
GPIO	0	GPIO157 Pin Control2 Register		400816BC
GPIO	0	GPIO161 Pin Control2 Register		400816C4
GPIO	0	GPIO162 Pin Control2 Register		400816C8
GPIO	0	GPIO163 Pin Control2 Register		400816CC
GPIO	0	GPIO165 Pin Control2 Register		400816D4
GPIO	0	GPIO170 Pin Control2 Register		400816E0
GPIO	0	GPIO171 Pin Control2 Register		400816E4
GPIO	0	GPIO172 Pin Control2 Register		400816E8
GPIO	0	GPIO175 Pin Control2 Register		400816F4

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
GPIO	0	GPIO200 Pin Control2 Register		40081700
GPIO	0	GPIO201 Pin Control2 Register		40081704
GPIO	0	GPIO202 Pin Control2 Register		40081708
GPIO	0	GPIO203 Pin Control2 Register		4008170C
GPIO	0	GPIO204 Pin Control2 Register		40081710
GPIO	0	GPIO205 Pin Control2 Register		40081714
GPIO	0	GPIO206 Pin Control2 Register		40081718
GPIO	0	GPIO207 Pin Control2 Register		4008171C
GPIO	0	GPIO211 Pin Control2 Register		40081724
GPIO	0	GPIO212 Pin Control2 Register		40081728
GPIO	0	GPIO213 Pin Control2 Register		4008172C
GPIO	0	GPIO221 Pin Control2 Register		40081744
GPIO	0	GPIO222 Pin Control2 Register		40081748
GPIO	0	GPIO223 Pin Control2 Register		4008174C
GPIO	0	GPIO224 Pin Control2 Register		40081750
GPIO	0	GPIO226 Pin Control2 Register		40081758
GPIO	0	GPIO227 Pin Control2 Register		4008175C
GPIO	0	GPIO240 Pin Control2 Register		40081780
GPIO	0	GPIO241 Pin Control2 Register		40081784
GPIO	0	GPIO242 Pin Control2 Register		40081788
GPIO	0	GPIO243 Pin Control2 Register		4008178C
GPIO	0	GPIO244 Pin Control2 Register		40081790
GPIO	0	GPIO245 Pin Control2 Register		40081794
GPIO	0	GPIO246 Pin Control2 Register		40081798
GPIO	0	GPIO250 Pin Control2 Register		400817A0
GPIO	0	GPIO253 Pin Control2 Register		400817AC
GPIO	0	GPIO254 Pin Control2 Register		400817B0
GPIO	0	GPIO255 Pin Control2 Register		400817B4
Mailbox	0	MBX_INDEX Register		400F0000
Mailbox	0	MBX_DATA Register		400F0001
Mailbox	0	HOST-to-EC Mailbox Register		400F0100
Mailbox	0	EC-to-Host Mailbox Register		400F0104
Mailbox	0	SMI Interrupt Source Register		400F0108
Mailbox	0	SMI Interrupt Mask Register		400F010C
Mailbox	0	Mailbox register [3:0]		400F0110
Mailbox	0	Mailbox register [7:4]		400F0114
Mailbox	0	Mailbox register [B:8]		400F0118
Mailbox	0	Mailbox register [F:C]		400F011C
Mailbox	0	Mailbox register [13:10]		400F0120
Mailbox	0	Mailbox register [17:14]		400F0124
Mailbox	0	Mailbox register [1B:18]		400F0128
Mailbox	0	Mailbox register [1F:1C]		400F012C
8042	0	EC_HOST Data / AUX Data Register	Runtime	400F0400
8042	0	Keyboard Status Read Register	Runtime	400F0404

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
8042	0	HOST2EC Data Register		400F0500
8042	0	EC Data Register		400F0500
8042	0	EC Keyboard Status Register		400F0504
8042	0	Keyboard Control Register		400F0508
8042	0	EC AUX Data Register		400F050C
8042	0	PCOBF Register		400F0514
8042	0	Activate Register	Config	400F0730
ACPI EC Channel	0	ACPI OS Data Register Byte 0 Register	Runtime	400F0800
ACPI EC Channel	0	ACPI OS Data Register Byte 1 Register	Runtime	400F0801
ACPI EC Channel	0	ACPI OS Data Register Byte 2 Register	Runtime	400F0802
ACPI EC Channel	0	ACPI OS Data Register Byte 3 Register	Runtime	400F0803
ACPI EC Channel	0	ACPI OS COMMAND Register	Runtime	400F0804
ACPI EC Channel	0	OS STATUS OS Register	Runtime	400F0804
ACPI EC Channel	0	OS Byte Control Register	Runtime	400F0805
ACPI EC Channel	0	Reserved	Runtime	400F0806
ACPI EC Channel	0	Reserved	Runtime	400F0807
ACPI EC Channel	0	EC2OS Data EC Byte 0 Register		400F0900
ACPI EC Channel	0	EC2OS Data EC Byte 1 Register		400F0901
ACPI EC Channel	0	EC2OS Data EC Byte 2 Register		400F0902
ACPI EC Channel	0	EC2OS Data EC Byte 3 Register		400F0903
ACPI EC Channel	0	EC STATUS Register		400F0904
ACPI EC Channel	0	EC Byte Control Register		400F0905
ACPI EC Channel	0	Reserved		400F0906
ACPI EC Channel	0	Reserved		400F0907
ACPI EC Channel	0	OS2EC Data EC Byte 0 Register		400F0908
ACPI EC Channel	0	OS2EC Data EC Byte 1 Register		400F0909
ACPI EC Channel	0	OS2EC Data EC Byte 2 Register		400F090A
ACPI EC Channel	0	OS2EC Data EC Byte 3 Register		400F090B
ACPI EC Channel	1	ACPI OS Data Register Byte 0 Register	Runtime	400F0C00
ACPI EC Channel	1	ACPI OS Data Register Byte 1 Register	Runtime	400F0C01
ACPI EC Channel	1	ACPI OS Data Register Byte 2 Register	Runtime	400F0C02
ACPI EC Channel	1	ACPI OS Data Register Byte 3 Register	Runtime	400F0C03
ACPI EC Channel	1	ACPI OS COMMAND Register	Runtime	400F0C04
ACPI EC Channel	1	OS STATUS OS Register	Runtime	400F0C04
ACPI EC Channel	1	OS Byte Control Register	Runtime	400F0C05
ACPI EC Channel	1	Reserved	Runtime	400F0C06
ACPI EC Channel	1	Reserved	Runtime	400F0C07
ACPI EC Channel	1	EC2OS Data EC Byte 0 Register		400F0D00
ACPI EC Channel	1	EC2OS Data EC Byte 1 Register		400F0D01
ACPI EC Channel	1	EC2OS Data EC Byte 2 Register		400F0D02
ACPI EC Channel	1	EC2OS Data EC Byte 3 Register		400F0D03
ACPI EC Channel	1	EC STATUS Register		400F0D04
ACPI EC Channel	1	EC Byte Control Register		400F0D05
ACPI EC Channel	1	Reserved		400F0D06

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
ACPI EC Channel	1	Reserved		400F0D07
ACPI EC Channel	1	OS2EC Data EC Byte 0 Register		400F0D08
ACPI EC Channel	1	OS2EC Data EC Byte 1 Register		400F0D09
ACPI EC Channel	1	OS2EC Data EC Byte 2 Register		400F0D0A
ACPI EC Channel	1	OS2EC Data EC Byte 3 Register		400F0D0B
ACPI EC Channel	2	ACPI OS Data Register Byte 0 Register	Runtime	400F1000
ACPI EC Channel	2	ACPI OS Data Register Byte 1 Register	Runtime	400F1001
ACPI EC Channel	2	ACPI OS Data Register Byte 2 Register	Runtime	400F1002
ACPI EC Channel	2	ACPI OS Data Register Byte 3 Register	Runtime	400F1003
ACPI EC Channel	2	ACPI OS COMMAND Register	Runtime	400F1004
ACPI EC Channel	2	OS STATUS OS Register	Runtime	400F1004
ACPI EC Channel	2	OS Byte Control Register	Runtime	400F1005
ACPI EC Channel	2	Reserved	Runtime	400F1006
ACPI EC Channel	2	Reserved	Runtime	400F1007
ACPI EC Channel	2	EC2OS Data EC Byte 0 Register		400F1100
ACPI EC Channel	2	EC2OS Data EC Byte 1 Register		400F1101
ACPI EC Channel	2	EC2OS Data EC Byte 2 Register		400F1102
ACPI EC Channel	2	EC2OS Data EC Byte 3 Register		400F1103
ACPI EC Channel	2	EC STATUS Register		400F1104
ACPI EC Channel	2	EC Byte Control Register		400F1105
ACPI EC Channel	2	Reserved		400F1106
ACPI EC Channel	2	Reserved		400F1107
ACPI EC Channel	2	OS2EC Data EC Byte 0 Register		400F1108
ACPI EC Channel	2	OS2EC Data EC Byte 1 Register		400F1109
ACPI EC Channel	2	OS2EC Data EC Byte 2 Register		400F110A
ACPI EC Channel	2	OS2EC Data EC Byte 3 Register		400F110B
ACPI EC Channel	3	ACPI OS Data Register Byte 0 Register	Runtime	400F1400
ACPI EC Channel	3	ACPI OS Data Register Byte 1 Register	Runtime	400F1401
ACPI EC Channel	3	ACPI OS Data Register Byte 2 Register	Runtime	400F1402
ACPI EC Channel	3	ACPI OS Data Register Byte 3 Register	Runtime	400F1403
ACPI EC Channel	3	ACPI OS COMMAND Register	Runtime	400F1404
ACPI EC Channel	3	OS STATUS OS Register	Runtime	400F1404
ACPI EC Channel	3	OS Byte Control Register	Runtime	400F1405
ACPI EC Channel	3	Reserved	Runtime	400F1406
ACPI EC Channel	3	Reserved	Runtime	400F1407
ACPI EC Channel	3	EC2OS Data EC Byte 0 Register		400F1500
ACPI EC Channel	3	EC2OS Data EC Byte 1 Register		400F1501
ACPI EC Channel	3	EC2OS Data EC Byte 2 Register		400F1502
ACPI EC Channel	3	EC2OS Data EC Byte 3 Register		400F1503
ACPI EC Channel	3	EC STATUS Register		400F1504
ACPI EC Channel	3	EC Byte Control Register		400F1505
ACPI EC Channel	3	Reserved		400F1506
ACPI EC Channel	3	Reserved		400F1507
ACPI EC Channel	3	OS2EC Data EC Byte 0 Register		400F1508

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
ACPI EC Channel	3	OS2EC Data EC Byte 1 Register		400F1509
ACPI EC Channel	3	OS2EC Data EC Byte 2 Register		400F150A
ACPI EC Channel	3	OS2EC Data EC Byte 3 Register		400F150B
ACPI PM1	0	Power Management 1 Status 1 Register	Runtime	400F1C00
ACPI PM1	0	Power Management 1 Status 2 Register	Runtime	400F1C01
ACPI PM1	0	Power Management 1 Enable 1 Register	Runtime	400F1C02
ACPI PM1	0	Power Management 1 Enable 2 Register	Runtime	400F1C03
ACPI PM1	0	Power Management 1 Control 1 Register	Runtime	400F1C04
ACPI PM1	0	Power Management 1 Control 2 Register	Runtime	400F1C05
ACPI PM1	0	Power Management 2 Control 1 Register	Runtime	400F1C06
ACPI PM1	0	Power Management 2 Control 2 Register	Runtime	400F1C07
ACPI PM1	0	Power Management 1 Status 1 Register		400F1D00
ACPI PM1	0	Power Management 1 Status 2 Register		400F1D01
ACPI PM1	0	Power Management 1 Enable 1 Register		400F1D02
ACPI PM1	0	Power Management 1 Enable 2 Register		400F1D03
ACPI PM1	0	Power Management 1 Control 1 Register		400F1D04
ACPI PM1	0	Power Management 1 Control 2 Register		400F1D05
ACPI PM1	0	Power Management 2 Control 1 Register		400F1D06
ACPI PM1	0	Power Management 2 Control 2 Register		400F1D07
ACPI PM1	0	EC_PM_STS Register		400F1D10
Port92-Legacy	0	Port 92 Register	Runtime	400F2000
Port92-Legacy	0	GATEA20 Control Register		400F2100
Port92-Legacy	0	SETGA20L Register		400F2108
Port92-Legacy	0	RSTGA20L Register		400F210C
Port92-Legacy	0	Port 92 Enable	Config	400F2330
UART	0	Receive Buffer Register	Runtime	400F2400
UART	0	Transmit Buffer Register	Runtime	400F2400
UART	0	Programmable Baud Rate Generator LSB Register	Runtime	400F2400
UART	0	Programmable Baud Rate Generator MSB Register	Runtime	400F2401
UART	0	Interrupt Enable Register	Runtime	400F2401
UART	0	FIFO Control Register	Runtime	400F2402
UART	0	Interrupt Identification Register	Runtime	400F2402
UART	0	Line Control Register	Runtime	400F2403
UART	0	Modem Control Register	Runtime	400F2404
UART	0	Line Status Register	Runtime	400F2405
UART	0	Modem Status Register	Runtime	400F2406
UART	0	Scratchpad Register	Runtime	400F2407
UART	0	Activate Register	Config	400F2730
UART	0	Configuration Select Register	Config	400F27F0
UART	1	Receive Buffer Register	Runtime	400F2800
UART	1	Transmit Buffer Register	Runtime	400F2800

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
UART	1	Programmable Baud Rate Generator LSB Register	Runtime	400F2800
UART	1	Programmable Baud Rate Generator MSB Register	Runtime	400F2801
UART	1	Interrupt Enable Register	Runtime	400F2801
UART	1	FIFO Control Register	Runtime	400F2802
UART	1	Interrupt Identification Register	Runtime	400F2802
UART	1	Line Control Register	Runtime	400F2803
UART	1	Modem Control Register	Runtime	400F2804
UART	1	Line Status Register	Runtime	400F2805
UART	1	Modem Status Register	Runtime	400F2806
UART	1	Scratchpad Register	Runtime	400F2807
UART	1	Activate Register	Config	400F2B30
UART	1	Configuration Select Register	Config	400F2BF0
UART	2	Receive Buffer Register	Runtime	400F2C00
UART	2	Transmit Buffer Register	Runtime	400F2C00
UART	2	Programmable Baud Rate Generator LSB Register	Runtime	400F2C00
UART	2	Programmable Baud Rate Generator MSB Register	Runtime	400F2C01
UART	2	Interrupt Enable Register	Runtime	400F2C01
UART	2	FIFO Control Register	Runtime	400F2C02
UART	2	Interrupt Identification Register	Runtime	400F2C02
UART	2	Line Control Register	Runtime	400F2C03
UART	2	Modem Control Register	Runtime	400F2C04
UART	2	Line Status Register	Runtime	400F2C05
UART	2	Modem Status Register	Runtime	400F2C06
UART	2	Scratchpad Register	Runtime	400F2C07
UART	2	Activate Register	Config	400F2F30
UART	2	Configuration Select Register	Config	400F2FF0
eSPI IO Component	0	Configuration port Index Register		400F3400
eSPI IO Component	0	Configuration port Data Register		400F3401
eSPI IO Component	0	Peripheral Channel Last Cycle Register		400F3500
eSPI IO Component	0	Peripheral Channel Error Address Register		400F350C
eSPI IO Component	0	Peripheral Channel Status Register		400F3514
eSPI IO Component	0	Peripheral Channel Interrupt Enable Register		400F3518
eSPI IO Component	0	Reserved		400F351C
eSPI IO Component	0	BAR Inhibit Register		400F3520
eSPI IO Component	0	eSPI BAR Init Register		400F3528
eSPI IO Component	0	EC IRQ Register		400F352C
eSPI IO Component	0	TEST		400F3530
eSPI IO Component	0	eSPI IO Component BAR		400F3534
eSPI IO Component	0	eSPI Memory Component BAR		400F3538
eSPI IO Component	0	Mailbox BAR		400F353C

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
eSPI IO Component	0	8042 Emulated Keyboard Controller BAR		400F3540
eSPI IO Component	0	ACIP EC Channel 0 BAR		400F3544
eSPI IO Component	0	ACIP EC Channel 1 BAR		400F3548
eSPI IO Component	0	ACIP EC Channel 2 BAR		400F354C
eSPI IO Component	0	ACIP EC Channel 3 BAR		400F3550
eSPI IO Component	0	ACPI PM1 BAR		400F3558
eSPI IO Component	0	Legacy (Fast Keyboard) BAR		400F355C
eSPI IO Component	0	UART 0 BAR		400F3560
eSPI IO Component	0	UART 1 BAR		400F3564
eSPI IO Component	0	Embedded Memory Interface (EMI) 0 BAR		400F3568
eSPI IO Component	0	Embedded Memory Interface (EMI) 1 BAR		400F356C
eSPI IO Component	0	BIOS Debug Port (Port 80) 0 BAR		400F3574
eSPI IO Component	0	BIOS Debug Port (Port 80) 1 BAR		400F3578
eSPI IO Component	0	RTC BAR		400F357C
eSPI IO Component	0	TEST		400F3584
eSPI IO Component	0	UART 2 BAR		400F3588
eSPI IO Component	0	Glue BAR		400F358C
eSPI IO Component	0	LTR Peripheral Status Register		400F3620
eSPI IO Component	0	LTR Peripheral Enable Register		400F3624
eSPI IO Component	0	LTR Peripheral Control Register		400F3628
eSPI IO Component	0	LTR Peripheral Message Register		400F362C
eSPI IO Component	0	OOB Channel Receive Address Register		400F3640
eSPI IO Component	0	OOB Channel Transmit Address Register		400F3648
eSPI IO Component	0	OOB Channel Receive Length Register		400F3650
eSPI IO Component	0	OOB Channel Transmit Length Register		400F3654
eSPI IO Component	0	OOB Channel Receive Control Register		400F3658
eSPI IO Component	0	OOB Channel Receive Interrupt Enable Register		400F365C
eSPI IO Component	0	OOB Channel Receive Status Register		400F3660
eSPI IO Component	0	OOB Channel Transmit Control Register		400F3664
eSPI IO Component	0	OOB Channel Transmit Interrupt Enable Register		400F3668
eSPI IO Component	0	OOB Channel Transmit Status Register		400F366C
eSPI IO Component	0	Flash Access Channel Flash Address Register		400F3680
eSPI IO Component	0	Flash Access Channel Buffer Address Register		400F3688
eSPI IO Component	0	Flash Access Channel Transfer Length Register		400F3690
eSPI IO Component	0	Flash Access Channel Control Register		400F3694
eSPI IO Component	0	Flash Access Channel Interrupt Enable Register		400F3698
eSPI IO Component	0	Flash Access Channel Configuration Register		400F369C
eSPI IO Component	0	Flash Access Channel Status Register		400F36A0
eSPI IO Component	0	Virtual Wire Status		400F36B0
eSPI IO Component	0	eSPI Capabilities ID Register		400F36E0
eSPI IO Component	0	eSPI Capabilities Global Capabilities 0 Register		400F36E1
eSPI IO Component	0	eSPI Capabilities Global Capabilities 1 Register		400F36E2
eSPI IO Component	0	eSPI Peripheral Channel Capabilities Register		400F36E3

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
eSPI IO Component	0	eSPI Virtual Wire Channel Capabilities Register		400F36E4
eSPI IO Component	0	eSPI OOB Channel Capabilities Register		400F36E5
eSPI IO Component	0	eSPI Flash Channel Capabilities Register		400F36E6
eSPI IO Component	0	eSPI Peripheral Channel Ready Register		400F36E7
eSPI IO Component	0	eSPI OOB Channel Ready Register		400F36E8
eSPI IO Component	0	eSPI Flash Channel Ready Register		400F36E9
eSPI IO Component	0	eSPI Reset Interrupt Status Register		400F36EA
eSPI IO Component	0	eSPI Reset Interrupt Enable Register		400F36EB
eSPI IO Component	0	PLTRST Source Register		400F36EC
eSPI IO Component	0	eSPI Virtual Channel Ready Register		400F36ED
eSPI IO Component	0	eSPI Activate Register	Config	400F3730
eSPI IO Component	0	eSPI IO Component BAR	Config	400F3734
eSPI IO Component	0	eSPI Memory Component BAR	Config	400F3738
eSPI IO Component	0	Mailbox BAR	Config	400F373C
eSPI IO Component	0	8042 Emulated Keyboard Controller BAR	Config	400F3740
eSPI IO Component	0	ACPI EC Channel 0 BAR	Config	400F3744
eSPI IO Component	0	ACPI EC Channel 1 BAR	Config	400F3748
eSPI IO Component	0	ACPI EC Channel 2 BAR	Config	400F374C
eSPI IO Component	0	ACPI EC Channel 3 BAR	Config	400F3750
eSPI IO Component	0	ACPI PM1 BAR	Config	400F3758
eSPI IO Component	0	Legacy (Fast Keyboard) BAR	Config	400F375C
eSPI IO Component	0	UART 0 BAR	Config	400F3760
eSPI IO Component	0	UART 1 BAR	Config	400F3764
eSPI IO Component	0	Embedded Memory Interface (EMI) 0 BAR	Config	400F3768
eSPI IO Component	0	Embedded Memory Interface (EMI) 1 BAR	Config	400F376C
eSPI IO Component	0	BIOS Debug Port (Port 80) 0 BAR	Config	400F3774
eSPI IO Component	0	BIOS Debug Port (Port 80) 1 BAR	Config	400F3778
eSPI IO Component	0	RTC BAR	Config	400F377C
eSPI IO Component	0	32 Byte Test Block BAR	Config	400F3784
eSPI IO Component	0	UART 2 BAR	Config	400F3788
eSPI IO Component	0	Glue BAR	Config	400F378C
eSPI IO Component	0	Mailbox SERIRQ 0	Config	400F37AC
eSPI IO Component	0	Mailbox SERIRQ 1	Config	400F37AD
eSPI IO Component	0	8042 SERIRQ 0	Config	400F37AE
eSPI IO Component	0	8042 SERIRQ 1	Config	400F37AF
eSPI IO Component	0	ACPI EC 0 SERIRQ	Config	400F37B0
eSPI IO Component	0	ACPI EC 1 SERIRQ	Config	400F37B1
eSPI IO Component	0	ACPI EC 2 SERIRQ	Config	400F37B2
eSPI IO Component	0	ACPI EC 3 SERIRQ	Config	400F37B3
eSPI IO Component	0	UART 0 SERIRQ	Config	400F37B5
eSPI IO Component	0	UART 1 SERIRQ	Config	400F37B6
eSPI IO Component	0	EMI 0 SERIRQ 0	Config	400F37B7
eSPI IO Component	0	EMI 0 SERIRQ 1	Config	400F37B8
eSPI IO Component	0	EMI 1 SERIRQ 0	Config	400F37B9

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
eSPI IO Component	0	EMI 1 SERIRQ 1	Config	400F37BA
eSPI IO Component	0	RTC SERIRQ	Config	400F37BD
eSPI IO Component	0	EC SERIRQ	Config	400F37BE
eSPI IO Component	0	UART 2 SERIRQ	Config	400F37BF
eSPI IO Component	0	eSPI Virtual Wire Error	Config	400F37F0
eSPI Memory Component	0	Mailbox BAR		400F3930
eSPI Memory Component	0	ACIP EC Channel 0 BAR		400F393A
eSPI Memory Component	0	ACIP EC Channel 1 BAR		400F3944
eSPI Memory Component	0	ACIP EC Channel 2 BAR		400F394E
eSPI Memory Component	0	ACIP EC Channel 3 BAR		400F3958
eSPI Memory Component	0	Embedded Memory Interface (EMI) 0 BAR		400F396C
eSPI Memory Component	0	Embedded Memory Interface (EMI) 1 BAR		400F3976
eSPI Memory Component	0	TEST		400F398A
eSPI Memory Component	0	SRAM BAR 0		400F39AC
eSPI Memory Component	0	SRAM BAR 1		400F39B6
eSPI Memory Component	0	Bus Master Status Register		400F3A00
eSPI Memory Component	0	Bus Master Interrupt Enable Register		400F3A04
eSPI Memory Component	0	Bus Master Configuration Register		400F3A08
eSPI Memory Component	0	Bus Master 1 Control Register		400F3A10
eSPI Memory Component	0	Bus Master 1 Host Address Register		400F3A14
eSPI Memory Component	0	Bus Master 1 Internal Address Register		400F3A1C
eSPI Memory Component	0	Bus Master 2 Control Register		400F3A24
eSPI Memory Component	0	Bus Master 2 Host Address Register		400F3A28
eSPI Memory Component	0	Bus Master 2 Internal Address Register		400F3A30
eSPI Memory Component	0	Mailbox BAR	Config	400F3B30
eSPI Memory Component	0	ACIP EC Channel 0 BAR	Config	400F3B3A
eSPI Memory Component	0	ACIP EC Channel 1 BAR	Config	400F3B44
eSPI Memory Component	0	ACIP EC Channel 2 BAR	Config	400F3B4E
eSPI Memory Component	0	ACIP EC Channel 3 BAR	Config	400F3B58
eSPI Memory Component	0	Embedded Memory Interface (EMI) 0 BAR	Config	400F3B6C
eSPI Memory Component	0	Embedded Memory Interface (EMI) 1 BAR	Config	400F3B76
eSPI Memory Component	0	TEST	Config	400F3B8A
eSPI Memory Component	0	SRAM BAR 0	Config	400F3BAC
eSPI Memory Component	0	SRAM BAR 1	Config	400F3BB6
eSPI SAF Bridge Component	0	Test		40008000
eSPI SAF Bridge Component	0	SAF EC Portal Command Register		40008018
eSPI SAF Bridge Component	0	SAF EC Portal Flash Address Register		4000801C
eSPI SAF Bridge Component	0	SAF EC Portal Start Register		40008020
eSPI SAF Bridge Component	0	SAF EC Portal Buffer Address Register		40008024
eSPI SAF Bridge Component	0	SAF EC Portal Status Register		40008028
eSPI SAF Bridge Component	0	SAF EC Portal Interrupt Enable Register		4000802C
eSPI SAF Bridge Component	0	SAF Flash Configuration Size Limit Register		40008030
eSPI SAF Bridge Component	0	SAF Flash Configuration Threshold Register		40008034
eSPI SAF Bridge Component	0	SAF Flash Configuration Misc Register		40008038

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
eSPI SAF Bridge Component	0	SAF eSPI Monitor Status Register		4000803C
eSPI SAF Bridge Component	0	SAF eSPI Monitor Interrupt Enable Register		40008040
eSPI SAF Bridge Component	0	SAF EC Busy Register		40008044
eSPI SAF Bridge Component	0	TEST		40008048
eSPI SAF Bridge Component	0	CS0 Opcode:SAF Flash Configuration Opcode Register A		4000804C
eSPI SAF Bridge Component	0	CS0 Opcode:SAF Flash Configuration Opcode Register B		40008050
eSPI SAF Bridge Component	0	CS0 opcode:SAF Flash Configuration Opcode Register C		40008054
eSPI SAF Bridge Component	0	CS0 Opcode;SAF Flash Configuration Per-Flash Descriptors Register		40008058
eSPI SAF Bridge Component	0	CS1 Opcode:SAF Flash Configuration Opcode Register A		4000805C
eSPI SAF Bridge Component	0	CS1 Opcode:SAF Flash Configuration Opcode Register B		40008060
eSPI SAF Bridge Component	0	CS1 opcode:SAF Flash Configuration Opcode Register C		40008064
eSPI SAF Bridge Component	0	CS1 Opcode;SAF Flash Configuration Per-Flash Descriptors Register		40008068
eSPI SAF Bridge Component	0	SAF Flash Configuration General Descriptors Register		4000806C
eSPI SAF Bridge Component	0	SAF Protection Lock Bit Register		40008070
eSPI SAF Bridge Component	0	SAF Protection Dirty Bit Register		40008074
eSPI SAF Bridge Component	0	SAF Tag Map Register 0		40008078
eSPI SAF Bridge Component	0	SAF Tag Map Register 1		4000807C
eSPI SAF Bridge Component	0	SAF Tag Map Register 2		40008080
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008084
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008088
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000808C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008090
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008094
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008098
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000809C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080A0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080A4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080A8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080AC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080B0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080B4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080B8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080BC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080C0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080C4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080C8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080CC

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080D0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080D4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080D8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080DC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080E0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080E4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080E8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080EC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		400080F0
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		400080F4
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		400080F8
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		400080FC
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008100
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008104
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008108
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000810C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008110
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008114
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008118
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000811C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008120
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008124
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008128
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000812C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008130
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008134
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008138
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000813C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008140
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008144
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008148
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000814C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008150
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008154
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008158
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000815C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008160
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008164
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008168
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000816C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008170
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008174
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008178
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000817C

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008180
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Start Register		40008184
eSPI SAF Bridge Component	0	SAF Protection Region [RR] Limit Register		40008188
eSPI SAF Bridge Component	0	SAF Write Protection Bitmap [RR] Register		4000818C
eSPI SAF Bridge Component	0	SAF Read Protection Bitmap [RR] Register		40008190
eSPI SAF Bridge Component	0	SAF Poll Timeout Register		40008194
eSPI SAF Bridge Component	0	SAF Poll Interval Register		40008198
eSPI SAF Bridge Component	0	SAF Suspend/Resume Interval Register		4000819C
eSPI SAF Bridge Component	0	SAF Consecutive Read Timeout Register		400081A0
eSPI SAF Bridge Component	0	SAF Flash Configuration Poll2 Mask Register		400081A4
eSPI SAF Bridge Component	0	SAF Flash Configuration Special Mode Register		400081A8
eSPI SAF Bridge Component	0	SAF Suspend Check Delay Register		400081AC
eSPI SAF Bridge Component	0	SAF Flash Configuration Special Mode Register		400081B0
eSPI SAF Bridge Component	0	SAF DnX Protection Bypass		400081B4
eSPI SAF Communication Registers	0	SAF Communication Mode Register		400712B8
EMI	0	HOST-to-EC Mailbox Register	Runtime	400F4000
EMI	0	EC-to-HOST Mailbox Register	Runtime	400F4001
EMI	0	EC Address LSB Register	Runtime	400F4002
EMI	0	EC Address MSB Register	Runtime	400F4003
EMI	0	EC Data Byte 0 Register	Runtime	400F4004
EMI	0	EC Data Byte 1 Register	Runtime	400F4005
EMI	0	EC Data Byte 2 Register	Runtime	400F4006
EMI	0	EC Data Byte 3 Register	Runtime	400F4007
EMI	0	Interrupt Source LSB Register	Runtime	400F4008
EMI	0	Interrupt Source MSB Register	Runtime	400F4009
EMI	0	Interrupt Mask LSB Register	Runtime	400F400A
EMI	0	Interrupt Mask MSB Register	Runtime	400F400B
EMI	0	Application ID Register	Runtime	400F400C
EMI	0	HOST-to-EC Mailbox Register		400F4100
EMI	0	EC-to-HOST Mailbox Register		400F4101
EMI	0	Memory Base Address 0 Register		400F4104
EMI	0	Memory Read Limit 0 Register		400F4108
EMI	0	Memory Write Limit 0 Register		400F410A
EMI	0	Memory Base Address 1 Register		400F410C
EMI	0	Memory Read Limit 1 Register		400F4110
EMI	0	Memory Write Limit 1 Register		400F4112
EMI	0	Interrupt Set Register		400F4114
EMI	0	Host Clear Enable Register		400F4116
EMI	1	HOST-to-EC Mailbox Register	Runtime	400F4400
EMI	1	EC-to-HOST Mailbox Register	Runtime	400F4401
EMI	1	EC Address LSB Register	Runtime	400F4402
EMI	1	EC Address MSB Register	Runtime	400F4403
EMI	1	EC Data Byte 0 Register	Runtime	400F4404

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
EMI	1	EC Data Byte 1 Register	Runtime	400F4405
EMI	1	EC Data Byte 2 Register	Runtime	400F4406
EMI	1	EC Data Byte 3 Register	Runtime	400F4407
EMI	1	Interrupt Source LSB Register	Runtime	400F4408
EMI	1	Interrupt Source MSB Register	Runtime	400F4409
EMI	1	Interrupt Mask LSB Register	Runtime	400F440A
EMI	1	Interrupt Mask MSB Register	Runtime	400F440B
EMI	1	Application ID Register	Runtime	400F440C
EMI	1	HOST-to-EC Mailbox Register		400F4500
EMI	1	EC-to-HOST Mailbox Register		400F4501
EMI	1	Memory Base Address 0 Register		400F4504
EMI	1	Memory Read Limit 0 Register		400F4508
EMI	1	Memory Write Limit 0 Register		400F450A
EMI	1	Memory Base Address 1 Register		400F450C
EMI	1	Memory Read Limit 1 Register		400F4510
EMI	1	Memory Write Limit 1 Register		400F4512
EMI	1	Interrupt Set Register		400F4514
EMI	1	Host Clear Enable Register		400F4516
Real Time Clock	0	Seconds Register	Runtime	400F5000
Real Time Clock	0	Seconds Alarm Register	Runtime	400F5001
Real Time Clock	0	Minutes Register	Runtime	400F5002
Real Time Clock	0	Minutes Alarm Register	Runtime	400F5003
Real Time Clock	0	Hours Register	Runtime	400F5004
Real Time Clock	0	Hours Alarm Register	Runtime	400F5005
Real Time Clock	0	Day of Week Register	Runtime	400F5006
Real Time Clock	0	Day of Month Register	Runtime	400F5007
Real Time Clock	0	Month Register	Runtime	400F5008
Real Time Clock	0	Year Register	Runtime	400F5009
Real Time Clock	0	Register A	Runtime	400F500A
Real Time Clock	0	Register B	Runtime	400F500B
Real Time Clock	0	Register C	Runtime	400F500C
Real Time Clock	0	Register D	Runtime	400F500D
Real Time Clock	0	Reserved	Runtime	400F500E
Real Time Clock	0	Reserved	Runtime	400F500F
Real Time Clock	0	RTC Control Register	Runtime	400F5010
Real Time Clock	0	Week Alarm Register	Runtime	400F5014
Real Time Clock	0	Daylight Savings Forward Register	Runtime	400F5018
Real Time Clock	0	Daylight Savings Backward Register	Runtime	400F501C
Real Time Clock	0	TEST	Runtime	400F5020
Port 80	0	Host Data Register	Runtime	400F8000
Port 80	0	EC Data Register		400F8100
Port 80	0	Configuration Register		400F8104
Port 80	0	Status Register		400F8108
Port 80	0	Count Register		400F810C

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
Port 80	0	Activate Register	Config	400F8330
Port 80	1	Host Data Register	Runtime	400F8400
Port 80	1	EC Data Register		400F8500
Port 80	1	Configuration Register		400F8504
Port 80	1	Status Register		400F8508
Port 80	1	Count Register		400F850C
Port 80	1	Activate Register	Config	400F8730
eSPI Virtual Wires	0	MSVW00 Register	Runtime	400F9C00
eSPI Virtual Wires	0	MSVW01 Register	Runtime	400F9C0C
eSPI Virtual Wires	0	MSVW02 Register	Runtime	400F9C18
eSPI Virtual Wires	0	MSVW03 Register	Runtime	400F9C24
eSPI Virtual Wires	0	MSVW04 Register	Runtime	400F9C30
eSPI Virtual Wires	0	MSVW05 Register	Runtime	400F9C3C
eSPI Virtual Wires	0	MSVW06 Register	Runtime	400F9C48
eSPI Virtual Wires	0	MSVW07 Register	Runtime	400F9C54
eSPI Virtual Wires	0	MSVW08 Register	Runtime	400F9C60
eSPI Virtual Wires	0	MSVW09 Register	Runtime	400F9C6C
eSPI Virtual Wires	0	MSVW10 Register	Runtime	400F9C78
eSPI Virtual Wires	0	SMVW00 Register		400F9E00
eSPI Virtual Wires	0	SMVW01 Register		400F9E08
eSPI Virtual Wires	0	SMVW02 Register		400F9E10
eSPI Virtual Wires	0	SMVW03 Register		400F9E18
eSPI Virtual Wires	0	SMVW04 Register		400F9E20
eSPI Virtual Wires	0	SMVW05 Register		400F9E28
eSPI Virtual Wires	0	SMVW06 Register		400F9E30
eSPI Virtual Wires	0	SMVW07 Register		400F9E38
eSPI Virtual Wires	0	SMVW08 Register		400F9E40
eSPI Virtual Wires	0	SMVW09 Register		400F9E49
eSPI Virtual Wires	0	SMVW10 Register		400F9E50
eSPI Virtual Wires	0	TEST	Config	400F9FF0
eSPI Virtual Wires	0	TEST	Config	400F9FF2
eSPI Virtual Wires	0	TEST	Config	400F9FF8
eSPI Virtual Wires	0	TEST	Config	400F9FFA
Global Configuration	0	Global Configuration Reserved	Runtime	400FFF00
Global Configuration	0	Control	Runtime	400FFF02
Global Configuration	0	Logical Device Number	Runtime	400FFF07
Global Configuration	0	Device Revision	Runtime	400FFF1C
Global Configuration	0	Device Sub ID	Runtime	400FFF1D
Global Configuration	0	Device ID[7:0]	Runtime	400FFF1E
Global Configuration	0	Device ID[15:0]	Runtime	400FFF1F
Global Configuration	0	Legacy Device ID	Runtime	400FFF20
Global Configuration	0	TEST	Runtime	400FFF28
Global Configuration	0	TEST	Runtime	400FFF29
Global Configuration	0	Test0	Runtime	400FFF2A

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
Global Configuration	0	Test1	Runtime	400FFF2B
Global Configuration	0	TEST	Runtime	400FFF2C
Global Configuration	0	TEST	Runtime	400FFF2D
Global Configuration	0	TEST	Runtime	400FFF2E
Global Configuration	0	TEST	Runtime	400FFF2F
ARM M4	0	Auxiliary_Control		DFFFE008
ARM M4	0	SystemTick_Ctrl_Status		DFFFE010
ARM M4	0	SystemTick_Reload_Value		DFFFE014
ARM M4	0	SystemTick_Current_Value		DFFFE018
ARM M4	0	SystemTick_Calibration_Value		DFFFE01C
ARM M4	0	CPU_ID		DFFFED00
ARM M4	0	Interrupt_Ctl_and_State		DFFFED04
ARM M4	0	Vector_Table_Offset		DFFFED08
ARM M4	0	Application_Interrupt_and_Reset_Ctl		DFFFED0C
ARM M4	0	System_Ctl		DFFFED10
ARM M4	0	Config_and_Ctl		DFFFED14
ARM M4	0	System_Handler_Priority1		DFFFED18
ARM M4	0	System_Handler_Priority2		DFFFED1C
ARM M4	0	System_Handler_Priority3		DFFFED20
ARM M4	0	System_Handler_Ctl_and_State		DFFFED24
ARM M4	0	Configurable_Fault_Status		DFFFED28
ARM M4	0	Hard_Fault_Status		DFFFED2C
ARM M4	0	Debug_Fault_Status		DFFFED30
ARM M4	0	Debug_Halting_Ctl_and_Status		DFFFEDF0
ARM M4	0	Debug_Core_Register_Selector		DFFFEDF4
ARM M4	0	Debug_Core_Register_Data		DFFFEDF8
ARM M4	0	Debug_Exception_and_Monitor_Ctl		DFFFEDFC
ARM M4	0	Bus_Fault_Address		DFFFED38
ARM M4	0	Auxiliary_Fault_Status		DFFFED3C
ARM M4	0	Processor_Feature0		DFFFED40
ARM M4	0	Processor_Feature1		DFFFED44
ARM M4	0	Debug_Features0		DFFFED48
ARM M4	0	Auxiliary_Features0		DFFFED4C
ARM M4	0	Memory_Model_Feature0		DFFFED50
ARM M4	0	Memory_Model_Feature1		DFFFED54
ARM M4	0	Memory_Model_Feature2		DFFFED58
ARM M4	0	Memory_Model_Feature3		DFFFED5C
ARM M4	0	Instruction_Set_Attributes0		DFFFED60
ARM M4	0	Instruction_Set_Attributes1		DFFFED64
ARM M4	0	Instruction_Set_Attributes2		DFFFED68
ARM M4	0	Instruction_Set_Attributes3		DFFFED6C
ARM M4	0	Instruction_Set_Attributes4		DFFFED70
ARM M4	0	Coprocessor_Access_Ctl		DFFFED88
ARM M4	0	Software_Triggered_Interrupt		DFFFEF00

© 2020 Microchip Technology Inc.

TABLE 3-5: REGISTER MAP (CONTINUED)

Block	Instance	Register	Host Type	Register Address
SPI Slave	0	SPI Communication Configuration Register		40007000
SPI Slave	0	SPI Slave Status Register		40007004
SPI Slave	0	SPI EC Status Register		40007008
SPI Slave	0	SPI Interrupt Enable Register		4000700C
SPI Slave	0	EC Interrupt Enable Register		40007010
SPI Slave	0	Memory Configuration Register		40007014
SPI Slave	0	Memory Base Address0 Register		40007018
SPI Slave	0	Memory Write Limit0 Register		4000701C
SPI Slave	0	Memory Read Limit0 Register		40007020
SPI Slave	0	Memory Base Address1 Register		40007024
SPI Slave	0	Memory Write Limit1 Register		40007028
SPI Slave	0	Memory Read Limit1 Register		4000702C
SPI Slave	0	RX FIFO Host BAR		40007030
SPI Slave	0	RX FIFO Byte CNTR		40007034
SPI Slave	0	TX FIFO Host BAR		40007038
SPI Slave	0	RX FIFO Byte CNTR		4000703C
SPI Slave	0	System Configuration Register		40007040
SPI Slave	0	SPI Master-to-EC Mailbox Register		40007044
SPI Slave	0	EC-to-SPI Master Mailbox Register		40007048
SPI Slave	0	Test Modes Register		4000704C

4.0 POWER, CLOCKS, AND RESETS

4.1 Introduction

The Power, Clocks, and Resets (PCR) chapter identifies all the power supplies, clock sources, and reset inputs to the chip and defines all the derived power, clock, and reset signals. In addition, this section identifies Power, Clock, and Reset events that may be used to generate an interrupt event, as well as, the Chip Power Management Features.

4.2 References

No references have been cited for this chapter.

4.3 Interrupts

The Power, Clocks, and Resets logic generates no events

4.4 Power

TABLE 4-1: POWER SOURCE DEFINITIONS

Power Well	Nominal Voltage	Description	Source
VTR_REG	1.8V - 3.3V	This supply is used to derive the chip's core power.	Pin Interface
VTR_ANALOG	3.3V	3.3V Analog Power Supply.	Pin Interface For the 128 pin package VTR_ANALOG source is VTR1
VTR1_ADC	3.3V	3.3V Power Supply for ADC and Bandgap.	Pin Interface (Only for 128 VTQFP)
VTR_PLL	3.3V	3.3V Power Supply for the 48MHz PLL. This must be connected to the same supply as VTR_ANALOG.	Pin Interface
VTR1	3.3V	3.3V System Power Supply. This is typically connected to the "Always-on" or "Suspend" supply rails in system. This supply must be on prior to the system RSMRST# signal being deasserted	Pin Interface
VTR2	3.3V or 1.8V	3.3V or 1.8V System Power Supply. This supply is used to power one bank of I/O pins. See Note 1.	Pin Interface
VTR3	3.3V or 1.8V	3.3V or 1.8V System Power Supply. This supply is used to power one bank of I/O pins. See Note 1.	Pin Interface
VTR_CORE	1.2V	The main power well for internal logic	Internal regulator
VBAT	3.0V - 3.3V	System Battery Back-up Power Well. This is the "coin-cell" battery. GPIOs that share pins with VBAT signals are powered by this supply.	Pin Interface VBAT
VSS	0V	Digital Ground	Pin Interface

Note 1: See Section 4.4.1, "I/O Rail Requirements" for connection requirements for VTRx.

2: The source for the Internal regulator is VTR_REG.

3: VTR refers to VTR_REG and VTR_ANALOG.

© 2020 Microchip Technology Inc.

4.4.1 I/O RAIL REQUIREMENTS

All pins are powered by the power supply pins: VBAT, VTR1, VTR2and VTR3. The VBAT supply must be 3V to 3.6V maximum, as shown in the following section. The VTR1 is fixed 3.3V and VTR2 and VTR3 pins may be connected to either a 3.3V or a 1.8V power supply as configured by the firmware.

If a power rail is not powered and stable when RESET_SYS is de-asserted and is not required for booting, software can configure the pins on that bank appropriately by setting the corresponding bit in the GPIO Bank Power Register, once software can determine that the power supply is up and stable. All GPIOs in the bank must be left in their default state and not modified until the Bank Power is configured properly.

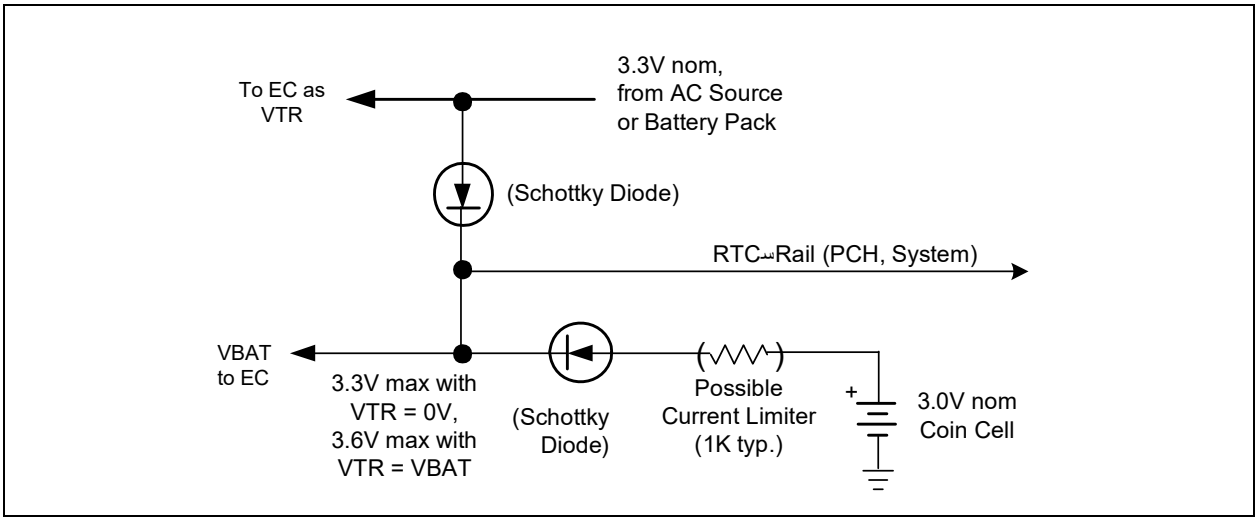
4.4.2 BATTERY CIRCUIT REQUIREMENTS

VBAT must always be present if VTR_ANALOG is present.

Microchip recommends removing all power sources to the device defined in Table 4-1, "Power Source Definitions" and all external voltage references defined in Table 4-2, "Voltage Reference Definitions" before removing and replacing the battery. In addition, upon removing the battery, discharge the battery pin before replacing the battery.

The following external circuit is recommended to fulfill this requirement:

FIGURE 4-1: RECOMMENDED BATTERY CIRCUIT



4.4.3 VOLTAGE REFERENCES

Table 4-2 lists the External Voltage References to which the MEC152x provides high impedance interfaces.

TABLE 4-2: VOLTAGE REFERENCE DEFINITIONS

Power Well	Nominal Input Voltage	Scaling Ratio	Nominal Monitored Voltage	Description	Source
VREF_VTT	Variable	n/a	Variable	Processor Voltage External Voltage Reference Used to scale Processor Interface signals. (See Note)	Pin Interface
VREF_ADC	Variable	n/a	Variable	ADC Reference Voltage	Pin Interface
CMP_VREF0	Variable	n/a	Variable	Determines reference voltage on the negative terminal of Comparator 0	Pin Interface
Note: In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_VTT Disable bit to 1. This bit is defined in PECI Disable Register bit 0					

TABLE 4-2: VOLTAGE REFERENCE DEFINITIONS (CONTINUED)

Power Well	Nominal Input Voltage	Scaling Ratio	Nominal Monitored Voltage	Description	Source
CMP_VREF1	Variable	n/a	Variable	Determines reference voltage on the negative terminal of Comparator 1	Pin Interface
Note: In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_VTT Disable bit to 1. This bit is defined in PECI Disable Register bit 0					

4.4.4 POWER GOOD SIGNALS

The power good timing and thresholds are defined in the Section 50.8, "VCC_PWRGD Timing".

TABLE 4-3: POWER GOOD SIGNAL DEFINITIONS

Power Good Signal	Description	Source
VCC_PWRGD	VCC_PWRGD is an input signal used to indicate when the main system power rail voltage is on and stable.	VCC_PWRGD Input pin
VTR_PWRGD	VTR_PWRGD is an internal power good signal used to indicate whether the VTR_CORE rail is good and stable.	VTR_PWRGD is asserted following a delay after the VTR_CORE power well exceeds its preset voltage threshold.
PWROK	PWROK is an output signal used to indicate that the main system power rail voltage is on and the Host may access Host devices in the EC.	The PWROK pin is asserted high whenever both the VCC_PWRGD input is asserted high and the PWR_INV bit in the Power Reset Control Register is de-asserted low.
PWRGD_STRAP	PWRGD_STRAP is an input signal that is asserted when the four Primary power rails in an Intel system (VCC_Prim 1.8V, VCC_Prim 3.3V, VCCPRIM_CORE and VCC_Prim 1.0V) are up. It is only used by Boot ROM code when booting over the eSPI Flash Channel.	GPIO227/SHD_IO2 pin. There is no special hardware associated with this signal

4.4.5 SYSTEM POWER SEQUENCING

The following table defines the behavior of the main power rails in each of the defined ACPI power states.

TABLE 4-4: TYPICAL POWER SUPPLIES VS. ACPI POWER STATES

Supply Name	ACPI Power State						Description
	S0 (FULL ON)	S1 (POS)	S3 (STR)	S4 (STD)	S5 (Soft Off)	G3 (MECH Off)	
VTR3	ON	ON	ON/OFF	ON/OFF	ON/OFF	OFF	3.3V/1.8V Power Supply for Bank 3
VTR1	ON	ON	ON	ON	ON	OFF	“Always-on” Supply
VTR2	ON	ON	ON	ON	ON	OFF	3.3V/1.8V Power Supply for Bank 2
VBAT	ON	ON	ON	ON Note	ON Note	ON Note	Battery Back-up Supply
Note: This device requires that the VBAT power is on when the VTR(Note 3) power supply is on. External circuitry, a diode isolation circuit, is implemented on the motherboard to extend the battery life. This external circuitry ensures the VBAT pin will derive power from the VTR power well when it is on. Therefore, the VBAT supply will never appear to be off when the VTR rail is on.							

© 2020 Microchip Technology Inc.

4.5 Clocks

The following section defines the clocks that are generated and derived.

4.5.1 RAW CLOCK SOURCES

The table defines raw clocks that are either generated externally or via an internal oscillator .

TABLE 4-5: SOURCE CLOCK DEFINITIONS

Clock Name	Frequency	Description	Source
32KHZ_IN	32.768 kHz (nominal)	Single-ended external clock input pin	32KHZ_IN pin
32.768 kHz Crystal Oscillator	32.768 kHz	A 32.768 kHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins. The accuracy of the clock depends on the accuracy of the crystal and the characteristics of the analog components used as part of the oscillator The crystal oscillator source can bypass the crystal with a single-ended clock input. This option is configured with the Clock Enable Register.	Pin Interface (XTAL1 and XTAL2) When used singled-ended, pin XTAL2
32.768 kHz Silicon Oscillator	32.768 kHz	32.768 kHz low power Internal Oscillator. The frequency is 32.768KHz ±2%.	Internal Oscillator powered by VBAT.
32 MHz Ring Oscillator	32MHz	The 32MHz Ring Oscillator is used to supply a clock for the 48MHz main clock domain while the 48MHz PLL is not locked. Its frequency can range from 12Mhz to 46MHz.	Powered by VTR_CORE.
48 MHz PLL	48MHz	The 48 MHz Phase Locked Loop generates a 48MHz clock locked to the VBAT 32KHz Clock	Powered by VTR_CORE. May be stopped by Chip Power Management Features.
eSPI Clock	20MHz to 50MHz	eSPI bus clock This clock is only used in the eSPI interface.	ESPI_CLK pin

4.5.2 CLOCK DOMAINS

TABLE 4-6: CLOCK DOMAIN DEFINITIONS

Clock Domain	Description
VBAT 32KHz Clock	The clock source used as reference for PLL lock and System Clock controls.
32KHz	The clock source used by internal blocks that require an always-on low speed clock
48MHz	The main clock source used by most internal blocks
2MHz	Internally generated 2 MHz clock from 48MHz clock.
100KHz	A low-speed clock derived from the 48MHz clock domain. Used as a time base for PWMs and Tachs.
EC_CLK	The clock used by the EC processor. The frequency is determined by the Processor Clock Control Register.

4.5.3 48MHZ PLL

The 48MHz clock domain is primarily driven by a 48MHz PLL, which derives 48MHz from the VBAT 32KHz Clock domain. In Heavy Sleep mode, the 48MHz PLL is shut off. When the PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset, the 32MHz ring oscillator becomes the clock source for the 48MHz clock domain until the PLL is stable. The PLL becomes stable after about 3ms after the VBAT 32KHz Clock is stable; until that time, the 48MHz clock domain may range from 16MHz to 48MHz, as this is the accuracy range of the 32MHz ring.

The PLL requires its own power 3.3V power supply, VTR_PLL. This power rail must be active and stable no later than the latest of VTR_REG and VTR_ANALOG. There is no hardware detection of VTR_PLL power good in the reset generator.

4.5.4 32KHZ CLOCK

The 32kHz Clock Domain may be sourced by a crystal oscillator, using an external crystal, by an internal 32kHz oscillator, or from a single-ended clock input. The external single-ended clock source can itself be sourced either from the 32KHZ_IN signal that is a GPIO alternate function or from the XTAL2 crystal pin. The Clock Enable Register is used to configure the source for the 32 kHz clock domain.

Internal 32khz Silicon Oscillator will be defaulted to OFF state. When VTR is on SUSCLK from PCH will be source for the 32kHz clock domain. When VTR goes down, HW will automatically switch to the 32.768 kHz Silicon Oscillator.Application code can enable the internal 32.768 kHz Silicon Oscillator if required.

When VTR_CORE is off, the 32 kHz clock domain can be disabled, for lowest standby power, or it can be kept running in order to provide a clock for the Real Time Clock or the Week Timer.

An external single-ended clock input for 32KHZ_IN may be supplied by any accurate 32KHz clock source in the system. The SUSCLK output from the chipset may be used as the 32KHz source whenever RSMRST# is de-asserted. SUSCLK must be present when VTR is on. See chipset documentation for details on the use of SUSCLK.

If firmware switches the 32KHz clock source, the 48MHz PLL will be shut off and then restarted. The 48MHz clock domain will become unlocked and be sourced from the 32 MHz Ring Oscillator until the 48MHz PLL is on and locked.

4.5.4.1 VBAT 32KHz Clock

This clock source is used to drive the 48MHz PLL. VBAT 32KHz Clock should remain on while the 48Mhz PLL is ON. The internal source provides a reference for the Activity Detect that monitors the external clock input, as well as providing a low latency backup clock source when the Activity Detector cannot detect a clock on the external input.

The VBAT 32KHz Clock Internal Clock Source can be driven either by the 32.768 kHz Silicon Oscillator or the 32.768 kHz Crystal Oscillator.

Note:	If the 32KHZ_SOURCE field in the Clock Enable Register selects the crystal oscillator as the source for the VBAT 32KHz Clock source, and the XOSEL field selects a single-ended input for the crystal oscillator, the system must ensure that the single-ended input remains on at all times. The Activity Detector will not monitor the single-ended input to the crystal oscillator.

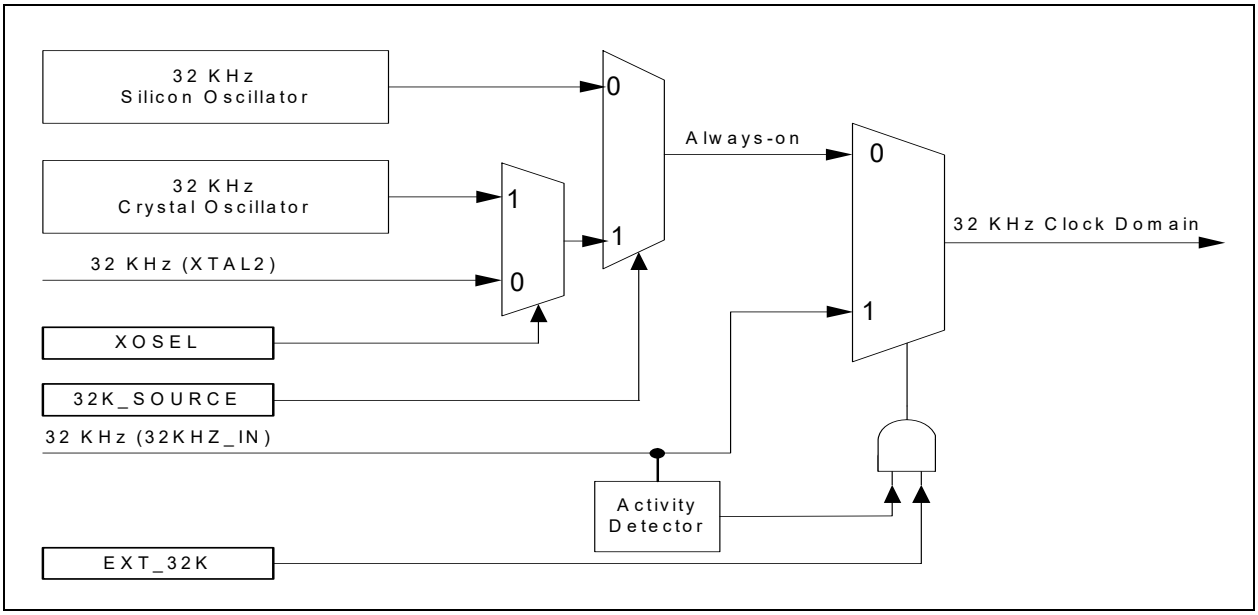
4.5.4.2 External 32KHz Clock Activity Detector

When the EXT_32K field in the Clock Enable Register is set for an external clock source an Activity Detector monitors the external 32KHz signal at all times. If there is no clock detected on the pin, the 32KHZ clock domain is switched to the internal 32KHz silicon oscillator. If a clock is again detected on the pin, the 32KHz clock domain is switched to the pin

The following figure illustrates the 32KHz clock domain sourcing.

© 2020 Microchip Technology Inc.

FIGURE 4-2: 32KHZ ACTIVITY DETECTOR



Note:	Once the internal 32KHz clock domain switches to an external single-ended clock source, the external source **must** remain active until VTR_CORE power is removed, or internal clocking may not function correctly.

4.5.4.3 32KHz Crystal Oscillator

If the 32KHz source will never be the crystal oscillator, then the XTAL2 pin should be grounded. The XTAL1 pin should be left unconnected.

4.6 Resets

TABLE 4-7: DEFINITION OF RESET SIGNALS

Reset	Description	Source
RESET_VBAT	Internal VBAT Reset signal. This signal is used to reset VBAT powered registers.	RESET_VBAT is a pulse that is asserted at the rising edge of VTR power if the VBAT voltage is below a nominal 1.25V. RESET_VBAT is also asserted as a level if, while VTR power is not present, the coin cell is replaced with a new cell that delivers at least a nominal 1.25V. In this latter case RESET_VBAT is de-asserted when VTR power is applied. No action is taken if the coin cell is replaced, or if the VBAT voltage falls below 1.25 V nominal, while VTR power is present.

TABLE 4-7: DEFINITION OF RESET SIGNALS (CONTINUED)

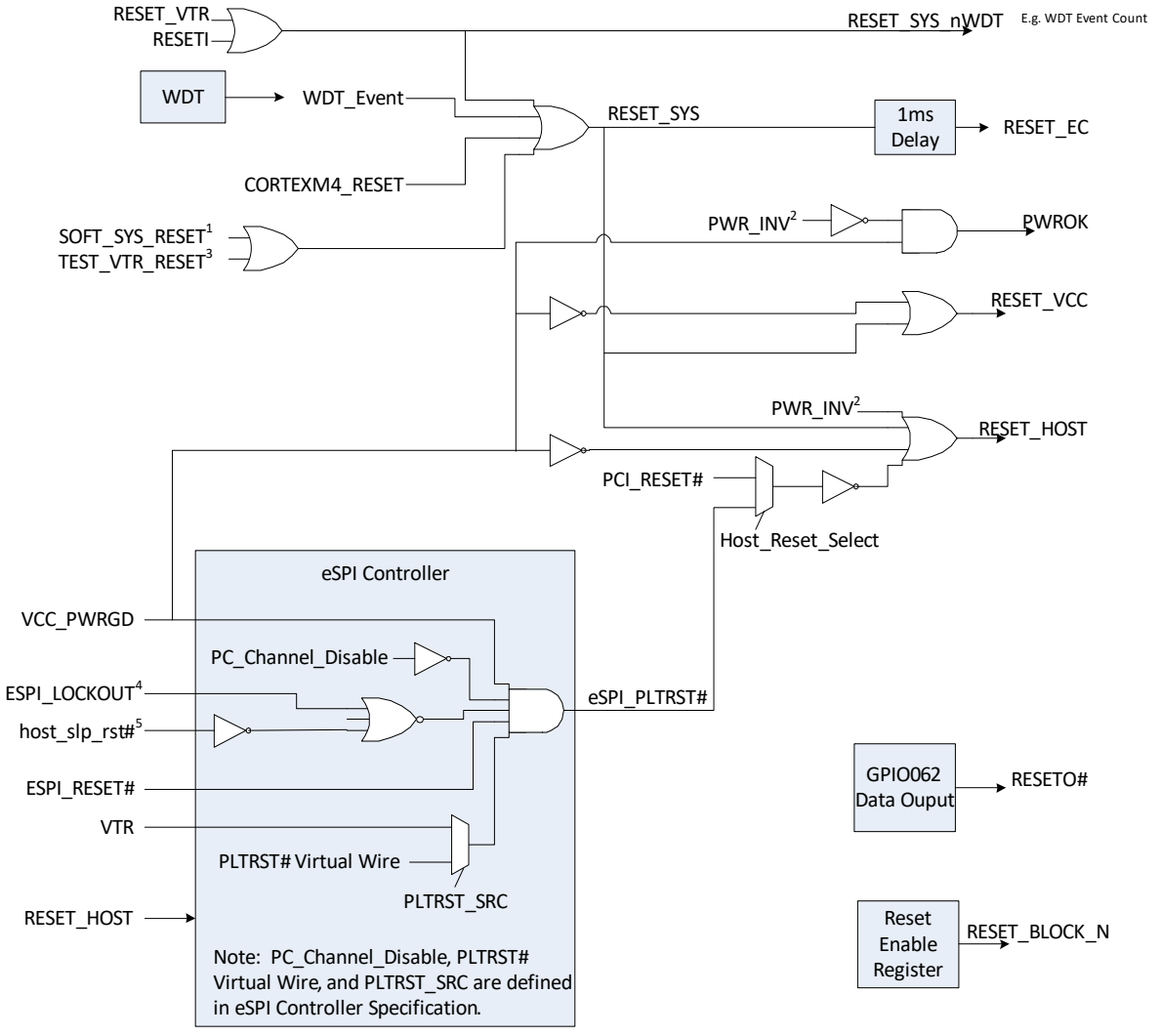
Reset	Description	Source
RESET_VTR	Internal VTR Reset signal.	This internal reset signal is asserted as long as the reset generator determines that the output of the internal regulator is stable at its target voltage and that the voltage rail supplying the main clock PLL is at 3.3V. Although most VTR_CORE-powered registers are reset on RESET_SYS, some registers are only reset on this reset.
RESET_SYS	Internal Reset signal. This signal is used to reset VTR_CORE powered registers.	RESET_SYS is the main global reset signal. This reset signal will be asserted if: • RESET_VTR is asserted • The nRESET_IN pin asserted • A WDT Event event is asserted • A soft reset is asserted by the SOFT_SYS_RESET bit in the System Reset Register • ARM M4 SYSRESETREQ
RESET_eSPI	System reset signal connected to the eSPI ESPI_RESET# pin.	Pin Interface, ESPI_RESET# pin.
RESET_VCC	Performs a reset when Host power (VCC) is turned off	This signal is asserted if • RESET_SYS is asserted • VCC_PWRGD is low The PWROK output pin is an inverted version of this reset; it is asserted when VCC_PWRGD is high and the PWR_INV bit is '0b'. **Note:** This reset is referred to as RESET_SIO in the eSPI Block Specification.
RESET_HOST	Performs a reset when VCC_PWRGD is low or when the system host resets the Host Interface.	This signal is asserted if • RESET_SYS is asserted • VCC_PWRGD is low • The PWR_INV bit in the Power Reset Control Register is '1b' • The HOST_RESET_SELECT bit in the Power Reset Control Register is configured for PCI_RESET# and the PCI_RESET# signal is asserted • The HOST_RESET_SELECT bit in the Power Reset Control Register is configured for eSPI_PLTSRT# and the eSPI_PLTRST# signal from the eSPI block is asserted.
RESET_HOST	Performs a reset when VCC_PWRGD is low	This signal is asserted if • RESET_SYS is asserted • VCC_PWRGD is low • The PWR_INV bit in the Power Reset Control Register is '1b'1

© 2020 Microchip Technology Inc.

TABLE 4-7: DEFINITION OF RESET SIGNALS (CONTINUED)

Reset	Description	Source
WDT Event	A WDT Event generates the RESET_SYS event. This signal resets VTR_CORE powered registers with the exception of the WDT Event Count Register register. Note that the glitch protect circuits do not activate on a WDT reset. WDT Event does not reset VBAT registers or logic.	This reset signal will be asserted if: • A WDT Event event is asserted This event is indicated by the WDT bit in the Power-Fail and Reset Status Register
RESET_SYS_nWDT	Internal Reset signal. This signal is used to reset VTR_CORE powered registers not effected by a WDT Event A RESET_SYS_nWDT is used to reset registers that need to be preserved through a WDT Event like a WDT Event Count Register.	This reset signal will be asserted if: • RESET_VTR is asserted • The nRESET_IN pin asserted
RESET_EC	Internal reset signal to reset the processor in the EC Subsystem.	This reset is a stretched version of RESET_SYS. This reset asserts at the same time that RESET_SYS asserts and is held asserted for 1ms after RESET_SYS deasserts.
RESET_BLOCK_N	Each IP block in the device may be configured to be reset by setting the RESET_ENABLE register.	This reset signal will be asserted if Block N RESET_ENABLE is set to 1 and Peripheral Reset Enable n Register is unlocked.

FIGURE 4-3: RESETS BLOCK DIAGRAM



Note 1: SOFT_SYS_RESET is implemented in bit[8] of the System Reset Register
Note 2: PWR_INV is implemented in bit[0] of the Power Reset Control Register
Note 3: TEST_VTR_RESET is implemented in bit [1] of the Host Global Test Register
Note 4: ESPI_LOCKOUT is implemented in bit[1] of the FEATURE LOCKOUT Register
Note 5: host_slp_rst# is asserted if the Host Sleep Enable bit and Host Reset Enable bit are set and the eSPI Controller is put to sleep
Note 6: RESET_HOST is fed back into the eSPI Controller to reset select registers/bits

© 2020 Microchip Technology Inc.

4.7 Chip Power Management Features

This device is designed to always operate in its lowest power state during normal operation. In addition, this device offers additional programmable options to put individual logical blocks to sleep as defined in the following section, Section 4.7.1.

4.7.1 BLOCK LOW POWER MODES

All power related control signals are generated and monitored centrally in the chip's Power, Clocks, and Resets (PCR) block. The power manager of the PCR block uses a sleep interface to communicate with all the blocks. The sleep interface consists of three signals:

- **SLEEP_ENABLE (request to sleep the block)** is generated by the PCR block. A group of SLEEP_ENABLE signals are generated for every clock segment. Each group consists of a SLEEP_ENABLE signal for every block in that clock segment.
- **CLOCK_REQUIRED (request clock on)** is generated by every block. They are grouped by blocks on the same clock segment. The PCR monitors these signals to see when it can gate off clocks.

A block can always drive CLOCK_REQUIRED low synchronously, but it *must* drive it high asynchronously since its internal clocks are gated and it has to assume that the clock input itself is gated. Therefore the block can only drive CLOCK_REQUIRED high as a result of a register access or some other input signal.

The following table defines a block's power management protocol:

TABLE 4-8: POWER MANAGEMENT PROTOCOL

Power State	SLEEP_ENABLE	CLOCK_REQUIRED	Description
Normal operation	Low	Low	Block is idle and NOT requesting clocks. The block gates its own internal clock.
Normal operation	Low	High	Block is NOT idle and requests clocks.
Request sleep	Rising Edge	Low	Block is IDLE and enters sleep mode immediately. The block gates its own internal clock. The block cannot request clocks again until SLEEP_ENABLE goes low.
Request sleep	Rising Edge	High then Low	Block is not IDLE and will stop requesting clocks and enter sleep when it finishes what it is doing. This delay is block specific, but should be less than 1 ms. The block gates its own internal clock. After driving CLOCK_REQUIRED low, the block cannot request clocks again until SLEEP_ENABLE goes low.
Register Access	X	High	Register access to a block is always available regardless of SLEEP_ENABLE. Therefore the block ungates its internal clock and drives CLOCK_REQUIRED high during the access. The block will regate its internal clock and drive CLOCK_REQUIRED low when the access is done.

A wake event clears all SLEEP_ENABLE bits momentarily, and then returns the SLEEP_ENABLE bits back to their original state. The block that needs to respond to the wake event will do so.

The Sleep Enable, Clock Required and Reset Enable Registers are defined in Section 4.8.

4.7.2 CONFIGURING THE CHIP'S SLEEP STATES

The chip supports two sleep states: LIGHT SLEEP and HEAVY SLEEP. The chip will enter one of these two sleep states only when all the blocks have been commanded to sleep and none of them require a 48MHz clock source (i.e., all CLOCK_REQUIRED status bits are 0), and the processor has executed its sleep instruction. These sleep states must be selected by firmware via the System Sleep Control bits implemented in the System Sleep Control Register prior to issuing the sleep instruction. Table 4-10, "System Sleep Modes" defines each of these sleep states.

There are two ways to command the chip blocks to enter sleep.

1. Assert the SLEEP_ALL bit located in the System Sleep Control Register
2. Assert all the individual block sleep enable bits

Blocks will only enter sleep after their sleep signal is asserted and they no longer require the 48MHz source. Each block has a corresponding clock required status bit indicating when the block has entered sleep. The general operation is that a block will keep the 48MHz clock source on until it completes its current transaction. Once the block has completed its work, it deasserts its clock required signal. Blocks like timers, PWMs, etc. will de-assert their clock required signals immediately. See the individual block Low Power Mode sections to determine how each individual block enters sleep.

4.7.3 DETERMINING WHEN THE CHIP IS SLEEPING

The TST_CLK_OUT pin can be used to verify the chip's clock has stopped, which indicates the device is in LIGHT SLEEP or HEAVY SLEEP, as determined by the System Sleep Control Register. If the clock is toggling the chip is in the full on running state. if the clock is not toggling the chip has entered the programmed sleep state.

4.7.4 WAKING THE CHIP FROM SLEEPING STATE

The chip will remain in the configured sleep state until it detects either a wake event or a full VTR_CORE POR. A wake event occurs when a wake-capable interrupt is enabled and triggered. Interrupts that are not wake-capable cannot occur while the system is in LIGHT SLEEP or HEAVY SLEEP.

In LIGHT SLEEP, the 48MHz clock domain is gated off, but the 48 MHz PLL remains operational and locked to the 32KHz clock domain. On wake, the PLL output is ungated and the 48MHz clock domain starts immediately, with the PLL_LOCK bit in the Oscillator ID Register set to '1'. Any device that requires an accurate clock, such as a UART, may be used immediately on wake.

In HEAVY SLEEP, the 48 MHz PLL is shut down. On wake, the 32 MHz Ring Oscillator is used to provide a clock source for the 48MHz clock domain until the PLL locks to the 32KHz clock domain. The ring oscillator starts immediately on wake, so there is no latency for the EC to start after a wake, However, the ring oscillator is only accurate to ±50%, so any device that requires an accurate 48MHz clock will not operate correctly until the PLL locks.The time to lock latency for the PLL is shown in Table 4-10, "System Sleep Modes".

The SLEEP_ALL bit is automatically cleared when the processor responds to an interrupt. This applies to non-wake interrupts as well as wake interrupts, in the event an interrupt occurs between the time the processor issued a WAIT FOR INTERRUPT instruction and the time the system completely enters the sleep state.

Any JTAG access to the ARM/STAP will cause a pseudo-wake event where the clocks are turned on, but the CHip is still in sleep (SLEEP_EN's and SLEEP_ALL stay in the same state).This way the access can occur over JTAG, without changing the parts state, and the part can go back to sleep once the JTAG access is over.

4.7.4.1 Wake-Only Events

Some devices which respond to an external master require the 48MHz clock domain to operate but do not necessarily require and immediate processing by the EC. Wake-only events provide the means to start the 48MHz clock domain without triggering an EC interrupt service routine. This events are grouped into a single GIRQ, GIRQ22. Events that are enabled in that GIRQ will start the clock domain when the event occurs, but will not invoke an EC interrupt. The SLEEP_ENABLE flags all remain asserted. If the activity for the event does not in turn trigger another EC interrupt, the CLOCK_REQUIRED for the block will re-assert and the configured sleep state will be re-entered.

Note:	For example, when RSMRST is high and there is a desire to wake from an ESPI cycle, GIRQ22[9] is the correct wake source to use. When Chip is asleep and there is a ESPI cycle, the falling edge of the CS will cause the chips clock to turn on the ESPI block, but not the processor itself. Upon conclusion of the ESPI cycle, if no ESPI interrupt was generated (i.e. most cycles), then the clock to the ESPI block will go off, and the chip will go back to sleep. If the ESPI cycle creates an interrupt to the processor (i.e. downstream wire or downstream OOB packet for example), then an processor interrupt will be generated if enabled and the clock will remain on and the processor can service the interrupt and the processor can put the chip back to sleep when it has completed its work.

© 2020 Microchip Technology Inc.

4.8 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the Power, Clocks, and Resets Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 4-9: REGISTER SUMMARY

Offset	Name
0h	System Sleep Control Register
4h	Processor Clock Control Register
8h	Slow Clock Control Register
Ch	Oscillator ID Register
10h	PCR Power Reset Status Register
14h	Power Reset Control Register
18h	System Reset Register
1Ch	Reserved
20h	TEST
30h	Sleep Enable 0 Register
34h	Sleep Enable 1 Register
38h	Sleep Enable 2 Register
3Ch	Sleep Enable 3 Register
40h	Sleep Enable 4 Register
50h	Clock Required 0 Register
54h	Clock Required 1 Register
58h	Clock Required 2 Register
5Ch	Clock Required 3 Register
60h	Clock Required 4 Register
70h	Reset Enable 0 Register
74h	Reset Enable 1 Register
78h	Reset Enable 2 Register
7Ch	Reset Enable 3 Register
80h	Reset Enable 4 Register
84h	Peripheral Reset Lock Register

Note: All register addresses are naturally aligned on 32-bit boundaries. Offsets for registers that are smaller than 32 bits are reserved and must not be used for any other purpose.

The bit definitions for the Sleep Enable, Clock Required and Reset Enable Registers are defined in the Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory".

4.9 Sleep Enable *n* Registers

4.9.1 SLEEP ENABLE *N* REGISTER

Offset	See Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	SLEEP_ENABLE 1=Block is commanded to sleep at next available moment 0=Block is free to use clocks as necessary Unassigned bits are reserved. They must be set to '1b' when written. When read, unassigned bits return the last value written.	R/W	0h	RESET_SYS

4.9.2 CLOCK REQUIRED *N* REGISTER

Offset	See Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CLOCK_REQUIRED 1=Bock requires clocks 0=Block does not require clocks Unassigned bits are reserved and always return 0 when read.	R	0h	RESET_SYS

4.9.3 PERIPHERAL RESET ENABLE *N* REGISTER

Offset	See Sleep Enable Register Assignments Table in Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PERIPHERAL_RESET_ENABLE 1= Will allow issue parallel reset to the peripherals. This is self clearing bit.	W	0h	RESET_SYS

4.9.4 SYSTEM SLEEP CONTROL REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	SLEEP_IMMEDIATE 0 = System will only allow entry into sleep after PLL locks. 1 = System will allow entry into Heavy Sleep before PLL locks. Heavy Sleep : Any sleep state where the PLL is OFF. Light Sleep : Any sleep state where the PLL is ON.	R/W	0h	RESET_SYS
7:4	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	SLEEP_ALL By setting this bit to '1b' and then issuing a WAIT FOR INTERRUPT instruction, the EC can initiate the System Sleep mode. When no device requires the main system clock, the system enters the sleep mode defined by the field SLEEP_MODE. This bit is automatically cleared when the processor vectors to an interrupt. 1=Assert all sleep enables 0=Do not sleep all	R/W	0h	RESET_SYS
2	TEST Test bit. Should always be written with a '0b'.	R/W	0h	RESET_SYS
1	Reserved	RES	-	-
0	SLEEP_MODE Sleep modes differ only in the time it takes for the 48MHz clock domain to lock to 48MHz. The wake latency in all sleep modes is 0ms. Table 4-10 shows the time to lock latency for the different sleep modes. 1=Heavy Sleep 0=Light Sleep	R/W	0h	RESET_SYS

TABLE 4-10: SYSTEM SLEEP MODES

SLEEP_MODE	Sleep State	Latency to Lock	Description
0	LIGHT SLEEP	0	Output of the PLL is gated in sleep. The PLL remains on.
1	HEAVY SLEEP	3ms	The PLL is shut down while in sleep.

4.9.5 PROCESSOR CLOCK CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	PROCESSOR_CLOCK_DIVIDE The following list shows examples of settings for this field and the resulting EC clock rate. 48=divide the 48MHz clock by 48 (1MHz processor clock) 16=divide the 48MHz clock by 16 (4MHz processor clock) 4=divide the 48MHz clock by 4 (12MHz processor clock) 3=divide the 48MHz clock by 3 (16MHz processor clock) 1=divide the 48MHz clock by 1 (48MHz processor clock) No other values are supported.	R/W	4h	RESET_SYS

4.9.6 SLOW CLOCK CONTROL REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	RES	-	-
9:0	SLOW_CLOCK_DIVIDE Configures the 100KHz clock domain. n=Divide by n 0=Clock off The default setting is for 100KHz.	R/W	1E0h	RESET_SYS

4.9.7 OSCILLATOR ID REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	PLL_LOCK Phase Lock Loop Lock Status	R	0h	RESET_SYS
7:0	TEST	R	N/A	RESET_SYS

4.9.8 PCR POWER RESET STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:12	Reserved	RES	-	-
11	ESPI_CLK_ACTIVE This bit monitors the state of the eSPI clock input. This status bit detects edges on the clock input but does not validate the frequency. 1=The eSPI clock is present. 0=The eSPI clock input is not present.	R	-	RESET_SYS
10	32K_ACTIVE 1=The 32K clock input is present. The internal 32K clock is derived from the pin and the ring oscillator is synchronized to the external 32K clock 0=The 32K clock input is not present. The internal 32K clock is derived from the ring oscillator	R	-	RESET_SYS
9	Reserved	RES	-	-
8	WDT_EVENT This bit allows the application code to determine WDT_EVENT against RESET_VTR	R/W1C	0h	RESET_SYS_n-WDT
Note 1:	This read-only status bit always reflects the current status of the event and is not affected by any Reset events.			

© 2020 Microchip Technology Inc.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	JTAG_RST# Indicates the JTAG_RST# pin status. The JTAG TRST# input is gated off low when Boundary scan mode is enabled and will not be set in this mode.	R	-	RESET_SYS
6	RESET_SYS_STATUS Indicates the status of RESET_SYS. The bit will not clear if a write 1 is attempted at the same time that a RESET_VTR occurs; this way a reset event is never missed. 1=A reset occurred 0=No reset occurred since the last time this bit was cleared	R/WC	1h	RESET_SYS
5	VBAT_RESET_STATUS Indicates the status of RESET_VBAT. The bit will not clear if a write of '1'b is attempted at the same time that a VBAT_RST_N occurs, this way a reset event is never missed. 1=A reset occurred 0=No reset occurred while VTR_CORE was off or since the last time this bit was cleared	R/WC	-	RESET_SYS
4	RESET_VTR_STATUS Indicates the status of RESET_VTR event.	R/W1C	1h	RESET_VTR
3	RESET_HOST_STATUS Indicates the status of RESET_HOST. 1=Reset not active 0=Reset active	R	-	Note 1
2	VCC_PWRGD_STATUS Indicates the status of VCC_PWRGD. 1=VCC_PWRGD asserted 0=VCC_PWRGD not asserted	R	xh	Note 1
1:0	Reserved	RES	-	-
Note 1:	This read-only status bit always reflects the current status of the event and is not affected by any Reset events.			

4.9.9 POWER RESET CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	HOST_RESET_SELECT This bit determines the platform reset signal. It should be set to '0b' if the eSPI interface is in use. 1=The PCI_RESET# pin is used to generate the internal Platform Reset 0=The eSPI_PLTRST# signal from the eSPI block is used to generate the internal Platform Reset	R/W	1h	RESET_SYS
7:1	Reserved	RES	-	-
0	PWR_INV This bit allows firmware to control when the Host receives an indication that the VCC power is valid, by controlling the state of the PWROK pin.	R / R/W	1h	RESET_SYS

4.9.10 SYSTEM RESET REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	SOFT_SYS_RESET A write of a '1' to this bit will force an assertion of the RESET_SYS reset signal, resetting the device. A write of a '0' has no effect. Reads always return '0'.	W	-	-
7:0	Reserved	RES	-	-

4.9.11 PERIPHERAL RESET LOCK REGISTER

Offset	84h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PCR_RST_EN _LOCK If the lock is enabled, the peripherals cannot be reset by writing to the Reset enable register. Once Unlocked the Registers remain in the unlocked state until FW re-locks it with the Lock pattern 0xA6382D4Dh = Lock Pattern 0xA6382D4Ch = Unlock Pattern	RW	A6382D4Dh	RESET_SYS

© 2020 Microchip Technology Inc.

5.0 ARM M4 BASED EMBEDDED CONTROLLER

5.1 Introduction

This chapter contains a description of the ARM M4 Embedded Controller (EC).

The EC is built around an ARM® Cortex®-M4 Processor provided by Arm Ltd. (the “ARM M4 IP”). The ARM Cortex® M4 is a full-featured 32-bit embedded processor, implementing the ARMv7-M THUMB instruction set in hardware.

The ARM M4 IP is configured as a Von Neumann, Byte-Addressable, Little-Endian architecture. It provides a single unified 32-bit byte-level address, for a total direct addressing space of 4GByte. It has multiple bus interfaces, but these express priorities of access to the chip-level resources (Instruction Fetch vs. Data RAM vs. others), and they do not represent separate addressing spaces.

The ARM M4 is configured as follows.

- **Little-Endian** byte ordering is selected at all times
- **Bit Banding** is included for efficient bit-level access
- **Debug** features are included at “Ex+” level, defined as follows:
  - **DWT** Unit provides 4 Data Watchpoint comparators and Execution Monitoring
- **Trace** features are included at “Full” level, defined as follows:
  - **DWT** for reporting breakpoints and watchpoints
  - **ITM** for profiling and to timestamp and output messages from instrumented firmware builds
  - **ETM** for instruction tracing, and for enhanced reporting of Core and DWT events
  - The ARM-defined **HTM** trace feature is **not included**
- **NVIC** Interrupt controller with 8 priority levels and up to 240 individually-vectored interrupt inputs
  - A Microchip-defined Interrupt Aggregator function (at chip level) may be used to group multiple interrupts onto single NVIC inputs
  - The ARM-defined **WIC** feature is **not included**. The Microchip Interrupt Aggregator function (at chip level) provides Wake control
- Single entry **Write Buffer** is incorporated

5.2 References

1. ARM Limited: Cortex®-M4 Technical Reference Manual, DDI0439C, 29 June 2010
2. ARM Limited: ARM®v7-M Architecture Reference Manual, DDI0403D, November 2010
3. NOTE: Filename DDI0403D_arm_architecture_v7m_reference_manual_errata_markup_1_0.pdf
4. ARM® Generic Interrupt Controller Architecture version 1.0 Architecture Specification, IHI0048A, September 2008
5. ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999
6. ARM Limited: AMBA® 3 AHB-Lite Protocol Specification, IHI0033A, 6 June 2006
7. ARM Limited: AMBA® 3 ATB Protocol Specification, IHI0032A, 19 June 2006
8. ARM Limited: Cortex-M™ System Design Kit Technical Reference Manual, DDI0479B, 16 June 2011
9. ARM Limited: CoreSight™ v1.0 Architecture Specification, IHI0029B, 24 March 2005
10. ARM Limited: CoreSight™ Components Technical Reference Manual, DDI0314H, 10 July 2009
11. ARM Limited: ARM® Debug Interface v5 Architecture Specification, IHI0031A, 8 February 2006
12. ARM Limited: ARM® Debug Interface v5 Architecture Specification ADIv5.1 Supplement, DSA09-PRDC-008772, 17 August 2009
13. ARM Limited: Embedded Trace Macrocell™ (ETMv1.0 to ETMv3.5) Architecture Specification, IHI0014Q, 23 September 2011
14. ARM Limited: CoreSight™ ETM™-M4 Technical Reference Manual, DDI0440C, 29 June 2010

5.3 Terminology

5.3.1 ARM IP TERMS AND ACRONYMS

- AHB

  Advanced High-Performance Bus, a system-level on-chip **AMBA 2** bus standard. See Reference[5], ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999.

- AHB-AP

  AHB Access Port, the **AP** option selected by Microchip for the **DAP**

- AHB-Lite

  A Single-Master subset of the **AHB** bus standard: defined in the **AMBA 3** bus standard. See Reference[6], ARM Limited: AMBA® 3 AHB-Lite Protocol Specification, IHI0033A, 6 June 2006.

- AMBA

  The collective term for bus standards originated by ARM Limited.

  AMBA 3 defines the IP's **AHB-Lite** and **ATB** bus interfaces.

  AMBA 2 (AMBA Rev. 2.0) defines the EC's **AHB** bus interface.

- AP

  Any of the ports on the **DAP** subblock for accessing on-chip resources on behalf of the Debugger, independent of processor operations. A single **AHB-AP** option is currently selected for this function.

- APB

  Advanced Peripheral Bus, a limited 32-bit-only bus defined in **AMBA 2** for I/O register accesses. This term is relevant only to describe the **PPB** bus internal to the EC core. See Reference [5], ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999.

- ARMv7

  The identifying name for the general architecture implemented by the **Cortex-M** family of IP products.

  The **ARMv7** architecture has no relationship to the older "ARM 7" product line, which is classified as an "ARMv3" architecture, and is very different.

- ATB

  Interface standard for Trace data to the **TPIU** from **ETM** and/or **ITM** blocks, Defined in **AMBA 3**. See Reference[7], ARM Limited: AMBA® 3 ATB Protocol Specification, IHI0032A, 19 June 2006.

- Cortex-M4

  The ARM designation for the specific IP selected for this product: a Cortex M4 processor core

- DAP

  Debug Access Port, a subblock consisting of **DP** and **AP** subblocks.

- DP

  Any of the ports in the **DAP** subblock for connection to an off-chip Debugger. A single **SWJ-DP** option is currently selected for this function, providing **JTAG** connectivity.

- DWT

  Data Watchdog and Trace subblock. This contains comparators and counters used for data watchpoints and Core activity tracing.

- ETM

  Embedded Trace Macrocell subblock. Provides enhancements for Trace output reporting, mostly from the **DWT** subblock. It adds enhanced instruction tracing, filtering, triggering and timestamping.

- FPB

  FLASH Patch Breakpoint subblock. Provides either Remapping (Address substitution) or Breakpointing (Exception or Halt) for a set of Instruction addresses and Data addresses. See Section 8.3 of Reference [1], ARM Limited: Cortex®-M4 Technical Reference Manual, DDI0439C, 29 June 2010.

© 2020 Microchip Technology Inc.

- HTM

  AHB Trace Macrocell. This is an optional subblock that is **not included**.

- ITM

  Instrumentation Trace Macrocell subblock. Provides a HW Trace interface for “printf”-style reports from instrumented firmware builds, with timestamping also provided.

- MEM-AP

  A generic term for an **AP** that connects to a memory-mapped bus on-chip. For this product, this term is synonymous with the AHB Access Port, **AHB-AP.**

- NVIC

  Nested Vectored Interrupt Controller subblock. Accepts external interrupt inputs. See References [2], ARM Limited: ARM®v7-M Architecture Reference Manual, DDI0403D, November 2010 and [4], ARM® Generic Interrupt Controller Architecture version 1.0 Architecture Specification, IHI0048A, September 2008.

- PPB

  Private Peripheral Bus: A specific **APB** bus with local connectivity within the EC.

- ROM Table

  A ROM-based data structure in the Debug section that allows an external Debugger and/or a FW monitor to determine which of the Debug features are present.

- SWJ-DP

  Serial Wire / **JTAG** Debug Port, the **DP** option selected by Microchip for the **DAP**.

- TPA

  Trace Port Analyzer: any off-chip device that uses the TPIU output.

- TPIU

  Trace Port Interface Unit subblock. Multiplexes and buffers Trace reports from the ETM and ITM subblocks.

- WIC

  Wake-Up Interrupt Controller. This is an optional subblock that is **not included**.

5.3.2 MICROCHIP TERMS AND ACRONYMS

- Interrupt Aggregator

  This is a module that may be present at the chip level, which can combine multiple interrupt sources onto single interrupt inputs at the EC, causing them to share a vector.

- PMU

  Processor Memory Unit, this is a module that may be present at the chip level containing any memory resources that are closely-coupled to the MEC152x EC. It manages accesses from both the EC processor and chip-level bus masters.

5.4 ARM M4 IP Interfaces

This section defines only the interfaces to the ARM IP itself. For the interfaces of the entire block, see Section 5.5, "Block External Interfaces".

The MEC152x IP has the following major external interfaces, as shown in Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram":

- ICode AHB-Lite Interface
- DCode AHB-Lite Interface
- System AHB-Lite Interface
- Debug (JTAG) Interface
- Trace Port Interface
- Interrupt Interface

The EC operates on the model of a single 32-bit addressing space of byte addresses (4Gbytes, Von Neumann architecture) with Little-Endian byte ordering. On the basis of an internal decoder (part of the Bus Matrix shown in Figure 5-1), it routes Read/Write/Fetch accesses to one of three external interfaces, or in some cases internally (shown as the PPB interface).

The EC executes instructions out of closely-coupled memory via the ICode Interface. Data accesses to closely-coupled memory are handled via the DCode Interface. The EC accesses the rest of the on-chip address space via the System AHB-Lite interface. The Debugger program in the host can probe the EC and all EC addressable memory via the JTAG debug interface.

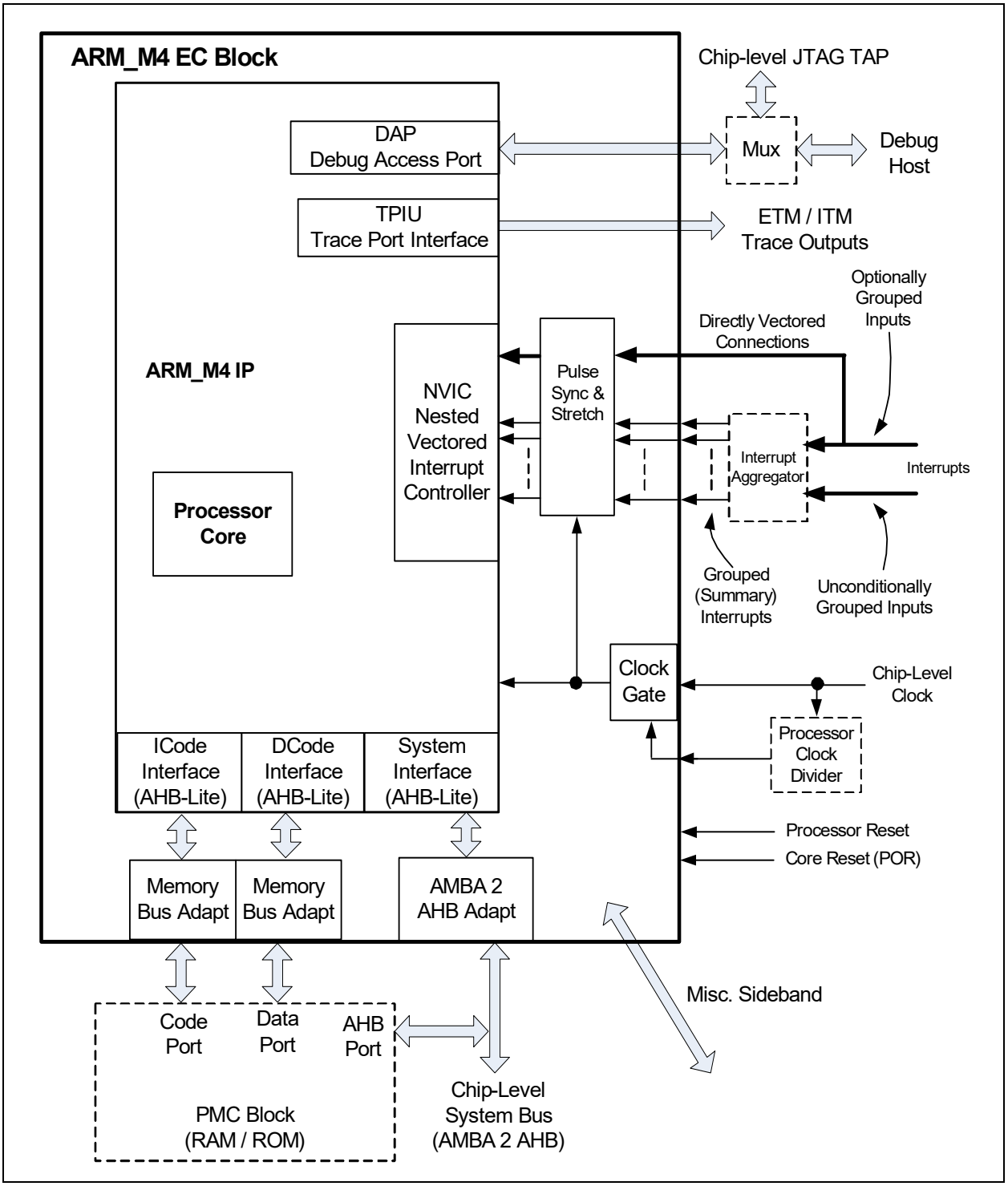
Aliased addressing spaces are provided at the chip level so that specific bus interfaces can be selected explicitly where needed. For example, the EC's Bit Banding feature uses the System AHB-Lite bus to access resources normally accessed via the DCode or ICode interface.

Note:	The EC executes most instructions in one clock cycle. If an instruction accesses code and data that are in different RAM blocks, then it takes one clock cycle to access both code and data (done in parallel). However, if the code and data blocks are in the same RAM block, then it takes two clock cycles (one clock for code access and one clock for data access) since it must do it sequentially.

© 2020 Microchip Technology Inc.

5.5 Block External Interfaces

FIGURE 5-1: ARM M4 BASED EMBEDDED CONTROLLER I/O BLOCK DIAGRAM



5.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

5.6.1 POWER DOMAINS

TABLE 5-1: POWER SOURCES

Name	Description
VTR_CORE	The ARM M4 Based Embedded Controller is powered by VTR_CORE.

5.6.2 CLOCK INPUTS

5.6.2.1 Basic Clocking

The basic clocking comes from a free-running Clock signal provided from the chip level.

TABLE 5-2: CLOCK INPUTS

Name	Description
48MHz	The clock source to the EC. Division of the clock rate is determined by the PROCESSOR_CLOCK_DIVIDE field in the Processor Clock Control Register.

5.6.2.2 System Tick Clocking

The System Tick clocking is controlled by a signal from chip-level logic. It is the 48MHz divided by the following:

- ((PROCESSOR_CLOCK_DIVIDE)x2)+1

5.6.2.3 Debug JTAG Clocking

The Debug JTAG clocking comes from chip-level logic, which may multiplex or gate this clock. See Section 5.10, "Debugger Access Support".

5.6.2.4 Trace Clocking

The Clock for the Trace interface is identical to the 48MHz input.

5.6.3 RESETS

The reset interface from the chip level is given below.

TABLE 5-3: RESET SIGNALS

Name	Description
RESET_EC	The ARM M4 Based Embedded Controller is reset by RESET_EC.

5.7 Interrupts

The ARM M4 Based Embedded Controller is equipped with an Interrupt Interface to respond to interrupts. These inputs go to the IP's NVIC block after a small amount of hardware processing to ensure their detection at varying clock rates. See Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram".

As shown in Figure 5-1, an Interrupt Aggregator block may exist at the chip level, to allow multiple related interrupts to be grouped onto the same NVIC input, and so allowing them to be serviced using the same vector. This may allow the same interrupt handler to be invoked for a group of related interrupt inputs. It may also be used to expand the total number of interrupt inputs that can be serviced.

The NMI (Non-Maskable Interrupt) connection is tied off and not used.

5.7.1 NVIC INTERRUPT INTERFACE

The NVIC interrupt unit can be wired to up to 240 interrupt inputs from the chip level. The interrupts that are actually connected from the chip level are defined in the Interrupt section.

All NVIC interrupt inputs can be programmed as either pulse or level triggered. They can also be individually masked, and individually assigned to their own hardware-managed priority level.

© 2020 Microchip Technology Inc.

5.7.2 NVIC RELATIONSHIP TO EXCEPTION VECTOR TABLE ENTRIES

The Vector Table consists of 4-byte entries, one per vector. Entry 0 is not a vector, but provides an initial Reset value for the Main Stack Pointer. Vectors start with the Reset vector, at Entry #1. Entries up through #15 are dedicated for internal exceptions, and do not involve the NVIC.

NVIC entries in the Vector Table start with Entry #16, so that NVIC Interrupt #0 is at Entry #16, and all NVIC interrupt numbers are incremented by 16 before accessing the Vector Table.

The number of connections to the NVIC determines the necessary minimum size of the Vector Table, as shown below. It can extend as far as 256 entries (255 vectors, plus the non-vector entry #0).

A Vector entry is used to load the Program Counter (PC) and the EPSR.T bit. Since the Program Counter only expresses code addresses in units of two-byte Halfwords, bit[0] of the vector location is used to load the EPSR.T bit instead, selecting THUMB mode for exception handling. Bit[0] must be '1' in all vectors, otherwise a UsageFault exception will be posted (INVSTATE, unimplemented instruction set). If the Reset vector is at fault, the exception posted will be HardFault instead.

TABLE 5-4: EXCEPTION AND INTERRUPT VECTOR TABLE LAYOUT

Table Entry	Exception Number	Exception
Special Entry for Reset Stack Pointer		
0	(none)	Holds Reset Value for the Main Stack Pointer. Not a Vector.
Core Internal Exception Vectors start here		
1	1	Reset Vector (PC + EPSR.T bit)
2	2	NMI (Non-Maskable Interrupt) Vector
3	3	HardFault Vector
4	4	MemManage Vector
5	5	BusFault Vector
6	6	UsageFault Vector
7	(none)	(Reserved by ARM Ltd.)
8	(none)	(Reserved by ARM Ltd.)
9	(none)	(Reserved by ARM Ltd.)
10	(none)	(Reserved by ARM Ltd.)
11	11	SVCall Vector
12	12	Debug Monitor Vector
13	(none)	(Reserved by ARM Ltd.)
14	14	PendSV Vector
15	15	SysTick Vector
NVIC Interrupt Vectors start here		
16	16	NVIC Interrupt #0 Vector
. . .	. . .	. . .
n + 16	n + 16	NVIC Interrupt #n Vector
. . .	. . .	. . .
max + 16	max + 16	NVIC Interrupt #max Vector (Highest-numbered NVIC connection.)
. . .	. . .	. Table size may (but need not) extend further. . .
255	255	NVIC Interrupt #239 (Architectural Limit of Exception Table)

5.8 Low Power Modes

The ARM processor can enter Sleep or Deep Sleep modes internally. This action will cause an output signal Clock Required to be turned off, allowing clocks to be stopped from the chip level. However, Clock Required will still be held active, or set to active, unless all of the following conditions exist:

- No interrupt is pending.
- An input signal Sleep Enable from the chip level is active.
- The Debug JTAG port is inactive (reset or configured not present).

In addition, regardless of the above conditions, a chip-level input signal Force Halt may halt the processor and remove Clock Required.

5.9 Description

5.9.1 BUS CONNECTIONS

There are three bus connections used from MEC152x EC block, which are directly related to the IP bus ports. See Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram".

For the mapping of addresses at the chip level, see Section 3.0, "Device Inventory".

5.9.1.1 Closely Coupled Instruction Fetch Bus

As shown in Figure 5-1, the AHB-Lite ICode port from the IP is converted to a more conventional SRAM memory-style bus and connected to the on-chip memory resources with routing priority appropriate to Instruction Fetches.

5.9.1.2 Closely Coupled Data Bus

As shown in Figure 5-1, the AHB-Lite DCode port from the IP is converted to a more conventional SRAM memory-style bus and connected to the on-chip memory resources with routing priority appropriate to fast Data Read/Write accesses.

5.9.1.3 Chip-Level System Bus

As shown in Figure 5-1, the AHB-Lite System port from the IP is converted from AHB-Lite to fully arbitrated multi-master capability (the AMBA 2 defined AHB bus: see Reference [5], ARM Limited: AMBA® Specification (Rev 2.0), IHI0011A, 13 May 1999). Using this bus, all addressable on-chip resources are available. The multi-mastering capability supports the Microchip DMA and EMI features if present, as well as the Bit-Banding feature of the IP itself.

As also shown in Figure 5-1, the Closely-Coupled memory resources are also available through this bus connection using aliased addresses. This is required in order to allow Bit Banding to be used in these regions, but it also allows them to be accessed by DMA and other bus masters at the chip level.

Note:	Registers with properties such as Write-1-to-Clear (W1C), Read-to-Clear and FIFOs need to be handled with appropriate care when being used with the bit band alias addressing scheme. Accessing such a register through a bit band alias address will cause the hardware to perform a read-modify-write, and if a W1C-type bit is set, it will get cleared with such an access. For example, using a bit band access to the Interrupt Aggregator, including the Interrupt Enables and Block Interrupt Status to clear an IRQ will clear all active IRQs.

5.9.2 INSTRUCTION PIPELINING

There are no special considerations except as defined by ARM documentation.

5.10 Debugger Access Support

An external Debugger accesses the chip through a JTAG standard interface. The ARM Debug Access Port supports both the 2-pin SWD (Serial Wire Debug) interface and the 4-pin JTAG interface.

As shown in Figure 5-1, "ARM M4 Based Embedded Controller I/O Block Diagram", other resources at the chip level that share the JTAG port pins; for example chip-level Boundary Scan.

By default, debug access is disabled when the EC begins executing code. EC code enables debugging by writing the Debug Enable Register in the EC Subsystem Registers block.

© 2020 Microchip Technology Inc.

TABLE 5-5: ARM JTAG ID

ARM Debug Mode	JTAG ID
SW-DP (2-wire)	0x2BA01477
JTAG (4-wire)	0x4BA00477

5.10.1 DEBUG AND ACCESS PORTS (SWJ-DP AND AHB-AP SUBBLOCKS)

These two subblocks work together to provide access to the chip for the Debugger using the Debug JTAG connection, as described in Chapter 4 of the ARM Limited: ARM® Debug Interface v5 Architecture Specification, IHI0031A, 8 February 2006.

5.10.2 BREAKPOINT, WATCHPOINT AND TRACE SUPPORT

See References [11], ARM Limited: ARM® Debug Interface v5 Architecture Specification, IHI0031A, 8 February 2006 and [12], ARM Limited: ARM® Debug Interface v5 Architecture Specification ADIv5.1 Supplement, DSA09-PRDC-008772, 17 August 2009. A summary of functionality follows.

Breakpoint and Watchpoint facilities can be programmed to do one of the following:

- Halt the processor. This means that the external Debugger will detect the event by periodically polling the state of the EC.
- Transfer control to an internal Debug Monitor firmware routine, by triggering the Debug Monitor exception (see Table 5-4, "Exception and Interrupt Vector Table Layout").

5.10.2.1 Instrumentation Support (ITM Subblock)

The Instrumentation Trace Macrocell (ITM) is for profiling software. This uses non-blocking register accesses, with a fixed low-intrusion overhead, and can be added to a Real-Time Operating System (RTOS), application, or exception handler. If necessary, product code can retain the register access instructions, avoiding probe effects.

5.10.2.2 HW Breakpoints and ROM Patching (FPB Subblock)

The Flash Patch and Breakpoint (FPB) block. This block can remap sections of ROM, typically Flash memory, to regions of RAM, and can set breakpoints on code in ROM. This block can be used for debug, and to provide a code or data patch to an application that requires field updates to a product in ROM.

5.10.2.3 Data Watchpoints and Trace (DWT Subblock)

The Debug Watchpoint and Trace (DWT) block provides watchpoint support, program counter sampling for performance monitoring, and embedded trace trigger control.

5.10.2.4 Trace Interface (ETM and TPIU)

The Embedded Trace Macrocell (ETM) provides instruction tracing capability. For details of functionality and usage, see References [13], ARM Limited: Embedded Trace Macrocell™ (ETMv1.0 to ETMv3.5) Architecture Specification, IHI0014Q, 23 September 2011 and [14], ARM Limited: CoreSight™ ETM™-M4 Technical Reference Manual, DDI0440C, 29 June 2010.

The Trace Port Interface Unit (TPIU) provides the external interface for the ITM, DWT and ETM.

5.11 Delay Register

5.11.1 DELAY REGISTER

Offset	1000_0000h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	RES	-	-
4:0	DELAY Writing a value *n,* from 0h to 31h, to this register will cause the ARM processor to stall for (*n*+1) microseconds (that is, from 1µS to 32µS). Reads will return the last value read immediately. There is no delay.	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

6.0 RAM AND ROM

6.1 SRAM

The MEC152x contains two blocks of SRAM. The two SRAM blocks in the MEC152x total 256KB. Both SRAM blocks can be used for either program or data accesses. Performance is enhanced when program fetches and data accesses are to different SRAM blocks, but a program will operate correctly even if both program and data accesses are targeting the same block simultaneously.

- The first SRAM, which is optimized for code access, is 224KB
- The second SRAM, which is optimized for data access, is32KB

6.2 ROM

The MEC152x contains a 64KB block of ROM, located at address 00000000h in the ARM address space. The ROM contains boot code that is executed after the de-assertion of RESET_SYS. The boot code loads an executable code image into SRAM. The ROM also includes a set of API functions that can be used for cryptographic functions, as well as loading SRAM with programs or data.

6.3 Additional Memory Regions

6.3.1 ALIAS RAM

The Alias RAM region, starting at address 20000000h, is an alias of the SRAM located at 118000h, and is the same size as that SRAM block. EC software can access memory in either the primary address or in the alias region; however, access is considerably slower to the alias region. The alias region exists in order to enable the ARM bit-band region located at address 20000000h.

6.3.2 RAM BIT-BAND REGION

The RAM bit-band region is an alias of the SRAM located at 118000h, except that each bit is aliased to bit 0 of a 32-bit doubleword in the bit-band region. The upper 31 bits in each doubleword of the bit-band region are always 0. The bit-band region is therefore 32 times the size of the SRAM region. It can be used for atomic updates of individual bits of the SRAM, and is a feature of the ARM architecture.

The bit-band region can only be accessed by the ARM processor. Accesses by any other bus master will cause a memory fault.

6.3.3 CRYPTOGRAPHIC RAM

The cryptographic RAM is used by the cryptographic API functions in the ROM

6.3.4 REGISTER BIT-BAND REGION

The Register bit-band region is an 32-to-1 alias of the device register space starting at address 40000000h and ending with the Host register space at 400FFFFF. Every bit in the register space is aliased to a byte in the Register bit-band region, and like the RAM bit-band region, can be used by EC software to read and write individual register bits. Only the EC Device Registers and the GPIO Registers can be accessed via the bit-band region.

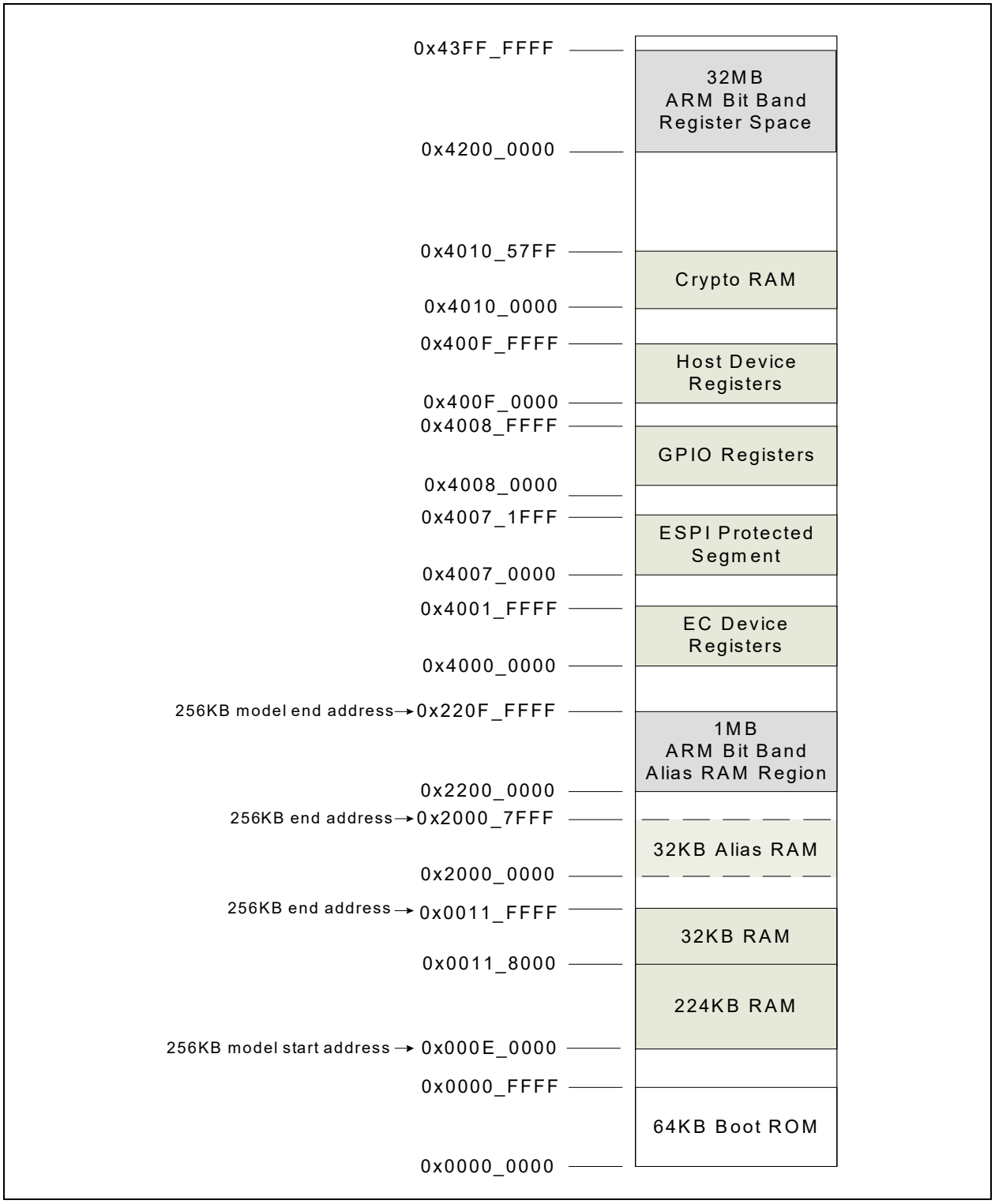
A one bit write operation to a register bit in the bit-band region is implemented by the ARM processor by performing a read, a bit modification, followed by a write back to the same register. Software must be careful when using bit-banding if a register contains bits have side effects triggered by a read.

The bit-band region can only be accessed by the ARM processor. Accesses by any other bus master will cause a memory fault.

6.4 Memory Map

The memory map of the RAM and ROM is represented as follows:

FIGURE 6-1: MEMORY LAYOUT



© 2020 Microchip Technology Inc.

7.0 INTERNAL DMA CONTROLLER

7.1 Introduction

The Internal DMA Controller transfers data to/from the source from/to the destination. The firmware is responsible for setting up each channel. Afterwards either the firmware or the hardware may perform the flow control. The hardware flow control exists entirely inside the source device. Each transfer may be 1, 2, or 4 bytes in size, so long as the device supports a transfer of that size. Every device must be on the internal 32-bit address space.

7.2 References

No references have been cited for this chapter.

7.3 Terminology

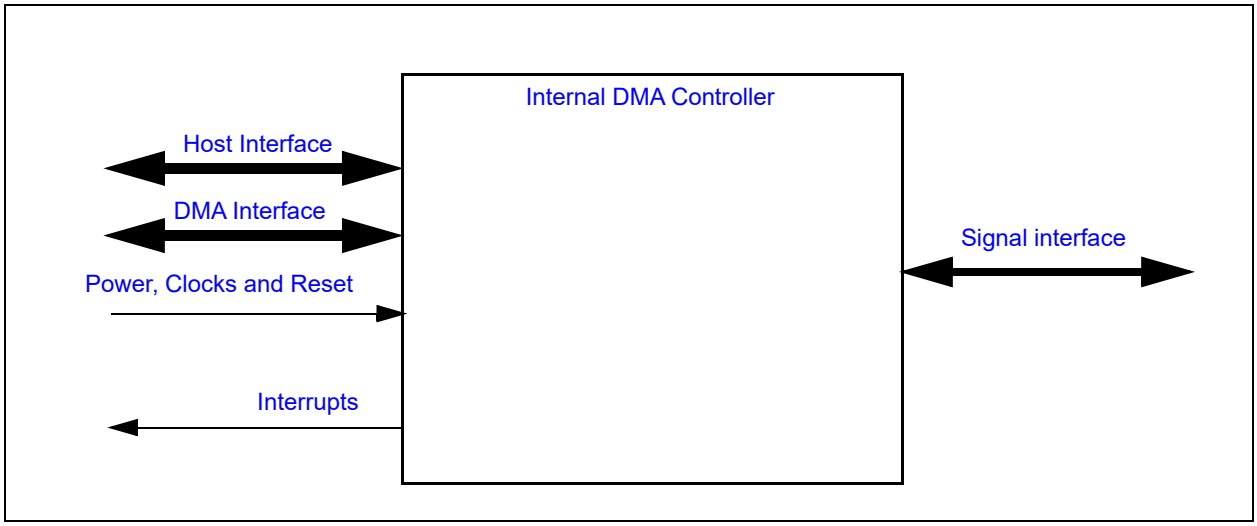
TABLE 7-1: TERMINOLOGY

Term	Definition
DMA Transfer	This is a complete **DMA Transfer** which is done after the **Master Device** terminates the transfer, the Firmware Aborts the transfer or the DMA reaches its transfer limit. A DMA Transfer may consist of one or more data packets.
Data Packet	Each data packet may be composed of 1, 2, or 4 bytes. The size of the data packet is limited by the max size supported by both the source and the destination. Both source and destination will transfer the same number of bytes per packet.
Channel	The Channel is responsible for end-to-end (source-to-destination) Data Packet delivery.
Device	A Device may refer to a Master or Slave connected to the DMA Channel. Each DMA Channel may be assigned one or more devices.
Master Device	This is the master of the DMA, which determines when it is active. The Firmware is the master while operating in Firmware Flow Control. The Hardware is the master while operating in Hardware Flow Control. The Master Device in Hardware Mode is selected by **DMA Channel Control:Hardware Flow Control Device**. It is the index of the **Flow Control** Port.
Slave Device	The Slave Device is defined as the device associated with the targeted Memory Address.
Source	The DMA Controller moves data from the Source to the Destination. The Source provides the data. The Source may be either the Master or Slave Controller.
Destination	The DMA Controller moves data from the Source to the Destination. The Destination receives the data. The Destination may be either the Master or Slave Controller.

7.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 7-1: INTERNAL DMA CONTROLLER I/O DIAGRAM



7.5 Signal interface

This block doesn't have any external signals that may be routed to the pin interface. This DMA Controller is intended to be used internally to transfer large amounts of data without the embedded controller being actively involved in the transfer.

7.6 Host Interface

The registers defined for the Internal DMA Controller are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

7.7 DMA Interface

Each DMA Master Device that may engage in a DMA transfer must have a compliant DMA interface. The following table lists the DMA Devices in the MEC152x.

TABLE 7-2: DMA CONTROLLER DEVICE SELECTION

Device Name	Device Number (Note 1)	Controller Source
SMB-I2C 0 Controller	0	Slave
	1	Master
SMB-I2C 1 Controller	2	Slave
	3	Master
SMB-I2C 2 Controller	4	Slave
	5	Master
SMB-I2C 3 Controller	6	Slave
	7	Master
SMB-I2C 4Controller	8	Transmit
	9	Receive

Note 1: The Device Number is programmed into field HARDWARE_FLOW_CONTROL_DEVICE of the DMA Channel N Control Register register.

© 2020 Microchip Technology Inc.

TABLE 7-2: DMA CONTROLLER DEVICE SELECTION (CONTINUED)

Device Name	Device Number (Note 1)	Controller Source
QMSPI Controller	10	Transmit
	11	Receive

Note 1: The Device Number is programmed into field HARDWARE_FLOW_CONTROL_DEVICE of the DMA Channel N Control Register register.

TABLE 7-3: DMA CONTROLLER MASTER DEVICES SIGNAL LIST

Device Name	Dev Num	Device Signal Name	Direction	Description
SMB-I2C 0 Controller	0	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	1	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
SMB-I2C 1 Controller	2	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	3	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
SMB-I2C 2 Controller	4	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	5	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Done	OUTPUT	DMA termination control from DMA Controller to Master channel.

TABLE 7-3: DMA CONTROLLER MASTER DEVICES SIGNAL LIST (CONTINUED)

Device Name	Dev Num	Device Signal Name	Direction	Description
SMB-I2C 3 Controller	6	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	7	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
SMB-I2C 4 Controller	8	SMB-I2C_SDMA_Req	INPUT	DMA request control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Term	INPUT	DMA termination control from SMB-I2C Slave channel.
		SMB-I2C_SDMA_Done	OUTPUT	DMA termination control from DMA Controller to Slave channel.
	9	SMB-I2C_MDMA_Req	INPUT	DMA request control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Term	INPUT	DMA termination control from SMB-I2C Master channel.
		SMB-I2C_MDMA_Done	OUTPUT	DMA termination control from DMA Controller to Master channel.
Quad SPI Controller	10	QSPI_TDMA_Req	INPUT	DMA request control from Quad SPI TX channel.
		QSPI_TDMA_Term	INPUT	DMA termination control from Quad SPI TX channel.
		QMSPI_TDMA_Done	OUTPUT	DMA termination control from DMA Controller to Quad SPI TDMA Channel.
	11	QSPI_RDMA_Req	INPUT	DMA request control from Quad SPI RX channel.
		QSPI_RDMA_Term	INPUT	DMA termination control from Quad SPI RX channel.
		QMSPI_RDMA_Done	OUTPUT	DMA termination control from DMA Controller to Quad SPI RDMA Channel.

© 2020 Microchip Technology Inc.

7.8 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

7.8.1 POWER DOMAINS

TABLE 7-4: POWER SOURCES

Name	Description
VTR_CORE	This power well sources the registers and logic in this block.

7.8.2 CLOCK INPUTS

TABLE 7-5: CLOCK INPUTS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).

7.8.3 RESETS

TABLE 7-6: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.
RESET	This reset is generated if either the RESET_SYS is asserted or the SOFT_RESET bit is asserted.

7.9 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 7-7: INTERRUPTS

Source	Description
DMAx	Direct Memory Access Channel x This signal is generated by the STATUS_DONE bit.

7.10 Low Power Modes

The Internal DMA Controller may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

When the block is commanded to go to sleep it will place the DMA block into sleep mode only after all transactions on the DMA have been completed. For Firmware Flow Controlled transactions, the DMA will wait until it hits its terminal count and clears the Go control bit. For Hardware Flow Control, the DMA will go to sleep after either the terminal count is hit, or the Master device flags the terminate signal.

7.11 Description

The MEC152x features a 12 channel DMA controller. The DMA controller can autonomously move data from/to any DMA capable master device to/from any populated memory location. This mechanism allows hardware IP blocks to transfer large amounts of data into or out of memory without EC intervention.

The DMA has the following characteristics:

- Data is only moved 1 Data Packet at a time
- Data only moves between devices that are accessible via the internal 32-bit address space
- The DMA Controller has 12 DMA Channels
- Each DMA Channel may be configured to communicate with any DMA capable device on the 32-bit internal address space. Each device has been assigned a device number. See Section 7.7, "DMA Interface".

The controller will access SRAM buffers only with incrementing addresses (that is, it cannot start at the top of a buffer, nor does it handle circular buffers automatically). The controller does not handle chaining (that is, automatically starting a new DMA transfer when one finishes).

7.11.1 CONFIGURATION

The DMA Controller is enabled via the ACTIVATE bit in DMA Main Control Register register.

Each DMA Channel must also be individually enabled via the CHANNEL_ACTIVATE bit in the DMA Channel N Activate Register to be operational.

Before starting a DMA transaction on a DMA Channel the host must assign a DMA Master to the channel via HARDWARE_FLOW_CONTROL_DEVICE. The host must not configure two different channels to the same DMA Master at the same time.

Data will be transfered between the DMA Master, starting at the programmed DEVICE_ADDRESS, and the targeted memory location, starting at the MEMORY_START_ADDRESS. The address for either the DMA Master or the targeted memory location may remain static or it may increment. To enable the DMA Master to increment its address set the INCREMENT_DEVICE_ADDRESS bit. To enable the targeted memory location to increment its addresses set the INCREMENT_MEMORY_ADDRESS. The DMA transfer will continue as long as the target memory address being accessed is less than the MEMORY_END_ADDRESS. If the DMA Controller detects that the memory location it is attempting to access on the Target is equal to the MEMORY_END_ADDRESS it will notify the DMA Master that the transaction is done. Otherwise the Data will be transferred in packets. The size of the packet is determined by the TRANSFER_SIZE.

7.11.2 OPERATION

The DMA Controller is designed to move data from one memory location to another.

7.11.2.1 Establishing a Connection

A DMA Master will initiate a DMA Transaction by requesting access to a channel. The DMA arbiter, which evaluates each channel request using a basic round robin algorithm, will grant access to the DMA master. Once granted, the channel will hold the grant until it decides to release it, by notifying the DMA Controller that it is done.

If Firmware wants to prevent any other channels from being granted while it is active it can set the LOCK_CHANNEL bit.

7.11.2.2 Initiating a Transfer

Once a connection is established the DMA Master will issue a DMA request to start a DMA transfer. If Firmware wants to have a transfer request serviced it must set the RUN bit to have its transfer requests serviced.

Firmware can initiate a transaction by setting the TRANSFER_GO bit. The DMA transfer will remain active until either the Master issues a Terminate or the DMA Controller signals that the transfer is DONE. Firmware may terminate a transaction by setting the TRANSFER_ABORT bit.

Note:	Before initiating a DMA transaction via firmware the hardware flow control must be disabled via the DISABLE_HARDWARE_FLOW_CONTROL bit.

Data may be moved from the DMA Master to the targeted Memory address or from the targeted Memory Address to the DMA Master. The direction of the transfer is determined by the TRANSFER_DIRECTION bit.

Once a transaction has been initiated firmware can use the STATUS_DONE bit to determine when the transaction is completed. This status bit is routed to the interrupt interface. In the same register there are additional status bits that indicate if the transaction completed successfully or with errors. These bits are OR'd together with the STATUS_DONE bit to generate the interrupt event. Each status be may be individually enabled/disabled from generating this event.

7.11.2.3 Reusing a DMA Channel

After a DMA Channel controller has completed, firmware **must** clear both the DMA Channel N Control Register and the DMA Channel N Interrupt Status Register. After both have been cleared to 0, the Channel Control Register can then be configured for the next transaction.

7.11.2.4 CRC Generation

A CRC generator can be attached to a DMA channel in order to generate a CRC on the data as it is transfered from the source to the destination. The CRC used is the CRC-32 algorithm used in IEEE 802.3 and many other protocols, using the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. The CRC generation takes place in parallel with the data transfer; enabling CRC will not increase the time to complete a DMA transaction. The CRC generator has the optional ability to automatically transfer the generated CRC to the destination after the data transfer has completed.

© 2020 Microchip Technology Inc.

CRC generation is subject to a number of restrictions:

- The CRC is only generated on channels that have the CRC hardware. See Table 7-10, "Channel Register Summary" for a definition of which channels have the ability to generate a CRC
- The DMA transfer must be 32-bits
- If CRC is enabled, DMA interrupts are inhibited until the CRC is completed, including the optional post-transfer copy of it is enabled
- The CRC must be initialized by firmware. The value FFFFFFFFh must be written to the Data Register in order to initialize the generator for the standard CRC-32-IEEE algorithm
- The CRC will be bit-order reversed and inverted as required by the CRC algorithm

7.11.2.5 Block Fill Option

A Fill engine can be attached to a DMA channel in order to provide a fast mechanism to set a block of memory to a fixed value (for example, clearing a block of memory to zero). The block fill operation runs approximately twice as fast as a memory-to-memory copy.

In order to fill memory with a constant value, firmware **must** configure the channel in the following order:

1. Set the DMA Channel N Fill Data Register to the desired fill value
2. Set the DMA Channel N Fill Enable Register to '1b', enabling the Fill engine
3. Set the DMA Channel N Control Register to the following values:
   - RUN = 0
   - TRANSFER_DIRECTION = 0 (memory destination)
   - INCREMENT_MEMORY_ADDRESS = 1 (increment memory address after each transfer)
   - INCREMENT_DEVICE_ADDRESS = 1
   - DISABLE_HARDWARE_FLOW_CONTROL = 1 (no hardware flow control)
   - TRANSFER_SIZE = 1, 2 or 4 (as required)
   - TRANSFER_ABORT = 0
   - TRANSFER_GO = 1 (this starts the transfer)

7.12 EC Registers

The DMA Controller consists of a Main Block and a number of Channels. Table 7-9, "Main Register Summary" lists the registers in the Main Block and Table 7-10, "Channel Register Summary" lists the registers in each channel. Addresses for each register are determined by adding the offset to the Base Address for the DMA Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

Registers are listed separately for the Main Block of the DMA Controller and for a DMA Channel. Each Channel has the same set of registers. The absolute register address for registers in each channel are defined by adding the Base Address for the DMA Controller Block, the Offset for the Channel shown in Table 7-8, "DMA Channel Offsets" to the offsets listed in Table 7-9, "Main Register Summary" or Table 7-10, "Channel Register Summary".

TABLE 7-8: DMA CHANNEL OFFSETS

Instance Name	Channel Number	Offset
DMA Controller	Main Block	000h
DMA Controller	0	040h
DMA Controller	1	080h
DMA Controller	2	0C0h
DMA Controller	3	100h
DMA Controller	4	140h
DMA Controller	5	180h
DMA Controller	6	1C0h
DMA Controller	7	200h
DMA Controller	8	240h
DMA Controller	9	280h

© 2020 Microchip Technology Inc.

TABLE 7-8: DMA CHANNEL OFFSETS (CONTINUED)

Instance Name	Channel Number	Offset
DMA Controller	10	2C0h
DMA Controller	11	300h

TABLE 7-9: MAIN REGISTER SUMMARY

Offset	Register Name
00h	DMA Main Control Register
04h	DMA Data Packet Register

7.12.1 DMA MAIN CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-
1	SOFT_RESET Soft reset the entire module. This bit is self-clearing.	W	0b	-
0	ACTIVATE Enable the blocks operation. 1=Enable block. Each individual channel must be enabled separately. 0=Disable all channels.	R/WS	0b	RESET

7.12.2 DMA DATA PACKET REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DATA_PACKET Debug register that has the data that is stored in the Data Packet. This data is read data from the currently active transfer source.	R	0000h	-

TABLE 7-10: CHANNEL REGISTER SUMMARY

Offset	Register Name (Note 1)
00h	DMA Channel N Activate Register
04h	DMA Channel N Memory Start Address Register
08h	DMA Channel N Memory End Address Register
0Ch	DMA Channel N Device Address
10h	DMA Channel N Control Register
14h	DMA Channel N Interrupt Status Register
18h	DMA Channel N Interrupt Enable Register

Note 1: The letter 'N' following DMA Channel indicates the Channel Number. Each Channel implemented will have these registers to determine that channel's operation.

2: These registers are only present on DMA Channel 0. They are reserved on all other channels.

3: These registers are only present on DMA Channel 1. They are reserved on all other channels.

© 2020 Microchip Technology Inc.

TABLE 7-10: CHANNEL REGISTER SUMMARY (CONTINUED)

Offset	Register Name (Note 1)
1Ch	TEST
20h (Note 2)	DMA Channel N CRC Enable Register
24h (Note 2)	DMA Channel N CRC Data Register
28h (Note 2)	DMA Channel N CRC Post Status Register
2Ch (Note 2)	TEST
20h (Note 3)	DMA Channel N Fill Enable Register
24h (Note 3)	DMA Channel N Fill Data Register
28h (Note 3)	DMA Channel N Fill Status Register
2Ch (Note 3)	TEST

Note 1: The letter 'N' following DMA Channel indicates the Channel Number. Each Channel implemented will have these registers to determine that channel's operation.

2: These registers are only present on DMA Channel 0. They are reserved on all other channels.

3: These registers are only present on DMA Channel 1. They are reserved on all other channels.

7.12.3 DMA CHANNEL N ACTIVATE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	CHANNEL_ACTIVATE Enable this channel for operation. The DMA Main Control:Activate must also be enabled for this channel to be operational.	R/W	0h	RESET

7.12.4 DMA CHANNEL N MEMORY START ADDRESS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MEMORY_START_ADDRESS This is the starting address for the Memory device. This field is updated by Hardware after every packet transfer by the size of the transfer, as defined by DMA Channel Control:Channel Transfer Size while the DMA Channel Control:Increment Memory Address is Enabled. The Memory device is defined as the device that is the slave device in the transfer. With Hardware Flow Control, the Memory device is the device that is not connected to the Hardware Flow Controlling device.	R/W	0000h	RESET

7.12.5 DMA CHANNEL N MEMORY END ADDRESS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MEMORY_END_ADDRESS This is the ending address for the Memory device. This will define the limit of the transfer, so long as DMA Channel Control:Increment Memory Address is Enabled. When the Memory Start Address is equal to this value, the DMA will terminate the transfer and flag the status DMA Channel Interrupt:Status Done. **Note:** If the TRANSFER_SIZE field in the DMA Channel N Control Register is set to 2 (for 2-byte transfers, this address **must** be evenly divisible by 2 or the transfer will not terminate properly. If the TRANSFER_SIZE field is set to 4 (for 4-byte transfers, this address **must** be evenly divisible by 4 or the transfer will not terminate properly.	R/W	0000h	RESET

7.12.6 DMA CHANNEL N DEVICE ADDRESS

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DEVICE_ADDRESS This is the Master Device address. This is used as the address that will access the Device on the DMA. The Device is defined as the Master of the DMA transfer; as in the device that is controlling the Hardware Flow Control. This field is updated by Hardware after every Data Packet transfer by the size of the transfer, as defined by DMA Channel Control:Transfer Size while the DMA Channel Control:Increment Device Address is Enabled.	R/W	0000h	RESET

7.12.7 DMA CHANNEL N CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:26	Reserved	RES	-	-
25	TRANSFER_ABORT This is used to abort the current transfer on this DMA Channel. The aborted transfer will be forced to terminate immediately.	R/W	0h	RESET
24	TRANSFER_GO This is used for the **Firmware Flow Control** DMA transfer. This is used to start a transfer under the **Firmware Flow Control**. Do not use this in conjunction with the **Hardware Flow Control**; DISABLE_HARDWARE_FLOW_CONTROL must be set in order for this field to function correctly.	R/W	0h	RESET
23	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
22:20	TRANSFER_SIZE This is the transfer size in Bytes of each Data Packet transfer. The transfer size must be a legal transfer size. Valid sizes are 1, 2 and 4 Bytes.	R/W	0h	RESET
19	DISABLE_HARDWARE_FLOW_CONTROL Setting this bit to '1'b will Disable **Hardware Flow Control**. When disabled, any DMA Master device attempting to communicate to the DMA over the DMA Flow Control Interface will be ignored. This should be set before using the DMA channel in **Firmware Flow Control** mode.	R/W	0h	RESET
18	LOCK_CHANNEL This is used to lock the arbitration of the Channel Arbiter on this channel once this channel is granted. Once this is locked, it will remain on the arbiter until it has completed it transfer (either the Transfer Aborted, Transfer Done or Transfer Terminated conditions). **Note:** This setting may starve other channels if the locked channel takes an excessive period of time to complete.	R/W	0h	RESET
17	INCREMENT_DEVICE_ADDRESS If this bit is '1'b, the DEVICE_ADDRESS will be incremented by TRANSFER_SIZE after every Data Packet transfer	R/W	0h	RESET
16	INCREMENT_MEMORY_ADDRESS If this bit is '1'b, the MEMORY_START_ADDRESS will be incremented by TRANSFER_SIZE after every Data Packet transfer **Note:** If this is not set, the DMA will never terminate the transfer on its own. It will have to be terminated through the Hardware Flow Control or through a DMA Channel Control:Transfer Abort.	R/W	0h	RESET
15:9	HARDWARE_FLOW_CONTROL_DEVICE This is the device that is connected to this channel as its Hardware Flow Control master. The Flow Control Interface is a bus with each master concatenated onto it. This selects which bus index of the concatenated Flow Control Interface bus is targeted towards this channel.	R/W	0h	RESET
8	TRANSFER_DIRECTION This determines the direction of the DMA Transfer. 1=Data Packet Read from MEMORY_START_ADDRESS followed by Data Packet Write to DEVICE_ADDRESS 0=Data Packet Read from DEVICE_ADDRESS followed by Data Packet Write to MEMORY_START_ADDRESS	R/W	0h	RESET
7:6	Reserved	RES	-	-
5	BUSY This is a status signal. 1=The DMA Channel is busy (FSM is not IDLE) 0=The DMA Channel is not busy (FSM is IDLE)	R	0h	RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4:3	TEST	R	0h	RESET
2	DONE This is a status signal. It is only valid while RUN is Enabled. This is the inverse of the **DMA Channel Control:Busy** field, except this is qualified with the **DMA Channel Control:Run** field. 1=Channel is done 0=Channel is not done or it is OFF	R	0h	RESET
1	REQUEST This is a status field. 1=There is a transfer request from the Master Device 0=There is no transfer request from the Master Device	R	0h	RESET
0	RUN This is a control field. It only applies to **Hardware Flow Control** mode. 1=This channel is enabled and will service transfer requests 0=This channel is disabled. All transfer requests are ignored	R/W	0h	RESET

7.12.8 DMA CHANNEL N INTERRUPT STATUS REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	-	-
2	STATUS_DONE This is an interrupt source register. This flags when the DMA Channel has completed a transfer successfully on its side. A completed transfer is defined as when the DMA Channel reaches its limit; **Memory Start Address** equals **Memory End Address**. A completion due to a **Hardware Flow Control Terminate** will not flag this interrupt. 1=MEMORY_START_ADDRESS equals MEMORY_END_ADDRESS 0=MEMORY_START_ADDRESS does not equal MEMORY_END_ADDRESS	R/WC	0h	RESET
1	STATUS_ENABLE_FLOW_CONTROL This is an interrupt source register. This flags when the DMA Channel has encountered a **Hardware Flow Control Request** after the DMA Channel has completed the transfer. This means the Master Device is attempting to overflow the DMA. 1=Hardware Flow Control is requesting after the transfer has completed 0=No Hardware Flow Control event	R/WC	0h	RESET

© 2020 Microchip Technology Inc.

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	STATUS_BUS_ERROR This is an interrupt source register. This flags when there is an Error detected over the internal 32-bit Bus. 1=Error detected.	R/WC	0h	RESET

7.12.9 DMA CHANNEL N INTERRUPT ENABLE REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	-	-
2	STATUS_ENABLE_DONE This is an interrupt enable for STATUS_DONE. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET
1	STATUS_ENABLE_FLOW_CONTROL_ERROR This is an interrupt enable for STATUS_ENABLE_FLOW_CONTROL. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET
0	STATUS_ENABLE_BUS_ERROR This is an interrupt enable for STATUS_BUS_ERROR. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET

7.12.10 DMA CHANNEL N CRC ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	CRC_POST_TRANSFER_ENABLE The bit enables the transfer of the calculated CRC-32 after the completion of the DMA transaction. If the DMA transaction is aborted by either firmware or an internal bus error, the transfer will not occur. If the target of the DMA transfer is a device and the device signaled the termination of the DMA transaction, the CRC post transfer will not occur. 1=Enable the transfer of CRC-32 for DMA Channel N after the DMA transaction completes 0=Disable the automatic transfer of the CRC	R/W	0h	RESET
0	CRC_MODE_ENABLE 1=Enable the calculation of CRC-32 for DMA Channel N 0=Disable the calculation of CRC-32 for DMA Channel N	R/W	0h	RESET

7.12.11 DMA CHANNEL N CRC DATA REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CRC Writes to this register initialize the CRC generator. Reads from this register return the output of the CRC that is calculated from the data transfered by DMA Channel N. The output of the CRC generator is bit-reversed and inverted on reads, as required by the CRC-32-IEEE definition. A CRC can be accumulated across multiple DMA transactions on Channel N. If it is necessary to save the intermediate CRC value, the result of the read of this register must be bit-reversed and inverted before being written back to this register.	R/W	0h	RESET

© 2020 Microchip Technology Inc.

7.12.12 DMA CHANNEL N CRC POST STATUS REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	CRC_DATA_READY This bit is set to ‘1b’ when the DMA controller is processing the post-transfer of the CRC data. This bit is cleared to ‘0b’ when the post-transfer completes.	R	0h	RESET
2	CRC_DATA_DONE This bit is set to ‘1b’ when the DMA controller has completed the post-transfer of the CRC data. This bit is cleared to ‘0b’ when the a new DMA transfer starts.	R	0h	RESET
1	CRC_RUNNING This bit is set to ‘1b’ when the DMA controller starts the post-transfer transmission of the CRC. It is only set when the post-transfer is enabled by the CRC_POST_TRANSFER_ENABLE field. This bit is cleared to ‘0b’ when the post-transfer completes.	R	0h	RESET
0	CRC_DONE This bit is set to ‘1b’ when the CRC calculation has completed from either normal or forced termination. It is cleared to ‘0b’ when the DMA controller starts a new transfer on the channel.	R	0h	RESET

7.12.13 DMA CHANNEL N FILL ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	FILL_MODE_ENABLE 1=Enable the Fill Engine for DMA Channel N 0=Disable the Fill Engine for DMA Channel N	R/W	0h	RESET

7.12.14 DMA CHANNEL N FILL DATA REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	DATA This is the data pattern used to fill memory.	R/W	0h	RESET

7.12.15 DMA CHANNEL N FILL STATUS REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	FILL_RUNNING This bit is ‘1b’ when the Fill operation starts and is cleared to ‘0b’ when the Fill operation completes.	R	0h	RESET
0	FILL_DONE This bit is set to ‘1b’ when the Fill operation has completed from either normal or forced termination. It is cleared to ‘0b’ when the DMA controller starts a new transfer on the channel.	R	0h	RESET

© 2020 Microchip Technology Inc.

8.0 EC INTERRUPT AGGREGATOR

8.1 Introduction

The EC Interrupt Aggregator works in conjunction with the processor's interrupt interface to handle hardware interrupts and exceptions.

Exceptions are synchronous to instructions, are not maskable, and have higher priority than interrupts. All three exceptions - reset, memory error, and instruction error - are hardwired directly to the processor. Interrupts are typically asynchronous and are maskable.

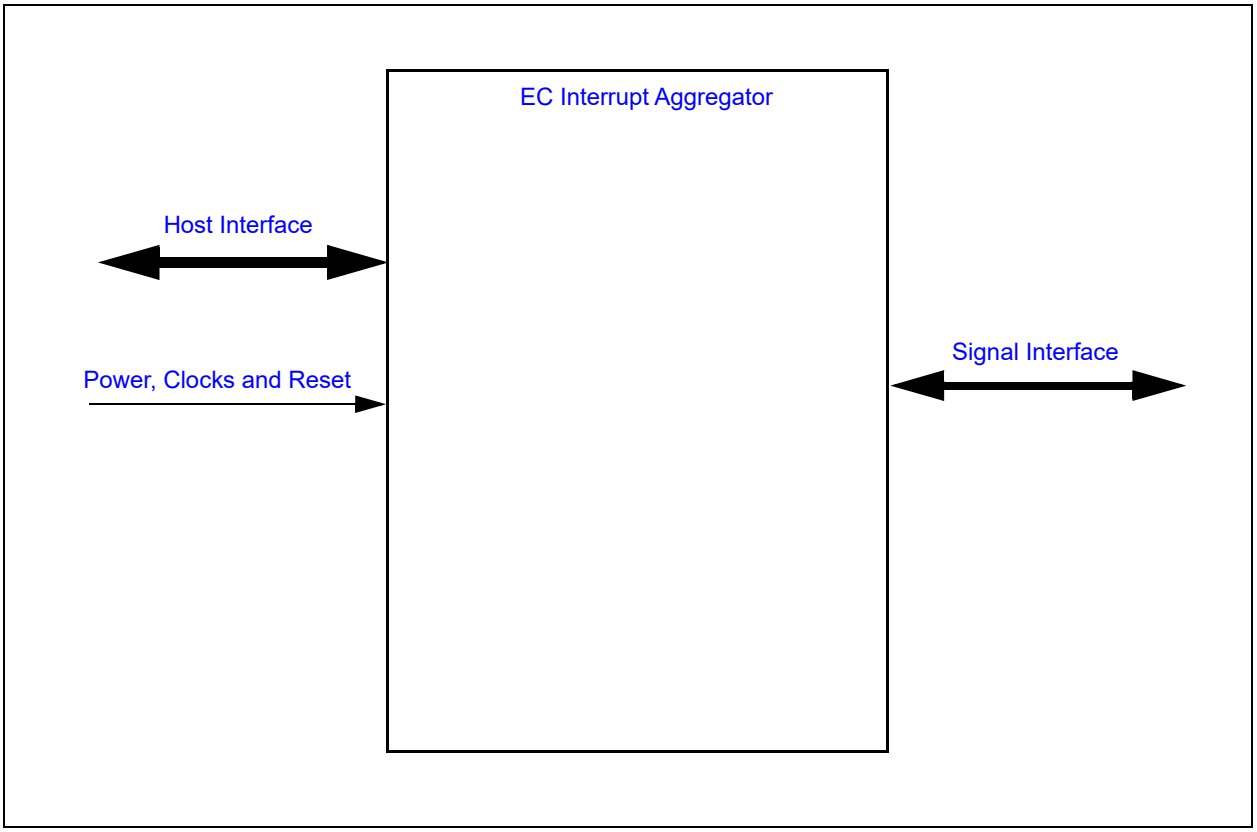
Interrupts classified as wake events can be recognized without a running clock, e.g., while the MEC152x is in sleep state.

This chapter focuses on the EC Interrupt Aggregator. Please refer to embedded controller's documentation for more information on interrupt and exception handling.

8.2 Interface

This block is designed to be accessed internally via a registered host interface. The following diagram illustrates the various interfaces to the block.

FIGURE 8-1: EC INTERRUPT AGGREGATOR INTERFACE DIAGRAM



8.3 Signal Description

8.3.1 SIGNAL INTERFACE

There are no external signals for this block.

8.4 Host Interface

The registers defined for the EC Interrupt Aggregator are only accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

8.5 Power, Clocks and Reset

8.5.1 BLOCK POWER DOMAIN

TABLE 8-1: BLOCK POWER

Power Well Source	Effect on Block
VTR_CORE	The EC Interrupt Aggregator block and registers operate on this single power well.

8.5.2 BLOCK CLOCKS

TABLE 8-2: CLOCK INPUTS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).

8.5.3 BLOCK RESET

TABLE 8-3: BLOCK RESETS

Reset Name	Reset Description
RESET_SYS	This signal is used to indicate when the VTR_CORE logic and registers in this block are reset.

8.6 Interrupts

This block aggregates all the interrupts targeted for the embedded controller into the Source Registers defined in Section 8.9, "EC Registers". The unmasked bits of each source register are then OR'd together and routed to the embedded controller's interrupt interface. The name of each Source Register identifies the IRQ number of the interrupt port on the embedded controller.

8.7 Low Power Modes

This block always automatically adjusts to operate in the lowest power mode by gating its clock when not required.

8.8 Description

The interrupt generation logic is made of groups of signals, each of which consist of a Status register, a Enable Set register, and Enable Clear register and a Result register.

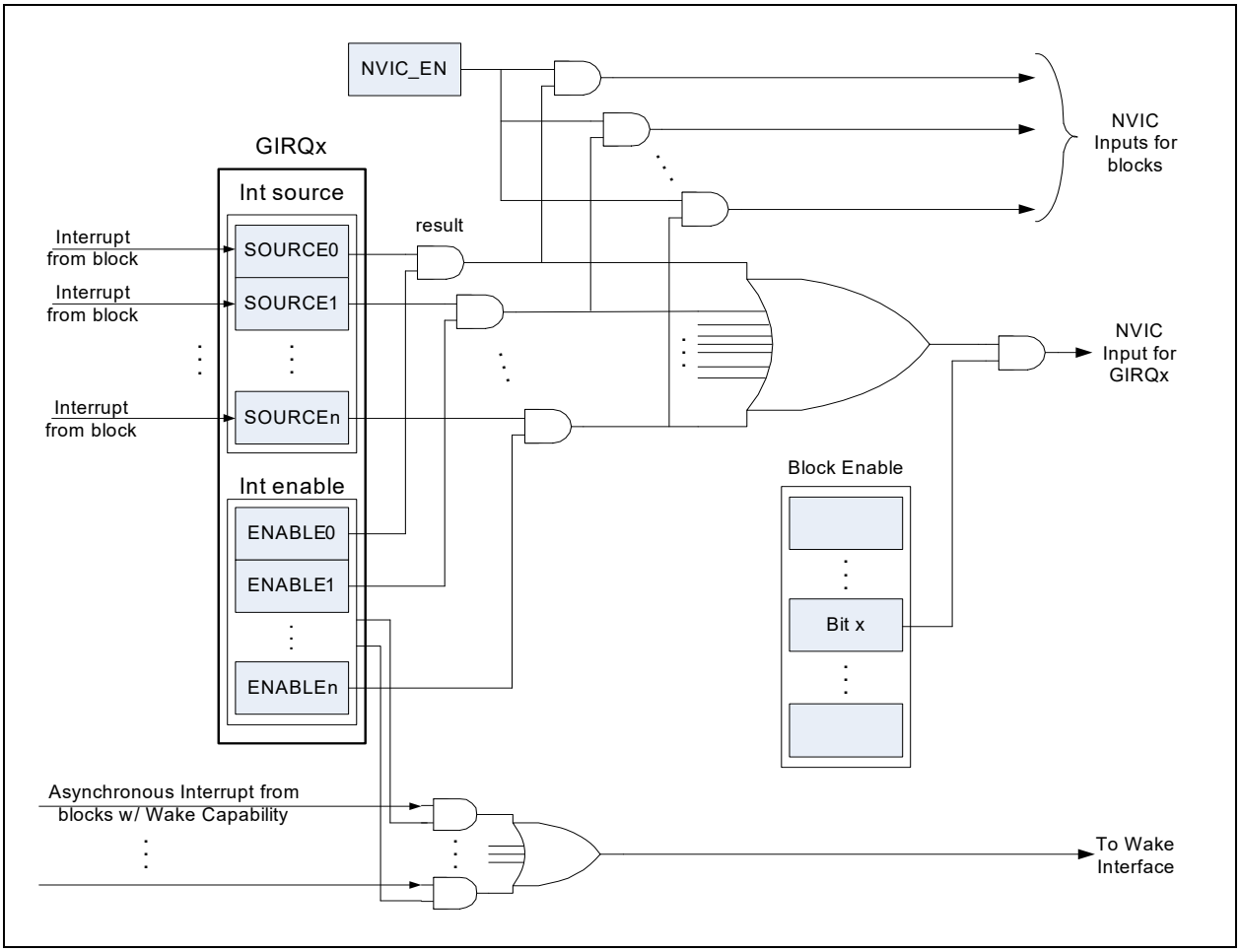
The Status and Enable are latched registers. There is one set of Enable register bits; both the Enable Set and Enable Clear registers return the same result when read. The Enable Set interface is used to set individual bits in the Enable register, and the Enable Clear is used to clear individual bits. The Result register is a bit by bit AND function of the Source and Enable registers. All the bits of the Result register are OR'ed together and AND'ed with the corresponding bit in the Block Select register to form the interrupt signal that is routed to the ARM interrupt controller.

The Result register bits may also be enabled to the NVIC block via the NVIC_EN bit in the Interrupt Control Register register. See Chapter 45.0, "EC Subsystem Registers"

Section 8.8.1 shows a representation of the interrupt structure.

© 2020 Microchip Technology Inc.

FIGURE 8-2: INTERRUPT STRUCTURE



8.8.1 AGGREGATED INTERRUPTS

All interrupts are routed to the ARM processor through the ARM Nested Vectored Interrupt Controller (NVIC). As shown in Figure 8-2, "Interrupt Structure", all interrupt sources are aggregated into the GIRQx Source registers. In many cases, the Result bit for an individual interrupt source is tied directly to the NVIC. These interrupts are shown in the “Direct NVIC” column in the Interrupt Bit Assignments table. In addition, all GIRQx can also generate an interrupt to the NVIC when any of the enabled interrupts in its group is asserted. The NVIC vectors for the aggregated GIRQ interrupts are shown tin the “Agg NVIC” column.

Firmware should not enable the group GIRQ NVIC interrupt at the same time individual direct interrupts for members of the group are enabled. If both are enabled, the processor will receive two interrupts for an event, one from the GIRQ and one from the direct interrupt.

Note:	The four Soft Interrupts that are defined by the RTOS Timer do not have individual NVIC vectors. If the use of the SWI interrupts is required, then all interrupts in the GIRQ must disable the individual NVIC vectors.
Note:	These four Soft Interrupts are only available in aggregate mode

8.8.2 WAKE GENERATION

Wake-capable interrupts are listed in Table 3-3, "GIRQ_mapping" with a designation of 'Yes' in the Wake Event column. All interrupts, except GIRQ22, generate an EC Interrupt event. They are routed to source bits that are synchronized to the 32 MHz Ring Oscillator. If enabled, the Interrupt Result is fed into the Priority Encoder/Decision Logic, which generates the interrupt vector to the NVIC Interrupt Interface.

Some Interrupts, which are labeled Wake-Capable, are also routed as Wake Events to the Chip's Wake Logic. These are asynchronous events that are used to resume the 32 MHz Ring Oscillator operation from a sleep state and wake the processor.

8.8.2.1 Wake Capable Interrupts

All GPIO inputs are wake-capable. In order for a GPIO input to wake the MEC152x from a sleep state, the Interrupt Detection field of the GPIO Pin Control Register must be set to Rising Edge Triggered, Falling Edge Triggered, or Either Edge Triggered. If the Interrupt Detection field is set to any other value, a GPIO input will not trigger a wake interrupt.

Some of the Wake Capable Interrupts are triggered by activity on pins that are shared with a GPIO. These interrupts will only trigger a wake if the Interrupt Detection field of the corresponding GPIO Pin Control Register is set to Rising Edge Triggered, Falling Edge Triggered, or Either Edge Triggered.

8.8.2.2 Wake-Only Events

Some devices which respond to an external master require the 48MHz clock domain to operate but do not necessarily require and immediate processing by the EC. Wake-only events provide the means to start the 48MHz clock domain without triggering an EC interrupt service routine. This events are grouped into a single GIRQ, GIRQ22. Events that are enabled in that GIRQ will start the clock domain when the event occurs, but will not invoke an EC interrupt. The SLEEP_ENABLE flags all remain asserted. If the activity for the event does not in turn trigger another EC interrupt, the CLOCK_REQUIRED for the block will re-assert and the configured sleep state will be re-entered.

Note:	For example, when RSMRST is high and there is a desire to wake from an ESPI cycle, GIRQ22[9] is the correct wake source to use. When Chip is asleep and there is a ESPI cycle, the falling edge of the CS will cause the chips clock to turn on the ESPI block, but not the processor itself. Upon conclusion of the ESPI cycle, if no ESPI interrupt was generated (i.e. most cycles), then the clock to the ESPI block will go off, and the chip will go back to sleep. If the ESPI cycle creates an interrupt to the processor (i.e. downstream wire or downstream OOB packet for example), then an processor interrupt will be generated if enabled and the clock will remain on and the processor can service the interrupt and the processor can put the chip back to sleep when it has completed its work.

8.8.3 INTERRUPT SUMMARY

Interrupt bit assignments, including wake capabilities and NVIC vector locations, are shown in the Interrupt Aggregator Bit Assignments Table in Section 3.0, "Device Inventory". The table lists all possible interrupt sources; the register bits for any interrupt source, such as a GPIO, that is not implemented in a particular part are reserved.

8.8.4 DISABLING INTERRUPTS

The Block Enable Clear Register and Block Enable Set Register should not be used for disabling and enabling interrupts for software operations i.e., critical sections. The ARM enable disable mechanisms should be used.

© 2020 Microchip Technology Inc.

8.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for of the EC Interrupt Aggregator Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 8-4: REGISTER SUMMARY

Offset	Register Name
00h	GIRQ8 Source Register
04h	GIRQ8 Enable Set Register
08h	GIRQ8 Result Register
0Ch	GIRQ8 Enable Clear Register
10h	Reserved
14h	GIRQ9 Source Register
18h	GIRQ9 Enable Set Register
1Ch	GIRQ9 Result Register
20h	GIRQ9 Enable Clear Register
24h	Reserved
28h	GIRQ10 Source Register
2Ch	GIRQ10 Enable Set Register
30h	GIRQ10 Result Register
34h	GIRQ10 Enable Clear Register
38h	Reserved
3Ch	GIRQ11 Source Register
40h	GIRQ11 Enable Set Register
44h	GIRQ11 Result Register
48h	GIRQ11 Enable Clear Register
4Ch	Reserved
50h	GIRQ12 Source Register
54h	GIRQ12 Enable Set Register
58h	GIRQ12 Result Register
5Ch	GIRQ12 Enable Clear Register
60h	Reserved
64h	GIRQ13 Source Register
68h	GIRQ13 Enable Set Register
6Ch	GIRQ13 Result Register
70h	GIRQ13 Enable Clear Register
74h	Reserved
78h	GIRQ14 Source Register
7Ch	GIRQ14 Enable Set Register
80h	GIRQ14 Result Register
84h	GIRQ14 Enable Clear Register
88h	Reserved
8Ch	GIRQ15 Source Register
90h	GIRQ15 Enable Set Register
94h	GIRQ15 Result Register
98h	GIRQ15 Enable Clear Register
9Ch	Reserved

TABLE 8-4: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
A0h	GIRQ16 Source Register
A4h	GIRQ16 Enable Set Register
A8h	GIRQ16 Result Register
ACh	GIRQ16 Enable Clear Register
B0h	Reserved
B4h	GIRQ17 Source Register
B8h	GIRQ17 Enable Set Register
BCh	GIRQ17 Result Register
C0h	GIRQ17 Enable Clear Register
C4h	Reserved
C8h	GIRQ18 Source Register
CCh	GIRQ18 Enable Set Register
D0h	GIRQ18 Result Register
D4h	GIRQ18 Enable Clear Register
D8h	Reserved
DCh	GIRQ19 Source Register
E0h	GIRQ19 Enable Set Register
E4h	GIRQ19 Result Register
E8h	GIRQ19 Enable Clear Register
ECh	Reserved
F0h	GIRQ20 Source Register
F4h	GIRQ20 Enable Set Register
F8h	GIRQ20 Result Register
FCh	GIRQ20 Enable Clear Register
100h	Reserved
104h	GIRQ21 Source Register
108h	GIRQ21 Enable Set Register
10Ch	GIRQ21 Result Register
110h	GIRQ21 Enable Clear Register
114h	Reserved
118h	GIRQ22 Source Register
11Ch	GIRQ22 Enable Set Register
120h	GIRQ22 Result Register
124h	GIRQ22 Enable Clear Register
128h	Reserved
12Ch	GIRQ23 Source Register
130h	GIRQ23 Enable Set Register
134h	GIRQ23 Result Register
138h	GIRQ23 Enable Clear Register
13Ch	Reserved
140h	GIRQ24 Source Register
144h	GIRQ24 Enable Set Register
148h	GIRQ24 Result Register
14Ch	GIRQ24 Enable Clear Register

© 2020 Microchip Technology Inc.

TABLE 8-4: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
150h	Reserved
154h	GIRQ25 Source Register
158h	GIRQ25 Enable Set Register
15Ch	GIRQ25 Result Register
160h	GIRQ25 Enable Clear Register
164h	Reserved
168h	GIRQ26 Source Register
16Ch	GIRQ26 Enable Set Register
170h	GIRQ26 Result Register
174h	GIRQ26 Enable Clear Register
200h	Block Enable Set Register
204h	Block Enable Clear Register
208h	Block IRQ Vector Register

All of the GIRQx Source, Enable Set, Enable Clear and Result registers have the same format. The following tables define the generic format for each of these registers. The bit definitions are defined in the sections that follow.

The behavior of the enable bit controlled by the GIRQx Enable Set and GIRQx Enable Clear Registers, the GIRQx Source bit, and the GIRQx Result bit is illustrated in Section 8.8.1, "Aggregated Interrupts".

8.9.1 GIRQ SOURCE REGISTERS

All of the GIRQx Source registers have the same format. The following table defines the generic format for each of these registers. The bit definitions are defined in the Interrupt Aggregator Bit Assignments Table in Section 3.0, "Device Inventory". Unassigned bits are Reserved and return 0.

Note:	If a GPIO listed in the tables does not appear in the pin list of a particular device, then the bits for that GPIO in the GIRQx Source, GIRQx Enable Clear, GIRQx Enable Set and GIRQx Result are reserved.

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	GIRQX_SOURCE The GIRQx Source bits are R/WC sticky status bits indicating the state of interrupt before the interrupt enable bit.	R/WC	0h	RESET_SYS

8.9.2 GIRQ ENABLE SET REGISTERS

All of the GIRQx Enable Set registers have the same format. The following table defines the generic format for each of these registers. Unassigned bits are Reserved and return 0.

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	GIRQX_ENABLE_SET Each GIRQx bit can be individually enabled to assert an interrupt event. Reads always return the current value of the internal GIRQX_ENABLE bit. The state of the GIRQX_ENABLE bit is determined by the corresponding GIRQX_ENABLE_SET bit and the GIRQX_ENABLE_CLEAR bit. (0=disabled, 1-enabled) 1=The corresponding interrupt in the GIRQx Source Register is enabled 0=No effect	R/WS	0h	RESET_SYS

8.9.3 GIRQ ENABLE CLEAR REGISTERS

All of the GIRQx Enable Clear registers have the same format. The following table defines the generic format for each of these registers. Unassigned bits are Reserved and return 0.

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	GIRQX_ENABLE_CLEAR Each GIRQx bit can be individually enabled to assert an interrupt event. Reads always return the current value of the internal GIRQX_ENABLE bit. The state of the GIRQX_ENABLE bit is determined by the corresponding GIRQX_ENABLE_SET bit and the GIRQX_ENABLE_CLEAR bit. (0=disabled, 1-enabled) 1=The corresponding interrupt in the GIRQx Source Register is disabled 0=No effect	R/WC	0h	RESET_SYS

© 2020 Microchip Technology Inc.

8.9.4 GIRQ RESULT REGISTERS

Offset	See Section 3.0, "Device Inventory"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	1h	-
30:0	GIRQX_RESULT The GIRQX_RESULT bits are Read-Only status bits indicating the state of an interrupt. The RESULT is asserted '1'b when both the GIRQX_SOURCE bit and the corresponding GIRQX_ENABLE bit are '1'b.	R	0h	RESET_SYS

8.9.5 BLOCK ENABLE SET REGISTER

Offset	200h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:27	Reserved	RES	-	-
26:8	IRQ_VECTOR_ENABLE_SET Each bit in this field enables the group GIRQ interrupt assertion to the NVIC. 1=Interrupts in the GIRQx Source Register may be enabled 0=No effect	R/WS	0h	RESET_SYS
7:0	Reserved	RES	-	-

8.9.6 BLOCK ENABLE CLEAR REGISTER

Offset	204h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:27	Reserved	RES	-	-
26:8	IRQ_VECTOR_ENABLE_CLEAR Each bit in this field disables the group GIRQ interrupt assertion to the NVIC. 1=Interrupts in the GIRQx Source Register are disabled 0=No effect	R/WC	0h	RESET_SYS
7:0	Reserved	RES	-	-

8.9.7 BLOCK IRQ VECTOR REGISTER

Offset	208h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:27	Reserved	RES	0h	-
26:8	IRQ_VECTOR Each bit in this field reports the status of the group GIRQ interrupt assertion to the NVIC. If the GIRQx interrupt is disabled as a group, by the Block Enable Clear Register, then the corresponding bit will be ‘0’b and no interrupt will be asserted.	R	0h	RESET_SYS
7:0	Reserved	RES	0h	-

© 2020 Microchip Technology Inc.

9.0 ENHANCED SERIAL PERIPHERAL INTERFACE (ESPI)

9.1 Introduction

The Intel® Enhanced Serial Peripheral Interface (eSPI) is Intel's successor to the Low Pin Count (LPC) bus, used in previous devices to provide System Host access to devices internal to the Embedded Controller. In addition, multiplexed

on the same physical pins, there are separate eSPI Channels: one transferring Host IRQs and other discrete pin inputs and outputs; another substituting for one or more SMBus channels; and another providing shared access to the BIOS

Flash memory, attached either to the Chipset or to the EC device.

The Peripheral Channel is the LPC replacement capability, which provides for I/O-Mapped and Memory-Mapped access to on-chip peripheral devices, with Plug-and-Play Configuration capability preserved. In addition, regions of the EC's internal memory space may be made available to the Host CPU directly, and Legacy DMA is replaced with Mastering capability so that the EC firmware may communicate with System DRAM.

The Virtual Wire Channel propagates IRQs to the Host system, replacing the Serial IRQ mechanism. It also provides a transport mechanism for other sideband signals such as SLP_Sx#, SMI#, SCI# and PLTRST#/PCI_RESET#.

The Out-of-Band (OOB) Channel provides a replacement for serial connections to the Chipset, replacing the SMLink1 port for PCH temperature and RTC readings, the PECI port for CPU temperature reading, and communication with the PCH's Management Engine for other purposes previously performed over SMBus.

The Flash Channel performs BIOS Flash Memory sharing. In Master-Attached (MAFS) configuration, the Chipset still connects to the Flash and shares with the EC over eSPI, also allowing the EC to load its firmware from it. In Slave-Attached (SAFS) configuration, the Flash is connected to the EC, which shares with the Chipset over eSPI, providing all information including low-level data such as Soft Straps and Management Engine firmware. Intel's Descriptor Mode protection mechanism is preserved in both configurations.

This chapter documents those registers whose offsets may change from product to product; especially those associated with Plug-and-Play configuration. For full details of the eSPI Block register set, see the Microchip document

"eSPI Controller with SAFS Support, Version 1.3" [1].

9.2 References

1. Microchip “eSPI Controller with SAFS Support, Version 1.3” Specification
2. Intel, *Enhanced Serial Peripheral Interface (eSPI): Interface Base Specification*
3. Intel, *eSPI Compatibility Specification for the specific Chipset*
4. Intel, *SPI Programming Guide for the specific Chipset*
5. The ROM Description Addendum document MEC152x

9.3 Terminology

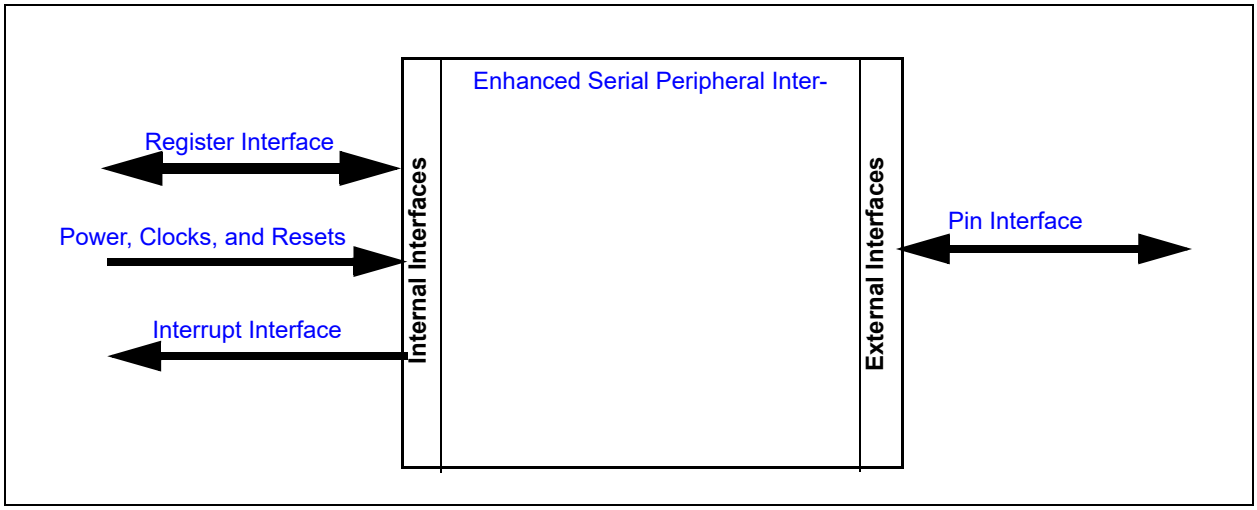
This table defines specialized terms localized to this feature.

TABLE 9-1: TERMINOLOGY

Term	Definition
System Host	Refers to the external CPU that communicates with this device via the eSPI Interface.
Logical Devices	Logical Devices are System Host-accessible features that are allocated a Base Address and range in the System Host I/O address space
Runtime Register	Runtime Registers are registers that are directly I/O accessible by the System Host via the eSPI interface.
Configuration Registers	Registers that are only accessible in CONFIG_MODE.
EC_Only Registers	Registers that are not accessible by the System Host. They are only accessible by an internal embedded controller.

9.4 Interface

FIGURE 9-1: Enhanced Serial Peripheral Interface (eSPI) INTERFACE DIAGRAM



9.4.1 PIN INTERFACE

Table 9-2, "Signal Description Table" lists the signals that are typically routed to the pin interface.

TABLE 9-2: SIGNAL DESCRIPTION TABLE

Signal Name	Direction	Description
eSPI_CS#	Input	eSPI Chip Select, Low-Active
eSPI_CLOCK	Input	eSPI Clock
eSPI_ALERT#	Output	eSPI Alert signal, Low-Active. Exercised only if ALERT# is configured to be presented separately from the IO1 pin.
eSPI_RESET#	Input	POR for eSPI bus power domain, and a Reset for serious errors. Low-Active.
eSPI_IO0	Input/Output	eSPI Data Bus, bit 0. Input (MOSI) in x1 Bus Mode. Else, it holds the LS data bit.
eSPI_IO1	Input/Output	eSPI Data Bus, bit 1. Output (MISO) in x1 Bus Mode. Also, by default, presents ALERT# state between frames.
eSPI_IO2	Input/Output	eSPI Data Bus, bit 2. Used only in x4 mode.
eSPI_IO3	Input/Output	eSPI Data Bus, bit 3. Used only in x4 mode, as MS bit.

9.4.2 REGISTER INTERFACE

Each of the four channels contains registers that may be accessed by both the Host and the EC, as well as a set of registers that can only be accessed by the EC.

9.4.3 POWER, CLOCKS, AND RESETS

This section defines the Power, Clock, and Reset parameters associated with this IP block.

© 2020 Microchip Technology Inc.

9.4.3.1 Power

Name	Description
VTR_CORE	The Enhanced Serial Peripheral Interface (eSPI) block and registers are powered by VTR_CORE. This power rail may be present to the block while external power to the eSPI pins is not present. Therefore, this block remains passive on the eSPI bus pins whenever eSPI_RESET# is low.

9.4.3.2 Clocks

This section describes all the clocks in the block, including those that are derived from the I/O Interface as well as the ones that are derived or generated internally.

Name	Description
48MHz	The main internal clock
eSPI_CLOCK	The eSPI clock provided by the System Host core logic **Note:** Max frequency supported is 50MHz.

9.4.3.3 Resets

This section describes all the resets associated with this IP block, including those that are derived from the I/O Interface as well as the ones that are derived or generated internally.

Name	Description
RESET_SYS	This is the power-on-reset signal, which is asserted when VTR_CORE power is applied. Asserting this reset signal resets the eSPI IP block, including all registers, FIFOs, and state machines to their initial POR state.
RESET_eSPI	A general reset signal for the eSPI block. This reset is asserted with the eSPI_RESET# pin is asserted by the System Host core logic. When this reset is asserted all eSPI Output signals and Input/Output signals are tri-stated. Any transaction in progress is terminated and all FIFOs are flushed. All interrupt status flags are reset and all interrupts to the EC except RESET_eSPI are suppressed. When this reset is asserted, all eSPI Configuration Registers in the slave device are set to the default values, as per the Intel eSPI Specification. Fields in the eSPI Configuration Registers that are set from the eSPI Capabilities registers (see Section 9.7, "eSPI Register Summary") are not modified. This reset is also asserted in the following cases: RESET_SYS is asserted
RESET_VCC	Performs a reset when the system main power rail is turned off.
RESET_HOST	Performs a reset when the system main power rail is turned off or when the system host resets the Host Interface.
eSPI_PLTRST#	This is a reset that affects the Peripheral Channel. It is received by the Slave as a Virtual Wire (PLTRST#) or through PCI_RESET# pin. This reset is also asserted in the following cases: • RESET_SYS is asserted • RESET_eSPI is asserted • VCC_PWRGD is de-asserted • The Peripheral Channel is disabled

9.4.4 INTERRUPT INTERFACE

This section defines the interrupt Interface signals routed to the chip interrupt aggregator.

Source	Description
Wake Only Event	
ESPI_WAKE_ONLY	This signal is asserted when the eSPI interface detects eSPI traffic. If enabled, it may be used to wake the main clock domain when the chip is in a sleep state.
Peripheral Channel	
INTR_PC	Peripheral Channel Interrupt
INTR_BM1	Bus Mastering Channel 1 Interrupt
INTR_BM2	Bus Mastering Channel 2 Interrupt
INTR_LTR	Peripheral Message (LTR) Interrupt
OOB Channel	
INTR_OOB_UP	Out of Band Channel Up Interrupt
INTR_OOB_DOWN	Out of Band Channel Down Interrupt
Flash Channel	
INTR_FLASH	Flash Channel Interrupt
EC_CMPLTN	EC Completion Event Interrupt-SAF mode
ESPI_ERROR	ESPI Error Event Interrupt-SAF mode
Virtual Wires Channel	
MSVW[00:10]_SRC[3:0]	Master-to-Slave Virtual Wire Interrupts
eSPI Global	
eSPI_RESET	eSPI Reset Interrupt This interrupt is generated whenever the external eSPI_RESET# pin changes state.

9.5 Low Power Modes

The eSPI block can enter a low power state when it is not in operation. When the eSPI block is operational it will keep the main system clock from shutting down and entering its sleep state. When the eSPI_CS# pin is asserted the eSPI block will wake the main system clock, if it is in a sleep state, and keep the system clock in its active state until the transaction started by the Master has completed.

The low power behavior of the block is controlled by the BAR Inhibit Register. The block is not affected by a SLEEP_ENABLE signal from the chip's Power, Clocks and Resets unit.

9.6 Description

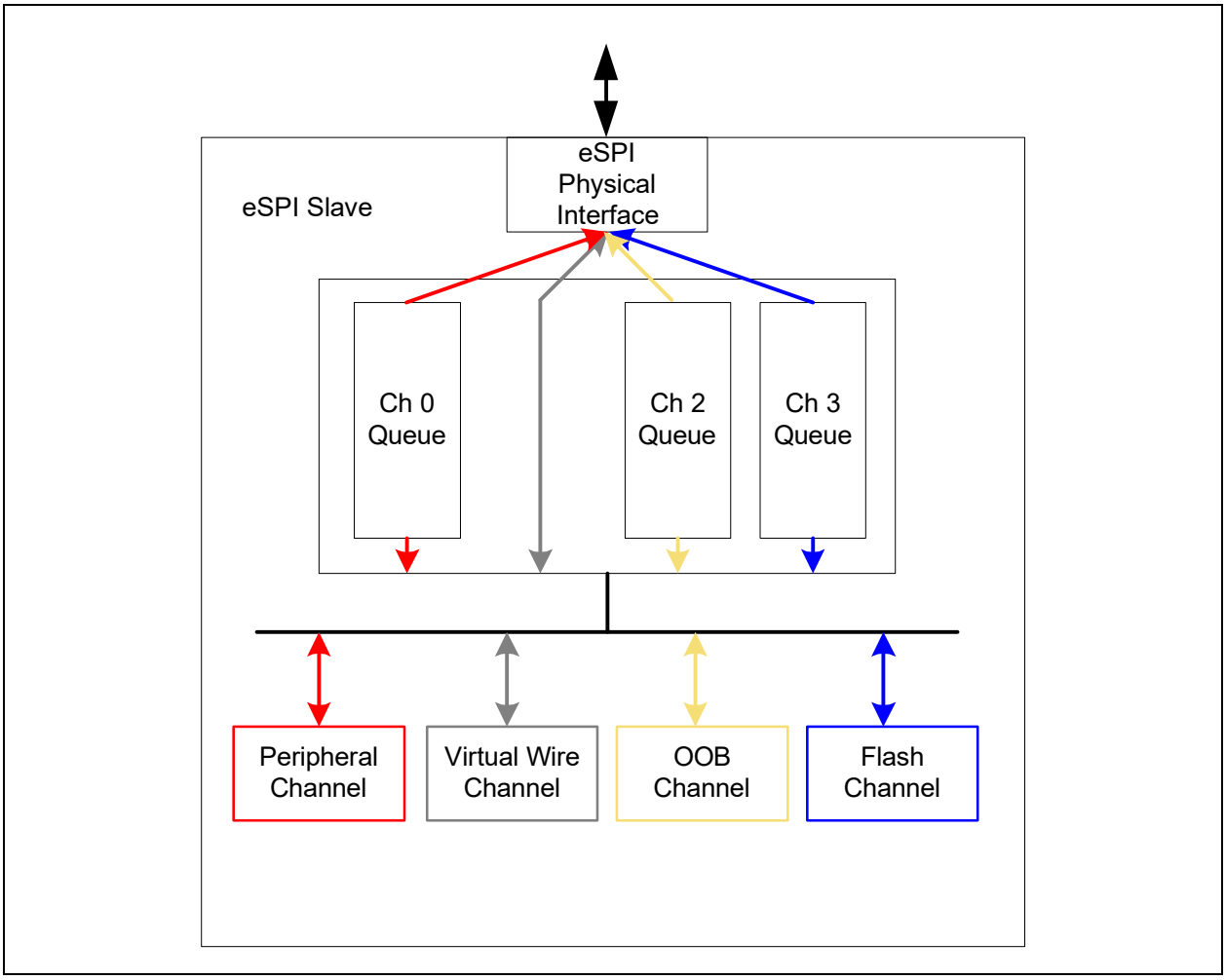
The eSPI interface consists of four channels:

- eSPI Peripheral Channel Interface
- eSPI Out Of Band Channel Interface
- eSPI Flash Channel Interface
- eSPI Virtual Wires Interface

These four channels are multiplexed onto the eSPI physical interface that connects the Embedded Controller device with the core logic of the Host. The following figure illustrates this multiplexing:

© 2020 Microchip Technology Inc.

FIGURE 9-2: ESPI BLOCK DIAGRAM



The Peripheral Channel (PC) enables the system Host to read and write locations inside the EC. The PC encapsulates legacy I/O and Memory-Mapped I\O operation as well as generic memory read and write operations in both directions.

EC can take control of Peripheral Channel and become the Bus Master. In this mode EC can transmit bulk data of size up to 4K to host memory directly. eSPI peripheral channel Interface chapter in eSPI block spec describes the settings for this mode.

The Virtual Wire Channel provides in-band emulation of sideband pin signals between the system Core Logic and the EC, including the legacy SERIRQ interrupt link to the system Host.

The Out of Band (OOB) Channel enables messaging between the Out-Of-Band Processor in the system chipset and the EC. This messaging is implemented by tunneling SMBus packets over the eSPI port.

The Flash Channel allows sharing of a Flash memory between the system Host and the EC. The Flash may be connected to either side (Host or EC), and is shared over eSPI with the other. A strap pin on the Chipset defines which direction the Flash traffic follows.

9.7 eSPI Register Summary

The following sections list the registers associated with the eSPI logic that may vary among products. The eSPI logic requires two Logical Devices in order to provide access to all the required registers. These Logical Devices are called the I/O Component and the Memory Component. The Base Addresses for these blocks are shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

The EC may access all registers in both Logical Devices. Host access is restricted to three ranges: Runtime Registers, located at offsets 00h through FFh from the Logical Device Base Address of the I/O Component, and Configuration Registers, located at offsets 330h through 3FFh in both the I/O Component and the Memory Component. The Runtime Registers may be mapped into the Hosts address space, either I/O or Memory, by setting the associated BAR for the Logical Device. The Configuration Registers are accessed through the Configuration Port. Registers located at offsets 330h through 3FFh are mapped to Configuration Port offsets 30h through FFh. Configuration Port offsets 00h through 2Fh, for all Host Logical Devices, are mapped to the Global Configuration Registers.

9.7.1 REGISTER SPACES

Each of the register spaces (eSPI I/O Component, eSPI Memory Component, Virtual Wire Component, SAF Communication and SAF Bridge Component) are assigned base addresses.

The base address of each of the eSPI components is shown in Table 3-1, "Base address".

9.7.2 ESPI I/O COMPONENT

TABLE 9-3: ESPI I/O COMPONENT REGISTER SUMMARY

INDEX 2	Host Offset 3	EC Offset 1	Register Name
RUNTIME REGISTERS			
Peripheral Channel			
-	00h	00h	INDEX Register
-	01h	01h	DATA Register
EC PRIVATE REGISTERS			
Peripheral Channel			
	-	120h	BAR Inhibit Register
	-	128h	eSPI Bar Init Register
	-	134h - 1A7h	I/O Base Address Register Format, Internal Component See Table 9-6, "ESPI I/O Base Address Register Default Values" Note
Note: Please refer to Microchip “eSPI Controller with SAFS Support, Version 1.3” Specification [1.] for the complete register set			
CONFIGURATION REGISTERS			
Peripheral Channel			
30h	-	330h	eSPI Activate Register
34h - 8Ch	-	334h - 38Ch	I/O Space Base Address Registers (BARs) See Table 9-6, "ESPI I/O Base Address Register Default Values"
Virtual Wire Channel			
ACh - BFh	-	3ACh - 3BFh	IRQ Selection. See Table 9-11, "IRQ Assignment Table"
C0h-EFh	-	3C0h-3EFh	Reserved
F0h	-	3F0h	eSPI Virtual Wire Errors

Note 1: eSPI Interface IO Component Base Address is defined in Device Inventory chapter.

2: Value is written to the INDEX Register.

3: Value offset from eSPI I/O Component. See Table 9-6, "ESPI I/O Base Address Register Default Values" for details.

© 2020 Microchip Technology Inc.

9.7.3 ESPI, MEMORY COMPONENT REGISTERS

TABLE 9-4: ESPI MEMORY COMPONENT REGISTER SUMMARY

INDEX 2	EC Offset 1	Host Offset
EC PRIVATE REGISTERS		
Peripheral Channel		
-	130h-164h	Memory Base Address Register, Internal Component See Table 9-7, "ESPI Memory Base Address Register Default Values"
-	1ACh-1FFh	SRAM Base Address Register Format, Internal Component See Table 9-8, "SRAM Base Address Register Default Values, Host Config" Note
CONFIGURATION REGISTERS		
30h-A7h	330h-3A7h	Memory Base Address Configuration Register See Table 9-7, "ESPI Memory Base Address Register Default Values"
A8h	3A8	Host MEM BAR Extend Register (16 bits)
ACh-BFh	3ACh-3BFh	SRAM Base Address Configuration Register See Table 9-10, "SRAM Base Address Register Default Values, EC-Only"
C0h - F7h	3C0h - 3F7h	Reserved
FCh	3FCh	SRAM BAR Extend Register (16 bits)

Note 1: eSPI Interface IO Component Base Address is defined in Device Inventory chapter.

2: Value is written to the Configuration Port INDEX Register.

9.7.4 VIRTUAL WIRE REGISTERS

The following registers are allocated in the Virtual Wire Component space. The MSVWxx registers hold Master-to-Slave Virtual Wires transmitted from the Chipset to the EC. The SMVWxx registers hold Slave-to-Master Virtual Wires transmitted from the EC to the Chipset.

Their offsets may differ from product to product. Their formats, and the assignments of the Virtual Wires to register bits, are defined in the Microchip document "eSPI Controller with SAFS Support, Version 1.3". Refer to the registers by name there, which will match the names defined here.

TABLE 9-5: VIRTUAL WIRES REGISTER SUMMARY

EC Offset	Register Name
0h	MSVW00 Register
Ch	MSVW01 Register
18h	MSVW02 Register
24h	MSVW03 Register
30h	MSVW04 Register
3Ch	MSVW05 Register
48h	MSVW06 Register
54h	MSVW07 Register
60h	MSVW08 Register
6Ch	MSVW09 Register
78h	MSVW10 Register
84h - 1FFh	Reserved
200h	SMVW00 Register

TABLE 9-5: VIRTUAL WIRES REGISTER SUMMARY (CONTINUED)

EC Offset	Register Name
208h	SMVW01 Register
210h	SMVW02 Register
218h	SMVW03 Register
220h	SMVW04 Register
228h	SMVW05 Register
230h	SMVW06 Register
238h	SMVW07 Register
240h	SMVW08 Register
248h	SMVW09 Register
250h	SMVW10 Register
242h - 3FFh	Reserved

9.8 Base Address Register Tables

TABLE 9-6: ESPI I/O BASE ADDRESS REGISTER DEFAULT VALUES

Host Config Index	EC Offset	Logical Device	Reset Default	EC-Only Offset	Reset Default	LDN	MASK
34h	334h	eSPI I/O Component (Configuration Port)	XXXX_0000h 1	134h	0000_0D01h	Dh	1h
38h	338h	eSPI Memory Component	0000_0000h	138h	0000_0E00h	Eh	0h
3Ch	33Ch	Mailbox	0000_0000h	13Ch	0000_0001h	0h	1h
40h	340h	8042 Emulated Key-board Controller	0060_0000h	140h	0000_0104h	1h	4h
44h	344h	ACPI EC Channel 0	0062_0000h	144h	0000_0204h	2h	4h
48h	348h	ACPI EC Channel 1	0000_0000h	148h	0000_0307h	3h	7h
4Ch	34Ch	ACPI EC Channel 2	0000_0000h	14Ch	0000_0407h	4h	7h
50h	350h	ACPI EC Channel 3	0000_0000h	150h	0000_0507h	5h	7h
54h	Reserved						
58h	358h	ACPI PM1	0000_0000h	158h	0000_0707h	7h	7h
5Ch	35Ch	Legacy (Fast Keyboard)	0092_0000h	15Ch	0000_0800h	8h	
60h	360h	UART 0	0000_0000h	160h	0000_0907h	9h	7h
64h	364h	UART 1	0000_0000h	164h	000_0A07h	Ah	7h
68h	368h	Embedded Memory Interface (EMI) 0	0000_0000h	168h	0000_100Fh	10h	Fh
6Ch	36Ch	Embedded Memory Interface (EMI) 1	0000_0000h	16Ch	0000_110Fh	11h	Fh
70h	370h	Reserved					
74h	374h	BIOS Debug Port (Port 80) 0	0000_0000h	174h	0000_2000h	20h	0h
78h	378h	BIOS Debug Port (Port 80) 1	0000_0000h	178h	0000_2100h	21h	0h
7Ch	37Ch	RTC	0000_0000h	17Ch	0000_141Fh	14h	1Fh
84h	384h	32 Byte Test Block	0000_0000h	184h	0000_2F1Fh	2Fh	1Fh
88h	388h	UART 2	0000_0000h	188h	0000_0B07h	Bh	7h

© 2020 Microchip Technology Inc.

Note 1: The higher 16bits (XXXX) come from the BAR INIT Register. See Microchip “eSPI Controller with SAFS Support, Version 1.3” Specification for details.

TABLE 9-7: ESPI MEMORY BASE ADDRESS REGISTER DEFAULT VALUES

Host Config Index	EC Offset	Logical Device	Reset Default	EC-Only Offset	Reset Default	LDN	MASK
30h	330h	Mailbox	00_0000h	130h	0000_0001h	0h	1h
3Ah	33Ah	ACPI EC Channel 0	62_0000h	13Ah	0000_0204h	2h	4h
44h	344h	ACPI EC Channel 1	00_0000h	144h	0000_0307h	3h	7h
4Eh	34Eh	ACPI EC Channel 2	00_0000h	14Eh	0000_0407h	4h	7h
58h	358h	ACPI EC Channel 3	00_0000h	158h	0000_0507h	5h	7h
62h	362h	Reserved					
6Ch	36Ch	Embedded Memory Interface (EMI) 0	00_0000h	16Ch	0000_100Fh	10h	Fh
76h	376h	Embedded Memory Interface (EMI) 1	00_0000h	176h	0000_110Fh	11h	Fh
80h	380h	Reserved					
8Ah	38Ah	32 Byte Test Block	00_0000h	18Ah	0000_2F1Fh	2Fh	1Fh

TABLE 9-8: SRAM BASE ADDRESS REGISTER DEFAULT VALUES, HOST CONFIG

Host Config Index	Logical Device	Reset Default	Host Address [47:16]	Size [7:4]	Access [2:1]
ACh	SRAM BAR 0	0h	0h	0h	0h
B6h	SRAM BAR 1	0h	0h	0h	0h

Note: The Host Address field will be zero by default until the Host writes something there, but the Size and Access fields (and therefore the Reset Default) may be re-initialized by Firmware and seen as non-zero values (read-only) by the Host.

TABLE 9-9: HOST ADDRESS EXTEND REGISTER DEFAULT VALUES, HOST CONFIG

Host Config Index	Memory / SRAM BAR Extend	Reset Default
3A8h	HOST MEM BAR	0h
3FCh	SRAM BAR	0h

TABLE 9-10: SRAM BASE ADDRESS REGISTER DEFAULT VALUES, EC-ONLY

EC Offset	Logical Device	Reset Default	Base Address [47:16]	Size [7:4]	Access [2:1]	Valid [0]
1ACh	SRAM BAR 0	00h	0h	0h	0h	0h
1B6h	SRAM BAR 1	00h	0h	0h	0h	0h

9.9 IRQ Table

TABLE 9-11: IRQ ASSIGNMENT TABLE

Host Config Index	EC Offset	Instance Name	Instance Number	Interrupt Source	Default Value
ACh	3ACh	Mailbox	0	MBX_Host_SIRQ	FFh
ADh	3ADh	Mailbox	0	MBX_Host_SMI	FFh
AEh	3AEh	8042	0	KIRQ	FFh
AFh	3AFh	8042	0	MIRQ	FFh
B0h	3B0h	ACPI EC	0	EC_OBF	FFh
B1h	3B1h	ACPI EC	1	EC_OBF	FFh
B2h	3B2h	ACPI EC	2	EC_OBF	FFh
B3h	3B3h	ACPI EC	3	EC_OBF	FFh
B4h	Reserved				
B5h	3B5h	UART	0	UART	FFh
B6h	3B6	UART	1	UART	FFh
B7h	3B7h	EMI	0	Host Event	FFh
B8h	3B8h	EMI	0	EC-to-Host	FFh
B9h	3B9h	EMI	1	Host Event	FFh
BAh	3BAh	EMI	1	EC-to-Host	FFh
BBh	Reserved				
BCh	Reserved				
BDh	3BDh	RTC	0	RTC	FFh
BEh	3BEh	EC	0	EC_IRQ	FFh
BFh	3BFh	UART	2	UART	FFh

9.10 Virtual Wires Table

TABLE 9-12: MASTER-TO-SLAVE VIRTUAL WIRE REGISTERS

Offset	Instance Name	Default Value
0h	MSVW00	00000000_04040404_00000002h
Ch	MSVW01	00000000_04040404_00000003h
18h	MSVW02	00000000_04040404_00000307h
24h	MSVW03	00000000_04040404_00000041h
30h	MSVW04	00000000_04040404_00000042h
3Ch	MSVW05	00000000_04040404_00000043h
48h	MSVW06	00000000_04040404_00000044h
54h	MSVW07	00000000_04040404_00000347h

© 2020 Microchip Technology Inc.

TABLE 9-12: MASTER-TO-SLAVE VIRTUAL WIRE REGISTERS

Offset	Instance Name	Default Value
60h	MSVW08	00000000_04040404_0000004Ah
6Ch	MSVW09	00000000_04040404_00000000h
78h	MSVW10	00000000_04040404_00000000h

TABLE 9-13: SLAVE-TO-MASTER VIRTUAL WIRE REGISTERS

Offset	Instance Name	Default Value
200h	SMVW00	01010000_0000C004h
208h	SMVW01	00000000_00000005h
210h	SMVW02	00010101_00007306h
218h	SMVW03	00000000_00000040h
220h	SMVW04	00000000_00000045h
228h	SMVW05	00000000_00000046h
230h	SMVW06	00000000_00000000h
238h	SMVW07	00000000_00000000h
240h	SMVW08	00000000_00000000h
248h	SMVW09	00000000_00000000h
250h	SMVW10	00000000_00000000h

10.0 CHIP CONFIGURATION

10.1 Introduction

This chapter defines the mechanism to configure the device. Each logical device or block in the design has their own set of configuration registers. The Global Configuration Registers are use for chip-level configuration. The chip's Device ID and Revision are located in the Global Configuration space and may be used to uniquely identify this chip.

10.2 Terminology

This section documents terms used locally in this chapter. Common terminology that is used in the chip specification is captured in the Chip-Level Terminology section.

TABLE 10-1: TERMINOLOGY

Term	Definition
Global Configuration Registers	Registers used to configure the chip that are always accessible via the Configuration Port
Logical Device Configuration Registers	Registers used to configure a logical device in the chip. These registers are only accessible via the Configuration Port when enabled via the Global Configuration registers.

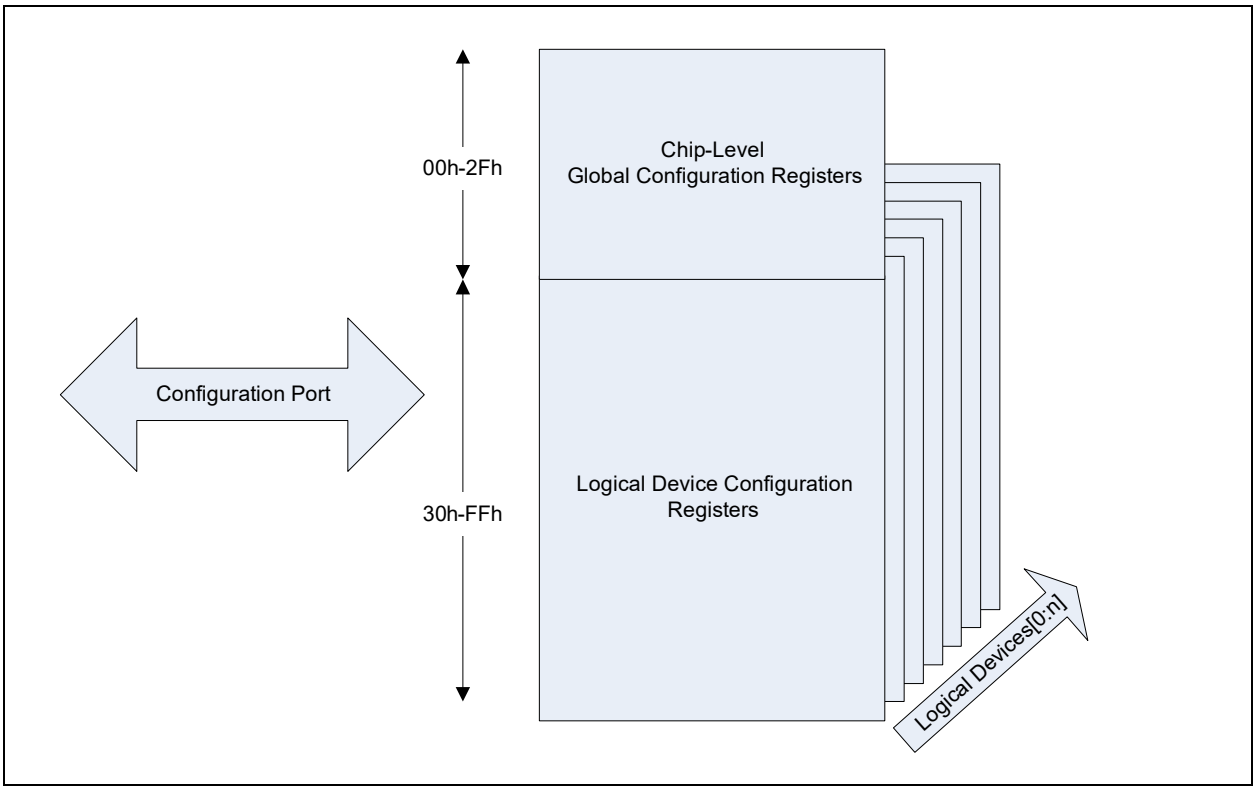
10.3 Interface

This block is designed to be accessed via the Host accessible Configuration Port.

© 2020 Microchip Technology Inc.

10.3.1 HOST INTERFACE

FIGURE 10-1: BLOCK DIAGRAM OF CONFIGURATION PORT



Note:	Each logical device has a bank of Configuration registers that are accessible at offsets 30h to FFh via the Configuration Port. The Logical Device number programmed in offset 07h determines which bank of configuration registers is currently accessible.

The registers defined for the Chip Configuration are accessible by the Configuration Port when the device is in CONFIG MODE. For a description of the Configuration Port and CONFIG MODE see the description of the eSPI Interface.

10.4 Power, Clocks and Reset.

This section defines the Power, Clock, and Reset input parameters to this block.

10.4.1 POWER DOMAINS

TABLE 10-2: POWER SOURCES

Name	Description
VTR_CORE	The logic and registers implemented in this block reside on this single power well.

10.4.2 CLOCK INPUTS

This block does not require any special clock inputs.

10.4.3 RESETS

TABLE 10-3: RESET SIGNALS

Name	Description
RESET_SYS	Power on Reset to the entire device. This signal resets all the register and logic in this block to its default state.
RESET_HOST	A reset that occurs when VCC is turned off or when the system host resets the Host Interface.
RESET_eSPI	For systems with eSPI, a general reset signal for the eSPI block.

10.5 Interrupts

This block does not generate any interrupts.

10.6 Low Power Modes

This block always automatically adjusts to operate in the lowest power mode.

10.7 Description

The Chip Configuration Registers are divided into two groups: Global Configuration Registers and Logical Device Configuration registers.

10.7.1 CONFIGURATION PORT

The eSPI Host can access the Chip's Configuration Registers through the Configuration Port when CONFIG MODE is enabled. The device defaults to CONFIG MODE being disabled.

Note: The data read from the Configuration Port Data register is undefined when CONFIG MODE is not enabled.

The Configuration Port is composed of an INDEX and DATA Register. The INDEX register is used as an address pointer to an 8-bit configuration register and the DATA register is used to read or write the data value from the indexed configuration register. Once CONFIG MODE is enabled, reading the Configuration Port Data register will return the data value that is in the indexed Configuration Register.

If no value was written to the INDEX register, reading the Data Register in the Configuration Port will return the value in Configuration Address location 00h (default).

TABLE 10-4: CONFIGURATION PORT

Default I/O Address	Type	Register Name	Relative Address	Default Value	Notes
002Eh	Read / Write	INDEX	Configuration Port's Base Address + 0	00h	Note 1
002Fh	Read / Write	DATA	Configuration Port's Base Address + 1	00h	

Note 1: The default Base I/O Address of the Configuration Port can be relocated by programming the BAR register for Logical Device Ch (eSPI, I/O Configuration Port). The Relative Address shows the general case for determining the I/O address for each register.

10.7.2 ENABLE CONFIG MODE

The INDEX and DATA registers are effective only when the chip is in CONFIG MODE. CONFIG MODE is enabled when the Config Entry Key is successfully written to the I/O address of the INDEX register of the CONFIG PORT while the CONFIG MODE is disabled (see following section).

```
Config Entry Key = < 55h>
```


© 2020 Microchip Technology Inc.

10.7.3 DISABLE CONFIG MODE

CONFIG MODE defaults to disabled on a RESET_SYS, RESET_HOST, and, for systems using eSPI, when RESET_HOST is asserted. CONFIG MODE is also disabled when the following Config Exit Key is successfully written to the I/O address of the INDEX PORT of the CONFIG PORT while CONFIG MODE is enabled.

```
Config Exit Key = < AAh>
```

10.7.4 CONFIGURATION SEQUENCE EXAMPLE

To program the configuration registers, the following sequence must be followed:

1. Enable Configuration State
2. Program the Configuration Registers
3. Disable Configuration State.

The following is an example of a configuration program in Intel 8086 assembly language.

```
;----------------------------.
; ENABLE CONFIGURATION STATE
;----------------------------'
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AX,055H; Config Entry Key
OUT     DX,AL
;----------------------------.
; CONFIGURE BASE ADDRESS,    |
; LOGICAL DEVICE 8           |
;----------------------------'
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AL,07H
OUT     DX,AL; Point to LD# Config Reg
MOV     DX,CONFIG_PORT_BASE_ADDRESS+1
MOV     AL, 08H
OUT DX,AL; Point to Logical Device 8
;
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AL,34H
OUT     DX,AL   ; Point to BASE ADDRESS REGISTER
MOV     DX,CONFIG_PORT_BASE_ADDRESS+1
MOV     AL,02H
OUT     DX,AL   ; Update BASE ADDRESS REGISTER
;----------------------------.
; DISABLE CONFIGURATION STATE
;----------------------------'
MOV     DX,CONFIG_PORT_BASE_ADDRESS
MOV     AX,0AAH; Config Exit Key
OUT     DX,AL.
```

10.7.5 GLOBAL CONFIGURATION

There are 48 8-bit Global Configuration Registers (at offsets 00h through 2Fh), plus up to 208 8-bit registers associated with each Logical Device. The Logical Device is selected with the Logical Device Number Register (Global Configuration Register 07h).

Sequence to Access Logical Device Configuration Register:

a) Write the number of the Logical Device being accessed in the Logical Device Number Configuration Register by writing 07h into the INDEX PORT and the Logical Device Number into the DATA PORT.

b) Write the address of the desired logical device configuration register to the INDEX PORT and then write or read the value of the configuration register through the DATA PORT.

© 2020 Microchip Technology Inc.

The following sections define the Global Configuration registers and the Logical Configuration registers.

10.7.6 GLOBAL CONTROL/CONFIGURATION REGISTERS

As with all Configuration Registers, the INDEX PORT is used to select a Global Configuration Register in the chip. The DATA PORT is then used to access the selected register. The INDEX and DATA PORTs are defined in the eSPI Interface description.

10.8 Configuration Registers

Host access to Global Configuration Registers is through the Configuration Port (the INDEX PORT and the DATA PORT) using the Logical Device Number 3Fh and the Index shown in the “Offset” column of the following table. The EC can access Configuration Registers directly.The EC address for each register is formed by adding the Base Address for Global Configuration block shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “Offset” column.

All Global Configuration registers are accessible to the Host through the Configuration Port for all Logical Devices. at offsets 00h through 2Fh.

TABLE 10-5: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS

Register	Host Offset	Description
Chip (Global) Control Registers		
Reserved	00h - 01h	Reserved - Writes are ignored, reads return 0.
TEST	02h	TEST. This register location is reserved for Microchip use. Modifying this location may cause unwanted results.
Reserved	03h - 06h	Reserved - Writes are ignored, reads return 0.
Logical Device Number	07h	A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device. **Note:** The Activate command operates only on the selected logical device.
Reserved	08h - 18h	Reserved - Writes are ignored, reads return 0.
Device Revision	1Ch	A read-only register which provides device revision information.
Device Sub ID	1Dh	Read-Only register which provides the device sub-identification.
Device ID[7:0]	1Eh	Read-Only register which provides Device ID LSB.
Device ID[15:8]	1Fh	Read-Only register which provides Device ID MSB.
Legacy Device ID	20h	A read-only register which provides Legacy device identification. The value of this register is FEh
TEST	22h - 23h	TEST. This register locations are reserved for Microchip use. Modifying these locations may cause unwanted results.
Reserved	24h	Reserved – writes are ignored, reads return “0”.
TEST	25h - 2Fh	TEST. This register locations are reserved for Microchip use. Modifying these locations may cause unwanted results.

© 2020 Microchip Technology Inc.

11.0 8042 EMULATED KEYBOARD CONTROLLER

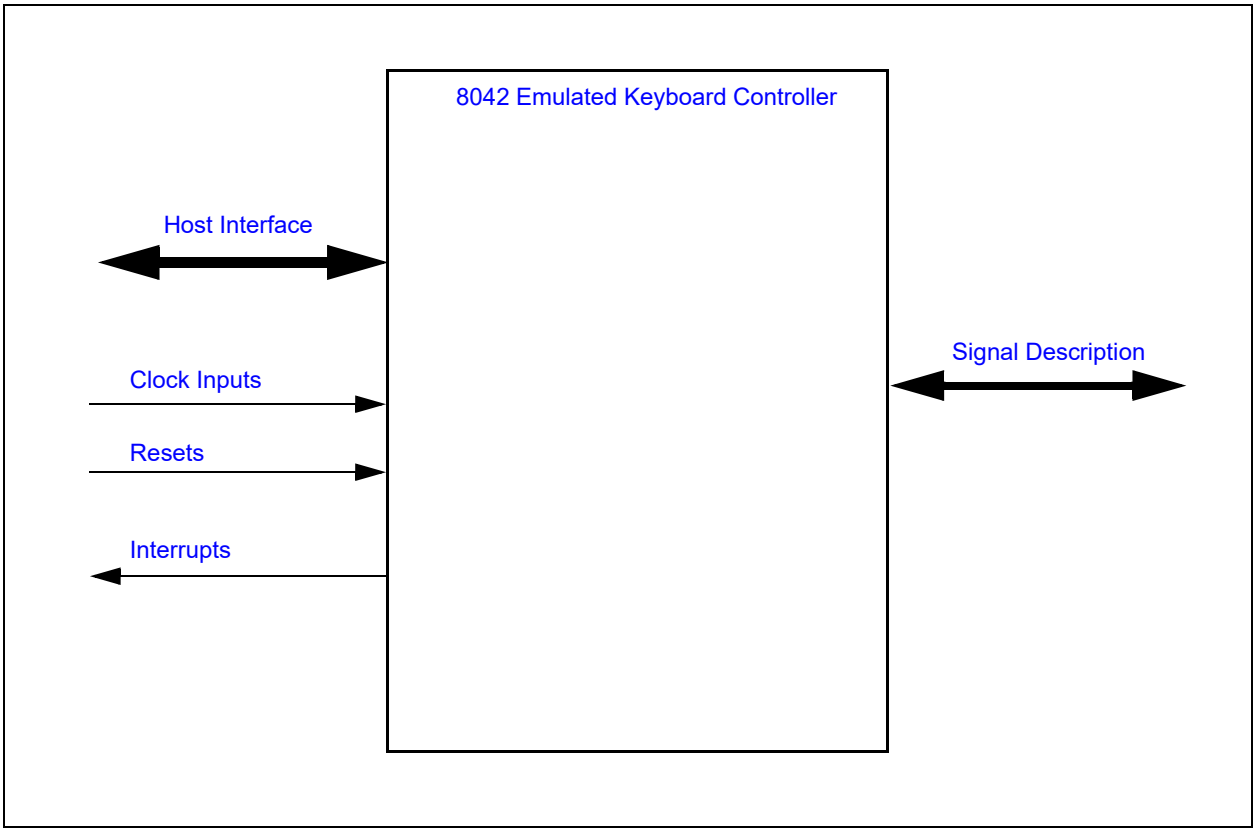
11.1 Introduction

The MEC152x keyboard controller uses the EC to produce a superset of the features provided by the industry-standard 8042 keyboard controller. The 8042 Emulated Keyboard Controller is a Host/EC Message Interface with hardware assists to emulate 8042 behavior and provide Legacy GATEA20 support.

11.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 11-1: I/O DIAGRAM OF BLOCK



11.3 Signal Description

TABLE 11-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
KBRST	Output	Keyboard Reset, routed to pin
A20M	Output	Keyboard gate A20 output pin

11.4 Host Interface

The registers defined for 8042 interface is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

11.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

11.5.1 POWER DOMAINS

TABLE 11-2: POWER SOURCES

Name	Description
VTR_CORE	This Power Well is used to power the registers and logic in this block.

11.5.2 CLOCK INPUTS

TABLE 11-3: CLOCK INPUTS

Name	Description
1MHz	Clock used for the counter in the CPU_RESET circuitry.

11.5.3 RESETS

TABLE 11-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset is asserted when VTR_CORE is applied.
RESET_VCC	This signal is asserted when the main power rail is off.
RESET_HOST	This signal is asserted when the main power rail is off or held off by the PWR_INV bit in the Power Reset Control Register, and also when the Host resets the Host-EC link via PCI_RESET# or PLTRST#.

11.6 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 11-5: SYSTEM INTERRUPTS

Source	Description
KIRQ	This interrupt source for the SIRQ logic, representing a Keyboard interrupt, is generated when the PCOBF status bit is '1'.
MIRQ	This interrupt source for the SIRQ logic, representing a Mouse interrupt, is generated when the AUXOBF status bit is '1'.

TABLE 11-6: EC INTERRUPTS

Source	Description
IBF	Interrupt generated by the host writing either data or command to the data register. This interrupt is asserted when the input buffer becomes not empty (i.e., when the IBF flag goes to 1).
OBE	Interrupt generated by the host reading either data or aux data from the data register. This interrupt is asserted when the output buffer becomes empty (i.e., when the OBF flag goes to 0).

11.7 Low Power Modes

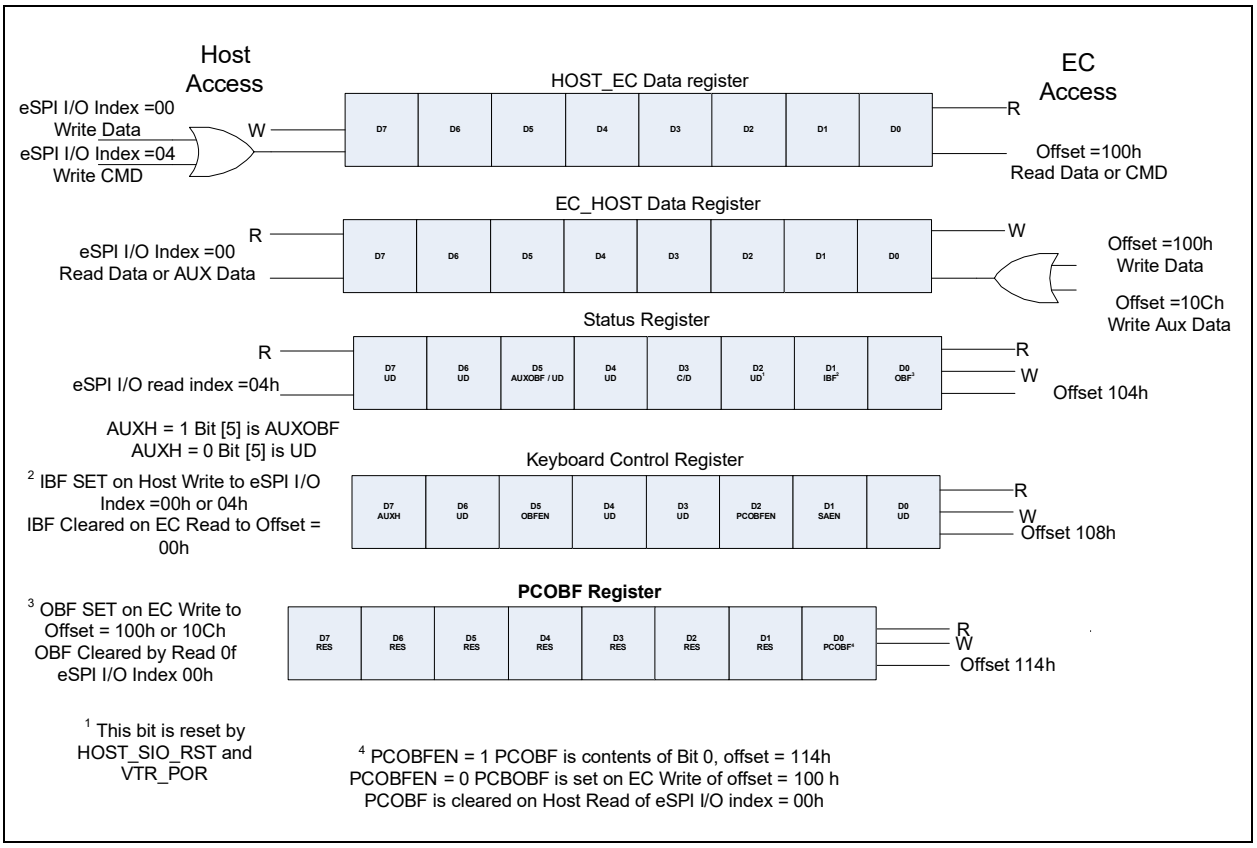
The 8042 Interface may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

© 2020 Microchip Technology Inc.

11.8 Description

11.8.1 BLOCK DIAGRAM

FIGURE 11-2: BLOCK DIAGRAM OF 8042 Emulated Keyboard Controller



11.9 EC-to-Host Keyboard Communication

The EC can write to the EC_HOST Data / AUX Data Register by writing to the HOST2EC Data Register at EC-Only offset 0h or the EC AUX Data Register at EC-Only offset Ch. A write to either of these addresses automatically sets bit 0 (OBF) in the Status register. A write to the HOST2EC Data Register may also set PCOBF. A write to the EC AUX Data Register may also set AUXOBF.

11.9.1 PCOBF DESCRIPTION

If enabled by the bit OBFEN, the bit PCOBF is gated onto KIRQ. The KIRQ signal is a system interrupt which signifies that the EC has written to the HOST2EC Data Register (EC-Only offset 0h). On power-up, PCOBF is reset to 0. PCOBF will normally reflect the status of writes to HOST2EC register, if PCOBFEN is “0”. PCOBF is cleared by hardware on a HOST read of the EC_HOST Data / AUX Data Register.

KIRQ is normally selected as IRQ1 for keyboard support.

Additional flexibility has been added which allows firmware to directly control the PCOBF output signal, independent of data transfers to the host-interface data output register. This feature allows the MEC152x to be operated via the host “polled” mode. Firmware control is active when PCOBFEN is ‘1’. Firmware sets PCOBF high by writing a “1” to the PCOBF field of the PCOBF Register. Firmware must also clear PCOBF by writing a “0” to the PCOBF field.

The PCOBF register is also readable; the value read back on bit 0 of the register always reflects the present value of the PCOBF output. If PCOBFEN = 1, then this value reflects the output of the firmware latch in the PCOBF Register. If PCOBFEN = 0, then the value read back reflects the in-process status of write cycles to the HOST2EC Data Register (i.e., if the value read back is high, the host interface output data register has just been written to). If OBFEN=0, then KIRQ is driven inactive (low).

11.9.2 AUXOBF DESCRIPTION

If enabled by the bit OBFEN, the bit AUXOBF is multiplexed onto MIRQ. The AUXOBF/MIRQ signal is a system interrupt which signifies that the EC has written to the EC_HOST Data / AUX Data Register. On power-up, after RESET_SYS, AUXOBF is reset to 0. AUXOBF will normally reflects the status of writes to EC EC AUX Data Register (EC-Only offset Ch). AUXOBF is cleared by hardware on a read of the Host Data Register. If OBFEN=0, then MIRQ is driven inactive (low).

MIRQ is normally selected as IRQ15 for mouse support.

Firmware can also directly control the AUXOBF output signal, similar to the mechanism it can use to control PCOBF. Firmware control is active when AUXH is '0'. Firmware sets AUXOBF high by writing a "1" to the AUXOBF field of the EC Keyboard Status Register. Firmware must also clear AUXOBF by writing a "0" to the AUXOBF field.

TABLE 11-7: OBFEN AND PCOBFEN EFFECTS ON KIRQ

OBFEN	PCOBFEN	
0	X	KIRQ is inactive and driven low
1	0	KIRQ = PCOBF (status of writes to HOST2EC Data Register)
1	1	KIRQ = PCOBF (status of writes to PCOBF Register)

TABLE 11-8: OBFEN AND AUXH EFFECTS ON MIRQ

OBFEN	AUXH	
0	X	MIRQ is inactive and driven low
1	0	MIRQ = AUXOBF (status of writes to EC AUX Data Register)
1	1	MIRQ = AUXOBF (status of writes to AUXOBF in EC Keyboard Status Register)

11.10 Legacy Port92/GATEA20 Support

The MEC152x supports I/O writes to port HOST I/O address 92h as a quick alternate mechanism for generating a CPU_RESET pulse or controlling the state of GATEA20. The Port92/GateA20 logic has a separate Logical Device Number and Base Address register (see Section 11.15, "Legacy Port92/GATEA20 Configuration Registers" and Section 11.16, "Legacy Port92/GATEA20 Runtime Registers". The Base Address Register for the Port92/GateA20 Logical Device has only one writable bit, the Valid Bit, since the only I/O accessible Register has a fixed address.

The Port 92 Register resides at HOST I/O address 92h and is used to support the alternate reset (ALT_RST#) and alternate GATEA20 (ALT_A20) functions. This register defaults to 00h on assertion of RESET_VCC.

Setting the Port92 Enable bit (Port 92 Enable Register) enables the Port92h Register. When Port92 is disabled, by clearing the Port92 Enable bit, then access to this register is completely disabled (I/O writes to host 92h are ignored and I/O reads float the system data bus SD[7:0]).

11.10.1 GATE A20 SPEEDUP

The MEC152x contains on-chip logic support for the GATEA20 hardware speed-up feature. GATEA20 is part of the control required to mask address line A20 to emulate 8086 addressing.

In addition to the ability for the host to control the GATEA20 output signal directly, a configuration bit called SAEN in the Keyboard Control Register is provided; when set, SAEN allows firmware to control the GATEA20 output. When SAEN is set, a 1 bit register (GATEA20 Control Register) controls the GATEA20 output.

Host control and firmware control of GATEA20 affect two separate register elements. Read back of GATEA20 through the use of EC OFFSET 100h reflects the present state of the GATEA20 output signal: if SAEN is set, the value read back corresponds to the last firmware-initiated control of GATEA20; if SAEN is reset, the value read back corresponds to the last host-initiated control of GATEA20.

Host control of the GATEA20 output is provided by the hardware interpretation of the "GATEA20 sequence" (see Table 11-9, "GATEA20 Command/Data Sequence Examples"). The foregoing description assumes that the SAEN configuration bit is reset.

© 2020 Microchip Technology Inc.

When the MEC152x receives a “D1” command followed by data (via the host interface), the on-chip hardware copies the value of data bit 1 in the received data field to the GATEA20 host latch. At no time during this host-interface transaction will PCOBF or the IBF flag (bit 1) in the EC Keyboard Status Register be activated; for example, this host control of GATEA20 is transparent to firmware, with no consequent degradation of overall system performance. Table 11-9 details the possible GATEA20 sequences and the MEC152x responses.

An additional level of control flexibility is offered via a memory-mapped synchronous set and reset capability. Any data written to the SETGA20L Register causes the GATEA20 host latch to be set; any data written to the RSTGA20L Register causes it to be reset. This control mechanism should be used with caution. It was added to augment the “normal” control flow as described above, not to replace it. Since the host and the firmware have asynchronous control capability of the host latch via this mechanism, a potential conflict could arise. Therefore, after using the SETGA20L and RSTGA20L registers, firmware should read back the GATEA20 status via the GATEA20 Control Register (with SAEN = 0) to confirm the actual GATEA20 response.

TABLE 11-9: GATEA20 COMMAND/DATA SEQUENCE EXAMPLES

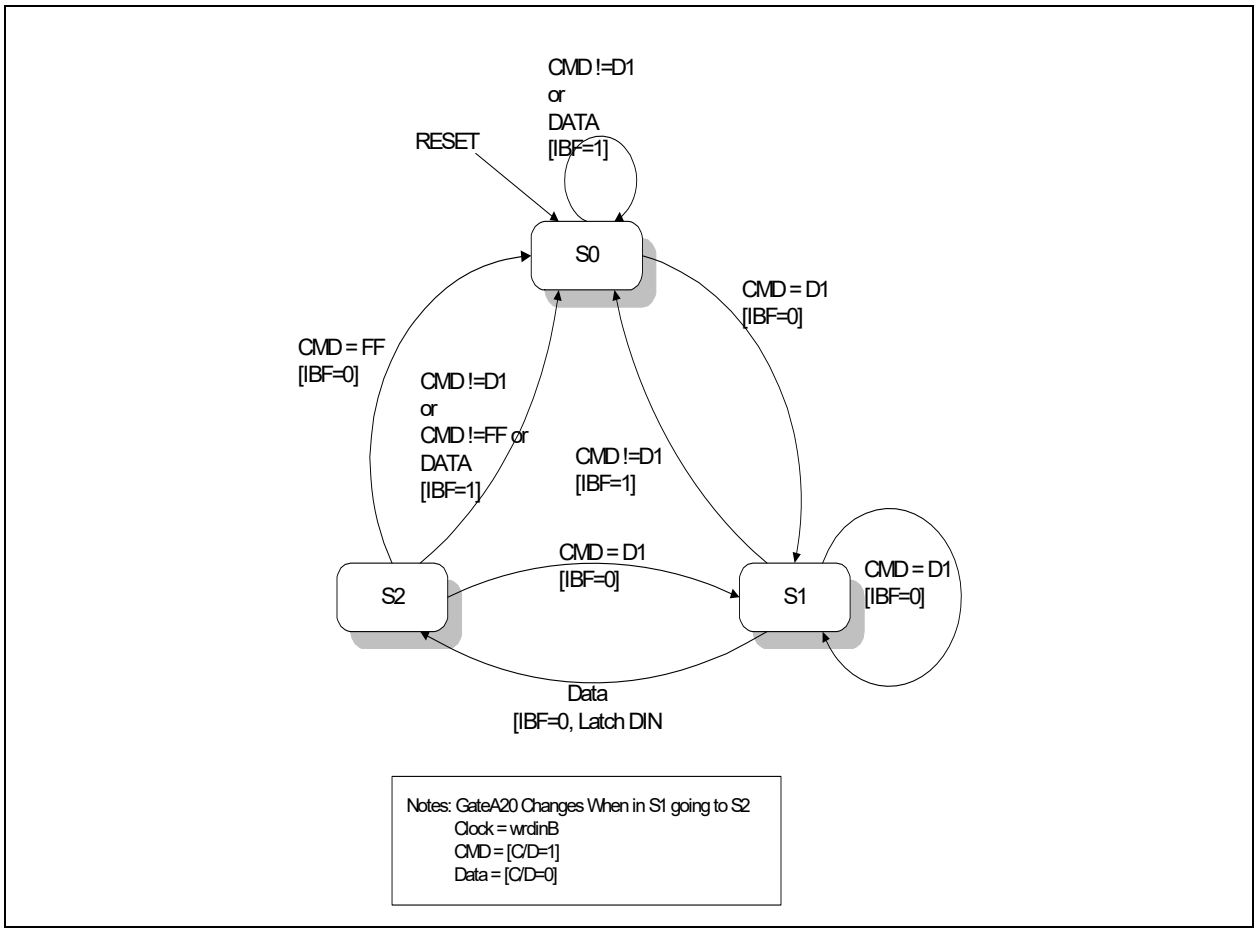
Data Byte	R/W	D[0:7]	IBF Flag	GATEA20	Comments
1 0 1	W W W	D1 DF FF	0 0 0	Q 1 1	GATEA20 Turn-on Sequence
1 0 1	W W W	D1 DD FF	0 0 0	Q 0 0	GATEA20 Turn-off Sequence
1 1 0 1	W W W W	D1 D1 DF FF	0 0 0 0	Q Q 1 1	GATEA20 Turn-on Sequence(*)
1 1 0 1	W W W W	D1 D1 DD FF	0 0 0 0	Q Q 0 0	GATEA20 Turn-off Sequence(*)
1 1 1	W W W	D1 XX** FF	0 1 1	Q Q Q	Invalid Sequence

Note: The following notes apply:

- All examples assume that the SAEN configuration bit is 0.
- “Q” indicates the bit remains set at the previous state.
- *Not a standard sequence.
- **XX = Anything except D1.
- If multiple data bytes, set IBF and wait at state 0. Let the software know something unusual happened.
- For data bytes, only D[1] is used; all other bits are don't care.
- Host Commands (FF, FE, and D1) do not cause IBF. The method of blocking IBF in Figure 11-4 is the nIOW not being asserted when FF, FE, and D1 Host commands are written”.

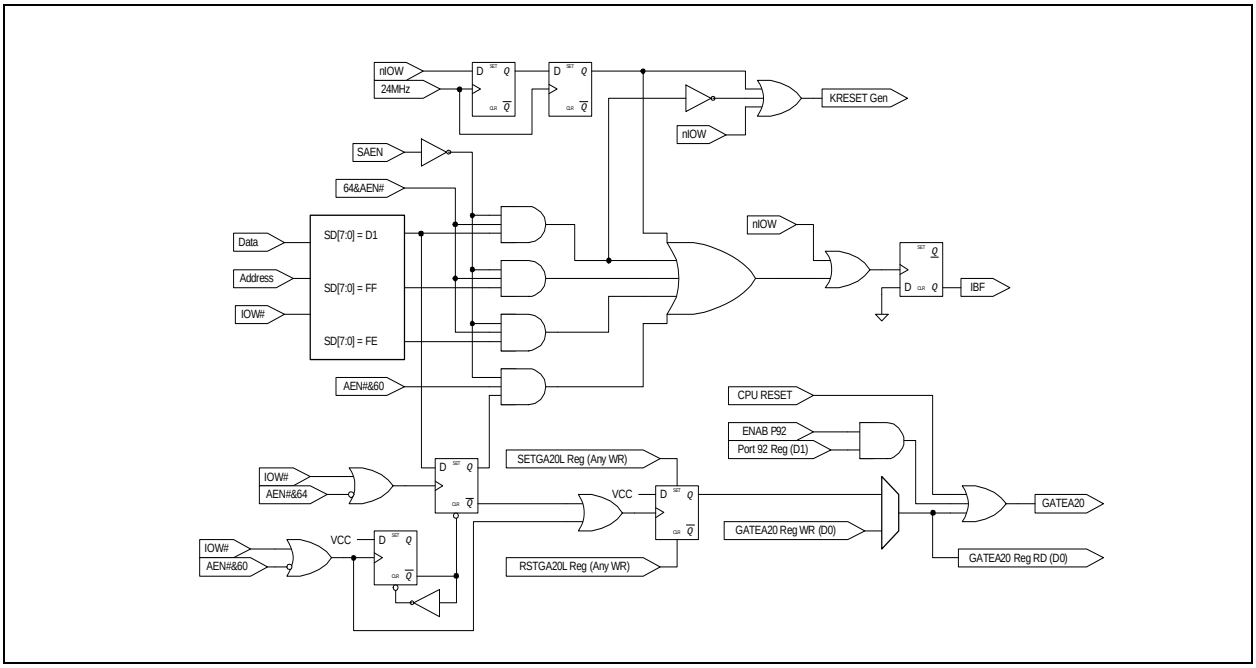
The hardware GATEA20 state machine returns to state S1 from state S2 when CMD = D1, as shown in the following figures:.

FIGURE 11-3: GATEA20 STATE MACHINE



© 2020 Microchip Technology Inc.

FIGURE 11-4: GATEA20 IMPLEMENTATION DIAGRAM



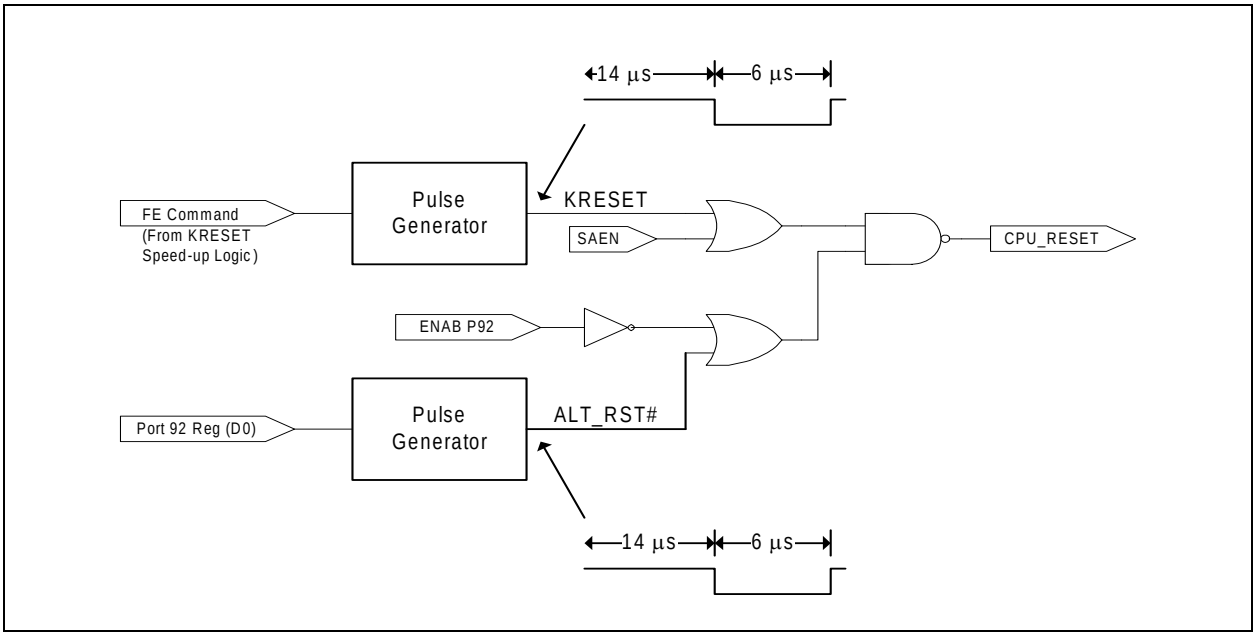
11.10.2 CPU_RESET HARDWARE SPEED-UP

The ALT_CPU_RESET bit generates, under program control, the ALT_RST# signal, which provides an alternate, means to drive the MEC152x CPU_RESET pin which in turn is used to reset the Host CPU. The ALT_RST# signal is internally NANDed together with the KBDRESET# pulse from the KRESET Speed up logic to provide an alternate software means of resetting the host CPU.

Before another ALT_RST# pulse can be generated, ALT_CPU_RESET must be cleared to '0' either by an RESET_VCC or by a write to the Port 92 Register with bit 0 = '0'. An ALT_RST# pulse is not generated in the event that the ALT_CPU_RESET bit is cleared and set before the prior ALT_RESET# pulse has completed.

If the 8042EM Sleep Enable is asserted, or the 8042 EM ACTIVATE bit is de-asserted, the 1MHz clocks source is disabled.

FIGURE 11-5: CPU_RESET IMPLEMENTATION DIAGRAM



11.11 Instance Description

There are two blocks defined in this chapter: Emulated 8042 Interface and the Port 92-Legacy Interface. The MEC152x has one instance of each block.

11.12 Configuration Registers

Configuration Registers for an instance of the 8042 Emulated Keyboard Controller are listed in the following table. Host access to Configuration Registers is through the Configuration Port using the Logical Device Number of the 8042 Emulated Keyboard Controller instance and the Index shown in the “Host Index” column of the table. The EC can access Configuration Registers directly. The EC address for each register is formed by adding the Base Address for each instance of the 8042 Emulated Keyboard Controller shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “EC Offset” column.

EC Offset	Host Index	Register Name
330h	30h	Activate Register

11.12.1 ACTIVATE REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	ACTIVATE 1=The 8042 Interface is powered and functional. 0=The 8042 Interface is powered down and inactive.	R/W	0b	RESET_VCC

11.13 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the 8042 Emulated Keyboard Controller. Host access for each register listed in this table is defined as an offset in the Host address space to the Block’s Base Address, as defined by the instance’s Base Address Register.

© 2020 Microchip Technology Inc.

The EC address for each register is formed by adding the Base Address for the 8042 Emulated Keyboard Controller shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “Offset” column.

TABLE 11-10: RUNTIME REGISTER SUMMARY

Offset	Register Name
0h/04h	HOST_EC Data / CMD Register
0h	EC_HOST Data / AUX Data Register
4h	Keyboard Status Read Register

11.13.1 HOST_EC DATA / CMD REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	WRITE_DATA This 8-bit register is write-only. When written, the C/D bit in the Keyboard Status Read Register is cleared to ‘0’, signifying data, and the IBF in the same register is set to ‘1’. When the Runtime Register at offset 0h is read by the Host, it functions as the EC_HOST Data / AUX Data Register.	W	0h	RESET_SYS

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	WRITE_CMD .This 8-bit register is write-only and is an alias of the register at offset 0h. When written, the C/D bit in the Keyboard Status Read Register is set to ‘1’, signifying a command, and the IBF in the same register is set to ‘1’. When the Runtime Register at offset 4h is read by the Host, it functions as the Keyboard Status Read Register.	W	0h	RESET_SYS

11.13.2 EC_HOST DATA / AUX DATA REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	READ_DATA This 8-bit register is read-only. When read by the Host, the PCOBF and/or AUXOBF interrupts are cleared and the OBF flag in the status register is cleared.	R	0h	RESET_SYS

11.13.3 KEYBOARD STATUS READ REGISTER

This register is a read-only alias of the EC Keyboard Status Register.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	UD2 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R	0h	RESET_SYS
5	AUXOBF Auxiliary Output Buffer Full. This bit is set to “1” whenever the EC writes the EC AUX Data Register. This flag is reset to “0” whenever the EC writes the EC Data Register.	R	0h	RESET_SYS
4	UD1 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R	0h	RESET_SYS
3	C/D Command Data. This bit specifies whether the input data register contains data or a command (“0” = data, “1” = command). During a Host command write operation (when the Host writes the HOST_EC Data / CMD Register at offset 04h), this bit is set to “1”. During a Host data write operation (when the Host writes the HOST_EC Data / CMD Register at offset 0h), this bit is set to “0”.	R	0h	RESET_SYS
2	UD0 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R	0h	RESET_HOST
1	IBF Input Buffer Full. This bit is set to “1” whenever the Host writes data or a command into the HOST_EC Data / CMD Registerr. When this bit is set, the EC's IBF interrupt is asserted, if enabled. When the EC reads the HOST_EC Data/CMD Register, this bit is automatically reset and the interrupt is cleared. This bit is not reset on RESET_VCC. To clear this bit, firmware must read the HOST2EC Data Register in the EC-Only address space.	R	0h	RESET_SYS
0	OBF Output Buffer Full. This bit is set when the EC writes a byte of Data or AUX Data into the EC_HOST Data / AUX Data Register. When the Host reads the HOST_EC Data / CMD Register, this bit is automatically cleared by hardware and an OBE interrupt is generated. This bit is not reset on RESET_VCC. To clear this bit, firmware must read the HOST2EC Data Register in the EC-Only address space.	R	0h	RESET_SYS

11.14 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the 8042 Emulated Keyboard Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

© 2020 Microchip Technology Inc.

TABLE 11-11: EC-ONLY REGISTER SUMMARY

Offset	Register Name
100h	HOST2EC Data Register
100h	EC Data Register
104h	EC Keyboard Status Register
108h	Keyboard Control Register
10Ch	EC AUX Data Register
114h	PCOBF Register

11.14.1 HOST2EC DATA REGISTER

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST2EC_DATA This register is an alias of the HOST_EC Data / CMD Register. When read at the EC-Only offset of 0h, it returns the data written by the Host to either Runtime Register offset 0h or Runtime Register offset 04h.	R	0h	RESET_SYS

11.14.2 EC DATA REGISTER

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA	W	0h	RESET_SYS

11.14.3 EC KEYBOARD STATUS REGISTER

This register is an alias of the Keyboard Status Read Register. The fields C/D, IBF, and OBF remain read-only.

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	UD2 User-defined data. Readable and writable by the EC.	R/W	0h	RESET_SYS
5	AUXOBF Auxiliary Output Buffer Full. This bit is set to ‘1’ whenever the EC writes the EC AUX Data Register. This flag is reset to ‘0’ whenever the EC writes the EC Data Register.	R/W	0h	RESET_SYS
4	UD1 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	RESET_SYS

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	C/D Command Data. This bit specifies whether the input data register contains data or a command. During a Host command write operation (when the Host writes the HOST_EC Data / CMD Register at offset 04h), this bit is set to '1'. During a Host data write operation (when the Host writes the HOST_EC Data / CMD Register at offset 0h), this bit is set to '0'. 1=Command 0=Data	R	0h	RESET_SYS
2	UD0 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	RESET_HOST
1	IBF Input Buffer Full. This bit is set to "1" whenever the Host writes data or a command into the HOST_EC Data / CMD Registerr. When this bit is set, the EC's IBF interrupt is asserted, if enabled. When the EC reads the Data/CMD Register, this bit is automatically reset and the interrupt is cleared. This bit is not reset on RESET_VCC. To clear this bit, firmware must read the HOST2EC Data Register in the EC-Only address space.	R	0h	RESET_SYS
0	OBF Output Buffer Full. This bit is set when the EC writes a byte of Data or AUX Data into the EC_HOST Data / AUX Data Register. When the Host reads the HOST_EC Data / CMD Register, this bit is automatically cleared by hardware and a OBE interrupt is generated. This bit is not reset on RESET_VCC. To clear this bit, firmware must read the HOST2EC Data Register in the EC-Only address space.	R	0h	RESET_SYS

11.14.4 KEYBOARD CONTROL REGISTER

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	AUXH AUX in Hardware. 1=AUXOBF of the Keyboard Status Read Register is set in hardware by a write to the EC AUX Data Register 0=AUXOBF is not modified in hardware, but can be read and written by the EC using the EC-Only alias of the EC Keyboard Status Register	R/W	0h	RESET_SYS
6	UD5 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	OBFEN When this bit is '1', the system interrupt signal KIRQ is driven by the bit PCOBF and MIRQ is driven by AUXOBF. When this bit is '0', KIRQ and MIRQ are driven low. This bit must not be changed when OBF of the status register is equal to '1'.	R/W	0h	RESET_SYS
4:3	UD4 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	RESET_SYS
2	PCOBFEN 1=reflects the value written to the PCOBF Register 0=PCOBF reflects the status of writes to the EC Data Register	R/W	0h	RESET_SYS
1	SAEN Software-assist enable. 1=This bit allows control of the GATEA20 signal via firmware 0=GATEA20 corresponds to either the last Host-initiated control of GATEA20 or the firmware write to the Keyboard Control Register or the EC AUX Data Register.	R/W	0h	RESET_SYS
0	UD3 User-defined data. Readable and writable by the EC when written by the EC at its EC-only alias.	R/W	0h	RESET_SYS

11.14.5 EC AUX DATA REGISTER

Offset	10Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_AUX_DATA This 8-bit register is write-only. When written, the C/D in the Keyboard Status Read Register is cleared to '0', signifying data, and the IBF in the same register is set to '1'. When the Runtime Register at offset 0h is read by the Host, it functions as the EC_HOST Data / AUX Data Register.	W	0h	RESET_SYS

11.14.6 PCOBF REGISTER

Offset	114h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	PCOBF For a description of this bit, see Section 11.9.1, "PCOBF Description".	R/W	0h	RESET_SYS

11.15 Legacy Port92/GATEA20 Configuration Registers

Configuration Registers for an instance of the Port92-Legacy block are listed in the following table. Host access to Configuration Registers is through the Configuration Port using the Logical Device Number of the Port 92 instance and the Index shown in the “Host Index” column of the table. The EC can access Configuration Registers directly. The EC address for each register is formed by adding the Base Address for each instance of the Port 92 block shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “EC Offset” column.

TABLE 11-12: CONFIGURATION REGISTER SUMMARY

EC Offset	Host Index	Register Name
330h	30h	Port 92 Enable Register

11.15.1 PORT 92 ENABLE REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	P92_EN When this bit is ‘1’, the Port92h Register is enabled. When this bit is ‘0’, the Port92h Register is disabled, and Host writes to address 92h are ignored.	R/W	0h	RESET_VCC

11.16 Legacy Port92/GATEA20 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Legacy Port92/GATEA20 logic. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the Runtime Register Base Address Table.

Block Instance	Instance Number	Host	Address Space	Base Address
Port92-Legacy	0	eSPI	I/O	0092h
		EC	32-bit address space	400F_2000h

The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 11-13: RUNTIME REGISTER SUMMARY

Offset	Register Namer
0h	Port 92 Register

11.16.1 PORT 92 REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-
1	ALT_GATE_A20 This bit provides an alternate means for system control of the GATEA20 pin. ALT_A20 low drives GATEA20 low, if A20 from the keyboard controller is also low. When Port 92 is enabled, writing a 1 to this bit forces ALT_A20 high. ALT_A20 high drives GATEA20 high regardless of the state of A20 from the keyboard controller. 0=ALT_A20 is driven low 1=ALT_A20 is driven high	R/W	0h	RESET_HOST

© 2020 Microchip Technology Inc.

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	ALT_CPU_RESET	R/W	0h	RESET_HOST

11.17 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Port92-Legacy Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 11-14: REGISTER SUMMARY

Offset	Register Name
100h	GATEA20 Control Register
108h	SETGA20L Register
10Ch	RSTGA20L Register

11.17.1 GATEA20 CONTROL REGISTER

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	GATEA20 See Section 11.10.1, "GATE A20 Speedup" for information on this register. 0=The GATEA20 output is driven low 1=The GATEA20 output is driven high	R/W	1h	RESET_HOST

11.17.2 SETGA20L REGISTER

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SETGA20L See Section 11.10.1, "GATE A20 Speedup" for information on this register. A write to this register sets GATEA20 in the GATEA20 Control Register.	W	-	-

11.17.3 RSTGA20L REGISTER

Offset	10Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	RSTGA20L See Section 11.10.1, "GATE A20 Speedup" for information on this register. A write to this register sets GATEA20 in the GATEA20 Control Register.	W	-	-

12.0 ACPI EMBEDDED CONTROLLER INTERFACE (ACPI-ECI)

12.1 Introduction

The ACPI Embedded Controller Interface (ACPI-ECI) is a Host/EC Message Interface. The ACPI specification defines the standard hardware and software communications interface between the OS and an embedded controller. This interface allows the OS to support a standard driver that can directly communicate with the embedded controller, allowing other drivers within the system to communicate with and use the EC resources; for example, Smart Battery and AML code.

The ACPI Embedded Controller Interface (ACPI-ECI) provides a four byte full duplex data interface which is a superset of the standard ACPI Embedded Controller Interface (ACPI-ECI) one byte data interface. The ACPI Embedded Controller Interface (ACPI-ECI) defaults to the standard one byte interface.

The MEC152x has two instances of the ACPI Embedded Controller Interface.

1. The EC host in Section 12.12, "Runtime Registers" and Section 12.13, "EC Registers" corresponds to the EC in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_EC.
2. The eSPI host in Section 12.12, "Runtime Registers" and Section 12.13, "EC Registers" corresponds to the “System Host Interface to OS” in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_OS.

12.2 References

- Advanced Configuration and Power Interface Specification, Revision 4.0 June 16, 2009, Hewlett-Packard Corporation Intel Corporation Microsoft Corporation Phoenix Technologies Ltd. Toshiba Corporation

12.3 Terminology

TABLE 12-1: TERMINOLOGY

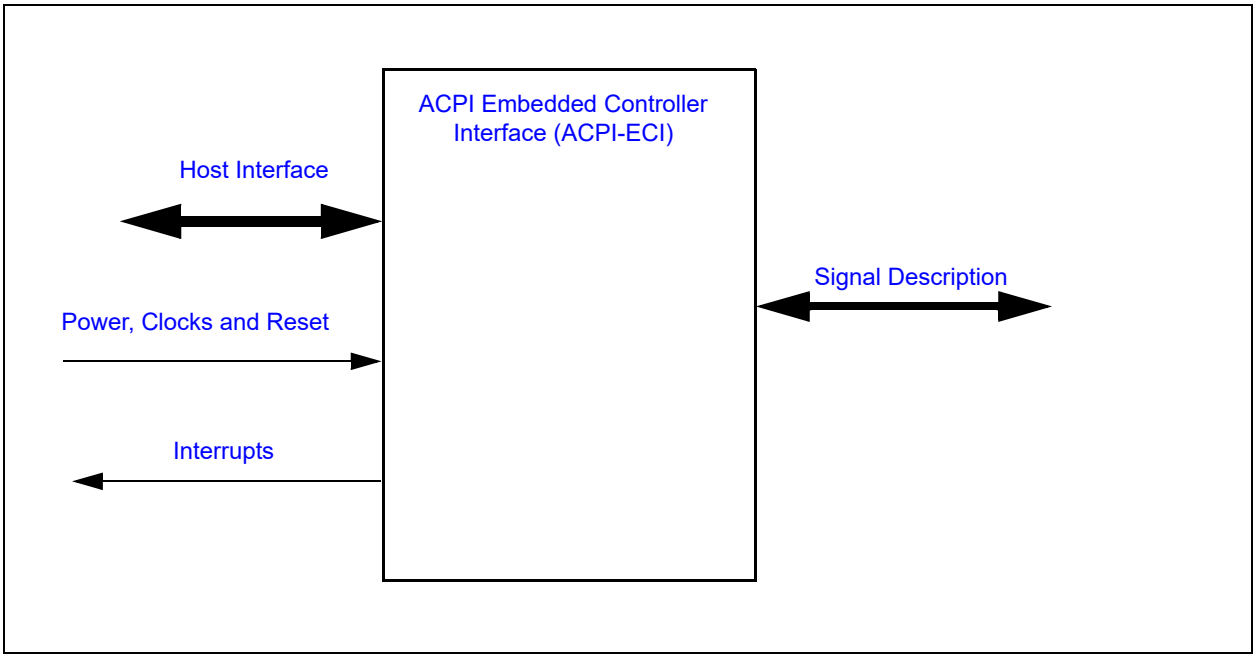
Term	Definition
ACPI_EC	The EC host corresponding to the ACPI specification interface to the EC.
ACPI_OS	The eSPI host corresponding to the ACPI specification interface to the “System Host Interface to OS”. ACPI_OS terminology is not meant to distinguish the ACPI System Management from Operating System but merely the hardware path upstream towards the CPU.

© 2020 Microchip Technology Inc.

12.4 Interface

This block is designed to be accessed internally via a register interface.

FIGURE 12-1: I/O DIAGRAM OF BLOCK



12.5 Signal Description

There are no external signals.

12.6 Host Interface

The registers defined for the ACPI Embedded Controller Interface (ACPI-ECI) are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses"..

12.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

12.7.1 POWER DOMAINS

TABLE 12-3: POWER SOURCES

Name	Description
VTR_CORE	The logic and registers implemented in this block reside on this single power well.

12.7.2 CLOCK INPUTS

This block only requires the Host interface clocks to synchronize registers access.

12.7.3 RESETS

TABLE 12-4: RESET SIGNALS

Name	Description
RESET_SYS	This signal resets all the logic and registers in this interface.

12.8 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 12-5: SYSTEM INTERRUPTS

Source	Description
EC_OBE	This host interrupt is asserted when the OBF bit in the EC STATUS Register is cleared to '0'.

TABLE 12-6: EC INTERRUPTS

Source	Description
IBF	Interrupt generated by the host writing either data or command to the data register. This interrupt is asserted when the input buffer becomes not empty (i.e., when the IBF flag goes to 1).
OBE	Interrupt generated by the host reading either data or aux data from the data register. This interrupt is asserted when the output buffer becomes empty (i.e., when the OBF flag goes to 0).

Note: The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI and SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the OS STATUS OS Register are software flags and this block do not initiate SMI or SCI events.

12.9 Low Power Modes

The ACPI Embedded Controller Interface (ACPI-ECI) automatically enters low power mode when no transaction targets it.

12.10 Description

The ACPI Embedded Controller Interface (ACPI-ECI) provides an APCI-EC interface that adheres to the ACPI specification. The ACPI Embedded Controller Interface (ACPI-ECI) includes two modes of operation: Legacy Mode and Four-byte Mode.

The ACPI Embedded Controller Interface (ACPI-ECI) defaults to Legacy Mode which provides single byte Full Duplex operation. Legacy Mode corresponds to the ACPI specification functionality as illustrated in Figure 12-2, "Block Diagram corresponding to the ACPI specification". The EC interrupts in Figure 12-2 are implemented as OBE and IBF. See Section 12.8, "Interrupts".

In Four-byte Mode, the ACPI Embedded Controller Interface (ACPI-ECI) provides four byte Full Duplex operation. Four-byte Mode is a superset of the ACPI specification functionality as illustrated in Figure 12-2, "Block Diagram corresponding to the ACPI specification".

Both Legacy Mode and Four-byte Mode provide Full Duplex Communications which allows data/command transfers in one direction while maintaining data from the other direction; communications can flow both ways simultaneously.

In Legacy Mode, ACPI Embedded Controller Interface (ACPI-ECI) contains three registers: ACPI OS COMMAND Register, OS STATUS OS Register, and OS2EC Data EC Byte 0 Register. The standard ACPI Embedded Controller Interface (ACPI-ECI) registers occupy two addresses in the ACPI_OS space (Table 12-9).

The OS2EC Data EC Byte 0 Register and ACPI OS COMMAND Register registers appear as a single 8-bit data register in the ACPI_EC. The CMD bit in the OS STATUS OS Register is used by the ACPI_EC to discriminate commands from data written by the ACPI_OS to the ACPI_EC. CMD bit is controlled by hardware: ACPI_OS writes to the OS2EC Data EC Byte 0 Register register clear the CMD bit; ACPI_OS writes to the ACPI OS COMMAND Register set the CMD bit.

© 2020 Microchip Technology Inc.

FIGURE 12-2: BLOCK DIAGRAM CORRESPONDING TO THE ACPI SPECIFICATION

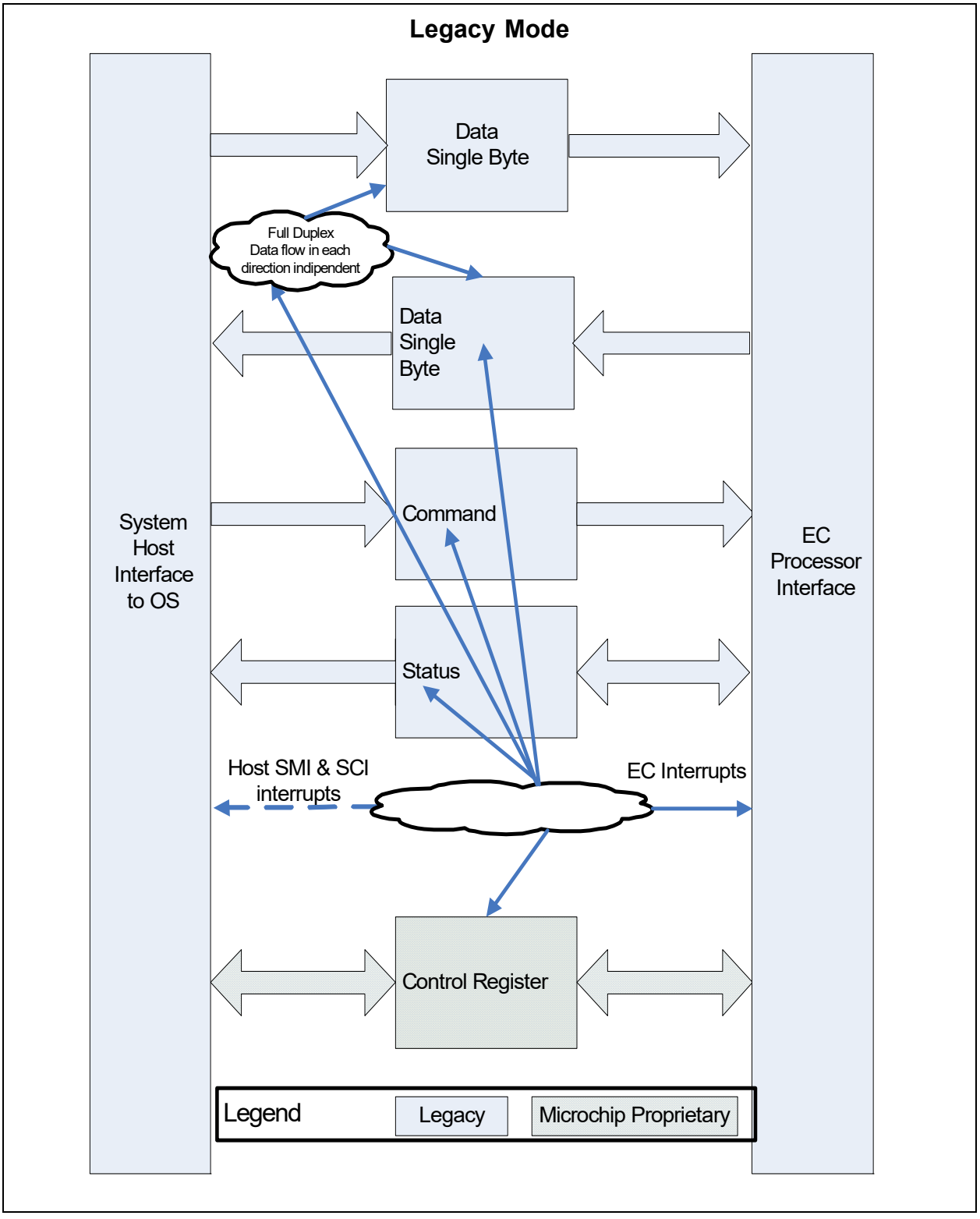
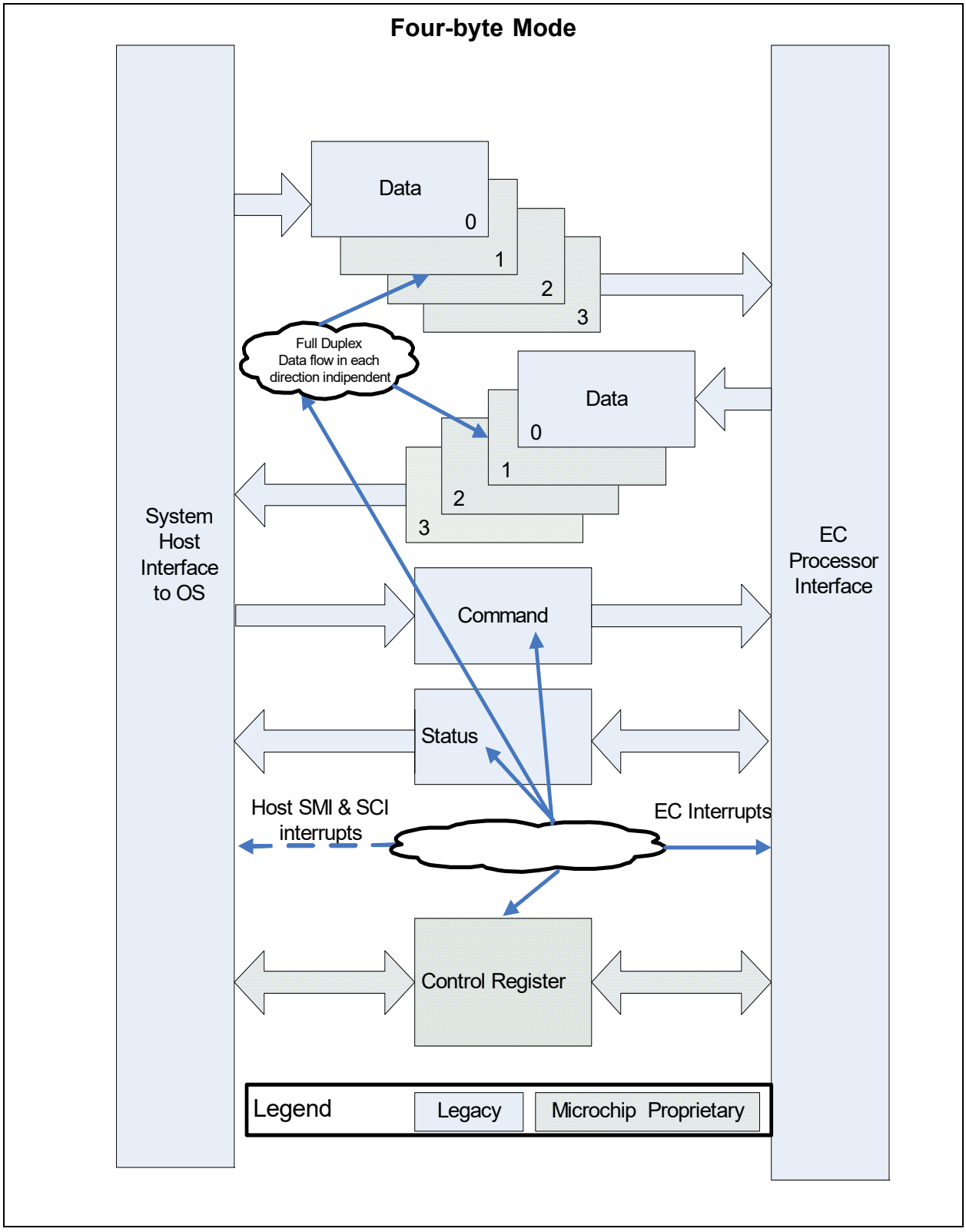


FIGURE 12-2: BLOCK DIAGRAM CORRESPONDING TO THE ACPI SPECIFICATION



© 2020 Microchip Technology Inc.

12.11 Register Aliasing between Runtime and EC-Only Registers

Table 12-7, "Runtime Register Aliasing into EC-Only Registers" indicates the aliasing from Runtime registers to EC-Only registers. The “Host/EC Access” column distinguishes the aliasing based on access type. See individual register descriptions for more details.

TABLE 12-7: RUNTIME REGISTER ALIASING INTO EC-ONLY REGISTERS

Host Offset	Runtime Register Register Name	Host Access	EC Offset	Aliased EC-Only Register Register Name	EC Access
00h	ACPI OS Data Register Byte 0 Register	W	108h	OS2EC Data EC Byte 0 Register	R
00h	ACPI OS Data Register Byte 0 Register	R	100h	EC2OS Data EC Byte 0 Register	W
01h	ACPI OS Data Register Byte 1 Register	W	109h	OS2EC Data EC Byte 1 Register	R
01h	ACPI OS Data Register Byte 1 Register	R	101h	EC2OS Data EC Byte 1 Register	W
02h	ACPI OS Data Register Byte 2 Register	W	10Ah	OS2EC Data EC Byte 2 Register	R
02h	ACPI OS Data Register Byte 2 Register	R	102h	EC2OS Data EC Byte 2 Register	W
03h	ACPI OS Data Register Byte 3 Register	W	10Bh	OS2EC Data EC Byte 3 Register	R
03h	ACPI OS Data Register Byte 3 Register	R	103h	EC2OS Data EC Byte 3 Register	W
04h	ACPI OS COMMAND Register	W	108h	OS2EC Data EC Byte 0 Register	R
04h	OS STATUS OS Register	R	104h	EC STATUS Register	W
05h	OS Byte Control Register	R	105h	EC Byte Control Register	R/W
06h	Reserved		106h	Reserved	
07h	Reserved		107h	Reserved	

Table 12-8, "EC-Only Registers Summary" indicates the aliasing from EC-Only to Runtime registers. The “Host/EC Access” column distinguishes the aliasing based on access type. See individual register descriptions for more details.

TABLE 12-8: EC-ONLY REGISTERS SUMMARY

EC Offset	EC-Only Registers Register Name	EC Access	Host Offset	Aliased Runtime Register Register Name	Host Access
108h	OS2EC Data EC Byte 0 Register	R	00h	ACPI OS Data Register Byte 0 Register	W
108h	OS2EC Data EC Byte 0 Register	R	04h	ACPI OS COMMAND Register	W
109h	OS2EC Data EC Byte 1 Register	R	01h	ACPI OS Data Register Byte 1 Register	W
10Ah	OS2EC Data EC Byte 2 Register	R	02h	ACPI OS Data Register Byte 2 Register	W
10Bh	OS2EC Data EC Byte 3 Register	R	03h	ACPI OS Data Register Byte 3 Register	W
104h	EC STATUS Register	W	04h	OS STATUS OS Register	W
105h	EC Byte Control Register	R/W	05h	OS Byte Control Register	R
106h	Reserved	R		Reserved	R
107h	Reserved	R		Reserved	R
100h	EC2OS Data EC Byte 0 Register	W	00h	ACPI OS Data Register Byte 0 Register	R
101h	EC2OS Data EC Byte 1 Register	W	01h	ACPI OS Data Register Byte 1 Register	R
102h	EC2OS Data EC Byte 2 Register	W	02h	ACPI OS Data Register Byte 2 Register	R
103h	EC2OS Data EC Byte 3 Register	W	03h	ACPI OS Data Register Byte 3 Register	R

12.12 Runtime Registers

Note: The Runtime registers may be accessed by the EC but typically the Host will access the Runtime Registers and the EC will access just the EC-Only registers.

The registers listed in the Runtime Register Summary table are for a single instance of the ACPI Embedded Controller Interface (ACPI-ECI). Host access for each register listed in this table is defined as an offset in the Host address space to the Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the ACPI Embedded Controller Interface (ACPI-ECI) shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

TABLE 12-9: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	ACPI OS Data Register Byte 0 Register
01h	ACPI OS Data Register Byte 1 Register
02h	ACPI OS Data Register Byte 2 Register
03h	ACPI OS Data Register Byte 3 Register
04h	ACPI OS COMMAND Register

© 2020 Microchip Technology Inc.

TABLE 12-9: RUNTIME REGISTER SUMMARY (CONTINUED)

Offset	Register Name
04h	OS STATUS OS Register
05h	OS Byte Control Register
06h	Reserved
07h	Reserved

12.12.1 ACPI OS DATA REGISTER BYTE 0 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_0 This is byte 0 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.12.1.1 ACPI-OS DATA BYTES[3:0]

Writes by the ACPI_OS to the ACPI-OS DATA BYTES[3:0] are aliased to the OS2EC DATA BYTES[3:0]. Reads by the ACPI_OS from the ACPI-OS DATA BYTES[3:0] are aliased to the EC2OS DATA BYTES[3:0].

All access to the ACPI-OS DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

Writes to any of the four ACPI-OS DATA BYTES[3:0] registers clears the CMD bit in the OS STATUS OS Register (the state of the FOUR_BYTE_ACCESS bit in the OS Byte Control Register has no impact.)

When the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is cleared to '0', the following access rules apply:

1. Writes to the ACPI OS Data Register Byte 0 Register sets the IBF bit in the OS STATUS OS Register.
2. Reads from the ACPI OS Data Register Byte 0 Register clears the OBF bit in the OS STATUS OS Register.
3. All writes to ACPI-OS DATA BYTES[3:1] complete without error but the data are not registered.
4. All reads from ACPI-OS DATA BYTES[3:1] return 00h without error.
5. Access to ACPI-OS DATA BYTES[3:1] has no effect on the IBF and OBF bits in the OS STATUS OS Register.

When the Four Byte Access bit in the OS Byte Control Register is set to '1', the following access rules apply:

1. Writes to the ACPI OS Data Register Byte 3 Register sets the IBF bit in the OS STATUS OS Register.
2. Reads from the ACPI OS Data Register Byte 3 Register clears the OBF bit in the OS STATUS OS Register.

12.12.2 ACPI OS DATA REGISTER BYTE 1 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_1 This is byte 1 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.12.3 ACPI OS DATA REGISTER BYTE 2 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_2 This is byte 2 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.12.4 ACPI OS DATA REGISTER BYTE 3 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OS_DATA_BYTE_3 This is byte 3 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.12.5 ACPI OS COMMAND REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI_OSS_COMMAND Writes to the this register are aliased in the OS2EC Data EC Byte 0 Register. Writes to the this register also set the CMD and IBF bits in the OS STATUS OS Register	W	0h	RESET_SYS

12.12.6 OS STATUS OS REGISTER

This read-only register is aliased to the EC STATUS Register. The EC STATUS Register has read write access.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UD0B User Defined	R	0b	RESET_SYS
6	SMI_EVT This bit is set when an SMI event is pending; i.e., the ACPI_EC is requesting an SMI query; This bit is cleared when no SMI events are pending. This bit is an ACPI_EC-maintained software flag that is set when the ACPI_EC has detected an internal event that requires system management interrupt handler attention. The ACPI_EC sets this bit before generating an SMI. **Note:** The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI and SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the OS STATUS OS Register are software flags and this block do not initiate SMI or SCI events.	R	0b	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	SCI_EVT This bit is set by software when an SCI event is pending; i.e., the ACPI_EC is requesting an SCI query; SCI Event flag is clear when no SCI events are pending. This bit is an ACPI_EC-maintained software flag that is set when the embedded controller has detected an internal event that requires operating system attention. The ACPI_EC sets this bit before generating an SCI to the OS. **Note:** The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI and SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the OS STATUS OS Register are software flags and this block do not initiate SMI or SCI events.	R	0b	RESET_SYS
4	BURST The BURST bit is set when the ACPI_EC is in Burst Mode for polled command processing; the BURST bit is cleared when the ACPI_EC is in Normal mode for interrupt-driven command processing. The BURST bit is an ACPI_EC-maintained software flag that indicates the embedded controller has received the Burst Enable command from the host, has halted normal processing, and is waiting for a series of commands to be sent from the host. Burst Mode allows the OS or system management handler to quickly read and write several bytes of data at a time without the overhead of SCIs between commands. The BURST bit is maintained by ACPI_EC software, only.	R	0b	RESET_SYS
3	CMD This bit is set when the OS2EC Data EC Byte 0 Register contains a command byte written into ACPI OS COMMAND Register; this bit is cleared when the OS2EC DATA BYTES[3:0] contains a data byte written into the ACPI-OS DATA BYTES[3:0]. This bit is hardware controlled: • ACPI_OS writes to any of the four ACPI-OS DATA BYTES[3:0] bytes clears this bit • ACPI_OS writes to the ACPI OS COMMAND Register sets this bit. **Note:** This bit allows the embedded controller to differentiate the start of a command sequence from a data byte write operation.	R	0b	RESET_SYS
2	UD1B User Defined	R	0b	RESET_SYS

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	IBF The Input Buffer Full bit is set to indicate that a the ACPI_OS has written a command or data to the ACPI_EC and that data is ready. This bit is automatically cleared when data has been read by the ACPI_EC. **Note:** The setting and clearing of this IBF varies depending on the setting of the following bits: CMD bit in this register and FOUR_BYTE_ACCESS bit in the OS Byte Control Register. Three scenarios follow: 1. The IBF is set when the ACPI_OS writes to the ACPI OS COMMAND Register. This same write autonomously sets the CMD bit in this register. The IBF is cleared if the CMD bit in this register is set and the ACPI_EC reads from the OS2EC Data EC Byte 0 Register. **Note:** When CMD bit in this register is set the FOUR_BYTE_-ACCESS bit in the OS Byte Control Register has no impact on the IBF bit behavior. 2. A write by the to the ACPI_OS to the ACPI OS Data Register Byte 0 Register sets the IBF bit if the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is in the cleared to '0' state prior to this write. This same write autonomously clears the CMD bit in this register. A read of the OS2EC Data EC Byte 0 Register clears the IBF bit if the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is in the cleared to '0' state prior to this read. 3. A write by the to the ACPI_OS to the ACPI OS Data Register Byte 3 Register sets the IBF bit if the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is in the set to '1' state prior to this write. This same write autonomously clears the CMD bit in this register. A read of the OS2EC Data EC Byte 3 Register clears the IBF bit if the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is in the set to '1' state prior to this read. An IBF interrupt signals the ACPI_EC that there is data available. The ACPI Specification usage model is as follows: 1. The ACPI_EC reads the EC STATUS Register and sees the IBF flag set, 2. The ACPI_EC reads all the data available in the OS2EC DATA BYTES[3:0]. This causes the IBF bit to be automatically cleared by hardware. 3. The ACPI_EC must then generate a software interrupt to alert the ACPI_OS that the data has been read and that the host is free to write more data to the ACPI_EC as needed.	R	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	OBF The Output Buffer Full bit is set to indicate that a the ACPI_EC has written a data to the ACPI_OS and that data is ready. This bit is automatically cleared when all the data has been read by the ACPI_OS. **Note:** The setting and clearing of this OBF varies depending on the setting FOUR_BYTE_ACCESS bit in the OS Byte Control Register. Two scenarios follow: 1. The OBF bit is set if the Four Byte Access bit in the OS Byte Control Register is '0' when the ACPI_EC writes to the EC2OS Data EC Byte 0 Register. The OBF is cleared if the Four Byte Access bit in the OS Byte Control Register is cleared to '0' when the ACPI_OS reads from the ACPI OS Data Register Byte 0 Register. 2. The OBF is set if the Four Byte Access bit in the OS Byte Control Register is set to '1' when the ACPI_EC writes to the EC2OS Data EC Byte 3 Register. The OBF is cleared if the Four Byte Access bit in the OS Byte Control Register is set to '1' when the ACPI_OS reads from the ACPI OS Data Register Byte 3 Register. The ACPI Specification usage model is as follows: 1. The ACPI_EC must generate a software interrupt (See the note in Section 12.8, "Interrupts") to alert the ACPI_OS that the data is available. 2. The ACPI_OS reads the OS STATUS OS Register and sees the OBF flag set, the ACPI_OS reads all the data available in the ACPI-OS DATA BYTES[3:0]. 3. The ACPI_OS reads all the data available in the ACPI-OS DATA BYTES[3:0]. This causes the OBF bit to be automatically cleared by hardware and the associated OBE interrupt to be asserted.	R	0h	RESET_SYS

12.12.7 OS BYTE CONTROL REGISTER

This register is aliased to the EC Byte Control Register. No behavioral differences occur due to address aliasing.

Offset	05			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	FOUR_BYTE_ACCESS When this bit is set to '1', the ACPI Embedded Controller Interface (ACPI-ECI) accesses four bytes through the ACPI-OS DATA BYTES[3:0]. When this bit is cleared to '0', the ACPI Embedded Controller Interface (ACPI-ECI) accesses one byte through the ACPI OS Data Register Byte 0 Register. The corresponds to Legacy Mode described in Section 12.10, "Description". This bit effects the behavior of the IBF and OBF bits in the OS STATUS OS Register. See also Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed description of access rules.	R	0b	RESET_SYS

Note: The ACPI_OS access Base Address Register (BAR) should be configured to match the access width selected by the Four Byte Access bit in the OS Byte Control Register. This BAR in not described in this chapter.

12.13 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the ACPI Embedded Controller Interface (ACPI-ECI) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 12-10: REGISTER SUMMARY

Offset	Register Name
100h	EC2OS Data EC Byte 0 Register
101h	EC2OS Data EC Byte 1 Register
102h	EC2OS Data EC Byte 2 Register
103h	EC2OS Data EC Byte 3 Register
104h	EC STATUS Register
105h	EC Byte Control Register
106h	Reserved
107h	Reserved
108h	OS2EC Data EC Byte 0 Register
109h	OS2EC Data EC Byte 1 Register
10Ah	OS2EC Data EC Byte 2 Register
10Bh	OS2EC Data EC Byte 3 Register

12.13.1 OS2EC DATA EC BYTE 0 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

© 2020 Microchip Technology Inc.

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS_TO_EC_DATA_BYTE_0 This is byte 0 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.13.1.1 OS2EC DATA BYTES[3:0]

When the CMD bit in the OS STATUS OS Register is cleared to '0', reads by the ACPI_EC from the OS2EC DATA BYTES[3:0] are aliased to the ACPI-OS DATA BYTES[3:0].

All access to the OS2EC DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

When the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is cleared to '0', the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the OS STATUS OS Register.
2. Reads from the OS2EC Data EC Byte 0 Register clears the IBF bit in the OS STATUS OS Register.
3. All reads from OS2EC DATA BYTES[3:1] return 00h without error.
4. Access to OS2EC DATA BYTES[3:1 has no effect on the IBF and OBF bits in the OS STATUS OS Register.

When the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is set to '1', the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the OS STATUS OS Register.
2. Reads from the OS2EC Data EC Byte 3 Register clears the IBF bit in the OS STATUS OS Register.

12.13.2 OS2EC DATA EC BYTE 1 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	109h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC_DATA_ BYTE_1 This is byte 1 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.13.3 OS2EC DATA EC BYTE 2 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	10Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC_DATA_BYTE_2 This is byte 2 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.13.4 OS2EC DATA EC BYTE 3 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	10Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC_DATA_BYTE_3 This is byte 3 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.13.5 EC2OS DATA EC BYTE 0 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_0 This is byte 0 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.13.5.1 EC2OS DATA BYTES[3:0]

Writes by the ACPI_EC to the EC2OS DATA BYTES[3:0] are aliased to the ACPI-OS DATA BYTES[3:0]

All access to the EC2OS DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

When the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is cleared to '0', the following access rules apply:

1. Writes to the EC2OS Data EC Byte 0 Register set the OBF bit in the OS STATUS OS Register.
2. Reads from the EC2OS DATA BYTES[3:0] have no effect on the IBF bit in the OS STATUS OS Register.
3. All reads from EC2OS DATA BYTES[3:1] return 00h without error.
4. All writes to EC2OS DATA BYTES[3:1] complete without error but the data are not registered.
5. Access to EC2OS DATA BYTES[3:1] have no effect on the IBF and OBF bits in the OS STATUS OS Register.

When the FOUR_BYTE_ACCESS bit in the OS Byte Control Register is set to '1', the following access rules apply:

1. Writes to the EC2OS Data EC Byte 3 Register set the OBF bit in the OS STATUS OS Register.
2. Reads from the EC2OS DATA BYTES[3:0] have no effect on the IBF bit in the OS STATUS OS Register.

12.13.6 EC2OS DATA EC BYTE 1 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	101h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_1 This is byte 1 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

12.13.7 EC2OS DATA EC BYTE 2 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	102h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_2 This is byte 2 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.13.8 EC2OS DATA EC BYTE 3 REGISTER

This register is aliased; see Section 12.12.1.1, "ACPI-OS DATA BYTES[3:0]", Section 12.13.1.1, "OS2EC DATA BYTES[3:0]", and Section 12.13.5.1, "EC2OS DATA BYTES[3:0]" for detailed descriptions of access rules.

Offset	103h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS_DATA_BYTE_3 This is byte 3 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	RESET_SYS

12.13.9 EC STATUS REGISTER

This register is aliased to the OS STATUS OS Register. The OS STATUS OS Register is a read only version of this register.

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UD0A User Defined	R/W	0b	RESET_SYS
6	SMI_EVT See the SMI_EVT bit in the OS STATUS OS Register for the bit description.	R/W	0b	RESET_SYS
5	SCI_EVT See the SMI_EVT bit in the OS STATUS OS Register for the bit description.	R/W	0b	RESET_SYS
4	BURST See the BURST bit in the OS STATUS OS Register for the bit description.	R/W	0b	RESET_SYS
3	CMD See the CMD bit in the OS STATUS OS Register for the bit description.	R	0b	RESET_SYS
2	UD1A User Defined	R/W	0b	RESET_SYS
1	IBF See the IBF bit in the OS STATUS OS Register for the bit description.	R	0h	RESET_SYS

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	OBF See the OBF bit in the OS STATUS OS Register for the bit description.	R	0h	RESET_SYS

Note:	The IBF and OBF bits are not de-asserted by hardware when the host is powered off, or the eSPI interface powers down; for example, following system state changes S3->S0, S5->S0, G3-> S0. For further information on how these bits are cleared, refer to IBF and OBF bit descriptions in the STATUS OS-Register definition.

12.13.10 EC BYTE CONTROL REGISTER

This register is aliased to the OS Byte Control Register. The OS Byte Control Register is a read only version of this register.

Offset	105h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	FOUR_BYTE_ACCESS See the FOUR_BYTE_ACCESS bit in the OS Byte Control Register for the bit description.	R/W	0b	RESET_SYS

© 2020 Microchip Technology Inc.

13.0 ACPI PM1 BLOCK

13.1 Introduction

The MEC152x supports ACPI as described in this section. These features comply with the ACPI Specification through a combination of hardware and EC software.

13.2 References

ACPI Specification, Revision 1.0

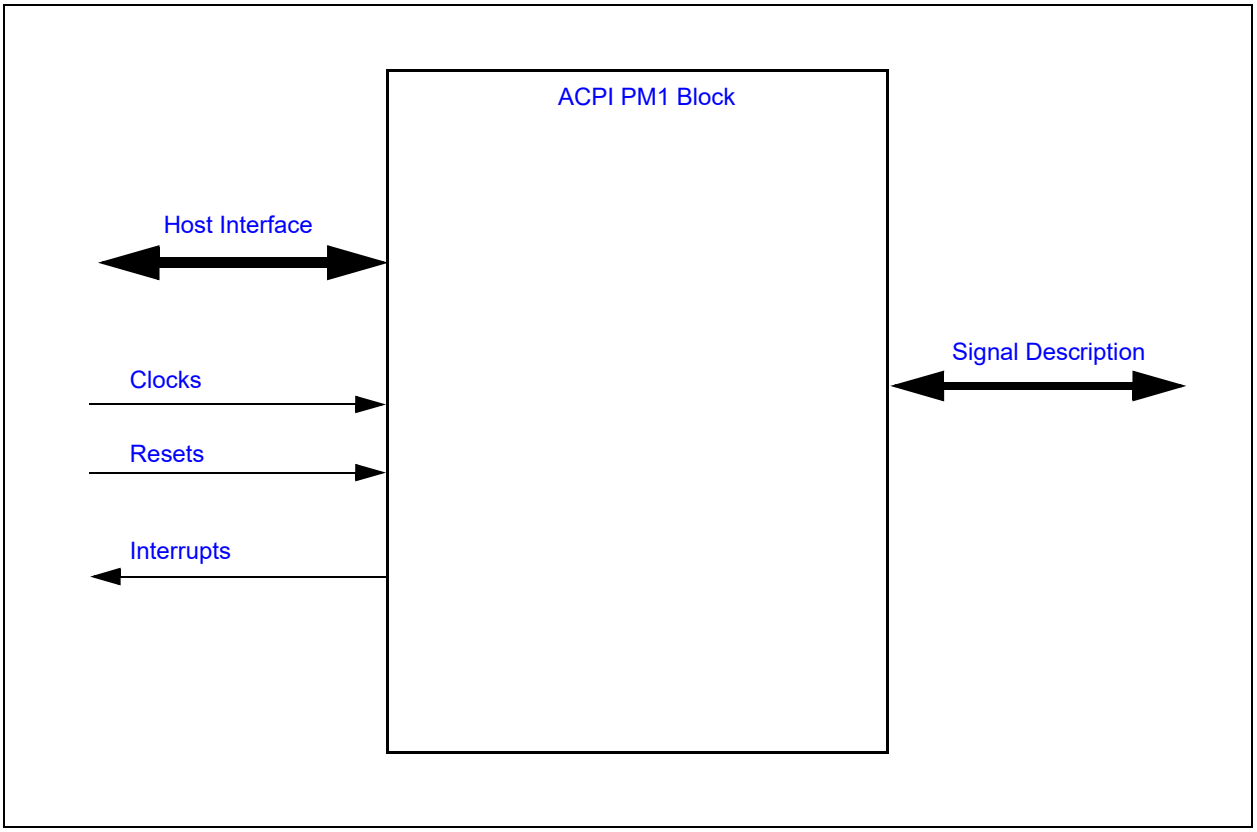
13.3 Terminology

None

13.4 Interface

This block is an IP block designed to be incorporated into a chip. It is designed to be accessed externally via the pin interface and internally via a registered host interface. The following diagram illustrates the various interfaces to the block.

FIGURE 13-1: I/O DIAGRAM OF BLOCK



13.5 Signal Description

Table 13-1, "ACPI PM1 Signal Description Table" lists the signals that are typically routed to the pin interface.

TABLE 13-1: ACPI PM1 SIGNAL DESCRIPTION TABLE

Name	Direction	Description
nEC_SCI	Output	Any or all of the PWRBTN_STS, SLPBTN_STS, and RTC_STS bits in the Power Management 1 Status 2 Register can assert the nEC_SCI pin if enabled by the associated bits in the Power Management 1 Enable 2 Register register. The EC_SCI_STS bit in the EC_PM_STS Register register can also be used to generate an SCI on the nEC_SCI pin.

13.6 Host Interface

The registers defined for the ACPI PM1 Block are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

13.7 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

13.7.1 POWER DOMAINS

TABLE 13-2: POWER SOURCES

Name	Description
VTR_CORE	This power well sources the registers and logic in this block.

13.7.2 CLOCKS

This section describes all the clocks in the block, including those that are derived from the I/O Interface as well as the ones that are derived or generated internally.

TABLE 13-3: CLOCKS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).

13.7.3 RESETS

TABLE 13-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.

13.8 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 13-5: EC INTERRUPTS

Source	Description
PM1_CTL	This Interrupt is generated to the EC by the Host writing to the Power Management 1 Control 2 Register register
PM1_EN	This Interrupt is generated to the EC by the Host writing to the Power Management 1 Enable 2 Register register
PM1_STS	This Interrupt is generated to the EC by the Host writing to the Power Management 1 Status 2 Register register

© 2020 Microchip Technology Inc.

13.9 Low Power Modes

The ACPI PM1 Block may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

13.10 Description

This section describes the functions of the ACPI PM1 Block in more detail.

The MEC152x implements the ACPI fixed registers but includes only those bits that apply to the power button sleep button and RTC alarm events. The ACPI WAK_STS, SLP_TYP, and SLP_EN bits are also supported.

The MEC152x can generate SCI Interrupts to the Host. The functions described in the following sub-sections can generate a SCI event on the nEC_SCI pin. In the MEC152x, an SCI event is considered the same as an ACPI wakeup or runtime event.

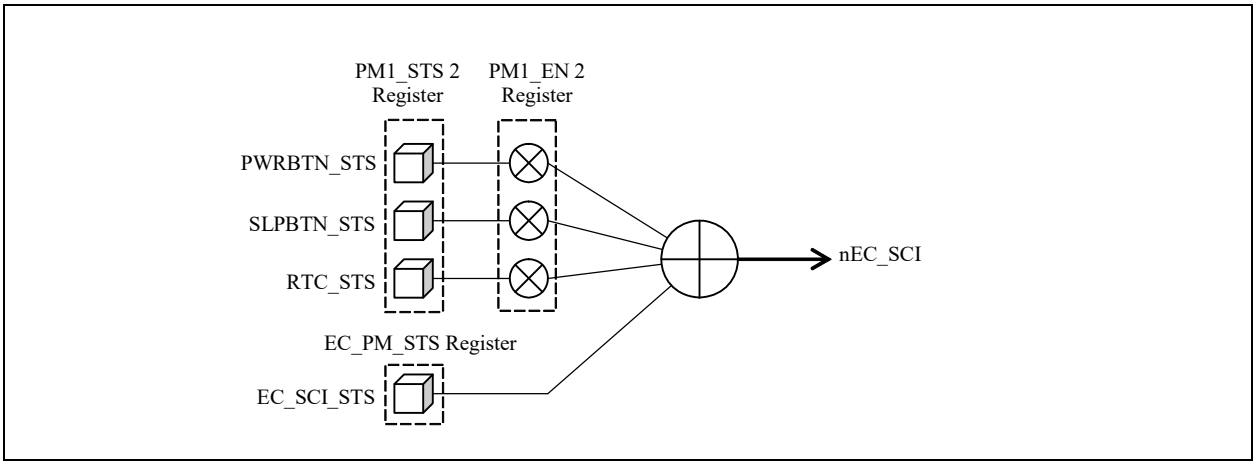
13.10.1 SCI EVENT-GENERATING FUNCTIONS

Event	Event Bit	Definition
Power Button with Override	PWRBTN_STS	The power button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. It also has a status and enable bit in the PM1_BLK of registers to indicate and control the power button override (fail-safe) event. These bits are not required by ACPI. The PWRBTN_STS bit is set by the Host to enable the generation of an SCI due to the power button event. The status bit is set by the EC when it generates a power button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC generates an SCI power management event.
	PWRBTNOR_STS	The power button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the power button override.The power button override event status bit is software Read/Writable by the EC; the enable bit is software read-only by the EC.The enable bit for the override event is located at bit 1 in the Power Management 1 Control Register 2 (PM1_CNTRL 2).The power button bit has a status and enable bit in the Runtime Registers to provide an SCI power management event on a button press The PWRBTNOR_STS bit is set by the Host to enable the generation of an SCI due to the power button override event. The status bit is set by the EC when it generates a power button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC generates an SCI power management event.
Sleep Button	SLPBTN_STS	The sleep button that has a status and an enable bit in the Runtime Registers to provide an SCI power management event on a button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. The SLPBTN_STS bit is set by the Host to enable the generation of an SCI due to the sleep button event. The status bit is set by the EC when it generates a sleep button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

Event	Event Bit	Definition
RTC Alarm	RTC_STS	The ACPI specification requires that the RTC alarm generate a hardware wake-up event from the sleeping state. The RTC alarm can be enabled as an SCI event and its status can be determined through bits in the Runtime Registers. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. The RTC_STS bit is set by the Host to enable the generation of an SCI due to the RTC alarm event. The status bit is set by the EC when the RTC generates an alarm event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

Figure 13-2 describes the relationship of PM1 Status and Enable bits to the nEC_SCI pin.

FIGURE 13-2: EC_SCI# INTERFACE



13.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the ACPI PM1 Block. Host access for each register listed in this table is defined as an offset in the Host address space to the Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for the ACPI PM1 Block shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

TABLE 13-6: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Power Management 1 Status 1 Register
01h	Power Management 1 Status 2 Register
02h	Power Management 1 Enable 1 Register
03h	Power Management 1 Enable 2 Register
04h	Power Management 1 Control 1 Register
05h	Power Management 1 Control 2 Register
06h	Power Management 2 Control 1 Register
07h	Power Management 2 Control 2 Register

© 2020 Microchip Technology Inc.

13.11.1 POWER MANAGEMENT 1 STATUS 1 REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	RES	-	-

13.11.2 POWER MANAGEMENT 1 STATUS 2 REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	WAK_STS This bit can be set or cleared by the EC. The Host writing a one to this bit can also clear this bit.	R/WC (Note 1)	00h	RESET_SYS
6:4	Reserved	RES	-	-
3	PWRBTNOR_STS This bit can be set or cleared by the EC to simulate a Power button override event status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated hardware event under software control.	R/WC (Note 1)	00h	RESET_SYS
2	RTC_STS This bit can be set or cleared by the EC to simulate a RTC status. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.	R/WC (Note 1)	00h	RESET_SYS
1	SLPBTN_STS This bit can be set or cleared by the EC to simulate a Sleep button status if the sleep state is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.	R/WC (Note 1)	00h	RESET_SYS
0	PWRBTN_STS This bit can be set or cleared by the EC to simulate a Power button status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.	R/WC (Note 1)	00h	RESET_SYS
Note 1:	These bits are set/cleared by the EC directly i.e., writing '1' sets the bit and writing '0' clears it. These bits can also be cleared by the Host software writing a one to this bit position and by RESET_SYS. Writing a 0 by the Host has no effect.			

13.11.3 POWER MANAGEMENT 1 ENABLE 1 REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	RES	-	-

13.11.4 POWER MANAGEMENT 1 ENABLE 2 REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	-	-
2	RTC_EN This bit can be read or written by the Host. It can be read by the EC.	R/W (Note 1)	00h	RESET_SYS
1	SLPBTN_EN This bit can be read or written by the Host. It can be read by the EC.	R/W (Note 1)	00h	RESET_SYS
0	PWRBTN_EN This bit can be read or written by the Host. It can be read by the EC.	R/W (Note 1)	00h	RESET_SYS
Note 1: These bits are read-only by the EC.				

13.11.5 POWER MANAGEMENT 1 CONTROL 1 REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	RES	0h	RESET_SYS

13.11.6 POWER MANAGEMENT 1 CONTROL 2 REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	RES	-	-
5	SLP_EN See Table 13-7.	See Table 13-7.	00h	RESET_SYS
4:2	SLP_TYP These bits can be set or cleared by the Host, read by the EC.	R/W (Note 1)	00h	RESET_SYS
1	PWRBTNOR_EN This bit can be set or cleared by the Host, read by the EC.	R/W (Note 1)	00h	RESET_SYS
0	Reserved	RES	-	-
Note 1: These bits are read-only by the EC.				

TABLE 13-7: SLP_EN DEFINITION

Host / EC	R/W	Description
Host	Read	Always reads 0
	Write	Writing a 0 has no effect, Writing a 1 sets this bit
EC	Read	Reads the value of the bit
	Write	Writing a 0 has no effect, Writing a 1 clears this bit

© 2020 Microchip Technology Inc.

13.11.7 POWER MANAGEMENT 2 CONTROL 1 REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	RES	-	-

13.11.8 POWER MANAGEMENT 2 CONTROL 2 REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Reserved	RES	-	-

13.12 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the ACPI PM1 Block Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 13-8: REGISTER SUMMARY

Offset	Register Name
100h	Power Management 1 Status 1 Register
101h	Power Management 1 Status 2 Register
102h	Power Management 1 Enable 1 Register
103h	Power Management 1 Enable 2 Register
104h	Power Management 1 Control 1 Register
105h	Power Management 1 Control 2 Register
106h	Power Management 2 Control 1 Register
107h	Power Management 2 Control 2 Register
110h	EC_PM_STS Register

Note: The Power Management Status, Enable and Control registers in Table 13-8, "Register Summary" are described in Section 13.11, "Runtime Registers".

13.12.1 EC_PM_STS REGISTER

Offset	110h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	UD	R/W	00h	RESET_SYS
0	EC_SCI_STS If the EC_SCI_STS bit is “1”, an interrupt is generated on the nEC_SCI pin.	R/W	00h	RESET_SYS

Note: This register is only accessed by the EC. There is no host access to this register.

14.0 EMBEDDED MEMORY INTERFACE (EMI)

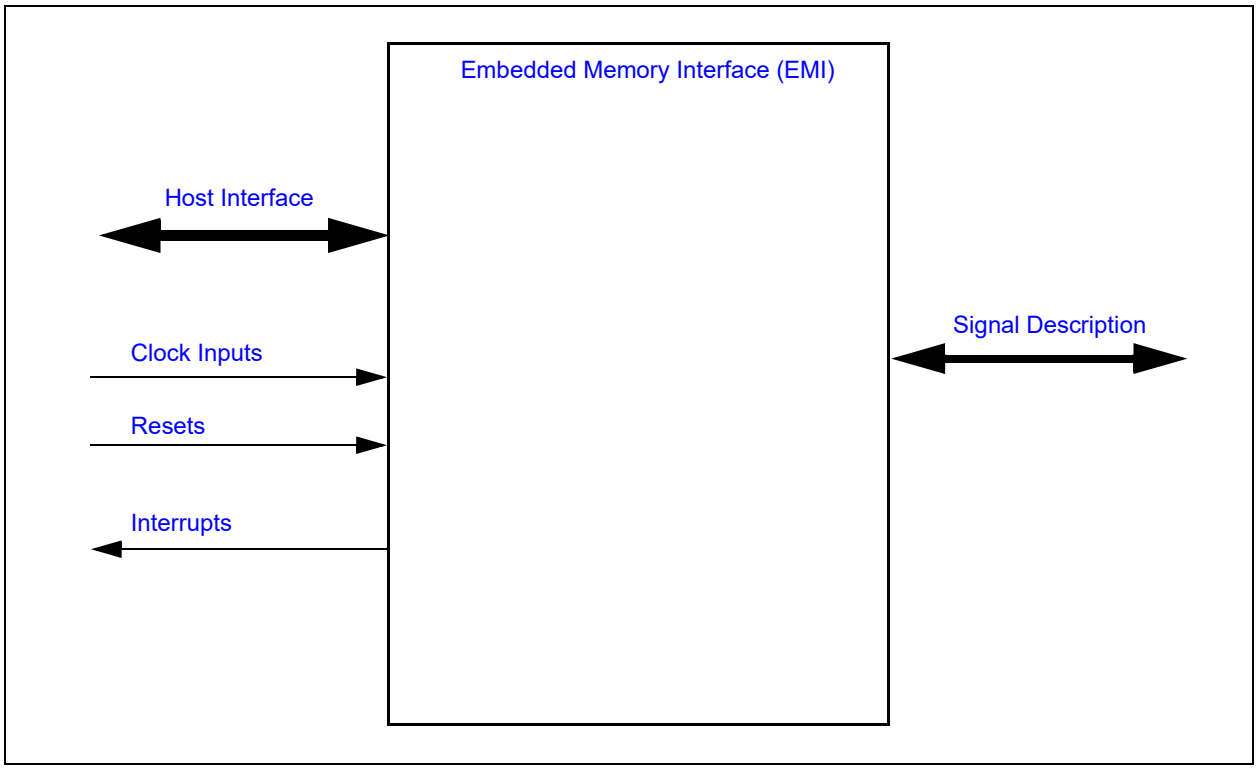
14.1 Introduction

The Embedded Memory Interface (EMI) provides a standard run-time mechanism for the system host to communicate with the Embedded Controller (EC) and other logical components. The Embedded Memory Interface includes 13 byte-addressable registers in the Host's address space, as well as 22 bytes of registers that are accessible only by the EC. The Embedded Memory Interface can be used by the Host to access bytes of memory designated by the EC without requiring any assistance from the EC. The EC may configure these regions of memory as read-only, write-only, or read/write capable.

14.2 Interface

This block is designed to be accessed externally via pin interface and internally via a register interface.

FIGURE 14-1: I/O DIAGRAM OF BLOCK



14.3 Signal Description

Name	Name	Description
nEMI_INT	OUTPUT	Active-low signal asserted when either the EC-to-Host or the Host_SWI_Event is asserted. This signal can be routed to nSMI and nPME inputs in the system as required.

14.4 Host Interface

The registers defined for the Embedded Memory Interface (EMI) are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

© 2020 Microchip Technology Inc.

14.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

14.5.1 POWER DOMAINS

TABLE 14-1: POWER SOURCES

Name	Description
VTR_CORE	The logic and registers implemented in this block reside on this single power well.

14.5.2 CLOCK INPUTS

This block has no special clocking requirements. Host register accesses are synchronized to the host bus clock and EC register accesses are synchronized to the EC bus clock, thereby allowing the transactions to complete in one bus clock.

14.5.3 RESETS

TABLE 14-2: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all the logic and register in this block.

14.6 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 14-3: SYSTEM INTERRUPTS

Source	Description
Host_SWI_Event	This interrupt source for the SERIRQ logic is generated when any of the EC_SWI bits are asserted and the corresponding EC_SWI_EN bits are asserted as well. This event is also asserted if the embedded controller (EC) writes the EC-to-HOST Mailbox Register.
EC-to-Host	This interrupt source for the SERIRQ logic is generated by the embedded controller (EC) writing the EC-to-HOST Mailbox Register.

TABLE 14-4: EC INTERRUPTS

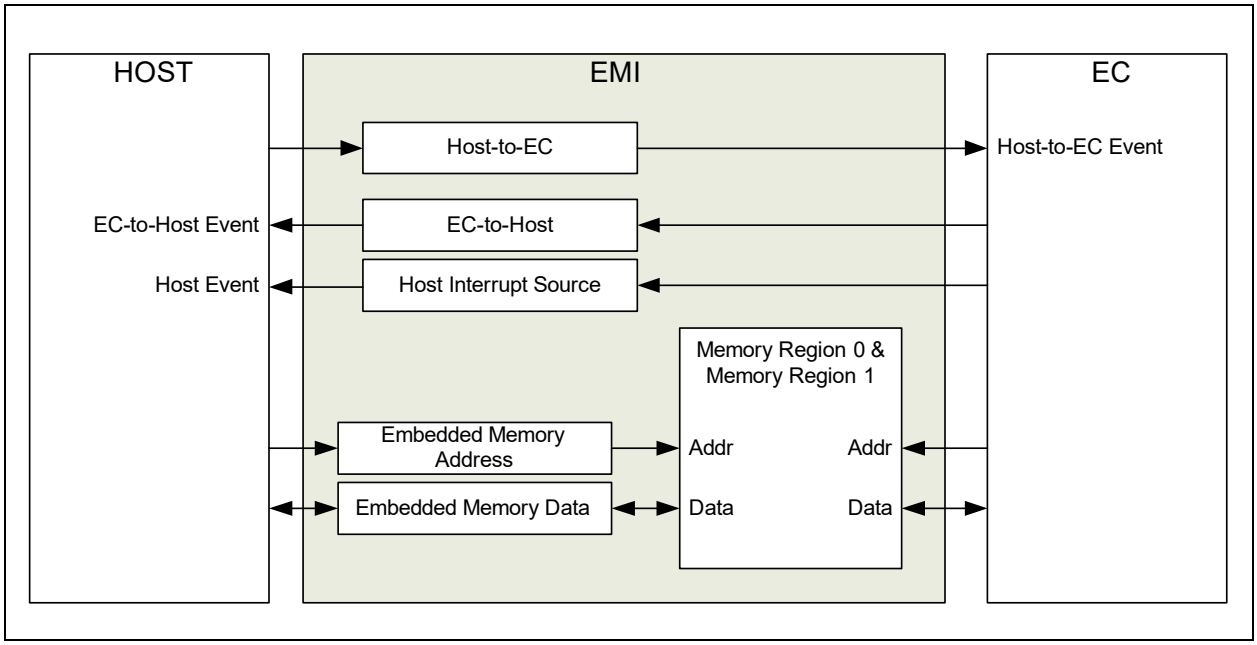
Source	Description
Host-to-EC	Interrupt source for the Interrupt Aggregator, generated by the host writing the HOST-to-EC Mailbox Register.

14.7 Low Power Modes

The Embedded Memory Interface (EMI) automatically enters low power mode when no transaction target it.

14.8 Description

FIGURE 14-2: EMBEDDED MEMORY INTERFACE BLOCK DIAGRAM



The Embedded Memory Interface (EMI) is composed of a mailbox, a direct memory interface, and an Application ID register.

The mailbox contains two registers, the HOST-to-EC Mailbox Register and the EC-to-HOST Mailbox Register, that act as a communication portal between the system host and the embedded controller. When the HOST-to-EC Mailbox Register is written an interrupt is generated to the embedded controller. Similarly, when the EC-to-HOST Mailbox Register is written an interrupt is generated to the system host. The source of the system host interrupt may be read in the Interrupt Source Register. These interrupt events may be individually prevented from generating a Host Event via the Interrupt Mask Register.

The direct memory interface, which is composed of a byte addressable 16-bit EC Address Register and a 32-bit EC Data Register, permits the Host to read or write a portion of the EC's internal address space. The embedded controller may enable up to two regions of the EC's internal address space to be exposed to the system host. The system host may access these memory locations without intervention or assistance from the EC.

The Embedded Memory Interface can be configured so that data transfers between the Embedded Memory Interface data bytes and the 32- bit internal address space may be multiple bytes, while Host I/O is always executed a byte at a time.

When the Host reads one of the four bytes in the Embedded Memory Interface data register, data from the internal 32-bit address space, at the address defined by the Embedded Memory Interface address register, is returned to the Host. This read access will load 1, 2, or 4 bytes into the Data register depending on the configuration of the ACCESS_TYPE bits. Similarly, writing one of the four bytes in the data register will write the corresponding byte(s) from the data register into the internal 32-bit address space as indicated by the ACCESS_TYPE bits. This configuration option is done to ensure that data the EC treats as 16-bit or 32-bit will be consistent in the Host, even though one byte of the data may change between two or more 8-bit accesses by the Host.

In addition, there is an auto-increment function for the Embedded Memory Interface address register. When enabled, the Host can read or write blocks of memory in the 32- bit internal address space by repeatedly accessing the Embedded Memory Interface data register, without requiring Host updates to the Embedded Memory Interface address register.

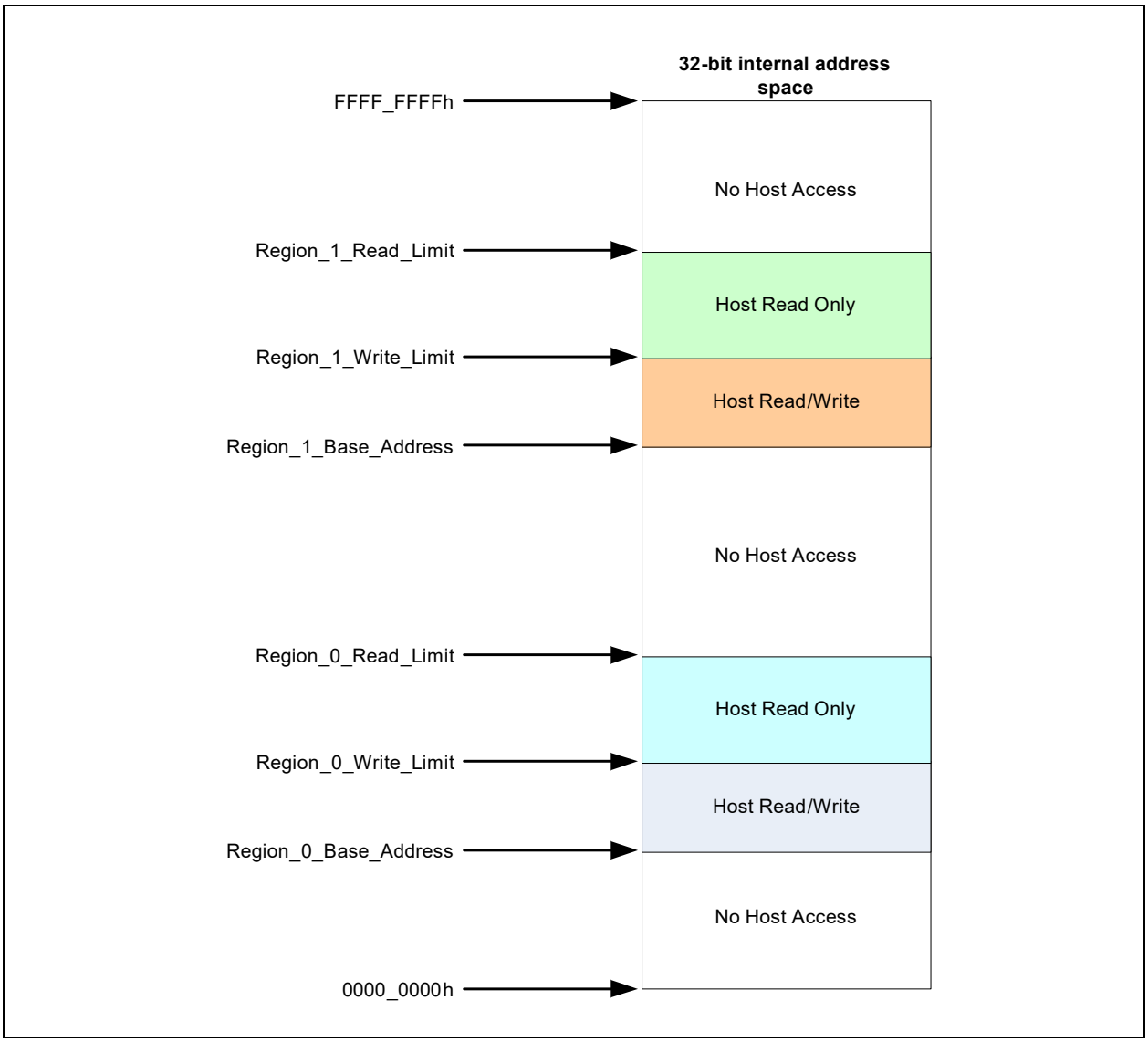
Finally, the Application ID Register may be used by the host to provide an arbitration mechanism if more than one software thread requires access through the EMI interface. See Section 14.8.4, "Embedded Memory Interface Usage" for more details.

© 2020 Microchip Technology Inc.

14.8.1 EMBEDDED MEMORY MAP

Each Embedded Memory interface provides direct access for the Host into two windows in the EC 32-bit internal address space. This mapping is shown in Figure 14-3, "Embedded Memory Addressing":

FIGURE 14-3: EMBEDDED MEMORY ADDRESSING



The Base addresses, the Read limits and the Write limits are defined by registers that are in the EC address space and cannot be accessed by the Host. In each region, the Read limit need not be greater than the Write limit. The regions can be contiguous or overlapping. For example, if the Region 0 Read limit is set to 0 and the Write limit is set to a positive number, then the Embedded Memory interface defines a region in the EC memory that the EC can read and write but is write-only for the host. This might be useful for storage of security data, which the Host might wish to send to the EC but should not be readable in the event a virus invades the Host.

Each window into the EC memory can be as large as 32k bytes in the 32-bit internal address space.

14.8.2 EC DATA REGISTER

The 4 1-byte EC Data Byte registers function as a 32-bit register, which creates a 4 byte window into the Memory REGION being accessed. The 4-byte window is always aligned on a 4-byte boundary. Depending on the read/write configuration of the memory region being accessed, the bytes may be extracted from or loaded into memory as a byte, word, or a DWord. The ACCESS_TYPE determines the size of the memory access. The address accessed is determined by the two EC_Address byte registers, which together function as a 15-bit EC Address Register.

- A write to the EC Data Register when the EC Address is in a read-only or a no-access region, as defined by the Memory Base and Limit registers, will update the EC Data Register but memory will not be modified.
- A read to the EC Data Register when the EC Address is in a no-access region, as defined by the Memory Base and Limit registers, will not trigger a memory read and will not modify the EC Data Register. In auto-increment mode (ACCESS_TYPE=11b), reads of Byte 3 of the EC Data Register will still trigger increments of the EC Address Register when the address is out of bounds, while writes of Byte 3 will not.

14.8.3 ACCESS TYPES

The access type field (ACCESS_TYPE in the EC Address LSB Register) defines the type of host access that occurs when the EC Data Register is read or written.

11: Auto-increment 32-bit access. This defines a 32-bit access, as in the 10 case. In addition, any read or write of Byte 3 in the EC Data Register causes the EC Data Register to be incremented by 1. That is, the EC_Address field will point to the next 32-bit double word in the 32- bit internal address space.

10: 32-bit access. A read of Byte 0 in the EC Data Register causes the 32 bits in the 32- bit internal address space at an offset of EC_Address to be loaded into the entire EC Data Register. The read then returns the contents of Byte 0. A read of Byte 1, Byte 2 or Byte 3 in the EC Data Register returns the contents of the register, without any update from the 32- bit internal address space.

A write of Byte 3 in the EC Data Register causes the EC Data Register to be written into the 32 bits in the 32- bit internal address space at an offset of EC_Address. A write of Byte 0, Byte 1 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the 32- bit internal address space.

01: 16-bit access. A read of Byte 0 in the EC Data Register causes the 16 bits in the 32- bit internal address space at an offset of EC_Address to be loaded into Byte 0 and Byte 1 of the EC Data Register. The read then returns the contents of Byte 0. A read of Byte 2 in the EC Data Register causes the 16 bits in the 32- bit internal address space at an offset of EC_Address+2 to be loaded into Byte 2 and Byte 3 of the EC Data Register. The read then returns the contents of Byte 2. A read of Byte 1 or Byte 3 in the EC Data Register return the contents of the register, without any update from the 32- bit internal address space.

A write of Byte 1 in the EC Data Register causes Bytes 1 and 0 of the EC Data Register to be written into the 16 bits in the 32- bit internal address space at an offset of EC_Address. A write of Byte 3 in the EC Data Register causes Bytes 3 and 2 of the EC Data Register to be written into the 16 bits in the 32- bit internal address space at an offset of EC_Address+2. A write of Byte 0 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the 32- bit internal address space.

00: 8-bit access. Any byte read of Byte 0 through Byte 3 in the EC Data Register causes the corresponding byte within the 32-bit double word addressed by EC_Address to be loaded into the byte of EC Data Register and returned by the read. Any byte write to Byte 0 through Byte 3 in the EC Data Register writes the corresponding byte within the 32-bit double word addressed by EC_Address, as well as the byte of the EC Data Register.

14.8.4 EMBEDDED MEMORY INTERFACE USAGE

The Embedded Memory Interface provides a generic facility for communication between the Host and the EC and can be used for many functions. Some examples are:

- Virtual registers. A block of memory in the 32-bit internal address space can be used to implement a set of virtual registers. The Host is given direct read-only access to this address space, referred to as peek mode. The EC may read or write this memory as needed.
- Program downloading. Because the Instruction Closely Coupled Memory is implemented in the same 32-bit internal address space, the Embedded Memory Interface can be used by the Host to download new program segments for the EC in the upper 32KB SRAM. The Read/Write window would be configured by the Host to point to the beginning of the loadable program region, which could then be loaded by the Host.
- Data exchange. The Read/Write portion of the memory window can be used to contain a communication packet. The Host, by default, “owns” the packet, and can write it at any time. When the Host wishes to communicate with the EC, it sends the EC a command, through the Host-to-EC message facility, to read the packet and perform

© 2020 Microchip Technology Inc.

some operations as a result. When it is completed processing the packet, the EC can inform the Host, either through a message in the EC-to-Host channel or by triggering an event such as an SMI directly. If return results are required, the EC can write the results into the Read/Write region, which the Host can read directly when it is informed that the EC has completed processing. Depending on the command, the operations could entail update of virtual registers in the 32-bit internal address space, reads of any register in the EC address space, or writes of any register in the EC address space. Because there are two regions that are defined by the base registers, the memory used for the communication packet does not have to be contiguous with a set of virtual registers.

Because there are two Embedded Memory Interface memory regions, the Embedded Memory Interface cannot be used for more than two of these functions at a time. The Host can request that the EC switch from one function to another through the use of the Host-to-EC mailbox register.

The Application ID Register is provided to help software applications track ownership of an Embedded Memory Interface. An application can write the register with its Application ID, then immediately read it back. If the read value is not the same as the value written, then another application has ownership of the interface.

Note: The protocol used to pass commands back and forth through the Embedded Memory Interface Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Embedded Memory Interface registers to gain access to all of the EC registers.

14.9 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Embedded Memory Interface (EMI). Host access for each register listed in this table is defined as an offset in the Host address space to the EMI Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the Embedded Memory Interface (EMI) shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

The Embedded Memory Interface (EMI) can be accessed from the eSPI Host Interface, the ESPI Host Interface, or the internal embedded controller (EC). The following table summarizes the host access types supported for each interface.

TABLE 14-5: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	HOST-to-EC Mailbox Register
01h	EC-to-HOST Mailbox Register
02h	EC Address LSB Register
03h	EC Address MSB Register
04h	EC Data Byte 0 Register
05h	EC Data Byte 1 Register
06h	EC Data Byte 2 Register
07h	EC Data Byte 3 Register
08h	Interrupt Source LSB Register
09h	Interrupt Source MSB Register
0Ah	Interrupt Mask LSB Register
0Bh	Interrupt Mask MSB Register
0Ch	Application ID Register

14.9.1 HOST-TO-EC MAILBOX REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST_EC_MBOX 8-bit mailbox used communicate information from the system host to the embedded controller. Writing this register generates an event to notify the embedded controller. The embedded controller has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The host must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to the HOST_EC_MBOX bit field in the HOST-to-EC Mailbox Register	R/W	0h	RESET_SYS

14.9.2 EC-TO-HOST MAILBOX REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_HOST_MBOX 8-bit mailbox used communicate information from the embedded controller to the system host. Writing this register generates an event to notify the system host. The system host has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The embedded controller must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to the EC_HOST_MBOX bit field in the EC-to-HOST Mailbox Register	R/WC	0h	RESET_SYS

14.9.3 EC ADDRESS LSB REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	EC_ADDRESS_LSB This field defines bits[7:2] of EC_Address [15:0]. Bits[1:0] of the EC_Address are always forced to 00b. The EC_Address is aligned on a DWord boundary. It is the address of the memory being accessed by EC Data Byte 0 Register, which is an offset from the programmed base address of the selected REGION.	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1:0	ACCESS_TYPE This field defines the type of access that occurs when the EC Data Register is read or written. 11b=Auto-increment 32-bit access. 10b=32-bit access. 01b=16-bit access. 00b=8-bit access. Each of these access types are defined in detail in Section 14.8.3, "Access Types".	R/W	0h	RESET_SYS

14.9.4 EC ADDRESS MSB REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	REGION The field specifies which of two segments in the 32-bit internal address space is to be accessed by the EC_Address[14:2] to generate accesses to the memory. 1=The address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 1 Register. 0=The address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 0 Register.	R/W	0h	RESET_SYS
6:0	EC_ADDRESS_MSB This field defines bits[14:8] of EC_Address. Bits[1:0] of the EC_Address are always forced to 00b. The EC_Address is aligned on a DWord boundary. It is the address of the memory being accessed by EC Data Byte 0 Register, which is an offset from the programmed base address of the selected REGION.	R/W	0h	RESET_SYS

14.9.5 EC DATA BYTE 0 REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_0 This is byte 0 (Least Significant Byte) of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 14.8.2, "EC Data Register".	R/W	0h	RESET_SYS

14.9.6 EC DATA BYTE 1 REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_1 This is byte 1 of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 14.8.2, "EC Data Register".	R/W	0h	RESET_SYS

14.9.7 EC DATA BYTE 2 REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_2 This is byte 2 of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 14.8.2, "EC Data Register".	R/W	0h	RESET_SYS

14.9.8 EC DATA BYTE 3 REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_DATA_BYTE_3 This is byte 3 (Most Significant Byte) of the 32-bit EC Data Register. Use of the Data Byte registers to access EC memory is defined in detail in Section 14.8.2, "EC Data Register".	R/W	0h	RESET_SYS

14.9.9 INTERRUPT SOURCE LSB REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI_LSB EC Software Interrupt Least Significant Bits. These bits are software interrupt bits that may be set by the EC to notify the host of an event. The meaning of these bits is dependent on the firmware implementation. Each bit in this field is cleared when written with a '1b'. The ability to clear the bit can be disabled by the EC if the corresponding bit in the Host Clear Enable Register is set to '0b'. This may be used by firmware for events that cannot be cleared while the event is still active.	R/WC	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	EC_WR EC Mailbox Write. This bit is set when the EC-to-HOST Mailbox Register has been written by the EC at offset 01h of the EC-Only registers. Note: there is no corresponding mask bit in the Interrupt Mask LSB Register.	R	0h	RESET_SYS

14.9.10 INTERRUPT SOURCE MSB REGISTER

Offset	09h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_SWI_MSB EC Software Interrupt Most Significant Bits. These bits are software interrupt bits that may be set by the EC to notify the host of an event. The meaning of these bits is dependent on the firmware implementation. Each bit in this field is cleared when written with a '1b'. The ability to clear the bit can be disabled by the EC. if the corresponding bit in the Host Clear Enable Register is set to '0b'. This may be used by firmware for events that cannot be cleared while the event is still active.	R/WC	0h	RESET_SYS

14.9.11 INTERRUPT MASK LSB REGISTER

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI_EN_LSB EC Software Interrupt Enable Least Significant Bits. Each bit that is set to '1b' in this field enables the generation of a Host Event interrupt by the corresponding bit in the EC_SWI field in the Interrupt Source LSB Register.	R/W	0h	RESET_SYS
0	TEST	R/W	0h	RESET_SYS

14.9.12 INTERRUPT MASK MSB REGISTER

Offset	0Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_SWI_EN_MSB EC Software Interrupt Enable Most Significant Bits. Each bit that is set to '1b' in this field enables the generation of a Host Event interrupt by the corresponding bit in the EC_SWI field in the Interrupt Source MSB Register.	R/W	0h	RESET_SYS

14.9.13 APPLICATION ID REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	APPLICATION_ID When this field is 00h it can be written with any value. When set to a non-zero value, writing that value will clear this register to 00h. When set to a non-zero value, writing any value other than the current contents will have no effect.	R/W	0h	RESET_SYS

14.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Embedded Memory Interface (EMI) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 14-6: EC-ONLY REGISTER SUMMARY

Offset	Register Name
100h	HOST-to-EC Mailbox Register
101h	EC-to-HOST Mailbox Register
104h	Memory Base Address 0 Register
108h	Memory Read Limit 0 Register
10Ah	Memory Write Limit 0 Register
10Ch	Memory Base Address 1 Register
110h	Memory Read Limit 1 Register
112h	Memory Write Limit 1 Register
114h	Interrupt Set Register
116h	Host Clear Enable Register

14.10.1 HOST-TO-EC MAILBOX REGISTER

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST_EC_MBOX 8-bit mailbox used communicate information from the system host to the embedded controller. Writing this register generates an event to notify the embedded controller. The embedded controller has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The host must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to the HOST_EC_MBOX bit field in the HOST-to-EC Mailbox Register.	R/WC	0h	RESET_SYS

© 2020 Microchip Technology Inc.

14.10.2 EC-TO-HOST MAILBOX REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_HOST_MBOX 8-bit mailbox used communicate information from the embedded controller to the system host. Writing this register generates an event to notify the system host. The system host has the option of clearing some or all of the bits in this register. This is dependent on the protocol layer implemented using the EMI Mailbox. The embedded controller must know this protocol to determine the meaning of the value that will be reported on a read. This bit field is aliased to EC_HOST_MBOX bit field in the EC-to-HOST Mailbox Register.	R/W	0h	RESET_SYS

14.10.3 MEMORY BASE ADDRESS 0 REGISTER

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	MEMORY_BASE_ADDRESS_0 This memory base address defines the beginning of region 0 in the Embedded Controller's 32-bit internal address space. Memory allocated to region 0 is intended to be shared between the Host and the EC. The region defined by this base register is used when bit 15 of the EC Address Register is 0. The access will be to a memory location at an offset defined by the EC_Address relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at Memory_Base_Address_0 + EC_Address.	R/W	0h	RESET_SYS
1:0	Reserved	RES	-	-

14.10.4 MEMORY READ LIMIT 0 REGISTER

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	RES	-	-
14:2	MEMORY_READ_LIMIT_0 Whenever a read of any byte in the EC Data Register is attempted, and bit 15 of EC_Address is 0, the field EC_Address[14:2] in the EC_Address_Register is compared to this field. As long as EC_Address[14:2] is less than this field the EC_Data_Register will be loaded from the 32-bit internal address space.	R/W	0h	RESET_SYS
1:0	Reserved	RES	-	-

14.10.5 MEMORY WRITE LIMIT 0 REGISTER

Offset	10Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	RES	-	-
14:2	MEMORY_WRITE_LIMIT_0 Whenever a write of any byte in EC DATA Register is attempted and bit 15 of EC_Address is 0, the field EC_ADDRESS_MSB in the EC_Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_0[14:2] the addressed bytes in the EC DATA Register will be written into the internal 32-bit address space. If EC_Address[14:2] is greater than or equal to the Memory_Write_Limit_0[14:2] no writes will take place.	R/W	0h	RESET_SYS
1:0	Reserved	RES	-	-

14.10.6 MEMORY BASE ADDRESS 1 REGISTER

Offset	10Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	MEMORY_BASE_ADDRESS_1 This memory base address defines the beginning of region 1 in the Embedded Controller's 32-bit internal address space. Memory allocated to region 1 is intended to be shared between the Host and the EC. The region defined by this base register is used when bit 15 of the EC Address Register is 1. The access will be to a memory location at an offset defined by the EC_Address relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at Memory_Base_Address_1 + EC_Address.	R/W	0h	RESET_SYS
1:0	Reserved	RES	-	-

14.10.7 MEMORY READ LIMIT 1 REGISTER

Offset	110h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	RES	-	-
14:2	MEMORY_READ_LIMIT_1 Whenever a read of any byte in the EC Data Register is attempted, and bit 15 of EC_ADDRESS is 1, the field EC_ADDRESS in the EC_Address_Register is compared to this field. As long as EC_ADDRESS is less than this value, the EC_Data_Register will be loaded from the 32-bit internal address space.	R/W	0h	RESET_SYS
1:0	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

14.10.8 MEMORY WRITE LIMIT 1 REGISTER

Offset	112h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15	Reserved	RES	-	-
14:2	MEMORY_WRITE_LIMIT_1 Whenever a write of any byte in EC DATA Register is attempted and bit 15 of EC_Address is 1, the field EC_Address[14:2] in the EC_Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_1[14:2] the addressed bytes in the EC DATA Register will be written into the internal 32-bit address space. If EC_Address[14:2] is greater than or equal to the Memory_Write_Limit_1[14:2] no writes will take place.	R/W	0h	RESET_SYS
1:0	Reserved	RES	-	-

14.10.9 INTERRUPT SET REGISTER

Offset	114h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:1	EC_SWI_SET EC Software Interrupt Set. This register provides the EC with a means of updating the Interrupt Source Registers. Writing a bit in this field with a ‘1b’ sets the corresponding bit in the Interrupt Source Register to ‘1b’. Writing a bit in this field with a ‘0b’ has no effect. Reading this field returns the current contents of the Interrupt Source Register.	R/WS	0h	RESET_SYS
0	Reserved	RES	-	-

14.10.10 HOST CLEAR ENABLE REGISTER

Offset	116h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:1	HOST_CLEAR_ENABLE When a bit in this field is ‘0b’, the corresponding bit in the Interrupt Source Register cannot be cleared by writes to the Interrupt Source Register. When a bit in this field is ‘1b’, the corresponding bit in the Interrupt Source Register can be cleared when that register bit is written with a ‘1b’. These bits allow the EC to control whether the status bits in the Interrupt Source Register are based on an edge or level event.	R/W	0h	RESET_SYS
0	Reserved	RES	-	-

15.0 MAILBOX INTERFACE

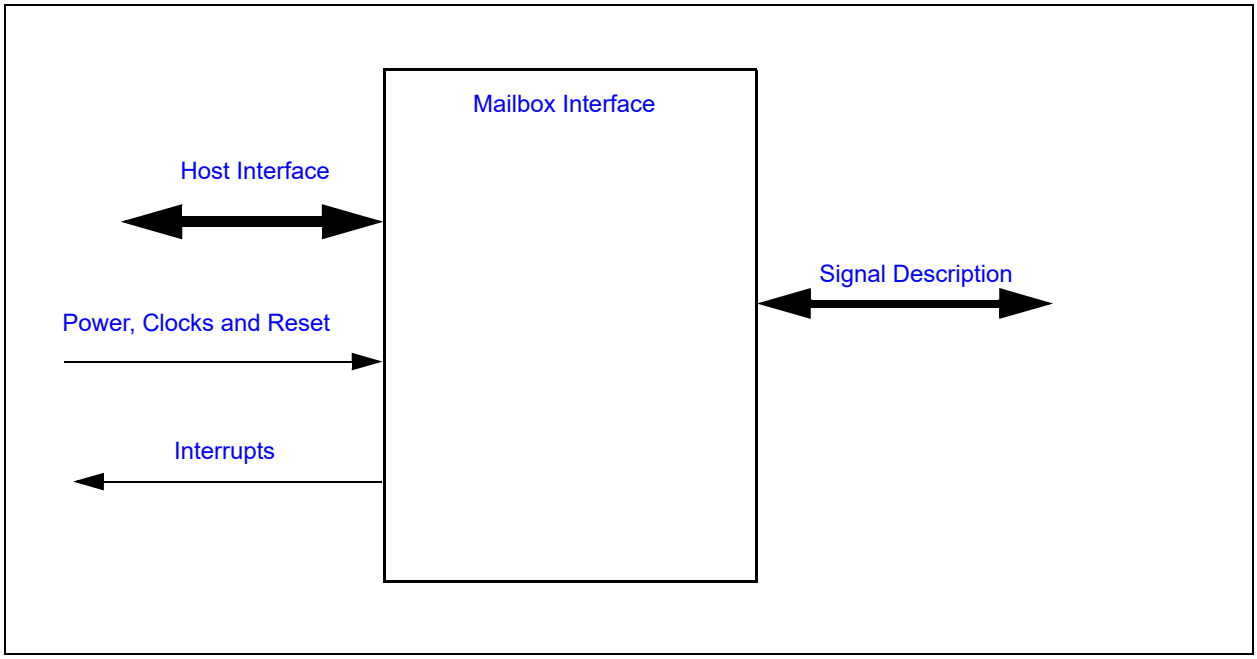
15.1 Overview

The Mailbox provides a standard run-time mechanism for the host to communicate with the Embedded Controller (EC)

15.2 Interface

This block is designed to be accessed externally via the pin interface and internally via registered host interface.

FIGURE 15-1: I/O DIAGRAM OF BLOCK



15.3 Signal Description

TABLE 15-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
nSMI	OUTPUT	SMI alert signal to the Host.

15.4 Host Interface

The registers defined for Mailbox Interface is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses"..

15.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

15.5.1 POWER DOMAINS

TABLE 15-2: POWER SOURCES

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

© 2020 Microchip Technology Inc.

15.5.2 CLOCK INPUTS

TABLE 15-3: CLOCK INPUTS

Name	Description
48MHz	This is the clock source for Mailbox logic.

15.5.3 RESETS

TABLE 15-4: RESET SIGNALS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.
RESET_VCC	This signal is asserted when the main power rail is asserted. The Host Access Port is reset when this signal is de-asserted.

15.6 Interrupts

TABLE 15-5: SYSTEM INTERRUPTS

Source	Description
MBX_Host_SERIRQ	This interrupt source for the SERIRQ logic is generated when the EC_WR bit is '1' and enabled by the EC_WR_EN bit.
MBX_Host_SMI	This interrupt source for the SERIRQ logic is generated when any of the EC_SWI bits are asserted and the corresponding EC_SWI_EN bit are asserted as well. This event is also asserted if the EC_WR/EC_WR_EN event occurs as well. This bit is also routed to the nSMI pin.

TABLE 15-6: EC INTERRUPTS

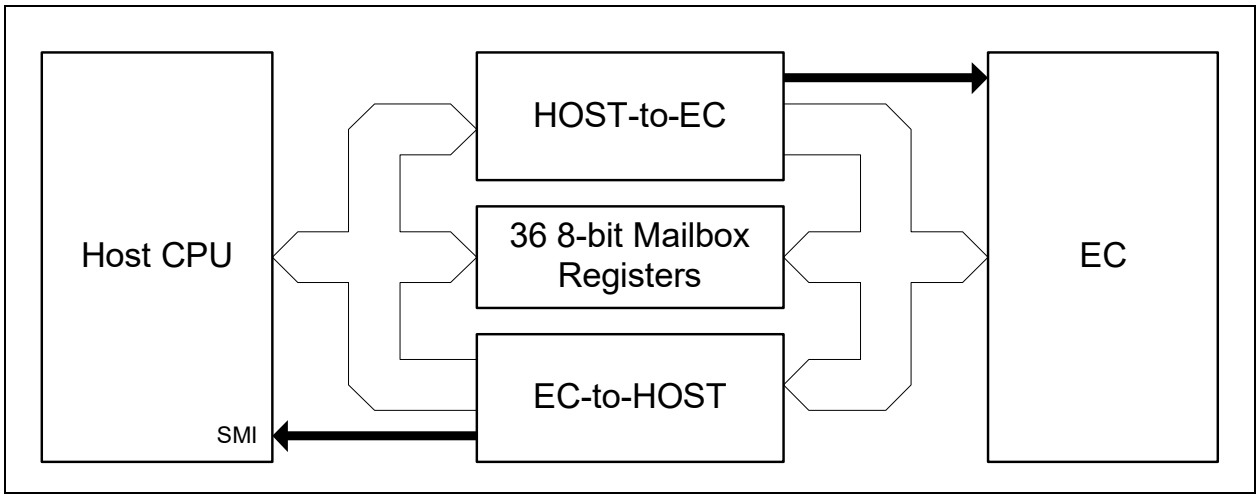
Source	Description
MBX	Interrupt generated by the host writing the HOST-to-EC Mailbox register.

15.7 Low Power Modes

The Mailbox automatically enters a low power mode whenever it is not actively.

15.8 Description

FIGURE 15-2: MAILBOX BLOCK DIAGRAM



15.8.1 HOST ACCESS PORT

The Mailbox includes a total of 36 index-addressable 8-bit Mailbox registers and a two byte Mailbox Registers Host Access Port. Thirty-two of the 36 index-addressable 8-bit registers are EC Mailbox registers, which can be read and written by both the EC and the Host. The remaining four registers are used for signaling between the Host and the EC. The Host Access Port consists of two 8-bit run-time registers that occupy two addresses in the HOST I/O space, MBX_INDEX Register and MBX_DATA Register. The Host Access Port is used by the host to access the 36 index-addressable 8-bit registers.

To access a Mailbox register once the Mailbox Registers Interface Base Address has been initialized, the Mailbox register index address is first written to the MBX Index port. After the Index port has been written, the Mailbox data byte can be read or written via the MBX data port.

The Host Access Port is intended to be accessed by the Host only, however it may be accessed by the EC at the Offset shown from its 32-bit internal address in Table 15-7, "Runtime Register Summary".

15.8.2 HOST INTERRUPT GENERATION

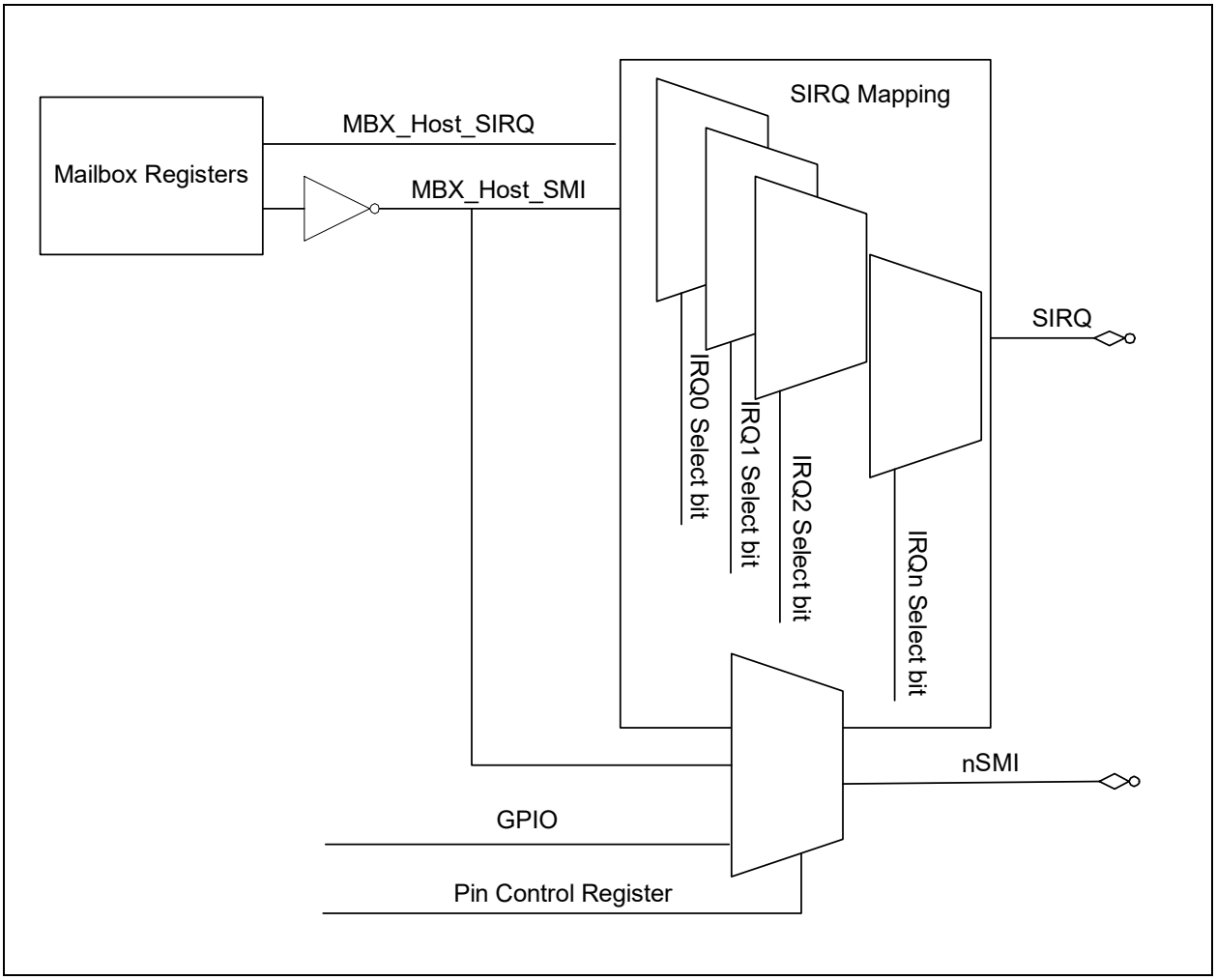
The Mailbox can generate a SERIRQ event for EC-to-HOST EC events, using the EC-to-Host Mailbox Register. This interrupt is routed to the SERIRQ block.

The Mailbox can also generate an SMI event, using SMI Interrupt Source Register. The SMI event can be routed to any frame in the SERIRQ stream as well as to the nSMI pin. The SMI event can be routed to nSMI pin by selecting the nSMI signal function in the associated GPIO Pin Control Register. The SMI event produces a standard active low frame on the serial IRQ stream and active low level on the open drain nSMI pin.

Routing for both the SERIRQ logic and the nSMI pin is shown in Figure 15-3.

© 2020 Microchip Technology Inc.

FIGURE 15-3: MAILBOX SERIRQ AND SMI ROUTING



15.8.3 EC MAILBOX CONTROL

The HOST-to-EC Mailbox Register and EC-to-Host Mailbox Register are designed to pass commands between the host and the EC. If enabled, these registers can generate interrupts to both the Host and the EC.

The two registers are not dual-ported, so the HOST BIOS and Keyboard BIOS must be designed to properly share these registers. When the host performs a write of the HOST-to-EC Mailbox Register, an interrupt will be generated and seen by the EC if unmasked. When the EC writes FFh to the Mailbox Register, the register resets to 00h, providing a simple means for the EC to inform the host that an operation has been completed.

When the EC writes the EC-to-Host Mailbox Register, an SMI may be generated and seen by the host if unmasked. When the Host CPU writes FFh to the register, the register resets to 00h, providing a simple means for the host to inform that EC that an operation has been completed.

Note:	The protocol used to pass commands back and forth through the Mailbox Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Mailbox registers to gain access to all of the EC registers.

15.9 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Mailbox Interface. Host access for each register listed in this table is defined as an offset in the Host address space to the Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for the Mailbox Interface shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

TABLE 15-7: RUNTIME REGISTER SUMMARY

Offset	Register Name
0h	MBX_INDEX Register
1h	MBX_DATA Register

15.9.1 MBX_INDEX REGISTER

Offset	0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	INDEX The index into the mailbox registers listed in Table 15-8, "Register Summary".	R/W	0h	RESET_VCC

15.9.2 MBX_DATA REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DATA Data port used to access the registers listed in Table 15-8, "Register Summary".	R/W	0h	RESET_VCC

15.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset shown in the "EC Offset" column to the Base Address for each instance of the Mailbox Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory". In addition, the registers can be accessed through the Host Access Port, at the indexes listed in the following tables as "MBX_INDEX".

TABLE 15-8: REGISTER SUMMARY

EC Offset	Host I/O Index (MBX_INDEX)	Register Name
100h	00h	HOST-to-EC Mailbox Register
104h	01h	EC-to-Host Mailbox Register
108h	02h	SMI Interrupt Source Register
10Ch	03h	SMI Interrupt Mask Register
110h	10h	Mailbox register [0]
	11h	Mailbox register [1]
	12h	Mailbox register [2]
	13h	Mailbox register [3]

© 2020 Microchip Technology Inc.

TABLE 15-8: REGISTER SUMMARY (CONTINUED)

EC Offset	Host I/O Index (MBX_INDEX)	Register Name
114h	14h	Mailbox register [4]
	15h	Mailbox register [5]
	16h	Mailbox register [6]
	17h	Mailbox register [7]
118h	18h	Mailbox register [8]
	19h	Mailbox register [9]
	1Ah	Mailbox register [A]
	1Bh	Mailbox register [B]
11Ch	1Ch	Mailbox register [C]
	1Dh	Mailbox register [D]
	1Eh	Mailbox register [E]
	1Fh	Mailbox register [F]
120h	20h	Mailbox register [10]
	21h	Mailbox register [11]
	22h	Mailbox register [12]
	23h	Mailbox register [13]
124h	24h	Mailbox register [14]
	25h	Mailbox register [15]
	26h	Mailbox register [16]
	27h	Mailbox register [17]
128h	28h	Mailbox register [18]
	29h	Mailbox register [19]
	2Ah	Mailbox register [1A]
	2Bh	Mailbox register [1B]
12Ch	2Ch	Mailbox register [1C]
	2Dh	Mailbox register [1D]
	2Eh	Mailbox register [1E]
	2Fh	Mailbox register [1F]

15.10.1 HOST-TO-EC MAILBOX REGISTER

Offset	100h			
MBX_ INDEX	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOST_EC_MBOX If enabled, an interrupt to the EC marked by the MBX bit in the Interrupt Aggregator will be generated whenever the Host writes this register. The interrupt is cleared when this register is read by the EC. This register is cleared when written with FFh. This field is Read/Write when accessed through the Host Access Port. When written at the EC offset, each bit in this field is cleared when written with a '1b'. Writes of '0b' have no effect.	Host Access Port: R/W EC: R/WC	0h	RESET_SYS

15.10.2 EC-TO-HOST MAILBOX REGISTER

Offset	104h			
MBX_ INDEX	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC_HOST_MBOX An EC write to this register will set bit EC_WR in the SMI Interrupt Source Register to '1b'. If enabled, this will generate a Host SMI or a Host SERIRQ. The SERIRQ is cleared when this register is read by the Host. This register is cleared when written with FFh. This field is Read/Write when accessed by the EC at the EC offset. When written through the Host Access Port, each bit in this field is cleared when written with a '1b'. Writes of '0b' have no effect.	Host Access Port: R/WC EC: R/W	0h	RESET_SYS

15.10.3 SMI INTERRUPT SOURCE REGISTER

Offset	108h			
MBX_ INDEX	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI EC Software Interrupt. An SERIRQ to the Host is generated when any bit in this register when this bit is set to '1b' and the corresponding bit in the SMI Interrupt Mask Register register is '1b'. This field is Read/Write when accessed by the EC at the EC offset. When written through the Host Access Port, each bit in this field is cleared when written with a '1b'. Writes of '0b' have no effect.	Host Access Port: R/WC EC: R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	108h			
MBX_ INDEX	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	EC_WR EC Mailbox Write. This bit is set automatically when the EC-to-Host Mailbox Register has been written. An SMI or SERIRQ to the Host is generated when n this bit is '1b' and the corresponding bit in the SMI Interrupt Mask Register register is '1b'. This bit is automatically cleared by a read of the EC-to-Host Mailbox Register through the Host Access Port. This bit is read-only when read through the Host Access Port. It is neither readable nor writable directly by the EC when accessed at the EC offset.	Host Access Port: R EC: -	0h	RESET_ SYS

15.10.4 SMI INTERRUPT MASK REGISTER

Offset	10Ch			
MBX_ INDEX	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	EC_SWI_EN EC Software Interrupt Enable. If this bit is '1b', the bit EC_WR in the SMI Interrupt Source Register is enabled for the generation of SERIRQ or nSMI events.	R/W	0h	RESET_ SYS
0	EC_WR_EN EC Mailbox Write.Interrupt Enable. Each bit in this field that is '1b' enables the generation of SERIRQ interrupts when the corresponding bit in the EC_SWI field in the SMI Interrupt Source Register is '1b'.	R/W	0h	RESET_ SYS

16.0 UART

16.1 Introduction

The 16550 UART (Universal Asynchronous Receiver/Transmitter) is a full-function Serial Port that supports the standard RS-232 Interface.

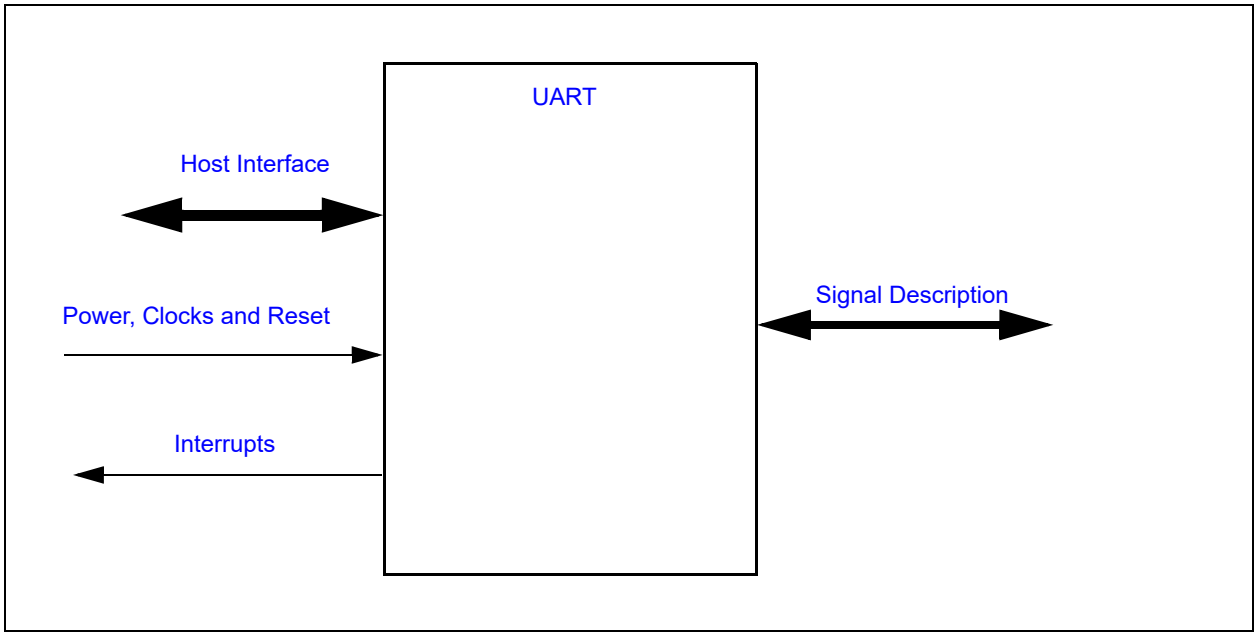
16.2 References

- EIA Standard RS-232-C specification

16.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 16-1: I/O DIAGRAM OF BLOCK



16.4 Signal Description

TABLE 16-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
DTR#	Output	Active low Data Terminal ready output for the Serial Port. Handshake output signal notifies modem that the UART is ready to transmit data. This signal can be programmed by writing to bit 1 of the Modem Control Register (MCR). **Note:** Defaults to tri-state on V3_DUAL power on.
DCD#	Input	Active low Data Carrier Detect input for the serial port. Handshake signal which notifies the UART that carrier signal is detected by the modem. The CPU can monitor the status of DCD# signal by reading bit 7 of Modem Status Register (MSR). A DCD# signal state change from low to high after the last MSR read will set MSR bit 3 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when DCD # changes state.

© 2020 Microchip Technology Inc.

TABLE 16-1: SIGNAL DESCRIPTION TABLE (CONTINUED)

Name	Direction	Description
DSR#	Input	Active low Data Set Ready input for the serial port. Handshake signal which notifies the UART that the modem is ready to establish the communication link. The CPU can monitor the status of DSR# signal by reading bit 5 of Modem Status Register (MSR). A DSR# signal state change from low to high after the last MSR read will set MSR bit 1 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when DSR# changes state.
RI#	Input	Active low Ring Indicator input for the serial port. Handshake signal which notifies the UART that the telephone ring signal is detected by the modem. The CPU can monitor the status of RI# signal by reading bit 6 of Modem Status Register (MSR). A RI# signal state change from low to high after the last MSR read will set MSR bit 2 to a 1. If bit 3 of Interrupt Enable Register is set, the interrupt is generated when RI# changes state.
RTS#	Output	Active low Request to Send output for the Serial Port. Handshake output signal notifies modem that the UART is ready to transmit data. This signal can be programmed by writing to bit 1 of the Modem Control Register (MCR). The hardware reset will reset the RTS# signal to inactive mode (high). RTS# is forced inactive during loop mode operation. Defaults to tri-state on V3_DUAL power on.
CTS#	Input	Active low Clear to Send input for the serial port. Handshake signal which notifies the UART that the modem is ready to receive data. The CPU can monitor the status of CTS# signal by reading bit 4 of Modem Status Register (MSR). A CTS# signal state change from low to high after the last MSR read will set MSR bit 0 to a 1. If bit 3 of the Interrupt Enable Register is set, the interrupt is generated when CTS# changes state. The CTS# signal has no effect on the transmitter.
TXD	Output	Transmit serial data output.
RXD	Input	Receiver serial data input.

16.5 Host Interface

The registers defined for UART is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

16.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

16.6.1 POWER DOMAINS

TABLE 16-2: POWER SOURCES

Name	Description
VTR_CORE	This Power Well is used to power the registers and logic in this block.

16.6.2 CLOCKS

TABLE 16-3: CLOCK INPUTS

Name	Description
UART_CLK	An external clock that may be used as an alternative to the internally-generated 1.8432MHz and 48MHz baud clocks. Selection between internal baud clocks and an external baud clock is configured by the CLK_SRC bit in the Configuration Select Register.
48MHz	This is the main clock domain. Because the clock input must be within ± 2% in order to generate standard baud rates, the 48MHz clock must be generated by a reference clock with better than 1% accuracy and locked to its frequency before the UART will work with the standard rates.

TABLE 16-4: BAUD CLOCKS

Name	Description
1.8432MHz	The UART requires a 1.8432 MHz ± 2% clock input for baud rate generation of standard baud rates up to 115,200 baud. It is derived from the system 48MHz clock domain.
48MHz	It may be used as an alternative to the 1.8432MHz clock, generating non-standard baud rates up to 1,500,000 baud.

16.6.3 RESETS

TABLE 16-5: RESET SIGNALS

Name	Description
RESET_SYS	This reset is asserted when VTR_CORE is applied.
RESET_HOST	This is an alternate reset condition, typically asserted when the main power rail is asserted.
RESET	This reset is determined by the POWER bit signal. When the power bit signal is 1, this signal is equal to RESET_VCC, if present. When the power bit signal is 0, this signal is equal to RESET_SYS.

16.7 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 16-6: SYSTEM INTERRUPTS

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 16-12, "Interrupt Control Table".

TABLE 16-7: EC INTERRUPTS

Source	Description
UART	The UART interrupt event output indicates if an interrupt is pending. See Table 16-12, "Interrupt Control Table".

16.8 Low Power Modes

The UART may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

© 2020 Microchip Technology Inc.

16.9 Description

The UART is compatible with the 16450, the 16450 ACE registers and the 16C550A. The UART performs serial-to-parallel conversions on received characters and parallel-to-serial conversions on transmit characters. Two sets of baud rates are provided. When the 1.8432 MHz source clock is selected, standard baud rates from 50 to 115.2K are available. When the source clock is 48MHz, baud rates up to 1,500K are available. The character options are programmable for 1 start; 1, 1.5 or 2 stop bits; even, odd, sticky or no parity; and prioritized interrupts. The UART contains a programmable baud rate generator that is capable of dividing the input clock signal by 1 to 32767. The UART is also capable of supporting the MIDI data rate. Refer to the Configuration Registers for information on disabling, powering down and changing the base address of the UART. The UART interrupt is enabled by programming OUT2 of the UART to logic “1.” Because OUT2 is logic “0,” it disables the UART's interrupt. The UART is accessible by both the Host and the EC.

16.9.1 PROGRAMMABLE BAUD RATE

The Serial Port contains a programmable Baud Rate Generator that is capable of dividing the internal clock source by any divisor from 1 to 32767. Unless an external clock source is configured, the clock source is either the 1.8432MHz clock source or the 48MHz clock source. The output frequency of the Baud Rate Generator is 16x the Baud rate. Two eight bit latches store the divisor in 16 bit binary format. These Divisor Latches must be loaded during initialization in order to ensure desired operation of the Baud Rate Generator. Upon loading either of the Divisor Latches, a 16 bit Baud counter is immediately loaded. This prevents long counts on initial load. If a 0 is loaded into the BRG registers, the output divides the clock by the number 3. If a 1 is loaded, the output is the inverse of the input oscillator. If a two is loaded, the output is a divide by 2 signal with a 50% duty cycle. If a 3 or greater is loaded, the output is low for 2 bits and high for the remainder of the count.

The following tables show possible baud rates.

TABLE 16-8: UART BAUD RATES USING CLOCK SOURCE 1.8432MHz

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
50	0	2304
75	0	1536
110	0	1047
134.5	0	857
150	0	768
300	0	384
600	0	192
1200	0	96
1800	0	64
2000	0	58
2400	0	48
3600	0	32
4800	0	24
7200	0	16
9600	0	12
19200	0	6
38400	0	3
57600	0	2
115200	0	1

© 2020 Microchip Technology Inc.

TABLE 16-9: UART BAUD RATES USING CLOCK SOURCE 48MHz

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
125000	1	24
136400	1	22
150000	1	20
166700	1	18
187500	1	16
214300	1	14
250000	1	12
300000	1	10
375000	1	8
500000	1	6
750000	1	4
1500000	1	2
3000000	1	1

16.10 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the UART. Host access for each register listed in this table is defined as an offset in the Host address space to the Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the UART shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

TABLE 16-10: RUNTIME REGISTER SUMMARY

DLAB Note 1	Offset	Register Name
0	0h	Receive Buffer Register
0	0h	Transmit Buffer Register
1	0h	Programmable Baud Rate Generator LSB Register
1	1h	Programmable Baud Rate Generator MSB Register
0	1h	Interrupt Enable Register
x	02h	FIFO Control Register
x	02h	Interrupt Identification Register
x	03h	Line Control Register
x	04h	Modem Control Register
x	05h	Line Status Register
x	06h	Modem Status Register
x	07h	Scratchpad Register
Note 1: DLAB is bit 7 of the Line Control Register.		

© 2020 Microchip Technology Inc.

16.10.1 RECEIVE BUFFER REGISTER

Offset	0h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	RECEIVED_DATA This register holds the received incoming data byte. Bit 0 is the least significant bit, which is transmitted and received first. Received data is double buffered; this uses an additional shift register to receive the serial data stream and convert it to a parallel 8 bit word which is transferred to the Receive Buffer register. The shift register is not accessible.	R	0h	RESET

16.10.2 TRANSMIT BUFFER REGISTER

Offset	0h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	TRANSMIT_DATA This register contains the data byte to be transmitted. The transmit buffer is double buffered, utilizing an additional shift register (not accessible) to convert the 8 bit data word to a serial format. This shift register is loaded from the Transmit Buffer when the transmission of the previous byte is complete.	W	0h	RESET

16.10.3 PROGRAMMABLE BAUD RATE GENERATOR LSB REGISTER

Offset	00h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	BAUD_RATE_DIVISOR_LSB See Section 16.9.1, "Programmable Baud Rate".	R/W	0h	RESET

16.10.4 PROGRAMMABLE BAUD RATE GENERATOR MSB REGISTER

Offset	01h (DLAB=1)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	BAUD_CLK_SEL If CLK_SRC is ‘0’: • 0=The baud clock is derived from the 1.8432MHz. • 1=IThe baud clock is derived from the 48MHz. If CLK_SRC is ‘1’: • This bit has no effect	R/W	0h	RESET
6:0	BAUD_RATE_DIVISOR_MSB See Section 16.9.1, "Programmable Baud Rate".	R/W	0h	RESET

16.10.5 INTERRUPT ENABLE REGISTER

The lower four bits of this register control the enables of the five interrupt sources of the Serial Port interrupt. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of this register. Similarly, setting the appropriate bits of this register to a high, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and disables any Serial Port interrupt out of the MEC152x. All other system functions operate in their normal manner, including the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are described below.

Offset	01h (DLAB=0)			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	Reserved	RES	-	-
3	EMSI This bit enables the MODEM Status Interrupt when set to logic “1”. This is caused when one of the Modem Status Register bits changes state.	R/W	0h	RESET
2	ELSI This bit enables the Received Line Status Interrupt when set to logic “1”. The error sources causing the interrupt are Overrun, Parity, Framing and Break. The Line Status Register must be read to determine the source.	R/W	0h	RESET
1	ETHREI This bit enables the Transmitter Holding Register Empty Interrupt when set to logic “1”.	R/W	0h	RESET
0	ERDAI This bit enables the Received Data Available Interrupt (and timeout interrupts in the FIFO mode) when set to logic “1”.	R/W	0h	RESET

16.10.6 FIFO CONTROL REGISTER

This is a write only register at the same location as the Interrupt Identification Register.

Note:	DMA is not supported.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	RECV_FIFO_TRIGGER_LEVEL These bits are used to set the trigger level for the RCVR FIFO interrupt.	W	0h	RESET
5:4	Reserved	RES	-	-
3	DMA_MODE_SELECT Writing to this bit has no effect on the operation of the UART. The RXRDY and TXRDY pins are not available on this chip.	W	0h	RESET
2	CLEAR_XMIT_FIFO Setting this bit to a logic “1” clears all bytes in the XMIT FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.	W	0h	RESET
1	CLEAR_RECv_FIFO Setting this bit to a logic “1” clears all bytes in the RCVR FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.	W	0h	RESET

© 2020 Microchip Technology Inc.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	EXRF Enable XMIT and RECV FIFO. Setting this bit to a logic “1” enables both the XMIT and RCVR FIFOs. Clearing this bit to a logic “0” disables both the XMIT and RCVR FIFOs and clears all bytes from both FIFOs. When changing from FIFO Mode to non-FIFO (16450) mode, data is automatically cleared from the FIFOs. This bit must be a 1 when other bits in this register are written to or they will not be properly programmed.	W	0h	RESET

TABLE 16-11: RECV FIFO TRIGGER LEVELS

Bit 7	Bit 6	RECV FIFO Trigger Level (BYTES)
0	0	1
	1	4
1	0	8
	1	14

16.10.7 INTERRUPT IDENTIFICATION REGISTER

By accessing this register, the host CPU can determine the highest priority interrupt and its source. Four levels of priority interrupt exist. They are in descending order of priority:

1. Receiver Line Status (highest priority)
2. Received Data Ready
3. Transmitter Holding Register Empty
4. MODEM Status (lowest priority)

Information indicating that a prioritized interrupt is pending and the source of that interrupt is stored in the Interrupt Identification Register (refer to Table 16-12). When the CPU accesses the IIR, the Serial Port freezes all interrupts and indicates the highest priority pending interrupt to the CPU. During this CPU access, even if the Serial Port records new interrupts, the current indication does not change until access is completed. The contents of the IIR are described below.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	FIFO_EN These two bits are set when the FIFO CONTROL Register bit 0 equals 1.	R	0h	RESET
5:4	Reserved	RES	-	-
3:1	INTID These bits identify the highest priority interrupt pending as indicated by Table 16-12, "Interrupt Control Table". In non-FIFO mode, Bit[3] is a logic “0”. In FIFO mode Bit[3] is set along with Bit[2] when a time-out interrupt is pending.	R	0h	RESET

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	IPEND This bit can be used in either a hardwired prioritized or polled environment to indicate whether an interrupt is pending. When bit 0 is a logic ‘0’ an interrupt is pending and the contents of the IIR may be used as a pointer to the appropriate internal service routine. When bit 0 is a logic ‘1’ no interrupt is pending.	R	1h	RESET

TABLE 16-12: INTERRUPT CONTROL TABLE

FIFO Mode Only	Interrupt Identification Register			Interrupt SET and RESET Functions			
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0	0	0	1	-	None	None	-
	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error or Break Interrupt	Reading the Line Status Register
		0		Second	Received Data Available	Receiver Data Available	Read Receiver Buffer or the FIFO drops below the trigger level.
1					Character Timeout Indication	No Characters Have Been Removed From or Input to the RCVR FIFO during the last 4 Char times and there is at least 1 char in it during this time	Reading the Receiver Buffer Register
0	0	1		Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if Source of Interrupt) or Writing the Transmitter Holding Register
	0	0		Fourth	MODEM Status	Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect	Reading the MODEM Status Register

© 2020 Microchip Technology Inc.

16.10.8 LINE CONTROL REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	DLAB Divisor Latch Access Bit (DLAB). It must be set high (logic “1”) to access the Divisor Latches of the Baud Rate Generator during read or write operations. It must be set low (logic “0”) to access the Receiver Buffer Register, the Transmitter Holding Register, or the Interrupt Enable Register.	R/W	0h	RESET
6	BREAK_CONTROL Set Break Control bit. When bit 6 is a logic “1”, the transmit data output (TXD) is forced to the Spacing or logic “0” state and remains there (until reset by a low level bit 6) regardless of other transmitter activity. This feature enables the Serial Port to alert a terminal in a communications system.	R/W	0h	RESET
5	STICK_PARITY Stick Parity bit. When parity is enabled it is used in conjunction with bit 4 to select Mark or Space Parity. When LCR bits 3, 4 and 5 are 1 the Parity bit is transmitted and checked as a 0 (Space Parity). If bits 3 and 5 are 1 and bit 4 is a 0, then the Parity bit is transmitted and checked as 1 (Mark Parity). If bit 5 is 0 Stick Parity is disabled. Bit 3 is a logic “1” and bit 5 is a logic “1”, the parity bit is transmitted and then detected by the receiver in the opposite state indicated by bit 4.	R/W	0h	RESET
4	PARITY_SELECT Even Parity Select bit. When bit 3 is a logic “1” and bit 4 is a logic “0”, an odd number of logic “1”'s is transmitted or checked in the data word bits and the parity bit. When bit 3 is a logic “1” and bit 4 is a logic “1” an even number of logic “1”s is transmitted and checked.	R/W	0h	RESET
3	ENABLE_PARITY Parity Enable bit. When bit 3 is a logic “1”, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and the first stop bit of the serial data. (The parity bit is used to generate an even or odd number of 1s when the data word bits and the parity bit are summed).	R/W	0h	RESET
2	STOP_BITS This bit specifies the number of stop bits in each transmitted or received serial character. Table 16-13 summarizes the information. The receiver will ignore all stop bits beyond the first, regardless of the number used in transmitting.	R/W	0h	RESET
1:0	WORD_LENGTH These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:	R/W	0h	RESET

TABLE 16-13: STOP BITS

Bit 2	Word Length	Number of Stop Bits
0	--	1
1	5 bits	1.5
	6 bits	2
	7 bits	
	8 bits	

TABLE 16-14: SERIAL CHARACTER

Bit 1	Bit 0	Word Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

The Start, Stop and Parity bits are not included in the word length.

16.10.9 MODEM CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:5	Reserved	RES	-	-
4	LOOPBACK This bit provides the loopback feature for diagnostic testing of the Serial Port. When bit 4 is set to logic “1”, the following occur: 1. The TXD is set to the Marking State (logic “1”). 2. The receiver Serial Input (RXD) is disconnected. 3. The output of the Transmitter Shift Register is “looped back” into the Receiver Shift Register input. 4. All MODEM Control inputs (CTS#, DSR#, RI# and DCD#) are disconnected. 5. The four MODEM Control outputs (DTR#, RTS#, OUT1 and OUT2) are internally connected to the four MODEM Control inputs (DSR#, CTS#, RI#, DCD#). 6. The Modem Control output pins are forced inactive high. 7. Data that is transmitted is immediately received. This feature allows the processor to verify the transmit and receive data paths of the Serial Port. In the diagnostic mode, the receiver and the transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.	R/W	0h	RESET

© 2020 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	OUT2 Output 2 (OUT2). This bit is used to enable an UART interrupt. When OUT2 is a logic “0”, the serial port interrupt output is forced to a high impedance state - disabled. When OUT2 is a logic “1”, the serial port interrupt outputs are enabled.	R/W	0h	RESET
2	OUT1 This bit controls the Output 1 (OUT1) bit. This bit does not have an output pin and can only be read or written by the CPU.	R/W	0h	RESET
1	RTS This bit controls the Request To Send (RTS#) output. When bit 1 is set to a logic “1”, the RTS# output is forced to a logic “0”. When bit 1 is set to a logic “0”, the RTS# output is forced to a logic “1”.	R/W	0h	RESET
0	DTR This bit controls the Data Terminal Ready (DTR#) output. When bit 0 is set to a logic “1”, the DTR# output is forced to a logic “0”. When bit 0 is a logic “0”, the DTR# output is forced to a logic “1”.	R/W	0h	RESET

16.10.10 LINE STATUS REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	FIFO_ERROR This bit is permanently set to logic “0” in the 450 mode. In the FIFO mode, this bit is set to a logic “1” when there is at least one parity error, framing error or break indication in the FIFO. This bit is cleared when the LSR is read if there are no subsequent errors in the FIFO.	R	0h	RESET
6	TRANSMIT_ERROR Transmitter Empty. Bit 6 is set to a logic “1” whenever the Transmitter Holding Register (THR) and Transmitter Shift Register (TSR) are both empty. It is reset to logic “0” whenever either the THR or TSR contains a data character. Bit 6 is a read only bit. In the FIFO mode this bit is set whenever the THR and TSR are both empty,	R	0h	RESET
5	TRANSMIT_EMPTY Transmitter Holding Register Empty Bit 5 indicates that the Serial Port is ready to accept a new character for transmission. In addition, this bit causes the Serial Port to issue an interrupt when the Transmitter Holding Register interrupt enable is set high. The THRE bit is set to a logic “1” when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic “0” whenever the CPU loads the Transmitter Holding Register. In the FIFO mode this bit is set when the XMIT FIFO is empty, it is cleared when at least 1 byte is written to the XMIT FIFO. Bit 5 is a read only bit.	R	0h	RESET

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	BREAK_INTERRUPT Break Interrupt. Bit 4 is set to a logic “1” whenever the received data input is held in the Spacing state (logic “0”) for longer than a full word transmission time (that is, the total time of the start bit + data bits + parity bits + stop bits). The BI is reset after the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. Restarting after a break is received, requires the serial data (RXD) to be logic “1” for at least 1/2 bit time. Bits 1 through 4 are the error conditions that produce a Receiver Line Status Interrupt BIT 3 whenever any of the corresponding conditions are detected and the interrupt is enabled	R	0h	RESET
3	FRAME_ERROR Framing Error. Bit 3 indicates that the received character did not have a valid stop bit. Bit 3 is set to a logic “1” whenever the stop bit following the last data bit or parity bit is detected as a zero bit (Spacing level). This bit is reset to a logic “0” whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. The Serial Port will try to resynchronize after a framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this 'start' bit twice and then takes in the 'data'.	R	0h	RESET
2	PARITY ERROR Parity Error. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. This bit is set to a logic “1” upon detection of a parity error and is reset to a logic “0” whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO.	R	0h	RESET
1	OVERRUN_ERROR Overrun Error. Bit 1 indicates that data in the Receiver Buffer Register was not read before the next character was transferred into the register, thereby destroying the previous character. In FIFO mode, an overrun error will occur only when the FIFO is full and the next character has been completely received in the shift register, the character in the shift register is overwritten but not transferred to the FIFO. This bit is set to a logic “1” immediately upon detection of an overrun condition, and reset whenever the Line Status Register is read.	R	0h	RESET
0	DATA_READY Data Ready. It is set to a logic ‘1’ whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to a logic ‘0’ by reading all of the data in the Receive Buffer Register or the FIFO.	R	0h	RESET

© 2020 Microchip Technology Inc.

16.10.11 MODEM STATUS REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	DCD This bit is the complement of the Data Carrier Detect (DCD#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to OUT2 in the MCR.	R	0h	RESET
6	RI This bit is the complement of the Ring Indicator (RI#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to OUT1 in the MCR.	R	0h	RESET
5	DSR This bit is the complement of the Data Set Ready (DSR#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to DTR# in the MCR.	R	0h	RESET
4	CTS This bit is the complement of the Clear To Send (CTS#) input. If bit 4 of the MCR is set to logic ‘1’, this bit is equivalent to RTS# in the MCR.	R	0h	RESET
3	DDCD Delta Data Carrier Detect (DDCD). Bit 3 indicates that the DCD# input to the chip has changed state. NOTE: Whenever bit 0, 1, 2, or 3 is set to a logic ‘1’, a MODEM Status Interrupt is generated.	R	0h	RESET
2	TERI Trailing Edge of Ring Indicator (TERI). Bit 2 indicates that the RI# input has changed from logic ‘0’ to logic ‘1’.	R	0h	RESET
1	DDSR Delta Data Set Ready (DDSR). Bit 1 indicates that the DSR# input has changed state since the last time the MSR was read.	R	0h	RESET
0	DCTS Delta Clear To Send (DCTS). Bit 0 indicates that the CTS# input to the chip has changed state since the last time the MSR was read.	R	0h	RESET

16.10.12 SCRATCHPAD REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SCRATCH This 8 bit read/write register has no effect on the operation of the Serial Port. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.	R/W	0h	RESET

16.11 Configuration Registers

Configuration Registers for an instance of the UART are listed in the following table. Host access to Configuration Registers is through the Configuration Port using the Logical Device Number of each instance of the UART and the Index shown in the “Host Index” column of the table. The EC can access Configuration Registers directly. The EC address for each register is formed by adding the Base Address for each instance of the UART shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “EC Offset” column.

TABLE 16-15: CONFIGURATION REGISTER SUMMARY

EC Offset	Host Index	Register Name
330h	30h	Activate Register
3F0h	F0h	Configuration Select Register

16.11.1 ACTIVATE REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	ACTIVATE When this bit is 1, the UART logical device is powered and functional. When this bit is 0, the UART logical device is powered down and inactive.	R/W	0b	RESET

16.11.2 CONFIGURATION SELECT REGISTER

Offset	F0h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	-	-
2	POLARITY 1=The UART_TX and UART_RX pins functions are inverted 0=The UART_TX and UART_RX pins functions are not inverted	R/W	0b	RESET
1	POWER 1=The RESET reset signal is derived from RESET_HOST 0=The RESET reset signal is derived from RESET_SYS	R/W	1b	RESET
0	CLK_SRC 1=The UART Baud Clock is derived from an external clock source 0=The UART Baud Clock is derived from one of the two internal clock sources	R/W	0b	RESET

© 2020 Microchip Technology Inc.

17.0 GPIO INTERFACE

17.1 General Description

The MEC152x GPIO Interface provides general purpose input monitoring and output control, as well as managing many aspects of pin functionality; including, multi-function Pin Multiplexing Control, GPIO Direction control, PU/PD (PU_PD) resistors, asynchronous wakeup and synchronous Interrupt Detection (int_det), GPIO Direction, and Polarity control, as well as control of pin drive strength and slew rate.

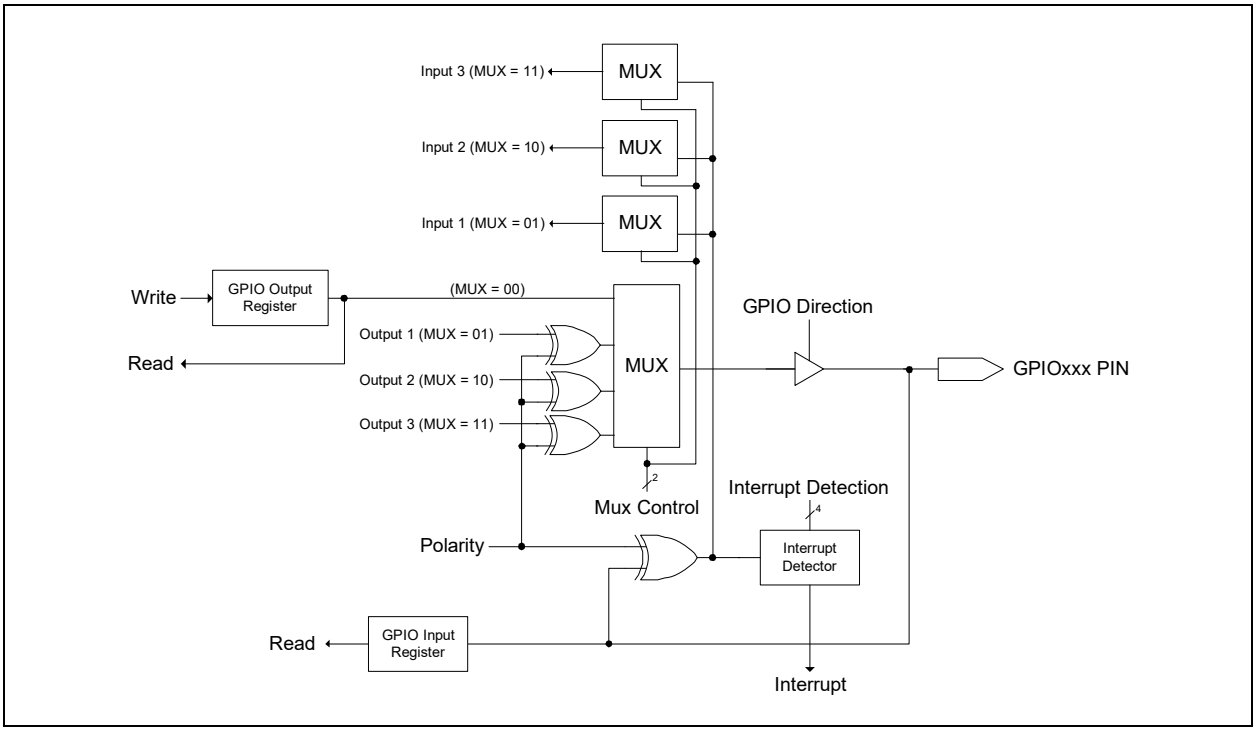
Features of the GPIO Interface include:

- Inputs:
  - Asynchronous rising and falling edge wakeup detection
  - Interrupt High or Low Level
- On Output:
  - Push Pull or Open Drain output
- Pull up or pull down resistor control
- Interrupt and wake capability available for all GPIOs
- Programmable pin drive strength and slew rate limiting
- Group- or individual control of GPIO data.
- Multiplexing of all multi-function pins are controlled by the GPIO interface

17.2 Block Diagram

The GPIO Interface Block Diagram shown in Figure 17-1 illustrates the functionality of a single MEC152x GPIO Interface pin. The source for the Pin Multiplexing Control, Interrupt Detection (int_det), GPIO Direction, and Polarity controls in Figure 17-1 is a Pin Control Register that is associated with each pin (see Section 17.6.1.1, "Pin Control Register," on page 260).

FIGURE 17-1: GPIO INTERFACE BLOCK DIAGRAM



17.3 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

17.3.1 POWER DOMAINS

Name	Description
VTR_CORE	The registers and logic in this block are powered by VTR_CORE.

17.3.2 CLOCK INPUTS

Name	Description
48MHz	The 48MHz is used for synchronizing the GPIO inputs.

17.3.3 RESETS

Name	Description
RESET_SYS	This reset is asserted when VTR_CORE is applied.
RESET_VCC	This is an alternate reset condition, typically asserted when the main power rail is asserted. This reset is used for VCC Power Well Emulation.

17.4 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
GPIO_Event	Each pin in the GPIO Interface has the ability to generate an interrupt event. This event may be used as a wake event. The GPIO Interface can generate an interrupt source event on a high level, low level, rising edge and falling edge, as configured by the Interrupt Detection (int_det) bits in the Pin Control Register associated with the GPIO signal function. **Note:** The minimum pulse width required to generate an interrupt/wakeup event is 5ns.

17.5 Description

The GPIO Interface refers to all the GPIOxxx pins implemented in the design. GPIO stands for General Purpose I/O.

The GPIO signals may be used by firmware to both monitor and control a pin in “bit-banged” mode. The GPIOs may be individually controlled via their Pin Control Register or group controlled via the Output and Input GPIO registers. The GPIO Output Control Select

The GPIO Pin control registers are used to select the alternate functions on GPIO pins (unless otherwise specified), to control the buffer direction, strength, and polarity, to control the internal pull-ups and pull-downs, for VCC emulation, and for selecting the event type that causes a GPIO interrupt.

The GPIO input is always live, even when an alternate function is selected. Firmware may read the GPIO input anytime to see the value on the pin. In addition, the GPIO interrupt is always functional, and may be used for either the GPIO itself or to support the alternate functions on the pin. See FIGURE 17-1: GPIO Interface Block Diagram on page 255.

© 2020 Microchip Technology Inc.

17.5.1 ACCESSING GPIOS

There are two ways to access GPIO output data. Bit [10] is used to determine which GPIO output data bit affects the GPIO output pin.

- Grouped Output GPIO Data
  - Outputs to individual GPIO ports are grouped into 32-bit GPIO Output Registers.
- Individual GPIO output data
  - Alternatively, each GPIO output port is individually accessible via Bit [16] in the port's Pin Control Register. On reads, Bit [16] returns the programmed value, not the value on the pin.

There are two ways to access GPIO input data.

- Input GPIO Data
  - Inputs from individual GPIO ports are grouped into 32-bit GPIO Input Registers and always reflect the current state of the GPIO input from the pad.
- GPIO input from pad
  - Alternatively, each GPIO input port is individually accessible via Bit [24] in the port's Pin Control Register. Bit [24] always reflects the current state of GPIO input from the pad.

17.5.2 GPIO INDEXING

Each GPIO signal function name consists of a 4-character prefix ("GPIO") followed by a 3-digit octal-encoded index number. In the MEC152x GPIO indexing is done sequentially starting from 'GPIO000.'

17.5.3 PIN CONTROL REGISTERS

Each GPIO has two Pin Control registers. The Pin Control Register, which is the primary register, is used to read the value of the input data and set the output either high or low. It is used to select the alternate function via the Mux Control bits, set the Polarity of the input, configure and enable the output buffer, configure the GPIO interrupt event source, enable internal pull-up/pull-down resistors, and to enable VCC Emulation via the Power Gating Signals (PGS) control bits. The Pin Control Register 2 is used to configure the output buffer drive strength and slew rate.

The following tables define the default settings for the two Pin Control registers for each GPIO in each product group.

17.5.3.1 Pin Control Register Defaults

Please refer to Section 3.5, "GPIO Register Assignments" for the Pin Control Register default information.

17.6 GPIO Registers

The registers listed in the Register Summary table are for a single instance of the MEC152x. The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in the Register Base Address Table.

TABLE 17-1: REGISTER BASE ADDRESS TABLE

Instance Name	Instance Number	Host	Address Space	Base Address
GPIO	0	eSPI	I/O	Note 17-2
	0	EC	32-bit internal address space	4008_1000h Note 17-1

Note 17-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

Note 17-2 The GPIO registers may be accessed by the eSPI Host via the EMI block via GPIO commands or by direct access if enabled by firmware. See the firmware documentation for a description of this access method.

Note: Registers and bits associated with GPIOs not implemented are Reserved. Please refer to Section 2.3, "Pin List" for GPIOs implemented in the chip.

TABLE 17-2: REGISTER SUMMARY

Offset	Register Name
000h - 01Ch	GPIO000-GPIO007 Pin Control Register
020h - 03Ch	GPIO010-GPIO017 Pin Control Register
040h - 05Ch	GPIO020-GPIO027 Pin Control Register
060h - 078h	GPIO030-GPIO036 Pin Control Register
080h - 09Ch	GPIO040-GPIO047 Pin Control Register
0A0h - 0BCh	GPIO050-GPIO057 Pin Control Register
0C0h - 0DCh	GPIO060-GPIO067 Pin Control Register
0E0h - 0F8h	GPIO070-GPIO077 Pin Control Register
100h - 11Ch	GPIO100-GPIO107 Pin Control Register
128h - 13Ch	GPIO112-GPIO117 Pin Control Register
140h - 15Ch	GPIO120-GPIO127 Pin Control Register
160h - 16Ch	GPIO130-GPIO137 Pin Control Register
180h - 19Ch	GPIO140-GPIO147 Pin Control Register
1A0h - 1BCh	GPIO150-GPIO157 Pin Control Register
1C0h - 1DCh	GPIO160-GPIO167 Pin Control Register
1E0h - 1F4h	GPIO170-GPIO177 Pin Control Register
200h - 21Ch	GPIO200-GPIO207 Pin Control Register
220h - 23Ch	GPIO210-GPIO217 Pin Control Register
240h - 25Ch	GPIO221-GPIO227 Pin Control Register
260h - 27Ch	Reserved
280h - 298h	GPIO240-GPIO247 Pin Control Register
2ACh - 2BCh	GPIO253-GPIO257 Pin Control Register
2C0h	GPIO260 Pin Control Register
300h	Input GPIO[000:036]
304h	Input GPIO[040:076]
308h	Input GPIO[100:127]

© 2020 Microchip Technology Inc.

TABLE 17-2: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
30Ch	Input GPIO[140:176]
310h	Input GPIO[200:236]
314h	Input GPIO[240:276]
380h	Output GPIO[000:036]
384h	Output GPIO[040:076]
388h	Output GPIO[100:127]
38Ch	Output GPIO[140:176]
390h	Output GPIO[200:236]
394h	Output GPIO[240:276]
500h - 51Ch	GPIO000-GPIO007 Pin Control Register 2
520h - 53Ch	GPIO010-GPIO017 Pin Control Register 2
540h - 55Ch	GPIO020-GPIO027 Pin Control Register 2
560h - 578h	GPIO030-GPIO036 Pin Control Register 2
580h - 59Ch	GPIO040-GPIO047 Pin Control Register 2
5A0h - 5BCh	GPIO050-GPIO057 Pin Control Register 2
5C0h - 5DCh	GPIO060-GPIO067 Pin Control Register 2
5E0h - 5F8h	GPIO070-GPIO076 Pin Control Register 2
600h - 61Ch	GPIO100-GPIO107 Pin Control Register 2
620h - 63Ch	GPIO110-GPIO117 Pin Control Register 2
640h - 65Ch	GPIO120-GPIO127 Pin Control Register 2
660h - 674h	GPIO130-GPIO135 Pin Control Register 2
680h - 69Ch	GPIO140-GPIO147 Pin Control Register 2
6A0h - 6BCh	GPIO150-GPIO157 Pin Control Register 2
6C0h - 6D8h	GPIO160-GPIO167 Pin Control Register 2
6E0h - 6F4h	GPIO170-GPIO175 Pin Control Register 2
700h - 71Ch	GPIO200-GPIO207 Pin Control Register 2
720h - 73Ch	GPIO210-GPIO217 Pin Control Register 2
740h - 75Ch	GPIO220-GPIO227 Pin Control Register 2
760h - 778h	Reserved
780h - 79Ch	GPIO240-GPIO247 Pin Control Register 2
7A0h - 7BCh	GPIO250-GPIO257 Pin Control Register 2
7C0h	GPIO260 Pin Control Register 2

© 2020 Microchip Technology Inc.

17.6.1 PIN CONTROL REGISTERS

Two Pin Control Registers are implemented for each GPIO. The Pin Control Register format is described in Section 17.6.1.1, "Pin Control Register," on page 260. The Pin Control Register 2 format is described in Section 17.6.1.2, "Pin Control Register 2," on page 263. Pin Control Register address offsets and defaults for each product are defined in Section 17.5.3.1, "Pin Control Register Defaults," on page 257.

17.6.1.1 Pin Control Register

Offset	See Table 17-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	RESERVED	RES	-	-
24	GPIO input from pad On reads, Bit [24] reflects the state of GPIO input from the pad regardless of setting of Bit [10]. **Note:** This bit is forced high when the selected power well is off as selected by the Power Gating Signal bits. See bits[3:2].	R	Note 17-1	RESET_SYS
23:17	RESERVED	RES	-	-
16	GPIO output data If enabled by the GPIO Output Control Select bit, the GPIO output data bit determines the level on the GPIO pin when the pin is configured for the GPIO output function. On writes: If enabled via the GPIO Output Control Select 0: GPIO[x] out = '0' 1: GPIO[x] out = '1' **Note:** If disabled via the GPIO Output Control Select then the GPIO[x] out pin is unaffected by writing this bit. On reads: Bit [16] returns the last programmed value, not the value on the pin.	R/W (GPIO Output Control Select = 0) R (GPIO Output Control Select=1)	Note 17-1	RESET_SYS
15	GPIO input disable This bit can be used to support undervoltage functionality. 1=disable input 0=do not disable input	R/W	Note 17-1	RESET_SYS
14	RESERVED	RES	-	-
13:12	Mux Control The Mux Control field determines the active signal function for a pin. 00 = GPIO Function Selected 01 = Signal Function 1 Selected 10 = Signal Function 2 Selected 11 = Signal Function 3 Selected	R/W	Note 17-1	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	See Table 17-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
11	Polarity 0 = Non-inverted 1 = Inverted When the Polarity bit is set to '1' and the Mux Control bits are greater than '00,' the selected signal function outputs are inverted and Interrupt Detection (int_det) sense defined in Table 17-3, "Edge Enable and Interrupt Detection Bits Definition" is inverted. When the Mux Control field selects the GPIO signal function (Mux = '00'), the Polarity bit does not effect the output. Regardless of the state of the Mux Control field and the Polarity bit, the state of the pin is always reported without inversion in the GPIO input register.	R/W	Note 17-1	RESET_SYS
10	GPIO Output Control Select Every GPIO has two mechanisms to set a GPIO data output: Output GPIO Bit located in the grouped GPIO Output Registers and the single GPIO output data bit located in bit 16 of this register. This control bit determines the source of the GPIO output. 0 = Pin Control Bit[16] GPIO output data bit enabled When this bit is zero the single GPIO output data bit is enabled. (GPIO output data is R/W capable and the Grouped Output GPIO is disabled (i.e., Read-Only). 1 = Grouped Output GPIO enable When this bit is one the GPIO output data write is disabled (i.e., Read-Only) and the Grouped Output GPIO is enabled (i.e., R/W). **Note:** See description in Section 17.5.1, "Accessing GPIOs".	R/W	Note 17-1	RESET_SYS
9	GPIO Direction 0 = Input 1 = Output The GPIO Direction bit controls the buffer direction only when the Mux Control field is '00' selecting the pin signal function to be GPIO. When the Mux Control field is greater than '00' (i.e., a non-GPIO signal function is selected) the GPIO Direction bit has no affect and the selected signal function logic directly controls the pin direction.	R/W	Note 17-1	RESET_SYS
8	Output Buffer Type 0 = Push-Pull 1 = Open Drain **Note:** Unless explicitly stated otherwise, pins with (I/O/OD) or (O/OD) in their buffer type column in the tables in are compliant with the following Programmable OD/PP Multiplexing Design Rule: Each compliant pin has a programmable open drain/push-pull buffer controlled by the Output Buffer Type bit in the associated Pin Control Register. The state of this bit controls the mode of the interface buffer for all selected functions, including the GPIO function.	R/W	Note 17-1	RESET_SYS

Offset	See Table 17-2, "Register Summary"			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Edge Enable (edge_en) 0 = Edge detection disabled 1 = Edge detection enabled **Note:** See Table 17-3, "Edge Enable and Interrupt Detection Bits Definition".	R/W	Note 17-1	RESET_SYS
6:4	Interrupt Detection (int_det) The interrupt detection bits determine the event that generates a GPIO_Event. **Note:** See Table 17-3, "Edge Enable and Interrupt Detection Bits Definition". **Note:** Since the GPIO input is always available, even when the GPIO is not selected as the alternate function, the GPIO interrupts may be used for detecting pin activity on alternate functions. The only exception to this is the analog functions (e.g., ADC, Comparator inputs)	R/W	Note 17-1	RESET_SYS
3:2	Power Gating Signals (PGS) The Power Gating Signals provide the chip Power Emulation options. The pin will be tristated when the selected power well is off (i.e., gated) as indicated. The Emulated Power Well column defined in Pin Multiplexing indicates the emulation options supported for each signal. The Signal Power Well column defines the buffer power supply per function. **Note:** Note that all GPIOs support Power Gating unless otherwise noted. 00 = VTR The output buffer is tristated when VTR_PWRGD = 0. 01 = VCC The output buffer is tristate when VCC_PWRGD = 0. 10 = Unpowered. The always unpowered setting on a GPIO will force the pin to tristate. The input and output are disabled, and the pad is in the lowest power state. 11 = Reserved **Note:** VBAT Powered Signals are always powered by the VBAT rail and power well emulation does not apply. For VBAT powered signals this field should be set to 00.	R/W	Note 17-1	RESET_SYS
1:0	PU/PD (PU_PD) These bits are used to enable an internal pull-up or pull-down resistor device on the pin. 00 = None. Pin tristates when no active driver is present on the pin. 01 = Pull Up Enabled 10 = Pull Down Enabled 11 = Repeater mode. Pin is kept at previous voltage level when no active driver is present on the pin.	R/W	Note 17-1	RESET_SYS

Note 17-1 See Section 3.5, "GPIO Register Assignments" for the default values and Table 17-2, "Register Summary" and Table 3.6, "Register Map" for register offset value for each GPIO Pin Control Register.

Note 17-2 Repeater mode is not available on over voltage protected pins.

© 2020 Microchip Technology Inc.

TABLE 17-3: EDGE ENABLE AND INTERRUPT DETECTION BITS DEFINITION

Edge Enable	Interrupt Detection Bits			Selected Function
D7	D6	D5	D4	
0	0	0	0	Low Level Sensitive
0	0	0	1	High Level Sensitive
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Interrupt events are disabled
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	1	0	1	Rising Edge Triggered
1	1	1	0	Falling Edge Triggered
1	1	1	1	Either Edge Triggered

Note: Only edge triggered interrupts can wake up the main ring oscillator. The GPIO must be enabled for edge-triggered interrupts and the GPIO interrupt must be enabled in the interrupt aggregator in order to wake up the ring when the ring is shut down.

APPLICATION NOTE:

1. All GPIO interrupt detection configurations default to '0000', which is low level interrupt. Having interrupt detection enabled will un-gated the clock to the GPIO module whenever the interrupt is active, which increases power consumption. Interrupt detection should be disabled when not required to save power.
2. Changing the configuration of the Interrupt edge and detection bits may generate an interrupt if it is enabled. The GPIO should be configured and associated status bits should be cleared before enabling the Interrupt.

17.6.1.2 Pin Control Register 2

Offset	See Note 17-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	-	-
5:4	DRIVE_STRENGTH These bits are used to select the drive strength on the pin. 00 = 2mA 01 = 4mA 10 = 8mA 11 = 12mA	R/W	00	RESET_SYS
3:1	Reserved	RES	-	-

Offset	See Note 17-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	Slew Rate This bit is used to select the slew rate on the pin 1= fast (Should be set to 1 always) 0= slow	R/W	0h	RESET_SYS

17.6.2 GPIO OUTPUT REGISTERS

If enabled by the GPIO Output Control Select bit, the grouped GPIO Output bits determine the level on the GPIO pin when the pin is configured for the GPIO output function.

On writes:

If enabled via the GPIO Output Control Select

0: GPIO[x] out = '0'

1: GPIO[x] out = '1'

If disabled via the GPIO Output Control Select then the GPIO[x] out pin is unaffected by writing the corresponding GPIO bit in the grouped Output GPIO[xxx:yyy] register.

On reads:

The GPIO output bit in the grouped Output GPIO[xxx:yyy] register returns the last programmed value, not the value on the pin.

17.6.2.1 Output GPIO[000:036]

Offset	380h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[036:030] Output	R/W	00h	RESET_SYS
23:16	GPIO[027:020] Output	R/W	00h	RESET_SYS
15:8	GPIO[017:010] Output	R/W	00h	RESET_SYS
7:0	GPIO[007:000] Output	R/W	00h	RESET_SYS

© 2020 Microchip Technology Inc.

17.6.2.2 Output GPIO[040:076]

Offset	384h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	RESERVED	RES	-	-
30:24	GPIO[076:070] Output	R/W	00h	RESET_SYS
23:16	GPIO[067:060] Output	R/W	00h	RESET_SYS
15:8	GPIO[057:050] Output	R/W	00h	RESET_SYS
7:0	GPIO[047:040] Output	R/W	00h	RESET_SYS

17.6.2.3 Output GPIO[100:127]

Offset	388h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[136:130] Output	R/W	00h	RESET_SYS
23:16	GPIO[127:120] Output	R/W	00h	RESET_SYS
15:8	GPIO[117:110] Output	R/W	00h	RESET_SYS
7:0	GPIO[107:100] Output	R/W	00h	RESET_SYS

17.6.2.4 Output GPIO[140:176]

Offset	38Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[176:170] Output	R/W	00h	RESET_SYS
23:16	GPIO[167:160] Output	R/W	00h	RESET_SYS
15:8	GPIO[157:150] Output	R/W	00h	RESET_SYS
7:0	GPIO[147:140] Output	R/W	00h	RESET_SYS

© 2020 Microchip Technology Inc.

17.6.2.5 Output GPIO[200:236]

Offset	390h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[236:230] Output	R/W	00h	RESET_SYS
23:16	GPIO[227:220] Output	R/W	00h	RESET_SYS
15:8	GPIO[217:210] Output	R/W	00h	RESET_SYS
7:0	GPIO[207:200] Output	R/W	00h	RESET_SYS

17.6.2.6 Output GPIO[240:276]

Offset	394h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	RESERVED	RES	-	-
30:24	GPIO[276:270] Output	R/W	00h	RESET_SYS
23:16	GPIO[267:260] Output	R/W	00h	RESET_SYS
15:8	GPIO[257:250] Output	R/W	00h	RESET_SYS
7:0	GPIO[247:240] Output	R/W	00h	RESET_SYS

17.6.3 GPIO INPUT REGISTERS

The GPIO Input Registers can always be used to read the state of a pin, even when the pin is in an output mode and/or when a signal function other than the GPIO signal function is selected; i.e., the Pin Control Register Mux Control bits are not equal to '00.'

The MSbit of the Input GPIO registers have been implemented as a read/write scratch pad bit to support processor specific instructions.

Note: Bits associated with GPIOs that are not implemented are shown as Reserved.

© 2020 Microchip Technology Inc.

17.6.3.1 Input GPIO[000:036]

Offset	300h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[036:030] Input	R	00h	RESET_SYS
23:16	GPIO[027:020] Input	R	00h	RESET_SYS
15:8	GPIO[017:010] Input	R	00h	RESET_SYS
7:0	GPIO[007:000] Input	R	00h	RESET_SYS

17.6.3.2 Input GPIO[040:076]

Offset	304h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[076:070] Input	R	00h	RESET_SYS
23:16	GPIO[067:060] Input	R	00h	RESET_SYS
15:8	GPIO[057:050] Input	R	00h	RESET_SYS
7:0	GPIO[047:040] Input	R	00h	RESET_SYS

17.6.3.3 Input GPIO[100:127]

Offset	308h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[136:130] Input	R	00h	RESET_SYS
23:16	GPIO[127:120] Input	R	00h	RESET_SYS
15:8	GPIO[117:110] Input	R	00h	RESET_SYS
7:0	GPIO[107:100] Input	R	00h	RESET_SYS

© 2020 Microchip Technology Inc.

17.6.3.4 Input GPIO[140:176]

Offset	30Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:16	GPIO[176:160] Input	R	00h	RESET_SYS
15:8	GPIO[157:150] Input	R	00h	RESET_SYS
7:0	GPIO[147:140] Input	R	00h	RESET_SYS

17.6.3.5 Input GPIO[200:236]

Offset	310h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[236:230] Input	R	00h	RESET_SYS
23:16	GPIO[227:220] Input	R	00h	RESET_SYS
15:8	GPIO[217:210] Input	R	00h	RESET_SYS
7:0	GPIO[207:200] Input	R	00h	RESET_SYS

17.6.3.6 Input GPIO[240:276]

Offset	314h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Scratchpad Bit	R/W	0b	RESET_SYS
30:24	GPIO[276:270] Input	R	00h	RESET_SYS
23:16	GPIO[267:260] Input	R	00h	RESET_SYS
15:8	GPIO[257:250] Input	R	00h	RESET_SYS
7:0	GPIO[247:240] Input	R	00h	RESET_SYS

© 2020 Microchip Technology Inc.

18.0 WATCHDOG TIMER (WDT)

18.1 Introduction

The function of the Watchdog Timer is to provide a mechanism to detect if the internal embedded controller has failed.

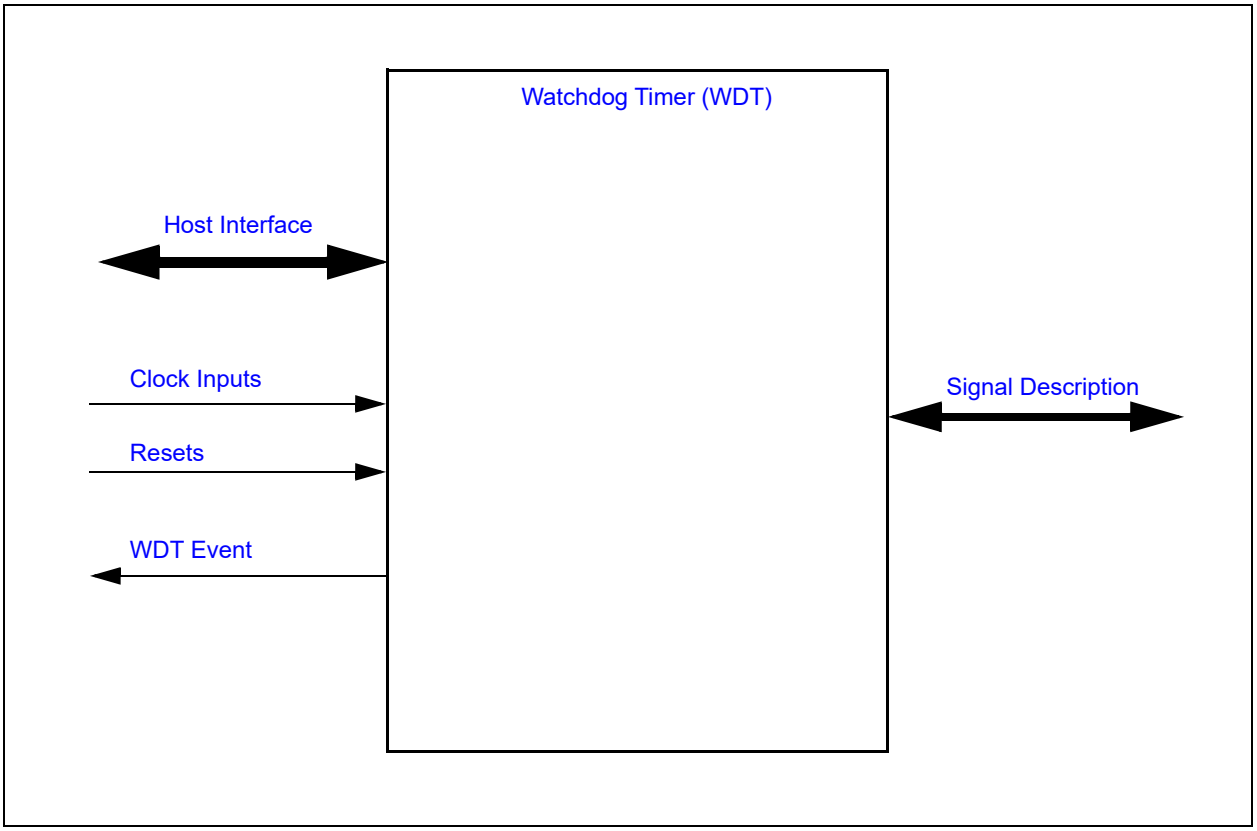
When enabled, the Watchdog Timer (WDT) circuit will generate a WDT Event if the user program fails to reload the WDT within a specified length of time known as the WDT Interval.

18.2 Interface

This block is designed to be accessed internally via a registered host interface

18.3 Host Interface

FIGURE 18-1: I/O DIAGRAM OF BLOCK



The registers defined for the Watchdog Timer (WDT) are accessible by the embedded controller as indicated in Section 18.7, "EC Registers". All registers accesses are synchronized to the host clock and complete immediately. Register reads/writes are not delayed by the 32KHz.

18.4 Signal Description

18.4.1 SIGNAL INTERFACE

There are no external signals for this block.

18.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

18.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block reside on this single power well.

18.5.2 CLOCK INPUTS

Name	Description
32KHz	The 32KHz clock input is the clock source to the Watchdog Timer functional logic, including the counter.

18.5.3 RESETS

TABLE 18-1: RESET INPUTS

Name	Description
RESET_SYS	Power on Reset to the block. This signal resets all the register and logic in this block to its default state following a POR or a WDT Event event.
RESET_SYS_nWDT	This reset signal is used on WDT registers/bits that need to be preserved through a WDT Event.

TABLE 18-2: RESET OUTPUTS

Source	Description
WDT Event	Pulse generated when WDT expires. This signal is used to either generate interrupt WDT_INT, if WDT Control Register bit 9 is set to 1b (WDT_INT_ENABLE), or reset the embedded controller and its subsystem, if WDT Control Register bit 9 is set to 0b. The event is cleared after a RESET_SYS.

18.6 Description

18.6.1 WDT OPERATION

18.6.1.1 WDT Activation Mechanism

The WDT is activated by the following sequence of operations during normal operation:

1. Load the WDT Load Register with the count value.
2. Set the WDT_ENABLE bit in the WDT Control Register.

The WDT Activation Mechanism starts the WDT decrementing counter.

18.6.1.2 WDT Deactivation Mechanism

The WDT is deactivated by the clearing the WDT_ENABLE bit in the WDT Control Register. The WDT Deactivation Mechanism places the WDT in a low power state in which clock are gated and the counter stops decrementing.

© 2020 Microchip Technology Inc.

18.6.1.3 WDT Reload Mechanism

The WDT must be reloaded within periods that are shorter than the programmed watchdog interval; otherwise, the WDT will underflow and a WDT Event will be generated and the WDT bit in Power-Fail and Reset Status Register on page 506 will be set. It is the responsibility of the user program to continually execute code which reloads the watchdog timer, causing the counter to be reloaded.

There are three methods of reloading the WDT: a write to the WDT Load Register, a write to the WDT Kick Register, or WDT event.

18.6.1.4 WDT Interval

The WDT Interval is the time it takes for the WDT to decrements from the WDT Load Register value to 0000h. The WDT Count Register value takes 33/32KHz seconds (ex. 33/32.768 KHz = 1.007ms) to decrement by 1 count.

18.6.1.5 WDT STALL Operation

There are three STALL_ENABLE control bits in the WDT Control Register. If enabled, and the STALL event is asserted, the WDT stops decrementing, and the WDT enters a low power state. When a WDT STALL event is de-asserted, the counter continues decrementing from the value it had when the STALL was asserted.

18.7 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Watchdog Timer (WDT) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 18-3: REGISTER SUMMARY

Offset	Register Name
00h	WDT Load Register
04h	WDT Control Register
08h	WDT Kick Register
0Ch	WDT Count Register
10h	WDT Status Register
14h	WDT Int Enable Register

18.7.1 WDT LOAD REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	WDT_LOAD Writing this field reloads the Watch Dog Timer counter.	R/W	FFFFh	RESET_SYS

18.7.2 WDT CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	RES	-	-
9	WDT_RESET If the WDT_RESET bit is set and the watch dog timer expires, the Watch dog module will generate interrupt and the WDT_RESET bit will be cleared. If this bit is not set, when the watch dog timer expires EC and its sub-system is reset.	R/W	0b	RESET_SYS
8:5	Reserved	RES	-	-
4	JTAG_STALL This bit enables the WDT Stall function if JTAG or SWD debug functions are active 1=The WDT is stalled while either JTAG or SWD is active 0=The WDT is not affected by the JTAG debug interface	R/W	0b	RESET_SYS
3	WEEK_TIMER_STALL This bit enables the WDT Stall function if the Week Timer is active. 1=The WDT is stalled while the Week Timer is active 0=The WDT is not affected by the Week Timer	R/W	0b	RESET_SYS
2	HIBERNATION_TIMER_STALL This bit enables the WDT Stall function if the Hibernation Timer 0 or Hibernation Timer 1 is active. 1=The WDT is stalled while the Hibernation Timer 0 is active 0=The WDT is not affected by Hibernation Timer 0	R/W	0b	RESET_SYS
1	TEST	R	0b	RESET_SYS
0	WDT_ENABLE In WDT Operation, the WDT is activated by the sequence of operations defined in Section 18.6.1.1, "WDT Activation Mechanism" and deactivated by the sequence of operations defined in Section 18.6.1.2, "WDT Deactivation Mechanism". 1=block enabled 0=block disabled	R/W	0b	RESET_SYS

18.7.3 WDT KICK REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	KICK The WDT Kick Register is a strobe. Reads of this register return 0. Writes to this register cause the WDT to reload the WDT Load Register value and start decrementing when the WDT_ENABLE bit in the WDT Control Register is set to '1'. When the WDT_ENABLE bit in the WDT Control Register is cleared to '0', writes to the WDT Kick Register have no effect.	W	n/a	RESET_SYS

© 2020 Microchip Technology Inc.

18.7.4 WDT COUNT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	WDT_COUNT This read-only register provide the current WDT count.	R	FFFFh	RESET_SYS

18.7.5 WDT STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	WDT_EVENT_IRQ This bit indicates the status of interrupt from Watch dog module.	R/W1C	0h	RESET_SYS_n-WDT

18.7.6 WDT INT ENABLE REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	WDT_INT_ENABLE This is the interrupt enables bit for WDT_INT interrupt. 1b - WDT_INT Interrupt Enable 0b - WDT_INT Interrupt Disabled	R/W	0h	RESET_SYS_n-WDT

19.0 16/32 BIT BASIC TIMER

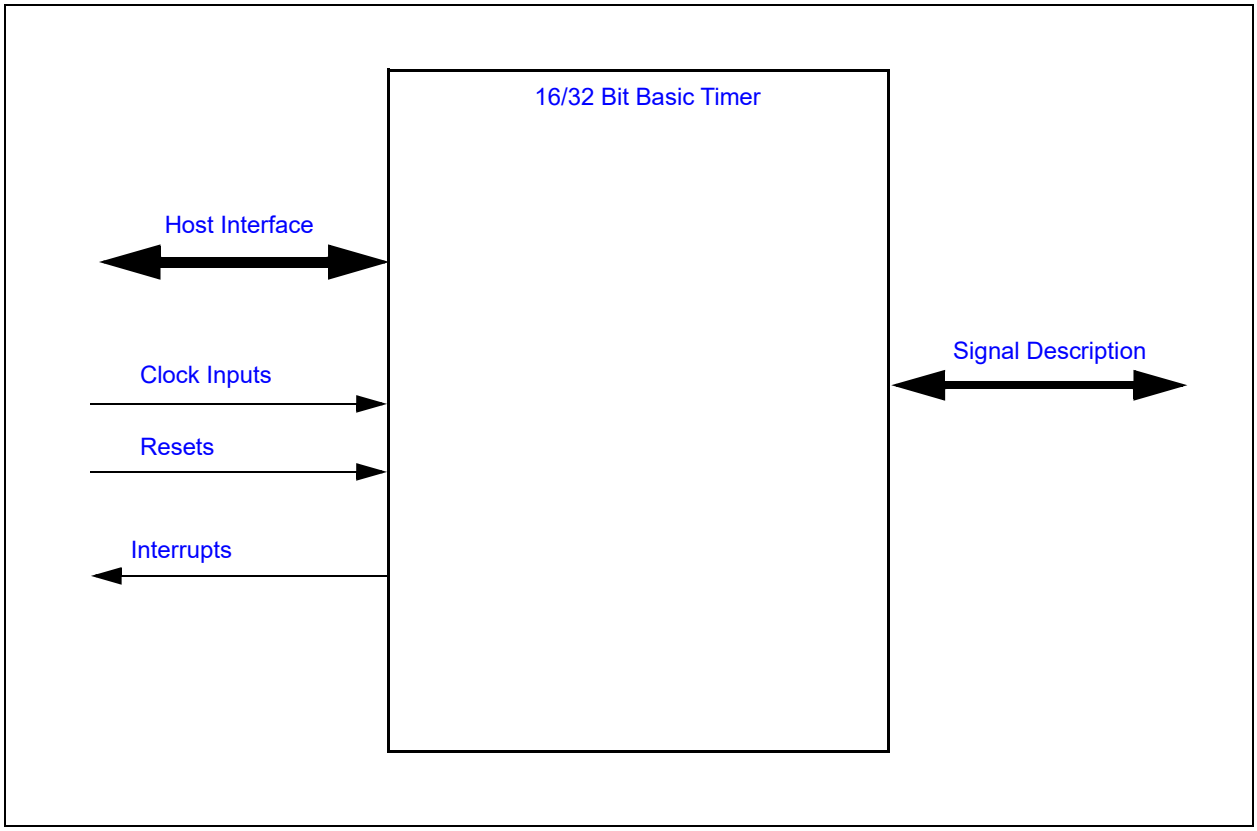
19.1 Introduction

This timer block offers a simple mechanism for firmware to maintain a time base. This timer may be instantiated as 16 bits or 32 bits. The name of the timer instance indicates the size of the timer.

19.2 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 19-1: I/O DIAGRAM OF BLOCK



19.3 Signal Description

There are no external signals for this block.

19.4 Host Interface

The Embedded Controller (EC) may access this block via the registers defined in Section 19.9, "EC-Only Registers," on page 277.

19.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

© 2020 Microchip Technology Inc.

19.5.1 POWER DOMAINS

TABLE 19-1: POWER SOURCES

Name	Description
VTR_CORE	The timer control logic and registers are all implemented on this single power domain.

19.5.2 CLOCK INPUTS

TABLE 19-2: CLOCK INPUTS

Name	Description
48MHz	This is the clock source to the timer logic. The Pre-scaler may be used to adjust the minimum resolution per bit of the counter.

19.5.3 RESETS

TABLE 19-3: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.
SOFT_RESET	This reset signal, which is created by this block, resets all the logic and registers to their initial default state. This reset is generated by the block when the SOFT_RESET bit is set in the Timer Control Register register.
Timer_Reset	This reset signal, which is created by this block, is asserted when either the RESET_SYS or the SOFT_RESET signal is asserted. The RESET_SYS and SOFT_RESET signals are OR'd together to create this signal.

19.6 Interrupts

TABLE 19-4: EC INTERRUPTS

Source	Description
TIMER_16_*x*	This interrupt event fires when a 16-bit timer *x* reaches its limit. This event is sourced by the EVENT_INTERRUPT status bit if enabled.
TIMER_32_*x*	This interrupt event fires when a 32-bit timer *x* reaches its limit. This event is sourced by the EVENT_INTERRUPT status bit if enabled.
Note: x represents the instance number.	

19.7 Low Power Modes

The Basic Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active.

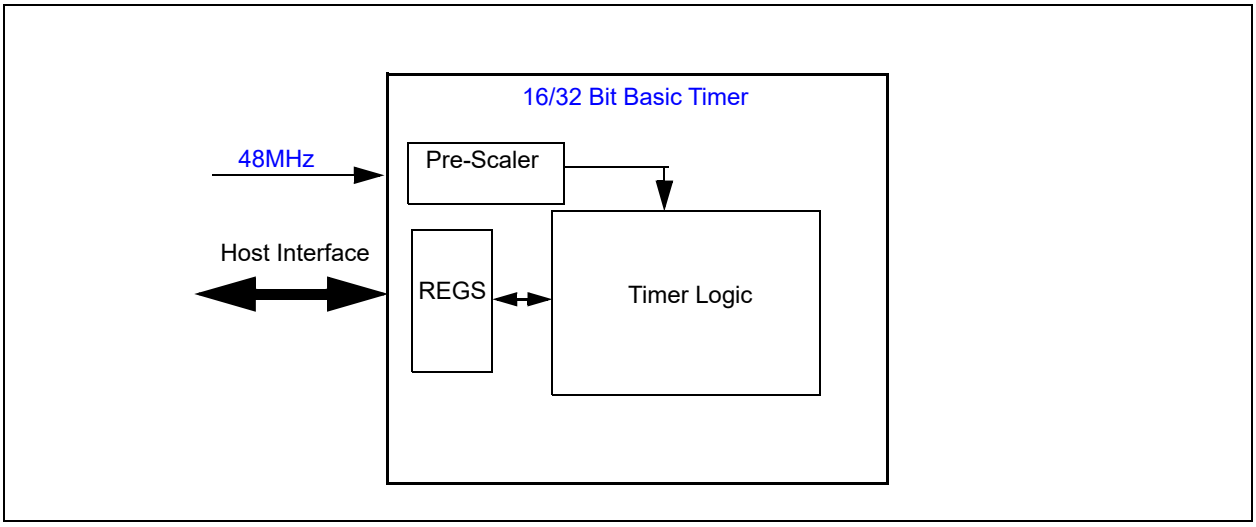
The sleep state of this timer is as follows:

- Asleep while the block is not Enabled
- Asleep while the block is not running (start inactive).
- Asleep while the block is halted (even if running).

The block is active while start is active.

19.8 Description

FIGURE 19-2: BLOCK DIAGRAM



This timer block offers a simple mechanism for firmware to maintain a time base in the design. The timer may be enabled to execute the following features:

- Programmable resolution per LSB of the counter via the Pre-scale bits in the Timer Control Register
- Programmable as either an up or down counter
- One-shot or Continuous Modes
- In one-shot mode the Auto Restart feature stops the counter when it reaches its limit and generates a level event.
- In Continuous Mode the Auto Restart feature restarts that counter from the programmed preload value and generates a pulse event.
- Counter may be reloaded, halted, or started via the Timer Control register
- Block may be reset by either a Power On Reset (POR) or via a Soft Reset.

© 2020 Microchip Technology Inc.

19.9 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Basic Timer. The addresses of each register listed in this table are defined as a relative offset to the “Base Address” of that instance, defined in the Device Inventory chapter and will follow the instance naming as listed in **TABLE 19-5: “MEC152x Instance Naming Convention”**.

TABLE 19-5: MEC152X INSTANCE NAMING CONVENTION

Block Instance	Host
16-Bit Basic Timer x	EC
32-Bit Basic Timer x	EC
Note: x represents the instance number.	

TABLE 19-6: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Timer Count Register
04h	Timer Preload Register
08h	Timer Status Register
0Ch	Timer Int Enable Register
10h	Timer Control Register

19.9.1 TIMER COUNT REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER This is the value of the Timer counter. This is updated by Hardware but may be set by Firmware. If it is set while the Hardware Timer is operating, functionality can not be guaranteed. When read, it is buffered so single byte reads will be able to catch the full 4 byte register without it changing. - For 16 bit Basic Timer, bits 0 to 15 are r/w counter bits. Bits 31 down to 16 are reserved. Reads of bits 31 down to 16 return 0 and writes have no effect. - For 32 bit Basic Timer, bits 0 to 31 are r/w counter bits.	R/W	0h	Timer_Reset

19.9.2 TIMER PRELOAD REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PRE_LOAD This is the value of the Timer pre-load for the counter. This is used by H/W when the counter is to be restarted automatically; this will become the new value of the counter upon restart. The size of the Pre-Load value is the same as the size of the counter. - For 16 bit Basic Timer, bits 0 to 15 are r/w pre-load bits. Bits 31 down to 16 are reserved. Reads of bits 31 down to 16 return 0 and writes have no effect. - For 32 bit Basic Timer, bits 0 to 31 are r/w pre-load bits.	R/W	0h	Timer_Reset

19.9.3 TIMER STATUS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	RES	-	-
0	EVENT_INTERRUPT This is the interrupt status that fires when the timer reaches its limit. This may be level or a self clearing signal cycle pulse, based on the AUTO_RESTART bit in the Timer Control Register. If the timer is set to automatically restart, it will provide a pulse, otherwise a level is provided.	R/WC	0h	Timer_Reset

19.9.4 TIMER INT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Reserved	RES	-	-
0	EVENT_INTERRUPT_ENABLE This is the interrupt enable for the status EVENT_INTERRUPT bit in the Timer Status Register	R/W	0h	Timer_Reset

© 2020 Microchip Technology Inc.

19.9.5 TIMER CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	PRE_SCALE This is used to divide down the system clock through clock enables to lower the power consumption of the block and allow slow timers. Updating this value during operation may result in erroneous clock enable pulses until the clock divider restarts. The number of clocks per clock enable pulse is (Value + 1); a setting of 0 runs at the full clock speed, while a setting of 1 runs at half speed.	R/W	0h	Timer_Reset
15:8	Reserved	RES	-	-
7	HALT This is a halt bit. This will halt the timer as long as it is active. Once the halt is inactive, the timer will start from where it left off. 1=Timer is halted. It stops counting. The clock divider will also be reset. 0=Timer runs normally	R/W	0h	Timer_Reset
6	RELOAD This bit reloads the counter without interrupting it operation. This will not function if the timer has already completed (when the START bit in this register is '0'). This is used to periodically prevent the timer from firing when an event occurs. Usage while the timer is off may result in erroneous behavior.	R/W	0h	Timer_Reset
5	START This bit triggers the timer counter. The counter will operate until it hits its terminating condition. This will clear this bit. It should be noted that when operating in restart mode, there is no terminating condition for the counter, so this bit will never clear. Clearing this bit will halt the timer counter. Setting this bit will: • Reset the clock divider counter. • Enable the clock divider counter. • Start the timer counter. • Clear all interrupts. Clearing this bit will: • Disable the clock divider counter. • Stop the timer counter.	R/W	0h	Timer_Reset
4	SOFT_RESET This is a soft reset. This is self clearing 1 cycle after it is written.	WO	0h	Timer_Reset
3	AUTO_RESTART This will select the action taken upon completing a count. 1=The counter will automatically restart the count, using the contents of the Timer Preload Register to load the Timer Count Register The interrupt will be set in edge mode 0=The counter will simply enter a done state and wait for further control inputs. The interrupt will be set in level mode.	R/W	0h	Timer_Reset

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	COUNT_UP This selects the counter direction. When the counter in incrementing the counter will saturate and trigger the event when it reaches all F's. When the counter is decrementing the counter will saturate when it reaches 0h. 1=The counter will increment 0=The counter will decrement	R/W	0h	Timer_Reset
1	Reserved	RES	-	-
0	ENABLE This enables the block for operation. 1=This block will function normally 0=This block will gate its clock and go into its lowest power state	R/W	0h	Timer_Reset

© 2020 Microchip Technology Inc.

20.0 INPUT CAPTURE AND COMPARE TIMER

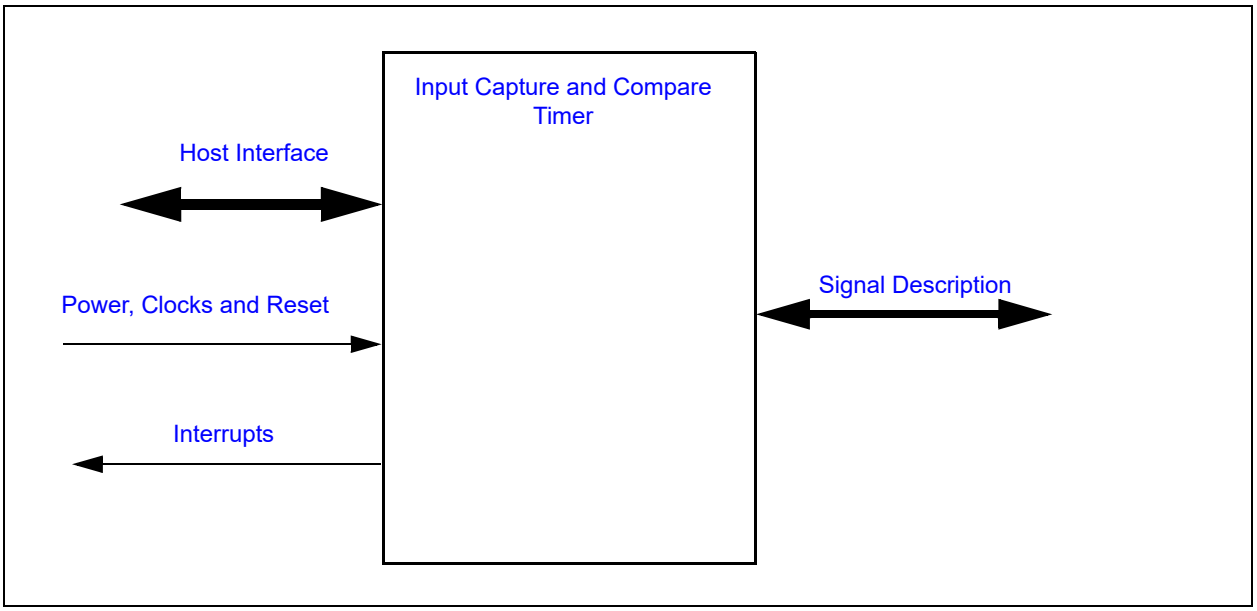
20.1 Introduction

The Input Capture and Compare Timers block contains a 32-bit timer running at the main system clock frequency. The timer is free-running and is associated with six 32-bit capture registers and two compare registers. Each capture register can record the value of the free-running timer based on a programmable edge of its associated input pin. An interrupt can be generated for each capture register each time it acquires a new timer value. The timer can also generate an interrupt when it automatically resets and can additionally generate two more interrupts when the timer matches the value in either of two 32-bit compare registers.

20.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 20-1: I/O DIAGRAM OF BLOCK



20.3 Signal Description

TABLE 20-1: SIGNAL DESCRIPTION

Name	Direction	Description
ICTx	INPUT	External capture trigger signal for Capture Register.
CTOUT0	OUTPUT	External compare match signal for Compare Register 0
CTOUT1	OUTPUT	External compare match signal for Compare Register 1
Note: Any ICTx can be connected to any Capture register using the ICT MUX Select Register.		

20.4 Host Interface

The registers defined for 16-bit Timers are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

20.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

20.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

20.5.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for this block.

20.5.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

20.6 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
CAPTURE TIMER	This interrupt event fires when the 32-bit free running counter overflows from FFFF_FFFFh to 0000_0000h.
CAPTURE 0	This interrupt event fires when Capture Register 0 acquires a new value.
CAPTURE 1	This interrupt event fires when Capture Register 1 acquires a new value.
CAPTURE 2	This interrupt event fires when Capture Register 2 acquires a new value.
CAPTURE 3	This interrupt event fires when Capture Register 3 acquires a new value.
CAPTURE 4	This interrupt event fires when Capture Register 4 acquires a new value.
CAPTURE 5	This interrupt event fires when Capture Register 5 acquires a new value.
COMPARE 0	This interrupt event fires when the contents of Compare 0 Register match the contents of the Free Running Counter.
COMPARE 1	This interrupt event fires when the contents of Compare 1 Register match the contents of the Free Running Counter.

© 2020 Microchip Technology Inc.

20.7 Low Power Modes

The Capture and Compare Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active. The block is inactive if the ACTIVATE bit is de-asserted, and will also become inactive when the block's SLEEP_EN signal is asserted.

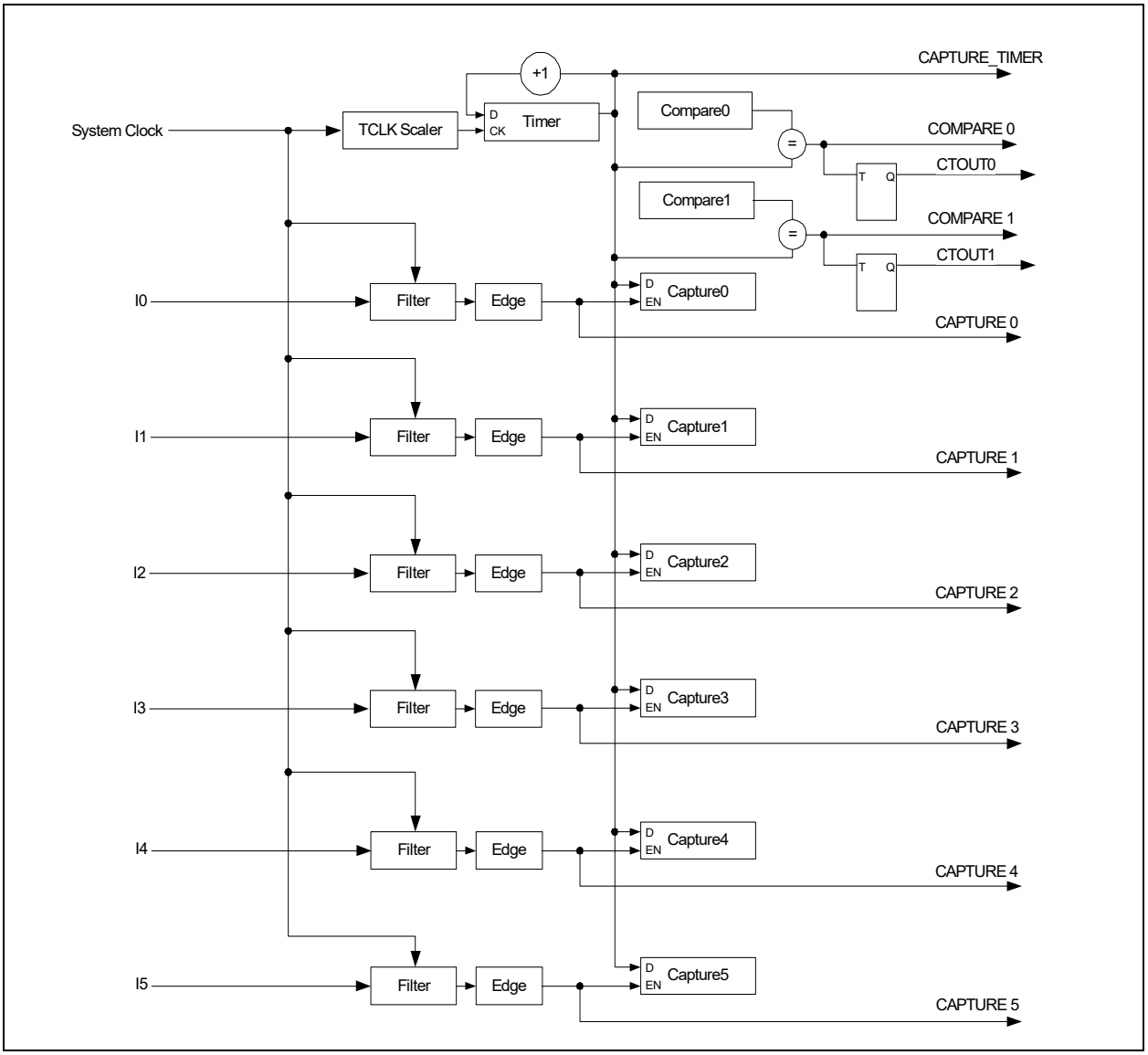
When the block returns from sleep, if enabled, the Free Running Timer Register value will continue counting from where it was when the block entered the Sleep state.

20.8 Description

The Input Capture and Compare Timer block has ICT Channels inputs and these can be connected to any of the 6 Capture Compare timer

Note:	The CCT0 to CCT5 blocks are expanded and shown in **FIGURE 20-2: "Capture and Compare Timer Block Diagram"**

FIGURE 20-2: CAPTURE AND COMPARE TIMER BLOCK DIAGRAM



20.8.1 TIMER CLOCK

Any of the frequencies listed in Table 20-2 may be used as the time base for the Free Running Counter.

TABLE 20-2: TIMER CLOCK FREQUENCIES

Timer Clock Select	Frequency Divide Select	Frequency Selected
0000b	Divide by 1	48MHz
0001b	Divide by 2	24MHz
0010b	Divide by 4	12MHz
0011b	Divide by 8	6MHz
0100b	Divide by 16	3MHz
0101b	Divide by 32	1.5MHz
0110b	Divide by 64	750KHz
0111b	Divide by 128	375KHz
1xxxb	Reserved	Reserved

For the Timer Clock, the **Timer Clock Select** value is defined by the TCLK field in the Capture and Compare Timer Control Register

20.8.2 FILTER CLOCK AND NOISE FILTER

The noise filter uses the Filter Clock (FCLK) to filter the signal on the Input Capture pins. An Input Capture pin must remain in the same state for three FCLK ticks before the internal state changes. The FILTER_BYPASS bit for the Input Capture pin may be used to bypass the input filter. Each Capture Register can individually bypass the filter.

When the input filter is bypassed, the minimum period of FCLK must be at least 2X the duration of an input signal pulse in order for an edge event to be captured reliably. When the input filter is enabled, the minimum period of FCLK must be at least 4X the duration of an input signal pulse in order for an edge event to be captured reliably.

20.9 Operation

20.9.1 INPUT CAPTURE

The Input Capture block consists of a free-running 32-bit timer and 2 capture registers. Each of the capture registers is associated with an input pin as well as an interrupt source bit in the Interrupt Aggregator: The Capture registers store the current value of the Free Running timer whenever the associated input signal changes, according to the programmed edge detection. An interrupt is also generated to the EC. The Capture registers are read-only. The registers are updated every time an edge is detected. If software does not read the register before the next edge, the value is lost.

20.9.2 COMPARE TIMER

There are two 32-bit Compare registers. Each of these registers can independently generate an interrupt to the EC when the 32-bit Free Running Timer matches the contents of the Compare register. The compare operation for each is enabled or disabled by a bit in the Capture and Compare Timer Control Register.

20.9.2.1 Interrupt Generation

Whenever a Compare Timer is enabled and the Compare register matches the Free Running Timer, a COMPARE event is sent to the Interrupt Aggregator. The event will trigger an EC interrupt if enabled by the appropriate Interrupt Enable register in the Aggregator.

20.9.2.2 Compare Output Generation

Each Compare Timer is associated with a toggle flip-flop. When the 32-bit Free Running Timer matches the contents of the Compare register the output off the flip-flop is complemented. Each of the toggle flip-flops can be independently set or cleared by using the COMPARE_SET or COMPARE_CLEAR fields, respectively, in the Capture and Compare Timer Control Register.

A Compare Timer should be disabled before setting or clearing the output, when updating the Compare register, or when updating the Free Running Timer, so spurious events are not generated by the matcher.

© 2020 Microchip Technology Inc.

20.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Input Capture and Compare Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

Note:	All registers in this block must be accessed as DWORDs.

TABLE 20-3: REGISTER SUMMARY

Offset	Register Name
00h	Capture and Compare Timer Control Register
04h	Capture Control 0 Register
08h	Capture Control 1 Register
0Ch	Free Running Timer Register
10h	Capture 0 Register
14h	Capture 1 Register
18h	Capture 2 Register
1Ch	Capture 3 Register
20h	Capture 4 Register
24h	Capture 5 Register
28h	Compare 0 Register
2Ch	Compare 1 Register
30h	ICT MUX Select Register

20.10.1 CAPTURE AND COMPARE TIMER CONTROL REGISTER

Note:	It is not recommended to use Read-Modify-Write operations on this register. May inadvertently cause the COMPARE_SET and COMPARE_CLEAR bits to be written to ‘1’ in error.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:26	Reserved	RES	-	-
25	COMPARE_CLEAR0 When read, returns the current value off the Compare Timer Output 0 state. If written with a ‘1b’, the output state is cleared to ‘0’. Writes have no effect if COMPARE_SET1 in this register is written with a ‘1b’ at the same time. Writes of ‘0b’ have no effect.	R/WC	0	RESET_SYS

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
24	COMPARE_CLEAR1 When read, returns the current value off the Compare Timer Output 1 state. If written with a ‘1b’, the output state is cleared to ‘0’. Writes have no effect if COMPARE_SET0 in this register is written with a ‘1b’ at the same time. Writes of ‘0b’ have no effect.	R/WC	0	RESET_SYS
23:18	Reserved	RES	-	-
17	COMPARE_SET0 When read, returns the current value off the Compare Timer Output 0 state. • If written with a ‘1b’, the output state is set to ‘1’. • Writes of ‘0b’ have no effect	R/WS	0	RESET_SYS
16	COMPARE_SET1 When read, returns the current value off the Compare Timer Output 1 state. If written with a ‘1b’, the output state is set to ‘1’. Writes of ‘0b’ have no effect	R/WS	0	RESET_SYS
15:10	Reserved	RES	-	-
9	COMPARE_ENABLE1 Compare Enable for Compare 1 Register. When enabled, a match between the Compare 1 Register and the Free Running Timer Register will cause the TOUT1 output to toggle and will send a COMPARE event to the Interrupt Aggregator. 1=Enabled 0=Disabled	R/W	0b	RESET_SYS
8	COMPARE_ENABLE0 Compare Enable for Compare 0 Register. When enabled, a match between the Compare 0 Register and the Free Running Timer Register will cause the TOUT0 output to toggle and will send a COMPARE event to the Interrupt Aggregator. 1=Enabled 0=Disabled	R/W	0b	RESET_SYS
7	Reserved	RES	-	-
6:4	TCLK This 3-bit field sets the clock source for the Free-Running Counter. See Table 20-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET_SYS
3	Reserved	RES	-	-
2	FREE_RESET Free Running Timer Reset. This bit stops the timer and resets the internal counter to 0000_0000h. This bit does not affect the FREE_ENABLE bit. This bit is self clearing after the timer is reset. 1=Timer reset 0=Normal timer operation	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	FREE_ENABLE Free-Running Timer Enable. This bit is used to start and stop the free running timer. This bit does not reset the timer count. The timer starts counting at 0000_0000h on reset and wraps around back to 0000_0000h after it reaches FFFF_FFFFh. The FREE_ENABLE bit is cleared after the RESET cycle is done. Firmware must poll the FREE_RESET bit to determine when it is safe to re-enable the timer. 1=Timer is enabled. The Free Running Timer Register is read-only. 0=Timer is disabled. The Free Running Timer Register is writable.	R/W	0h	RESET_SYS
0	ACTIVATE 1=The timer block is in a running state 0=The timer block is powered down and all clocks are gated	R/W	0h	RESET_SYS

20.10.2 CAPTURE CONTROL 0 REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	FCLK_SEL3 This 3-bit field sets the clock source for the input filter for Capture Register 3. See Table 20-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
28:27	Reserved	RES	-	-
26	FILTER_BYP3 This bit enables bypassing the input noise filter for Capture Register 3, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
25:24	CAPTURE_EDGE3 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 3. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS
23:21	FCLK_SEL2 This 3-bit field sets the clock source for the input filter for Capture Register 2. See Table 20-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
20:19	Reserved	RES	-	-

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
18	FILTER_BYP2 This bit enables bypassing the input noise filter for Capture Register 2, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
17:16	CAPTURE_EDGE2 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 2. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS
15:13	FCLK_SEL1 This 3-bit field sets the clock source for the input filter for Capture Register 1. See Table 20-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET_SYS
12:11	Reserved	RES	-	-
10	FILTER_BYP1 This bit enables bypassing the input noise filter for Capture Register 1, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
9:8	CAPTURE_EDGE1 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 1. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS
7:5	FCLK_SEL0 This 3-bit field sets the clock source for the input filter for Capture Register 0. See Table 20-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
4:3	Reserved	RES	-	-
2	FILTER_BYP0 This bit enables bypassing the input noise filter for Capture Register 0, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1:0	CAPTURE_EDGE0 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 0. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS

20.10.3 CAPTURE CONTROL 1 REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:13	FCLK_SEL5 This 3-bit field sets the clock source for the input filter for Capture Register 5. See Table 20-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0b	RESET_SYS
12:11	Reserved	RES	-	-
10	FILTER_BYP5 This bit enables bypassing the input noise filter for Capture Register 5, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS
9:8	CAPTURE_EDGE5 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 5. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS
7:5	FCLK_SEL4 This 3-bit field sets the clock source for the input filter for Capture Register 4. See Table 20-2, "Timer Clock Frequencies" for a list of available frequencies.	R/W	0h	RESET_SYS
4:3	Reserved	RES	-	-
2	FILTER_BYP4 This bit enables bypassing the input noise filter for Capture Register 4, so that the input signal goes directly into the timer. 1=Input filter bypassed 0=Input filter enabled	R/W	0h	RESET_SYS

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1:0	CAPTURE_EDGE4 This field selects the edge type that triggers the capture of the Free Running Counter into Capture Register 4. 3=Capture event disabled 2=Both rising and falling edges 1=Rising edges 0=Falling edges	R/W	0h	RESET_SYS

20.10.4 FREE RUNNING TIMER REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	FREE_RUNNING_TIMER This register contains the current value of the Free Running Timer. A Capture Timer interrupt is signaled to the Interrupt Aggregator when this register transitions from FFFF_FFFFh to 0000_0000h. When FREE_ENABLE in the Capture and Compare Timer Control Register is '1', this register is read-only. When FREE_ENABLE is '0', this register may be written.	R/W	0h	RESET_SYS

20.10.5 CAPTURE 0 REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_0 This register saves the value copied from the Free Running timer on a programmed edge of ICT0.	R	0h	RESET_SYS

20.10.6 CAPTURE 1 REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_1 This register saves the value copied from the Free Running timer on a programmed edge of ICT1.	R	0h	RESET_SYS

20.10.7 CAPTURE 2 REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_2 This register saves the value copied from the Free Running timer on a programmed edge of ICT2.	R	0h	RESET_SYS

© 2020 Microchip Technology Inc.

20.10.8 CAPTURE 3 REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_3 This register saves the value copied from the Free Running timer on a programmed edge of ICT3.	R	0h	RESET_SYS

20.10.9 CAPTURE 4 REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_4 This register saves the value copied from the Free Running timer on a programmed edge of ICT4.	R	0h	RESET_SYS

20.10.10 CAPTURE 5 REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	CAPTURE_5 This register saves the value copied from the Free Running timer on a programmed edge of ICT5.	R	0h	RESET_SYS

20.10.11 COMPARE 0 REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COMPARE_0 A COMPARE 0 interrupt is generated when this register matches the value in the Free Running Timer.	R/W	0h	RESET_SYS

20.10.12 COMPARE 1 REGISTER

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COMPARE_1 A COMPARE 1 interrupt is generated when this register matches the value in the Free Running Timer.	R/W	0h	RESET_SYS

20.10.13 ICT MUX SELECT REGISTER

This register selects the pin mapping to the capture register.

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
23:20	Mux Select for Capture 5 register.	R/W	5h	RESET_SYS
19:16	Mux Select for Capture 4 register.	R/W	4h	RESET_SYS
15:12	Mux Select for Capture 3 register.	R/W	3h	RESET_SYS
11:8	Mux Select for Capture 2 register.	R/W	2h	RESET_SYS
7:4	Mux Select for Capture 1 register.	R/W	1h	RESET_SYS
3:0	Mux Select for Capture 0 register.	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

21.0 HIBERNATION TIMER

21.1 Introduction

The Hibernation Timer can generate a wake event to the Embedded Controller (EC) when it is in a hibernation mode. This block supports wake events up to 2 hours in duration. The timer is a 16-bit binary count-down timer that can be programmed in 30.5µs and 0.125 second increments for period ranges of 30.5µs to 2s or 0.125s to 136.5 minutes, respectively. Writing a non-zero value to this register starts the counter from that value. A wake-up interrupt is generated when the count reaches zero.

21.2 References

No references have been cited for this chapter

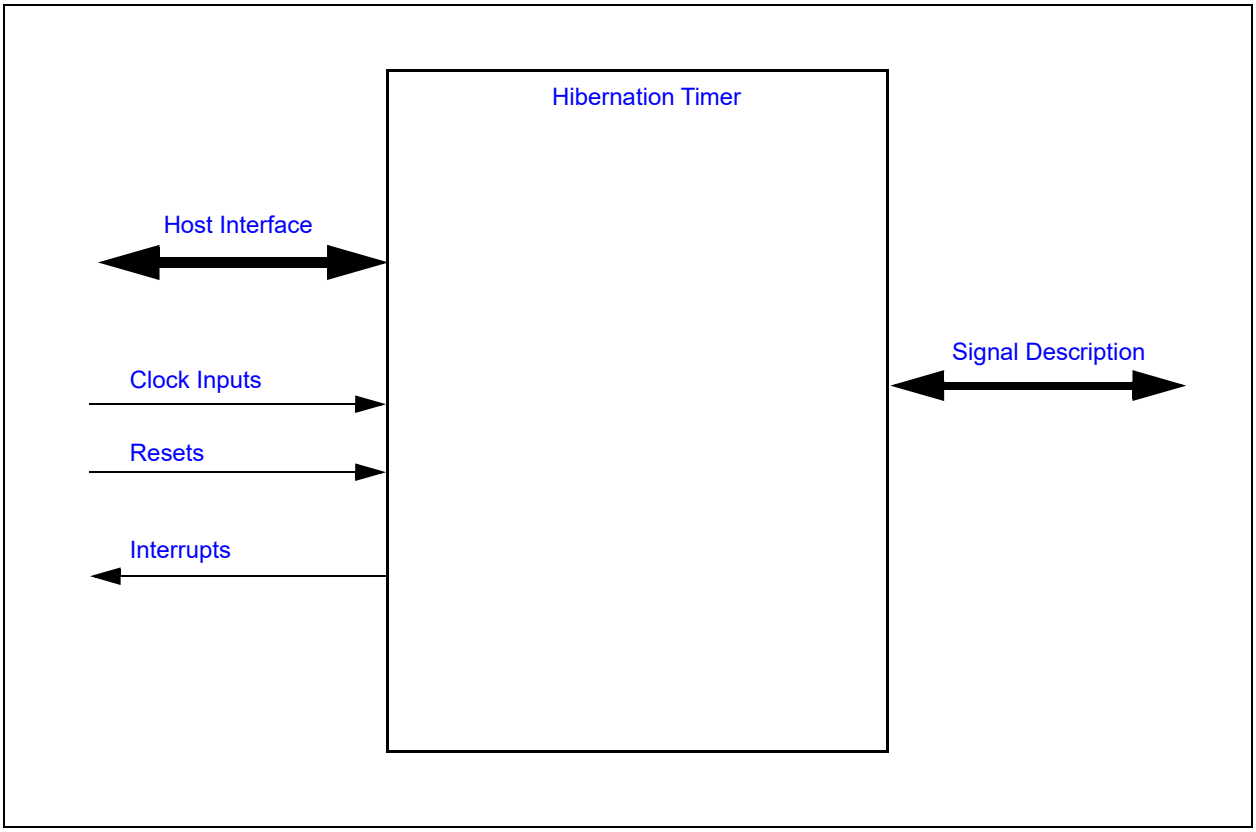
21.3 Terminology

No terms have been cited for this chapter.

21.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 21-1: HIBERNATION TIMER INTERFACE DIAGRAM



21.5 Signal Description

There are no external signals for this block.

21.6 Host Interface

The registers defined for the Hibernation Timer are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

21.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

21.7.1 POWER DOMAINS

TABLE 21-1: POWER SOURCES

Name	Description
VTR_CORE	The timer control logic and registers are all implemented on this single power domain.

21.7.2 CLOCK INPUTS

TABLE 21-2: CLOCK INPUTS

Name	Description
32KHz	This is the clock source to the timer logic. The Pre-scaler may be used to adjust the minimum resolution per bit of the counter. if the main oscillator is stopped then an external 32.768kHz clock source must be active for the Hibernation Timer to continue to operate.

21.7.3 RESETS

TABLE 21-3: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

21.8 Interrupts

This section defines the interrupt Interface signals routed to the chip interrupt aggregator.

Each instance of the Hibernation Timer in the MEC152x can be used to generate interrupts and wake-up events when the timer decrements to zero.

TABLE 21-4: INTERRUPT INTERFACE SIGNAL DESCRIPTION TABLE

Name	Direction	Description
HTIMER	Output	Signal indicating that the timer is enabled and decrements to 0. This signal is used to generate an Hibernation Timer interrupt event.

21.9 Low Power Modes

The timer operates off of the 32KHz clock, and therefore will operate normally when the main oscillator is stopped.

The sleep enable inputs have no effect on the Hibernation Timer and the clock required outputs are only asserted during register read/write cycles for as long as necessary to propagate updates to the block core.

21.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Hibernation Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 21-5: REGISTER SUMMARY

Offset	Register Name
00h	HTimer Preload Register
04h	HTimer Control Register
08h	HTimer Count Register

© 2020 Microchip Technology Inc.

21.10.1 HTIMER PRELOAD REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	HT_PRELOAD This register is used to set the Hibernation Timer Preload value. Writing this register to a non-zero value resets the down counter to start counting down from this programmed value. Writing this register to 0000h disables the hibernation counter. The resolution of this timer is determined by the CTRL bit in the HTimer Control Register. Writes to the HTimer Control Register are completed with an EC bus cycle.	R/W	000h	RESET_SYS

21.10.2 HTIMER CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:1	Reserved	RES	-	-
0	CTRL 1=The Hibernation Timer has a resolution of 0.125s per LSB, which yields a maximum time in excess of 2 hours. 0=The Hibernation Timer has a resolution of 30.5µs per LSB, which yields a maximum time of ~2seconds.	R	0000h	RESET_SYS

21.10.3 HTIMER COUNT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:0	COUNT The current state of the Hibernation Timer.	R	0000h	RESET_SYS

22.0 RTOS TIMER

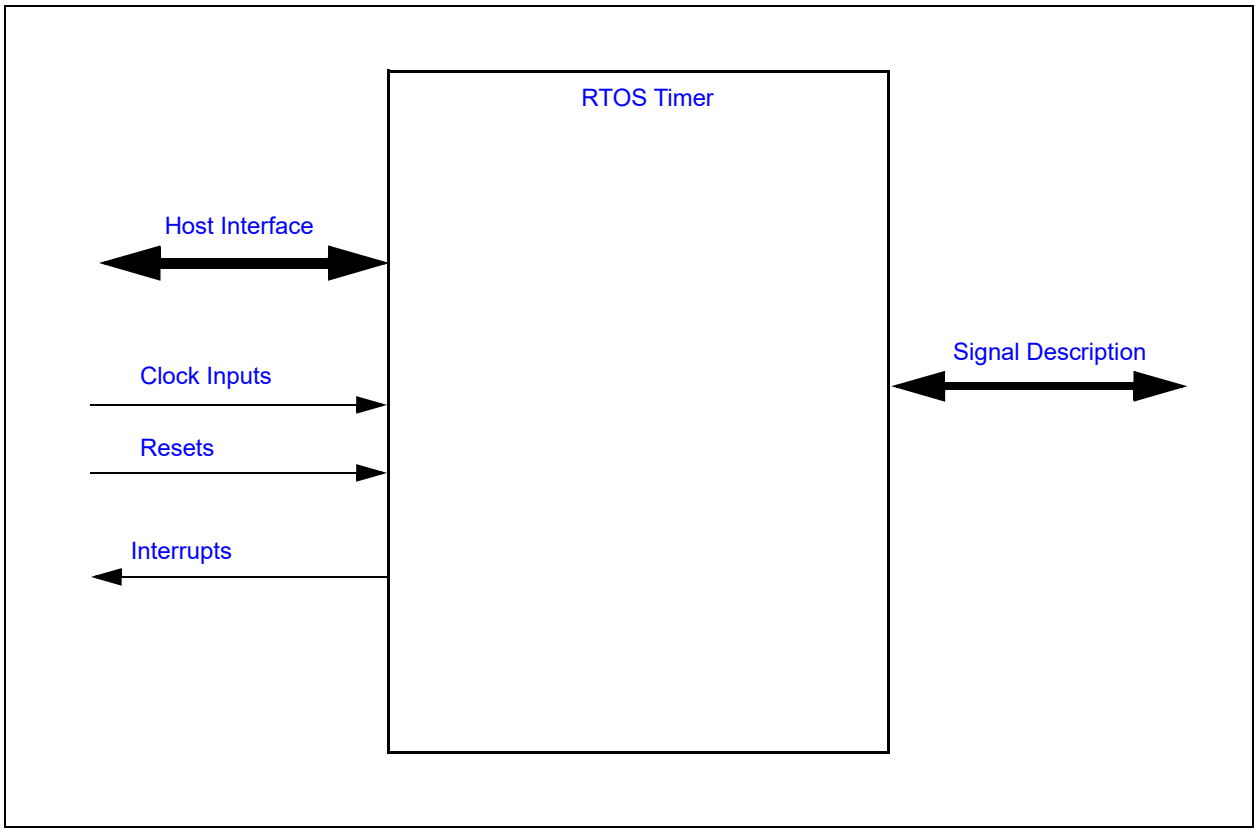
22.1 Introduction

The RTOS Timer is a low-power, 32-bit timer designed to operate on the 32kHz oscillator which is available during all chip sleep states. This allows firmware the option to sleep the processor and wake after a programmed amount of time. The timer may be used as a one-shot timer or a continuous timer. When the timer transitions to 0 it is capable of generating a wake-capable interrupt to the embedded controller. This timer may be halted during debug by hardware or via a software control bit.

22.2 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 22-1: I/O DIAGRAM OF BLOCK



22.3 Signal Description

Name	Description
HALT	RTOS Timer Halt signal. This signal is connected to the same signal that halts the embedded controller during debug (e.g., JTAG Debugger is active, break points, etc.).

22.4 Host Interface

The Embedded Controller (EC) may access this block via the registers defined in Section 22.9, "EC Registers".

© 2020 Microchip Technology Inc.

22.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

22.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The timer control logic and registers are all implemented on this single power domain.

22.5.2 CLOCK INPUTS

Name	Description
32KHz	This is the clock source to the timer logic.

22.5.3 RESETS

Name	Description
RESET_SYS	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

22.6 Interrupts

Source	Description
RTOS_TIMER	RTOS Timer interrupt event. The interrupt is signaled when the timer counter transitions from 1 to 0 while counting.

22.7 Low Power Modes

The Basic Timer may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. This block is only be permitted to enter low power modes when the block is not active.

22.8 Description

The RTOS Timer is a basic down counter that can operate either as a continuous timer or a one-shot timer. When it is started, the counter is loaded with a pre-load value and counts towards 0. When the counter counts down from 1 to 0, it will generate an interrupt. In one-shot mode (the AUTO_RELOAD bit is '0'), the timer will then halt; in continuous mode (the AUTO_RELOAD bit is '1'), the counter will automatically be restarted with the pre-load value.

The timer counter can be halted by firmware by setting the FIRMWARE_TIMER_HALT bit to '1'. In addition, if enabled, the timer counter can be halted by the external HALT signal.

22.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the RTOS Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 22-1: REGISTER SUMMARY

Offset	Register Name
00h	RTOS Timer Count Register
04h	RTOS Timer Preload Register
08h	RTOS Timer Control Register
0Ch	Soft Interrupt Register

22.9.1 RTOS TIMER COUNT REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER This register contains the current value of the RTOS Timer counter. This register should be read as a DWORD. There is no latching mechanism of the upper bytes implemented if the register is accessed as a byte or word. Reading the register with byte or word operations may give incorrect results.	R/W	0h	RESET_SYS

22.9.2 RTOS TIMER PRELOAD REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	PRE_LOAD The this register is loaded into the RTOS Timer counter either when the TIMER_START bit is written with a ‘1’, or when the timer counter counts down to ‘0’ and the AUTO_RELOAD bit is ‘1’. This register must be programmed with a new count value before the TIMER_START bit is set to ‘1’. If this register is updated while the counter is operating, the new count value will only take effect if the counter transitions form 1 to 0 while the AUTO_RELOAD bit is set.	R/W	0h	RESET_SYS

22.9.3 RTOS TIMER CONTROL REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	RES	-	-
4	FIRMWARE_TIMER_HALT 1=The timer counter is halted. If the counter was running, clearing this bit will restart the counter from the value at which it halted 0=The timer counter, if enabled, will continue to run	R/W	0h	RESET_SYS
3	EXT_HARDWARE_HALT_EN 1=The timer counter is halted when the external HALT signal is asserted. Counting is always enabled if HALT is de-asserted. 0=The HALT signal does not affect the RTOS Timer	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	TIMER_START Writing a '1' to this bit will load the timer counter with the RTOS Timer Preload Register and start counting. If the Preload Register is 0, counting will not start and this bit will be cleared to '0'. Writing a '0' to this bit will halt the counter and clear its contents to 0. The RTOS timer interrupt will not be generated. This bit is automatically cleared if the AUTO_RELOAD bit is '0' and the timer counter transitions from 1 to 0.	R/W	0h	RESET_SYS
1	AUTO_RELOAD 1=The the RTOS Timer Preload Register is loaded into the timer counter and the counter is restarted when the counter transitions from 1 to 0 0=The timer counter halts when it transitions from 1 to 0 and will not restart	R/W	0h	RESET_SYS
0	BLOCK_ENABLE 1=RTOS timer counter is enabled 0=RTOS timer disabled. All register bits are reset to their default state	R/W	0h	RESET_SYS

22.9.4 SOFT INTERRUPT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	SWI_3 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS
2	SWI_2 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS
1	SWI_1 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS
0	SWI_0 Software Interrupt. A write of a '1' to this bit will generate an SWI interrupt to the EC. Writes of a '0' have no effect. Reads return '0'.	W	0h	RESET_SYS

23.0 REAL TIME CLOCK

23.1 Introduction

This block provides the capabilities of an industry-standard 146818B Real-Time Clock module, without CMOS RAM. Enhancements to this architecture include:

- Industry standard Day of Month Alarm field, allowing for monthly alarms
- Configurable, automatic Daylight Savings adjustment
- Week Alarm for periodic interrupts and wakes based on Day of Week
- System Wake capability on interrupts.

23.2 References

1. Motorola 146818B Data Sheet, available on-line
2. Intel Lynx Point PCH EDS specification

23.3 Terminology

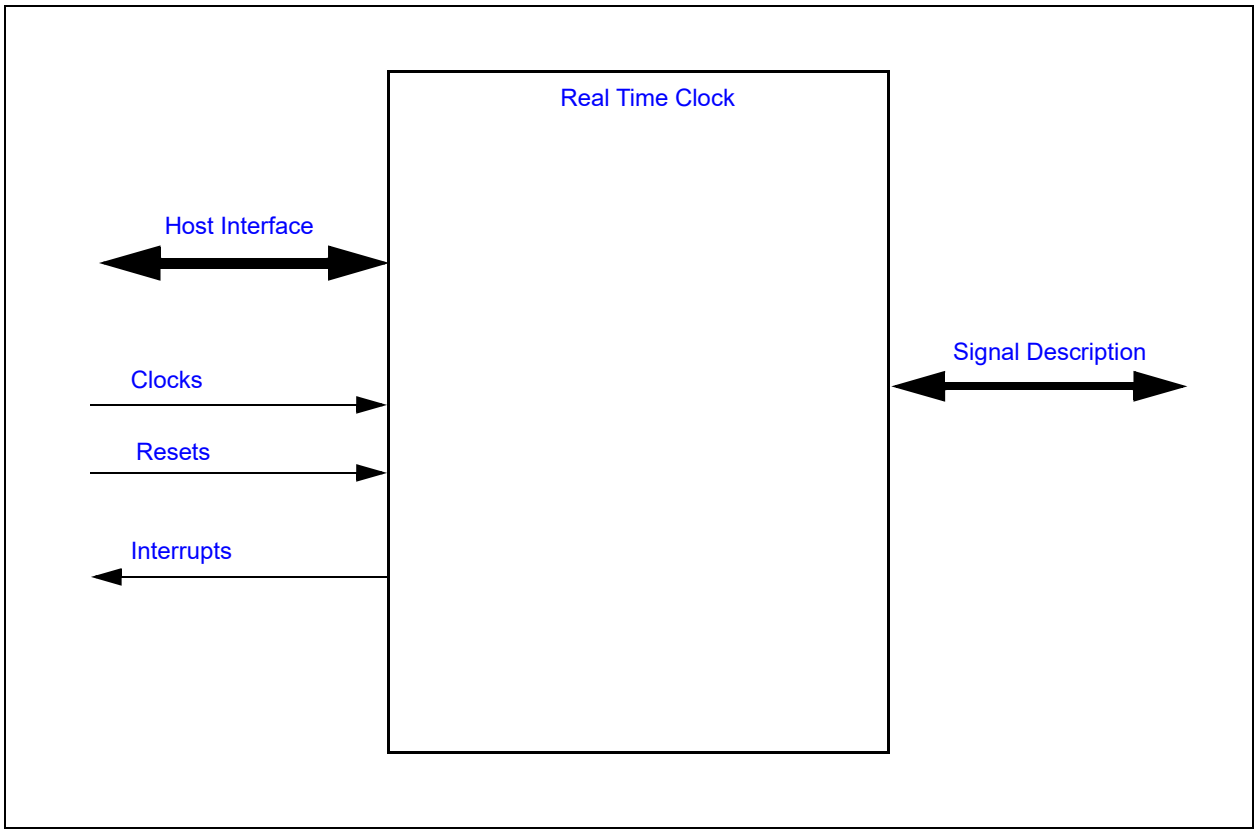
Time and Date Registers:

This is the set of registers that are automatically counted by hardware every 1 second while the block is enabled to run and to update. These registers are: **Seconds**, **Minutes**, **Hours**, **Day of Week**, **Day of Month**, **Month**, and **Year**.

23.4 Interface

This block's connections are entirely internal to the chip.

FIGURE 23-1: I/O DIAGRAM OF BLOCK



© 2020 Microchip Technology Inc.

23.5 Signal Description

There are no external signals.

23.6 Host Interface

The registers defined for the Real Time Clock are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

23.7 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

23.7.1 POWER DOMAINS

TABLE 23-1: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR_CORE	This power well sources only host register accesses. The block continues to operate internally while this rail is down.

23.7.2 CLOCKS

TABLE 23-2: CLOCKS

Name	Description
32KHz	This clock input drives all internal logic, and will be present at all times that the VBAT well is powered.

23.7.3 RESETS

TABLE 23-3: RESET SIGNALS

Name	Description
RESET_VBAT	This reset signal is used in the RESET_RTC signal to reset all of the registers and logic in this block. It directly resets the Soft Reset bit in the RTC Control Register.
RESET_RTC	This reset signal resets all of the registers and logic in this block, except for the Soft Reset bit in the RTC Control Register. It is triggered by RESET_VBAT, but can also be triggered by a SOFT_RESET from the RTC Control Register.
RESET_SYS	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR_CORE powered circuitry on-chip. Otherwise it has no effect on the internal state.
SOFT_RESET	This is the block reset and resets all the registers and logic in the block

23.8 Interrupts

TABLE 23-4: SYSTEM INTERRUPTS

Source	Description
RTC	This interrupt source for the SIRQ logic is generated when any of the following events occur: • Update complete. This is triggered, at 1-second intervals, when the Time register updates have completed • Alarm. This is triggered when the alarm value matches the current time (and date, if used) • Periodic. This is triggered at the chosen programmable rate

TABLE 23-5: EC INTERRUPTS

Source	Description
RTC	This interrupt is signaled to the Interrupt Aggregator when any of the following events occur: • Update complete. This is triggered, at 1-second intervals, when the Time register updates have completed • Alarm. This is triggered when the alarm value matches the current time (and date, if used) • Periodic. This is triggered at the chosen programmable rate
RTC ALARM	This wake interrupt is signaled to the Interrupt Aggregator when an Alarm event occurs.

23.9 Low Power Modes

The RTC has no low-power modes. It runs continuously while the VBAT well is powered.

23.10 Description

This block provides the capabilities of an industry-standard 146818B Real-Time Clock module, excluding the CMOS RAM and the SQW output. See the following registers, which represent enhancements to this architecture. These enhancements are listed below.

See the Date Alarm field of Register D for a Day of Month qualifier for alarms.

See the Week Alarm Register for a Day of Week qualifier for alarms.

See the registers Daylight Savings Forward Register and Daylight Savings Backward Register for setting up hands-off Daylight Savings adjustments.

See the RTC Control Register for enhanced control over the block's operations.

23.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Real Time Clock. Host access for each register listed in this table is defined as an offset in the Host address space to the Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the Real Time Clock shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column.

TABLE 23-6: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Seconds Register
01h	Seconds Alarm Register
02h	Minutes Register
03h	Minutes Alarm Register
04h	Hours Register
05h	Hours Alarm Register
06h	Day of Week Register
07h	Day of Month Register
08h	Month Register
09h	Year Register
0Ah	Register A
0Bh	Register B

© 2020 Microchip Technology Inc.

TABLE 23-6: RUNTIME REGISTER SUMMARY (CONTINUED)

Offset	Register Name
0Ch	Register C
0Dh	Register D
0Eh	Reserved
0Fh	Reserved
10h	RTC Control Register
14h	Week Alarm Register
18h	Daylight Savings Forward Register
1Ch	Daylight Savings Backward Register
20h	TEST

Note: This extended register set occupies offsets that have historically been used as CMOS RAM. Code ported to use this block should be examined to ensure that it does not assume that RAM exists in this block.

23.11.1 SECONDS REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SECONDS Displays the number of seconds past the current minute, in the range 0--59. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

23.11.2 SECONDS ALARM REGISTER

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	SECONDS_ALARM Holds a match value, compared against the Seconds Register to trigger the Alarm event. Values written to this register must use the format defined by the current setting of the DM bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RESET_RTC

23.11.3 MINUTES REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MINUTES Displays the number of minutes past the current hour, in the range 0--59. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

23.11.4 MINUTES ALARM REGISTER

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MINUTES_ALARM Holds a match value, compared against the Minutes Register to trigger the Alarm event. Values written to this register must use the format defined by the current setting of the DM bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RESET_RTC

23.11.5 HOURS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	HOURS_AM_PM In 12-hour mode (see bit "24/12" in register B), this bit indicates AM or PM. 1=PM 0=AM	R/W	0b	RESET_RTC
6:0	HOURS Displays the number of the hour, in the range 1--12 for 12-hour mode (see bit "24/12" in register B), or in the range 0--23 for 24-hour mode. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

23.11.6 HOURS ALARM REGISTER

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	HOURS_ALARM Holds a match value, compared against the Hours Register to trigger the Alarm event. Values written to this register must use the format defined by the current settings of the DM bit and the 24/12 bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RESET_RTC

© 2020 Microchip Technology Inc.

23.11.7 DAY OF WEEK REGISTER

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DAY_OF_WEEK Displays the day of the week, in the range 1 (Sunday) through 7 (Saturday). Numbers in this range are identical in both binary and BCD notation, so this register's format is unaffected by the DM bit.	R/W	00h	RESET_RTC

23.11.8 DAY OF MONTH REGISTER

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DAY_OF_MONTH Displays the day of the current month, in the range 1--31. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

23.11.9 MONTH REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	MONTH Displays the month, in the range 1--12. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

23.11.10 YEAR REGISTER

Offset	09h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	YEAR Displays the number of the year in the current century, in the range 0 (year 2000) through 99 (year 2099). Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RESET_RTC

© 2020 Microchip Technology Inc.

23.11.11 REGISTER A

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UPDATE_IN_PROGRESS '0' indicates that the Time and Date registers are stable and will not be altered by hardware soon. '1' indicates that a hardware update of the Time and Date registers may be in progress, and those registers should not be accessed by the host program. This bit is set to '1' at a point 488us (16 cycles of the 32K clock) before the update occurs, and is cleared immediately after the update. See also the Update-Ended Interrupt, which provides more useful status.	R	0b	RESET_RTC
6:4	DIVISION_CHAIN_SELECT This field provides general control for the Time and Date register updating logic. 11xb=Halt counting. The next time that 010b is written, updates will begin 500ms later. 010b=Required setting for normal operation. It is also necessary to set the Block Enable bit in the RTC Control Register to '1' for counting to begin 000b=Reserved. This field should be initialized to another value before Enabling the block in the RTC Control Register Other values Reserved	R/W	000b	RESET_RTC
3:0	RATE_SELECT This field selects the rate of the Periodic Interrupt source. See Table 23-7	R/W	0h	RESET_RTC

TABLE 23-7: REGISTER A FIELD RS: PERIODIC INTERRUPT SETTINGS

RS (hex)	Interrupt Period
0	Never Triggered
1	3.90625 ms
2	7.8125 ms
3	122.070 us
4	244.141 us
5	488.281 us
6	976.5625 us
7	1.953125 ms
8	3.90625 ms
9	7.8125 ms
A	15.625 ms
B	31.25 ms
C	62.5 ms
D	125 ms
E	250 ms
F	500 ms

© 2020 Microchip Technology Inc.

23.11.12 REGISTER B

Offset	0Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	UPDATE_CYCLE_INHIBIT In its default state '0', this bit allows hardware updates to the Time and Date registers, which occur at 1-second intervals. A '1' written to this field inhibits updates, allowing these registers to be cleanly written to different values. Writing '0' to this bit allows updates to continue.	R/W	0b	RESET_RTC
6	PERIODIC_INTERRUPT_ENABLE 1=Alows the Periodic Interrupt events to be propagated as interrupts 0=Periodic events are not propagates as interrupts	R/W	0b	RESET_RTC
5	ALARM_INTERRUPT_ENABLE 1=Alows the Alarm Interrupt events to be propagated as interrupts 0=Alarm events are not propagates as interrupts	R/W	0b	RESET_RTC
4	UPDATE_ENDED_INTERRUPT_ENABLE 1=Alows the Update Ended Interrupt events to be propagated as interrupts 0=Update Ended events are not propagates as interrupts	R/W	0b	RESET_RTC
3	Reserved	RES	-	-
2	DATA_MODE 1=Binary Mode for Dates and Times 0=BCD Mode for Dates and Times	R/W	0b	RESET_RTC
1	HOUR_FORMAT_24_12 1=24-Hour Format for Hours and Hours Alarm registers. 24-Hour format keeps the AM/PM bit off, with value range 0--23 0=12-Hour Format for Hours and Hours Alarm registers. 12-Hour format has an AM/PM bit, and value range 1--12	R/W	0b	RESET_RTC
0	DAYLIGHT_SAVINGS_ENABLE 1=Enables automatic hardware updating of the hour, using the registers Daylight Savings Forward and Daylight Savings Backward to select the yearly date and hour for each update 0=Automatic Daylight Savings updates disabled	R/W	0b	RESET_RTC

Note: The DATA_MODE and HOUR_FORMAT_24_12 bits affect only how values are presented as they are being read and how they are interpreted as they are being written. They do not affect the internal contents or interpretations of registers that have already been written, nor do they affect how those registers are represented or counted internally. This mode bits may be set and cleared dynamically, for whatever I/O data representation is desired by the host program.

23.11.13 REGISTER C

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	INTERRUPT_REQUEST_FLAG 1=Any of bits[6:4] below is active after masking by their respective Enable bits in Register B. 0=No bits in this register are active This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
6	PERIODIC_INTERRUPT_FLAG 1=A Periodic Interrupt event has occurred since the last time this register was read. This bit displays status regardless of the Periodic Interrupt Enable bit in Register B 0=A Periodic Interrupt event has not occurred This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
5	ALARM_FLAG 1=An Alarm event has occurred since the last time this register was read. This bit displays status regardless of the Alarm Interrupt Enable bit in Register B. 0=An Alarm event has not occurred This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
4	UPDATE_ENDED_INTERRUPT_FLAG 1=A Time and Date update has completed since the last time this register was read. This bit displays status regardless of the Update-Ended Interrupt Enable bit in Register B. Presentation of this status indicates that the Time and Date registers will be valid and stable for over 999ms 0=A Time and Data update has not completed since the last time this register was read This bit is automatically cleared by every Read access to this register.	RC	0b	RESET_RTC
3:0	Reserved	RES	-	-

23.11.14 REGISTER D

Offset	0Dh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	RES	-	-
5:0	DATE_ALARM This field, if set to a non-zero value, will inhibit the Alarm interrupt unless this field matches the contents of the Month register also. If this field contains 00h (default), it represents a don't-care, allowing more frequent alarms.	R/W	00h	RESET_RTC

© 2020 Microchip Technology Inc.

23.11.15 RTC CONTROL REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	Reserved	RES	-	-
3	ALARM_ENABLE 1=Enables the Alarm features 0=Disables the Alarm features	R/W	0b	RESET_RTC
2	VCI_ENABLE 1= RTC Alarm event is routed to chip level VCI Circuitry 0= RTC alarm event is inhibited from affecting the VCI circuitry	R/W	0b	RESET_RTC
1	SOFT_RESET A '1' written to this bit position will trigger the RESET_RTC reset, resetting the block and all registers except this one and the Test Register. This bit is self-clearing at the end of the reset.	R/W	0b	RESET_VBAT
0	BLOCK_ENABLE This bit must be '1' in order for the block to function internally. Registers may be initialized first, before setting this bit to '1' to start operation.	R/W	0b	RESET_RTC

23.11.16 WEEK ALARM REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ALARM_DAY_OF_WEEK This register, if written to a value in the range 1--7, will inhibit the Alarm interrupt unless this field matches the contents of the Day of Week Register also. If this field is written to any value 11xxxxxxb (like the default FFh), it represents a don't-care, allowing more frequent alarms, and will read back as FFh until another value is written.	R/W	FFh	RESET_RTC

23.11.17 DAYLIGHT SAVINGS FORWARD REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	DST_FORWARD_AM_PM This bit selects AM vs. PM, to match bit[7] of the Hours Register if 12-Hour mode is selected in Register B at the time of writing.	R/W	0b	RESET_RTC
30:24	DST_FORWARD_HOUR This field holds the matching value for bits[6:0] of the Hours register. The written value will be interpreted according to the 24/12 Hour mode and DM mode settings at the time of writing.	R/W	00h	RESET_RTC
23:19	Reserved	RES	-	-

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
18:16	DST_FORWARD_WEEK This value matches an internally-maintained week number within the current month. Valid values for this field are: 5=Last week of month 4 =Fourth week of month 3=Third week of month 2=Second week of month 1=First week of month	R/W	0h	RESET_RTC
15:11	Reserved	RES	-	-
10:8	DST_FORWARD_DAY_OF_WEEK This field matches the Day of Week Register bits[2:0].	R/W	0h	RESET_RTC
7:0	DST_FORWARD_MONTH This field matches the Month Register.	R/W	00h	RESET_RTC

This is a 32-bit register, accessible also as individual bytes. When writing as individual bytes, ensure that the DSE bit (in Register B) is off first, or that the block is disabled or stopped (SET bit), to prevent a time update while this register may have incompletely-updated contents.

When enabled by the DSE bit in Register B, this register defines an hour and day of the year at which the Hours register will be automatically incremented by 1 additional hour.

There are no don't-care fields recognized. All fields must be already initialized to valid settings whenever the DSE bit is '1'.

Fields other than Week and Day of Week use the current setting of the DM bit (binary vs. BCD) to interpret the information as it is written to them. Their values, as held internally, are not changed by later changes to the DM bit, without subsequently writing to this register as well.

Note: An Alarm that is set inside the hour after the time specified in this register will not be triggered, because that one-hour period is skipped. This period includes the exact time (0 minutes: 0 seconds) given by this register, through the 59 minutes: 59 seconds point afterward.

23.11.18 DAYLIGHT SAVINGS BACKWARD REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	DST_BACKWARD_AM_PM This bit selects AM vs. PM, to match bit[7] of the Hours register if 12-Hour mode is selected in Register B at the time of writing.	R/W	0b	RESET_RTC
30:24	DST_BACKWARD_HOUR This field holds the matching value for bits[6:0] of the Hours register. The written value will be interpreted according to the 24/12 Hour mode and DM mode settings at the time of writing.	R/W	00h	RESET_RTC
23:19	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
18:16	DST_BACKWARD_WEEK This value matches an internally-maintained week number within the current month. Valid values for this field are: 5=Last week of month 4 =Fourth week of month 3=Third week of month 2=Second week of month 1=First week of month	R/W	0h	RESET_RTC
15:11	Reserved	RES	-	-
10:8	DST_BACKWARD_DAY_OF_WEEK This field matches the Day of Week Register bits[2:0].	R/W	0h	RESET_RTC
7:0	DST_BACKWARD_MONTH This field matches the Month Register.	R/W	00h	RESET_RTC

This is a 32-bit register, accessible also as individual bytes. When writing as individual bytes, ensure that the DSE bit (in Register B) is off first, or that the block is disabled or stopped (SET bit), to prevent a time update while this register may have incompletely-updated contents.

When enabled by the DSE bit in Register B, this register defines an hour and day of the year at which the Hours register increment will be inhibited from occurring. After triggering, this feature is automatically disabled for long enough to ensure that it will not retrigger the second time this Hours value appears, and then this feature is re-enabled automatically.

There are no don’t-care fields recognized. All fields must be already initialized to valid settings whenever the DSE bit is ‘1’.

Fields other than Week and Day of Week use the current setting of the DM bit (binary vs. BCD) to interpret the information as it is written to them. Their values, as held internally, are not changed by later changes to the DM bit, without subsequently writing to this register as well.

Note: An Alarm that is set inside the hour before the time specified in this register will be triggered twice, because that one-hour period is repeated. This period will include the exact time (0 minutes: 0 seconds) given by this register, through the 59 minutes: 59 seconds point afterward.

24.0 WEEK TIMER

24.1 Introduction

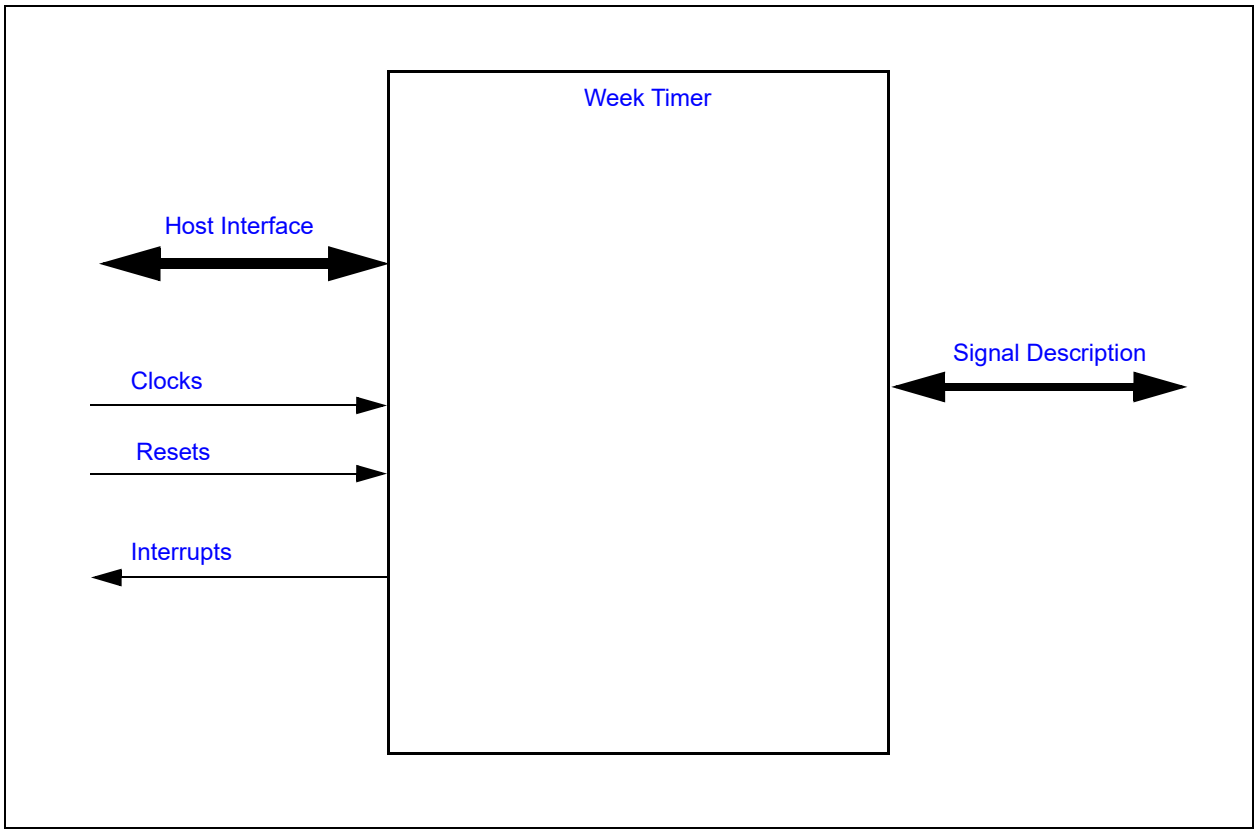
The Week Alarm Interface provides two timekeeping functions: a Week Timer and a Sub-Week Timer. Both the Week Timer and the Sub-Week Timer assert the Power-Up Event Output which automatically powers-up the system from the G3 state. Features include:

- EC interrupts based on matching a counter value
- Repeating interrupts at 1 second and sub-1 second intervals
- System Wake capability on interrupts, including Wake from Heavy Sleep

24.2 Interface

This block's connections are entirely internal to the chip.

FIGURE 24-1: I/O DIAGRAM OF BLOCK



24.3 Signal Description

TABLE 24-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
BGPO	OUTPUT	Battery-powered general purpose outputs
SYSPWR_PRES	INPUT	Input signal used to gate the POWER_UP_EVENT

© 2020 Microchip Technology Inc.

Note 1: Please refer to TABLE 1-1: for the number of BGPO's and SYSPWR_PRES availability in the package.

TABLE 24-2: INTERNAL SIGNAL DESCRIPTION TABLE

Name	Direction	Description
POWER_UP_EVENT	OUTPUT	Signal to the VBAT-Powered Control Interface. When this signal is asserted, the VCI output signal asserts. See Section 24.8, "Power-Up Events".

24.4 Host Interface

The registers defined for the Week Timer are accessible only by the EC.

24.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

24.5.1 POWER DOMAINS

TABLE 24-3: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR_CORE	This power well sources only host register accesses. The block continues to operate internally while this rail is down.

24.5.2 CLOCKS

TABLE 24-4: CLOCKS

Name	Description
48MHz	Clock used for host register access
32KHz	This 32KHz clock input drives all internal logic, and will be present at all times that the VBAT well is powered.

24.5.3 RESETS

TABLE 24-5: RESET SIGNALS

Name	Description
RESET_VBAT	This reset signal is used reset all of the registers and logic in this block.
RESET_SYS	This reset signal is used to inhibit the Host register access and isolates this block from VTR_CORE powered circuitry on-chip. Otherwise it has no effect on the internal state.

24.6 Interrupts

TABLE 24-6: EC INTERRUPTS

Source	Description
WEEK_ALARM_INT	This interrupt is signaled to the Interrupt Aggregator when the Week Alarm Counter Register is greater than or equal to the Week Timer Compare Register. The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
SUB_WEEK_ALARM_INT	This interrupt is signaled to the Interrupt Aggregator when the Sub-Week Alarm Counter Register decrements from '1' to '0'. The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
ONE_SECOND	This interrupt is signaled to the Interrupt Aggregator at an isochronous rate of once per second. The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.
SUB_SECOND	This interrupt is signaled to the Interrupt Aggregator at an isochronous rate programmable between 0.5Hz and 32.768KHz. The rate interrupts are signaled is determined by the SPISR field in the Sub-Second Programmable Interrupt Select Register. See Table 24-9, "SPISR Encoding". The interrupt signal is always generated by the Week Timer if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.

24.7 Low Power Modes

The Week Alarm has no low-power modes. It runs continuously while the VBAT well is powered.

24.8 Power-Up Events

The Week Timer POWER_UP_EVENT can be used to power up the system after a timed interval. The POWER_UP_EVENT is routed to the VBAT-Powered Control Interface (VCI). The VCI_OUT pin that is part of the VCI is asserted if the POWER_UP_EVENT is asserted.

The POWER_UP_EVENT can be asserted under the following two conditions:

1. The Week Alarm Counter Register is greater than or equal to the Week Timer Compare Register
2. The Sub-Week Alarm Counter Register decrements from '1' to '0'

The assertion of the POWER_UP_EVENT is inhibited if the POWERUP_EN field in the Control Register is '0'

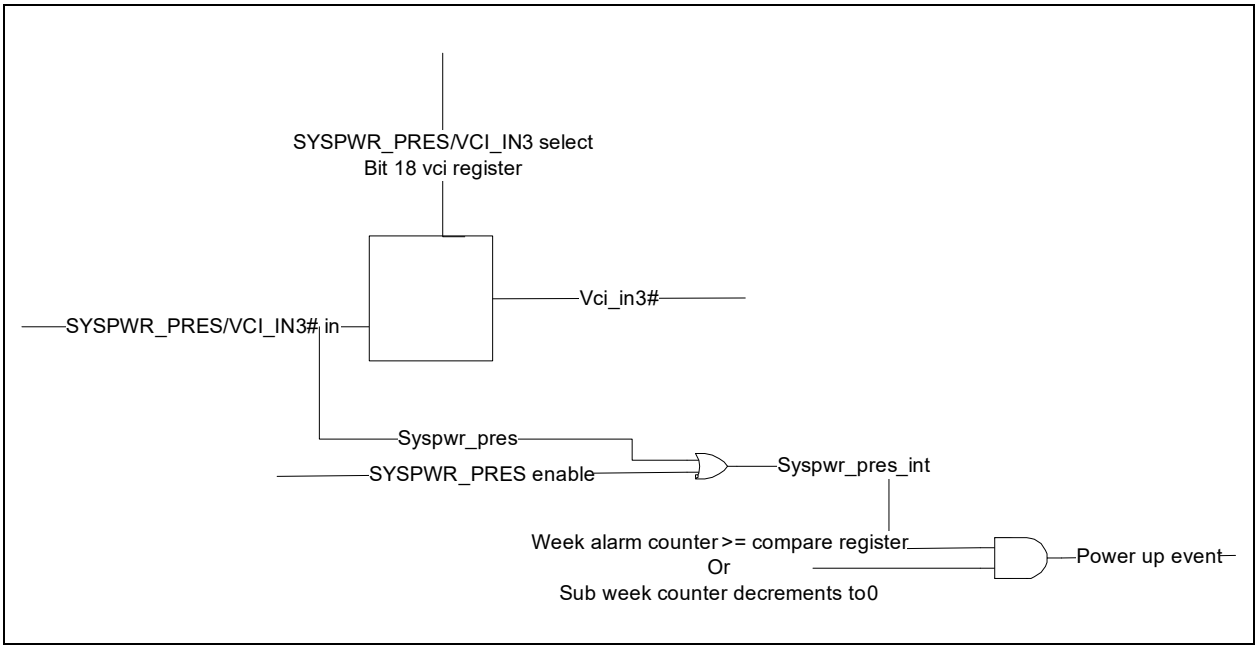
Once a POWER_UP_EVENT is asserted the POWERUP_EN bit must be cleared to reset the output. Clearing POWERUP_EN is necessary to avoid unintended power-up cycles.

24.8.1 SYSPWR_PRES PIN

The SYSPWR_PRES input signal gates the POWER_UP_EVENT. If the SYSPWR_PRES gating function is enabled, the POWER_UP_EVENT is inhibited if the SYSPWR_PRES input is low. The SYSPWR_PRES pin is muxed with VCI_IN3# and the select is in bit 18 of VCI Register.

© 2020 Microchip Technology Inc.

FIGURE 24-2: SYSPWR_PRES SELECT



24.9 Description

The Week Alarm block provides battery-powered timekeeping functions, derived from a low-power 32KHz clock, that operate even when the device's main power is off. The block contains a set of counters that can be used to generate one-shot and periodic interrupts to the EC for periods ranging from about 30 microseconds to over 8 years. The Week Alarm can be used in conjunction with the VBAT-Powered Control Interface to power up a sleeping system after a configurable period.

In addition to basic timekeeping, the Week Alarm block can be used to control the battery-powered general purpose BGPO outputs.

24.9.1 INTERNAL COUNTERS

The Week Timer includes 3 counters:

24.9.1.1 28-bit Week Alarm Counter

This counter is 28 bits wide. The clock for this counter is the overflow of the Clock Divider, and as long as the Week Timer is enabled, it is incremented at a 1 Hz rate.

Both an interrupt and a power-up event can be generated when the contents of this counter matches the contents of the Week Timer Compare Register.

24.9.1.2 9-bit Sub-Week Alarm Counter

This counter is 9 bits wide. It is decremented by 1 at each tick of its selected clock. It can be configured either as a one-shot or repeating event generator.

Both an interrupt and a power-up event can be generated when this counter decrements from 1 to 0.

The Sub-Week Alarm Counter can be configured with a number of different clock sources for its time base, derived from either the Week Alarm Counter or the Clock Divider, by setting the SUBWEEK_TICK field of the Sub-Week Control Register.

© 2020 Microchip Technology Inc.

TABLE 24-7: SUB-WEEK ALARM COUNTER CLOCK

SUBWEEK_ TICK	Source	SPISR	Frequency	Minimum Duration	Maximum Duration
0	Counter Disabled				
1	Sub-Second	0	Counter Disabled		
		1	2 Hz	500 ms	255.5 sec
		2	4 Hz	250 ms	127.8 sec
		3	8 Hz	125 ms	63.9 sec
		4	16 Hz	62.5	31.9 sec
		5	32 Hz	31.25 ms	16.0 sec
		6	64 Hz	15.6 ms	8 sec
		7	128 Hz	7.8 ms	4 sec
		8	256 Hz	3.9 ms	2 sec
		9	512 Hz	1.95 ms	1 sec
		10	1024 Hz	977 µS	499 ms
		11	2048 Hz	488 µS	249.5 ms
		12	4096 Hz	244 µS	124.8 ms
		13	8192 Hz	122 µS	62.4 ms
		14	16.384 KHz	61.1 µS	31.2 ms
		15	32.768 KHz	30.5 µS	15.6 ms
2	Second	n/a	1 Hz	1 sec	511 sec
3	Reserved				
4	Week Counter bit 3	n/a	125 Hz	8 sec	68.1 min
5	Week Counter bit 5	n/a	31.25 Hz	32 sec	272.5 min
6	Week Counter bit 7	n/a	7.8125 Hz	128 sec	18.17 hour
7	Week Counter bit 9	n/a	1.95 Hz	512 sec	72.68 hour

Note 1: The Week Alarm Counter **must not** be modified by firmware if Sub-Week Alarm Counter is using the Week Alarm Counter as its clock source (i.e., the SUBWEEK_TICK field is set to any of the values 4, 5, 6 or 7). The Sub-Week Alarm Counter must be disabled before changing the Week Alarm Counter. For example, the following sequence may be used:

1.Write 0h to the Sub-Week Alarm Counter Register (disabling the Sub-Week Counter)

2.Write the Week Alarm Counter Register

3.Write a new value to the Sub-Week Alarm Counter Register, restarting the Sub-Week Counter

24.9.1.3 15-bit Clock Divider

This counter is 15 bits wide. The clock for this counter is 32KHz, and as long as the Week Timer is enabled, it is incremented at 32.768KHz rate. The Clock Divider automatically The Clock Divider generates a clock out of 1 Hz when the counter wraps from 7FFFh to 0h.

By selecting one of the 15 bits of the counter, using the Sub-Second Programmable Interrupt Select Register, the Clock Divider can be used either to generate a time base for the Sub-Week Alarm Counter or as an isochronous interrupt to the EC, the SUB_SECOND interrupt. See Table 24-9, "SPISR Encoding" for a list of available frequencies.

© 2020 Microchip Technology Inc.

24.9.2 TIMER VALID STATUS

If power on reset occurs on the VBAT power rail while the main device power is off, the counters in the Week Alarm are invalid. If firmware detects a POR on the VBAT power rail after a system boot, by checking the status bits in the Power, Clocks and Resets registers, the Week Alarm block must be reinitialized.

24.9.3 APPLICATION NOTE: REGISTER TIMING

Register writes in the Week Alarm complete within two cycles of the 32KHz clock.The write completes even if the main system clock is stopped before the two cycles of the 32K clock complete. Register reads complete in one cycle of the internal bus clock.

All Week Alarm interrupts that are asserted within the same cycle of the 32KHz clock are synchronously asserted to the EC.

24.9.4 APPLICATION NOTE: USE OF THE WEEK TIMER AS A 43-BIT COUNTER

The Week Timer cannot be directly used as a 42-bit counter that is incremented directly by the 32.768KHz clock domain. The upper 28 bits (28-bit Week Alarm Counter) are incremented at a 1Hz rate and the lower 16 bits (15-bit Clock Divider) are incremented at a 32.768KHz rate, but the increments are not performed in parallel. In particular, the upper 28 bits are incremented when the lower 15 bits increment from 0 to 1, so as long as the Clock Divider Register is 0 the two registers together, treated as a single value, have a smaller value then before the lower register rolled over from 7FFFh to 0h.

The following code can be used to treat the two registers as a single large counter. This example extracts a 32-bit value from the middle of the 43-bit counter:

```
dword TIME_STAMP(void)
{
    AHB_dword wct_value;
    AHB_dword cd_value1;
    AHB_dword cd_value2;
    dword irqEnableSave;

    //Disable interrupts
    irqEnableSave = IRQ_ENABLE;
    IRQ_ENABLE = 0;

    //Read 15-bit clk divider reading register, save result in A
    cd_value1 = WTIMER->CLOCK_DIVIDER;
    //Read 28 bit up-counter timer register, save result in B
    wct_value = WTIMER->WEEK_COUNTER_TIMER;
    //Read 15-bit clk divider reading register, save result in C
    cd_value2 = WTIMER->CLOCK_DIVIDER;

    if (0 == cd_value2)
    {
        wct_value = wct_value + 1;
    }
    else if ( (cd_value2 < cd_value1) || (0 == cd_value1))
    {
        wct_value = WTIMER->WEEK_COUNTER_TIMER;
    }

    //Enable interrupts
    IRQ_ENABLE = irqEnableSave;

    return (WTIMER_BASE + ((wct_value << 10) | (cd_value2>>5)));
}
```

24.10 Battery-Powered General Purpose Outputs

The Week Timer contains the control logic for Battery-Powered General Purposes Outputs (BGPOs). These are output-only pins whose state can be controlled by firmware and preserved when the device is operating on VBAT power alone.

When a BGPO function is selected on a pin that can also serve as a GPIO, using the BGPO Power Register, the GPIO input register is readable and reads the pin value.

24.11 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Week Timer Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 24-8: REGISTER SUMMARY

Offset	Register Name
00h	Control Register
04h	Week Alarm Counter Register
08h	Week Timer Compare Register
0Ch	Clock Divider Register
10h	Sub-Second Programmable Interrupt Select Register
14h	Sub-Week Control Register
18h	Sub-Week Alarm Counter Register
1Ch	BGPO Data Register
20h	BGPO Power Register
24h	BGPO Reset Register

24.11.1 CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6	POWERUP_EN This bit controls the state of the Power-Up Event Output and enables Week POWER-UP Event decoding in the VBAT-Powered Control Interface. See Section 24.8, "Power-Up Events" for a functional description of the POWER-UP_EN bit. 1=Power-Up Event Output Enabled 0=Power-Up Event Output Disabled and Reset	R/W	00h	RESET_VBAT
5:1	Reserved	RES	-	-
0	WT_ENABLE The WT_ENABLE bit is used to start and stop the Week Alarm Counter Register and the Clock Divider Register. The value in the Counter Register is held when the WT_ENABLE bit is not asserted (‘0’) and the count is resumed from the last value when the bit is asserted (‘1’). The 15-Bit Clock Divider is reset to 00h and the Week Alarm Interface is in its lowest power consumption state when the WT_ENABLE bit is not asserted.	R/W	1h	RESET_VBAT

© 2020 Microchip Technology Inc.

24.11.2 WEEK ALARM COUNTER REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	RES	-	-
27:0	WEEK_COUNTER While the WT_ENABLE bit is '1', this register is incremented at a 1 Hz rate. Writes of this register may require one second to take effect. Reads return the current state of the register. Reads and writes complete independently of the state of WT_ENABLE.	R/W	00h	RESET_VBAT

24.11.3 WEEK TIMER COMPARE REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	RES	-	-
27:0	WEEK_COMPARE A Week Alarm Interrupt and a Week Alarm Power-Up Event are asserted when the Week Alarm Counter Register is greater than or equal to the contents of this register. Reads and writes complete independently of the state of WT_ENABLE.	R/W	FFFFFFFh	RESET_VBAT

24.11.4 CLOCK DIVIDER REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:0	CLOCK_DIVIDER Reads of this register return the current state of the Week Timer 15-bit clock divider.	R	-	RESET_VBAT

24.11.5 SUB-SECOND PROGRAMMABLE INTERRUPT SELECT REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	SPISR This field determines the rate at which Sub-Second interrupt events are generated. Table 24-9, "SPISR Encoding" shows the relation between the SPISR encoding and Sub-Second interrupt rate.	R/W	00h	RESET_VBAT

TABLE 24-9: SPISR ENCODING

SPISR Value	Sub-Second Interrupt Rate, Hz	Interrupt Period
0	Interrupts disabled	
1	2	500 ms
2	4	250 ms
3	8	125 ms
4	16	62.5 ms
5	32	31.25 ms
6	64	15.63 ms
7	128	7.813 ms
8	256	3.906 ms
9	512	1.953 ms
10	1024	977 µS
11	2048	488 µS
12	4096	244 µS
13	8192	122 µS
14	16384	61 µS
15	32768	30.5 µS

24.11.6 SUB-WEEK CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:10	Reserved	RES	-	-
9:7	SUBWEEK_TICK This field selects the clock source for the Sub-Week Counter. See Table 24-7, "Sub-Week Alarm Counter Clock" for the description of the options for this field. See also Note 1.	R/W	0	RESET_VBAT
6	AUTO_RELOAD 1= No reload occurs when the Sub-Week Counter expires 0= Reloads the SUBWEEK_COUNTER_LOAD field into the Sub-Week Counter when the counter expires.	R/W	0	RESET_VBAT
5	SYSPWR_PRES_ENABLE This bit controls whether the SYSPWR_PRES input pin has an effect on the POWER_UP_EVENT signal from this block. 1=The POWER_UP_EVENT will only be asserted if the SYSPWR_PRES input is high. If the SYSPWR_PRES input is low, the POWER_UP_EVENT will not be asserted 0=The SYSPWR_PRES input is ignored. It has no effect on the POWER_UP_EVENT	R/W	0	RESET_VBAT
4	SYSPWR_PRES_STATUS This bit provides the current state of the SYSPWR_PRES input pin.	R	-	RESET_VBAT

© 2020 Microchip Technology Inc.

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	WEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Week Alarm Counter Register is greater than or equal the contents of the Week Timer Compare Register and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit does not have to be cleared to remove a Week Timer Power-Up Event.	R/WC	0	RESET_VBAT
0	SUBWEEK_TIMER_POWERUP_EVENT_STATUS This bit is set to '1' when the Sub-Week Alarm Counter Register decrements from '1' to '0' and the POWERUP_EN is '1'. Writes of '1' clear this bit. Writes of '0' have no effect. **Note:** This bit MUST be cleared to remove a Sub-Week Timer Power-Up Event.	R/WC	0	RESET_VBAT

24.11.7 SUB-WEEK ALARM COUNTER REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	Reserved	RES	-	-
24:16	SUBWEEK_COUNTER_STATUS Reads of this register return the current state of the 9-bit Sub-Week Alarm counter.	R	00h	RESET_VBAT
15:9	Reserved	RES	-	-
8:0	SUBWEEK_COUNTER_LOAD Writes with a non-zero value to this field reload the 9-bit Sub-Week Alarm counter. Writes of 0 disable the counter. If the Sub-Week Alarm counter decrements to 0 and the AUTO_RELOAD bit is set, the value in this field is automatically loaded into the Sub-Week Alarm counter.	R/W	00h	RESET_VBAT

24.11.8 BGPO DATA REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2:0	BGPO Battery powered General Purpose Output. Each output pin may be individually configured to be either a VBAT-power BGPO or a VTR-powered GPIO, based on the corresponding settings in the BGPO Power Register. Additionally, each output pin may be individually configured to reset to 0 on either RESET_VBAT or RESET_SYS, based on the corresponding settings in the BGPO Reset Register. For each bit [*i*] in the field: 1=BGPO[*i*] output is high 0=BGPO[*i*] output is low If a BGPO[*i*] does not appear in a package, the corresponding bit **must** be written with a 0 or undesirable results will occur.	R/W	0h	RESET_VBAT or RESET_SYS

24.11.9 BGPO POWER REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2:0	BGPO_POWER Battery powered General Purpose Output power source. For each bit [*i*] in the field: 1=BGPO[*i*] is powered by VBAT. The BGPO[*i*] pin is always determined by the corresponding bit in the BGPO Data Register. The GPIO Input register for the GPIO that is multiplexed with the BGPO always returns a ‘1b’. 0=The pin for BGPO[*i*] functions as a GPIO. When VTR is powered, the pin associated with BGPO[*i*] is determined by the GPIO associated with the pin. When VTR is unpowered, the pin is tri-stated	R/W	1Fh	RESET_VBAT

24.11.10 BGPO RESET REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2:0	BGPO_RESET Battery powered General Purpose Output reset event. For each bit [*i*] in the field: 1=BGPO[*i*] is reset to 0 on RESET_VBAT 0=BGPO[*i*] is reset to 0 on RESET_SYS	R/W	0h	RESET_VBAT

© 2020 Microchip Technology Inc.

25.0 TACH

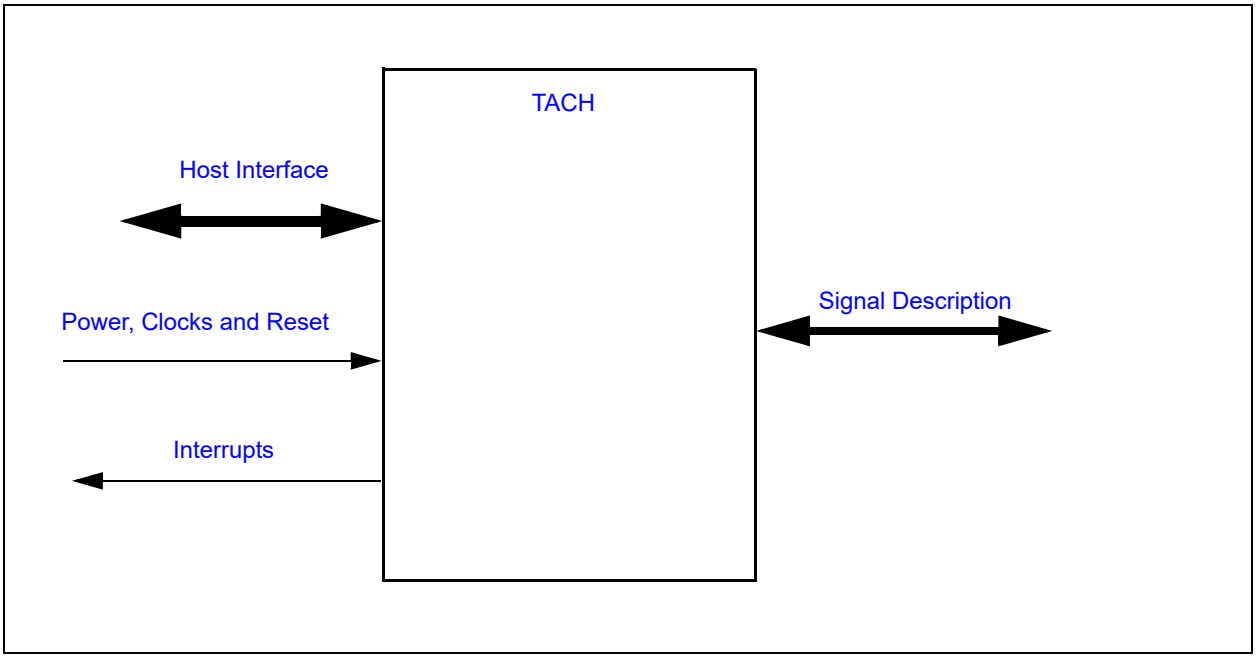
25.1 Introduction

This block monitors TACH output signals (or locked rotor signals) from various types of fans, and determines their speed.

25.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 25-1: I/O DIAGRAM OF BLOCK



25.3 Signal Description

TABLE 25-1: SIGNAL DESCRIPTION

Name	Direction	Description
TACH INPUT	Input	Tachometer signal from TACHx Pin.

25.4 Host Interface

The registers defined for the TACH are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

25.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

25.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

25.5.2 CLOCK INPUTS

Name	Description
100KHz	This is the clock input to the tachometer monitor logic. In Mode 1, the TACH is measured in the number of these clocks. This clock is derived from the main clock domain.

25.5.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

25.6 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 25-2: EC INTERRUPTS

Source	Description
TACH	This internal signal is generated from the OR'd result of the status events, as defined in the TACHx Status Register.

25.7 Low Power Modes

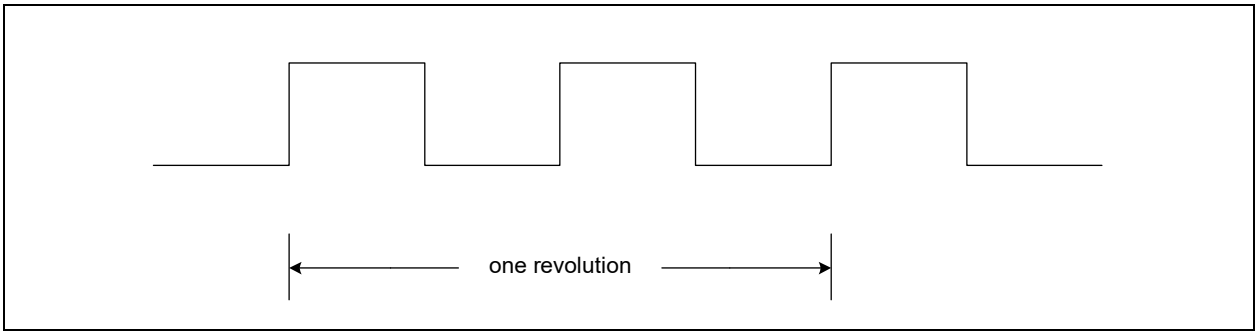
The TACH may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

25.8 Description

The TACH block monitors Tach output signals or locked rotor signals generated by various types of fans. These signals can be used to determine the speed of the attached fan. This block is designed to monitor fans at fan speeds from 100 RPMs to 30,000 RPMs.

Typically, these are DC brushless fans that generate (with each revolution) a 50% duty cycle, two-period square wave, as shown in Figure 25-2 below.

FIGURE 25-2: FAN GENERATED 50%DUTY CYCLE WAVEFORM



In typical systems, the fans are powered by the main power supply. Firmware may disable this block when it detects that the main power rail has been turned off by either clearing the <enable> TACH_ENABLE bit or putting the block to sleep via the supported Low Power Mode interface (see Low Power Modes).

© 2020 Microchip Technology Inc.

25.8.1 MODES OF OPERATION

The Tachometer block supports two modes of operation. The mode of operation is selected via the TACH_READING_-MODE_SELECT bit.

25.8.1.1 Free Running Counter

In Mode 0, the Tachometer block uses the TACH input as the clock source for the internal TACH pulse counter (see TACHX_COUNTER). The counter is incremented when it detects a rising edge on the TACH input. In this mode, the firmware may periodically poll the TACHX_COUNTER field to determine the average speed over a period of time. The firmware must store the previous reading and the current reading to compute the number of pulses detected over a period of time. In this mode, the counter continuously increments until it reaches FFFFh. It then wraps back to 0000h and continues counting. The firmware must ensure that the sample rate is greater than the time it takes for the counter to wrap back to the starting point.

Note:	Tach interrupts should be disabled in Mode 0.

25.8.1.2 Mode 1 -- Number of Clock Pulses per Revolution

In Mode 1, the Tachometer block uses its 100KHz clock input to measure the programmable number of TACH pulses. In this mode, the internal TACH pulse counter (TACHX_COUNTER) returns the value in number of 100KHz pulses per programmed number of TACH_EDGES. For fans that generate two square waves per revolution, these bits should be configured to five edges.

When the number of edges is detected, the counter is latched and the COUNT_READY_STATUS bit is asserted. If the COUNT_READY_INT_EN bit is set a TACH interrupt event will be generated.

25.8.2 OUT-OF-LIMIT EVENTS

The TACH Block has a pair of limit registers that may be configured to generate an event if the Tach indicates that the fan is operating too slow or too fast. If the <TACH reading> exceeds one of the programmed limits, the TACHx High Limit Register and the TACHx Low Limit Register, the bit TACH_OUT_OF_LIMIT_STATUS will be set. If the TACH_OUT_OF_LIMIT_STATUS bit is set, the Tachometer block will generate an interrupt event.

25.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the TACH Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 25-3: REGISTER SUMMARY

Offset	Register Name
00h	TACHx Control Register
04h	TACHx Status Register
08h	TACHx High Limit Register
0Ch	TACHx Low Limit Register

25.9.1 TACHX CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TACHX_COUNTER This 16-bit field contains the latched value of the internal Tach pulse counter, which may be configured by the Tach Reading Mode Select field to operate as a free-running counter or to be gated by the Tach input signal. If the counter is free-running (Mode 0), the internal Tach counter increments (if enabled) on transitions of the raw Tach input signal and is latched into this field every time it is incremented. The act of reading this field will not reset the counter, which rolls over to 0000h after FFFFh. The firmware will compute the delta between the current count reading and the previous count reading, to determine the number of pulses detected over a programmed period. If the counter is gated by the Tach input and clocked by 100KHz (Mode 1), the internal counter will be latched into the reading register when the programmed number of edges is detected or when the counter reaches FFFFh. The internal counter is reset to zero after it is copied into this register. **Note:** In Mode 1, a counter value of FFFFh means that the Tach did not detect the programmed number of edges in 655ms. A stuck fan can be detected by setting the TACHx High Limit Register to a number less than FFFFh. If the internal counter then reaches FFFFh, the reading register will be set to FFFFh and an out-of-limit interrupt can be sent to the EC.	R	00h	RESET_SYS
15	TACH_INPUT_INT_EN 1=Enable Tach Input toggle interrupt from Tach block 0=Disable Tach Input toggle interrupt from Tach block	R/W	0b	RESET_SYS
14	COUNT_READY_INT_EN 1=Enable Count Ready interrupt from Tach block 0=Disable Count Ready interrupt from Tach block	R/W	0b	RESET_SYS
13	Reserved	RES	-	-
12:11	TACH_EDGES A Tach signal is a square wave with a 50% duty cycle. Typically, two Tach periods represents one revolution of the fan. A Tach period consists of three Tach edges. This programmed value represents the number of Tach edges that will be used to determine the interval for which the number of 100KHz pulses will be counted 11b=9 Tach edges (4 Tach periods) 10b=5 Tach edges (2 Tach periods) 01b=3 Tach edges (1 Tach period) 00b=2 Tach edges (1/2 Tach period)	R/W	00b	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TACH_READING_MODE_SELECT 1=Counter is incremented on the rising edge of the 100KHz input. The counter is latched into the TACHX_COUNTER field and reset when the programmed number of edges is detected. 0=Counter is incremented when Tach Input transitions from low-to-high state (default)	R/W	0b	RESET_SYS
9	Reserved	RES	-	-
8	FILTER_ENABLE This filter is used to remove high frequency glitches from Tach Input. When this filter is enabled, Tach input pulses less than two 100KHz periods wide get filtered. 1=Filter enabled 0=Filter disabled (default) It is recommended that the Tach input filter always be enabled.	R/W	0b	RESET_SYS
7:2	Reserved	RES	-	-
1	TACH_ENABLE This bit gates the clocks into the block. When clocks are gated, the TACHx pin is tristated. When re-enabled, the internal counters will continue from the last known state and stale status events may still be pending. Firmware should discard any status or reading values until the reading value has been updated at least one time after the enable bit is set. 1=TACH Monitoring enabled, clocks enabled. 0=TACH Idle, clocks gated	R/W	0b	RESET_SYS
0	TACH_OUT_OF_LIMIT_ENABLE This bit is used to enable the TACH_OUT_OF_LIMIT_STATUS bit in the TACHx Status Register to generate an interrupt event. 1=Enable interrupt output from Tach block 0=Disable interrupt output from Tach block (default)	R/W	0b	RESET_SYS

25.9.2 TACHX STATUS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	COUNT_READY_STATUS This status bit is asserted when the Tach input changes state and when the counter value is latched. This bit remains cleared to '0' when the TACH_READING_MODE_SELECT bit in the TACHx Control Register is '0'. When the TACH_READING_MODE_SELECT bit in the TACHx Control Register is set to '1', this bit is set to '1' when the counter value is latched by the hardware. It is cleared when written with a '1'. If COUNT_READY_INT_EN in the TACHx Control Register is set to 1, this status bit will assert the Tach Interrupt signal. 1=Reading ready 0=Reading not ready	R/WC	0b	RESET_SYS
2	TOGGLE_STATUS This bit is set when Tach Input changes state. It is cleared when written with a '1b'. If TACH_INPUT_INT_EN in the TACHx Control Register is set to '1b', this status bit will assert the Tach Interrupt signal. 1=Tach Input changed state (this bit is set on a low-to-high or high-to-low transition) 0=Tach stable	R/WC	0b	RESET_SYS
1	TACH_PIN_STATUS This bit reflects the state of Tach Input. This bit is a read only bit that may be polled by the embedded controller. 1=Tach Input is high 0=Tach Input is low	R	0b	RESET_SYS
0	TACH_OUT_OF_LIMIT_STATUS This bit is set when the Tach Count value is greater than the high limit or less than the low limit. It is cleared when written with a '1b'. To disable this status event set the limits to their extreme values. If TACH_OUT_OF_LIMIT_ENABLE in the TACHx Control Register is set to 1', this status bit will assert the Tach Interrupt signal. 1=Tach is outside of limits 0=Tach is within limits	R/WC	0b	RESET_SYS

Note:

- Some fans offer a Locked Rotor output pin that generates a level event if a locked rotor is detected. This bit may be used in combination with the Tach pin status bit to detect a locked rotor signal event from a fan.
- Tach Input may come up as active for Locked Rotor events. This would not cause an interrupt event because the pin would not toggle. Firmware must read the status events as part of the initialization process, if polling is not implemented.

© 2020 Microchip Technology Inc.

25.9.3 TACHX HIGH LIMIT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	TACH_HIGH_LIMIT This value is compared with the value in the TACHX_COUNTER field. If the value in the counter is greater than the value programmed in this register, the TACH_OUT_OF_LIMIT_STATUS bit will be set. The TACH_OUT_OF_LIMIT_STATUS status event may be enabled to generate an interrupt to the embedded controller via the TACH_OUT_OF_LIMIT_ENABLE bit in the TACHx Control Register.	R/W	FFFFh	RESET_SYS

25.9.4 TACHX LOW LIMIT REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	TACHX_LOW_LIMIT This value is compared with the value in the TACHX_COUNTER field of the TACHx Control Register. If the value in the counter is less than the value programmed in this register, the TACH_OUT_OF_LIMIT_STATUS bit will be set. The TACH_OUT_OF_LIMIT_STATUS status event may be enabled to generate an interrupt to the embedded controller via the TACH_OUT_OF_LIMIT_ENABLE bit in the TACHx Control Register To disable the TACH_OUT_OF_LIMIT_STATUS low event, program 0000h into this register.	R/W	0000h	RESET_SYS

26.0 PWM

26.1 Introduction

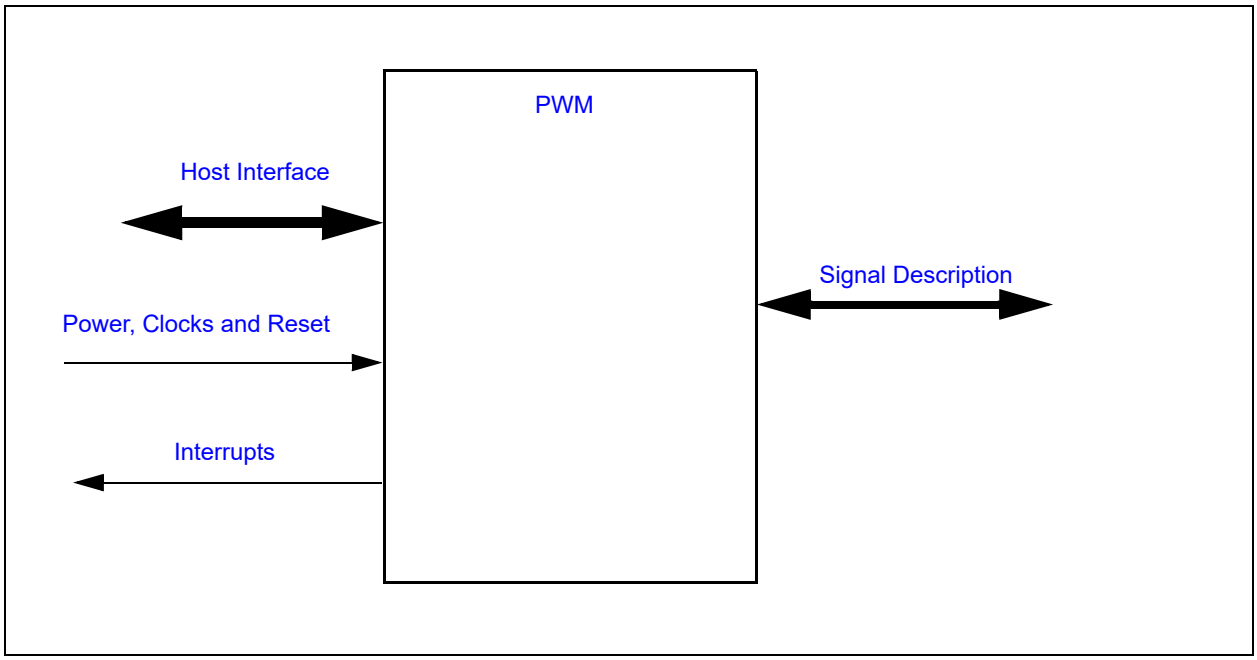
This block generates a PWM output that can be used to control 4-wire fans, blinking LEDs, and other similar devices. Each PWM can generate an arbitrary duty cycle output at frequencies from less than 0.1 Hz to 24 MHz.

The PWMx Counter ON Time registers and PWMx Counter OFF Time registers determine the operation of the PWM_OUTPUT signals. See Section 26.9.1, "PWMx Counter ON Time Register" and Section 26.9.2, "PWMx Counter OFF Time Register" for a description of the PWM_OUTPUT signals.

26.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 26-1: I/O DIAGRAM OF BLOCK



26.3 Signal Description

TABLE 26-1: SIGNAL DESCRIPTION

Name	Direction	Description
PWMx	OUTPUT	Pulse Width Modulated signal to PWMx pin.

26.4 Host Interface

The registers defined for the PWM Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

© 2020 Microchip Technology Inc.

26.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

26.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

26.5.2 CLOCK INPUTS

Name	Description
48MHz	Clock input for generating high PWM frequencies, such as 15 kHz to 30 kHz.
100KHz	This is the clock input for generating low PWM frequencies, such as 10 Hz to 100 Hz.

26.5.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

26.6 Interrupts

The PWM block does not generate any interrupt events.

26.7 Low Power Modes

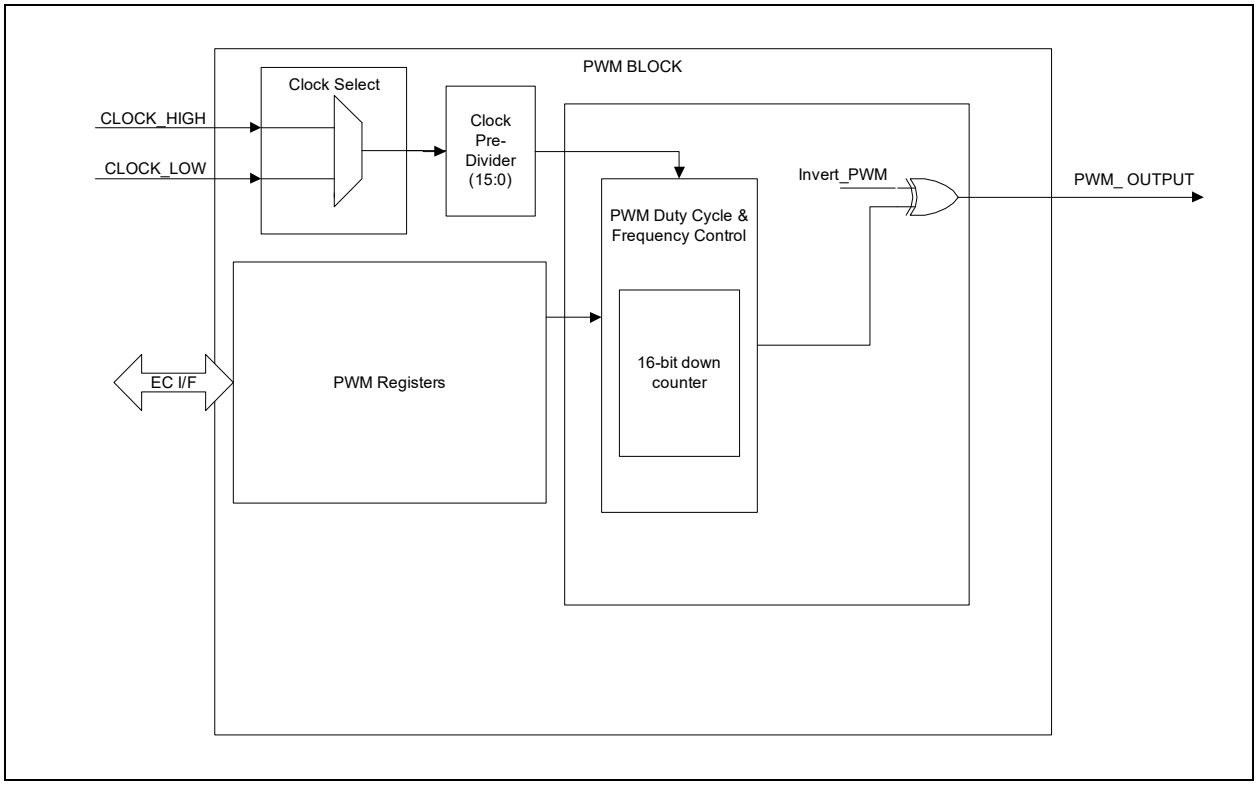
The PWM may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. When the PWM is in the sleep state, the internal counters reset to 0 and the internal state of the PWM and the PWM_OUTPUT signal set to the OFF state.

26.8 Description

The PWM_OUTPUT signal is used to generate a duty cycle of specified frequency. This block can be programmed so that the PWM signal toggles the PWM_OUTPUT, holds it high, or holds it low. When the PWM is configured to toggle, the PWM_OUTPUT alternates from high to low at the rate specified in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register.

The following diagram illustrates how the clock inputs and registers are routed to the PWM Duty Cycle & Frequency Control logic to generate the PWM output.

FIGURE 26-2: BLOCK DIAGRAM OF PWM CONTROLLER



Note: In Figure 26-2, the 48MHz clock is represented as CLOCK_HIGH and the 100KHz clock is represented as CLOCK_LOW.

The PWM clock source to the PWM Down Counter, used to generate a duty cycle and frequency on the PWM, is determined through the Clock select[1] and Clock Pre-Divider[6:3] bits in the PWMx Configuration Register register.

The PWMx Counter ON/OFF Time registers determine both the frequency and duty cycle of the signal generated on PWM_OUTPUT as described below.

The PWM frequency is determined by the selected clock source and the total on and off time programmed in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers. The frequency is the time it takes (at that clock rate) to count down to 0 from the total on and off time.

The PWM duty cycle is determined by the relative values programmed in the PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers.

The PWM Frequency Equation and PWM Duty Cycle Equation are shown below.

EQUATION 26-1: PWM FREQUENCY EQUATION

$$\text{PWM Frequency} = \frac{1}{(PreDivisor + 1)} \times \frac{(ClockSourceFrequency)}{((PWMCounterOnTime + 1) + (PWMCounterOffTime + 1))}$$

In this equation, the ClockSourceFrequency variable is the frequency of the clock source selected by the Clock Select bit in the PWMx Configuration Register, and PreDivisor is a field in the PWMx Configuration Register. The PWMCounterOnTime, PWMCounterOffTime are registers that are defined in Section 26.9, "EC Registers".

© 2020 Microchip Technology Inc.

EQUATION 26-2: PWM DUTY CYCLE EQUATION

$$\text{PWM Duty Cycle} = \frac{(PWMCounterOnTime + 1)}{((PWMCounterOnTime + 1) + (PWMCounterOffTime + 1))}$$

The PWMx Counter ON Time Register and PWMx Counter OFF Time Register registers should be accessed as 16-bit values.

26.8.1 PWM REGISTER UPDATES

The PWMx Counter ON Time Register and PWMx Counter OFF Time Register may be updated at any time. Values written into the two registers are kept in holding registers. The holding registers are transferred into the two user-visible registers when all four bytes have been written with new values and the internal counter completes the OFF time count. If the PWM is in the Full On state then the two user-visible registers are updated from the holding registers as soon as all four bytes have been written. Once the two registers have been updated the holding registers are marked empty. and all four bytes must again be written before the holding registers will be reloaded into the On Time Register and the Off Time Register. Reads of both registers return the current contents of the registers that are used to load the counter and not the holding registers.

26.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the PWM Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 26-2: REGISTER SUMMARY

Offset	Register Name
00h	PWMx Counter ON Time Register
04h	PWMx Counter OFF Time Register
08h	PWMx Configuration Register

26.9.1 PWMX COUNTER ON TIME REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	PWMX_COUNTER_ON_TIME This field determine both the frequency and duty cycle of the PWM signal. Setting this field to a value of *n* will cause the On time of the PWM to be *n*+1 cycles of the PWM Clock Source. When this field is set to zero and the PWMX_COUNTER_OFF_-TIME is not set to zero, the PWM_OUTPUT is held low (Full Off).	R/W	0000h	RESET_SYS

26.9.2 PWMX COUNTER OFF TIME REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	PWMX_COUNTER_OFF_TIME This field determine both the frequency and duty cycle of the PWM signal. Setting this field to a value of *n* will cause the Off time of the PWM to be *n*+1 cycles of the PWM Clock Source. When this field is set to zero, the PWM_OUTPUT is held high (Full On).	R/W	FFFFh	RESET_SYS

26.9.3 PWMX CONFIGURATION REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6:3	CLOCK_PRE_DIVIDER The Clock source for the 16-bit down counter (see PWMx Counter ON Time Register and PWMx Counter OFF Time Register) is determined by bit D1 of this register. The Clock source is then divided by the value of Pre-Divider+1 and the resulting signal determines the rate at which the down counter will be decremented. For example, a Pre-Divider value of 1 divides the input clock by 2 and a value of 2 divides the input clock by 3. A Pre-Divider of 0 will disable the Pre-Divider option.	R/W	0000b	RESET_SYS
2	INVERT 1=PWM_OUTPUT ON State is active low 0=PWM_OUTPUT ON State is active high	R/W	0b	RESET_SYS
1	CLOCK_SELECT This bit determines the clock source used by the PWM duty cycle and frequency control logic. 1=CLOCK_LOW 0=CLOCK_HIGH	R/W	0b	RESET_SYS
0	PWM_ENABLE When the PWM_ENABLE is set to 0 the internal counters are reset and the internal state machine is set to the OFF state. In addition, the PWM_OUTPUT signal is set to the inactive state as determined by the Invert bit. The PWMx Counter ON Time Register and PWMx Counter OFF Time Register are not affected by the PWM_ENABLE bit and may be read and written while the PWM enable bit is 0. 1=Enabled (default) 0=Disabled (gates clocks to save power)	R/W	0b	RESET_SYS

© 2020 Microchip Technology Inc.

27.0 PECI INTERFACE

27.1 Overview

The MEC152x includes a PECI Interface to allow the EC to retrieve temperature readings from PECI-compliant devices. The PECI Interface implements the PHY and Link Layer of a PECI host controller as defined in References[1] and includes hardware support for the PECI 3.1 command set.

This chapter focuses on MEC152x-specific PECI Interface configuration information such as Power Domains, Clock Inputs, Resets, Interrupts, and other chip specific information. For a functional description of the MEC152x PECI Interface refer to References [1].

27.2 References

1. PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, Microchip Confidential, 4/15/11

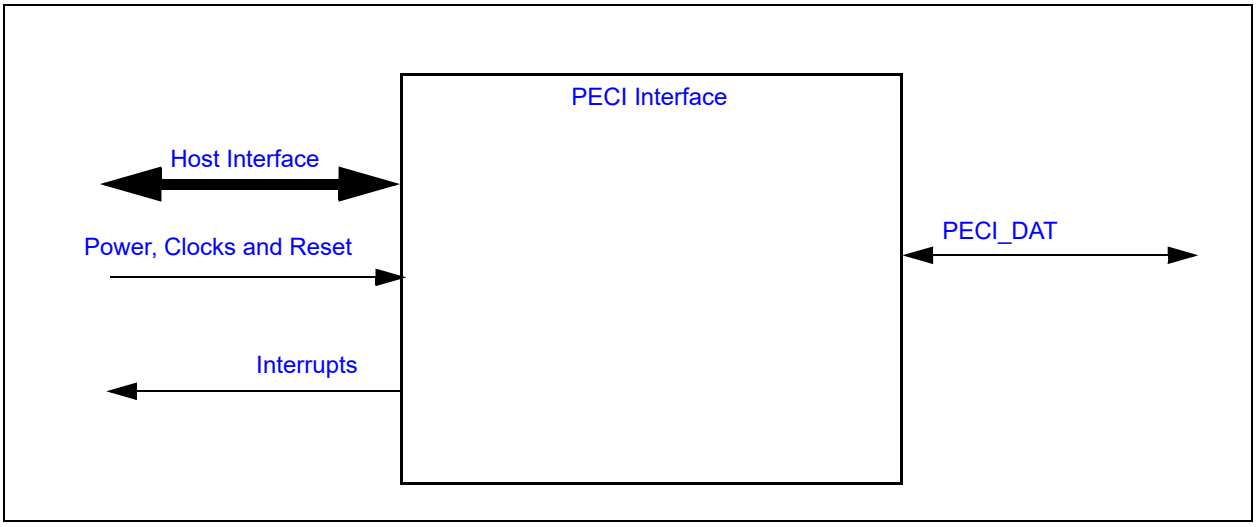
27.3 Terminology

No terminology has been defined for this chapter.

27.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 27-1: PECI INTERFACE I/O DIAGRAM



27.5 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 27-1: SIGNAL DESCRIPTION

Name	Direction	Description
PECI_DAT	Input/Output	PECI Data signal pin

Note: Routing guidelines for the PECI_DAT pin is provided in Intel Platform design guides. Refer to the appropriate Intel document for current information. See Table 27-2.

TABLE 27-2: PECI ROUTING GUIDELINES

Trace Impedance	50 Ohms +/- 15%
Spacing	10 mils
Routing Layer	Microstrip
Trace Width	Calculate to match impedance
Length	1” - 15”

27.6 Host Interface

The registers defined for the PECI Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

27.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

27.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

27.7.2 CLOCK INPUTS

Name	Description
48MHz	This is the main system clock.
PECI_CORE_CLK	This is the PECI_CORE_CLK derived from the 48MHz. This clock divided by the OPTIMAL BIT TIME REGISTER value will generate the bit clock for the PECI_DAT.

27.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.
RST	This is the Soft reset to the PECI block, and resets all the registers and logic to their default state

27.8 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 27-3: EC INTERRUPTS

Source	Description
PECI_INT	PECI Host Event

27.9 Low Power Modes

The PECI Interface may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry.

© 2020 Microchip Technology Inc.

27.10 Instance Description

There is one instance of the PECI Core implemented in the PECI Interface in the MEC152x. See Reference [1], PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, Microchip Confidential, 4/15/11, for a description of the PECI Core.

Note:	If the PECI interface is not in use, the PECI_DISABLE bit in the PECI Disable Register must be set to '1b' in order to minimize leakage current.

27.11 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the PECI Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 27-4: REGISTER SUMMARY

Offset	Register Name
00h	WRITE DATA Register
04h	READ DATA Register
08h	CONTROL Register
0Ch	STATUS Register 1
10h	STATUS Register 2
14h	ERROR Register
18h	Interrupt Enable 1 Register
1Ch	Interrupt Enable 2 Register
20h	Optimal Bit Time Register (LOW BYTE)
24h	Optimal Bit Time Register (HIGH BYTE)
28h	TEST
2Ch	TEST
30h	PECI Baud Control Register
40h	PECI Block ID Register
44h	PECI Revision Register
48h - 7Ch	Test

27.11.1 WRITE DATA REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Write Data. The Write Data Register provides access to a 32-byte Transmit FIFO.The Write Data Register status indicators WFF, WFE,WROV and WRUN can affect the Interrupt.The Transmit FIFO pointers as well as status indicators WFF and WFE are reset to their default values when the FRST bit in the Control Register is asserted.	R/W	00h	RESET_SYS

27.11.2 READ DATA REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Read Data. The Read Data Register provides access to a 32-byte Receive FIFO.The Read Data Register status indicators RFF, RFE, RDOV can affect the Interrupt.The Receive FIFO pointers as well as status indicators RFF and RFE are reset to their default values when the FRST bit in the Control Register is asserted.	R/W	00h	RESET_SYS

27.11.3 CONTROL REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	MIEN Master Interrupt enable	R/W	00h	RESET_SYS
6	TXEN TXEN is the Transmit Enable bit. When TXEN is not asserted (‘0’) Message Transmission is disabled; and data can be queued into the transmit FIFO using the Write Data Register. TXEN is asserted by software and de-asserted by hardware following EOF.	R/W	0	RESET_SYS
5	FRST FRST is the FIFO Reset bit. When FRST is asserted ‘1,’ the Transmit FIFO and the Receive FIFO are reset and the FIFO status indicators in STATUS Register 2 are returned to their default state. The FRST bit is only controlled by software and must be de-asserted by the host before continuing normal operation.	R/W	0	RESET_SYS
4	Reserved	RES	0	
3	RST RST is PECI Core Soft reset. The RST bit must be de-asserted by the host before continuing the normal operation. 0- Normal operation 1- in reset.	R/W	0	RESET_SYS
2:1	Reserved	RES	00	
0	PD Power Down (PD) along with RST controls the entry and exit from the Low Power modes. RST and PD can be asserted at the same time.	R/W	1	RESET_SYS

27.11.4 STATUS REGISTER 1

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	MINT MINT is the Master Interrupt Status bit. MINT is asserted when any of the interrupt status bit is set and de-asserted when all of the interrupt status bits are cleared. The MINT interrupt enable bit (MIEN) is located in the CONTROL Register.	R	0h	RESET_SYS
6	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	ERR ERR Indicates that an error for the current transaction has been detected. This bit will be set when any of the conditions in the ERROR Register are asserted. The ERR bit remains set until the offending condition(s) is eliminated.	R	0	RESET_SYS
1	EOF EOF (End of Frame) is asserted following Message Stop.	R/WC	0	RESET_SYS
0	BOF BOF (Beginning of Frame) is asserted when the PECI Core begins Address Timing Negotiation.	R/WC	0	RESET_SYS

27.11.5 STATUS REGISTER 2

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	IDLE The IDLE status bit indicates when the PECI bus is idle and a new transaction may begin. IDLE is de-asserted during a PECI message transaction and asserted following the Message Setup Time. The host must only initiate PECI transactions when IDLE is asserted. Note that the IDLE status bit does not generate an interrupt.	R	1h	RESET_SYS
6:4	RSVD	R	0	
3	RFE RFE indicates that the Read Data Register FIFO is empty. RFE does not generate an interrupt.	R	1	RESET_SYS
2	RFF RFF indicates that the Read Data Register FIFO is full.	R	0	RESET_SYS
1	WFE WFE indicates that the Write Data Register FIFO is empty.	R	1	RESET_SYS
0	WFF WFF indicates that the Write Data Register FIFO is full. WFF does not generate an interrupt.	R	0	RESET_SYS

27.11.6 ERROR REGISTER

Software handles the bulk of the error recovery process. The different error conditions are captured in this register.

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	CLKERR CLKERR indicates that the READY signal function was de-asserted in the middle of a transaction. In the event of a Clock Error, the PECI Core hardware completes the message normally (EOF) with incorrect data in the Receive FIFO and an FERR.	R/WC	0h	RESET_SYS
6	RDOV Read Overrun, RDOV indicates that the internal read buffer has overflowed. In the event of a Read Buffer Overrun, the PECI Core hardware completes the message normally (EOF) with incorrect data in the Receive FIFO.	R/WC	0	RESET_SYS
5	WRUN Write Underrun, WRUN is set by the PECI Core hardware to indicate that the host did not write data required to be sent over the PECI Bus i.e., the internal write buffer is empty, but data must be sent according to the protocol.	R/WC	0	RESET_SYS
4	WROV Write Overrun, WROV is set by the PECI Core hardware to indicate that the data byte written to the Input register has been ignored since the transmit buffer is full. In this case, the controller takes no action, the data written is ignored, and the transaction continues normally.	R/WC	0	RESET_SYS
3	Reserved	RES	-	
2	Reserved	RES	-	-
1	BERR Bus Error, Bus contention has been detected. BERR is asserted when the PECI host controller reads a value that is different from what it has driven. Following Message Transmit Initiation if the PECI bus is stuck high the BERR is asserted. Note that the BERR bit is the only indication that a PECI Bus Stuck-High Fault has been detected; the IDLE and EOF bits are never re-asserted, TXEN is never de-asserted. A Reset must be asserted after a PECI Bus Stuck-High Fault has been detected.	R/WC	0	RESET_SYS
0	FERR Frame Check Sequence Error occurs when the controller calculates a FCS value that is different from that returned by the target. In the case of a Frame Check Sequence Error the FERR bit is asserted and the transaction continues normally.	R/WC	0	RESET_SYS

© 2020 Microchip Technology Inc.

27.11.7 INTERRUPT ENABLE 1 REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	Reserved	RES	0h	
3	Reserved	RES	0	
2	EREN When the EREN bit is asserted '1' the ERR interrupt is enabled.	R/W	0	RESET_SYS
1	EIEN When the EIEN bit is asserted '1' the EOF interrupt is enabled.	R/W	0	RESET_SYS
0	BIEN When the BIEN bit is asserted '1' the BOF interrupt is enabled.	R/W	0	RESET_SYS

27.11.8 INTERRUPT ENABLE 2 REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:3	Reserved	RES	0h	-
2	ENRFF When the ENRFF bit is asserted '1' the RFF interrupt is enabled.	R/W	0	RESET_SYS
1	ENWFE When the ENWFE bit is asserted '1' the WFE interrupt is enabled.	R/W	0	RESET_SYS
0	Reserved	RES	0	-

27.11.9 OPTIMAL BIT TIME REGISTER

The 16-bit Optimal Bit Time Register determines the 'high' pulse width driven by the host during speed negotiation phases for all PECI transactions. The Optimal Bit Time Register includes the Optimal Bit Time Register (Low Byte) and the Optimal Bit Time Register (High Byte). The 16-bit Optimal Bit Time Register determines PULSE (in seconds) according to the expression.

PULSE = PERIOD * OBT/4

PERIOD is the period of the PECI_CORE_CLK and OBT is the value in the Optimal Bit Time Register,OBT reg value (Hex). The Host Optimal Bit Time (HOBT) (Hz) is determined according to the expression.

HOBT = 1/(PULSE * 4)

The Optimal Bit Time Register (LOW BYTE) and Optimal Bit Time Register (HIGH BYTE) can be written in any order but must not be written while a transaction is in process; i.e., while IDLE in STATUS Register 2 is not asserted ('0').

TABLE 27-5: CORE CLOCK VS OBT REG VALUE FOR MAX. HOBT

Baud Value	**3**	**4**	**1**	**1**	**1**
PECI_CORE_CLK[a] **(MHz)**	**16**	**12**	**48**	**48**	**48**
OBT reg value[b] (Hex)	0x0020	0x0018	0x5DC0	0x0060	0x0030
OBT bit rate	500Kbps	500Kbps	2Kbps	500Kbps	1Mbps

a. The main system clock frequency is 48MHz.

b. Valid OPTIMAL BIT TIME REGISTER values are 0010h - FFFFh.

27.11.9.1 Optimal Bit Time Register (LOW BYTE)

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Optimal Bit Time Register Low byte Valid OBT low values are 10h to FFh	R/W	16h	RESET_SYS

27.11.9.2 Optimal Bit Time Register (HIGH BYTE)

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Optimal Bit Time Register High byte Valid OBT high values are 00h to FFh	R/W	00h	RESET_SYS

27.11.10 PECI BAUD CONTROL REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:4	Reserved	-	-	-
3:0	Baud Value This register is used to divide the main system clock (48MHz) with the Baud Value to generate the PECI_CORE_CLK. By using this register, we can lower the dynamic power of the block.	R/W	0001h	RESET_SYS

27.11.11 PECI BLOCK ID REGISTER

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Block ID Register This register contains the PECI Interface Block ID.	R/W	000000C0h	RESET_SYS

27.11.12 PECI REVISION REGISTER

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	Revision Register This register contains the PECI Interface revision number.	R/W	00000000h	RESET_SYS

© 2020 Microchip Technology Inc.

28.0 HDMI-CEC INTERFACE CONTROLLER

28.1 Overview

This chapter describes an implementation for an HDMI Consumer Electronics Control (CEC) Interface as defined in References [1].

The HDMI-CEC Interface Controller described in this document handles in hardware the transfer of CEC data across the physical interface; including, all initiator/follower data/framing bit timing, logical address decoding, contention detection/Lost Arbitration, Line Error Handling and message block acknowledgment.

Firmware is required to configure physical addressing and the CEC high level protocol, which is beyond the scope of this document.

28.2 References

1. High-Definition Multimedia Interface Specification Version 1.3a, Supplement 1 CEC, November 10, 2006.

Note:	Italicized text in this chapter typically refers to content defined in References [1].

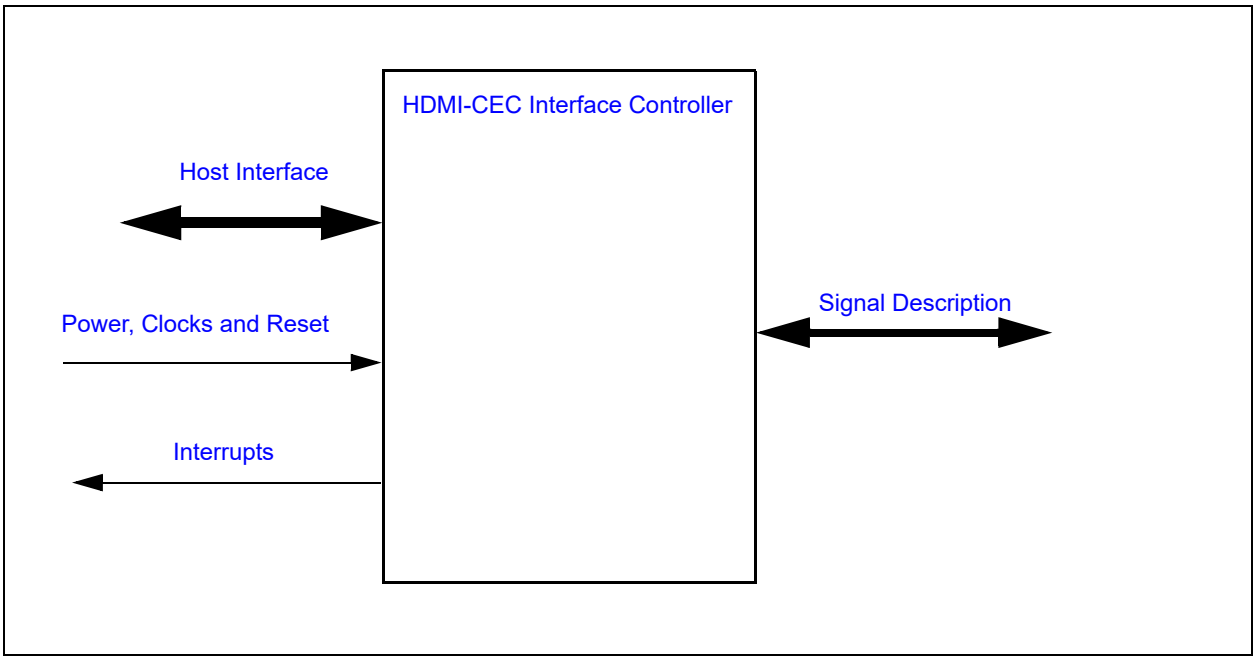
28.3 Terminology

There is no terminology defined for this section.

28.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 28-1: I/O DIAGRAM OF BLOCK



28.5 Signal Description

TABLE 28-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
CEC_IN	INPUT	CEC Bus input signal.
CEC_OUT	OUTPUT	CEC Bus output signal. Note: In MEC152x polarity of the GPIO to which CEC_OUT is mapped should be inverted

28.6 Host Interface

The registers defined for this block are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

28.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

28.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

28.7.2 CLOCK INPUTS

Name	Description
100KHz	The main clock domain, used for CEC bus operations.

28.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.
RESET	This block level reset resets all the CEC state machines and the block registers to their default state.

28.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
CEC_INT	This internal signal is generated when any of the following status bits are enabled and ‘1’: IFDONE, IFE, FFDONE, FDR, FFF

28.9 Low Power Modes

This block may be put into a low power state by the chip’s Power, Clocks, and Reset (PCR) circuitry. The block will only release its clock_required status when the Initiator status is Idle.

28.10 Description

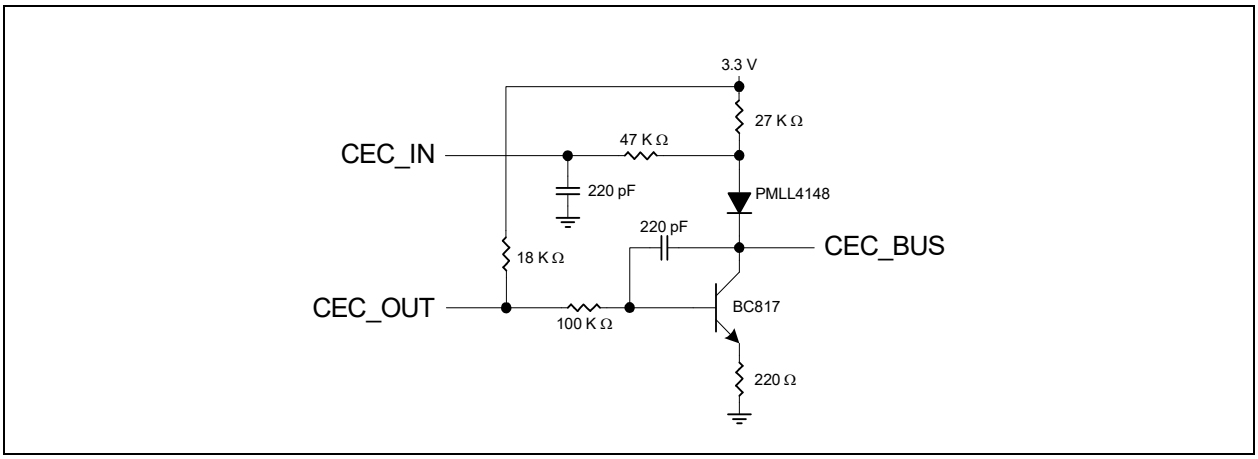
The CEC bus is a single wire bus that provides high-level control functions between all of the various audiovisual products in a user’s environment (see *3, “Overview”* in References [1]). The CEC Bus interface includes two pins, CEC_IN and CEC_OUT. The Controller includes an Initiator Interface and a Follower Interface that function independently within

© 2020 Microchip Technology Inc.

the constraints defined in the subsections that follow. Data transfer buffering in each interface include a 9-bit wide x 16 deep FIFO and a shift register. Each interface can generate interrupts to the EC and the Initiator Interface hardware senses the Bus Idle Condition to initiate message frame transfers.

The following figure illustrates an example of an external CEC Bus interface network. CEC_IN and CEC_OUT pin characteristics, including polarity, direction and buffer type, can be programmed using the GPIO Pin Control facility.

FIGURE 28-2: CEC BUS APPLICATION EXAMPLE



28.10.1 FILTERING

When enabled, the HDMI-CEC Interface Controller filters the CEC_IN pin as defined in Table 28-2. Filtering is enabled using the FLTEN bit in the CEC Control Register.

TABLE 28-2: HDMI-CEC INTERFACE INPUT FILTER

Parameter	Symbol	Value			Units	Notes
		MIN	TYP	MAX		
Filtered Pulse Width	t_{FPW}	100	–	–	ns	

28.10.2 INITIATOR INTERFACE

The Initiator Interface describes data transfers in the HDMI-CEC Interface Controller as an initiator when the ACTIVATE bit is asserted. The Initiator Interface is disabled when the ACTIVATE bit is not asserted.

Specific accesses to the Initiator Data Register, Initiator Control Register and Initiator Status Register are required for successful Message Initiation. The Initiator Interface hardware is also capable of detecting Lost Arbitration.

Note that enforcing CEC message frame size constraints in the HDMI-CEC Interface Controller is the responsibility of firmware because the Initiator Interface can send any number of message blocks in a frame.

28.10.2.1 Message Initiation

Initiator Interface begins a message frame transaction when the START bit in the Initiator Control Register is asserted, the IFE bit is not asserted, and hardware detects a Bus Idle Condition as described in Section 28.10.4, "Bus Idle Condition".

The recommended procedure for Message Initiation is:

1. Assert the IFLUSH in the Initiator Interface FIFO
2. Write all message data to the Initiator Interface FIFO using the Initiator Data Register
3. Assert the START bit

The message frame transaction is complete, or terminated because of an error (see Section 28.10.2.3, "Error Handling"), when the IFDONE bit in the Initiator Status Register is asserted.

To transfer message frames larger than 16 blocks, additional message data can be written to the Initiator Data Register whenever the IFE interrupt is asserted. In this case, data must be written within 20 milliseconds to avoid an underrun error.

28.10.2.2 Lost Arbitration

Lost Arbitration occurs when the Initiator Interface detects bus contention during CEC line arbitration, in which case the LAB bit in the Initiator Status Register is asserted. As a result of Lost Arbitration the transfer is aborted by the Initiator Interface and the task of decoding the rest of the message header is passed to the Follower Interface. If the destination logical address matches one of the addresses configured in the Claimed Logical Address Register, the Follower Interface continues decoding the rest of the message frame.

Following Lost Arbitration, the Initiator Interface firmware is responsible for flushing the Initiator Interface FIFO, and re-initiating the transaction (see also Section 28.10.2.3, "Error Handling").

28.10.2.3 Error Handling

If at any time during an Initiator Interface message transfer the UNDRN, ACKERR, LAB, or CE bits in the Initiator Status Register are asserted, the transfer is aborted by the initiator and the IFDONE bit is asserted. Note that in all cases the transfer is terminated, except during Lost Arbitration as described in Section 28.10.2.2.

Following message frame errors as defined above, the IFLUSH bit in the Initiator Control Register may need to be asserted to delete residual data in the Initiator Interface FIFO.

28.10.3 FOLLOWER INTERFACE

The Follower Interface in the HDMI-CEC Interface Controller responds to data transfers when the ACTIVATE bit is asserted and the Claimed Logical Address Register is greater than zero. The Follower Interface is disabled when the ACTIVATE bit is not asserted; if the ACTIVATE bit is asserted, no message frames will be decoded when the Claimed Logical Address Register is zero.

Specific accesses to the Follower Data Register, Follower Control Register and Follower Status Register are required for successful Follower Message Response. The Follower Interface hardware is also capable of Line Error Handling as described in Section 28.10.3.3. The Follower Interface can respond to frames that contain any number of message blocks.

When the HDMI-CEC Interface Controller is activated the Follower Interface responds to all signaling on the CEC Bus when the Initiator Interface is not active, or following Lost Arbitration. The Follower Interface supports Follower Directed Messages and Follower Broadcast Messages.

When the FFDONE bit in the Follower Status Register is asserted, the Follower Interface has successfully decoded a single message frame and placed it in the Follower Interface FIFO. It is the responsibility of firmware to read the message data using the Follower Data Register before another message arrives. If the Follower Interface tries to write received message data to the FIFO when the FFF bit is asserted, an overrun error occurs (see the definition for OVRN), the current data block is negatively acknowledged, and the message is terminated.

In addition to detecting overrun errors, the Follower Interface implements Line Error Handling for message blocks that cannot be decoded successfully, as described in Section 28.10.3.3, "Line Error Handling".

28.10.3.1 Follower Directed Messages

When the Follower Interface decodes a message header addressed to one of the logical device addresses configured in the Claimed Logical Address Register that are less than 15, the message data blocks are decoded until the EOM bit is detected. These are Follower Directed Messages.

Acknowledge bit handling for Follower Directed Messages is different than for Follower Broadcast Messages (see the definition for ACKERR).

28.10.3.2 Follower Broadcast Messages

When the Follower Interface decodes a message header addressed to logical device address 15, which like directed message addresses is configured in the Claimed Logical Address Register, the message data blocks are decoded until the EOMBERR bit is detected. These are Follower Broadcast Messages.

Acknowledge bit handling for Follower Broadcast Messages is different than for Follower Directed Messages (see the definition for ACKERR).

© 2020 Microchip Technology Inc.

28.10.3.3 Line Error Handling

Whenever the Follower Interface detects spurious pulses on the CEC Bus while decoding a message frame, the BERR bit in the Follower Status Register is asserted and the initiator is notified that a potential error has occurred. Spurious pulses and initiator notification are defined in *CEC 7.4, "CEC Line Error Handling"* in References [1].

Line Error Handling in the HDMI-CEC Interface Controller also includes bus time-out detection, which is indicated by the BTO bit in the Follower Status Register. In all cases, the Follower Interface response to Line Error Handling includes terminating message frame reception.

28.10.4 BUS IDLE CONDITION

The Bus Idle Condition is uniquely determined for each device on the CEC Bus and depends upon the state of the CEC Bus itself and the device system state. In the HDMI-CEC Interface Controller, the Bus Idle Condition is considered 'asserted' when the IDLE bit in the Initiator Status Register is '1,' which occurs according to the signal free time as defined in *CEC 9, "CEC Arbitration"* in References [1].

The start of a message frame by the HDMI-CEC Interface Controller as an initiator (see Section 28.10.2, "Initiator Interface") requires that the Bus Idle Condition be asserted, which is determined by device hardware as described in Table 28-3.

TABLE 28-3: Bus Idle Condition ASSERTION

SFT (Note 1)	FACTOR
3	The previous message was send by the internal initiator and was not successful.
5	RESET (including a hardware reset), or the previous message was send by an external initiator.
7	The previous message was send by the internal initiator and was successful.

Note 1: the Signal Free Time (SFT) is determined by the specified number, or greater, of nominal data bit periods where the CEC Bus remains high following the end of the last message.

For example, if the HDMI-CEC Interface Controller sends a successful message frame (A) it will not be eligible to send a subsequent frame for 16.8 ms (2.4 ms x 7) nominal following the end of frame A. In this same scenario, if an external initiator begins a message frame (B) 12 ms (2.4 ms x 5) after the end of frame A, the internal HDMI-CEC Interface Controller initiator will then be eligible to send a subsequent frame 12 ms after the end of frame B.

In all cases, the IDLE bit is de-asserted in the same bit period as a valid (or invalid) start bit. The IDLE bit does not generate an interrupt.

Immediately following the end of an Initiator Interface message transfer, successful or otherwise, the Bus Idle Condition can be temporarily overridden using the SFT5 bit in the CEC Control Register.

28.11 EC-Only Registers

TABLE 28-4: EC-ONLY REGISTER BASE ADDRESS TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
CEC	0	EC	24-bit internal address space	F0_4000h

TABLE 28-5: POWER, CLOCKS AND RESET VTR-POWERED REGISTERS SUMMARY

Offset	Register Name
00h	CEC Control Register
04h	Claimed Logical Address Register
08h	Initiator Data Register
0Ch	Follower Data Register
10h	Initiator Status Register
12h	Follower Status Register

TABLE 28-5: POWER, CLOCKS AND RESET VTR-POWERED REGISTERS SUMMARY

Offset	Register Name
18h	Initiator Control Register
1Ch	Follower Status Register
20h	Reserved

28.11.1 CEC CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3	SFT5 When this bit is written to ‘1’, the Signal Free Time (SFT) is changed to ‘5’ once only. This bit is cleared and normal SFT behavior resumes when the Bus Idle condition is asserted.	R/W	0h	RESET_SYS
2	FLTEN 1=Filtering on the CEC bus is enabled 0=Filtering is disabled	R/W	0h	RESET_SYS
1	RESET Setting this bit to ‘1’ resets the CEC state machines and the block registers to their default state. This bit is cleared by hardware within one register access cycle. If there is an error condition that requires the CEC controller to be reset, the operation should be completed within the minimum CEC bus idle time in order to avoid unnecessary message NACKs.	R/W	0h	RESET_SYS
0	ACTIVATE 1=The block is enabled for normal operation 0=The block is disabled and placed in its lowest power state	R/W	0h	RESET_SYS

28.11.2 CLAIMED LOGICAL ADDRESS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	CLAR Only one bit in CLAR can be set. 2 Nibbles in Header block correspond to the location of the bit set in Initiator and Follower Logical Address Register Eg. Header value 0x23 correspond to Bit 2 in Initiator CLAR and bit 3 in Follower CLAR Note: CLAR is not a hex value, its one bit in the 16 bit register	R	0h	RESET_SYS

© 2020 Microchip Technology Inc.

28.11.3 INITIATOR DATA REGISTER

The Initiator Data Register is used to write the Initiator Interface FIFO for CEC header and data block values. For 8-bit access cycles, the most significant byte of the Initiator Data Register must be written first. See also *CEC 6.1, "Header/Data Block Description"* in References [1].

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	EOM This bit is used for the End of Message bit in header/data blocks in CEC message frames. When the bit is '0,' more data blocks follow; a '1' specifies that the message is complete. Reads of this field return 0.	W	0h	RESET_SYS
7:0	INITIATOR_DATA Reads of this field return 0.	W	0h	RESET_SYS

28.11.4 FOLLOWER DATA REGISTER

This register is used to read the Follower Interface FIFO for CEC header and data block values. For 8-bit access cycles, the most significant byte of the Follower Data Register must be read first.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	EOM This bit is the received End of Message bit from header/data blocks in CEC message frames. When the bit is '0,' more data blocks follow; a '1' specifies that the message is complete.	R	0h	RESET_SYS
7:0	FOLLOWER_DATA This field is used for the received Information bits from header/data blocks in CEC message frames. Reads of this register when the FFNE bit in the Follower Status Register is not asserted ('0') return undefined data.	R	0h	RESET_SYS

28.11.5 INITIATOR STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:7	Reserved	RES	-	-
6	IFDONE The Initiator Frame Done bit is asserted ('1') when a message block with the EOM bit asserted ('1') has been transferred by the Initiator Interface, or the message frame has been terminated because an error occurred as defined by the LAB, UNDRN, ACKERR, and CE bits.	R/WC	0h	RESET_SYS
5	IFE The Initiator FIFO Empty bit is asserted ('1') when the Initiator Interface retrieves the last data entry from the FIFO.	R	1h	RESET_SYS

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	CE The Contention Error bit is asserted (‘1’) when bus contention is detected by the Initiator Interface at any point following CEC line arbitration. This is likely to be a consequence of CEC Line Error Handling (see *CEC 7.4, “CEC Line Error Handling”* in References [1]).	R/WC	0h	RESET_SYS
3	ACKERR The Acknowledge Error bit is asserted (‘1’) when the follower NACKs an acknowledge bit, both for Follower Directed Messages and Follower Broadcast Messages. For example, in Follower Directed Messages if the follower acknowledge bit response is ‘1,’ this bit is ‘1;’ in Follower Broadcast Messages if a follower acknowledge bit response is ‘0,’ the bit is ‘1.’	R/W	0h	RESET_SYS
2	UNDRN Initiator Underrun. 1=The Initiator shift register requires data from the Initiator Interface FIFO and the IFE bit is asserted 0=No underrun detected	R/WC	0h	RESET_SYS
1	LAB Lost Arbitration. 1=Bus contention is detected by the Initiator Interface during CEC line arbitration 0=No bus contention detected	R/WC	0h	RESET_SYS
0	IDLE 1=The CEC bus is idle 0=The CEC bus is busy	R	1h	RESET_SYS

28.11.6 FOLLOWER STATUS REGISTER

Offset	12h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:7	Reserved	RES	-	-
6	FFDONE The Follower Frame Done bit is asserted (‘1’) when a message block with the EOM bit asserted (‘1’) has been received by the Follower Interface, or the message frame has been terminated because an error occurred as defined by the OVRN, BERR, and BTO bits.	R/WC	0h	RESET_SYS
5	FDR The Follower Data Ready bit is asserted (‘1’) whenever received data is written to the Follower Interface FIFO.	R/WC	0h	RESET_SYS
4	FFF The Follower FIFO Full bit is asserted (‘1’) when there is no room in the Follower Interface FIFO for more data.	R	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	12h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	FFNE The Follower FIFO Not Empty bit is asserted (‘1’) when there is data in the Follower Interface FIFO. The bit is not asserted (‘0’) when the Follower Interface FIFO is empty.	R	0h	RESET_SYS
2	BTO The Bus Time Out Detected bit is asserted (‘1’) when the follower detects that the CEC Bus is held high too long	R/WC	0h	RESET_SYS
1	BERR The Bus Error Detected bit is asserted (‘1’) when the follower detects spurious pulses on the CEC Bus.	R/WC	0h	RESET_SYS
0	OVRN The Follower Overrun bit is asserted (‘1’) when data from the follower shift register is transferred to the Follower Interface FIFO when the FFF bit is asserted.	R/WC	0h	RESET_SYS

28.11.7 INITIATOR CONTROL REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6	IFDONE_EN Enable bit for the IFDONE interrupt. 1=Interrupt is enabled with IFDONE is asserted 0=Interrupt is disabled	R/WC	0h	RESET_SYS
5	IFE_EN Enable bit for the IFE interrupt. 1=Interrupt is enabled with IFE is asserted 0=Interrupt is disabled	R/WC	0h	RESET_SYS
4:2	Reserved	RES	-	-
1	START When this bit is asserted (‘1’), the Initiator Interface begins a message frame transfer if the IFE bit is not asserted. If the Initiator Interface FIFO is empty when this bit is asserted, the message frame transfer begins as soon as a write to the Initiator Data Register occurs. In both cases, the bit is automatically cleared by hardware within one register access cycle. The START bit only needs to be asserted once per message frame.	R/WC	0h	RESET_SYS
0	IFLUSH When this bit is asserted (‘1’), data in the Initiator Interface FIFO is cleared. When set, the bit is automatically cleared by hardware within one register access cycle. This bit can be used to terminate an Initiator Interface message frame transfer.	R/WC	0h	RESET_SYS

28.11.8 FOLLOWER CONTROL REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6	FFDONE_EN Enable bit for the FFDONE interrupt. 1=Interrupt is enabled with FFDONE is asserted 0=Interrupt is disabled	R/WC	0h	RESET_SYS
5	FDR_EN Enable bit for the FDR interrupt. 1=Interrupt is enabled with FDR is asserted 0=Interrupt is disabled	R/WC	0h	RESET_SYS
4	FFF_EN Enable bit for the FFF interrupt. 1=Interrupt is enabled with FFF is asserted 0=Interrupt is disabled	R/WC	0h	RESET_SYS
3:1	Reserved	RES	-	-
0	FFLUSH When this bit is asserted (‘1’), data in the Follower Interface FIFO is cleared. When set, the bit is automatically cleared by hardware within one register access cycle.	R/WC	0h	RESET_SYS

© 2020 Microchip Technology Inc.

29.0 ANALOG TO DIGITAL CONVERTER

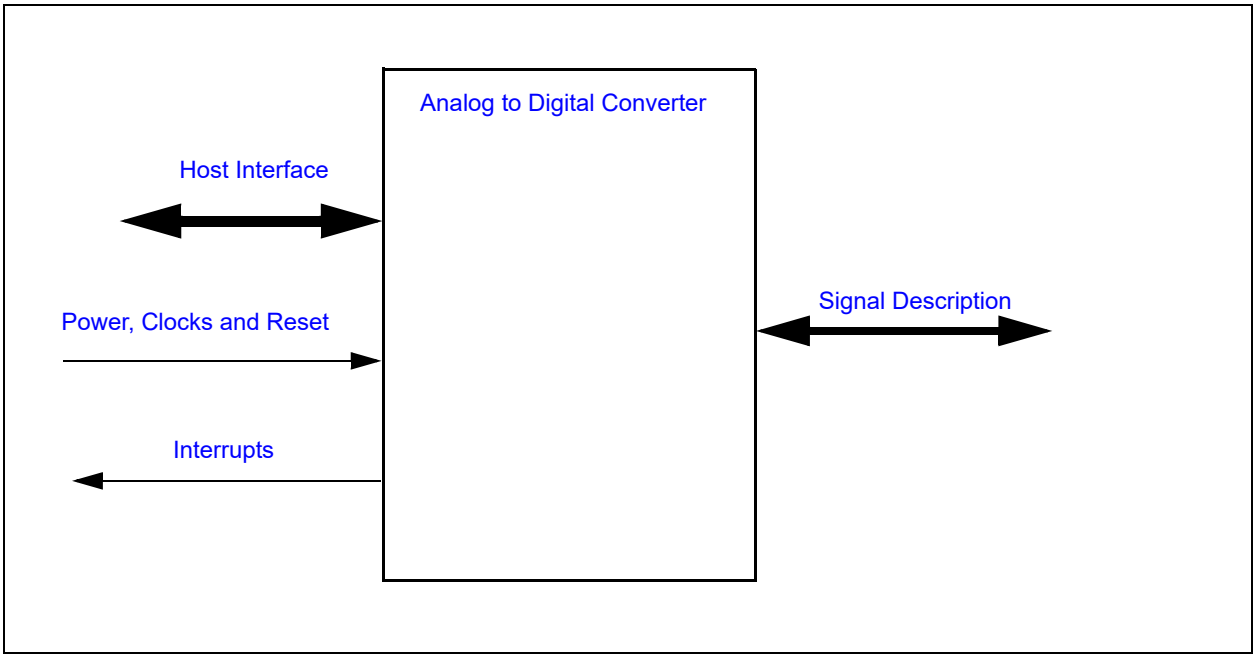
29.1 Introduction

This block is designed to convert external analog voltage readings into digital values. It consists of a single successive-approximation Analog-Digital Converter that can be shared among multiple inputs.

29.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 29-1: I/O DIAGRAM OF BLOCK



29.3 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 29-1: SIGNAL DESCRIPTION

Name	Direction	Description
ADC [7:0]	Input	ADC Analog Voltage Input from pins. **Note:** The ADC IP supports up to 12 channels. The number of channels implemented is package dependent. Refer to the Pin Chapter for the number of channels implemented in a package.
VREF_ADC	Input	ADC Reference Voltage input. ADC Reference Voltage. This pin must either be connected to a very accurate 3.3V reference or connected to the same VTR_ANALOG power supply that is powering the ADC logic.
VREF2_ADC	Input	ADC Reference Voltage input. ADC Reference Voltage can have 2 sources. 1. Internal Reference voltage sourced internal to the chip. This voltage will also be available on a GPIO pin for Thermistor reference voltage 2. External Reference voltage fed through GPIO pin

29.4 Host Interface

The registers defined for the ADC are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

29.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

29.5.1 POWER DOMAINS

TABLE 29-2: POWER SOURCES

Name	Description
VTR_CORE	This power well supplies power for the registers tn this block.
VTR_ANALOG	This power well supplies power for the analog circuitry in this block.

29.5.2 CLOCK INPUTS

TABLE 29-3: CLOCK INPUTS

Name	Description
48MHz	This clock signal is the master clock input to the ADC. This clock is internally divided to generate the ADC sampling clock. At 24MHz, the ADC does one channel conversion in 499.6nS for 12 bit resolution.

29.5.3 RESETS

TABLE 29-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.
SOFT_RESET	This is the Soft reset to the block and resets the Hardware in this block and does not affect the registers.

29.6 Interrupts

TABLE 29-5: EC INTERRUPTS

Source	Description
ADC_Single_Int	Interrupt signal from ADC controller to EC for Single-Sample ADC conversion.
ADC_Repeat_Int	Interrupt signal from ADC controller to EC for Repeated ADC conversion.

29.7 Low Power Modes

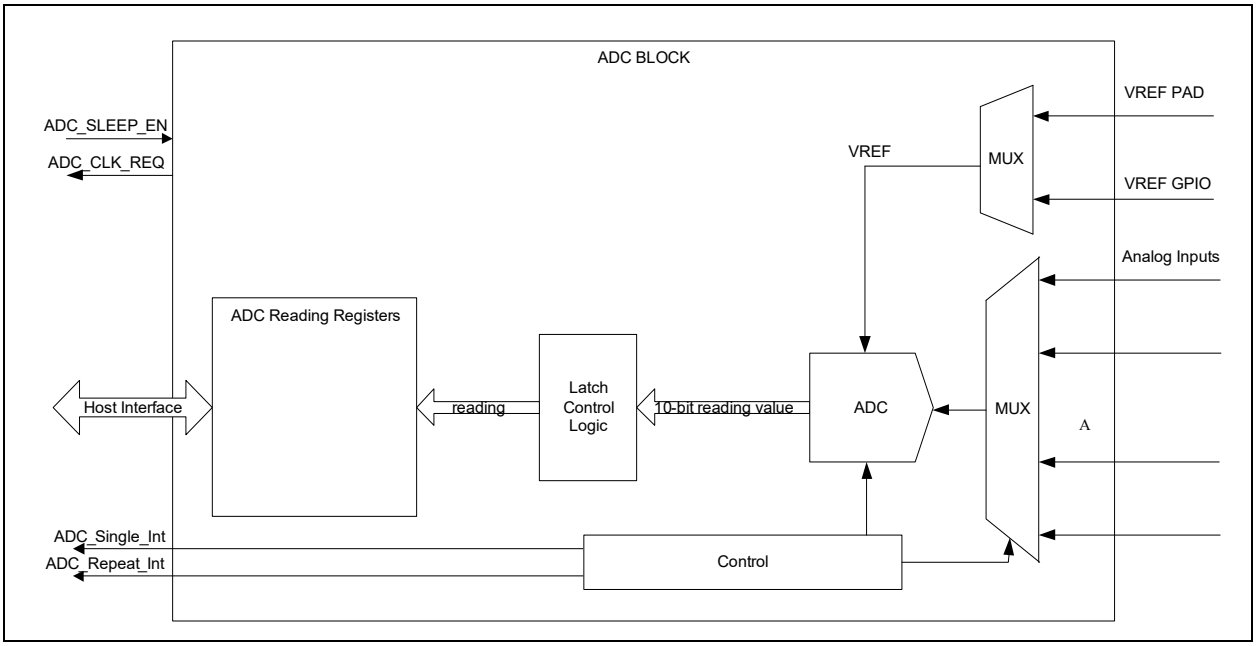
The ADC may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

The ADC is designed to conserve power when it is either sleeping or disabled. It is disabled via the ACTIVATE Bit and sleeps when the ADC_SLEEP_EN signal is asserted. The sleeping state only controls clocking in the ADC and does not power down the analog circuitry. For lowest power consumption, the ADC ACTIVATE bit must be set to '0.'

© 2020 Microchip Technology Inc.

29.8 Description

FIGURE 29-2: ADC BLOCK DIAGRAM



The MEC152x features a twelve channel successive approximation Analog to Digital Converter. The ADC architecture features excellent linearity and converts analog signals to 12 bit words. Conversion takes 499.6 nanoseconds per 12-bit word. The twelve channels are implemented with a single high speed ADC fed by a twelve input analog multiplexer. The multiplexer cycles through the twelve voltage channels, starting with the lowest-numbered channel and proceeding to the highest-number channel, selecting only those channels that are programmed to be active.

The input range on the voltage channels spans from 0V to the voltage reference. With an voltage reference of 3.3V, this provides resolutions of 3.2mV. The range can easily be extended with the aid of resistor dividers. The accuracy of any voltage reading depends on the accuracy and stability of the voltage reference input.

Note:	The ADC pins are 3.3V tolerant.
Note:	Transitions on ADC GPIOs are not permitted when Analog to Digital Converter readings are being taken.
Note:	If GPIO and VREF2_ADC pins are shared and used as a GPIO noise can be injected into the ADC. Hence care should be taken in system design to make sure GPIOs doesn't switch when ADC is active.

The ADC conversion cycle starts either when the START_SINGLE bit in the ADC to set to 1 or when the ADC Repeat Timer counts down to 0. When the START_SINGLE is set to 1 the conversion cycle converts channels enabled by configuration bits in the ADC Single Register. When the Repeat Timer counts down to 0 the conversion cycle converts channels enabled by configuration bits in the ADC Repeat Register. When both the START_SINGLE bit and the Repeat Timer request conversions the START_SINGLE conversion is completed first.

Conversions always start with the lowest-numbered enabled channel and proceed to the highest-numbered enabled channel.

If software repeatedly sets Start_Single to 1 at a rate faster than the Repeat Timer count down interval, the conversion cycle defined by the ADC Repeat Register will not be executed.

29.8.1 REPEAT MODE

- Repeat Mode will start a conversion cycle of all ADC channels enabled by bits RPT_EN in the ADC Repeat Register. The conversion cycle will begin after a delay determined by WARM_UP_DELAY in SAR ADC Control Register and START_DELAY in the ADC Delay Register.Every channel that is enabled will be is converted in 500nS for 12 bit mode and 416.6nS for 10bit mode, for 24MHz internal reference clock. The conversion time formula is **Resolution * Sampling clock time period.**
- After all channels enabled by RPT_EN are complete, REPEAT_DONE_STATUS will be set to 1. The firmware will have to clear the REPEAT_DONE_STATUS bit.
- As long as START_REPEAT is 1 the ADC will repeatedly begin conversion cycles with a period defined by REPEAT_DELAY.
- If the delay period expires and a conversion cycle is already in progress because START_SINGLE was written with a 1, the cycle in progress will complete, followed immediately by a conversion cycle using RPT_EN to control the channel conversions.

29.8.2 SINGLE MODE

- The Single Mode conversion cycle will begin after WARM_UP_DELAY time. After all channels enabled by SINGLE_EN are complete, SINGLE_DONE_STATUS will be set to 1. The firmware will have to clear the SINGLE_DONE_STATUS bit.
- If START_SINGLE is written with a 1, while a conversion cycle is in progress because START_REPEAT is set, the current repeat conversion cycle will complete, followed immediately by a conversion cycle using SINGLE_EN to control the channel conversions.

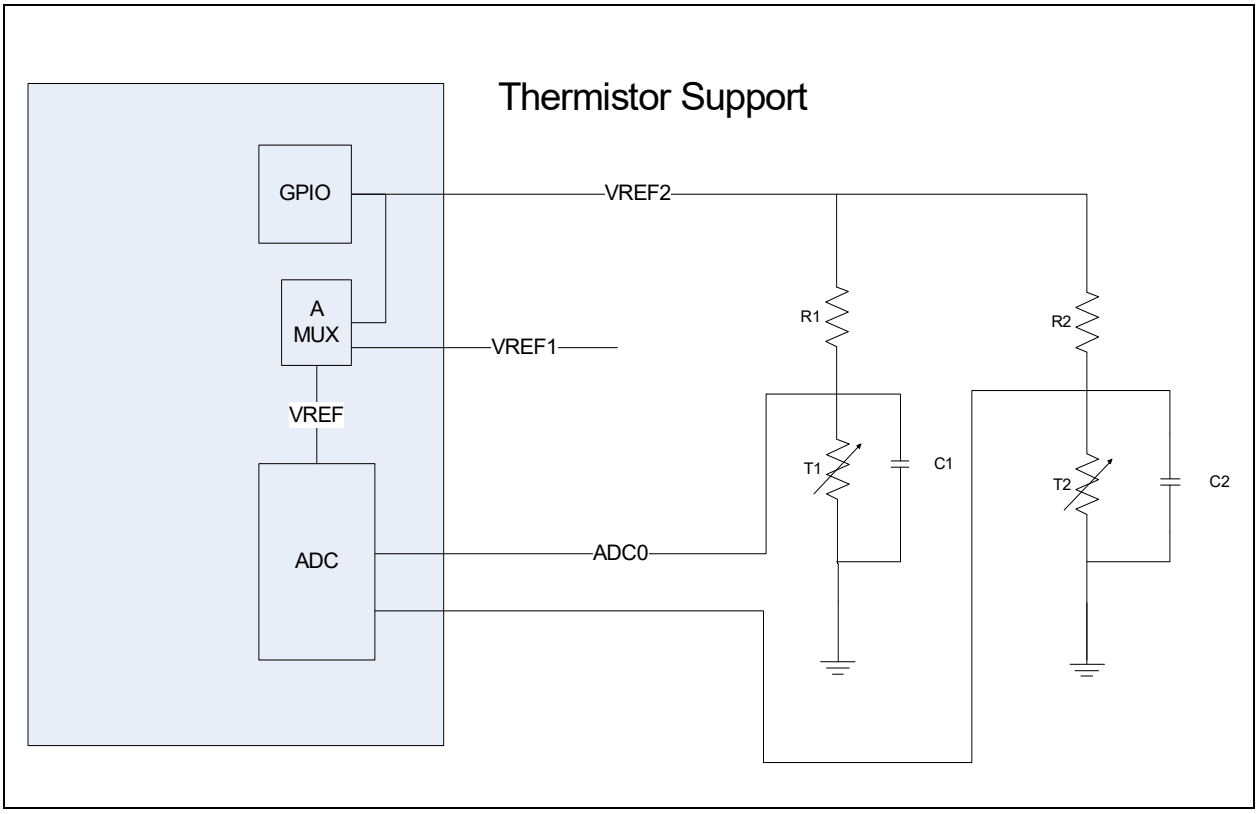
29.8.3 APPLICATION NOTES

Please refer to white paper on “Accurate Temperature measurement using Thermistor” for details on how to use ADC for better than 1 degree C temperature measurement accuracy. Refer to FIGURE 29-3: ADC Reference Voltage Connection on page 357 for details of ADC reference voltage usage.

Note 1:	ADC inputs require at least a 0.1 uF capacitor to filter glitches.
2:	Use 1% Tolerance resistors with the ADC inputs

© 2020 Microchip Technology Inc.

FIGURE 29-3: ADC REFERENCE VOLTAGE CONNECTION



29.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Analog to Digital Converter Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 29-6: REGISTER SUMMARY

Offset	Register Name
00h	ADC Control Register
04h	ADC Delay Register
08h	ADC Status Register
0Ch	ADC Single Register
10h	ADC Repeat Register
14h	ADC Channel Reading Registers 0
18h	ADC Channel Reading Registers 1
1Ch	ADC Channel Reading Registers 2
20h	ADC Channel Reading Registers 3
24h	ADC Channel Reading Registers 4
28h	ADC Channel Reading Registers 5
2Ch	ADC Channel Reading Registers 6
30h	ADC Channel Reading Registers 7
7Ch	ADC Configuration Register

© 2020 Microchip Technology Inc.

TABLE 29-6: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
80h	It is recommended to operate at ADC sampling clock = 24MHz; Do not set the ADC sampling clock to be less than 3MHz.
84h	VREF Control Register
88h	SAR ADC Control Register

29.9.1 ADC CONTROL REGISTER

The ADC Control Register is used to control the behavior of the Analog to Digital Converter.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7	SINGLE_DONE_STATUS This bit is cleared when it is written with a 1. Writing a 0 to this bit has no effect. This bit can be used to generate an EC interrupt. 1= ADC single-sample conversion is completed. This bit is set to 1 when conversion completes for all enabled channels in the single conversion cycle 0= ADC single-sample conversion is not complete. This bit is cleared whenever the software writes a 1b to this bit. **Note:** Only firmware is able to clear SINGLE_DONE_STATUS and REPEAT_DONE_STATUS status bits by writing a 1 to these bits, even when multiple repeat_done or single_-done events occurs before firmware services the interrupt. **Note:** This bit is not self clearing bit.	R/WC	0h	RESET_SYS
6	REPEAT_DONE_STATUS This bit is cleared when it is written with a 1. Writing a 0 to this bit has no effect. This bit can be used to generate an EC interrupt. 1= ADC repeat-sample conversion is completed. This bit is set to 1 when all enabled channels in a repeating conversion cycle complete 0= ADC repeat-sample conversion is not complete.This bit is cleared whenever the software writes to this bit to clear it. **Note:** Only firmware is able to clear SINGLE_DONE_STATUS and REPEAT_DONE_STATUS status bits by writing a 1 to these bits, even when multiple repeat_done or single_-done events occurs before firmware services the interrupt. **Note:** This bit is not self clearing bit.	R/WC	0h	RESET_SYS
5	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	SOFT_RESET 1=writing one causes a reset of the ADC block hardware (not the registers) 0=writing zero takes the ADC block out of reset	R/W	0h	RESET_SYS
3	POWER_SAVER_DIS 1=Power saving feature is disabled **Note:** 0=Power saving feature is enabled. The Analog to Digital Converter controller powers down the ADC between conversion sequences.	R/W	0h	RESET_SYS
2	START_REPEAT 1=The ADC Repeat Mode is enabled. This setting will start a conversion cycle of all ADC channels enabled by bits RPT_EN in the ADC Repeat Register. 0=The ADC Repeat Mode is disabled. Note: This setting will not terminate any conversion cycle in process, but will clear the Repeat Timer and inhibit any further periodic conversions.	R/W	0h	RESET_SYS
1	START_SINGLE 1=The ADC Single Mode is enabled. This setting starts a single conversion cycle of all ADC channels enabled by bits SINGLE_EN in the ADC Single Register. 0=The ADC Single Mode is disabled. This bit is self-clearing	R/W	0h	RESET_SYS
0	ACTIVATE 1=ADC block is enabled for operation. START_SINGLE or START_REPEAT can begin data conversions by the ADC. Note: A reset pulse is sent to the ADC core when this bit changes from 0 to 1. 0=The ADC is disabled and placed in its lowest power state. Note: Any conversion cycle in process will complete before the block is shut down, so that the reading registers will contain valid data but no new conversion cycles will begin.	R/W	0h	RESET_SYS

29.9.2 ADC DELAY REGISTER

The ADC Delay register determines the delay from setting START_REPEAT in the ADC Control Register and the start of a conversion cycle. This register also controls the interval between conversion cycles in repeat mode.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	REPEAT_DELAY This field determines the interval between conversion cycles when START_REPEAT is 1. The delay is in units of 40µs. A value of 0 means no delay between conversion cycles, and a value of 0xFFFF means a delay of 2.6 seconds. This field has no effect when START_SINGLE is written with a 1. **Note 1:** The REPEAT_DELAY is the delay before the start of each successive repeat cycle (not the first cycle. START_DELAY will be used for the first cycle) when the ADC is in low power state and the only after this delay the enable to the actual ADC block is asserted. The delay is also relative to the source being monitored. for eg. for a 10K impedance the recommended REPEAT_DELAY is 0x10 **2:** If the ADC sampling clock is 24MHz, the default REPEAT_DELAY of 0 will work, for lower sampling rate the value will change	R/W	0000h	RESET_SYS
15:0	START_DELAY This field determines the starting delay before a conversion cycle is begun when START_REPEAT is written with a 1. The delay is in units of 40µs. A value of 0 means no delay before the start of a conversion cycle, and a value of 0xFFFF means a delay of 2.6 seconds. This field has no effect when START_SINGLE is written with a 1. **Note:** The START_DELAY is the delay before the start of new repeat cycle when the ADC is disabled and only after this delay the enable to the actual ADC core is asserted.	R/W	0000h	RESET_SYS

29.9.3 ADC STATUS REGISTER

The ADC Status Register indicates whether the ADC has completed a conversion cycle.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	ADC_CH_STATUS All bits are cleared by being written with a '1'. 1=conversion of the corresponding ADC channel is complete 0=conversion of the corresponding ADC channel is not complete For enabled single cycles, the SINGLE_DONE_STATUS bit in the ADC Control Register is also set after all enabled channel conversion are done; for enabled repeat cycles, the REPEAT_DONE_STATUS in the ADC Control Register is also set after all enabled channel conversion are done.	R/WC	00h	RESET_SYS

© 2020 Microchip Technology Inc.

29.9.4 ADC SINGLE REGISTER

The ADC Single Register is used to control which ADC channel is captured during a Single-Sample conversion cycle initiated by the START_SINGLE bit in the ADC Control Register.

Note: Do not change the bits in this register in the middle of a conversion cycle to insure proper operation.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	SINGLE_EN Each bit in this field enables the corresponding ADC channel when a single cycle of conversions is started when the START_SINGLE bit in the ADC Control Register is written with a 1. 1=single cycle conversions for this channel are enabled 0=single cycle conversions for this channel are disabled	R/W	0h	RESET_SYS

29.9.5 ADC REPEAT REGISTER

The ADC Repeat Register is used to control which ADC channels are captured during a repeat conversion cycle initiated by the START_REPEAT bit in the ADC Control Register.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	RPT_EN Each bit in this field enables the corresponding ADC channel for each pass of the Repeated ADC Conversion that is controlled by bit START_REPEAT in the ADC Control Register. 1=repeat conversions for this channel are enabled 0=repeat conversions for this channel are disabled	R/W	00h	RESET_SYS

29.9.6 ADC CHANNEL READING REGISTERS

All 8 ADC channels return their results into a 32-bit reading register. In each case the low 10/12 bits of the reading register return the result of the Analog to Digital conversion and the upper 22/20 bits return 0. Table 29-6, "Register Summary" shows the addresses of all the reading registers.

Note: The ADC Channel Reading Registers access require single 16, or 32 bit reads; i.e., two 8 bit reads will not provide data coherency.

29.9.7 ADC CONFIGURATION REGISTER

Offset	7Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	TEST	R	-	-
15:8	ADC_CLK_HIGH_TIME High Time Count ADC Clock: Programmable from 1 to 255. 0 is not used. **Note:** The High Time Count must be programmed to be equal to the Low Time Count (must be programmed to 50% duty cycle).	R/W	01h	RESET_SYS
7:0	ADC_CLK_LOW_TIME Low Time Count ADC Clock: Programmable from 1 to 255. 0 is not used. The High Time Count must be programmed to be equal to the Low Time Count (must be programmed to 50% duty cycle).	R/W	01h	RESET_SYS

Note: It is recommended to operate at ADC sampling clock = 24MHz; Do not set the ADC sampling clock to be less than 3MHz.

29.9.8 VREF CHANNEL REGISTER

Offset	80h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:22	VREF Select for Channel 11 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
21:20	VREF Select for Channel 10 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
19:18	VREF Select for Channel 9 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
17:16	VREF Select for Channel 8 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	80h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:14	VREF Select for Channel 7 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
13:12	VREF Select for Channel 6 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
11:10	VREF Select for Channel 5 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
9:8	VREF Select for Channel 4 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
7:6	VREF Select for Channel 3 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
5:4	VREF Select for Channel 2 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
3:2	VREF Select for Channel 1 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS
1:0	VREF Select for Channel 0 00 = VREF Pad 01 = VREF GPIO 10 = Reserved 11 = Reserved	R/W	0h	RESET_SYS

29.9.9 VREF CONTROL REGISTER

Offset	84h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:30	VREF Select Status These bits show the VREF selected at this time of reading the register.	R	0h	RESET_SYS
29	VREF_PAD_CTL This is the VREF Pad Control 0 = Leave unused pad floating 1 = Drive unused pad low	R/W	0h	RESET_SYS
28:16	VREF Switch Delay This is the time delay required to switch VREF selects	R/W	0h	RESET_SYS
15:0	VREF Charge Delay This is the time delay required to charge the external VREF capacitor	R/W	0h	RESET_SYS

29.9.10 SAR ADC CONTROL REGISTER

Offset	88h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:17	Reserved	RES	-	-
16:7	WARM_UP_DELAY This is the warm up time delay required for ADC. The delay is in terms of number of ADC clock cycles.	R/W	202h	RESET_SYS
6-4	Reserved	RES	-	-
3	SHIFT_DATA Right justify ADC output data 0 = ADC_DOUT is not shifted and lower bits are 0 1 = ADC_DOUT is shifted right following resolution selection	R/W	0h	RESET_SYS
2:1	SEL_RES These bits define the SAR ADC resolution 00b = Reserved 01b = Reserved 10b = 10 bit resolution 11b = 12 bit resolution	R/W	3h	RESET_SYS
0	SELDIFF This bit define the single ended / differential mode of ADC operation 0 = ADC is enabled for single ended input operation 1 = ADC is enabled for differential mode input operation	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

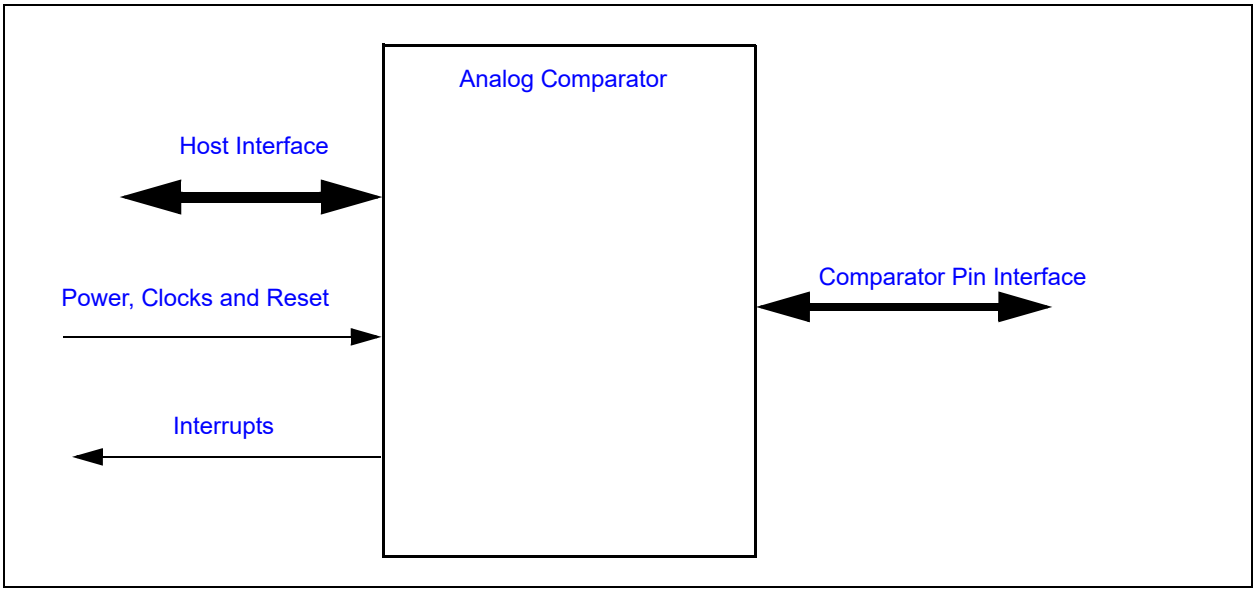
30.0 ANALOG COMPARATOR

30.1 Overview

30.2 The Analog Comparator compares the analog voltage on an input pin to a reference voltage and generates an output that indicates the result of the comparison.**Interface**

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 30-1: I/O DIAGRAM OF BLOCK



30.3 Comparator Pin Interface

TABLE 30-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
CMP_VREF0	Input	Negative voltage input for Comparator 0
CMP_VREF1	Input	Negative voltage input for Comparator 1
CMP_VIN0	Input	Positive voltage input for Comparator 0
CMP_VIN1	Input	Positive voltage input for Comparator 1
CMP_VOUT0	Output	Comparator 0 output
CMP_VOUT1	Output	Comparator 1 output

30.4 Host Interface

The registers defined for the Comparator Interface are only accessible by the embedded controller. The Comparator Registers for both comparators are located in one register in the EC Subsystem register bank. See Section 45.8.11, "Comparator Control Register," on page 515.

30.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

30.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic implemented in this block are powered by this power well.

30.5.2 CLOCK INPUTS

This component does not require a clock input.

30.5.3 RESETS

Name	Description
RESET_VTR	This signal resets all the register in the EC Subsystem that interact with the comparators.

30.6 Interrupts

The comparators do not have a dedicated interrupt output event. An interrupt can be generated by the GPIO which shares the pin with the comparator output signal.

- GPIO124/CMP_VOUT0
- GPIO120/CMP_VOUT1

The GPIO interrupt is configurable, thereby allowing CMP_VOUTx signal to generate an event when the CMP_VINx input is greater than the CMP_VREFx input or when it is less than the CMP_VREFx input. See the definition of Bits[7:4] of the Pin Control Registers on page 260.

30.7 Low Power Modes

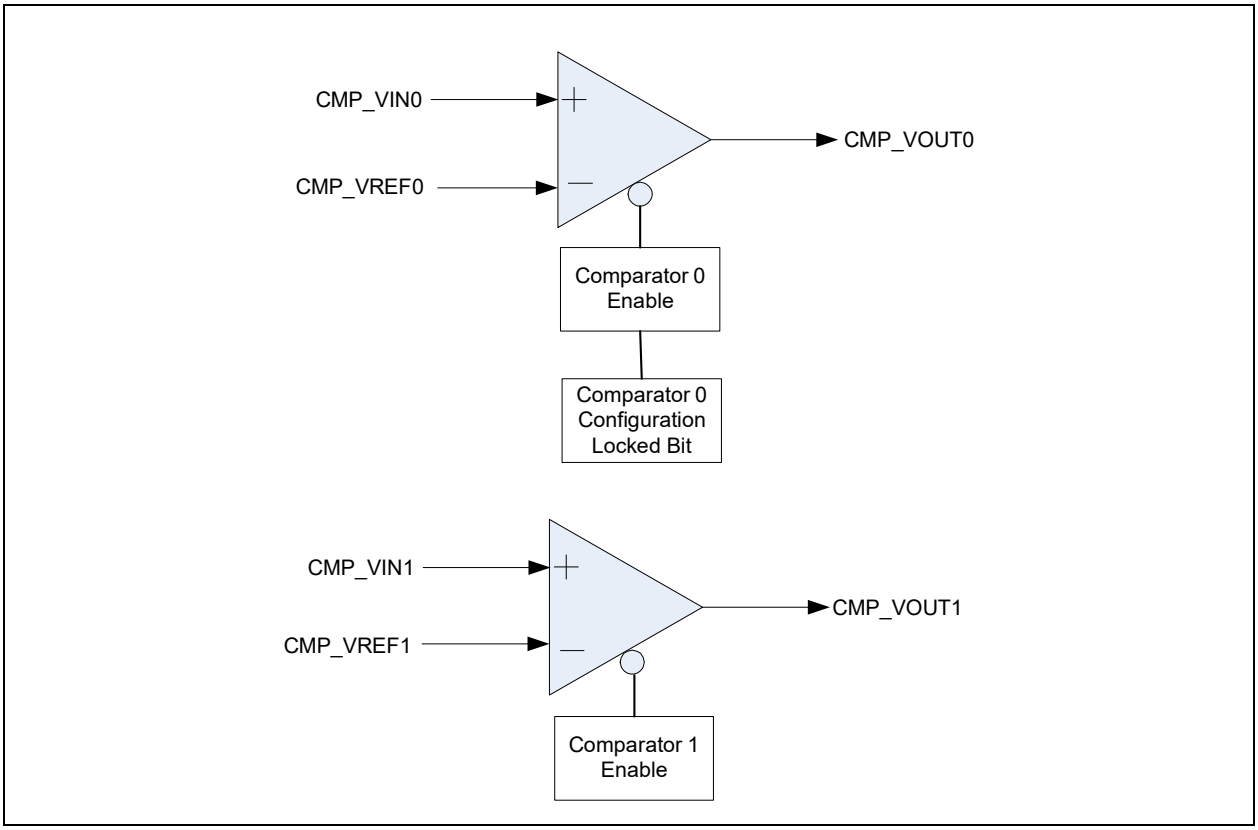
Each comparator is in its lowest powered state when its ENABLE bit is ‘0’.

© 2020 Microchip Technology Inc.

30.8 Description

The Analog Comparator compares the analog voltage on an input pin to a reference voltage and generates an output that indicates the result of the comparison. The reference voltage can be derived either from an external pin or from the internal Digital Analog Converter.

FIGURE 30-2: COMPARATOR BLOCK DIAGRAM



The Analog Comparator compares the analog voltage on the CMP_VINx input pin to a reference voltage and generates an output that indicates the result of the comparison. The reference voltage is derived from the CMP_VREFx input.

The GPIO that shares a pin with the CMP_VOUT signal can be used to generate an interrupt to the EC when the pin multiplexer is configured for CMP_VOUT. The GPIO Pin Control Register is configured for the desired interrupt behavior (level or edge). Changes in the CMP_VOUT output signal will be reflected in the Interrupt Status register field for the GPIO, as configured in the GPIO Pin Control Register.

The control bits for Comparator 0 can be locked. The COMPARATOR 0 ENABLE bit is locked if the LOCK bit for Comparator 0 is set. Once the LOCK bit is set, the COMPARATOR 0 ENABLE cannot be modified until the device is power cycled.

30.9 Comparator Registers

Control and status for both comparators are located in the EC Subsystem register bank. See Section 45.8.11, "Comparator Control Register," on page 515.

31.0 BLINKING/BREATHING LED

31.1 Introduction

LEDs are used in computer applications to communicate internal state information to a user through a minimal interface. Typical applications will cause an LED to blink at different rates to convey different state information. For example, an LED could be full on, full off, blinking at a rate of once a second, or blinking at a rate of once every four seconds, in order to communicate four different states.

As an alternative to blinking, an LED can “breathe”, that is, oscillate between a bright state and a dim state in a continuous, or apparently continuous manner. The rate of breathing, or the level of brightness at the extremes of the oscillation period, can be used to convey state information to the user that may be more informative, or at least more novel, than traditional blinking.

The blinking/breathing hardware is implemented using a PWM. The PWM can be driven either by the Main system clock or by a 32.768 KHz clock input. When driven by the Main system clock, the PWM can be used as a standard 8-bit PWM in order to control a fan. When used to drive blinking or breathing LEDs, the 32.768 KHz clock source is used.

Features:

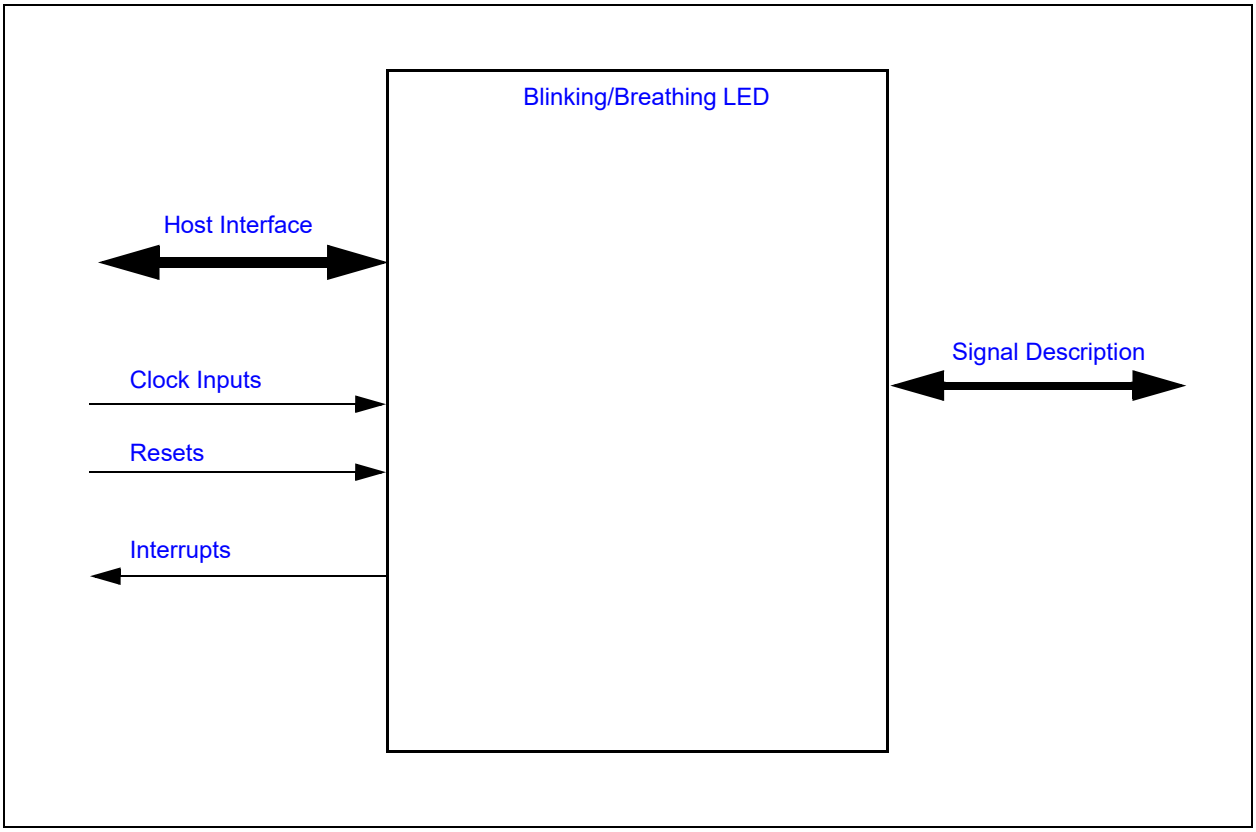
- Each PWM independently configurable
- Each PWM configurable for LED blinking and breathing output
- Highly configurable breathing rate from 60ms to 1min
- Non-linear brightness curves approximated with 8 piece wise-linear segments
- All LED PWMs can be synchronized
- Each PWM configurable for 8-bit PWM support
- Multiple clock rates
- Configurable Watchdog Timer

© 2020 Microchip Technology Inc.

31.2 Interface

This block is designed to drive a pin on the pin interface and to be accessed internally via a registered host interface.

FIGURE 31-1: I/O DIAGRAM OF BLOCK



31.3 Signal Description

Name	Direction	Description
LEDx	Output	PWM LED Output[a] By default, the LEDx pin is configured to be active high: when the LED is configured to be fully on, the pin is driving high. When the LED is configured to be fully off, the pin is low. If firmware requires the Blinking/Breathing PWM to be active low, the Polarity bit in the GPIO Pin Control Register associated with the LED can be set to 1, which inverts the output polarity.

a. Refer to the Table 1-1, "MEC152x Feature List by Package" table to know the number of LED pins available in the chip.

31.4 Host Interface

The blinking/breathing PWM block is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

31.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

31.5.1 POWER DOMAINS

Name	Description
VTR_CORE	Main power. The source of main power for the device is system dependent.

31.5.2 CLOCK INPUTS

Name	Description
32KHz	32.768 KHz clock
48MHz	Main system clock

31.5.3 RESETS

Name	Description
RESET_SYS	This reset signal resets all the logic and register in this block.
RESET	This reset signal, resets the PWM registers to their default values.

31.6 Interrupts

Each PWM can generate an interrupt. The interrupt is asserted for one Main system clock period whenever the PWM WDT times out. The PWM WDT is described in Section 31.8.3.1, "PWM WDT".

Source	Description
PWM_WDT	PWM watchdog time out

31.7 Low Power Mode

The Blinking/Breathing LED may be put into a low power mode by the chip-level power, clocks, and reset (PCR) circuitry. The low power mode is only applicable when the Blinking/Breathing PWM is operating in the General Purpose PWM mode. When the low speed clock mode is selected, the blinking/breathing function continues to operate, even when the 48MHz is stopped. Low power mode behavior is summarized in the following table:

TABLE 31-1: LOW POWER MODE BEHAVIOR

CLOCK_SOURCE	CONTROL	Mode	Low Power Mode	Description
X	‘00’b	PWM ‘OFF’	Yes	32.768 KHz clock is required.
X	‘01’b	Breathing	Yes	
1	‘10’b	General Purpose PWM	No	Main system clock is required, even when a sleep command to the block is asserted.
0	‘10’b	Blinking	Yes	32.768 KHz clock is required.
X	‘11’b	PWM ‘ON’	Yes	

© 2020 Microchip Technology Inc.

Note:	In order for the MEC152x to enter its.Heavy Sleep state, the SLEEP_ENABLE input for all Blinking/Breathing PWM instances must be asserted, even if the PWMs are configured to use the low speed clock.

31.8 Description

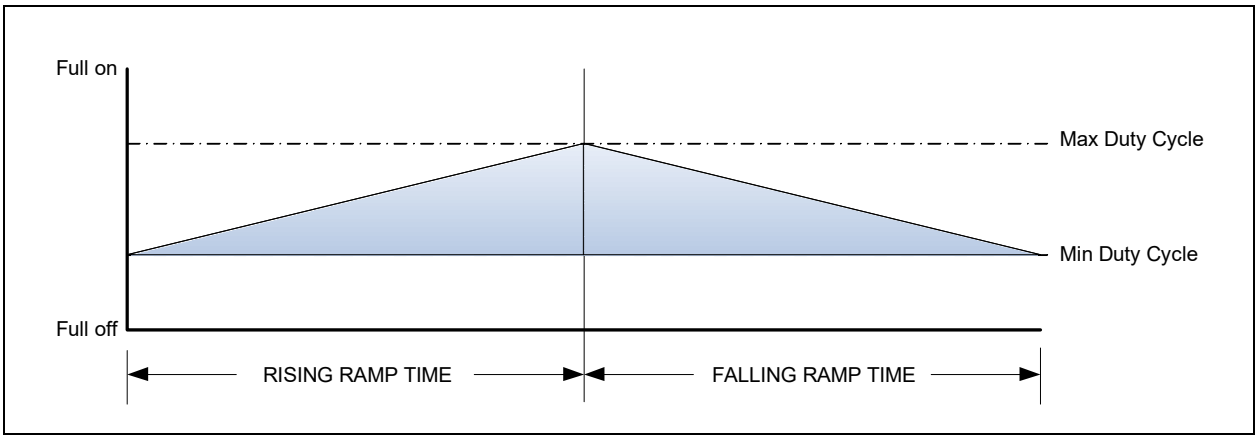
31.8.1 BREATHING

If an LED blinks rapidly enough, the eye will interpret the light as reduced brightness, rather than a blinking pattern. Therefore, if the blinking period is short enough, modifying the duty cycle will set the apparent brightness, rather than a blinking rate. At a blinking rate of 128Hz or greater, almost all people will perceive a continuous light source rather than an intermittent pattern.

Because making an LED appear to breathe is an aesthetic effect, the breathing mechanism must be adjustable or customers may find the breathing effect unattractive. There are several variables that can affect breathing appearance, as described below.

The following figure illustrates some of the variables in breathing:

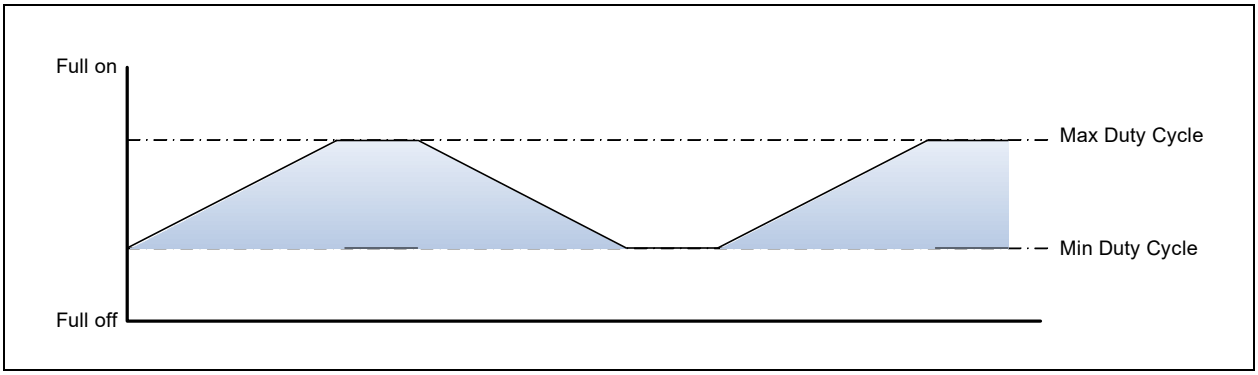
FIGURE 31-2: BREATHING LED EXAMPLE



The breathing range of and LED can range between full on and full off, or in a range that falls within the full-on/full-off range, as shown in this figure. The ramp time can be different in different applications. For example, if the ramp time was 1 second, the LED would appear to breathe quickly. A time of 2 seconds would make the LED appear to breathe more leisurely.

The breathing pattern can be clipped, as shown in the following figure, so that the breathing effect appears to pause at its maximum and minimum brightnesses:

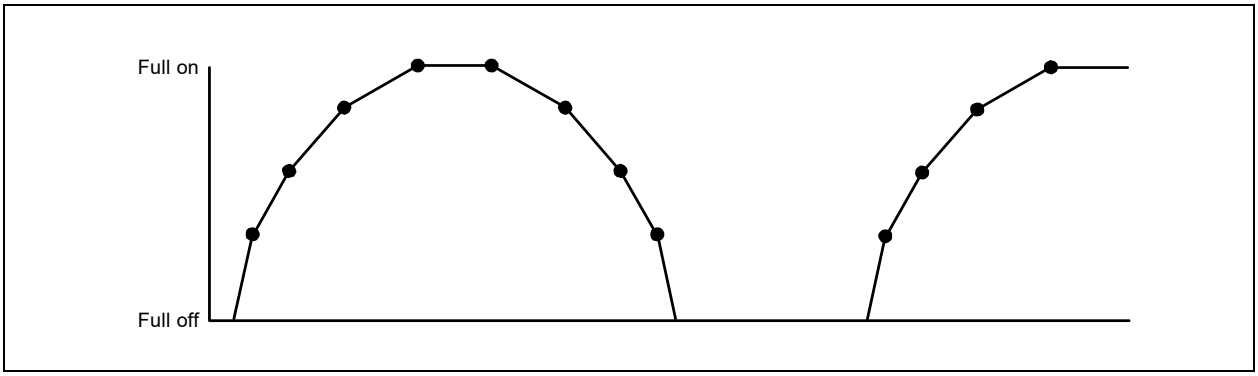
FIGURE 31-3: CLIPPING EXAMPLE



The clipping periods at the two extremes can be adjusted independently, so that for example an LED can appear to breathe (with a short delay at maximum brightness) followed by a longer “resting” period (with a long delay at minimum brightness).

The brightness can also be changed in a non-linear fashion, as shown in the following figure:

FIGURE 31-4: EXAMPLE OF A SEGMENTED CURVE



In this figure, the rise and fall curves are implemented in 4 linear segments and are the rise and fall periods are symmetric.

The breathing mode uses the 32.768 KHz clock for its time base.

31.8.2 BLINKING

When configured for blinking, a subset of the hardware used in breathing is used to implement the blinking function. The PWM (an 8-bit accumulator plus an 8-bit duty cycle register) drives the LED directly. The Duty Cycle register is programmed directly by the user, and not modified further. The PWM accumulator is configured as a simple 8-bit up counter. The counter uses the 32.768 KHz clock, and is pre-scaled by the Delay counter, to slow the PWM down from the 128Hz provided by directly running the PWM on the 32.768 KHz clock.

With the pre-scaler, the blink rate of the LED could be as fast as 128Hz (which, because it is blinking faster than the eye can distinguish, would appear as a continuous level) to 0.03125Hz (that is, with a period of 7.8ms to 32 seconds). Any duty cycle from 0% (0h) to 100% (FFh) can be configured, with an 8-bit precision. An LED with a duty cycle value of 0h will be fully off, while an LED with a duty cycle value of FFh will be fully on.

In Blinking mode the PWM counter is always in 8-bit mode.

Table 31-2, "LED Blink Configuration Examples" shows some example blinking configurations:

TABLE 31-2: LED BLINK CONFIGURATION EXAMPLES

Prescale	Duty Cycle	Blink Frequency	Blink
000h	00h	128Hz	full off
000h	FFh	128Hz	full on
001h	40h	64Hz	3.9ms on, 11.5ms off
003h	80h	32Hz	15.5ms on, 15.5ms off
07Fh	20h	1Hz	125ms on, 0.875s off
0BFh	16h	0.66Hz	125ms on, 1.375s off
0FFh	10h	0.5Hz	125ms on, 1.875s off
180h	0Bh	0.33Hz	129ms on, 2.875s off
1FFh	40h	0.25Hz	1s on, 3s off

© 2020 Microchip Technology Inc.

The Blinking and General Purpose PWM modes share the hardware used in the breathing mode. The Prescale value is derived from the LD field of the LED_DELAY register and the Duty Cycle is derived from the MIN field of the LED_LIMITS register.

TABLE 31-3: BLINKING MODE CALCULATIONS

Parameter	Unit	Equation
Frequency	Hz	(32KHz frequency) /(PRESCALE + 1)/256
'H' Width	Seconds	(1/Frequency) x (DutyCycle/256)
'L' Width	Seconds	(1/Frequency) x ((1-DutyCycle)/256)

31.8.3 GENERAL PURPOSE PWM

When used in the Blinking configuration with the 48MHz, the LED module can be used as a general-purpose programmable Pulse-Width Modulator with an 8-bit programmable pulse width. It can be used for fan speed control, sound volume, etc. With the 48MHz source, the PWM frequency can be configured in the range shown in Table 31-4.

TABLE 31-4: PWM CONFIGURATION EXAMPLES

Prescale	PWM Frequency
000h	187.5 KHz
001h	94 KHz
003h	47 KHz
006h	26.8 KHz
00Bh	15.625 KHz
07Fh	1.46 KHz
1FFh	366 Hz
FFFh	46 Hz

TABLE 31-5: GENERAL PURPOSE PWM MODE CALCULATIONS

Parameter	Unit	Equation
Frequency	Hz	(48MHz frequency) / (PRESCALE + 1) / 256
'H' Width	Seconds	(1/Frequency) x (DutyCycle/256)
'L' Width	Seconds	(1/Frequency) x (256 - DutyCycle)

31.8.3.1 PWM WDT

When the PWM is configured as a general-purpose PWM (in the Blinking configuration with the Main system clock), the PWM includes a Watch Dog Timer (WDT). The WDT consists of an internal 8-bit counter and an 8-bit reload value (the field WDTLD in LED Configuration Register). The internal counter is loaded with the reset value of WDTLD (14h, or 4 seconds) on system RESET_SYS and loaded with the contents of WDTLD whenever either the LED Configuration Register register is written or the MIN byte in the LED Limits Register register is written (the MIN byte controls the duty cycle of the PWM).

Whenever the internal counter is non-zero, it is decremented by 1 for every tick of the 5 Hz clock. If the counter decrements from 1 to 0, a WDT Terminal Count causes an interrupt to be generated and reset sets the CONTROL bit in the LED Configuration Register to 3h, which forces the PWM to be full on. No other PWM registers or fields are affected.

If the 5 Hz clock halts, the watchdog timer stops decrementing but retains its value, provided the device continues to be powered. When the 5 Hz clock restarts, the watchdog counter will continue decrementing where it left off.

Setting the WDTLD bits to 0 disables the PWM WDT. Other sample values for WDTLD are:

01h = 200 ms

02h = 400 ms

03h = 600 ms

04h = 800 ms

...

14h = 4seconds

FFh = 51 seconds

31.9 Implementation

In addition to the registers described in Section 31.10, "EC Registers", the PWM is implemented using a number of components that are interconnected differently when configured for breathing operation and when configured for blinking/PWM operation.

31.9.1 BREATHING CONFIGURATION

The **PSIZE** parameter can configure the PWM to one of three modes: 8-bit, 7-bit and 6-bit. The **PERIOD CTR** counts ticks of its input clock. In 8-bit mode, it counts from 0 to 255 (that is, 256 steps), then repeats continuously. In this mode, a full cycle takes 7.8ms (128Hz). In 7-bit mode it counts from 0 to 127 (128 steps), and a full cycle takes 3.9ms (256Hz). In 6-bit mode it counts from 0 to 63 (64 steps) and a full cycle takes 1.95ms (512Hz).

The output of the LED circuit is asserted whenever the **PERIOD CTR** is less than the contents of the **DUTY CYCLE** register. The appearance of breathing is created by modifying the contents of the **DUTY CYCLE** register in a continuous manner. When the LED control is off the internal counters and registers are all reset to 0 (i.e. after a write setting the RESET bit in the LED Configuration Register Register.) Once enabled, the **DUTY CYCLE** register is increased by an amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. Once the duty cycle reaches its maximum value (determined by the field MAX), the duty cycle is held constant for a period determined by the field HD. Once the hold time is complete, the **DUTY CYCLE** register is decreased, again by an amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. When the duty cycle then falls at or below the minimum value (determined by the field MIN), the duty cycle is held constant for a period determined by the field HD. Once the hold time is complete, the cycle repeats, with the duty cycle oscillating between MIN and MAX.

The rising and falling ramp times as shown in Figure 31-2, "Breathing LED Example" can be either symmetric or asymmetric depending on the setting of the SYMMETRY bit in the LED Configuration Register Register. In Symmetric mode the rising and falling ramp rates have mirror symmetry; both rising and falling ramp rates use the same (all) 8 segments fields in each of the following registers (see Table 31-6): the LED Update Stepsize Register register and the LED Update Interval Register register. In Asymmetric mode the rising ramp rate uses 4 of the 8 segments fields and the falling ramp rate uses the remaining 4 of the 8 segments fields (see Table 31-6).

The parameters MIN, MAX, HD, LD and the 8 fields in LED_STEP and LED_INT determine the brightness range of the LED and the rate at which its brightness changes. See the descriptions of the fields in Section 31.10, "EC Registers", as well as the examples in Section 31.9.3, "Breathing Examples" for information on how to set these fields.

TABLE 31-6: SYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 31-3, "Clipping Example"	Duty Cycle	Segment Index	Symmetric Mode Register Fields Utilized	
X	000xxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
X	001xxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
X	010xxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
X	011xxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
X	100xxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
X	101xxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]
X	110xxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
X	111xxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]
Note: In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]				

© 2020 Microchip Technology Inc.

TABLE 31-7: ASYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in Figure 31-3, "Clipping Example"	Duty Cycle	Segment Index	Asymmetric Mode Register Fields Utilized	
Rising	00xxxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
Rising	01xxxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
Rising	10xxxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
Rising	11xxxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
falling	00xxxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
falling	01xxxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]
falling	10xxxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
falling	11xxxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]
Note: In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 31-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].				

31.9.2 BLINKING CONFIGURATION

The Delay counter and the PWM counter are the same as in the breathing configuration, except in this configuration they are connected differently. The Delay counter is clocked on either the 32.768 KHz clock or the Main system clock, rather than the output of the PWM. The PWM counter is clocked by the zero output of the Delay counter, which functions as a prescalar for the input clocks to the PWM. The Delay counter is reloaded from the LD field of the LED_DELAY register. When the LD field is 0 the input clock is passed directly to the PWM counter without prescaling. In Blinking/PWM mode the PWM counter is always 8-bit, and the PSIZE parameter has no effect.

The frequency of the PWM pulse waveform is determined by the formula:

$$f_{PWM} = \frac{f_{clock}}{(256 \times (LD + 1))}$$

where f_{PWM} is the frequency of the PWM, f_{clock} is the frequency of the input clock (32.768 KHz clock or Main system clock) and LD is the contents of the LD field.

Note:	At a duty cycle value of 00h (in the MIN register), the LED output is fully off. At a duty cycle value of 255h, the LED output is fully on. Alternatively, In order to force the LED to be fully on, firmware can set the CONTROL field of the Configuration register to 3 (always on).

The other registers in the block do not affect the PWM or the LED output in Blinking/PWM mode.

31.9.3 BREATHING EXAMPLES

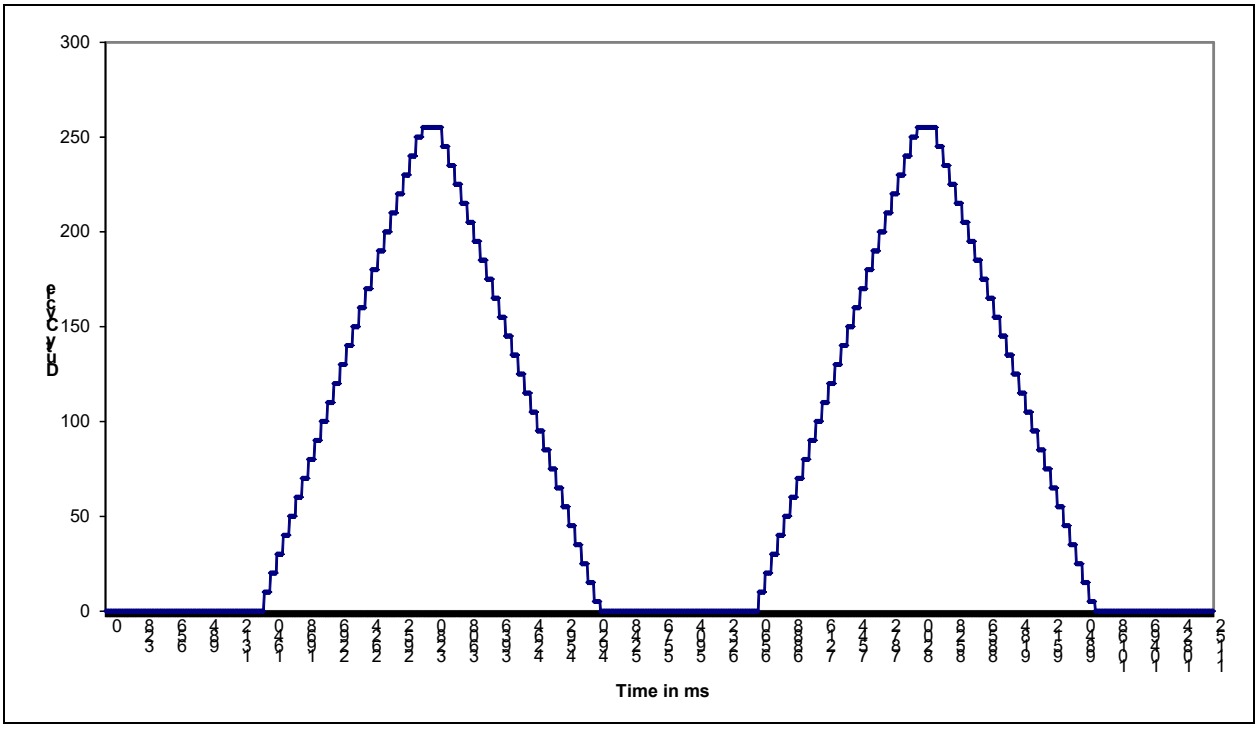
31.9.3.1 Linear LED brightness change

In this example, the brightness of the LED increases and diminishes in a linear fashion. The entire cycle takes 5 seconds. The rise time and fall time are 1.6 seconds, with a hold time at maximum brightness of 200ms and a hold time at minimum brightness of 1.6 seconds. The LED brightness varies between full off and full on. The PWM size is set to 8-bit, so the time unit for adjusting the PWM is approximately 8ms. The registers are configured as follows:

TABLE 31-8: LINEAR EXAMPLE CONFIGURATION

Field	Value							
PSIZE	8-bit							
MAX	255							
MIN	0							
HD	25 ticks (200ms)							
LD	200 ticks (1.6s)							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	8	8	8	8	8	8	8	8
LED_STEP	10	10	10	10	10	10	10	10

FIGURE 31-5: LINEAR BRIGHTNESS CURVE EXAMPLE



31.9.3.2 Non-linear LED brightness change

In this example, the brightness of the LED increases and diminishes in a non-linear fashion. The brightness forms a curve that is approximated by four piece wise-linear line segments. The entire cycle takes about 2.8 seconds. The rise time and fall time are about 1 second, with a hold time at maximum brightness of 320ms and a hold time at minimum brightness of 400ms. The LED brightness varies between full off and full on. The PWM size is set to 7-bit, so the time unit for adjusting the PWM is approximately 4ms. The registers are configured as follows:

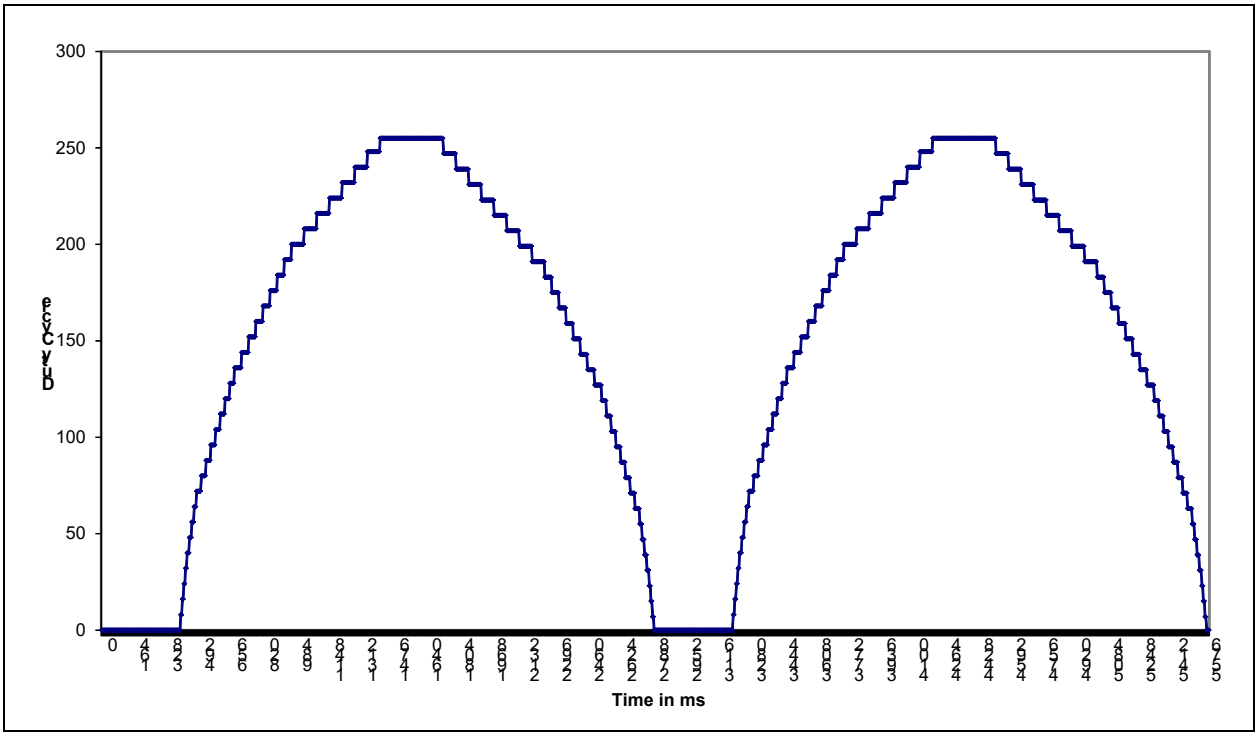
© 2020 Microchip Technology Inc.

TABLE 31-9: NON-LINEAR EXAMPLE CONFIGURATION

Field	Value							
PSIZE	7-bit							
MAX	255 (effectively 127)							
MIN	0							
HD	80 ticks (320ms)							
LD	100 ticks (400ms)							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	2	3	6	6	9	9	16	16
LED_STEP	4	4	4	4	4	4	4	4

The resulting curve is shown in the following figure:

FIGURE 31-6: NON-LINEAR BRIGHTNESS CURVE EXAMPLE



31.10 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Blinking/Breathing LED Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 31-10: REGISTER SUMMARY

Offset	Register Name
00h	LED Configuration Register
04h	LED Limits Register
08h	LED Delay Register

TABLE 31-10: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
0Ch	LED Update Stepsize Register
10h	LED Update Interval Register
14h	LED Output Delay

In the following register definitions, a “PWM period” is defined by time the PWM counter goes from 000h to its maximum value (FFh in 8-bit mode, FEh in 7-bit mode and FCh in 6-bit mode, as defined by the PSCALE field in register LED_CFG). The end of a PWM period occurs when the PWM counter wraps from its maximum value to 0.

The registers in this block can be written 32-bits, 16-bits or 8-bits at a time. Writes to LED Configuration Register take effect immediately. Writes to LED Limits Register are held in a holding register and only take effect only at the end of a PWM period. The update takes place at the end of every period, even if only one byte of the register was updated. This means that in blink/PWM mode, software can change the duty cycle with a single 8-bit write to the MIN field in the LED_LIMIT register. Writes to LED Delay Register, LED Update Stepsize Register and LED Update Interval Register also go initially into a holding register. The holding registers are copied to the operating registers at the end of a PWM period only if the Enable Update bit in the LED Configuration Register is set to 1. If LED_CFG is 0, data in the holding registers is retained but not copied to the operating registers when the PWM period expires. To change an LED breathing configuration, software should write these three registers with the desired values and then set LED_CFG to 1. This mechanism ensures that all parameters affecting LED breathing will be updated consistently, even if the registers are only written 8 bits at a time.

31.10.1 LED CONFIGURATION REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
16	SYMMETRY 1=The rising and falling ramp times are in Asymmetric mode. Table 31-7, "Asymmetric Breathing Mode Register Usage" shows the application of the Stepsize and Interval registers to the four segments of rising duty cycles and the four segments of falling duty cycles. 0=The rising and falling ramp times (as shown in Figure 31-2, "Breathing LED Example") are in Symmetric mode. Table 31-6, "Symmetric Breathing Mode Register Usage" shows the application of the Stepsize and Interval registers to the 8 segments of both rising and falling duty cycles.	R/W	0b	RESET_SYS
15:8	WDT_RELOAD The PWM Watchdog Timer counter reload value. On system reset, it defaults to 14h, which corresponds to a 4 second Watchdog timeout value.	R/W	14h	RESET_SYS
7	RESET Writes of’1’ to this bit resets the PWM registers to their default values. This bit is self clearing. Writes of ‘0’ to this bit have no effect.	W	0b	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6	ENABLE_UPDATE This bit is set to 1 when written with a '1'. Writes of '0' have no effect. Hardware clears this bit to 0 when the breathing configuration registers are updated at the end of a PWM period. The current state of the bit is readable any time. This bit is used to enable consistent configuration of LED_DELAY, LED_STEP and LED_INT. As long as this bit is 0, data written to those three registers is retained in a holding register. When this bit is 1, data in the holding register are copied to the operating registers at the end of a PWM period. When the copy completes, hardware clears this bit to 0.	R/WS	0b	RESET_SYS
5:4	PWM_SIZE This bit controls the behavior of PWM: 3=Reserved 2=PWM is configured as a 6-bit PWM 1=PWM is configured as a 7-bit PWM 0=PWM is configured as an 8-bit PWM	R/W	0b	RESET_SYS
3	SYNCHRONIZE When this bit is '1', all counters for all LEDs are reset to their initial values. When this bit is '0' in the LED Configuration Register for all LEDs, then all counters for LEDs that are configured to blink or breathe will increment or decrement, as required. To synchronize blinking or breathing, the SYNCHRONIZE bit should be set for at least one LED, the control registers for each LED should be set to their required values, then the SYNCHRONIZE bits should all be cleared. If the all LEDs are set for the same blink period, they will all be synchronized.	R/W	0b	RESET_SYS
2	CLOCK_SOURCE This bit controls the base clock for the PWM. It is only valid when CNTRL is set to blink (2). 1=Clock source is the Main system clock 0=Clock source is the 32.768 KHz clock	R/W	0b	RESET_SYS
1:0	CONTROL This bit controls the behavior of PWM: 3=PWM is always on 2=LED blinking (standard PWM) 1=LED breathing configuration 0=PWM is always off. All internal registers and counters are reset to 0. Clocks are gated	R/W	00b 11b	RESET_SYS WDT TC

31.10.2 LED LIMITS REGISTER

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period. The two byte fields may be written independently. Reads of this register return the current contents and not the value of the holding register.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:8	MAXIMUM In breathing mode, when the current duty cycle is greater than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field HD in register LED_DELAY, then starts decrementing the current duty cycle	R/W	0h	RESET_SYS
7:0	MINIMUM In breathing mode, when the current duty cycle is less than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field LD in register LED_DELAY, then starts incrementing the current duty cycle In blinking mode, this field defines the duty cycle of the blink function.	R/W	0h	RESET_SYS

31.10.3 LED DELAY REGISTER

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:12	HIGH_DELAY In breathing mode, the number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MAX in register LED_LIMIT. 4095=The current duty cycle is decremented after 4096 PWM periods … 1=The delay counter is bypassed and the current duty cycle is decremented after two PWM period 0=The delay counter is bypassed and the current duty cycle is decremented after one PWM period	R/W	000h	RESET_SYS
11:0	LOW_DELAY The number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MIN in register LED_LIMIT. 4095=The current duty cycle is incremented after 4096 PWM periods … 0=The delay counter is bypassed and the current duty cycle is incremented after one PWM period In blinking mode, this field defines the prescalar for the PWM clock	R/W	000h	RESET_SYS

© 2020 Microchip Technology Inc.

31.10.4 LED UPDATE STEPSIZE REGISTER

This register has eight segment fields which provide the amount the current duty cycle is adjusted at the end of every PWM period. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the SYMMETRY bit in the LED Configuration Register Register)

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 31-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

In 8-bit mode, each 4-bit STEPSIZE field represents 16 possible duty cycle modifications, from 1 to 16 as the duty cycle is modified between 0 and 255:

15: Modify the duty cycle by 16

...

1: Modify the duty cycle by 2

0=Modify the duty cycle by 1

In 7-bit mode, the least significant bit of the 4-bit field is ignored, so each field represents 8 possible duty cycle modifications, from 1 to 8, as the duty cycle is modified between 0 and 127:

14, 15: Modify the duty cycle by 8

...

2, 3: Modify the duty cycle by 2

0, 1: Modify the duty cycle by 1

In 6-bit mode, the two least significant bits of the 4-bit field is ignored, so each field represents 4 possible duty cycle modifications, from 1 to 4 as the duty cycle is modified between 0 and 63:

12, 13, 14, 15: Modify the duty cycle by 4

8, 9, 10, 11: Modify the duty cycle by 3

4, 5, 6, 7: Modify the duty cycle by 2

0, 1, 2, 3: Modify the duty cycle by 1

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	UPDATE_STEP7 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 111.	R/W	0h	RESET_SYS
27:24	UPDATE_STEP6 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 110.	R/W	0h	RESET_SYS
23:20	UPDATE_STEP5 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 101	R/W	0h	RESET_SYS
19:16	UPDATE_STEP4 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 100.	R/W	0h	RESET_SYS
15:12	UPDATE_STEP3 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 011.	R/W	0h	RESET_SYS

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
11:8	UPDATE_STEP2 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 010.	R/W	0h	RESET_SYS
7:4	UPDATE_STEP1 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 001.	R/W	0h	RESET_SYS
3:0	UPDATE_STEP0 Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 000.	R/W	0h	RESET_SYS

31.10.5 LED UPDATE INTERVAL REGISTER

This register has eight segment fields which provide the number of PWM periods between updates to current duty cycle. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the SYMMETRY bit in the LED Configuration Register Register)

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in Figure 31-3, "Clipping Example") and Segment_Index[1:0] = Duty Cycle Bits[7:6].

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	UPDATE_INTERVAL7 The number of PWM periods between updates to current duty cycle when the segment index is equal to 111b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
27:24	UPDATE_INTERVAL6 The number of PWM periods between updates to current duty cycle when the segment index is equal to 110b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
23:20	UPDATE_INTERVAL5 The number of PWM periods between updates to current duty cycle when the segment index is equal to 101b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
19:16	UPDATE_INTERVAL4 The number of PWM periods between updates to current duty cycle when the segment index is equal to 100b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
15:12	UPDATE_INTERVAL3 The number of PWM periods between updates to current duty cycle when the segment index is equal to 011b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
11:8	UPDATE_INTERVAL2 The number of PWM periods between updates to current duty cycle when the segment index is equal to 010b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
7:4	UPDATE_INTERVAL1 The number of PWM periods between updates to current duty cycle when the segment index is equal to 001b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS
3:0	UPDATE_INTERVAL0 The number of PWM periods between updates to current duty cycle when the segment index is equal to 000b. 15=Wait 16 PWM periods … 0=Wait 1 PWM period	R/W	0h	RESET_SYS

31.10.6 LED OUTPUT DELAY

This register permits the transitions for multiple blinking/breathing LED outputs to be skewed, so as not to present too great a current load. The register defines a count for the number of clocks the circuitry waits before turning on the output, either on initial enable, after a resume from Sleep, or when multiple outputs are synchronized through the Sync control in the LED CONFIGURATION (LED_CFG) register.

When more than one LED outputs are used simultaneously, the LED OUTPUT DELAY fields of each should be configured with different values so that the outputs are skewed. When used with the 32KHz clock domain as a clock source, the differences can be as small as 1.

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	OUTPUT_DELAY The delay, in counts of the clock defined in Clock Source (CLKSRC), in which output transitions are delayed. When this field is 0, there is no added transition delay. When the LED is programmed to be Always On or Always Off, the Output Delay field has no effect.	R/W	000h	RESET_SYS

© 2020 Microchip Technology Inc.

32.0 KEYBOARD SCAN INTERFACE

32.1 Overview

The Keyboard Scan Interface block provides a register interface to the EC to directly scan an external keyboard matrix of size up to 18x8.

The maximum configuration of the Keyboard Scan Interface is 18 outputs by 8 inputs. For a smaller matrix size, firmware should configure unused KSO pins as GPIOs or another alternate function, and it should mask out unused KSIs and associated interrupts.

32.2 References

No references have been cited for this feature.

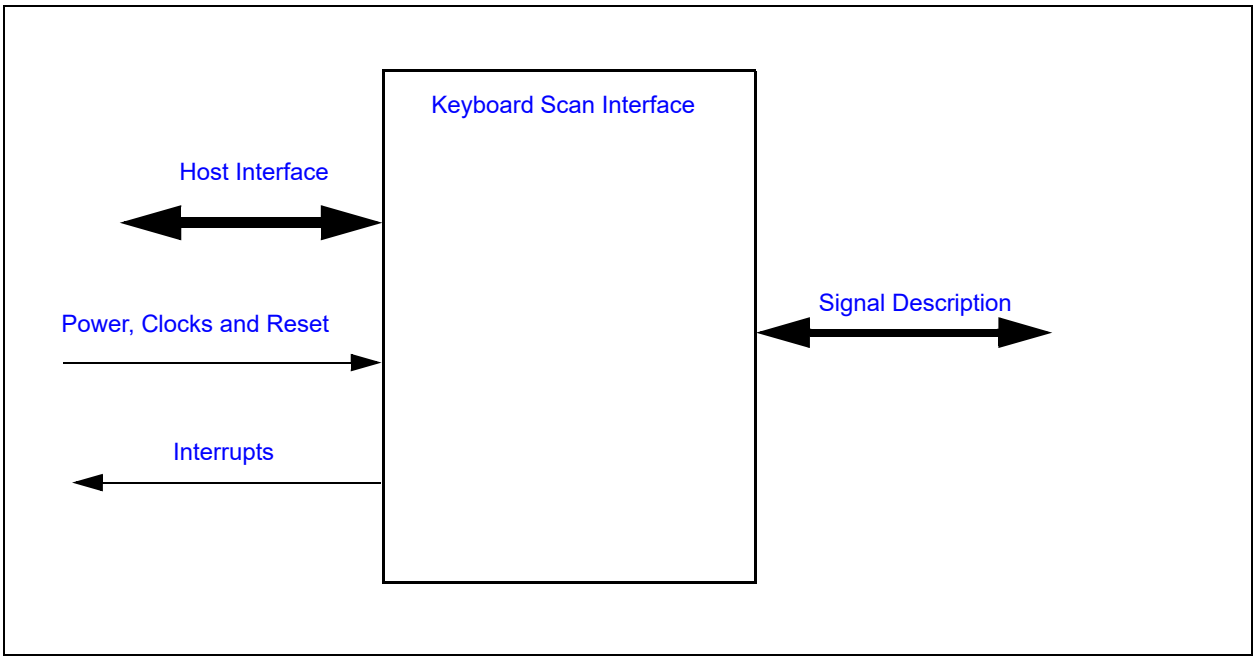
32.3 Terminology

There is no terminology defined for this section.

32.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 32-1: I/O DIAGRAM OF BLOCK



32.5 Signal Description

Name	Direction	Description
KSI[7:0]	Input	Column inputs from external keyboard matrix.
KSO[17:0]	Output	Row outputs to external keyboard matrix.

32.6 Host Interface

The registers defined for the Keyboard Scan Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

32.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

32.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

32.7.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for Keyboard Scan Interface logic.

32.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

32.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
KSC_INT	Wake capable Interrupt request to the Interrupt Aggregator.

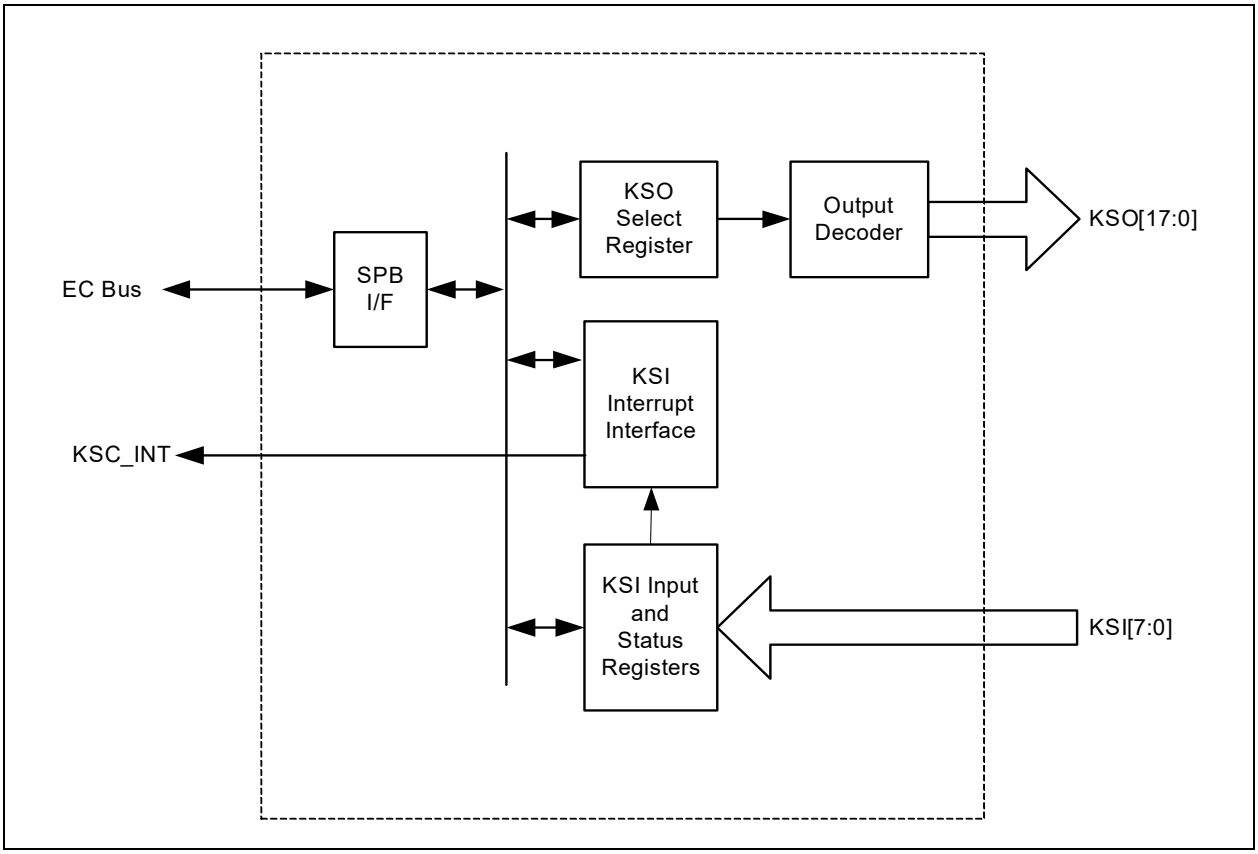
32.9 Low Power Modes

The Keyboard Scan Interface automatically enters a low power mode whenever it is not actively scanning the keyboard matrix. The block is also placed in a low-power state when it is disabled by the KSEN bit. When the interface is in a low-power mode it will not prevent the chip from entering a sleep state. When the interface is active it will inhibit the chip sleep state until the interface has re-entered its low power mode.

© 2020 Microchip Technology Inc.

32.10 Description

FIGURE 32-2: KEYBOARD SCAN INTERFACE BLOCK DIAGRAM



During scanning the firmware sequentially drives low one of the rows (KSO[17:0]) and then reads the column data line (KSI[7:0]). A key press is detected as a zero in the corresponding position in the matrix. Keys that are pressed are debounced by firmware. Once confirmed, the corresponding keycode is loaded into host data read buffer in the 8042 Host Interface module. Firmware may need to buffer keycodes in memory in case this interface is stalled or the host requests a Resend.

32.10.1 INITIALIZATION OF KSO PINS

If the Keyboard Scan Interface is not configured for PREDRIVE Mode, KSO pins should be configured as open-drain outputs. Internal or external pull-ups should be used so that the GPIO functions that share the pins do not have a floating input when the KSO pins are tri-stated.

If the Keyboard Scan Interface is configured for PREDRIVE Mode, KSO pins must be configured as push-pull outputs. Internal or external pull-ups should be used to protect the GPIO inputs associated with the KSO pins from floating inputs.

32.10.2 PREDRIVE MODE

There is an optional Predrive Mode that can be enabled to actively drive the KSO pins high before switching to open-drain operation. The PREDRIVE ENABLE bit in the Keyscan Extended Control Register is used to enable the PREDRIVE option. Timing for the Predive mode is shown in Section 50.12, Keyboard Scan Matrix Timing.

32.10.2.1 Predrive Mode Programming

The following precautions should be taken to prevent output pad damage during Predrive Mode Programming.

32.10.2.2 Asserting PREDRIVE_ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Enable Predrive function (PREDRIVE_ENABLE = '1')
3. Program buffer type for all KSO pins to "push-pull”
4. Enable Keyscan Interface (KSEN ='0')

32.10.2.3 De-asserting PREDRIVE_ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Program buffer type for all KSO pins to "open-drain”
3. Disable Predrive function (PREDRIVE_ENABLE = '0')
4. Enable Keyscan Interface (KSEN ='0')

32.10.3 INTERRUPT GENERATION

To support interrupt-based processing, an interrupt can optionally be generated on the high-to-low transition on any of the KSI inputs. A running clock is not required to generate interrupts.

32.10.3.1 Runtime interrupt

KSC_INT is the block’s runtime active-high level interrupt. It is connected to the interrupt interface of the Interrupt Aggregator, which then relays interrupts to the EC.

Associated with each KSI input is a status register bit and an interrupt enable register bit. A status bit is set when the associated KSI input goes from high to low. If the interrupt enable bit for that input is set, an interrupt is generated. An Interrupt is de-asserted when the status bit and/or interrupt enable bit is clear. A status bit cleared when written to a ‘1’.

Interrupts from individual KSIs are logically ORed together to drive the KSC_INT output port. Once asserted, an interrupt is not asserted again until either all KSI[7:0] inputs have returned high or the has changed.

32.10.4 WAKE PROGRAMMING

Using the Keyboard Scan Interface to ‘wake’ the MEC152x can be accomplished using either the Keyboard Scan Interface wake interrupt, or using the wake capabilities of the GPIO Interface pins that are multiplexed with the Keyboard Scan Interface pins. Enabling the Keyboard Scan Interface wake interrupt requires only a single interrupt enable access and is recommended over using the GPIO Interface for this purpose.

32.11 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Keyboard Scan Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 32-1: EC-ONLY REGISTER SUMMARY

Offset	Register Name
0h	Reserved
4h	KSO Select Register
8h	KSI INPUT Register
Ch	KSI STATUS Register
10h	KSI INTERRUPT ENABLE Register
14h	Keyscan Extended Control Register

© 2020 Microchip Technology Inc.

32.11.1 KSO SELECT REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
7	KSO_INVERT This bit controls the output level of KSO pins when selected. 0=KSO[x] driven low when selected 1=KSO[x] driven high when selected.	R/W	0h	RESET_SYS
6	KSEN This field enables and disables keyboard scan 0=Keyboard scan enabled 1=Keyboard scan disabled. All KSO output buffers disabled.	R/W	1h	RESET_SYS
5	KSO_ALL 0=When key scan is enabled, KSO output controlled by the KSO_SELECT field. 1=KSO[x] driven high when selected.	R/W	0h	RESET_SYS
4:0	KSO_SELECT This field selects a KSO line (00000b = KSO[0] etc.) for output according to the value off KSO_INVERT in this register. See Table 32-2, "KSO Select Decode"	R/W	0h	RESET_SYS

TABLE 32-2: KSO SELECT DECODE

KSO Select [4:0]	KSO Selected
00h	KSO00
01h	KSO01
02h	KSO02
03h	KSO03
04h	KSO04
05h	KSO05
06h	KSO06
07h	KSO07
08h	KSO08
09h	KSO09
0Ah	KSO10
0Bh	KSO11
0Ch	KSO12
0Dh	KSO13
0Eh	KSO14
0Fh	KSO15
10h	KSO16
11h	KSO17

© 2020 Microchip Technology Inc.

TABLE 32-3: KEYBOARD SCAN OUT CONTROL SUMMARY

KSO_INVERTt	KSEN	KSO_ALL	KSO_SELECT	Description
X	1	x	x	Keyboard Scan disabled. KSO[17:0] output buffers disabled.
0	0	0	10001b-00000b	KSO[Drive Selected] driven low. All others driven high
1	0	0	10001b-00000b	KSO[Drive Selected] driven high. All others driven low
0	0	0	11111b-10010b	All KSO's driven high
1	0	0	11111b-10010b	All KSO's driven low
0	0	1	x	All KSO's driven high
1	0	1	x	All KSO's driven low

32.11.2 KSI INPUT REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	KSI This field returns the current state of the KSI pins.	R	0h	RESET_SYS

32.11.3 KSI STATUS REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	KSI_STATUS Each bit in this field is set on the falling edge of the corresponding KSI input pin. A KSI interrupt is generated when its corresponding status bit and interrupt enable bit are both set. KSI interrupts are logically ORed together to produce KSC_INTWriting a '1' to a bit will clear it. Writing a '0' to a bit has no effect.	R/WC	0h	RESET_SYS

32.11.4 KSI INTERRUPT ENABLE REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	KSI_INT_EN Each bit in KSI_INT_EN enables interrupt generation due to high-to-low transition on a KSI input. An interrupt is generated when the corresponding bits in KSI_STATUS and KSI_INT_EN are both set.	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

32.11.5 KEYSCAN EXTENDED CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
32:1	Reserved	RES	-	-
0	PREDRIVE_ENABLE PREDRIVE_ENABLE enables the PREDRIVE mode to actively drive the KSO pins high for two 48 MHz PLL clocks before switching to open-drain operation. 0=Disable predrive on KSO pins 1=Enable predrive on KSO pins.	R/W	0h	RESET_SYS

33.0 I2C/SMBUS INTERFACE

33.1 Introduction

This section describes the Power Domain, Resets, Clocks, Interrupts, Registers and the Physical Interface of the I2C/SMBus interface. In I2C mode, this block supports Promiscuous mode when configured as I2C slave. For a General Description, Features, Block Diagram, Functional Description, Registers Interface and other core-specific details, *see* Ref [1] (note: in this chapter, *italicized text* typically refers to SMB-I2C Controller core interface elements as described in Ref [1]).

33.2 References

1. I2C_SMB Controller Core with Network Layer Support (SMB2) - 16MHz I2C Baud Clock“, Revision 3.6, Core-Level Architecture Specification, Microchip

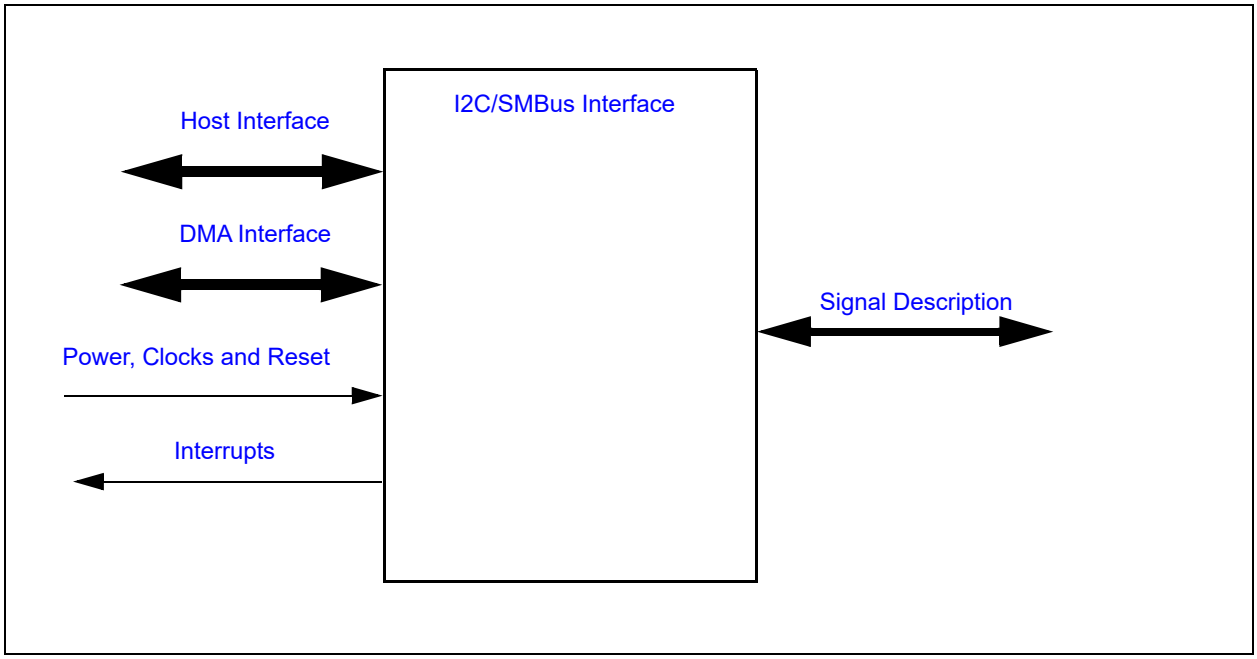
33.3 Terminology

There is no terminology defined for this chapter.

33.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface. In addition, this block is equipped with:

FIGURE 33-1: I/O DIAGRAM OF BLOCK



33.5 Signal Description

see the Pin Configuration section for a description of the SMB-I2C pin configuration.

33.6 Host Interface

The registers defined for the I2C/SMBus Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

© 2020 Microchip Technology Inc.

33.7 DMA Interface

This block is designed to communicate with the Internal DMA Controller. This feature is defined in the SMB-I2C Controller Core Interface specification (*See* Ref [1]).

Note: For a description of the Internal DMA Controller implemented in this design see Section 7.0, "Internal DMA Controller".

33.8 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

33.8.1 POWER DOMAINS

Name	Description
VTR_CORE	This power well sources all of the registers and logic in this block, except where noted.

33.8.2 CLOCK INPUTS

Name	Description
16MHz	This is the clock signal drives the SMB-I2C Controller core. The core also uses this clock to generate the SMB-I2C_CLK on the pin interface. It is derived from the main system clock

33.8.3 RESETS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in the SMB-I2C Controller core.

33.9 Interrupts

Source	Description
SMB-I2C	I^2C Activity Interrupt Event
SMB-I2C_WAKE	This interrupt event is triggered when an SMB/I2C Master initiates a transaction by issuing a START bit (a high-to-low transition on the SDA line while the SCL line is high) on the bus currently connected to the SMB-I2C Controller. The EC interrupt handler for this event only needs to clear the interrupt SOURCE bit and return; if the transaction results in an action that requires EC processing, that action will trigger the SMB-I2C interrupt event.

33.10 Low Power Modes

The SMB-I2C Controller may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

33.11 Description

33.11.1 SMB-I2C CONTROLLER CORE

The SMB-I2C Controller behavior is defined in the SMB-I2C Controller Core Interface specification (*See* Ref [1]).

33.11.2 PHYSICAL INTERFACE

The Physical Interface for the SMB-I2C Controller core is configurable for up to 10 ports. Each I2C_WAKE Controller can be connected to any of the ports defined in Table 33-1, "SMB-I2C Port Selection". The *PORT SEL [3:0]* bit field in each controller independently sets the port for the controller. The default for each field is Fh, Reserved, which means that the SMB-I2C Controller is not connected to a port.

An I^2C port should be connected to a single controller. An attempt to configure the *PORT SEL [3:0]* bits in one controller to a value already assigned to another controller may result in unexpected results.

The port signal-function names and pin numbers are defined in Pin Configuration section. The I^2C port selection is made using the *PORT SEL [3:0]* bits in the *Configuration Register* as described in Ref [1].. In the Pin section, the SDA (Data) pins are listed asi2Cxx_SDA and the SCL (Clock) pins are listed as I2C*xx*_SCL, where *xx* represents the port number 00 through 15. The CPU-voltage-level port SB_TSI is also listed in the pin section with the SD-TSI_DAT and SD-TSI_CLK.

For I^2C port signal functions that are alternate functions of GPIO pins, the buffer type for these pins must be configured as open-drain outputs when the port is selected as an I^2C port.

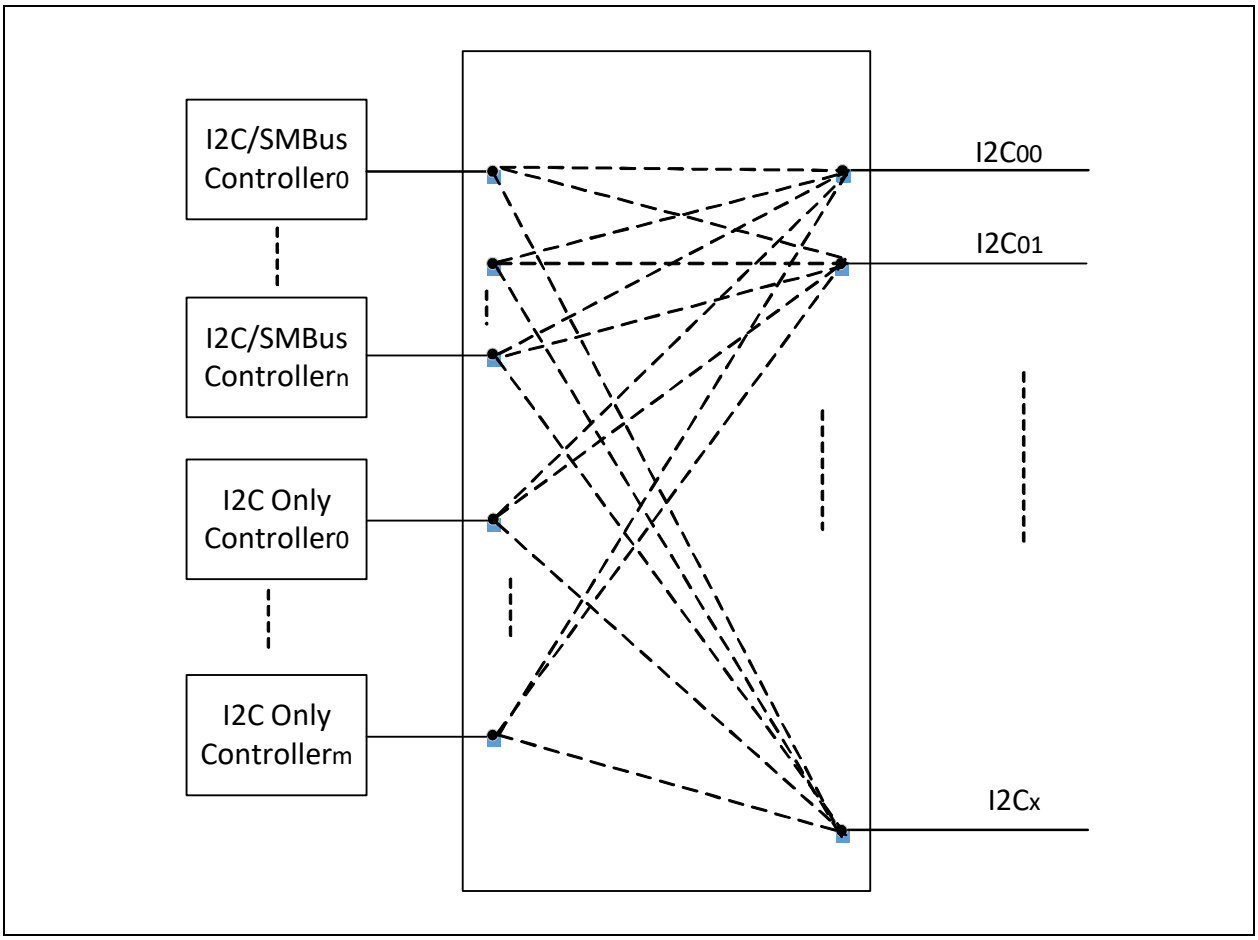
For more information regarding the SMB-I2C Controller core see *Section 2.2,* "*Physical Interface*" in Ref[1].

TABLE 33-1: SMB-I2C PORT SELECTION

PORT_SEL[3:0]				Port
3	**2**	**1**	**0**	
0	0	0	0	I2C00
0	0	0	1	I2C01
0	0	1	0	I2C02
0	0	1	1	I2C03
0	1	0	0	I2C04
0	1	0	1	I2C05
0	1	1	0	I2C06
0	1	1	1	I2C07
1	0	0	0	I2C08
1	0	0	1	I2C09
1	0	1	0	I2C10
1	0	1	1	I2C11
1	1	0	0	I2C12
1	1	0	1	I2C13
1	1	1	0	I2C14
1	1	1	1	I2C15
Note: Refer toSection 2.4.12for the pin mapping				

© 2020 Microchip Technology Inc.

FIGURE 33-2: SMB-I2C PORT CONNECTIVITY



33.12 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the SMB-I2C Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

Registers for the SMB-I2C Controllers are listed in Reference[1].

34.0 I2C CONTROLLER

34.1 Introduction

The I2C Controller functions as master and slave.It has support for clock synchronization, arbitration for multiple masters, bit-banging, and 7-bit addressing.

34.2 References

1. The I^2C-Bus Specification, Version 2.1, Philips Semiconductors, January 2000.
2. PCF8584 I^2C-Bus Controller, Product specification, Philips Semiconductors, 1997 Oct 21.
3. I2C_SMB Controller Core with Network Layer Support (SMB2) - 16MHz I2C Baud Clock“, Revision 3.6, Core-Level Architecture Specification.,Microchip

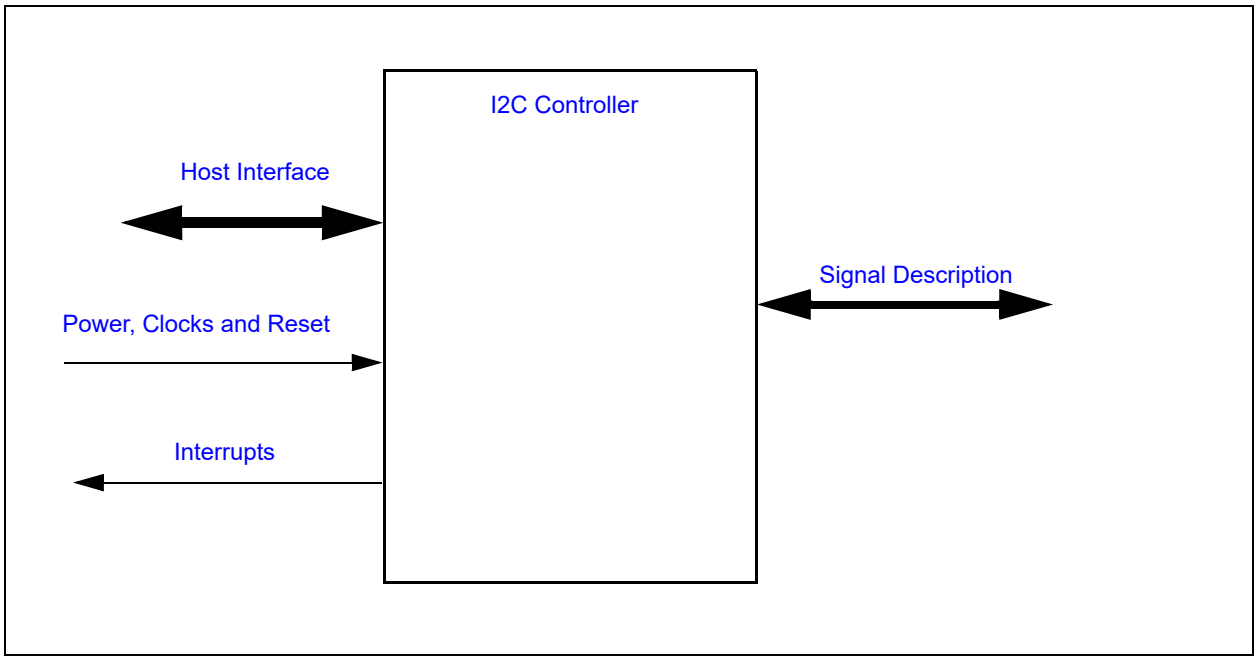
34.3 Terminology

There is no terminology defined for this section.

34.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 34-1: I/O DIAGRAM OF BLOCK



© 2020 Microchip Technology Inc.

34.5 Signal Description

TABLE 34-1: SIGNAL DESCRIPTION

Name	Direction	Description
I2C_SDA	BiDi	This is the Bidirectional I2C serial Data. An external pull up to VTR is required on this line
I2C_SCL	BiDi	This is the Bidirectional I2C serial clock. An external pull up to VTR is required on this line

34.6 Host Interface

The registers defined for the I2C Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

34.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

34.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

34.7.2 CLOCK INPUTS

Name	Description
I2C_BAUD_CLOCK	This is the 16MHz Baud clock for the module
48MHz	System Clock.

34.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

34.8 Interrupts

The section defines the Interrupt Sources generated from this block

The I2C_INTERRUPT signal is asserted from either of the 2 sources mentioned below.

FIGURE 34-2: I2C INTERRUPTS

Source	Description
I2C_intr	I2C Activity interrupt event
I2C_prom_int	Interrupt for a Promiscuous event
I2C_wake	This event is triggered when a START bit is detected on the line. This event will self clear when SCL goes high

34.9 Low Power Modes

The I2C Controller may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry

34.10 Description

34.10.1 CONTROLLER CORE

The MEC152x implements physical and data link layer support as defined in Ref [1] and Ref [2]. The I2C block is directly controlled by firmware through the use of a register interface which can initiate and respond to traffic on the external I2C bus.The core set of registers provide a byte-at-a-time interface to the external bus.

34.10.1.1 Master Controller

The I2C master Controller is always monitoring the filtered I2C clock and data lines as long as the block is enabled. The Master controls the flow of address & data and appropriately sends the ACK/NACK if it is receiving data. During the ADDRESS phase the master will control the shift-register to shift address into the shift-register. If the slave address matched, the Master will put the address in the data register, and assert the AAS bit

If the master is trying to address external slave but it lost arbitration, the master will update the data register with the received slave address (even though the address did not match), assert the LAB bit and assert the PIN (0) bit. The external I2C SCL will be stretched until the F/W de-asserts the PIN bit. This feature is added for multi-master support.

34.10.1.2 Slave Controller

The I2C Slave Control module responds to transfers initiated by I2C bus masters. The slave uses the Own Address Register to determine that an external master is attempting an access to it, and responds as indicated by firmware through the register bank interface

The I2C Controller when configured in Slave mode supports the Promiscuous mode of operation. Promiscuous Mode allows F/W to decide upon receiving the address packet, whether or not to ACK and packet and respond, or NAK the packet and ignore. This is for every address received, including addresses from Repeated Start protocols. When promiscuous mode is enabled by setting Promiscuous enable in Configuration Register

1. The General Call Address is disabled.
2. Promiscuous Address Interrupt function is enabled
3. Stall 9th clock of address byte enabled.
4. Address byte ACK/NAK done by Promiscuous ACK setting

34.10.2 PHYSICAL INTERFACE

The Physical Interface for the I2C Controller core is configurable for up to 16 ports. Each I2C_wake Controller can be connected to any of the ports defined in Table 34-2, "I2C port selection". The PORT_SEL[3:0] bit field in each controller independently sets the port for the controller. The default for each field is 0h, which means that the I2C Controller is connected to port 0.An I2C port should be connected to a single controller. An attempt to configure the PORT_SEL[3:0] bits in one controller to a value already assigned to another controller may result in unexpected results. Hence before enabling a Controller, care should be taken to fix the PORT_SEL[3:0] to a unique port.
The port signal-function names and pin numbers are defined in Pin Configuration section. The I2C port selection is made using the PORT_SEL[3:0] bits in the Configuration Register. In the Pin section, the SDA (Data) pins are listed as I2Cxx_SDA and the SCL (Clock) pins are listed as I2C*xx*_SCL, where *xx* represents the port number 00 through 15. The CPU-voltage-level port SB_TSI is also listed in the pin section with the pins SB-TSI_DAT and SB-TSI_CLK.For I2C port signal functions that are alternate functions of GPIO pins, the buffer type for these pins must be configured as open-drain outputs when the port is selected as an I2C port. Please refer to Figure 33-2, "SMB-I2C Port Connectivity" for more details.

TABLE 34-2: I2C PORT SELECTION

PORT_SEL[3:0]				Port
3	2	1	0	
0	0	0	0	I2C00
0	0	0	1	I2C01
0	0	1	0	I2C02
0	0	1	1	I2C03

© 2020 Microchip Technology Inc.

TABLE 34-2: I2C PORT SELECTION (CONTINUED)

PORT_SEL[3:0]				Port
3	2	1	0	
0	1	0	0	I2C04
0	1	0	1	I2C05
0	1	1	0	I2C06
0	1	1	1	I2C07
1	0	0	0	I2C08
1	0	0	1	I2C09
1	0	1	0	I2C10
1	0	1	1	I2C11
1	1	0	0	I2C12
1	1	0	1	I2C13
1	1	1	0	I2C14
1	1	1	1	I2C15

34.10.3 INPUT FILTERING

The MEC152x includes an analog filter on both the SCL and SDA input pins. The filter is designed to reject pulses less than 50ns and always pass pulses greater than 140ns.

By default the filter is disabled. To enable the filter, set the FEN bit located in the Configuration Register.

34.11 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the I2C Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 34-3: REGISTER SUMMARY

Offset (HEX)	Register Name
00h	Control Register
	Status Register
04h	Own Address Register
08h	Data Register
18h	Repeated START Hold Time Register
20h	Completion Register
28h	Configuration Register
2Ch	Bus Clock Register
30h	Block ID Register
34h	Revision Register
38h	Bit-Bang Control Register
3Ch	Test Data Timing Register
40h	Data Timing Register
44h	Time-Out Scaling Register
58h	Test
5Ch	Test

TABLE 34-3: REGISTER SUMMARY (CONTINUED)

Offset (HEX)	Register Name
60h	Wake Status Register
64h	Wake Enable Register
68h	Test
6Ch	Slave Address Register
70h	Promiscuous interrupt Register
74h	Promiscuous interrupt Enable Register
78h	Promiscuous Control Register

Registers marked “Test Register” must not be modified.

TABLE 34-4: RECOMMENDED PROGRAMMING VALUES

16MHz BAUD Clock:				
Register	**Default**	**Value for 100k**	**Value for 400k**	**Value for 1M**
Bus Clock Register	00_00_4F_4Fh	00_00_4F_4Fh	00_00_0F_17h	00_00_05_09h
Data Timing Register	0C_4D_50_06h	0C_4D_50_06h	04_0A_0A_06h	04_06_06_01h
Repeated START Hold Time Register	00_00_00_4Dh	00_00_00_4Dh	00_00_00_0Ah	00_00_00_06h
Configuration Register	01_FC_01_EDh	01_FC_01_EDh	01_00_00_50h	01_00_00_50h
Time-Out Scaling Register	4B_9C_C2_C7h	4B_9C_C2_C7h	15_9C_C2_C7h	08_9C_C2_C7h

TABLE 34-5: BUS CLOCK REGISTER VS. FREQUENCY

Bus Frequency (KHz)	Bus Clock Register value
1000	0509h
400	0F17h
333	0F1Fh
100	4F4Fh (default)
80	6363h
40	C7C7h

34.12 Register Description

34.12.1 CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7	PIN The Pending Interrupt Not bit serves as a software reset function. Writing the PIN bit to a logic ‘1’ de-asserts all status bits except for the HBB bit which is not affected by the PIN bit. The PIN bit is a self-clearing bit. Writing this bit to a logic ‘0’ has no effect.	W	N/A	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6	ESO The Enable Serial Output bit enables and disables the serial data output (I2C_SDA). When ESO is asserted (‘1’), I2C_SDA is enabled. When ESO is not asserted (‘0’) I2C_SDA is disabled. The ESO bit does not affect access to registers in the controller. The ESO bit must not be de-asserted when the I^2C Controller is actively involved in a bus transaction.	W	N/A	RESET_SYS
5:4	Reserved	RES	-	-
3	ENI The Enable Interrupt bit controls the Interrupts as described in Section 34.8, Interrupts.	W	N/A	RESET_SYS
2	STA Transmit START. The STA and STO bits control the generation of the I^2C Start condition and the transmission of the Slave Address and R/nW bit (from the Data Register), generation of repeated Start condition, and generation of the Stop condition as described in Table 34-6.	W	N/A	RESET_SYS
1	STO Transmit STOP. The STA and STO bits control the generation of the I^2C Start condition and the transmission of the Slave Address and R/nW bit (from the Data Register), generation of repeated Start condition, and generation of the Stop condition as described in Table 34-6.	W	N/A	RESET_SYS
0	ACK Acknowledge. The ACK must normally be asserted (‘1’). This causes the controller to send an acknowledge automatically after each byte (this occurs during the 9th clock pulse). The ACK bit must be de-asserted (‘0’) when the controller is operating in master/receiver mode and requires no further data to be sent from the slave transmitter. This causes a negative acknowledge on the I^2C bus, which halts further transmission from the slave device.	W	N/A	RESET_SYS

TABLE 34-6: INSTRUCTION TABLE FOR SERIAL BUS CONTROL

STA	STO	Mode	Function	Operation
1	0	SLV/REC	START	Transmit START+address, remain MST/TRM if Data Register bit 0 (R/nW) = 0; go to MST/REC if Data Register bit 0 (R/nW) =1.
		MST/TRM	REPEAT START	Same as for SLV/REC
0	1	MST/REC; MST/TRM	STOP READ; STOP WRITE	Transmit STOP go to SLV/REC mode. In Master Receive mode, the last received byte must be terminated with a NACK.
1	1	Reserved		Reserved
0	0	ANY	NOP	No operation

34.12.2 STATUS REGISTER

Use of this register is only required for legacy I^2C operation. Access to this register is not required if Network Layer functions are used.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7	PIN The operation of PIN (Pending Interrupt)	R	1h	RESET_SYS
6	Test	R	0h	RESET_SYS
5	STS This bit is asserted (‘1’) when an externally generated STOP condition is detected. It is used only in slave receiver mode.	R	0h	-
4	BER When BER (Bus Error) is asserted, a misplaced START or STOP condition or Bus Time-outs have been detected. If this bit is asserted, NBB (‘1’) and PIN (‘0’) are de-asserted. After BER is asserted, the controller must be reset before taking any further action.	R	0h	-
3	LRB/AD0 LBR/AB0 (“Last Received Bit” or “Address 0”) is valid only while the PIN bit is asserted (‘0’). When the AAS bit is not asserted (‘0’) (i.e., the controller has not addressed as a slave), this bit holds the value of the last received bit over the bus. Normally this will be the value of the slave acknowledgment; thus, checking for slave acknowledgment is done via testing this bit. When the AAS bit is asserted (‘1’) (i.e., the controller has been addressed as a slave), this bit will be set to logic ‘1’ if the slave address received was the ‘general call’ (00h) address, or logic ‘0’ if the received slave address matches the value programmed in the Own Address Register.	R	0h	RESET_SYS
2	AAS AAS (Addressed As Slave) is valid only when PIN is asserted (‘0’). When acting as a slave, this bit is set when an incoming address over the bus matches the value in the Own Address Register (shifted by one bit) or if the ‘general call’ address (00h) has been received (‘general call’ is indicated by the LRB/AD0 bit)	R	0h	RESET_SYS
1	LAB LAB (Lost Arbitration) is set when, in multi-master operation, arbitration is lost to another master on the bus as described in Section 34.10.1.1, "Master Controller".	R	0h	RESET_SYS
0	NBB The Bus Busy bit indicates when the bus is in use. A zero indicates that the bus is busy and access is not possible. This bit is asserted (‘0’) by a START condition and de-asserted (‘1’) following a STOP condition.	R	1h	RESET_SYS

© 2020 Microchip Technology Inc.

34.12.3 OWN ADDRESS REGISTER

This register provides the ability for the MEC152x to respond to two different slave addresses.

Note: The internal master should not attempt transactions to a slave with the same address as one that is programmed in the **Own Address Register**. This represents an illegal operation. If it occurs the controller must be reset.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:8	OWN_ADDRESS_2 This field configures one of the two addresses to which the controller will respond when addressed as a slave. The Own Address fields is offset by one bit, so that programming this field with a value of 55h will result in the value AAh being recognized as the slave address. The AAS bit in the Status Register is set when this field matches an incoming slave address. The address '0000000b' is the General Call Address. An Own_Address of 0h will match the General Call Address unless this match is disabled by setting GC_DIS in the Configuration Register.	R/W	0h	RESET_SYS
7	Reserved	RES	-	-
6:0	OWN_ADDRESS_1 This field configures one of the two addresses to which the controller will respond when addressed as a slave. The Data Register and Own Address fields are offset by one bit, so that programming this field with a value of 55h will result in the value AAh being recognized as the slave address. The AAS bit in the Status Register is set when this field matches an incoming slave address. The address '0000000b' is the General Call Address. An Own_Address of 0h will match the General Call Address unless this match is disabled by setting GC_DIS in the Configuration Register.	R/W	0h	RESET_SYS

Note: Device OWN Address should not be set to the SMBUS HOST ADDRESS (0001_000b) or SMBUS DEVICE DEFAULT ADDRESS (1100_001b).

34.12.4 DATA REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	DATA This register holds the data that are either shifted out to or shifted in from the I^2C port.	R/W	0h	RESET_SYS

34.12.5 REPEATED START HOLD TIME REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	RPT_START_HOLD_TIME This is the value of the timing requirement $t_{Hd:Sta}$ in the I^2C specification for a repeated START bit. This is used to hold the clock until the Hold Time for the repeated Start Bit has been satisfied. See Table 34-4, "Recommended Programming Values" for recommended values for this field for different clock rates.	R/W	4Dh	RESET_SYS

34.12.6 COMPLETION REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	SDONE Slave Done. 1=Slave transaction completed 0=Slave transaction not completed	R/WC	0h	RESET_SYS
30	MDONE 1=Master transaction completed 0=Master transaction not completed	R/WC	0h	RESET_SYS
29	IDLE This bit is set to '1' when: • The bus becomes idle (on the rising edge of NBB in the Status Register) • The bus was idle when the Idle Interrupt was enabled • The bus was idle when the controller was enabled (the ENAB bit in the Configuration Register was set to '1')	R/WC	0h	RESET_SYS
28:26	Reserved	RES	-	-
25	MTR Master Transmit/Receive. This bit reports the phase of the Master State Machine when it asserts MDONE. 1=Master has just finished the transmit phase of a transaction 0=Master has just finished the receive phase of a transaction	R	0h	RESET_SYS
24	MNAKX Master received a NACK while Transmitting. 1=The external Slave sent a NACK to the Master while the Master was transmitting data over the I^2C port 0=No NACK was received during the Master transmission	R/WC	0h	RESET_SYS
23:22	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
21	REPEAT_WRITE 1=The Slave state machine paused because it detected a Repeat START Write (bit[0] of the byte containing the Slave address was '0') 0=No Repeat Write detected	R/WC	0h	RESET_SYS
20	REPEAT_READ 1=The Slave state machine paused because it detected a Repeat START Read (bit[0] of the byte containing the Slave address was '1') 0=No Repeat Read detected	R/WC	0h	RESET_SYS
19	Test	R/WC	0h	RESET_SYS
18	Reserved	RES	-	-
17	STR Slave Transmit/Receive 1=The Slave just finished the receive phase of a transaction 0=The Slave just finished the transmit phase of a transaction	R	0h	RESET_SYS
16	SNAKR Slave NACK sent while Receiving 1=The Slave sent a NACK to the external transmitting Master while the Slave was receiving data from the I^2C port 0=A NACK was not sent to the Master	R/WC	0h	RESET_SYS
15	Reserved	RES	-	-
14	LAB Lost Arbitration Status. This bit is set to '1' if the LAB bit in the Status Register was set to '1' while either a Master or a Slave transaction was in progress.	R/WC	0h	RESET_SYS
13	BER Bus Error Status. This bit is set to '1' if the BER bit in the Status Register was set to '1' while either a Master or a Slave transaction was in progress.	R/WC	0h	RESET_SYS
12	CHDH Clock High Data High time-out status.This is the bus idle time-out status. 1=Time-Out occurred 0=Time-Out did not occur	R/WC	0h	RESET_SYS
11	CHDL Clock High Data Low time-out status. This status is asserted if the SCL remains high while SDA remains low. 1=Time-Out occurred 0=Time-Out did not occur	R/WC	0h	RESET_SYS

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	SCTO Slave Cumulative Time-Out status. 1=Time-Out occurred 0=Time-Out did not occur	R/WC	0h	RESET_SYS
9	MCTO Master Cumulative Time-Out status. 1=Time-Out occurred 0=Time-Out did not occur	R/WC	0h	RESET_SYS
8	DTO Device Time-Out status. 1=Time-Out occurred 0=Time-Out did not occur	R/WC	0h	RESET_SYS
7	Reserved	RES	-	-
6	TIMERR Time Out Error Detected. This bit is ‘1’ if any of the Time-Out status bits (CHDH, CHDL, MCTO and DTO) are both asserted ‘1’ and enabled by their respective Time Out Enable bits.	R	0h	RESET_SYS
5	BIDEN This bit enables Bus Idle Time-Out checking, as described in. 1=Time-Out checking is enabled 0=Time-Out checking is disabled	R/W	0h	RESET_SYS
4	SCEN This bit enables Slave Cumulative Time-Out checking. 1=Time-Out checking is enabled 0=Time-Out checking is disabled	R/W	0h	RESET_SYS
3	MCEN This bit enables Master Cumulative Time-Out checking 1=Time-Out checking is enabled 0=Time-Out checking is disabled	R/W	0h	RESET_SYS
2	DTEN This bit enables Device Time-Out checking. 1=Time-Out checking is enabled 0=Time-Out checking is disabled	R/W	0h	RESET_SYS
1:0	Reserved	RES	-	-

Note: The Bus Time outs are defined in Ref[3].

© 2020 Microchip Technology Inc.

34.12.7 CONFIGURATION REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Test	-	0h	RESET_SYS
15	Promiscuous enable 0: Normal Operation 1: Promiscuous mode enable	R/W	0	RESET_SYS
14	GC_DIS General Call disable 1=The response to the General Call address (0h) as a slave is enabled. 0=The response to the General Call address (0h) as a slave is disabled.	R/W	0h	RESET_SYS
13	Test Must be always written with 0.	R/W	0h	RESET_SYS
12	Test	R/W	0h	RESET_SYS
11	Test	R/W	0h	RESET_SYS
10	ENAB I^2C controller enable. 1=Controller is enabled 0=Controller disabled and in its lowest power state	R/W	0h	RESET_SYS
9	RESET When RESET is asserted (‘1’), all logic and registers except for the RESET bit itself are initialized to the power-on default state. RESET is not self-clearing. It must remain asserted for at least one baud clock period. Register reads while RESET is asserted return default register values. The RESET bit is itself reset on the main system reset and not on the block reset signal	R/W	0h	RESET_EC
8	FEN Input filtering enable. See Section 34.10.3, "Input Filtering". Input filtering is required by the I^2C specification if external filtering is not available. 1=Input filtering is enabled 0=Input filtering is disabled	R/W	0h	RESET_SYS
7	Test	R/W	0h	RESET_SYS
6	Test Must be always written with 0.	R/W	0h	RESET_SYS

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	SLOW_CLOCK 1=The base period for the Bus Clock Register is multiplied by 4, and thus the frequency is divided by 4. It does not affect other timing calculations (such as those in the Data Timing Register or the Time-Out Scaling Register 0=The base period for the Bus Clock Register is the controller's baud clock	R/W	0h	RESET_SYS
4	TCEN Timing Check Enable. 1=Bus Time-Outs are enabled. Time-Outs must not be enabled when Bit-Banging BBEN is enabled. 0=Bus Time-Outs are disabled	R/W	0h	RESET_SYS
3:0	PORT_SEL This field selects which of the 16 possible bus ports is the currently active I^2C port. Unselected bus ports remain inactive. See the specific part Data Sheet for the configuration of ports per I^2C controller.	R/W	0h	RESET_SYS

Note:	All fields in this register should be set to their desired values before setting the ENAB bit in this register to '1b'. Failure to do so could result in glitches on the signal interface, and incorrect state machine behavior.

34.12.8 BUS CLOCK REGISTER

The Bus Clock Register is used to determine I^2C bus clock when the controller is a master. The bus clock period is determined by the following equation:

EQUATION 34-1: $\mathbf{BUS_CLOCK_{PERIOD} = ((LOW_PERIOD+1) + (HIGH_PERIOD+1)) \times I2C_BAUD_CLOCK_{PERIOD}}$

Examples of appropriate settings for the Bus Clock Register for several desired bus frequencies are shown in Table 34-5, "Bus Clock Register vs. Frequency".

Changes to the Bus Clock Register must only occur when the ESO bit in the Control Register is not asserted; i.e., the register must not be changed in the middle of a transaction. In order to comply with the I^2C the registers listed in Table 34-4, "Recommended Programming Values" may have to be adjusted when changing bus speeds.

If the SLOW_CLOCK bit in the Configuration Register is set to '1', the $I2C_BAUD_CLOCK_{PERIOD}$ term is multiplied by 4, and the frequencies shown in Table 34-5, "Bus Clock Register vs. Frequency" are divided by 4.

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:8	HIGH_PERIOD This field defines the number of I2C_BAUD_CLOCK periods that make up the high phase of the I^2C bus clock. The number of clock periods is one greater than the contents of this field. When both HIGH_PERIOD and LOW_PERIOD are 00h, the bus clock is disabled and master transactions cannot occur.	R/W	4Fh	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	LOW_PERIOD This field defines the number of I2C_BAUD_CLOCK periods that make up the low phase of the I^2C bus clock. The number of clock periods is one greater than the contents of this field	R/W	4Fh	RESET_SYS

34.12.9 BLOCK ID REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	ID Block ID	R	11h	RESET_SYS

34.12.10 REVISION REGISTER

Offset	34h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	REVISION Block Revision Number	R	0	RESET_SYS

34.12.11 BIT-BANG CONTROL REGISTER

Offset	38h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
6	BBDATI Bit-Bang Data In. This bit always returns the state of the SDA pin.	R	1h	RESET_SYS
5	BBCLKI Bit-Bang Clock In. This bit always returns the state of the SCL pin.	R	1h	RESET_SYS
4	BBDAT Bit-Bang Mode Data.state. 1=The SDA pin is tri-stated 0=If BBEN='1' and DADIR='1', the SDA pin is driven low. If either BBEN='1' or DADIR='1', the SDA pin is tri-stated	R/W	0h	RESET_SYS

Offset	38h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	BBCLK Bit-Bang Mode Clock.state. 1=The SCL pin is tri-stated 0=If BBEN='1' and DADIR='1', the SCL pin is driven low. If either BBEN='1' or DADIR='1', the SCL pin is tri-stated	R/W	0h	RESET_SYS
2	DADIR Bit-Bang Mode Data direction. 1=If BBEN='1', the SDA pin is an output 0=If BBEN='1', the SDA pin is an input	R/W	0h	RESET_SYS
1	CLDIR Bit-Bang Mode Clock direction. 1=If BBEN='1', the SCL pin is an output 0=If BBEN='1', the SCL pin is an input	R/W	0h	RESET_SYS
0	BBEN Bit-Bang Mode Enable. 1=Bit-Bang Mode Enabled 0=Bit Bang Mode Disabled	R/W	0h	RESET_SYS

34.12.12 DATA TIMING REGISTER

Recommended values for the Data Timing Register are shown in Table 34-4, "Recommended Programming Values".

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	FIRST_START_HOLD This field determines the SCL hold time following SDA driven low during the first START bit in a transfer. It is the parameter $T_{HD:STA}$ in the I^2C Specification for an initial START bit. Repeated START hold time is determined by the Repeated START Hold Time Register.	R/W	OCh	RESET_SYS
23:16	STOP_SETUP This field determines the SDA setup time from the rising edge of SCL for a repeated START condition. It is the parameter $T_{SU:STOP}$ in the I^2C Specification.	R/W	4Dh	RESET_SYS
16:8	RESTART_SETUP This field determines the SDA setup time from the rising edge of SCL for a repeated START condition. It is the parameter $T_{SU:STA}$ in the I^2C Specification.	R/W	50h	RESET_SYS
7:0	DATA_HOLD This field determines the SDA hold time following SCL driven low. It is the parameter $T_{HD:DAT}$ in the I^2C Specification.	R/W	06h	RESET_SYS

© 2020 Microchip Technology Inc.

34.12.13 TIME-OUT SCALING REGISTER

Time-Outs and Bus Busy calculation are described in Ref[3]. Recommended values for the Time-Out Scaling Register are shown in Table 34-4, "Recommended Programming Values".

The Baud_Clock_Period is the cycle time of the I2C_BAUD_CLOCK

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	BUS_IDLE_MIN This field determines the minimum bus idle time, also defined as the time between a STOP and START condition. It is the parameter T_{BUF} in the I^2C Specification. It is defined here as BUS_IDLE_MIN × Baud_Clock_Period	R/W	4Bh	RESET_SYS
23:0	Test		9CC2C7h	

34.12.14 WAKE STATUS REGISTER

Offset	60h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	START_BIT_DETECTION This bit is set to ‘1’ when a START bit is detected while the controller is enabled. This bit is cleared to ‘0’ when written with a ‘1’. Writes of ‘0’ have no effect.	R/WC	0h	RESET_SYS

34.12.15 WAKE ENABLE REGISTER

Offset	64h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	START_DETECT_INT_EN Enable Start Bit Detection Interrupt. The Start Bit Detection Interrupt is wake-capable. 1=Start Bit Detection Interrupt enabled 0=Start Bit Detection Interrupt disabled	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

34.12.16 SLAVE ADDRESS REGISTER

Offset	6Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	Slave Address This field is the Slave Address and Direction bit from the 1st byte of a Slave transfer. This is loaded immediately upon receipt of the address byte, before the ACK/NAK 9th clock	R/W	0h	RESET_SYS

34.12.17 PROMISCUOUS INTERRUPT REGISTER

Offset	70h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	Promiscuous Address Interrupt Functional only in Promiscuous Mode. Will not be set otherwise. Set when the 1st byte of a slave transfer has completed, but before the 9th ACK/NAK clock has been sent. During this period the 9th clock will be stretched by the slave until this interrupt status is cleared	R/W	0h	RESET_SYS

34.12.18 PROMISCUOUS INTERRUPT ENABLE REGISTER

Offset	74h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	Promiscuous Address Interrupt Enable Enables Slave Address	R/W	0h	RESET_SYS

34.12.19 PROMISCUOUS CONTROL REGISTER

Offset	78h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	Promiscuous Ack 0: NAK the address byte 1: ACK the address byte	R/W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

35.0 PROCHOT MONITOR

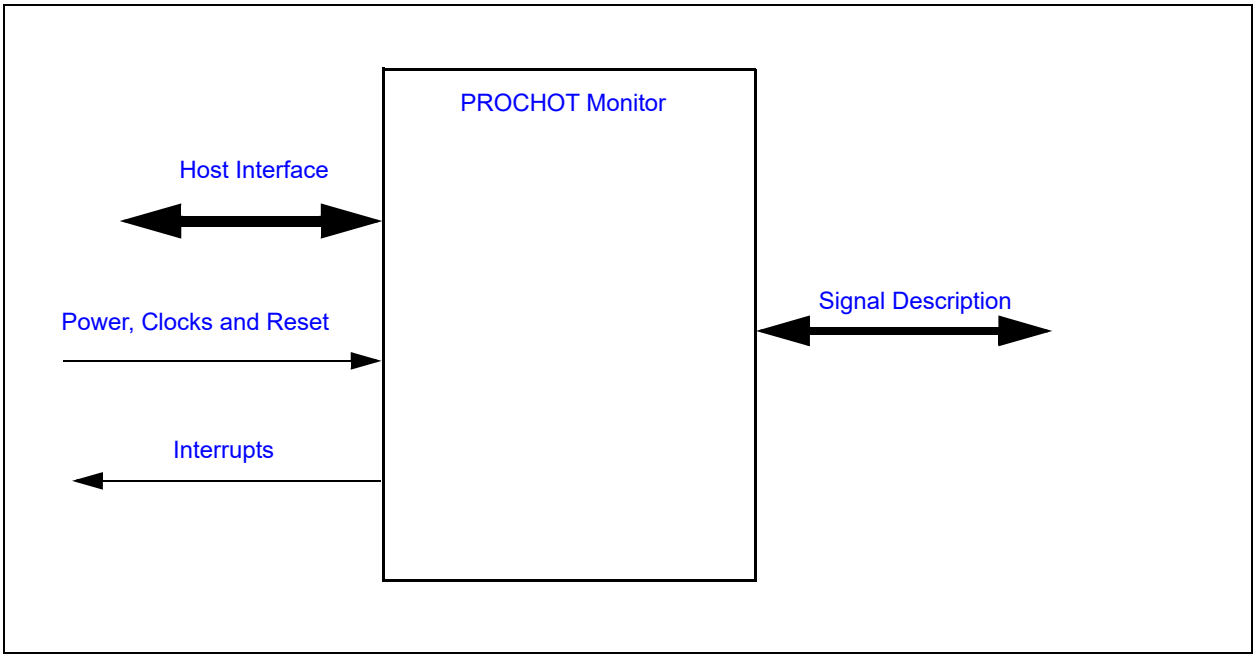
35.1 Overview

This block monitors the PROCHOT_IN# signal generated by the host processor. It is designed to detect single assertions and monitor cumulative PROCHOT active time.

35.2 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 35-1: I/O DIAGRAM OF BLOCK



35.3 Signal Description

TABLE 35-1: SIGNAL DESCRIPTION

Name	Direction	Description
PROCHOT_IN#	Input	PROCHOT_IN# is an active low signal generated by some processors to indicate the processor is running hot. This signal is used to throttle the processors clocks and as notification to the system. Some processors are equipped with a bi-directional PROCHOT pin. This PROCHOT block in combination with a PWM can be used to support a bi-directional PROCHOT pin.

35.4 Host Interface

The registers defined for the PROCHOT Monitor are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

35.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

35.5.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

35.5.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for this block
100KHz	This is the clock source for filters and counters

35.5.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.
RESET_PHOT	This signal resets all the logic and registers in this block when RESET_SYS is asserted and when the PHOT_RESET bit is asserted by software. This reset is not affected by Low Power Modes.
RESET_SLP	This signal resets all the logic and registers in this block when PHOT_RESET is asserted and when the block is commanded to sleep by the Sleep_Enable signal.

35.6 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
PHOT	PHOT events are generated when either a single PROCHOT Assertion event is detected or a Cumulative PROCHOT Assertion is detected. A PROCHOT Assertion event is generated when the PHOT_ASSERT bit is asserted if the ASSERT_ENABLE bit is set to one. A Cumulative PROCHOT Assertion event is generated when the PHOT_PERIOD bit is asserted if the PERIOD_ENABLE bit is set to one.

35.7 Low Power Modes

The PROCHOT Monitor may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. The block will immediately enter sleep when commanded, resetting any internal counters and filters back to their initial state.

35.8 Description

The PROCHOT logic defaults to disabled (i.e., not monitoring). To enable the PROCHOT function, set the PROCHOT Enable bit in the PROCHOT Status/Control Register to one.

The PROCHOT Monitor block supports two types of monitoring features:

- PROCHOT Assertion Monitoring
- PROCHOT Cumulative Duty Cycle Monitoring

© 2020 Microchip Technology Inc.

Note: A 3-stage filter is implemented to prevent system noise from being detected as a processor hot event. If enabled via the FILT_ENABLE bit, this filter will always reject an high or low pulses that are less than two times the 100KHz clock period.

35.8.1 PROCHOT ASSERTION MONITORING

The PROCHOT Monitor block is equipped with an internal counter, referred to as the PROCHOT Assertion Counter, that increments when the PROCHOT_IN# signal is active (i.e., low) and is reset to zero when the PROCHOT_IN# signal is inactive (i.e., high). The host may read the current value of this counter at any time via the PROCHOT Assertion Counter Register. If the value of this counter is greater than or equal to the value programmed in the PROCHOT Assertion Counter Register the PHOT_ASSERT status bit is set to one. This sticky status bit will remain set until either the bit is reset by the defined hardware reset event or software clears the bit. If the ASSERT_ENABLE bit is set, the PHOT_ASSERT status bit will trigger a PHOT.

35.8.2 PROCHOT CUMULATIVE DUTY CYCLE MONITORING

The PROCHOT Cumulative Duty Cycle Monitoring feature measures the total amount of time the PROCHOT_IN# signal is active (i.e., low) for a specified period of time. The period is programmable in the PROCHOT Duty Cycle Period Register.

To enable the PROCHOT Cumulative Duty Cycle Monitoring feature the PROCHOT Monitor block must be enabled via the PHOT_ENABLE bit and the PROCHOT Duty Cycle Period Register must be set to a value greater than zero. When the PROCHOT Cumulative Duty Cycle Monitoring feature is enabled the internal counters associated with this feature are initialized. The PROCHOT Active Counter, which is used to measure the total amount of time the PROCHOT_IN# signal is active during a specified period, is reset to zero. The PROCHOT Period Counter, which is used to determine the PROCHOT period being monitored, is initialized to the value programmed in the PROCHOT Duty Cycle Period Register. Once enabled, these internal counters will run continuously until either the block is disabled or the PROCHOT Duty Cycle Period Register is written to zero.

35.8.2.1 PROCHOT Active Counter

- Increments when the PROCHOT_IN# signal is active (i.e., low) and is held static when the PROCHOT_IN# signal is inactive (i.e., high).
- The PROCHOT Active Counter is reset to zero when the PROCHOT Monitor block is disabled, when the PROCHOT Duty Cycle Period Register is written, and when the PROCHOT Period Counter reaches zero. The host may read the current value of this counter at any time via the PROCHOT Cumulative Count Register.

35.8.2.2 PROCHOT Period Counter

- Decrements from the programmed PROCHOT Duty Cycle Period until counter reaches zero.
- When the duty cycle counter reaches zero:
  - The PROCHOT Cumulative Count Register, which contains the filtered PROCHOT Active Counter value, is loaded into the PROCHOT Duty Cycle Count Register
  - The PROCHOT Active Counter is reset to zero.
  - The PHOT_PERIOD bit is set in the PROCHOT Status/Control Register.
  - The PROCHOT Period Counter is reloaded from the PROCHOT Duty Cycle Period Register.

35.8.2.3 PROCHOT Cumulative Duty Cycle Filter

A filter has been implemented to discard any invalid active time. The minimum required assertion period for a valid PROCHOT_IN# signal is 500µS. Pulses less than 500µS should be ignored. The PROCHOT Active Counter is used to remove low-going pulses that are less than this threshold. This is done by discarding any counter incremental values that are less than or equal to 50 100KHz clock pulses.

© 2020 Microchip Technology Inc.

Figure 35-2, "Effects of PROCHOT# Filtering" shows the effect of PROCHOT_IN# input filtering:

FIGURE 35-2: EFFECTS OF PROCHOT# FILTERING

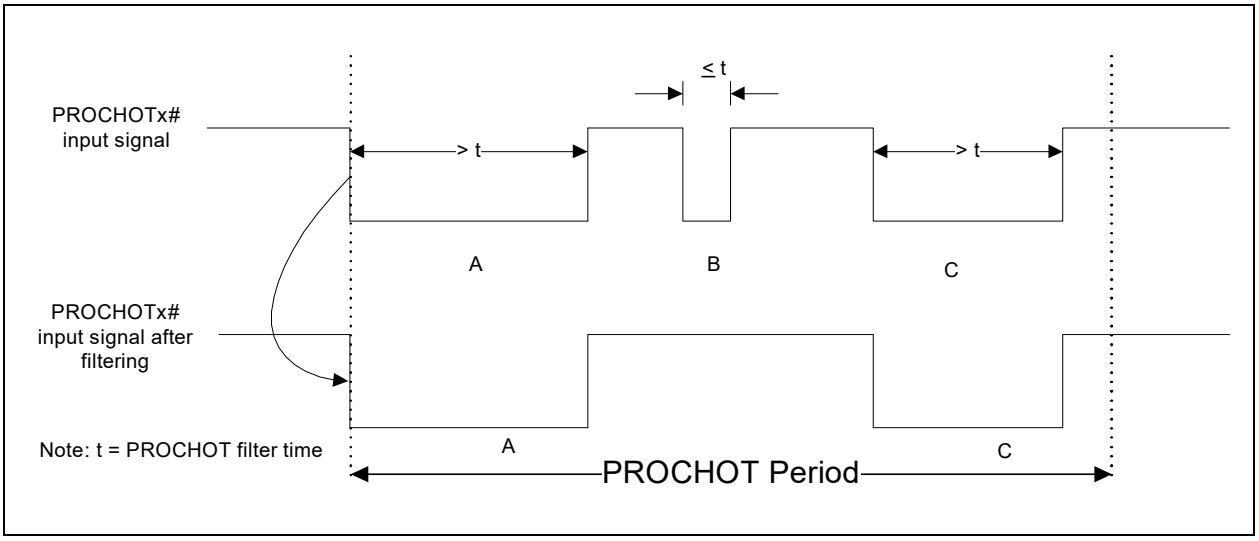
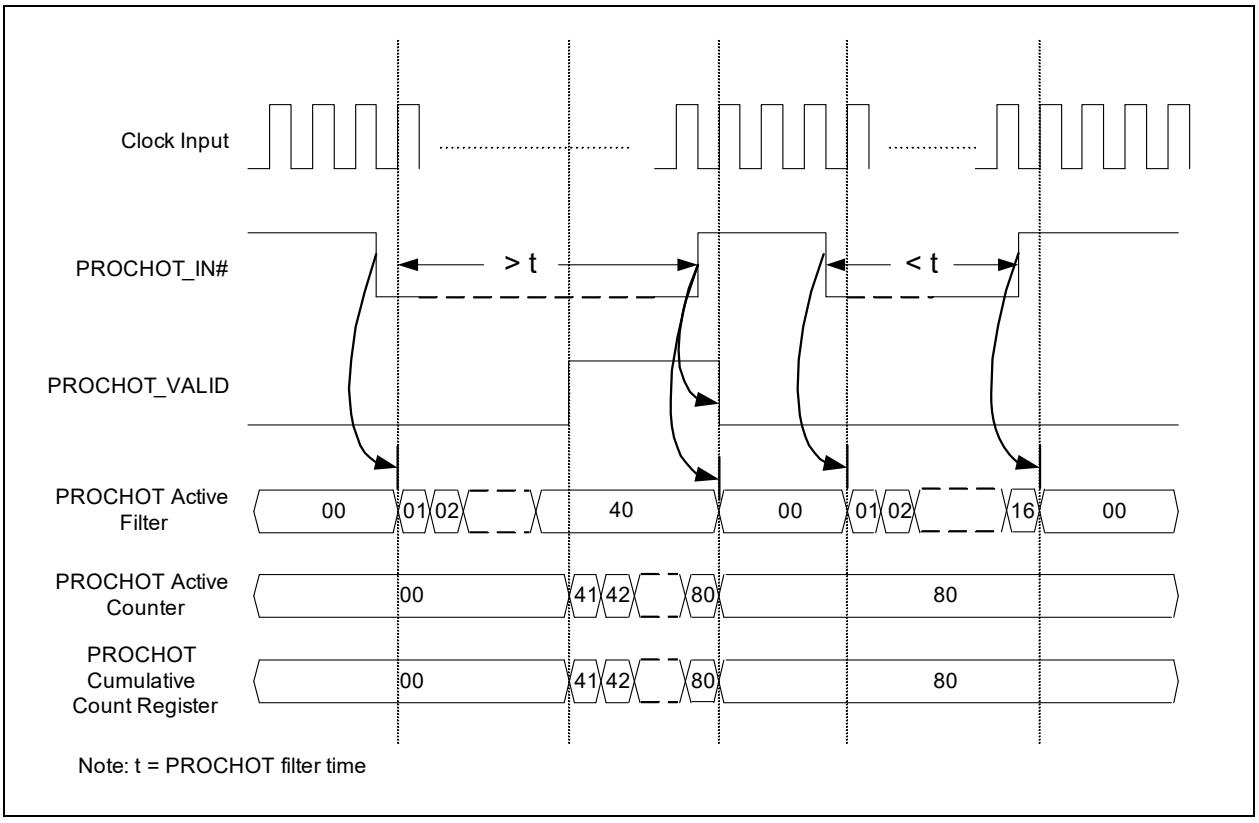


Figure 35-3, "Example of PROCHOT Active Counter" provides an example of the interaction between the internal PROCHOT Cumulative Duty Cycle Filter, PROCHOT Active Counter and the PROCHOT Cumulative Count Register.

FIGURE 35-3: EXAMPLE OF PROCHOT Active Counter



© 2020 Microchip Technology Inc.

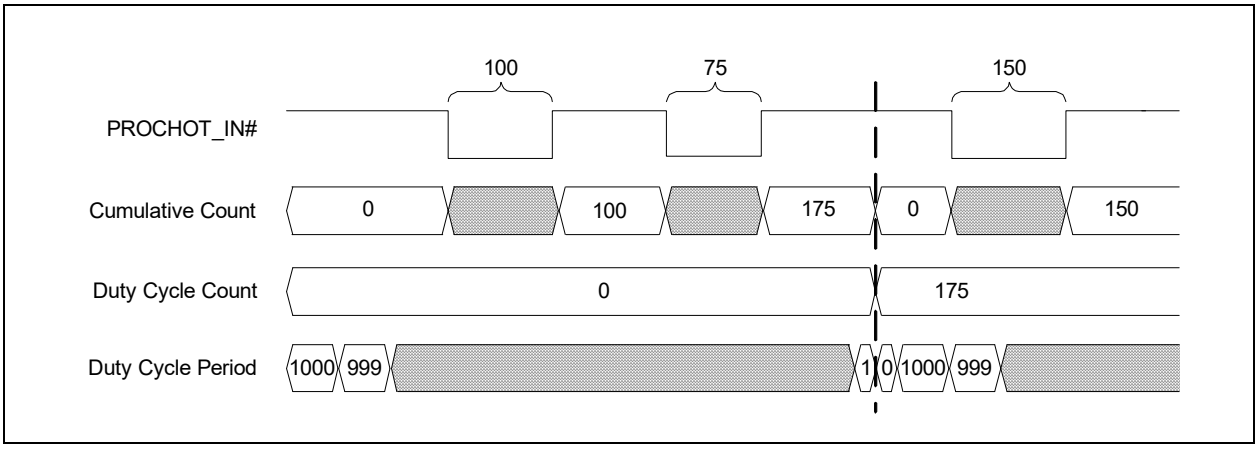
35.8.2.4 Determining a PROCHOT Cumulative Duty Cycle

This hardware may be used by firmware to determine the cumulative duty cycle of the PROCHOT_IN# signal. The cumulative duty cycle (the percentage of time that PROCHOT was asserted for a period longer than the filter time) is derived from the PROCHOT Duty Cycle Count Register and the PROCHOT Duty Cycle Period Register. The duty cycle can be calculated using the relation:

$$DutyCycle = \frac{PROCHOT\ Duty\ Cycle\ Count}{PROCHOT\ Duty\ Cycle\ Period}$$

Figure 35-4, "PROCHOT Duty Cycle Example" shows an example of how the Cumulative Count, Duty Cycle Count and Duty Cycle Period registers relate. The numbers are arbitrary and are for illustration purposes only:

FIGURE 35-4: PROCHOT Duty Cycle Example



35.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the PROCHOT Monitor Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 35-2: REGISTER SUMMARY

Offset	Register Name
00h	PROCHOT Cumulative Count Register
04h	PROCHOT Duty Cycle Count Register
08h	PROCHOT Duty Cycle Period Register
0Ch	PROCHOT Status/Control Register
10h	PROCHOT Assertion Counter Register
14h	PROCHOT Assertion Counter Limit Register

35.9.1 PROCHOT CUMULATIVE COUNT REGISTER

Whenever byte 0 of this register is read, bytes 1, 2 and 3 are held in a holding register. Subsequent reads of 8-bits or 16-bits to bytes other than byte 0 return data from the holding register, although the actual register may continue to be updated by hardware. A 32-bit read always returns the current register value.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:0	CUMULATIVE_PROCHOT_ACTIVE This register contains the current filtered PROCHOT Active Counter value. This register returns the value of the internal PROCHOT Active Counter. When PROCHOT_IN# transitions from low to high (from active to inactive) this register retains its most recent value. This register, as well as the internal PROCHOT Active Counter, are cleared to 0 when this register is copied into the PROCHOT Duty Cycle Count Register on the 1 to 0 transition of the internal Duty Cycle Counter.	R	00_0000h	RESET_SLP

35.9.2 PROCHOT DUTY CYCLE COUNT REGISTER

Whenever byte 0 of this register is read, bytes 1, 2 and 3 are held in a holding register. Subsequent reads of 8-bits or 16-bits to bytes other than byte 0 return data from the holding register, although the actual register may continue to be updated by hardware. A 32-bit read always returns the current register value.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:0	DUTY_CYCLE_COUNT The contents of the PROCHOT Cumulative Count Register is copied into this register when the PROCHOT Duty Cycle Period Register transitions from 1 to 0.	R	0h	RESET_SLP

© 2020 Microchip Technology Inc.

35.9.3 PROCHOT DUTY CYCLE PERIOD REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:0	DUTY_CYCLE_PERIOD This register defines the number of 100KHz periods required for a duty cycle measurement. It can be programmed for periods from one 100KHz period to 2^{24}-1 100KHz periods. As long as the PROCHOT device is enabled, the PROCHOT Period Counter repeatedly counts down from this value to 0. When the counter transitions from 1 to 0, the contents of the PROCHOT Cumulative Count Register are copied into the PROCHOT Duty Cycle Count Register. The status bit in the register is set and the counter is reloaded from this register. Setting this register to 0 disables duty cycle measurement. When this register is written, both the internal PROCHOT Active Counter and the PROCHOT Cumulative Count Register are reset to 0.	R/W	0b	RESET_PHOT

35.9.4 PROCHOT STATUS/CONTROL REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
15:12	Reserved	RES	-	-
11	PHOT_PERIOD This sticky status bit is set to ‘1b’ when the PROCHOT Period Counter transitions from ‘1b’ to ‘0b.’ It is cleared when written by software with a ‘1b’ or when a hardware reset event occurs. Writes of ‘0b’ have no affect.	R/WC	0b	RESET_SLP
10	PHOT_ASSERT This bit is set when the PROCHOT Assertion Counter Register value is greater than or equal to the PROCHOT Assertion Counter Limit Register value. It is cleared when written with a ‘1b,’ if the counter value is no longer violating the limit or by a hardware reset event. Writes of ‘0b,’ and writes of ‘1b’ when the counter is violating the limit, have no affect.	R/WC	0b	RESET_SLP
9:6	Reserved	RES	-	-
5	FILT_ENABLE This bit determines whether a digital filter eliminates pulses less than 3 100KHz pulses wide and potentially up to 4 100KHz pulses wide on the PROCHOT_IN# signal before PROCHOT_IN# is sampled by the Assertion counter or the Active counter. 1=PROCHOT_IN# input filtered 0=PROCHOT_IN# input not filtered	R/W	0b	RESET_PHOT

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4	PHOT_RESET Writing this self-clearing bit to one resets all the registers and logic in the PROCHOT Monitor block to its defined initial state. Writing a zero to this bit has no effect	R/W	0b	RESET_PHOT
3	PERIOD_ENABLE This bit determines whether or not an interrupt will be generated when the PHOT_PERIOD bit is set. 1=PROCHOT Duty Cycle Period Event interrupt enabled 0=PROCHOT Duty Cycle Period Event interrupt blocked	R/W	0b	RESET_PHOT
2	ASSERT_ENABLE This bit determines whether or not an interrupt will be generated when the PHOT_ASSERT bit is set. 1=PROCHOT Assertion Event interrupt enabled 0=PROCHOT Assertion Event interrupt blocked	R/W	0b	RESET_PHOT
1	PHOT_PIN When PHOT_ENABLE is ‘1’b, this bit reflects the state of the PROCHOT_IN# Pin input. When PHOT_ENABLE is ‘0’b, the pin is not monitored and this bit is not updated. 1=PROCHOT Pin is high 0=PROCHOT Pin is low	R	1b	RESET_SLP
0	PHOT_ENABLE This bit enables the PROCHOT Monitor logic. When Enable is 0, no status bits in this register or any of the counters in this block will be updated, although the registers can still be read by the EC. 1=PROCHOT Monitoring 0=PROCHOT Idle (default). This mode gates the clocks to the PROCHOT I/O block Contents of the registers are not affected.	R/W	0b	RESET_PHOT

© 2020 Microchip Technology Inc.

35.9.5 PROCHOT ASSERTION COUNTER REGISTER

Whenever byte 0 of this register is read, bytes 1, 2 and 3 are held in a holding register. Subsequent reads of 8-bits or 16-bits to bytes other than byte 0 return data from the holding register, although the actual register may continue to be updated by hardware. A 32-bit read always returns the current register value.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	ASSERTION_COUNTER The PROCHOT Assertion Counter is a 16-bit up-counter that is clocked by the 100KHz and is gated and reset by the PROCHOT_IN# input signal. If enabled, this counter increments when the PROCHOT_IN# input signal is active (low) and is reset to 0000h when the pin is inactive (high). This counter is used to measure a single PROCHOT assertion. This register allows the firmware to read the current count value. This counter is a saturating counter: When it reaches FFFFh, it stops counting rather than rolling over to 0000h.	R	0000h	RESET_SLP

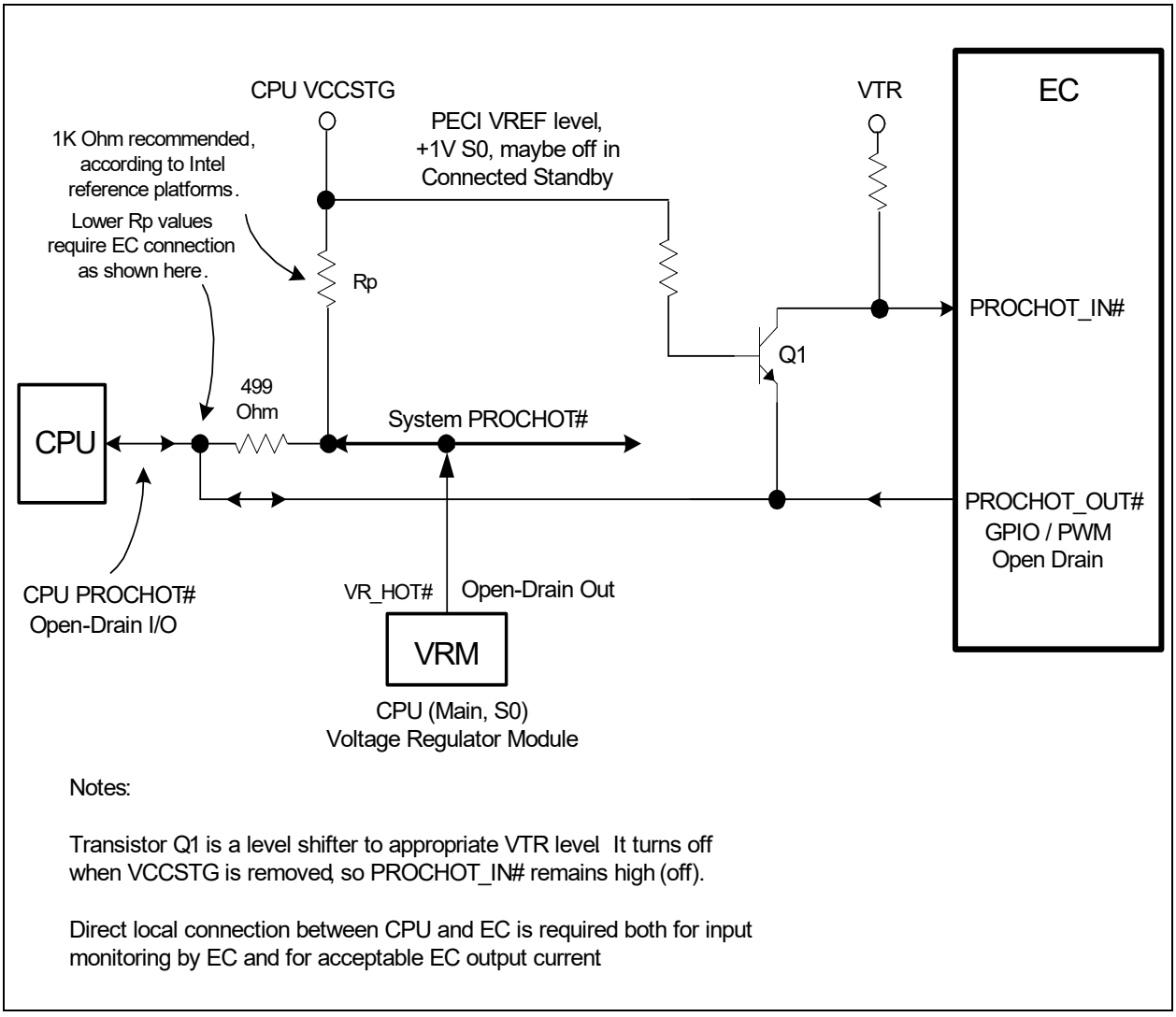
35.9.6 PROCHOT ASSERTION COUNTER LIMIT REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	ASSERTION_COUNT_LIMIT The PROCHOT Assertion Counter Limit register is compared to the 16-bit PROCHOT Assertion Counter. If the value in the PROCHOT Assertion counter is greater than or equal to the value in the limit register, then the PHOT_ASSERT bit contained in the PROCHOT Status/Control Register is set. In addition, an interrupt will be generated if the ASSERT_ENABLE bit in the PROCHOT Status/Control Register is set. A value of 0000h disables the comparison process.	R/w	0000h	RESET_PHOT

35.10 Usage Models

The recommended usage of this block is shown in Figure 35-5, "Recommended PROCHOT Connections".

FIGURE 35-5: RECOMMENDED PROCHOT CONNECTIONS



© 2020 Microchip Technology Inc.

36.0 QUAD SPI MASTER CONTROLLER

36.1 Overview

The Quad SPI Master Controller may be used to communicate with various peripheral devices that use a Serial Peripheral Interface, such as EEPROMS, DACs and ADCs. The controller can be configured to support advanced SPI Flash devices with multi-phase access protocols. Data can be transfered in Half Duplex, Single Data Rate, Dual Data Rate and Quad Data Rate modes. In all modes and all SPI clock speeds, the controller supports back-to-back reads and writes without clock stretching if internal bandwidth permits.

36.2 References

No references have been cited for this feature.

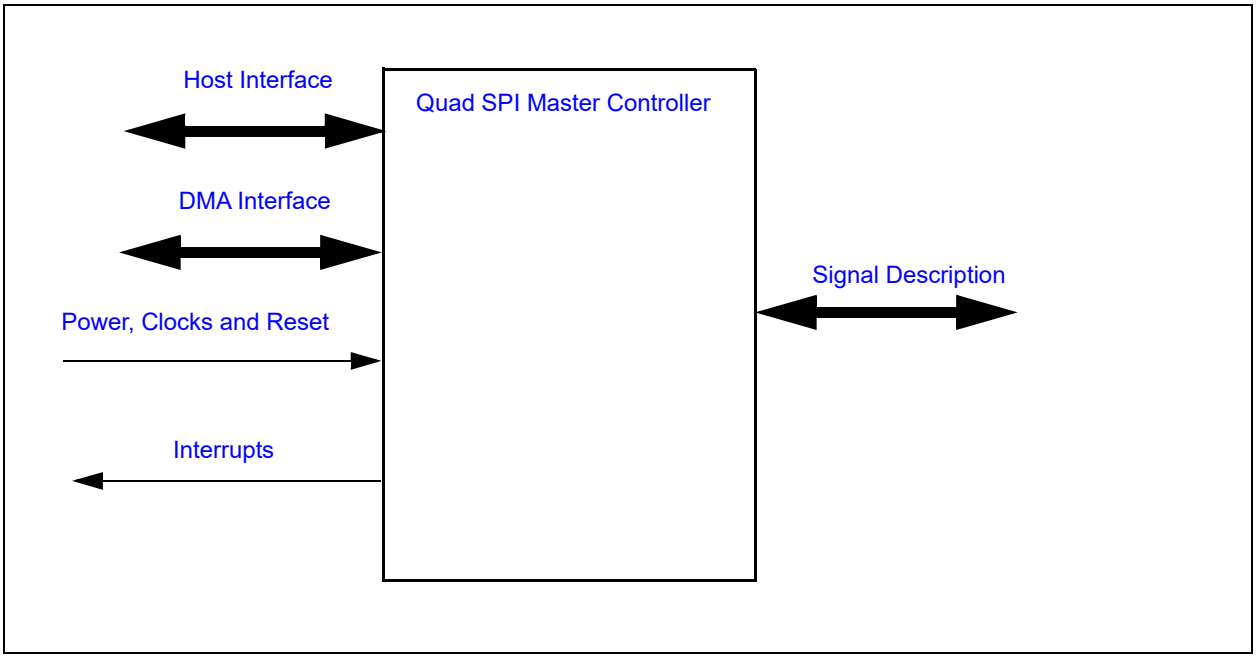
36.3 Terminology

No terminology for this block.

36.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 36-1: I/O DIAGRAM OF BLOCK



36.5 Signal Description

TABLE 36-1: EXTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
SPI_CLK	Output	SPI Clock output used to drive the SPCLK pin.
SPI_CS#	Output	SPI chip select
SPI_IO0	Input/Output	SPI Data pin 0. Also used as SPI_MOSI, Master-Out/Slave-In when the interface is used in Single wire mode
SPI_IO1	Input/Output	SPI Data pin 1. Also used as SPI_MISO, Master-In/Slave-Out when the interface is used in Single wire mode

© 2020 Microchip Technology Inc.

TABLE 36-1: EXTERNAL SIGNAL DESCRIPTION (CONTINUED)

Name	Direction	Description
SPI_IO2	Input/Output	SPI Data pin 2 when the SPI interface is used in Quad Mode. Also can be used by firmware as WP.
SPI_IO3	Input/Output	SPI Data pin 3 when the SPI interface is used in Quad Mode. Also can be used by firmware as HOLD.

36.6 Host Interface

The registers defined for the General Purpose Serial Peripheral Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

36.7 DMA Interface

This block is designed to communicate with the Internal DMA Controller.

Note: For a description of the Internal DMA Controller implemented in this design see Section 7.0, "Internal DMA Controller".

36.8 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

36.8.1 POWER

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

36.8.2 CLOCKS

Name	Description
48MHz	This is a clock source for the SPI clock generator.

36.8.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.QMSPI Status Register
RESET	This reset is generated if either the RESET_SYS is asserted or the SOFT_RESET is asserted.

36.9 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
QMSPI_INT	Interrupt generated by the Quad SPI Master Controller. Events that may cause the interrupt to be asserted are stored in the QMSPI Status Register.

36.10 Low Power Modes

The Quad SPI Master Controller is always in its lowest power state unless a transaction is in process. A transaction is in process between the time the START bit is written with a ‘1’ and the TRANSFER_DONE bit is set by hardware to ‘1’.

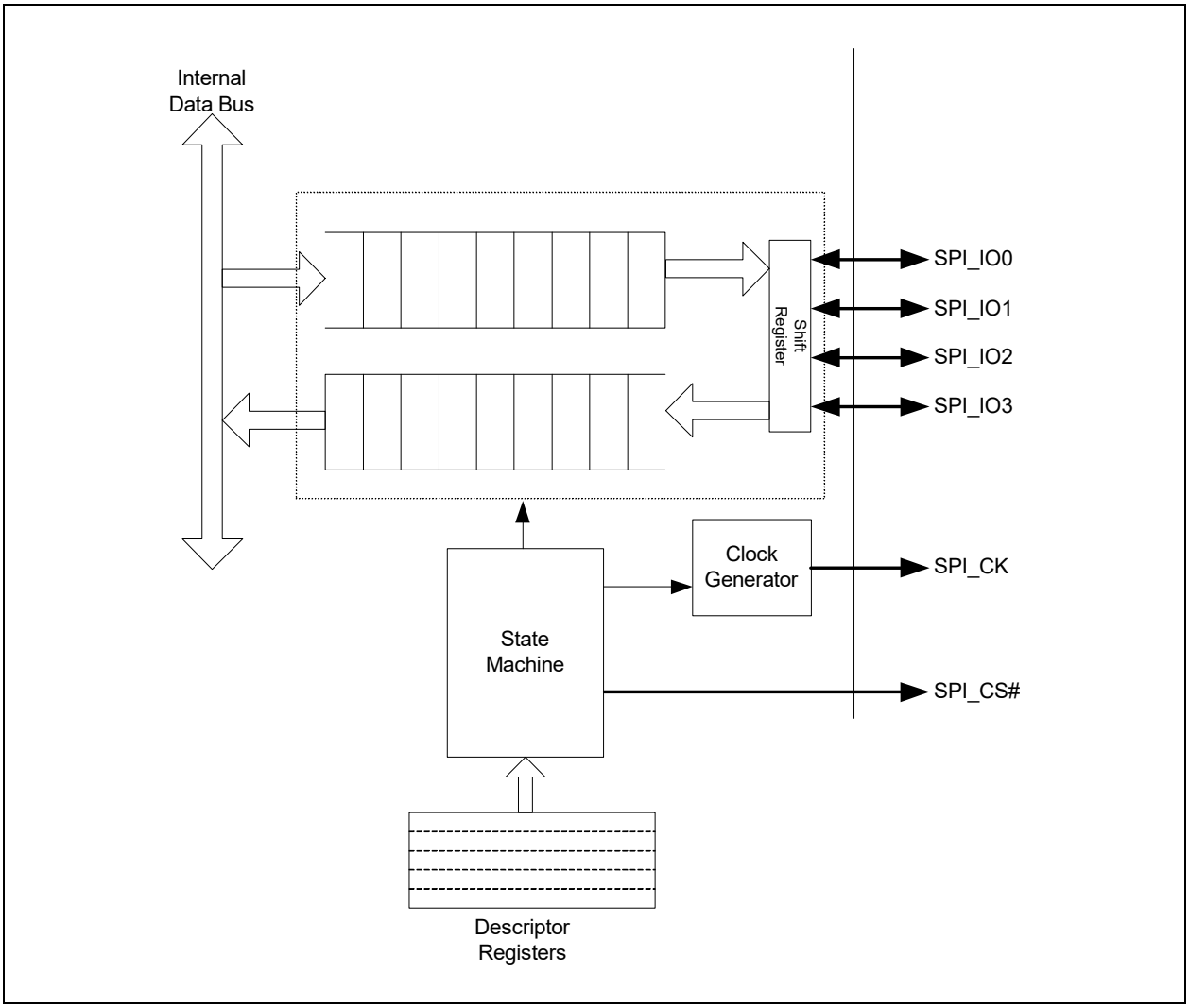
If the QMSPI SLEEP_ENABLE input is asserted, writes to the START bit are ignored and the Quad SPI Master Controller will remain in its lowest power state.

© 2020 Microchip Technology Inc.

36.11 Description

- Support for multiple SPI pin configurations
  - Single wire half duplex
  - Two wire full duplex
  - Two wire double data rate
  - Four wire quad data rate
- Separate FIFO buffers for Receive and Transmit
  - 8 byte FIFO depth in each FIFO
  - Each FIFO can be 1 byte, 2 bytes or 4 bytes wide
- Support for all four SPI clock formats
- Programmable SPI Clock generator, with clock polarity and phase controls
- Separate DMA support for Receive and Transmit data transfers
- Configurable interrupts, for errors, individual bytes, or entire transactions
- Descriptor Mode, in which a set of sixteen descriptor registers can configure the controller to autonomously perform multi-phase SPI data transfers
- Capable of wire speed transfers in all SPI modes and all configurable SPI clock rates (internal bus contention may cause clock stretching)

FIGURE 36-2: QUAD MASTER SPI BLOCK DIAGRAM



36.11.1 SPI CONFIGURATIONS MODES

- Half Duplex. All SPI data transfers take place on a single wire, SPI_IO0
- Full Duplex. This is the legacy SPI configuration, where all SPI data is transfered one bit at a time and data from the SPI Master to the SPI Slave takes place on SPI_MOSI (SPI_IO0) and at the same time data from the SPI Slave to the SPI Master takes place on SPI_MISO (SPI_IO1)
- Dual Data Rate. Data transfers between the SPI Master and the SPI Slave take place two bits at a time, using SPI_IO0 and SPI_IO1
- Quad Data Rate. Data transfers between the SPI Master and the SPI Slave take place four bits at a time, using all four SPI data wires, SPI_IO0, SPI_IO1, SPI_IO2 and SPI_IO3

36.11.2 SPI CONTROLLER MODES

- Manual. In this mode, firmware control all SPI data transfers byte at a time
- DMA. Firmware configures the SPI Master controller for characteristics like data width but the transfer of data between the FIFO buffers in the SPI controller and memory is controlled by the DMA controller. DMA transfers can take place from the Slave to the Master, from the Master to the Slave, or in both directions simultaneously
- Descriptor. Descriptor Mode extends the SPI Controller so that firmware can configure a multi-phase SPI transfer, in which each phase may have a different SPI bus width, a different direction, and a different length. For example, firmware can configure the controller so that a read from an advanced SPI Flash, which consists of a command phase, an address phase, a dummy cycle phase and the read phase, can take place as a single operation, with a single interrupt to firmware when the entire transfer is completed

36.11.3 SPI CLOCK

The SPI output clock is derived from the 48MHz, divided by a value programmed in the CLOCK_DIVIDE field of the QMSPI Mode Register. Sample frequencies are shown in the following table:

TABLE 36-2: EXAMPLE SPI FREQUENCIES

CLOCK_DIVIDE	**SPI Clock Frequency**
0	187.5 KHz
1	48 MHz
2	24 MHz
3	16 MHz
6	8 MHz
48	1 MHz
128	375 KHz
255	188.25 KHz

36.11.4 ERROR CONDITIONS

The Quad SPI Master Controller can detect some illegal configurations. When these errors are detected, an error is signaled via the PROGRAMMING_ERROR status bit. This bit is asserted when any of the following errors are detected:

- Both Receive and the Transmit transfers are enabled when the SPI Master Controller is configured for Dual Data Rate or Quad Data Rate
- Both Pull-up and Pull-down resistors are enabled on either the Receive data pins or the Transmit data pins
- The transfer length is programmed in bit mode, but the total number of bits is not a multiple of 2 (when the controller is configured for Dual Data Rate) or 4 (when the controller is configured for Quad Data Rate)
- Both the STOP bit and the START bits in the QMSPI Execute Register are set to ‘1’ simultaneously

© 2020 Microchip Technology Inc.

36.12 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Quad SPI Master Controller Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 36-3: REGISTER SUMMARY

Offset	Register Name
0h	QMSPI Mode Register
4h	QMSPI Control Register
8h	QMSPI Execute Register
Ch	QMSPI Interface Control Register
10h	QMSPI Status Register
14h	QMSPI Buffer Count Status Register
18h	QMSPI Interrupt Enable Register
1Ch	QMSPI Buffer Count Trigger Register
20h	QMSPI Transmit Buffer Register
24h	QMSPI Receive Buffer Register
28h	QMSPI Chip Select Timing Register
30h	QMSPI Description Buffer 0 Register
34h	QMSPI Description Buffer 1 Register
38h	QMSPI Description Buffer 2 Register
3Ch	QMSPI Description Buffer 3 Register
40h	QMSPI Description Buffer 4 Register
44h	QMSPI Description Buffer 5 Register
48h	QMSPI Description Buffer 6 Register
4Ch	QMSPI Description Buffer 7 Register
50h	QMSPI Description Buffer 8 Register
54h	QMSPI Description Buffer 9 Register
58h	QMSPI Description Buffer 10 Register
5Ch	QMSPI Description Buffer 11 Register
60h	QMSPI Description Buffer 12 Register
64h	QMSPI Description Buffer 13 Register
68h	QMSPI Description Buffer 14 Register
6Ch	QMSPI Description Buffer 15 Register
B0	Test

36.12.1 QMSPI MODE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
24:16	CLOCK_DIVIDE The SPI clock divide in number of system clocks. A value of 1 divides the master clock by 1, a value of 255 divides the master clock by 255. A value of 0 divides the master clock by 256. See Table 36-2, "Example SPI Frequencies" for examples.	R/W	0h	RESET
15:14	Reserved	RES	-	-

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
13:12	CHIP_SELECT Selects which Chip Select line is active. The non-active CS line is driven high. 00=Chip Select 0 01=Chip Select 1 1x=unused.	R/W	0h	RESET
11	Reserved	RES	-	-
10	CHPA_MISO If CPOL=1: 1=Data are captured on the rising edge of the SPI clock 0=Data are captured on the falling edge of the SPI clock If CPOL=0: 1=Data are captured on the falling edge of the SPI clock 0=Data are captured on the rising edge of the SPI clock Application Notes: Common SPI Mode configurations: Common SPI Modes require the CHPA_MISO and CHPA_MOSI programmed to the same value. E.g., - Mode 0: CPOL=0; CHPA_MISO=0; CHPA_MOSI=0 - Mode 3: CPOL=1; CHPA_MISO=1; CHPA_MOSI=1 Alternative SPI Mode configurations When configured for quad mode, applications operating at 48MHz may find it difficult to meet the minimum setup timing using the default Mode 0. It is recommended to configure the Master to sample and change data on the same edge when operating at 48MHz as shown in these examples. E.g, - Mode 0: CPOL=0; CHPA_MISO=1; CHPA_MOSI=0 - Mode 3: CPOL=1; CHPA_MISO=0; CHPA_MOSI=1	R/W	0h	RESET
9	CHPA_MOSI If CPOL=1: 1=Data changes on the falling edge of the SPI clock 0=Data changes on the rising edge of the SPI clock If CPOL=0: 1=Data changes on the rising edge of the SPI clock 0=Data changes on the falling edge of the SPI clock	R/W	0h	RESET
8	CPOL Polarity of the SPI clock line when there are no transactions in process. 1=SPI Clock starts High 0=SPI Clock starts Low	R/W	0h	RESET
7:3	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	SAF DMA Mode This mode enables the H/W to allow a DMA to access the part with accesses that are not a multiple of 4 bytes. 0 = Standard DMA functionality 1 = SAF DMA Mode: Non-standard DMA functionality with arbitrary (unaligned) sizes and FIFO underflow allowed.	R/W	0h	RESET
1	SOFT_RESET Writing this bit with a '1' will reset the Quad SPI block. It is self-clearing.	W	0h	RESET_SYS
0	ACTIVATE 1=Enabled. The block is fully operational 0=Disabled. Clocks are gated to conserve power and the output signals are set to their inactive state	R/W	0h	RESET

36.12.2 QMSPI CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:17	TRANSFER_LENGTH The length of the SPI transfer. The count is in bytes or bits, depending on the value of TRANSFER_UNITS. A value of '0' means an infinite length transfer.	R/W	0h	RESET
16	DESCRIPTION_BUFFER_ENABLE This enables the Description Buffers to be used. 1=Description Buffers in use. The first buffer is defined in DESCRIPTION_BUFFER_POINTER 0=Description Buffers disabled	R/W	0h	RESET
15:12	DESCRIPTION_BUFFER_POINTER This field selects the first buffer used if Description Buffers are enabled.	R/W	0h	RESET
11:10	TRANSFER_UNITS 3=TRANSFER_LENGTH defined in units of 16-byte segments 2=TRANSFER_LENGTH defined in units of 4-byte segments 1=TRANSFER_LENGTH defined in units of bytes 0=TRANSFER_LENGTH defined in units of bits	R/W	0h	RESET
9	CLOSE_TRANSFER_ENABLE This selects what action is taken at the end of a transfer. When the transaction closes, the Chip Select de-asserts, the SPI interface returns to IDLE and the DMA transfer terminates. When Description Buffers are in use this bit must be set only on the Last Buffer. 1=The transaction is terminated 0=The transaction is not terminated	R/W	0h	RESET
8:7	RX_DMA_ENABLE This bit enables DMA support for Receive Transfer. If enabled, DMA will be requested to empty the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Receive Buffer must be emptied by firmware	R/W	0h	RESET
6	RX_TRANSFER_ENABLE This bit enables the receive function of the SPI interface. 1=Receive is enabled. Data received from the SPI Slave is stored in the Receive Buffer 0=Receive is disabled	R/W	0h	RESET

© 2020 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5:4	TX_DMA_ENABLE This bit enables DMA support for Transmit Transfer. If enabled, DMA will be requested to fill the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Transmit Buffer must be emptied by firmware	R/W	0h	RESET
3:2	TX_TRANSFER_ENABLE This field bit selects the transmit function of the SPI interface. 3=Transmit Enabled in 1 Mode. The MOSI or IO Bus will send out only 1's. The Transmit Buffer will not be used 2=Transmit Enabled in 0 Mode. The MOSI or IO Bus will send out only 0's. The Transmit Buffer will not be used. 1=Transmit Enabled. Data will be fetched from the Transmit Buffer and sent out on the MOSI or IO Bus. 0=Transmit is Disabled. Not data is sent. This will cause the MOSI be to be undriven, or the IO bus to be undriven if Receive is also disabled.	R/W	0h	RESET
1:0	INTERFACE_MODE This field sets the transmission mode. If this field is set for Dual Mode or Quad Mode then either TX_TRANSFER_ENABLE or RX_TRANSFER_ENABLE **must** be 0. 3=Reserved 2=Quad Mode 1=Dual Mode 0=Single/Duplex Mode	R/W	0h	RESET

36.12.3 QMSPI EXECUTE REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2	CLEAR_DATA_BUFFER Writing a ‘1’ to this bit will clear out the Transmit and Receive FIFOs. Any data stored in the FIFOs is discarded and all count fields are reset. Writing a ‘0’ to this bit has no effect. This bit is self-clearing.	W	0h	RESET
1	STOP Writing a ‘1’ to this bit will stop any transfer in progress at the next byte boundary. Writing a ‘0’ to this bit has no effect. This bit is self-clearing. This bit must not be set to ‘1’ if the field START in this register is set to ‘1’.	W	0h	RESET

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	START Writing a '1' to this bit will start the SPI transfer. Writing a '0' to this bit has no effect. This bit is self-clearing. This bit must not be set to '1' if the field STOP in this register is set to '1'.	W	0h	RESET

36.12.4 QMSPI INTERFACE CONTROL REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7	PULLUP_ON_NOT_DRIVEN 1=Enable pull-up resistors on Transmit pins while the pins are not driven 0=No pull-up resistors enabled ion Transmit pins	R/W	0h	RESET
6	PULLDOWN_ON_NOT_DRIVEN 1=Enable pull-down resistors on Transmit pins while the pins are not driven 0=No pull-down resistors enabled ion Transmit pins	R/W	0h	RESET
5	PULLUP_ON_NOT_SELECTED 1=Enable pull-up resistors on Receive pins while the SPI Chip Select signal is not asserted 0=No pull-up resistors enabled on Receive pins	R/W	0h	RESET
4	PULLDOWN_ON_NOT_SELECTED 1=Enable pull-down resistors on Receive pins while the SPI Chip Select signal is not asserted 0=No pull-down resistors enabled on Receive pins	R/W	0h	RESET
3	HOLD_OUT_ENABLE 1=HOLD SPI Output Port is driven 0=HOLD SPI Output Port is not driven	R/W	0h	RESET
2	HOLD_OUT_VALUE This bit sets the value on the HOLD SPI Output Port if it is driven. 1=HOLD is driven to 1 0=HOLD is driven to 0	R/W	0h	RESET
1	WRITE_PROTECT_OUT_ENABLE 1=WRITE PROTECT SPI Output Port is driven 0=WRITE PROTECT SPI Output Port is not driven	R/W	0h	RESET

© 2020 Microchip Technology Inc.

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	WRITE_PROTECT_OUT_VALUE This bit sets the value on the WRITE PROTECT SPI Output Port if it is driven. 1=WRITE PROTECT is driven to 1 0=WRITE PROTECT is driven to 0	R/W	0h	RESET

36.12.5 QMSPI STATUS REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Reserved	RES	-	-
27:24	CURRENT_DESCRIPTION_BUFFER This field shows the Description Buffer currently active. This field has no meaning if Description Buffers are not enabled.	R	0h	RESET
23:17	Reserved	RES	-	-
16	TRANSFER_ACTIVE 1=A transfer is currently executing 0=No transfer currently in progress	R	0h	RESET
15	RECEIVE_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to write to a full Receive Buffer) 0=No stalls occurred	R/WC	0h	RESET
14	RECEIVE_BUFFER_REQUEST This status is asserted if the Receive Buffer reaches a high water mark established by the RECEIVE_BUFFER_TRIGGER field. 1=RECEIVE_BUFFER_COUNT is greater than or equal to RECEIVE_BUFFER_TRIGGER 0=RECEIVE_BUFFER_COUNT is less than RECEIVE_BUFFER_TRIGGER	R/WC	0h	RESET
13	RECEIVE_BUFFER_EMPTY 1=The Receive Buffer is empty 0=The Receive Buffer is not empty	R	1h	RESET
12	RECEIVE_BUFFER_FULL 1=The Receive Buffer is full 0=The Receive Buffer is not full	R	0h	RESET
11	TRANSMIT_BUFFER_STALL 1=The SPI interface had been stalled due to a flow issue (an attempt by the interface to read from an empty Transmit Buffer) 0=No stalls occurred	R/WC	0h	RESET

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TRANSMIT_BUFFER_REQUEST This status is asserted if the Transmit Buffer reaches a high water mark established by the TRANSMIT_BUFFER_TRIGGER field. 1=TRANSMIT_BUFFER_COUNT is less than or equal to TRANSMIT_BUFFER_TRIGGER 0=TRANSMIT_BUFFER_COUNT is greater than TRANSMIT_BUFFER_TRIGGER	R/WC	0h	RESET
9	TRANSMIT_BUFFER_EMPTY 1=The Transmit Buffer is empty 0=The Transmit Buffer is not empty	R	1h	RESET
8	TRANSMIT_BUFFER_FULL 1=The Transmit Buffer is full 0=The Transmit Buffer is not full	R	0h	RESET
7:5	Reserved	RES	-	-
4	PROGRAMMING_ERROR This bit if a programming error is detected. Programming errors are listed in Section 36.11.4, "Error Conditions". 1=Programming Error detected 0=No programming error detected	R/WC	0h	RESET
3	RECEIVE_BUFFER_ERROR 1=Underflow error occurred (attempt to read from an empty Receive Buffer) 0=No underflow occurred	R/WC	0h	RESET
2	TRANSMIT_BUFFER_ERROR 1=Overflow error occurred (attempt to write to a full Transmit Buffer) 0=No overflow occurred	R/WC	0h	RESET
1	DMA_COMPLETE This field has no meaning if DMA is not enabled. This bit will be set to ‘1’ when the DMA controller asserts the DONE signal to the SPI controller. This occurs either when the SPI controller has closed the DMA transfer, or the DMA channel has completed its count. If both Transmit and Receive DMA transfers are active, then this bit will only assert after both have completed. If CLOSE_TRANSFER_ENABLE is enabled, DMA_COMPLETE and TRANSFER_COMPLETE will be asserted simultaneously. This status is not inhibited by the description buffers, so it can fire on all valid description buffers while operating in that mode. 1=DMA completed 0=DMA not completed	R/WC	0h	RESET

© 2020 Microchip Technology Inc.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	TRANSFER_COMPLETE In Manual Mode (neither DMA nor Description Buffers are enabled), this bit will be set to '1' when the transfer matches TRANSFER_LENGTH. If DMA Mode is enabled, this bit will be set to '1' when DMA_COMPLETE is set to '1'. In Description Buffer Mode, this bit will be set to '1' only when the Last Buffer completes its transfer. In all cases, this bit will be set to '1' if the STOP bit is set to '1' and the controller has completed the current 8 bits being copied. 1=Transfer completed 0=Transfer not complete	R/WC	0h	RESET

36.12.6 QMSPI BUFFER COUNT STATUS REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RECEIVE_BUFFER_COUNT This is a count of the number of bytes currently valid in the Receive Buffer.	R	0h	RESET
15:0	TRANSMIT_BUFFER_COUNT This is a count of the number of bytes currently valid in the Transmit Buffer.	R	0h	RESET

36.12.7 QMSPI INTERRUPT ENABLE REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14	RECEIVE_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
13	RECEIVE_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	1h	RESET
12	RECEIVE_BUFFER_FULL_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_FULL is asserted 0=Disable the interrupt	R/W	0h	RESET
11	Reserved	RES	-	-

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	TRANSMIT_BUFFER_REQUEST_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_REQUEST is asserted 0=Disable the interrupt	R/W	0h	RESET
9	TRANSMIT_BUFFER_EMPTY_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_EMPTY is asserted 0=Disable the interrupt	R/W	0h	RESET
8	TRANSMIT_BUFFER_FULL_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_FULL is asserted 0=Disable the interrupt	R/W	0h	RESET
7:5	Reserved	RES	-	-
4	PROGRAMMING_ERROR_ENABLE 1=Enable an interrupt if PROGRAMMING_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
3	RECEIVE_BUFFER_ERROR_ENABLE 1=Enable an interrupt if RECEIVE_BUFFER_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
2	TRANSMIT_BUFFER_ERROR_ENABLE 1=Enable an interrupt if TRANSMIT_BUFFER_ERROR is asserted 0=Disable the interrupt	R/W	0h	RESET
1	DMA_COMPLETE_ENABLE 1=Enable an interrupt if DMA_COMPLETE is asserted 0=Disable the interrupt	R/W	0h	RESET
0	TRANSFER_COMPLETE_ENABLE 1=Enable an interrupt if TRANSFER_COMPLETE is asserted 0=Disable the interrupt	R/W	0h	RESET

36.12.8 QMSPI BUFFER COUNT TRIGGER REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RECEIVE_BUFFER_TRIGGER An interrupt is triggered if the RECEIVE_BUFFER_COUNT field is greater than or equal to this value. A value of ‘0’ disables the interrupt.	R/W	0h	RESET
15:0	TRANSMIT_BUFFER_TRIGGER An interrupt is triggered if the TRANSMIT_BUFFER_COUNT field is less than or equal to this value. A value of ‘0’ disables the interrupt.	R/W	0h	RESET

© 2020 Microchip Technology Inc.

36.12.9 QMSPI TRANSMIT BUFFER REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	TRANSMIT_BUFFER Writes to this register store data to be transmitted from the SPI Master to the external SPI Slave. Writes to this block will be written to the Transmit FIFO. A 1 Byte write fills 1 byte of the FIFO. A Word write fills 2 Bytes and a Doubleword write fills 4 bytes. The data must always be aligned to the bottom most byte (so 1 byte write is on bits [7:0] and Word write is on [15:0]). An overflow condition,TRANSMIT_BUFFER_ERROR will happen, if a write to a full FIFO occurs. Write accesses to this register increment the TRANSMIT_BUFFER_COUNT field.	W	0h	RESET

36.12.10 QMSPI RECEIVE BUFFER REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	RECEIVE_BUFFER Buffer that stores data from the external SPI Slave device to the SPI Master (this block), which is received over MISO or IO. Reads from this register will empty the Rx FIFO. A 1 Byte read will have valid data on bits [7:0] and a Word read will have data on bits [15:0]. It is possible to request more data than the FIFO has (underflow condition), but this will cause an error (RECEIVE_BUFFER_ERROR). Read accesses to this register decrement the RECEIVE_BUFFER_COUNT field.	R	0h	RESET

36.12.11 QMSPI CHIP SELECT TIMING REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	DELAY_CS_OFF_TO_CS_ON This selects the number of system clock cycles between CS deassertion to CS assertion. This is the minimum pulse width of CS deassertion. **Note:** this field delays the start of the next transaction, it does not delay the status of the current transaction.	R/W	06h	RESET
23:20	Reserved	RES	0h	RESET

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
19:16	DELAY_LAST_DATA_HOLD This selects the number of system clock cycles between CS deassertion to the data ports for WP and HOLD switching from input to output. This is only used if the WP/HOLD functions are in use and only on IO2/WP and IO3/HOLD pins.	R/W	6h	RESET
15:12	Reserved	RES	0h	RESET
11:8	DELAY_CLK_STOP_TO_CS_OFF This selects the number of system clock cycles between the last clock edge and the deassertion of CS.	R/W	4h	RESET
7:4	Reserved	RES	0h	RESET
3:0	DELAY_CS_ON_TO_CLOCK_START This selects the number of system clock cycles between CS assertion to the start of the SPI Clock. An additional ½ SPI Clock delay is inherently added to allow pre-set-up of the data ports.	R/W	6h	RESET

36.12.12 QMSPI DESCRIPTION BUFFER 0 REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:17	TRANSFER_LENGTH The length of the SPI transfer. The count is in bytes or bits, depending on the value of TRANSFER_LENGTH_BITS. A value of '0' means an infinite length transfer.	R/W	0h	RESET
16	DESCRIPTION_BUFFER_LAST If this bit is '1' then this is the last Description Buffer in the chain. When the transfer described by this buffer completes the TRANSFER_COMPLETE status will be set to '1'. If this bit is '0', then this is not the last buffer in use. When the transfer completes the next buffer will be activated, and no additional status will be asserted.	R/W	0h	RESET
15:12	DESCRIPTION_BUFFER_NEXT_POINTER This defines the next buffer to be used if Description Buffers are enabled and this is not the last buffer. This can point to the current buffer, creating an infinite loop.	R/W	0h	RESET
11:10	TRANSFER_UNITS 3=TRANSFER_LENGTH defined in units of 16-byte segments 2=TRANSFER_LENGTH defined in units of 4-byte segments 1=TRANSFER_LENGTH defined in units of bytes 0=TRANSFER_LENGTH defined in units of bits	R/W	0h	RESET

© 2020 Microchip Technology Inc.

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9	CLOSE_TRANFSER_ENABLE This selects what action is taken at the end of a transfer. This bit must be set only on the Last Buffer. 1=The transfer is terminated. The Chip Select de-asserts, the SPI interface returns to IDLE and the DMA interface completes the transfer. 0=The transfer is not closed. Chip Select remains asserted and the DMA interface and the SPI interface remain active	R/W	0h	RESET
8:7	RX_DMA_ENABLE This bit enables DMA support for Receive Transfer. If enabled, DMA will be requested to empty the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Receive Buffer must be emptied by firmware	R/W	0h	RESET
6	RX_TRANSFER_ENABLE This bit enables the receive function of the SPI interface. 1=Receive is enabled. Data received from the SPI Slave is stored in the Receive Buffer 0=Receive is disabled	R/W	0h	RESET
5:4	TX_DMA_ENABLE This bit enables DMA support for Transmit Transfer. If enabled, DMA will be requested to fill the FIFO until either the interface reaches TRANSFER_LENGTH or the DMA sends a termination request. The size defined here must match DMA programmed access size. 1=DMA is enabled.and set to 1 Byte 2=DMA is enabled and set to 2 Bytes 3=DMA is enabled and set to 4 Bytes 0=DMA is disabled. All data in the Transmit Buffer must be emptied by firmware	R/W	0h	RESET
3:2	TX_TRANSFER_ENABLE This field bit selects the transmit function of the SPI interface. 3=Transmit Enabled in 1 Mode. The MOSI or IO Bus will send out only 1's. The Transmit Buffer will not be used 2=Transmit Enabled in 0 Mode. The MOSI or IO Bus will send out only 0's. The Transmit Buffer will not be used. 1=Transmit Enabled. Data will be fetched from the Transmit Buffer and sent out on the MOSI or IO Bus. 0=Transmit is Disabled. No data is sent. This will cause the MOSI be to be undriven, or the IO bus to be undriven if Receive is also disabled.	R/W	0h	RESET

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1:0	INTERFACE_MODE This field sets the transmission mode. If this field is set for Dual Mode or Quad Mode then either TX_TRANSFER_ENABLE or RX_TRANSFER_ENABLE **must** be 0. 3=Reserved 2=Quad Mode 1=Dual Mode 0=Single/Duplex Mode	R/W	0h	RESET

36.12.13 QMSPI DESCRIPTION BUFFER 1 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.14 QMSPI DESCRIPTION BUFFER 2 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.15 QMSPI DESCRIPTION BUFFER 3 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.16 QMSPI DESCRIPTION BUFFER 4 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.17 QMSPI DESCRIPTION BUFFER 5 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.18 QMSPI DESCRIPTION BUFFER 6 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.19 QMSPI DESCRIPTION BUFFER 7 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.20 QMSPI DESCRIPTION BUFFER 8 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.21 QMSPI DESCRIPTION BUFFER 9 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.22 QMSPI DESCRIPTION BUFFER 10 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.23 QMSPI DESCRIPTION BUFFER 11 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.24 QMSPI DESCRIPTION BUFFER 12 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.25 QMSPI DESCRIPTION BUFFER 13 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

© 2020 Microchip Technology Inc.

36.12.26 QMSPI DESCRIPTION BUFFER 14 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

36.12.27 QMSPI DESCRIPTION BUFFER 15 REGISTER

The format for this register is the same as the format of the QMSPI Description Buffer 0 Register.

37.0 SERIAL PERIPHERAL INTERFACE (SPI) SLAVE

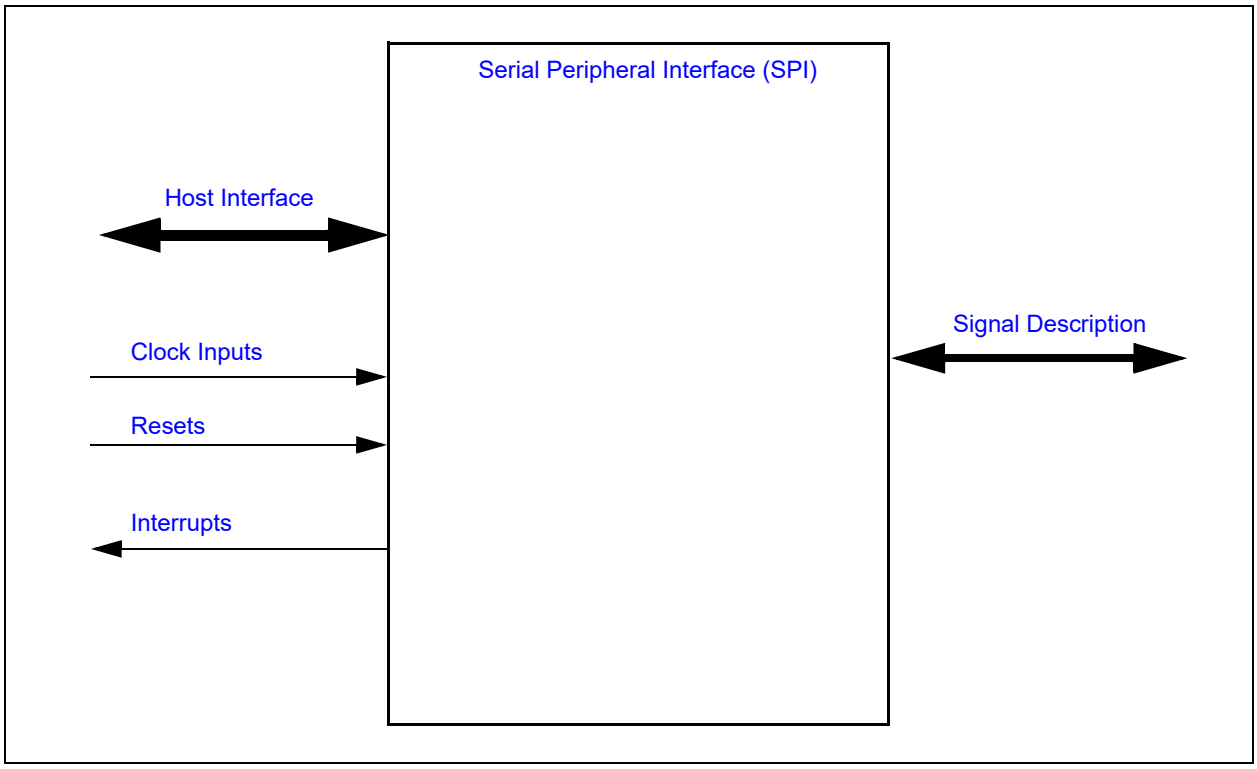
37.1 Introduction

The Serial Peripheral Interface (SPI) Slave provides a standard run-time mechanism for the SPI Master to communicate with the Embedded Controller (EC) and other logical components. The SPI includes 2 byte-addressable registers in the SPI Master's address space, as well as by the EC. The SPI slave can be used by the SPI Master to access bytes of memory designated by the EC without requiring any assistance from the EC. The SPI slave can also be used by the SPI Master to access FIFO data without requiring any assistance from the EC.

37.2 Interface

This block is designed to be accessed externally and internally via a register interface.

FIGURE 37-1: I/O DIAGRAM OF BLOCK



37.3 Host Interface

The registers defined for the Serial Peripheral Interface (SPI) Slave are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

© 2020 Microchip Technology Inc.

37.4 Signal Description

The registers defined for the Serial Peripheral Interface (SPI) Slave are accessible by the SPI Master and the Embedded Controller (EC) as indicated in Section 37.9, "Configuration and Runtime Registers".

TABLE 37-1: SPI SLAVE PORTS

Name	Direction	Description
SLV_SPI_SCLK	INPUT	Clock signal from SPI Master.
SLV_SPI_CS#	INPUT	Chip Select for SPI Slave from Master.
SLV_SPI_IOx	INOUT	SPI Slave data pins to Master. This is a 4 bit data bus.
SLV_SPI_MSTR_INT	OUTPUT	This interrupt is for the SPI Master and is asynchronous to clock.

37.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

37.5.1 POWER DOMAINS

This block works on the VTR_CORE power domain as listed in Table 37-2 below.

TABLE 37-2: POWER SOURCES

Name	Description
VTR_CORE	The logic and registers implemented in this block reside on this single power well.

37.5.2 CLOCK INPUTS

This block has two clock inputs as listed in Table 37-3, "Clock Signals". For both the 48MHz and SPI_CLK domain crossing low latency clock domain crossing synchronizers are used. Both clock are treated as asynchronous to each other.

TABLE 37-3: CLOCK SIGNALS

Name	Description
SLV_SPI_SCLK	This is the SPI clock from the Master. All SPI transfers take place with respect to this clock.
48MHz	Clock used for EC register access

37.5.3 RESETS

Resets to the SPI Slave are from the system reset which will reset the entire block or a write to the self-clearing reset bit. The SPI chip select is treated as reset to the SPI Interface.

TABLE 37-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all the logic and register in this block.

37.6 Interrupts

This section lists the Interrupt pins from this block. Refer to Table 37-5 below for details.

TABLE 37-5: SYSTEM INTERRUPTS

Source	Description
SLV_SPI_MSTR_INT	This interrupt is for the SPI Master and is asynchronous to clock.

The SPI_HOST_INTERRUPT signal is asserted when any of the enabled interrupt in SPI Interrupt Enable Register is set and the corresponding condition for interrupt assertion is met.

TABLE 37-6: EC INTERRUPTS

Source	Description
SPI_EC_INTERRUPT	This interrupt is synchronous to the EC clock domain and is generated if the interrupt is enabled.
SPI_ASYNC_WAKE	Clock wake is a asynchronous output signal.

The SPI_EC_INTERRUPT signal is asserted when any of the enabled interrupt in EC Interrupt Enable Register is set and the corresponding condition for interrupt assertion is met.

37.7 Low Power Modes

The Serial Peripheral Interface (SPI) Slave automatically enters low power mode when no transaction is targeted to it. The SPI Slave is a wake interface; at de-assert of chip select a wake event will occur.

37.8 Description

Some of the features of this block are listed below

1. SPI Slave module supports Simple Mode (SM) and Advanced Mode (AM).

Simple Mode

1. SPI Slave module is Wake Capable.
2. SPI Slave module supports Single Wire and Mode 0 / Mode 3 transfers in this mode. SPI Communication Configuration Register configuration settings are ignored.
3. SPI Slave module only supports byte transfer with Undefined length in this mode.
4. SPI Slave module supports only one window with programmable Memory Base Address0 Register, Memory Write Limit0 Register for write data and another window Memory Base Address1 Register, Memory Read Limit1 Register for reads.
5. SPI Slave module Interrupt are don't care in Simple Mode. GPIO Interrupt for SLV_SPI_CS# toggle should be used in this mode.
6. SPI Slave module does not support register access from SPI Master in this mode.
7. SPI Slave module supports Full Duplex mode.
8. This mode uses an application code (Software) controlled data flow.
9. SPI Slave module uses byte counter to count the number of Bytes received or transmitted.
10. The Max packet length of an undefined length transfer is 32K Byte but recommended the master limits the size according to the limits placed by the EC. The data above the value written in Memory Write Limit0 Register will be ignored for writes and data above the value read from Memory Read Limit1 Register will be invalid data for reads.
11. The wake up timing of the SPI Slave have to be accounted for by the SPI Master.

Advanced Mode

1. SPI Slave module is Wake Capable.
2. SPI Slave module supports Single / Quad Wire and Mode 0 / Mode 3 transfers.
3. SPI Slave module supports programmable number of turn-around (TAR) cycles for Quad mode.
4. SPI Slave module supports standalone 8,16, 32 bit or block 2-8 DWords read/write memory accesses with error response and minimum possible latency.
5. SPI Slave module supports Base Address Enable and Memory Access Window of 256 – 4K bytes and error if disabled or out of range.
6. SPI Slave module supports Poll command for quick read of status register.
7. SPI Slave module supports Status Register which is not transaction specific. There are a set of flags for errors or done transactions for Master or System to be aware. Please refer to Section 37.9.2, "SPI Slave Status Register" and Section 37.9.3, "SPI EC Status Register" for details.
8. SPI Slave module supports important set of commands (described in Section 37.10, "Commands Supported") to allow direct access to the SPI Slave's registers (referred to as SREG accesses) with 8, 16, 32 bit size. These registers can be locked by system, with error response support.

© 2020 Microchip Technology Inc.

9. SPI Slave module supports accesses to external register bank situated in the SLV_SPI_SCLK domain with 8-bit Read and Write Commands with test entry mechanism support.
10. SPI Slave module supports programmable number of wait cycles for transactions between SLV_SPI_SCLK and 48MHz.
11. SPI Slave module supports programmable interrupt enables for both the system on chip and the SPI Interface.
12. SPI Slave module supports separate interrupt (SLV_SPI_MSTR_INT) to the SPI Master.

The Serial Peripheral Interface (SPI) Slave is composed of register interface, Memory interface and a mailbox interface. Fully on the SPI CLK domain, the SPI_IF's function is to transmit and receive data to and from the SPI Master using the SPI protocol. The block captures the incoming command and along with the dispatcher units determine if the rest of the command can be accepted.

37.8.1 SPI CLOCK FREQUENCY SUPPORTED

The Table 37-7, "Supported SLV_SPI_SCLK clock Frequency" lists the supported SPI clock frequency of this block. Running the chip outside the specified SLV_SPI_SCLK clock frequency may cause unspecified results.

TABLE 37-7: SUPPORTED SLV_SPI_SCLK CLOCK FREQUENCY

SPI Slave Mode	Supported SLV_SPI_SCLK Frequency
Single Mode	1MHz to 48 MHz
Advanced Mode Byte Command (Single Wire Interface)	1MHz to 48 MHz
Advanced Mode Byte Command (Quad Wire Interface)	1MHz to 32 MHz
Advanced Mode DWORD Command (Single Wire Interface)	1MHz to 48 MHz
Advanced Mode DWORD Command (DWORD Wire Interface)	1MHz to 48 MHz

37.8.2 EMBEDDED MEMORY MAP

Each Serial Peripheral Interface (SPI) Slave provides direct access for the SPI Master into two windows of 32K Byte each in the EC's internal address space. This mapping is programmable through a register, programmed during boot up:

The Base addresses, the Read limits and the Write limits are defined by registers that are in the EC address space and cannot be written by the SPI Master if the register is locked. In each region, the Read limit need not be greater than the Write limit. The regions can be contiguous or overlapping.

Each window into the EC memory can be as large as 32k bytes in the 32-bit internal address space. In Advanced Mode, the register Memory Base Address0 Register defines the address that SPI Master can write/read data to in EC space and register Memory Base Address1 Register defines the second set of address that the SPI Master can write/read data from the EC space.

In Simple Mode, the register Memory Base Address0 Register defines the address that SPI Master can write data to in EC space and register Memory Base Address1 Register defines the second set of address that the SPI Master can read data from the EC space.

37.8.3 EC AND SPI MASTER DATA REGISTERS

There are 14 32-bit EC registers. The lock register determines the type of access for SPI Master. Once the register is locked, the SPI Master can only read the data from these registers.

37.9 Configuration and Runtime Registers

The registers listed in the below Table 37-8, "Register Summary" table are for a single instance of the Serial Peripheral Interface (SPI) Slave. EC access for each register listed in this table is defined as an offset in the EC address space to the SPI Slave Block's Base Address, as defined by the instance's Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the Serial Peripheral Interface (SPI) Slave shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the "Offset" column in Table 37-8, "Register Summary".

The Serial Peripheral Interface (SPI) Slave can be accessed from the internal embedded controller (EC) and SPI Master.

Note:	SPI Master only has read access to these register when lock bit is set or is totally hidden if Mask EC Registers is set.
Note:	If Mask EC Registers bit is set in the System Configuration Register then only SPI Slave Status Register, SPI Master-to-EC Mailbox Register, EC-to-SPI Master Mailbox Register and SPI Interrupt Enable Register are accessible to the SPI Master.
Note:	Serial Peripheral Interface (SPI) Slave should be enabled by firmware only after the PLL has locked.
Note:	The SPI Master has to wait for the wake up timing requirements from heavy sleep after asserting SLV_SPI_CS# and before initiating the read / write transfer. This time is required for the PLL to come up and lock and all blocks to be functioning properly.

TABLE 37-8: REGISTER SUMMARY

Offset	Register Name
00h	SPI Communication Configuration Register
04h	SPI Slave Status Register
08h	SPI EC Status Register
0Ch	SPI Interrupt Enable Register
10h	EC Interrupt Enable Register
14h	Memory Configuration Register
18h	Memory Base Address0 Register
1Ch	Memory Write Limit0 Register
20h	Memory Read Limit0 Register
24h	Memory Base Address1 Register
28h	Memory Write Limit1 Register
2Ch	Memory Read Limit1 Register
30h	RX FIFO Host Bar Register
34h	RX FIFO Byte Counter Register
38h	TX FIFO Host Bar Register
3Ch	TX FIFO Byte Counter Register
40h	System Configuration Register
44h	SPI Master-to-EC Mailbox Register
48h	EC-to-SPI Master Mailbox Register

© 2020 Microchip Technology Inc.

Note 1: SPI Access is limited by the corresponding <Lock> bit and the <Mask> bit as follows:

- (NL-NM): “Not Locked and not Masked” has same access as EC.
- (L-NM): “Locked and not Masked” has RO access.
- (NL-M): “Not Locked and Masked” is reserved.
- (L-M): “Locked and Masked” is reserved.

2: SPI Access is limited by the only corresponding <Lock> bit as follows:

- (NL-NM): “Not Locked and not Masked” has same as EC access
- (L-NM): “Locked and not Masked” has RO access.
- (NL-M): “Not Locked and Masked” has same as EC access.
- (L-M): “Locked and Masked” has RO access.

3: SPI Access is limited by only the <Mask> bit as follows:

- (NL-NM): “Not Locked and not Masked” has RO* access.
- (L-NM): “Locked and not Masked” not applicable.
- (NL-M): “Not Locked and Masked” is reserved.
- (L-M): “Locked and Masked” is not applicable.

4: SPI Master has full access.

5: SPI Master does not have access.

37.9.1 SPI COMMUNICATION CONFIGURATION REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	Reserved	RES	-	-
23:16	Wait time: Set amount of wait time in cycles before transmitting data back to master. During this wait time status bits will be transmitted	R/W Note 2	4h	RESET_SYS
15:10	Reserved	RES	-	-
9:8	Tar Time: Turn Around Time select for Quad Wire • 0h = 1 cycle • 1h = 2 cycles • 2h = 4 cycles • 3h = 8 cycles • Other values are reserved.	R/W Note 2	0h	RESET_SYS
7:1	Reserved	RES	-	-
0	Single / Quad Wire Select • 0 = Single Wire • 1 = Quad Wire	R/W Note 2	0h	RESET_SYS

The lock register can be enabled to lock all or certain fields from SPI Master.

Note: If the programmed wait cycles aren’t enough, once the transaction passes a cycle after all data has been capture, the transaction will go through as long as the destination modules isn’t busy. An example of this could be seen when writing to the SPI_COMM_CONFIG register to make the cycles number larger, the write’s final status may return a BUSY STATUS but the write will still take effect; so on the next transaction the new wait cycle length will be used. Technically these can be posted Reads/Writes.

37.9.2 SPI SLAVE STATUS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow If SPI Master writes more than the space in the FIFO, the FIFO will flag an overflow error and data will not be stored.	R/WC Note 2	0h	RESET_SYS
27	RX FIFO Underflow If the SPI Slave reads RX FIFO when it is empty, RX FIFO Underflow flag will be set. This condition will never happen under normal situation.	R/WC Note 2	0h	RESET_SYS
26	TX FIFO Overflow If Master doesn't read all of the data it requested from the posted read block cycle, than data will still be left in the FIFO. This will cause misalignment with the following transactions and a new read cycle can cause overflow.	R/WC Note 2	0h	RESET_SYS
25	TX FIFO Underflow If Master reads more than what is in FIFO, FIFO will flag an underflow error and the data returned will just be the last valid pointer value.	R/WC Note 2	0h	RESET_SYS
24	RX FIFO Size Error If size requested is more than what Master provided and the Master terminates early→ error flag shut down request signal to ARM Bus. Size requested is less than what Master provided → ignored and continue transaction, may be taking in garbage.	R/WC Note 2	0h	RESET_SYS
23	DV_Busy If the Master requested a transaction whose destination is busy the request is ignored. Should use the poll or wait for interrupts.	R/WC Note 2	0h	RESET_SYS
22	Undefined Command Command isn't known.	R/WC Note 2	0h	RESET_SYS
21	ARM BUS Error Bus Error returned an error	R/WC Note 2	0h	RESET_SYS
20	Out of Limit 1 Error Address requested out of range or request when the BAR is disabled	R/WC Note 2	0h	RESET_SYS
19	Out of Limit 0 Error Address requested out of range or request when the BAR is disabled.	R/WC Note 2	0h	RESET_SYS
18	TX FIFO Reset Done Set after the SPI Master initiates a reset and the reset has been processed. FIFO is cleared.Note 2	R/WC Note 2	0h	RESET_SYS
17	RX FIFO Reset Done Set after the SPI Master initiates a reset and the reset has been processed. FIFO is cleared.Note 2	R/WC Note 2	0h	RESET_SYS
16	SPI Master Requested Reset Set when the SPI Master Requested a Configuration Reset.	R/WC Note 2	0h	RESET_SYS
15	OBF Flag Set when the EC writes to the Output Buffer signaling there is data for the Host to read.	R	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
14	IBF Flag Set when the Host writes to the Input Buffer signaling there is data for the EC to read.	R	0h	RESET_SYS
13	TM SPI Clock Count Error This bit is set when the SPI Clock Count Test Mode is set and there is an uneven amount of clocks.	R/WC Note 2	0h	RESET_SYS
12	Reserved	RES	-	-
11	TX FIFO Full The TX FIFO is full of data that was read from Memory.	R	0h	RESET_SYS
10	TX FIFO Empty Signifies SPI Master has read the data requested from Memory. Can be used to show there is data the SPI Master has requested and not been read yet. New read transactions will be aligned.	R	1h	RESET_SYS
9	RX FIFO Full The RX FIFO is full of data to be written to Memory.	R	0h	RESET_SYS
8	RX FIFO Empty Signifies all Memory write transactions for the SPI Masters requested size have been performed. New transactions are allowed.	R	1h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req If this bit is set, then something in the high 16-bit of status register is set and needs to be checked. SPI Master should take action to clear this.	R	0h	RESET_SYS
5	SREG Trans Busy When an SREG transaction is currently being processed.	R	0h	RESET_SYS
4	Memory Read Busy When an Memory Read transaction is currently being processed.	R	0h	RESET_SYS
3	Memory Write Busy When an Memory Write transaction is currently being processed.	R	0h	RESET_SYS
2	Reserved	RES	-	-
1	Memory Read Done When the ARM BUS side has fully finished writing the last written DWord to the FIFO for a set of data read from Memory for Posted Reads. - cleared with new Read request.	R/WC Note 2	0h	RESET_SYS
0	Memory Write Done When the ARM BUS side has fully finished the last transaction from the FIFO to write the data to Memory for Posted Writes .- clear with new Write request.	R/WC Note 2	0h	RESET_SYS

Note 1: This register is accessible by the SPI Master only in Advanced Mode.

2: Upon reset of the SPI Slave block, RX FIFO Reset Done and TX FIFO Reset Done bits will get asserted only after several SPI Clocks have been received as the FIFOs require SPI Clock to have a complete reset.

37.9.3 SPI EC STATUS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow: If SPI Master writes more than the space in the FIFO, the FIFO will flag an overflow error and data will not be stored.	R/WC	0h	RESET_SYS
27	RX FIFO Underflow: If the SPI Slave reads RX FIFO when it is empty, RX FIFO Underflow flag will be set. This condition will never happen under normal situation.	R/WC	0h	RESET_SYS
26	TX FIFO Overflow: If Master doesn't read all of the data it requested from the posted read block cycle, than data will still be left in the FIFO. This will cause misalignment with the following transactions and a new read cycle can cause overflow.	R/WC	0h	RESET_SYS
25	TX FIFO Underflow: If Master reads more than what is in FIFO, FIFO will flag an underflow error and the data returned will just be the last valid pointer value.	R/WC	0h	RESET_SYS
24	RX FIFO Size Error: If size requested is more than what Master provided and the Master terminates early→ error flag shut down request signal to ARM bus. Size requested is less than what Master provided → ignored and continue transaction, may be taking in garbage.	R/WC	0h	RESET_SYS
23	DV_Busy: If the Master requested a transaction whose destination is busy the request is ignored. Should use the poll or wait for interrupts.	R/WC	0h	RESET_SYS
22	Undefined Command: Command isn't known.	R/WC	0h	RESET_SYS
21	ARM BUS Error: ARM Bus returned an error	R/WC	0h	RESET_SYS
20	Out of Limit 1 Error: Address requested out of range or request when the BAR is disabled	R/WC	0h	RESET_SYS
19	Out of Limit 0 Error: Address requested out of range or request when the BAR is disabled.	R/WC	0h	RESET_SYS
18	TX FIFO Reset Done: Set after the SPI Master initiates a reset and the reset has been processed. FIFO is cleared.	R/WC	0h	RESET_SYS
17	RX FIFO Reset Done: Set after the SPI Master initiates a reset and the reset has been processed. FIFO is cleared.	R/WC	0h	RESET_SYS
16	SPI Master Requested Reset: Set when the SPI Master Requested a Configuration Reset.	R/WC	0h	RESET_SYS
15	OBF Flag: Set when the EC writes to the Output Buffer signaling there is data for the Host to read.	R	0h	RESET_SYS
14	IBF Flag: Set when the SPI Master writes to the Input Buffer signaling there is data for the EC to read.	R	0h	RESET_SYS
13	TM SPI Clock Count Error: This bit is set when the SPI Clock Count Test Mode is set and there is an uneven amount of clocks.	R/WC	0h	RESET_SYS
12	Reserved	RES	-	-
11	TX FIFO Full: The FIFO is full of data that was read from Memory.	R	0h	RESET_SYS
10	TX FIFO Empty: Signifies SPI Master has read the data requested from Memory. Can be used to show there is data the SPI Master has requested and not been read yet. New read transactions will be aligned.	R	1h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
9	RX FIFO Full: The FIFO is full of data to be written to Memory.	R	0h	RESET_SYS
8	RX FIFO Empty: Signifies all Memory write transactions for the SPI Masters requested size have been performed. New transactions are allowed.	R	1h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req: If this bit is set, then something in the high 16-bit of status register is set and needs to be checked. SPI Master should take action to clear this.	R	0h	RESET_SYS
5	SREG Trans Busy: When an SREG transaction is currently being processed.	R	0h	RESET_SYS
4	Memory Read Busy: When an Memory Read transaction is currently being processed.	R	0h	RESET_SYS
3	Memory Write Busy: When an Memory Write transaction is currently being processed.	R	0h	RESET_SYS
2	Reserved	RES	0h	RESET_SYS
1	Memory Read Done: When the ARM bus side has fully finished writing the last written DWord to the FIFO for a set of data read from Memory for Posted Reads. - cleared with new Read request.	R/WC	0h	RESET_SYS
0	Memory Write Done: When the ARM bus side has fully finished the last transaction from the FIFO to write the data to Memory for Posted Writes .- clear with new Write request.	R/WC	0h	RESET_SYS

37.9.4 SPI INTERRUPT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
27	RX FIFO Underflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
26	TX FIFO Overflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
25	TX FIFO Underflow: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
24	RX FIFO Size Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
23	DV_Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
22	Undefined Command: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
21	ARM BUS Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
20	Out of Limit 1 Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
19	Out of Limit 0 Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
18	TX FIFO Reset Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
17	RX FIFO Reset Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
16	SPI Master Requested Reset: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
15	OBF Flag: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
14	Reserved	RES	-	-
13	TM SPI Clock Count Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
12	Reserved	RES Note 2	-	-
11	TX FIFO Full: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
10	TX FIFO Empty: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
9	RX FIFO Full: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
8	RX FIFO Empty: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
5	SREG Trans Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
4	Memory Read Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
3	Memory Write Busy: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
2	Reserved	RES Note 2	0h	RESET_SYS
1	Memory Read Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
0	Memory Write Done: Set SPI interrupt to SPI Master when corresponding Status Bit is set.	R/W Note 2	0h	RESET_SYS
Note:	This register may be accessible by the SPI Master only in Advanced Mode.			

© 2020 Microchip Technology Inc.

37.9.5 EC INTERRUPT ENABLE REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28	RX FIFO Overflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
27	RX FIFO Underflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
26	TX FIFO Overflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
25	TX FIFO Underflow: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
24	RX FIFO Size Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
23	DV_Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
22	Undefined Command: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
21	ARM BUS Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
20	Out of Limit 1 Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
19	Out of Limit 0 Error: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
18	TX FIFO Reset Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
17	RX FIFO Reset Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
16	SPI Master Requested Reset: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
15	OBF Flag: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
14	IBF Flag: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
13	TM SPI Clock Count Error: Set SPI interrupt to SPI Master when corresponding Status Bit is set	R/WC Note 3	0h	RESET_SYS
12	Reserved	RES	-	-
11	TX FIFO Full: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
10	TX FIFO Empty: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
9	RX FIFO Full: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
8	RX FIFO Empty: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
7	Reserved	RES	-	-
6	Poll High Req: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
5	SREG Trans Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
4	Memory Read Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
3	Memory Write Busy: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
2	Reserved	RES	0h	RESET_SYS
1	Memory Read Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS
0	Memory Write Done: Set interrupt to EC when corresponding Status Bit is set.	R/W Note 3	0h	RESET_SYS

37.9.6 MEMORY CONFIGURATION REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	BAR 1 Enable: Enables Region 1	R/W Note 1	0h	RESET_SYS
0	BAR 0 Enable: Enables Region 0	R/W Note 1	0h	RESET_SYS

37.9.7 MEMORY BASE ADDRESS0 REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Base Address for Region 0 **Note:** Base Address for Region 0 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

37.9.8 MEMORY WRITE LIMIT0 REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Write Limit for Region 0 **Note:** Write Limit for Region 0 is DWORD aligned.	R/W Note 1	0	RESET_SYS
1:0	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

37.9.9 MEMORY READ LIMIT0 REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Read Limit for Region 0 **Note:** Read Limit for Region 0 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

37.9.10 MEMORY BASE ADDRESS1 REGISTER

Offset	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Base Address for Region 1 **Note:** Base Address for Region 1 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

37.9.11 MEMORY WRITE LIMIT1 REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Write Limit for Region 1 **Note:** Write Limit for Region 1 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

37.9.12 MEMORY READ LIMIT1 REGISTER

Offset	2Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:15	Reserved	RES	-	-
14:2	Read Limit for Region 1 **Note:** Read Limit for Region 1 is DWORD aligned.	R/W Note 1	0h	RESET_SYS
1:0	Reserved	RES	-	-

37.9.13 RX FIFO HOST BAR REGISTER

Offset	30h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	RX FIFO Bar Latest offset address requested by the SPI Master for a write transfer. This register gets set for a new transaction request.	R	0h	RESET_SYS

37.9.14 RX FIFO BYTE COUNTER REGISTER

Offset	34h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
14:0	RX FIFO Byte Count Number of Bytes written through the AHB transfer. This register gets cleared for every new request	R	0h	RESET_SYS

37.9.15 TX FIFO HOST BAR REGISTER

Offset	38h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
15:0	TX FIFO Bar: Latest offset address requested by the SPI Master for a read transfer. This register gets set for a new transaction request.	R	0h	RESET_SYS

37.9.16 TX FIFO BYTE COUNTER REGISTER

Offset	3Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	Reserved	RES	-	-
14:0	RX FIFO Byte Count: Number of Bytes written through the AHB transfer. This register gets cleared for every new request	R	0h	RESET_SYS

37.9.17 SYSTEM CONFIGURATION REGISTER

Note: EC access to registers remains the same in all configurations. However the SPI Master access can be changed depending on the configurations set in this register.

Note: The System Configuration Register is a read only register from SPI Master. It can be read and written by the EC at all times. The Lock bits in this register only define the type of access for the SPI Master.

Note: Any read of the SPI Slave registers when Mask EC Registers bit is set, will always return 0h value to the SPI Master.

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:20	Reserved	RES	-	-
19	EC Data Available: Notification to TX FIFO Engine that data is available for AHB Transfer. This register but is cleared by Hardware at the end of the transaction, with SLV_SPI_CS# de-assertion.	R/WC Note 3	0h	RESET_SYS
18	Simple Mode: Enable SPI Slave Simple Mode operation	R/W Note 3	0h	RESET_SYS
17	Mask EC Registers Mask EC registers Mask EC Registers from SPI Master. All the register are neither readable now writable from SPI Master.	R/W Note 3	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
16	Activate: SPI Slave Block Enabled / Disabled	R/W Note 3	0h	RESET_SYS
15:11	Reserved	RES	-	-
10	Lock Test Modes: Lock test modes write access from SPI Master	R/W Note 3	1h	RESET_SYS
9	Reserved	RES	-	-
8	Reserved	RES	-	-
7	Lock Mem Bar1: Lock writes to Region 1 Addresses from SPI Master.	R/W Note 3	1h	RESET_SYS
6	Lock Mem Bar0: Lock writes to Region 0 Addresses from SPI Master	R/W Note 3	1h	RESET_SYS
5	Lock SPI Int En: Lock SPI interrupt enable register	R/W Note 3	0h	RESET_SYS
4	Lock SPI Stats: Lock write access to SPI Status field from SPI Master.	R/W Note 3	0h	RESET_SYS
3	Lock Wait Cycles: Lock Wait Cycle register write access from SPI Master.	R/W Note 3	0h	RESET_SYS
2	Lock Tar Time: Lock Tar Time register write access from SPI Master.	R/W Note 3	0h	RESET_SYS
1	Lock Quad / Single Write Mode: Lock Write Mode register write access from SPI Master.	R/W Note 3	0h	RESET_SYS
0	Soft reset Soft reset for entire SPI Slave Block. This bit is self clearing	WO Note 5	0h	RESET_SYS

37.9.18 SPI MASTER-TO-EC MAILBOX REGISTER

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	SPI Master to EC: Write only register for the Host. When data is written to this register the IBF Flag is set. EC can read the data and writes of 0xFFFF_FFFF will clear this register. Any form of read will clear the flag for this register.	R/WC Note 4	0h	RESET_SYS

37.9.19 EC-TO-SPI MASTER MAILBOX REGISTER

Offset	48h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	EC to SPI Master: Read only register for the Host. When data is written to this register the OBF Flag is set. Host can read the data and writes of 0xFFFF_FFFF will clear this register, also clearing the flag. Any form of read will clear the flag for this register	R/WC Note 4	0h	RESET_SYS

Note: Because any form of read (8/16/32 bit read) will clear the flag – it is necessary for the EC and the external FW of which the SPI Master resides agrees upon a protocol.

© 2020 Microchip Technology Inc.

37.10 Commands Supported

The list of commands supported by SPI slave is given below in

TABLE 37-9: SPI COMMANDS

Command Name	Code	Description
CMD_IN_BAND_RST	FFh	In Band Reset.
CMD_UNDEF_DWORD_W	01h	Undefined Size DWord Write
CMD_UNDEF_BYTE_W	02h	Undefined Size Byte Write
CMD_UNDEF_DWORD_R	05h	Undefined Size DWord Read
CMD_UNDEF_BYTE_R	06h	Undefined Size Byte Read
CMD_RST_RX_FIFO	12h	Reset RX FIFO pointers
CMD_RST_TX_FIFO	14h	Reset TX FIFO pointers
CMD_RST_RXTX_FIFO	16h	Reset RX and TX FIFO pointers
CMD_EXT_REG_W8	41h	External Register Bank 8 bit write
CMD_EXT_REG_R8	45h	External Register Bank 8 bit read
CMD_SREG_W8	09h	SPI Slave Register 8 bit Write
CMD_SREG_W16	0Ah	SPI Slave Register 16 bit Write
CMD_SREG_W32	0Bh	SPI Slave Register 32 bit Write
CMD_SREG_R8	0Dh	SPI Slave Register 8 bit Read
CMD_SREG_R16	0Eh	SPI Slave Register 16 bit Read
CMD_SREG_R32	0Fh	SPI Slave Register 32 bit Read
CMD_MEM_W8	21h	Standalone 8 bit Memory Write
CMD_MEM_W16	22h	Standalone 16 bit Memory Write
CMD_MEM_W32	23h	Standalone 32 bit Memory Write
CMD_MEM_R8	25h	Standalone 8 bit Memory Read
CMD_MEM_R16	26h	Standalone 16 bit Memory Read
CMD_MEM_R32	27h	Standalone 32 bit Memory Read
CMD_RD_SNGL_FIFO8	28h	Standalone 8 bit Memory Read FIFO
CMD_RD_SNGL_FIFO16	29h	Standalone 16 bit Memory Read FIFO
CMD_RD_SNGL_FIFO32	2Bh	Standalone 32 bit Memory Read FIFO
CMD_RD_SNGL_FIFO8_FSR	68h	8 bit Memory Read FIFO with Status
CMD_RD_SNGL_FIFO16_FSR	69h	16 bit Memory Read FIFO with Status
CMD_RD_SNGL_FIFO32_FSR	6Bh	32 bit Memory Read FIFO with Status
CMD_POLL_LOW	2Ch	Read lower 16 bits of the Status Register
CMD_POLL_HIGH	2Dh	Read higher 16 bits of the Status Register
CMD_POLL_ALL	2Fh	Read all 32 bits of the Status Register
CMD_EXTEND	6Ch	Declare Second Command Byte
CMD_MEM_BLK_W	80h-87h	Block 1-8 D word Memory Write
CMD_MEM_BLK_R	A0h-A7h	Block 1-8 D word Memory Read
CMD_RD_BLK_FIFO	C0h-C7h	Block 1-8 D word Read FIFO
CMD_BLK_RD_FIFO_FSR	E0h-E7h	Block 1-8 D word Read FIFO with status

© 2020 Microchip Technology Inc.

Note:	For CMD_MEM_BLK_W, CMD_MEM_BLK_R, CMD_RD_BLK_FIFO, CMD_BLK_RD_FIFO_FSR commands last nibble 0-7 represent the 1 to 8 D Word operation. where 0 represents 1 D word and 7 represents 8 D word operation. This is equivalent to saying that there are 8 commands for each of these commands.
Note:	For CMD_UNDEF_DWORD_W, CMD_UNDEF_BYTE_W, CMD_UNDEF_DWORD_R and CMD_UNDEF_BYTE_R commands, the SPI Master first needs to issue a CMD_EXTEND command.
Note:	Commands CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32 and CMD_MEM_BLK_W are be used to generate AHB write transaction in the EC, pointed by Memory Base Address0 Register and Memory Base Address1 Register and the offset address received from the master.
Note:	Commands CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32 and CMD_MEM_BLK_R are be used to generate AHB read transaction in the EC, pointed by Memory Base Address0 Register and Memory Base Address1 Register and the offset address received from the master.
Note:	For CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32, CMD_MEM_BLK_W, CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32 and CMD_MEM_BLK_R the address bit 15 received from the SPI Master determines to which memory base address region the transaction is targeted. If bit 15 of the address is 0b, the transaction if for Memory Base Address0 Register and if bit 15 of the address is 1b, the transaction is for Memory Base Address1 Register

37.10.1 COMMAND FORMAT

This section lists the command format for each command category

37.10.1.1 Non-Posted Memory (Block) and SREG Write Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-2 for non-posted writes. CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32, CMD_MEM_BLK_W, CMD_SREG_W8, CMD_SREG_W16 and CMD_SREG_W32 commands can be used for this type of transfer.

FIGURE 37-2: NON-POSTED MEMORY (BLOCK) AND SREG WRITE COMMAND FORMAT



For these commands, the SPI Slave sends the write command status at the end of the transaction. These commands are ideal for cases where the SPI Master wants to see the status without issuing another command to confirm the transfer status.

37.10.1.2 Non-Posted Memory (Block) and SREG Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-3 for non-posted reads. CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32, CMD_MEM_BLK_R, CMD_SREG_R8, CMD_SREG_R16 and CMD_SREG_R32 commands can be used for this type of transfer.

FIGURE 37-3: NON-POSTED MEMORY (BLOCK) AND SREG READ COMMAND FORMAT



For these commands, the SPI Slave sends the status and the read data. These commands are ideal for cases where the SPI Master wants to see the status without issuing another command to confirm the transfer status.

37.10.1.3 Posted Memory Write Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-4 for posted memory writes. CMD_MEM_W8, CMD_MEM_W16, CMD_MEM_W32 and CMD_MEM_BLK_W commands can be used for this type of transfer. The SPI Master needs to end the command by de-asserting the SLV_SPI_CS# after the command, address and data is transfered. The early termination of the command indicates that the current transfer is Posted transfer to the SPI slave. The SPI Master can do other work and later check the status of the transfer via the CMD_POLL_LOW, CMD_POLL_HIGH and CMD_POLL_ALL command.

FIGURE 37-4: POSTED MEMORY WRITE COMMAND FORMAT



37.10.1.4 Posted Memory Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-5 for posted memory read.CMD_MEM_R8, CMD_MEM_R16, CMD_MEM_R32 and CMD_MEM_BLK_R commands can be used for this type of transfer. The SPI Master needs to end the command by de-asserting the SLV_SPI_CS# after the command and address is transfered. The early termination of the command indicates that the current transfer is Posted transfer to the SPI slave. The SPI Master can do other work and later check the status of the transfer via the CMD_POLL_LOW, CMD_POLL_HIGH and CMD_POLL_ALL command. Once the status indicates that data is available, SPI Master initiates a FIFO read command to get the data. See Section 37.10.1.7, FIFO Read Command Format for details.

FIGURE 37-5: POSTED MEMORY READ COMMAND FORMAT



37.10.1.5 Undefined Size Memory Write Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-6 for Undefined size memory write transfer. CMD_EXTEND, CMD_UNDEF_DWORD_W and CMD_UNDEF_BYTE_W commands can be used for this type of transfer. The SPI Master needs to end the Undefined Size Memory Write transfer by de-asserting the SLV_SPI_CS#. There will be no status for this transfer provided to the SPI Master, however the SPI Master may use CMD_POLL_LOW, CMD_POLL_HIGH or CMD_POLL_ALL command to know the status of the transfer.

FIGURE 37-6: UNDEFINED SIZE MEMORY WRITE COMMAND FORMAT



37.10.1.6 Undefined Size Memory Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-7 for Undefined size memory read transfer. CMD_EXTEND, CMD_UNDEF_DWORD_R and CMD_UNDEF_BYTE_R commands can be used for this type of transfer. The SPI Master needs to end the Undefined Size Memory Read transfer by de-asserting the SLV_SPI_CS#. The status returned by the Serial Peripheral Interface (SPI) Slave informs whether there is data in the FIFO available for the SPI Master to read.

FIGURE 37-7: UNDEFINED SIZE MEMORY READ COMMAND FORMAT



© 2020 Microchip Technology Inc.

37.10.1.7 FIFO Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-8 for FIFO Read transfer. CMD_RD_SNGL_FIFO8, CMD_RD_SNGL_FIFO16, CMD_RD_SNGL_FIFO32 and CMD_RD_BLK_FIFO commands can be used for this type of transfer. This command is used along with Posted Memory read command. See section Section 37.10.1.4, Posted Memory Read Command Format for more information.

FIGURE 37-8: FIFO READ COMMAND FORMAT



37.10.1.8 FIFO Read with Status Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-9 for FIFO Read with Status transfer. CMD_RD_SNGL_FIFO8_FSR, CMD_RD_SNGL_FIFO16_FSR, CMD_RD_SNGL_FIFO32_FSR and CMD_BLK_RD_FIFO_FSR commands can be used for this type of transfer. This command is used along with the posted read command and helps avoid constant polling the SPI Slave via CMD_POLL_LOW, CMD_POLL_HIGH or CMD_POLL_ALL commands. The SPI Master may issue the FIFO Read with Status Command read the data if data is available (as indicated in the Status) or terminate the transaction after status read, if data is not available.

FIGURE 37-9: FIFO READ WITH STATUS COMMAND FORMAT



37.10.1.9 Poll Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-10 for Poll transfer. CMD_POLL_LOW, CMD_POLL_HIGH or CMD_POLL_ALL commands can be used for this type of transfer. This command will immediately return the contents of SPI Slave Status Register. This command should be used by the SPI Master before issuing the command of the same type, check for errors that may have occurred with any previous transaction or check for data availability in case of posted transactions.

FIGURE 37-10: POLL COMMAND FORMAT



37.10.1.10 External Register Write Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-11 for External Register write transfer. CMD_EXT_REG_W8 command can be used for this type of transfer. This command is used to write to the SPI Slave register bank. The status of this command will be immediately returned back to the SPI Master. This type of access is used to write to registers in the SPI clock domain.

FIGURE 37-11: EXTERNAL REGISTER WRITE COMMAND FORMAT



37.10.1.11 External Register Read Command Format

The SPI Master should expect the command and data in below order as shown in Figure 37-12 for External Register read transfer. CMD_EXT_REG_R8 command can be used for this type of transfer. This command is used to read the register bank. The status of this command will be immediately returned back to the SPI Master. This type of access is used to read registers in the SPI clock domain.

FIGURE 37-12: EXTERNAL REGISTER READ COMMAND FORMAT



37.10.1.12 Simple Mode

The SPI Master should expect the command and data in below order as shown in Figure 37-13 for External Register read transfer.

FIGURE 37-13: SIMPLE MODE



37.10.1.13 Reset Commands Format

The SPI Master should expect the Reset command behavior as shown in Figure 37-14. CMD_RST_RX_FIFO, CMD_RST_TX_FIFO and CMD_RST_RXTX_FIFO commands show the behavior shown in Figure 37-14. There should be at least 1 dummy cycle of SLV_SPI_SCLK clock for the reset operation to complete.

FIGURE 37-14: RESET COMMAND FORMAT



37.11 Examples

This section shows an example usage of this block. The example algorithm for initialization from EC (EC Initialization) and SPI Master (SREG Accesses by SPI Master) are discussed in this section

37.11.1 EC INITIALIZATION

1. SPI Slave configured as a Bridge
   a) Configure EC Interrupt Enable Register and SPI Interrupt Enable Register
   b) Enable SPI Slave Block
      i. configure System Configuration Register with 32'h0001_0000
         1. Activate Slave (bit 16) and unlocks write access to registers
2. SPI Slave configured in non bridge mode:
   a) Configure Memory Configuration Register, Memory Base Address0 Register and Memory Base Address1 Register
   b) Configure SPI Communication Configuration Register
   c) Configure EC Interrupt Enable Register and SPI Interrupt Enable Register
   d) Enable SPI Slave Block
      i. Write System Configuration Register with 32'h0001_04CE
         1. Activate Slave (bit 16) and lock all write access
   e) EC Configures GPIO's

© 2020 Microchip Technology Inc.

3. SPI Slave configured in non bridge mode (solely prohibiting Memory Configuration write accesses from SPI Master):
   a) Configure Memory Configuration Register, Memory Write Limit0 Register, Memory Read Limit0 Register, Memory Base Address0 Register, Memory Write Limit1 Register, Memory Read Limit1 Register and Memory Base Address1 Register.
   b) Configure SPI Communication Configuration Register
   c) Enable SPI Slave Block
      i. Write System Configuration Register with 32'h0001_00C0
         1. Activate Slave (bit 16) and lock all write access
   d) EC Configures GPIO's

Note: In 3, the SPI Master can still access/write SPI Communication Configuration Register and SPI Interrupt Enable Register.

4. If SPI Master does not have any access to EC registers (this completely hides the Memory Configuration Register (other registers as well) ie. the SPI Master will not be able to read/write the base address or limits):
   a) Write Memory Configuration Register
   b) Configure EC Interrupt Enable Register and SPI Interrupt Enable Register
   c) Enable SPI Slave Block
      i. Write System Configuration Register with 32'h0003_0xxx
         1. Activate Slave (bit 16) and lock all write access

Note: EC should make any configurations to Memory Register while block is not activated and as long as Mask EC Registers is set to confirm the SPI Master does not have access to these registers

Note: In 4, the SPI Master still has access to write its own SPI Communication Configuration Register. See Note 2 for details about register access when Mask bit is set.

37.11.2 SREG ACCESSES BY SPI MASTER

Note: After any reset to the Serial Peripheral Interface (SPI) Slave block, we need to clear RX FIFO RESET DONE and TX FIFO RESET DONE SPI EC Status Register and SPI Slave Status Register.

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 or CMD_SREG_W16 or CMD_SREG_W32 command with address 0x06 (access to SPI EC Status Register) and data 0x06.
2. Insert 4 Wait Cycles
3. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty

Note: When SPI Master is allowed to change communication Interconnect Configuration the following two are doable, otherwise EC will have to set configuration through SPB accesses

Initialization of SPL Slave from SPI Master: Changing from 1 Turn around cycle (Current Default) to 4 Turn around cycle & 4 (Current Default) Wait Bytes to 12 Wait Bytes

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 or CMD_SREG_W16 or CMD_SREG_W32 command with address 0x00 (access to SPI Communication Configuration Register) and data 0x000C_0200 (Still in Single Wire).
2. Insert 4 Wait Cycles
3. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty. One may get SREG BUSY if current Wait Cycles isn't large enough
4. Next transaction with be 4 Cycles TAR/Dummy and 12 Bytes of Wait Cycles

Initialization of SPL Slave from SPI Master: Change from Single Wire Transfer to Quad Wire

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 command with address 0x00 (access to SPI Communication Configuration Register) and data 0x01.
2. Insert 4 TAR/Dummy and 12 wait cycles and then read the Status back from SPI slave.
3. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty.
4. Next transaction will use quad wire.

37.11.3 MEMORY WRITE BY SPI MASTER

1. Initiate a transaction from SPI Master with either CMD_MEM_W8 or CMD_MEM_W16 or CMD_MEM_W32 or CMD_MEM_BLK_W command with address 0x0100 and the required data bytes.
2. Insert 4 TAR/Dummy and 12 wait cycles.
3. Read the 2 byte Status back and check that all bits should be cleared except TX FIFO empty, Memory Write done and Memory Write Busy (RX FIFO not empty).

If Memory Write Busy bit is set: Poll until Done then clear bit in Status Register

1. Initiate a transaction from SPI Master with either CMD_POLL_LOW or CMD_POLL_ALL command.
2. Insert 4 TAR/Dummy
3. Read the Status back and check if Memory Write Busy is cleared.
4. If Memory Write Busy is cleared move to the next set of command, else repeat this loop.
5. Write to SPI Slave Status Register to clear the Memory Write Done bit.

If Memory Write Done bit set: Using CMD_SREG_W8 command write to SPI Slave Status Register to the clear Memory Write Done

1. Initiate a transaction from SPI Master with either CMD_SREG_W8 or CMD_SREG_W16 or CMD_SREG_R32 command with address 0x04 and data 0x01.
2. Insert 4 TAR/Dummy and 12 wait cycles.
3. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty.

37.11.4 MEMORY READ BY SPI MASTER

1. Initiate a transaction from SPI Master with either CMD_MEM_R8 or CMD_MEM_R16 or CMD_MEM_R32 or CMD_MEM_BLK_R command with address 0x0100.
2. Insert 4 TAR/Dummy and 12 wait cycles.
3. Read the 2 byte Status back and check that all bits should be cleared except RX FIFO empty, Memory Read done / Memory Read Busy.

Using CMD_RD_SNGL_FIFO8_FSR or CMD_RD_SNGL_FIFO16_FSR or CMD_RD_SNGL_FIFO32_FSR command poll for Memory Read Done bit in the SPI Slave Status Register and then read the data and clear Memory Read Done bit.

1. Initiate a transaction from SPI Master with either CMD_RD_SNGL_FIFO8_FSR or CMD_RD_SNGL_FIFO16_FSR or CMD_RD_SNGL_FIFO32_FSR command.
2. Insert 4 TAR/Dummy.
3. Read status back from SPI slave as long as Memory Read Done bit is not set and after the Memory Read Done bit is set the Data from the SPI slave is valid.
4. Read the Status back and check that all bits should be cleared except RX/TX FIFO empty.
5. Using CMD_SREG_W8 command write to SPI Slave Status Register to the clear Memory Read Done bit

© 2020 Microchip Technology Inc.

37.11.5 SIMPLE MODE

1. Configuration
   - EC application code needs to program Memory Base Address0 Register and Memory Write Limit0 Register for setting up the write region for SPI Master.
   - EC application code needs to program Memory Base Address1 Register and Memory Read Limit1 Register for setting up the read region for SPI Master.
   - EC application code needs to program EC Interrupt Enable Register.
   - EC application code needs to program SPI Interrupt Enable Register if interrupt is enabled to the SPI Master.
   - EC application code needs to program 0x0005_0000 in System Configuration Register to enable Simple Mode and activate the block
2. Write from SPI Master
   - Wait for SPI Master to write start the transaction
   - EC can read the RX FIFO Byte Counter Register to know how may bytes have been written to the Memory
   - EC may read the data from Memory Base Address0 Register when it is ready.
3. Read from SPI Master
   - The EC application code sets up response at Memory Base Address1 Register
   - The EC application code then writes to System Configuration Register bit 19 EC Data Available: Notification to TX FIFO Engine that data is available for AHB Transfer. This register but is cleared by Hardware at the end of the transaction, with SLV_SPI_CS# de-assertion. to indicate to the SPI slave that the data is available for sending to SPI Master.
4. Transaction completion
   - The SPI Master terminates the transfer after it has received all the bits.
   - EC Data Available: Notification to TX FIFO Engine that data is available for AHB Transfer. This register but is cleared by Hardware at the end of the transaction, with SLV_SPI_CS# de-assertion. is cleared.
   - TX FIFO is cleared
   - Wait for 2.56micro second for RX FIFO to be empty.
5. Start new transfer by jumping to step 2 and repeating the steps

Note:	The content of the Simple Mode data transfer need to be interpreted by the application code.

38.0 EEPROM

38.1 Overview

MEC1523 includes a 2K x 8bit EEPROM (Electrically Erasable Programmable Read Only Memory).

38.2 References

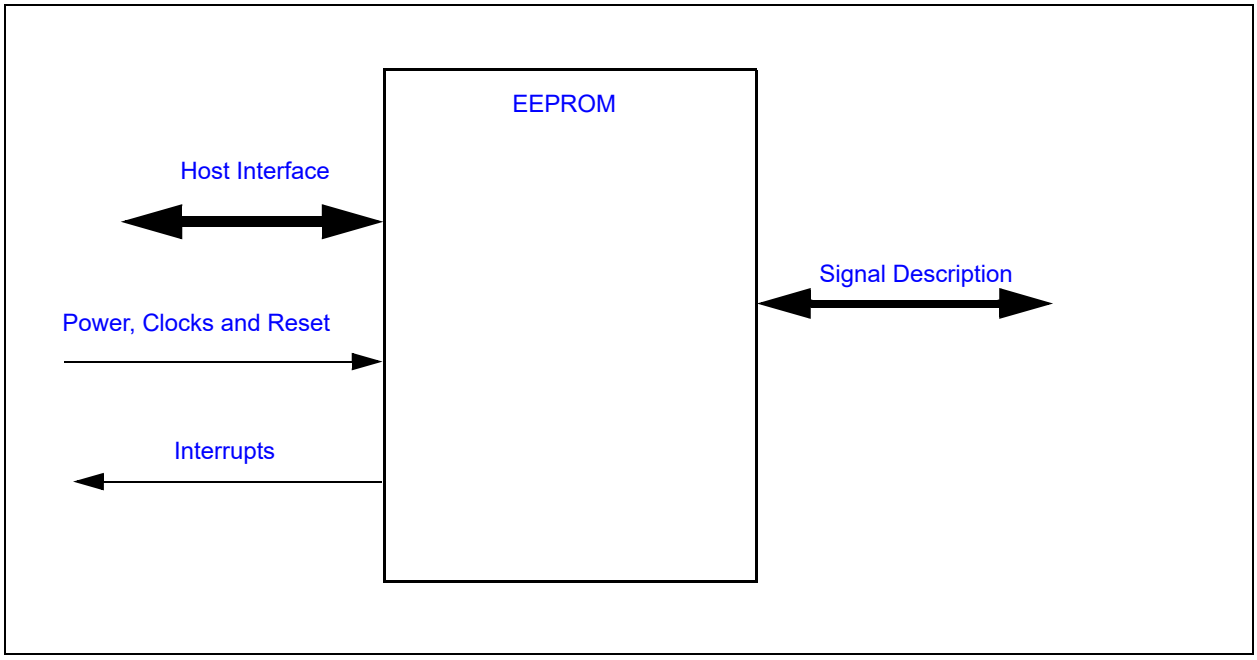
No references have been cited for this feature.

38.3 Terminology

There is no terminology defined for this section.

38.4 Interface

FIGURE 38-1: I/O DIAGRAM OF BLOCK



38.5 Signal Description

There are no external signals for this block.

38.6 Host Interface

The EEPROM interface is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

38.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

38.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

© 2020 Microchip Technology Inc.

38.7.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for EEPROM logic.

38.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.
RESET_EEPROM	This signal resets most of the registers and logic in this block to their default state. It is asserted either on a RESET_SYS is asserted, or when the SOFT_RESET bit in the EEPROM Mode Register is written with a '1b'.

38.8 Interrupts

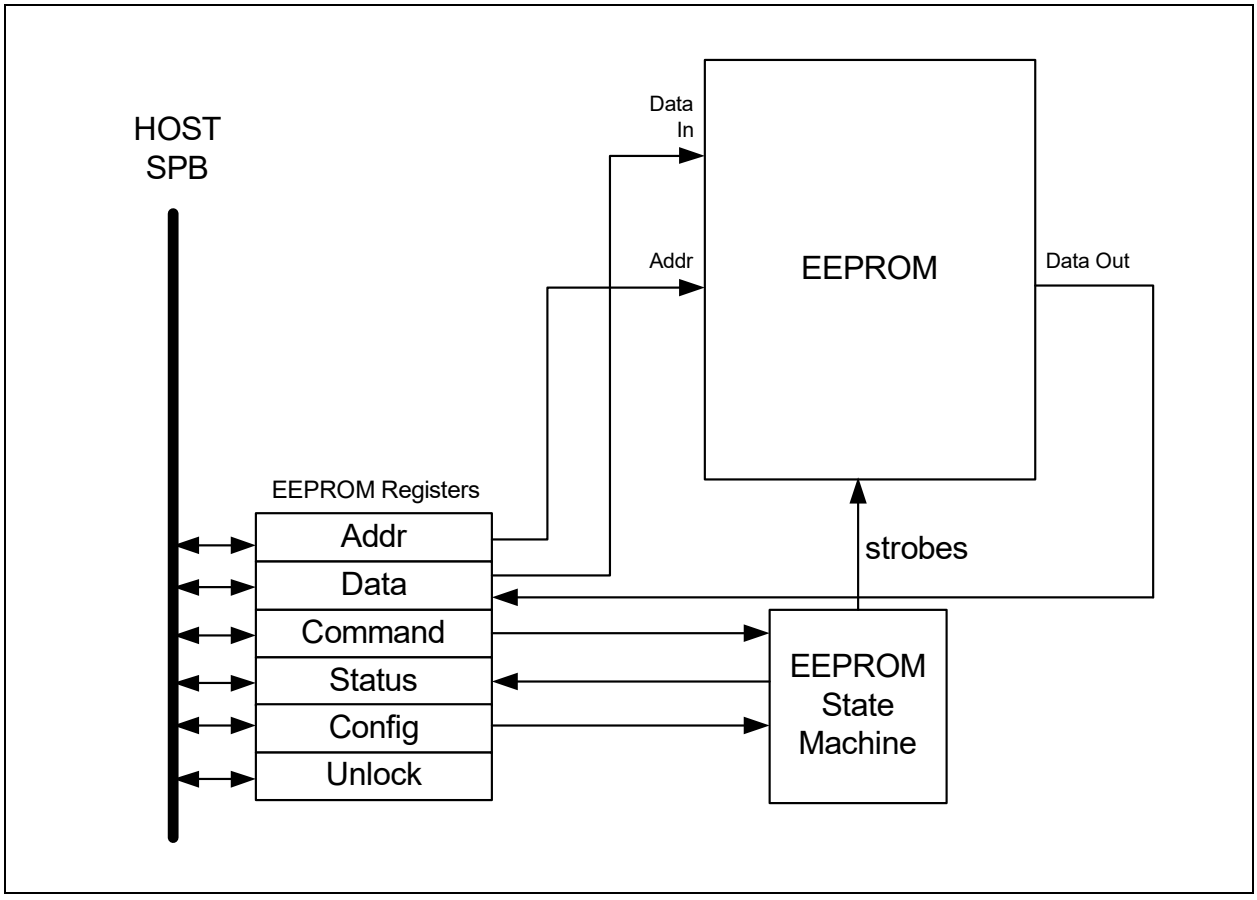
Source	Description
EEPROM	EEPROM transfer completed.

38.9 Low Power Modes

The EEPROM Controller enters its lowest power state whenever it is not busy (the TRANSFER_ACTIVE bit in the EEPROM Status Register is 0).

38.10 Description

TABLE 38-1: EEPROM BLOCK DIAGRAM



38.11 EEPROM Operation

The EEPROM consists of a 2K array of bytes, organized into 32-byte pages. Data are transfered to and from the EEPROM array to a register buffer in transfers of 1 to 32 bytes at a time. Reads can start at any byte address in the array and will transfer the quantity from the EEPROM array to the buffer. Writes can start at any byte address but cannot cross 32 byte page boundaries.

A read transfer of 1 byte from the EEPROM fabric to the buffer takes approximately 7µS. A transfer of 32 bytes takes an additional 50µS.

A write transfer of 1 byte to 32 bytes to the EEPROM fabric from the buffer takes approximately 5ms.

38.11.1 ENDURANCE

Each 32-byte page can be written up to 1,000,000 times before the page's ability to retain data is compromised. Data retention is >100years.

38.11.2 EEPROM INITIALIZATION

Data in offsets 7FCh to 7FFh from the EEPROM Memory Block are examined a boot. If bit 31 is a '1,' the 31 least significant bits of the data are written to the EEPROM Unlock Register and the LOCK bit in the EEPROM Lock Register is set to '1b' and the EEPROM Memory Block becomes inaccessible.

© 2020 Microchip Technology Inc.

38.11.3 PASSWORD USE

Offset 7FCh is the last 32-bit word in the EEPROM memory block. If bit 31 is a '0' in that word, then the location can be treated as EEPROM data that stores only positive integers in the range 0 to 2 billion (or so). If bit 31 is a '1' then the word contains a password that can be used to unlock the EEPROM memory block.

To keep the EEPROM memory block secure, and to insure that the EC firmware is authorized to read its contents, EC firmware can use this password. The following code can be executed as part of the EC's initialization code:

```
int password;
#define SECRET_PASSWORD 0xXXXXXXXX
#define Unlock_register 0x40002C14 // address of key register

password = eeprom_read(0x7FC);
if( password == 0xFFFFFFFF )
{
        // EEPROM is unlocked and the key not yet installed
        // establish key and force a reboot
        eeprom_write(0x7FF, SECRET_PASSWORD);
        force_reset();
} else {
        // unlock the EEPROM
        *Unlock_register = SECRET_PASSWORD;
}
```

The subroutines force_reset() forces a system reset, using, for example, the watchdog timer. If the EC firmware knows the right password, it can unblock the EEPROM memory block; if it does not, then the block will remain inaccessible. The code can run in the Boot Block so that the SECRET_PASSWORD (a constant in the code) cannot be read via JTAG or over the Host bus, and thus can only be known to a valid EC firmware block.

38.11.4 STATUS BYTE

The READ STATUS and WRITE STATUS commands affect the Status Byte. The Status Byte can be used to control write-protection in the EEPROM fabric, as well as providing a mechanism for reporting on status of the EEPROM fabric.The Status Byte is non-volatile and retains its value even after a RESET_SYS system reset.

The byte is defined as follows:

Bits	Description	Type
7:4	Reserved	RES
3:2	WRITE_PROTECT Write protection of the EEPROM fabric. 3=The entire EEPROM fabric is write protected (read only) 2=The upper half (byte offsets 0400h to 07FFh) of the EEPROM fabric is write protected 1=The upper quarter (byte offsets 0600h to 07FFh) of the EEPROM fabric is write protected 0=The entire EEPROM fabric is writable	R/W
1	WRITE_ENABLE In normal use, this bit is always 0. Write enable of the EEPROM fabric is handled automatically.	R
0	WRITE_IN_PROGRESS 1=The EEPROM fabric is processing a write 0=The EEPROM fabric is idle	R

38.12 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the EEPROM Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 38-2: EC-ONLY REGISTER SUMMARY

EC Offset	Register Name
00h	EEPROM Mode Register
04h	EEPROM Execute Register
08h	EEPROM Status Register
0Ch	EEPROM Interrupt Enable Register
10h	EEPROM Password Register
14h	EEPROM Unlock Register
18h	EEPROM Lock Register
20h	EEPROM Buffer Register

38.12.1 EEPROM MODE REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	SOFT_RESET This is self clearing bit. Writing a one to this bit resets the entire EEPROM interface.	W	-	RESET_EEPROM
0	ACTIVATE This 8-bit register holds the data to be written into the EEPROM memory array during a program cycle, as well as the data returned from an EEPROM read during a read cycle. It should be set up before the EEPROM Execute Register is configured. 1=The EEPROM controller is enabled 0=The EEPROM controller is disabled and placed in its lowest power state	R/W	0h	RESET_EEPROM

38.12.2 EEPROM EXECUTE REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:29	Reserved	RES	-	-
28:24	TRANSFER_SIZE The number of bytes to be transfered between the EEPROM fabric and the buffer. A count of 0 is means a transfer of 32 bytes. This field is only applicable for WRITE and READ commands.	R/W	0h	RESET_EEPROM
23:19	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
18:16	COMMAND A write to this register automatically starts an EEPROM transfer between the fabric and the buffer 3=WRITE STATUS 2=READ STATUS 1=WRITE 0=READ	R/W	0h	RESET_EEPROM
15:0	EEPROM_ADRESS This register represents a byte address in the EEPROM. Bits[15:11] should be 0, but there is error flagged if they are not. This field only applies to READ and WRITE commands. It does not apply to READ STATUS and WRITE STATUS commands.	R/W	0h	RESET_EEPROM

38.12.3 EEPROM STATUS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:9	Reserved	RES	-	-
8	TRANSFER_ACTIVE A transfer between the EEPROM fabric and the EEPROM Buffer Register is in progress	R	0h	RESET_EEPROM
6:3	Reserved	RES	-	-
1	EXECUTION_ERROR This bit is set to '1b' if an illegal command has been programmed into the block. A command is illegal if: • The EEPROM Execute Register is written while a previous command has not yet completed (that is, while TRANSFER_ACTIVE is '1b') • An illegal value was written into the COMMAND field	R/WC	0h	RESET_EEPROM
0	TRANSFER_COMPLETE 1=The transfer between the EEPROM fabric and the EEPROM Buffer Register has completed 0=The transfer between the EEPROM fabric and the EEPROM Buffer Register has not yet completed	R/WC	0h	RESET_EEPROM

38.12.4 EEPROM INTERRUPT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	EXECUTION_ERROR_IE Assert an EEPROM interrupt when the EXECUTION_ERROR status is asserted. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET_SYS
0	TRANSFER_COMPLETE_IE Assert an EEPROM interrupt when the TRANSFER_COMPLETE status is asserted. 1=Enable Interrupt 0=Disable Interrupt	R/W	0h	RESET_SYS

38.12.5 EEPROM PASSWORD REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	EEPROM_PASSWORD This register stores a key that can be used to unlock the EEPROM controller if it is locked. This register is write-once only. Once written, it can be neither read nor written until the next system reset.	W	0h	RESET_SYS

38.12.6 EEPROM UNLOCK REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	Reserved	RES	-	-
30:0	EEPROM_UNLOCK When this 31-bit register is written, the least significant 31 bits of the write are compared to the EEPROM Password Register that stores the key. If all bits match, the LOCK bit in the EEPROM Lock Register cleared, and the EEPROM array can be read or written. This register is write only.	W	0h	RESET_SYS

© 2020 Microchip Technology Inc.

38.12.7 EEPROM LOCK REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	LOCK EEPROM Access Lock. When this bit is set to '1b', the EEPROM is locked from all accesses, including reads, writes and status queries. Once set to '1b' it can only be cleared to '0b' by a RESET_SYS or by writing the EEPROM Unlock Register with a value that matches the contents of the EEPROM Password Register. 1=EEPROM is locked and cannot be accessed 0=EEPROM is unlocked and may be accessed	R/W	0h	RESET_SYS
0	JTAG_LOCK If this bit is set to '1b' the LOCK bit is set to '1b' whenever the JTAG/SWD test interface is activated. This has priority over the EEPROM Unlock Register register, so that writing the EEPROM Unlock Register register with a value that matches the EEPROM Password Register if JTAG/SWD is active at the time. 1=The LOCK bit is set to '1b' whenever the JTAG/SWD test interface is activated 0=The JTAG/SWD test interface has no effect on the LOCK bit	R/W	0h	RESET_SYS

38.12.8 EEPROM BUFFER REGISTER

Offset	20h to 3Fh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
	BUFFER Data buffer of 32 bytes used to transfer data to and from the EEPROM fabric. For WRITES, it must be written before the WRITE command is started. For READ STATUS and WRITE STATUS commands, only the first byte (offset 0 in this register) is used.	R/W	0h	RESET_SYS

39.0 PS/2 INTERFACE

39.1 Introduction

PS/2 controllers are directly controlled by the EC. The hardware implementation eliminates the need to bit bang I/O ports to generate PS/2 traffic, however bit banging is available via the associated GPIO pins.

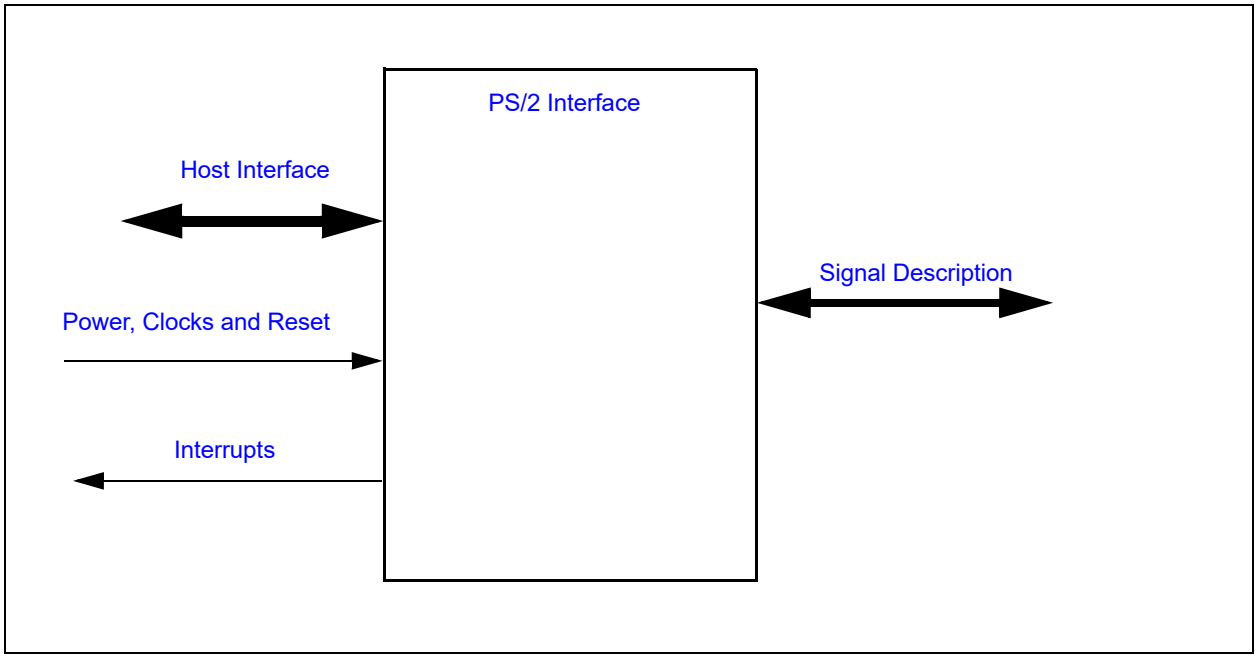
39.2 References

No references have been cited for this feature.

39.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 39-1: I/O DIAGRAM OF BLOCK



39.4 Signal Description

TABLE 39-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
PS2_DAT	INPUT/ OUTPUT	Data from the PS/2 device
PS2_CLK	INPUT/ OUTPUT	Clock from the PS/2 device

39.5 Host Interface

The registers defined for the PS/2 Interface are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

© 2020 Microchip Technology Inc.

39.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

39.6.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

39.6.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for PS/2 Interface logic.
2 MHz Clock	The PS/2 state machine is clocked using the 2 MHz clock.

39.6.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

39.7 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
PS2_x_ACT	Interrupt request to the Interrupt Aggregator for PS2 controller instance *x*, based on PS2 controller activity. Section 39.13.4, "PS2 Status Register" defines the sources for the interrupt request.
PS2_*x*_WK	Wake-up request to the Interrupt Aggregator's wake-up interface for PS2 port *x*. In order to enable PS2 wakeup interrupts, the pin control registers for the PS2DAT pin must be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

39.8 Low Power Modes

The PS/2 Interface may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

When set to be in Low power mode, PS2 interface will only sleep while the PS2 is disabled or in Rx mode with no traffic on the bus.

39.9 Description

Each EC PS/2 serial channels use a synchronous serial protocol to communicate with the auxiliary device. Each PS/2 channel has Clock and Data signal lines. The signal lines are bi-directional and employ open drain outputs capable of sinking 12m, as required by the PS/2 specification. A pull-up resistor, typically 10K, is connected to both lines. This allows either the EC PS/2 logic or the auxiliary device to drive the lines. Regardless of the drive source, the auxiliary device always provides the clock for transmit and receive operations. The serial packet is made up of eleven bits, listed in the order they appear on the data line: start bit, eight data bits (least significant bit first), odd parity, and stop bit. Each bit cell is from 60µS to 100µS long.

All PS/2 Serial Channel signals (PS2CLK and PS2DAT) are driven by open drain drivers which can be pulled to VTR1 or the main power rail (+3.3V nominal) through 10K-ohm resistors.

The PS/2 controller supports a PS/2 Wake Interface that can wake the EC from the IDLE or SLEEP states. The Wake Interface can generate wake interrupts without a clock. The PS/2 Wake Interface is only active when the peripheral device and external pull-up resisters are powered by the VTR1 supply.

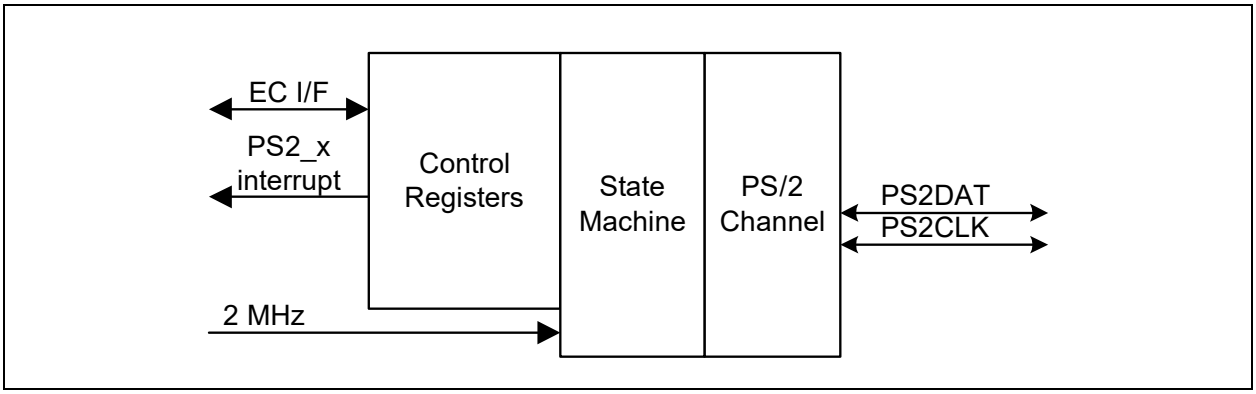
© 2020 Microchip Technology Inc.

There are no special precautions to be taken to prevent back drive of a PS/2 peripheral powered by the main power well when the power well is off, as long as the external 10K pull-up resistor is tied to the same power source as the peripheral.

PS/2 controllers may have one or two ports. Only one port may be active at a time. See the pin chapter for a definition of the PS/2 ports.

39.10 Block Diagram

FIGURE 39-2: PORT PS/2 BLOCK DIAGRAM



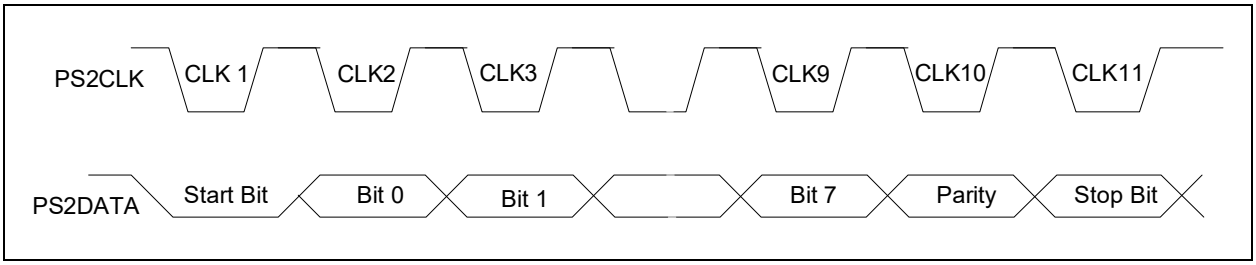
39.11 PS/2 Port Physical Layer Byte Transmission Protocol

The PS/2 physical layer transfers a byte of data via an eleven bit serial stream as shown in Table 39-2. A logic 1 is sent at an active high level. Data sent from a Keyboard or mouse device to the host is read on the falling edge of the clock signal. The Keyboard or mouse device always generates the clock signal. The Host may inhibit communication by pulling the Clock line low. The Clock line must be continuously high for at least 50 microseconds before the Keyboard or mouse device can begin to transmit its data. See Table 39-3, "PS/2 Port Physical Layer Bus States".

TABLE 39-2: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL

Bit	Function
1	Start bit (always 0)
2	Data bit 0 (least significant bit)
3	Data bit 1
4	Data bit 2
5	Data bit 3
6	Data bit 4
7	Data bit 5
8	Data bit 6
9	Data bit 7 (most significant bit)
10	Parity bit (odd parity)
11	Stop Bit (always 1)

FIGURE 39-3: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL



© 2020 Microchip Technology Inc.

TABLE 39-3: PS/2 PORT PHYSICAL LAYER BUS STATES

Data	Clock	State
high	high	Idle
high	low	Communication Inhibited
low	low	Request to Send

39.12 Controlling PS/2 Transactions

PS/2 transfers are controlled by fields in the PS2 Control Register.

The interface is enabled by the PS2_EN bit. Transfers are enabled when PS2_EN is '1' and disabled when PS2_EN is '0'. If the PS2_EN bit is cleared to '0' while a transfer is in progress but prior to the leading edge (falling edge) of the 10th (parity bit) clock edge, the receive data is discarded (RDATA_RDY remains low). If the PS2_EN bit is cleared following the leading edge of the 10th clock signal, then the receive data is saved in the Receive Register (RDATA_RDY goes high) assuming no parity error.

The direction of a PS/2 transfer is controlled by the PS2_T/R bit.

39.12.1 RECEIVE

If PS2_T/R is '0' while the PS2 Interface is enabled, the interface is configured to receive data. If while PS2_T/R is '0' RDATA_RDY is '0', the channel's PS2CLK and PS2DAT will float waiting for the external PS/2 device to signal the start of a transmission. If RDATA_RDY is '1', the channel's PS2DAT line will float but its PS2CLK line will be held low, holding off the peripheral, until the Receive Register is read.

The peripheral initiates a reception by sending a start bit followed by the data bits. After a successful reception, data are placed in the PS2 Receive Buffer Register, the RDATA_RDY bit in the PS2 Status Register is set and the PS2CLK line is forced low. Further receive transfers are inhibited until the EC reads the data in the PS2 Receive Buffer Register. RDATA_RDY is cleared and the PS2CLK line is tri-stated following a read of the PS2 Receive Buffer Register.

The Receive Buffer Register is initialized to FFh after a read or after a Time-out has occurred.

39.12.2 TRANSMIT

If PS2_T/R is '1' while the PS2 Interface is enabled, the interface is configured to transmit data. When the PS2_T/R bit is written to '1' while the state machine is idle, the channel prepares for a transmission: the interface will drive the PS2-CLK line low and then float the PS2DAT line, holding this state until a write occurs to the Transmit Register or until the PS2_T/R bit is cleared. A transmission is started by writing the PS2 Transmit Buffer Register. Writes to the Transmit Buffer Register are blocked when PS2_EN is '0', PS2_T/R is '0' or when the transmit state machine is active (the XMIT_IDLE bit in the PS/2 Status Register is '0'). The transmission of data will not start if there is valid data in the Receive Data Register (when the status bit RDATA_RDY is '1'). When a transmission is started, the transmission state machine becomes active (the XMIT_IDLE bit is set to '1' by hardware), the PS2DAT line is driven low and within 80ns the PS2CLK line floats (externally pulled high by the pull-up resistor).

The transmission terminates either on the 11th clock edge of the transmission or if a Transmit Time-Out error condition occurs. When the transmission terminates, the PS2_T/R bit is cleared to '0'and the state machine becomes idle, setting XMIT_IDLE to '1'.

The PS2_T/R bit must be written to a '1' before initiating another transmission to the remote device. If the PS2_T/R bit is set to '1' while the channel is actively receiving data (that is, while the status bit RDATA_RDYis '1') prior to the leading edge of the 10th (parity bit) clock edge, the receive data is discarded. If the bit is set after the 10th edge, the receive data is saved in the Receive Register.

© 2020 Microchip Technology Inc.

39.13 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the PS/2 Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 39-4: REGISTER SUMMARY

Offset	Register Name
0h	PS2 Transmit Buffer Register
0h	PS2 Receive Buffer Register
4h	PS2 Control Register
8h	PS2 Status Register

39.13.1 PS2 TRANSMIT BUFFER REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	TRANSMIT_DATA Writes to this register start a transmission of the data in this register to the peripheral.	W	0h	RESET_SYS

39.13.2 PS2 RECEIVE BUFFER REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	RECEIVE_DATA Data received from a peripheral are recorded in this register. A transmission initiated by writing the PS2 Transmit Buffer Register will not start until valid data in this register have been read and RDATA_RDY has been cleared by hardware. The Receive Buffer Register is initialized to FFh after a read or after a Time-out has occurred.	R	FFh	RESET_SYS

© 2020 Microchip Technology Inc.

39.13.3 PS2 CONTROL REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	-	-
5:4	STOP These bits are used to set the level of the stop bit expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set. 00b=Receiver expects an active high stop bit. 01b=Receiver expects an active low stop bit. 10b=Receiver ignores the level of the Stop bit (11th bit is not interpreted as a stop bit). 11b=Reserved.	R/W	0h	RESET_SYS
3:2	PARITY These bits are used to set the parity expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set. 00b=Receiver expects Odd Parity (default). 01b=Receiver expects Even Parity. 10b=Receiver ignores level of the parity bit (10th bit is not interpreted as a parity bit). 11b=Reserved	R/W	0h	RESET_SYS
1	PS2_EN PS/2 Enable. 0=The PS/2 state machine is disabled. The CLK pin is driven low and the DATA pin is tri-stated. 1=The PS/2 state machine is enabled, allowing the channel to perform automatic reception or transmission, depending on the state of PS2_T/R.	R/W	0h	RESET_SYS
0	PS2_T/R PS/2 Transmit/Receive 0=The P2/2 channel is enabled to receive data. 1=The PS2 channel is enabled to transmit data.	R/W	0h	RESET_SYS

Changing values in the PS2 CONTROL REGISTER at a rate faster than 2 MHz, may result in unpredictable behavior.

39.13.4 PS2 STATUS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7	XMIT_START_TIMEOUT Transmit Start Timeout. 0=No transmit start timeout detected 1=A start bit was not received within 25 ms following the transmit start event. The transmit start bit time-out condition is also indicated by the XMIT_TIMEOUT bit.	R/WC	0h	RESET_SYS
6	RX_BUSY Receive Channel Busy. 0=The channel is idle 1=The channel is actively receiving PS/2 data	R	0h	RESET_SYS
5	XMIT_TIMEOUT When the XMIT_TIMEOUT bit is set, the PS2_T/R bit is held clear, the PS/2 channel's CLK line is pulled low for a minimum of 300µs until the PS/2 Status register is read. The XMIT_TIMEOUT bit is set on one of three transmit conditions: when the transmitter bit time (the time between falling edges) exceeds 300µs, when the transmitter start bit is not received within 25ms from signaling a transmit start event or if the time from the first bit (start) to the 10th bit (parity) exceeds 2ms	R/WC	0h	RESET_SYS
4	XMIT_IDLE Transmitter Idle. 0=The channel is actively transmitting PS/2 data. Writing the PS2 Transmit Buffer Register will cause the XMIT_IDLE bit to clear 1=The channel is not transmitting. This bit transitions from '0' to '1' in the following cases: • The falling edge of the 11th CLK • XMIT_TIMEOUT is set • The PS2_T/R bit is cleared • The PS2_EN bit is cleared. A low to high transition on this bit generates a PS2 Activity interrupt.	R	1h	RESET_SYS
3	FE Framing Error When receiving data, the stop bit is clocked in on the falling edge of the 11th CLK edge. If the channel is configured to expect either a high or low stop bit and the 11th bit is contrary to the expected stop polarity, then the FE and REC_TIMEOUT bits are set following the falling edge of the 11th CLK edge and an interrupt is generated.	R/WC	0h	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	PE Parity Error When receiving data, the parity bit is clocked in on the falling edge of the 10th CLK edge. If the channel is configured to expect either even or odd parity and the 10th bit is contrary to the expected parity, then the PE and REC_TIMEOUT bits are set following the falling edge of the 10th CLK edge and an interrupt is generated.	R/WC	0h	RESET_SYS
1	REC_TIMEOUT Receive Timeout Following assertion of the REC_TIMEOUT bit, the channel's CLK line is automatically pulled low for a minimum of 300µs until the PS/2 status register is read. Under PS2 automatic operation, PS2_EN is set, this bit is set on one of three receive error conditions: • When the receiver bit time (the time between falling edges) exceeds 300µs. • If the time from the first bit (start) to the 10th bit (parity) exceeds 2ms. • On a receive parity error along with the Parity Error (PE) bit. • On a receive framing error due to an incorrect STOP bit along with the framing error (FE) bit. A low to high transition on this bit generates a PS2 Activity interrupt.	R/WC	0h	RESET_SYS
0	RDATA_RDY Receive Data Ready Under normal operating conditions, this bit is set following the falling edge of the 11th clock given successful reception of a data byte from the PS/2 peripheral (i.e., no parity, framing, or receive time-out errors) and indicates that the received data byte is available to be read from the Receive Register. This bit may also be set in the event that the PS2_EN bit is cleared following the 10th CLK edge. Reading the Receive Register clears this bit. A low to high transition on this bit generates a PS2 Activity interrupt.	R	0h	RESET_SYS

40.0 TRACE FIFO DEBUG PORT (TFDP)

40.1 Introduction

The TFDP serially transmits Embedded Controller (EC)-originated diagnostic vectors to an external debug trace system.

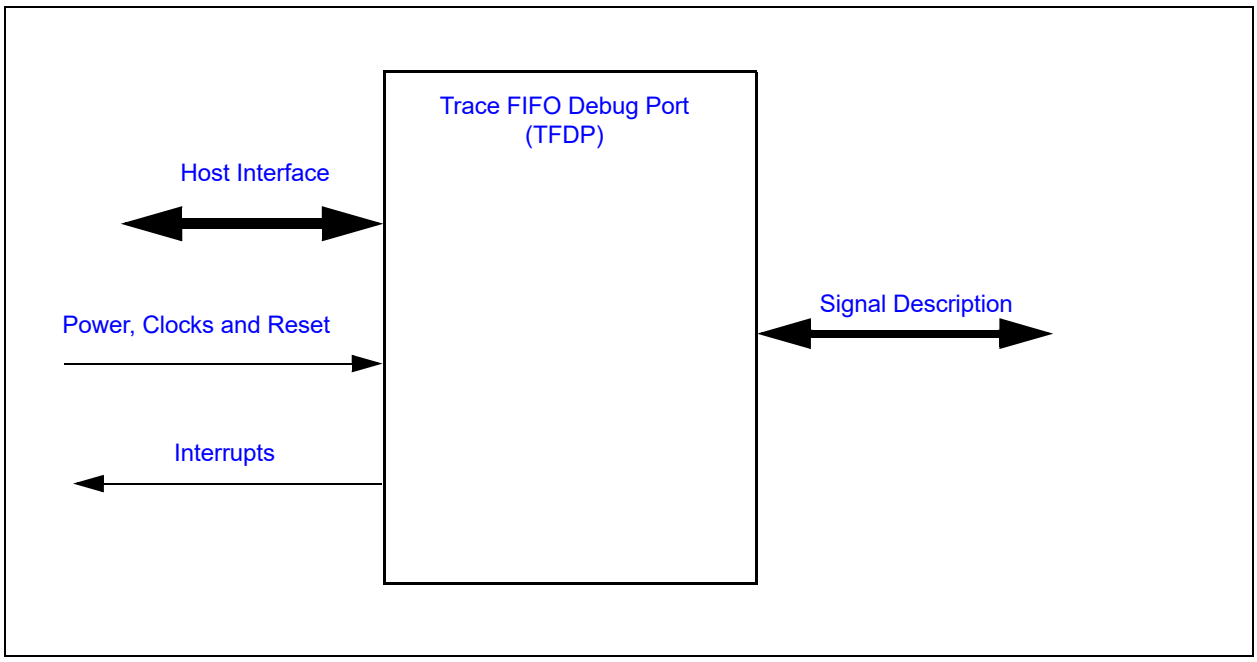
40.2 References

No references have been cited for this chapter.

40.3 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 40-1: I/O DIAGRAM OF BLOCK



40.4 Signal Description

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 40-1: SIGNAL DESCRIPTION

Name	Direction	Description
TFDP Clk	Output	Derived from EC Bus Clock.
TFDP Data	Output	Serialized data shifted out by TFDP Clk.

40.5 Host Interface

The registers defined for the Trace FIFO Debug Port (TFDP) are accessible by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

© 2020 Microchip Technology Inc.

40.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

40.6.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

40.6.2 CLOCK INPUTS

Name	Description
48MHz	This is the main system clock.

40.6.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

40.7 Interrupts

There are no interrupts generated from this block.

40.8 Low Power Modes

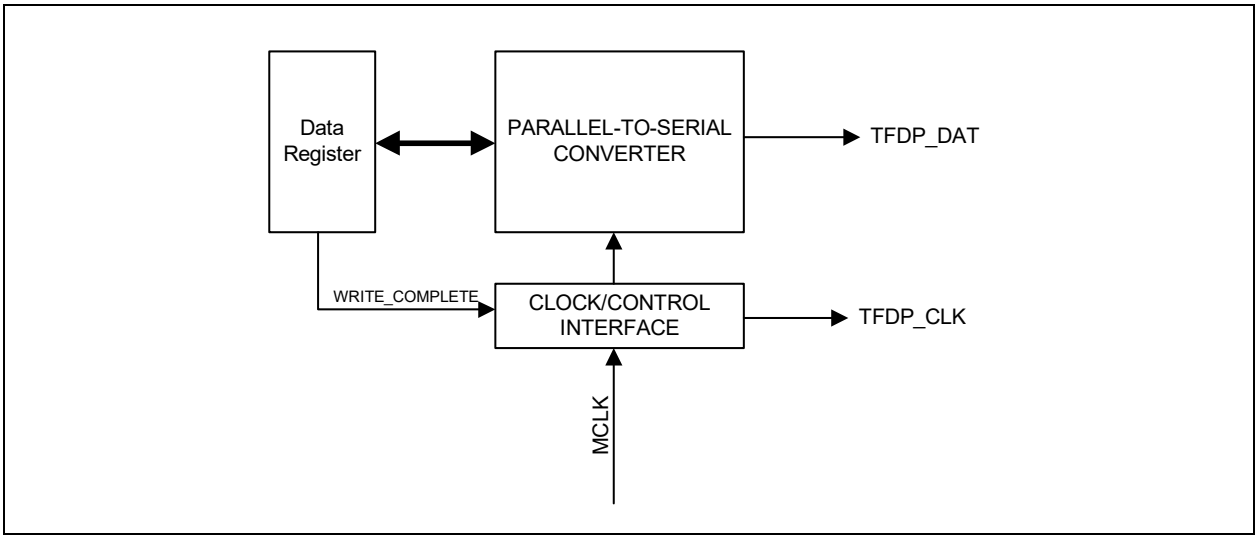
The Trace FIFO Debug Port (TFDP) may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

40.9 Description

The TFDP is a unidirectional (from processor to external world) two-wire serial, byte-oriented debug interface for use by processor firmware to transmit diagnostic information.

The TFDP consists of the Debug Data Register, Debug Control Register, a Parallel-to-Serial Converter, a Clock/Control Interface and a two-pin external interface (TFDP Clk, TFDP Data). See Figure 40-2.

FIGURE 40-2: BLOCK DIAGRAM OF TFDP DEBUG PORT

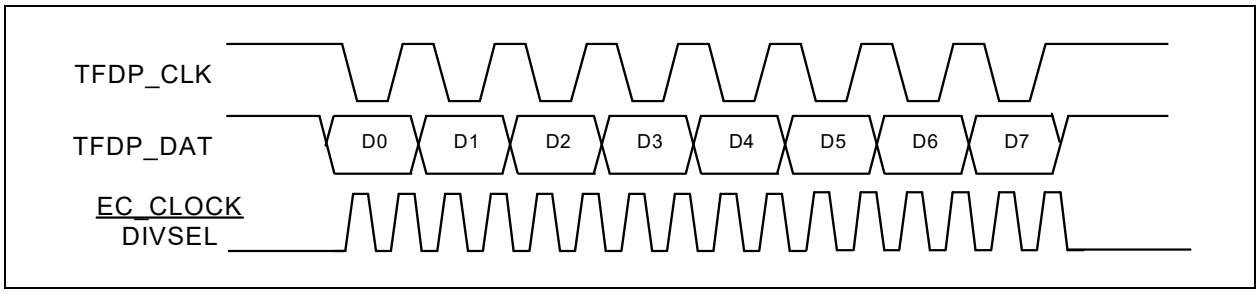


The firmware executing on the embedded controller writes to the Debug Data Register to initiate a transfer cycle (Figure 40-2). The Debug Data Register is loaded into a shift register and shifted out on TFDP_DAT LSB first at the programmed TFDP_CLK Clock rate (Figure 40-3).

Data is transferred in one direction only from the Debug Data Register to the external interface. The data is shifted out at the clock edge. The clock edge is selected by the EDGE_SEL bit in the Debug Control Register. After being shifted out, valid data will be presented at the opposite edge of the TFDP_CLK. For example, when the EDGE_SEL bit is '0' (default), valid data will be presented on the falling edge of the TFDP_CLK. The Setup Time (to the falling edge of TFDP_CLK) is 10 ns, minimum. The Hold Time is 1 ns, minimum.

When the Serial Debug Port is inactive, the TFDP_CLK and TFDP_DAT outputs are '1.' The EC Bus Clock clock input is the transfer clock.

FIGURE 40-3: DATA TRANSFER



40.10 EC-Only Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the Trace FIFO Debug Port (TFDP) Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 40-2: REGISTER SUMMARY

Offset	Register Name
00h	Debug Data Register
04h	Debug Control Register

40.10.1 DEBUG DATA REGISTER

The Debut Data Register is Read/Write. It always returns the last data written by the TFDP or the power-on default '00h'.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	DATA Debug data to be shifted out on the TFDP Debug port. While data is being shifted out, the Host Interface will 'hold-off' additional writes to the data register until the transfer is complete.	R/W	00h	RESET_SYS

© 2020 Microchip Technology Inc.

40.10.2 DEBUG CONTROL REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Reserved	RES	-	-
6:4	IP_DELAY Inter-packet Delay. The delay is in terms of TFDP Debug output clocks. A value of 0 provides a 1 clock inter-packet period, while a value of 7 provides 8 clocks between packets:	R/W	000b	RESET_SYS
3:2	DIVSEL Clock Divider Select. The TFDP Debug output clock is determined by this field, according to Table 40-3, "TFDP Debug Clocking":	R/W	00b	RESET_SYS
1	EDGE_SEL 1=Data is shifted out on the falling edge of the debug clock 0=Data is shifted out on the rising edge of the debug clock (Default)	R/W	0b	RESET_SYS
0	EN Enable. 1=Clock enabled 0=Clock is disabled (Default)	R/W	0b	RESET_SYS

TABLE 40-3: TFDP DEBUG CLOCKING

divsel	TFDP Debug Clock
00	24 MHz
01	12 MHz
10	6 MHz
11	Reserved

41.0 PORT 80 BIOS DEBUG PORT

41.1 Overview

The Port 80 BIOS Debug Port emulates the functionality of a “Port 80” ISA plug-in card. In addition, a timestamp for the debug data can be optionally added.

Diagnostic data is written by the Host Interface to the Port 80 BIOS Debug Port, which is located in the Host I/O address space. The Port 80 BIOS Debug Port generates an interrupt to the EC when host data is available. The EC reads this data along with the timestamp, if enabled.

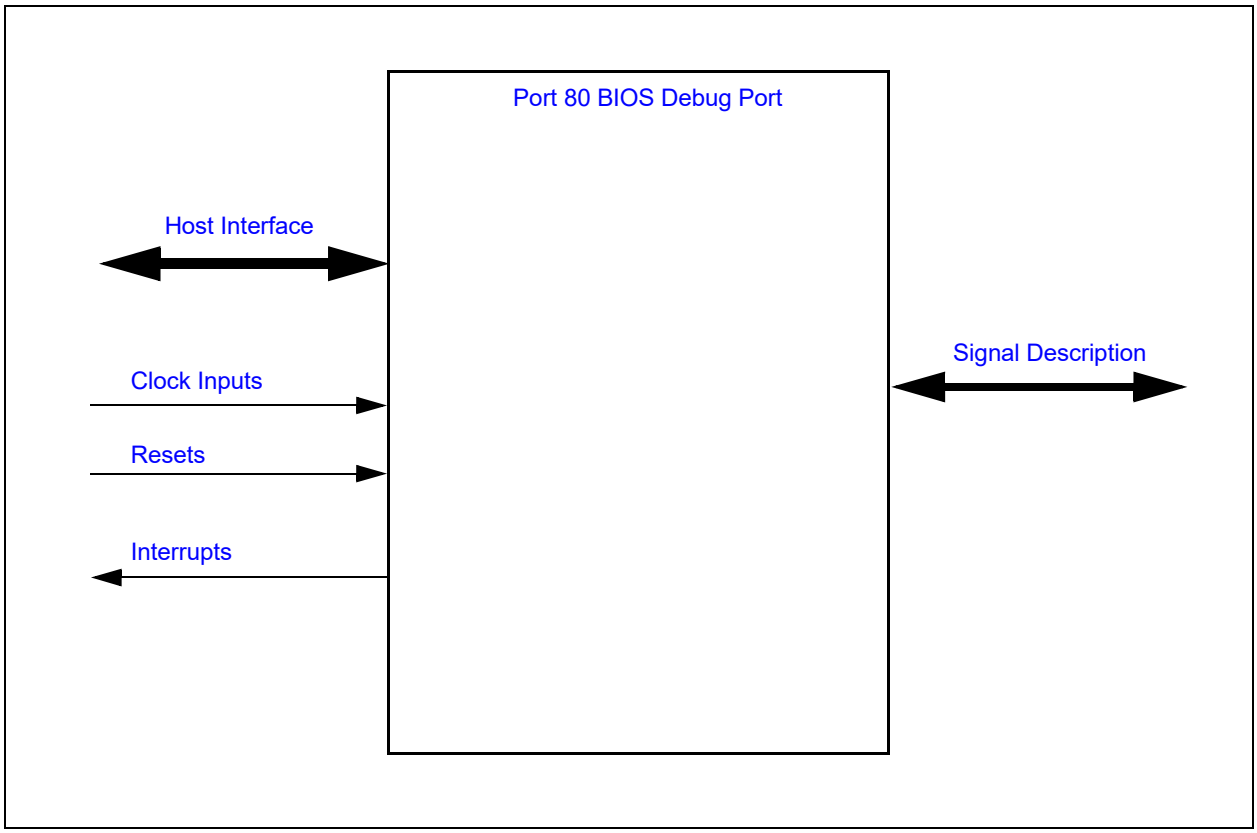
41.2 References

There are no references for this block.

41.3 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 41-1: I/O DIAGRAM OF BLOCK



41.4 Signal Description

There are no external signals for this block.

41.5 Host Interface

The registers for Port 80 block is accessed by the various hosts as indicated in Section 3.2, "Block Overview and Base Addresses".

© 2020 Microchip Technology Inc.

41.6 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

41.6.1 POWER DOMAINS

Name	Description
VTR_CORE	This Power Well is used to power the registers and logic in this block.

41.6.2 CLOCK INPUTS

Name	Description
48MHz	This is the clock source for Port 80 block logic.

41.6.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

41.7 Interrupts

This section defines the Interrupt Sources generated from this block.

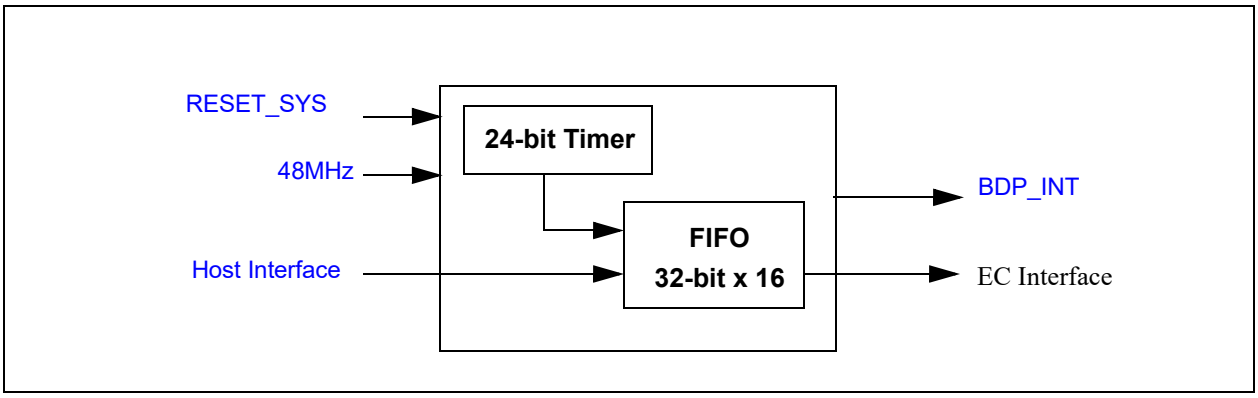
Source	Description
BDP_INT	The Port 80 BIOS Debug Port generates an EC interrupt when the amount of data in the Port 80 FIFO equals or exceeds the FIFO Threshold defined in the Configuration Register. The interrupt signal is always generated by the Port 80 block if the block is enabled; the interrupt is enabled or disabled in the Interrupt Aggregator.

41.8 Low Power Modes

The Port 80 block may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

41.9 Description

FIGURE 41-2: PORT 80 BLOCK DIAGRAM



The Port 80 BIOS Debug Port consists of a 32-bit wide x 16 deep FIFO and a 24-bit free running timer. Host and EC access to the Port 80 device is through a set of registers. The Host can write the FIFO via the Runtime Registers and the EC can read the FIFO can control the device via the EC Registers.

Writes to the Host Data Register are concatenated with the 24-bit timestamp and written to the FIFO. Reads of the Host Data Register return zero. If writes to the Host Data Register overrun the FIFO, the oldest data are discarded and the OVERRUN status bit in the Status Register is asserted.

Only the EC can read data from the FIFO, using the EC Data Register. The use of this data is determined by EC Firmware alone.

41.10 Configuration Registers

Configuration Registers for an instance of the Port 80 BIOS Debug Port are listed in the following table. Host access to Configuration Registers is through the Configuration Port using the Logical Device Number of each instance of the Port 80 BIOS Debug Port and the Index shown in the “Host Index” column of the table. The EC can access Configuration Registers directly. The EC address for each register is formed by adding the Base Address for each instance of the Port 80 BIOS Debug Port shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “EC Offset” column.

TABLE 41-1: CONFIGURATION REGISTER SUMMARY

EC Offset	Host Index	Register Name
330h	30h	Activate Register

41.10.1 ACTIVATE REGISTER

Offset	330h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	Reserved	RES	-	-
0	ACTIVATE When this bit is asserted ‘1’, the block is enabled. When this bit is ‘0’, writes by the Host interface to the Host Data Register are not claimed, the FIFO is flushed, the 24-bit Timer is reset, and the timer clock is stopped. Control bits in the Configuration Register are not affected by the state of ACTIVATE.	R/W	0h	RESET_SYS

41.11 Runtime Registers

The registers listed in the Runtime Register Summary table are for a single instance of the Port 80 BIOS Debug Port. Host access for each register listed in this table is defined as an offset in the Host address space to the Block’s Base Address, as defined by the instance’s Base Address Register.

The EC address for each register is formed by adding the Base Address for each instance of the Port 80 BIOS Debug Port shown in the Block Overview and Base Address Table in Section 3.0, "Device Inventory" to the offset shown in the “Offset” column.

TABLE 41-2: RUNTIME REGISTER SUMMARY

Offset	Register Name
00h	Host Data Register

© 2020 Microchip Technology Inc.

41.11.1 HOST DATA REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	HOST_DATA	W	0h	RESET_SYS

41.12 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for each instance of the Port 80 BIOS Debug Port Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 41-3: EC REGISTER SUMMARY

Offset	Register Name
100h	EC Data Register
104h	Configuration Register
108h	Status Register
10Ch	Count Register

41.12.1 EC DATA REGISTER

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	TIME_STAMP	R	0h	RESET_SYS
7:0	EC_DATA	R	0h	RESET_SYS

41.12.2 CONFIGURATION REGISTER

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:6	FIFO_THRESHOLD This field determines the threshold for the Port 80 BIOS Debug Port Interrupts. 3=14 entry threshold 2=8 entry threshold 1=4 entry threshold 0=1 entry threshold	R/W	0h	RESET_SYS
5	TIMER_ENABLE When the TIMER_ENABLE bit is ‘1’, the 24-bit Timer is actively counting at a rate determined by the TIMEBASE_SELECT bits. When the TIMER ENABLE bit is ‘0’, counting is stopped.	R/W	0h	RESET_SYS

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
4:3	TIMEBASE_SELECT The TIMEBASE SELECT bits determine the clock for the 24-bit Timer. 3=48MHz/64 2=48MHz/32 1=48MHz/16 0=48MHz/8	R/W	0h	RESET_SYS
2	RESET_TIMESTAMP When this field is written with a '1', the 24-bit Timer is reset to '0'. Writing zero to the Count Register has the same effect. Writes of a '0' to this field have no effect. Reads always return '0'.	W	-	RESET_SYS
1	FLUSH When this field is written with a '1', the FIFO is flushed. Writes of a '0' to this field have no effect. Reads always return '0'.	W	-	RESET_SYS
0	Reserved	RES	-	-

41.12.3 STATUS REGISTER

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	Reserved	RES	-	-
1	OVERRUN The OVERRUN bit is '1' when the host writes the Host Data Register when the FIFO is full.	R	0h	RESET_SYS
0	NOT_EMPTY The NOT EMPTY bit is '1' when there is data in the FIFO. The NOT EMPTY bit is '0' when the FIFO is empty.	R	0h	RESET_SYS

41.12.4 COUNT REGISTER

Offset	10Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
32:8	COUNT Writes load data into the 24-bit Timer. Reads return the 24-bit Timer current value.	R/W	0h	–
7:0	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

42.0 VBAT-POWERED CONTROL INTERFACE

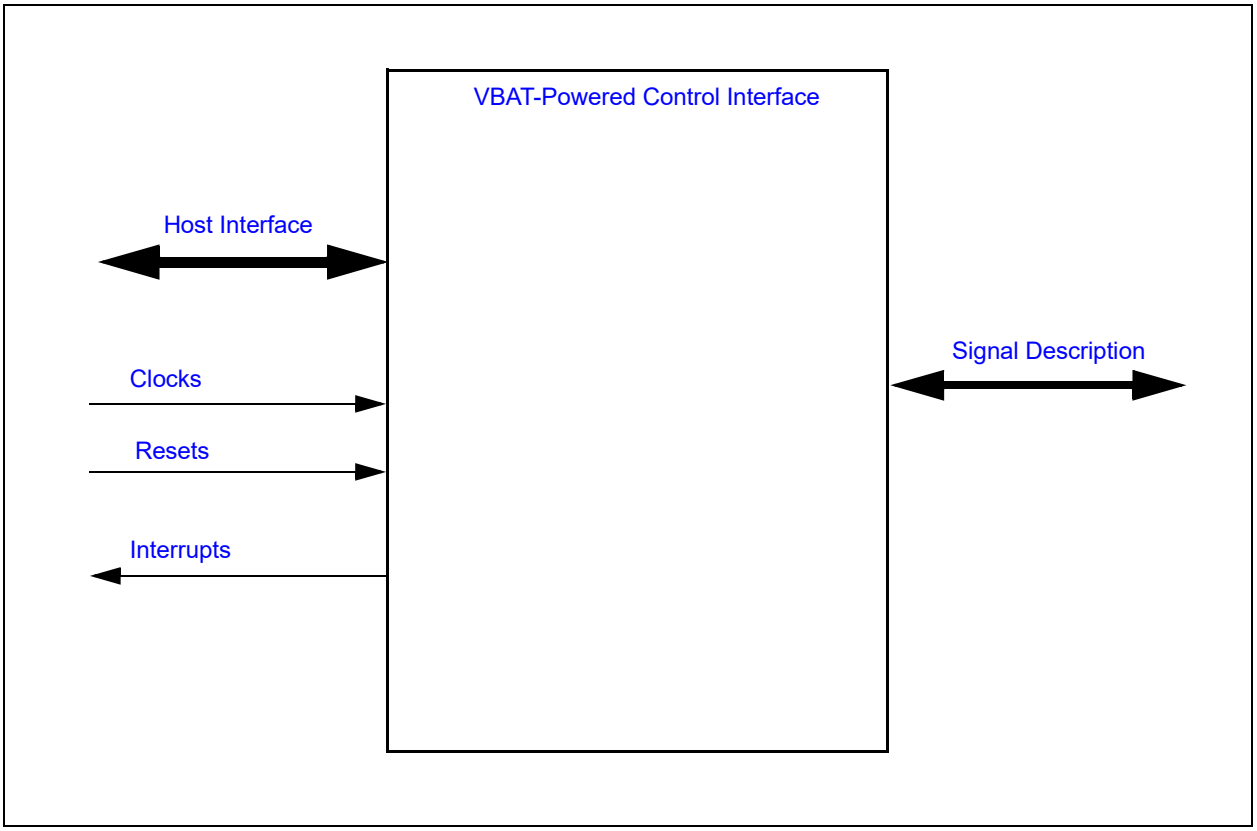
42.1 General Description

The VBAT-Powered Control Interface (VCI) has VBAT-powered combinational logic and input and output signal pins. The block interfaces with the Real Time Clock as well as the Week Alarm.

42.2 Interface

This block is designed to be accesses externally via the pin interface and internally via a registered host interface.

FIGURE 42-1: I/O DIAGRAM OF BLOCK



42.3 Signal Description

TABLE 42-1: EXTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
VCI_IN[3:0]	INPUT	Active-low inputs that can cause wakeup or interrupt events. **Note:** The VCI IP supports up to seven VCI_IN inputs. These inputs are generically referred to as VCI_INx. Input signals not routed to pins or balls on the package are connected to VBAT.
VCI_OVRD_IN	INPUT	Active high input that can cause wakeup or interrupt events.
VCI_OUT	OUTPUT	Output status driven by this block.

TABLE 42-2: INTERNAL SIGNAL DESCRIPTION

Name	Direction	Description
Week_Alarm	INPUT	Signal from the Week Timer block. The alarm is asserted by the timer when the Week_Alarm Power-Up Output is asserted
RTC_Alarm	INPUT	Signal from the Real Time Clock block. The alarm is asserted by the RTC when the RTC_ALRM signal is asserted.
VTR_PWRGD	INPUT	Status signal for the state of the VTR power rail. This signal is high if the power rail is on, and low if the power rail is off.

42.4 Host Interface

The registers defined for the VBAT-Powered Control Interface are accessible only by the EC.

42.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

42.5.1 POWER DOMAINS

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR_CORE	The power well sources register access by the host. The block continues to operate internally while this rail is down

42.5.2 CLOCKS

This block does not require clocks.

42.5.3 RESETS

Name	Description
RESET_VBAT	This reset signal is used reset all of the registers and logic in this block.
RESET_SYS	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR_CORE powered circuitry on-chip. Otherwise it has no effect on the internal state.

42.6 Interrupts

Source	Description
VCI_IN[3:0]	These interrupts are routed to the Interrupt Aggregator. They are only asserted when both VBAT and VTR_CORE are powered. Edge detection and assertion level for the interrupt are configured in the GPIO Pin Control Registers for the GPIOs that shares pins with VCI_INx# inputs. The interrupts are equivalent to the GPIO interrupts for the GPIOs that share the pins, but appear on different registers in the Interrupt Aggregator.
VCI_OVRD_IN	This interrupt is routed to the Interrupt Aggregator. It is only asserted when both VBAT and VTR_CORE are powered.

42.7 Low Power Modes

The VBAT-powered Control Interface has no low-power modes. It runs continuously while the VBAT well is powered.

© 2020 Microchip Technology Inc.

42.8 General Description

The VBAT-Powered Control Interface (VCI) is used to drive the VCI_OUT pin. The output pin can be controlled either by VBAT-powered inputs, or by firmware when the VTR_CORE is active and the EC is powered and running. When the VCI_OUT pin is controlled by hardware, either because VTR_CORE is inactive or because the VCI block is configured for hardware control, the VCI_OUT pin can be asserted by a number of inputs:

- When one or more of the VCI_INx# pins are asserted. By default, the VCI_INx# pins are active low, but firmware can switch each input individually to an active-high input. See Section 42.8.1, "Input Polarity".
- When the Week Alarm from the Week Alarm Interface is asserted
- When the RTC Alarm from the Real Time Clock is asserted

Firmware can configure which of the hardware pin inputs contribute to the VCI_OUT output by setting the enable bits in the VCI Input Enable Register. Even if the input pins are not configured to affect VCI_OUT, firmware can monitor their current state through the status bits in the VCI Register. Firmware can also enable EC interrupts from the state of the input pins.

Each of the VCI_INx# pins can be configured for additional properties.

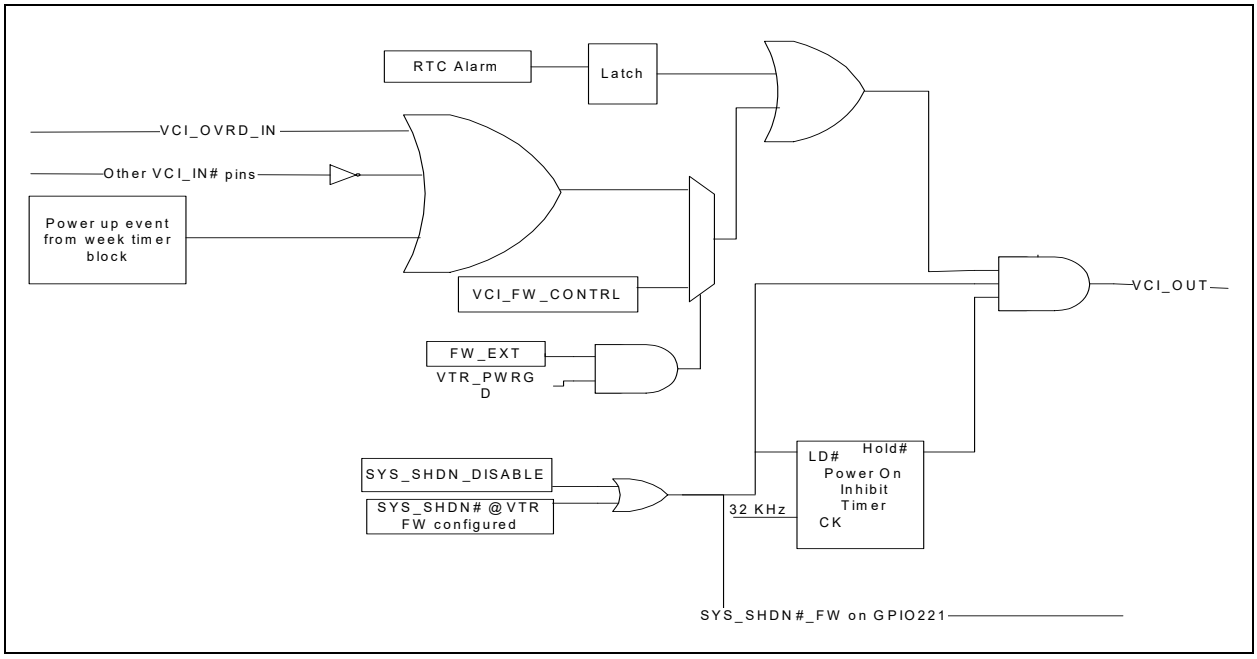
- By default, each of the VCI_INx# pins have an input glitch filter. All glitch filters can be disabled by the FILTERS_BYPASS bit in the VCI Register
- Assertions of each of the VCI_INx# pins can optionally be latched, so hardware can maintain the assertion of a VCI_INx# even after the physical pin is de-asserted, or so that firmware can determine which of the VCI_INx# inputs contributed to VCI_OUT assertion. See the Latch Enable Register and the Latch Resets Register.
- Rising edges and falling edges on the VCI_INx# pins are latched, so firmware can detect transitions on the VCI_INx# pins even if the transitions occurred while EC power was not available. See Section 42.8.2, "Edge Event Status".

If none of the additional properties are required, firmware can disable a VCI_INx# pin completely, by clearing both the corresponding bit in the VCI Input Enable Register and the corresponding bit in the VCI Buffer Enable Register. When both bits are ‘0’, the input is disabled and will not be a drain on the VBAT power rail.

When VTR_CORE power is present and the EC is operating, firmware can configure the VCI_OUT pin to operate as a general-purpose output pin. The VCI_OUT pin is firmware-controlled when the FW_EXT bit in the VCI Register is ‘1’. When firmware is controlling the output, the state of VCI_OUT is defined by the VCI_FW_CNTRL bit in the same register. When VTR_CORE is not present (the VTR_PWRGD input is low), the VCI_OUT pin is also determined by the hardware circuit.

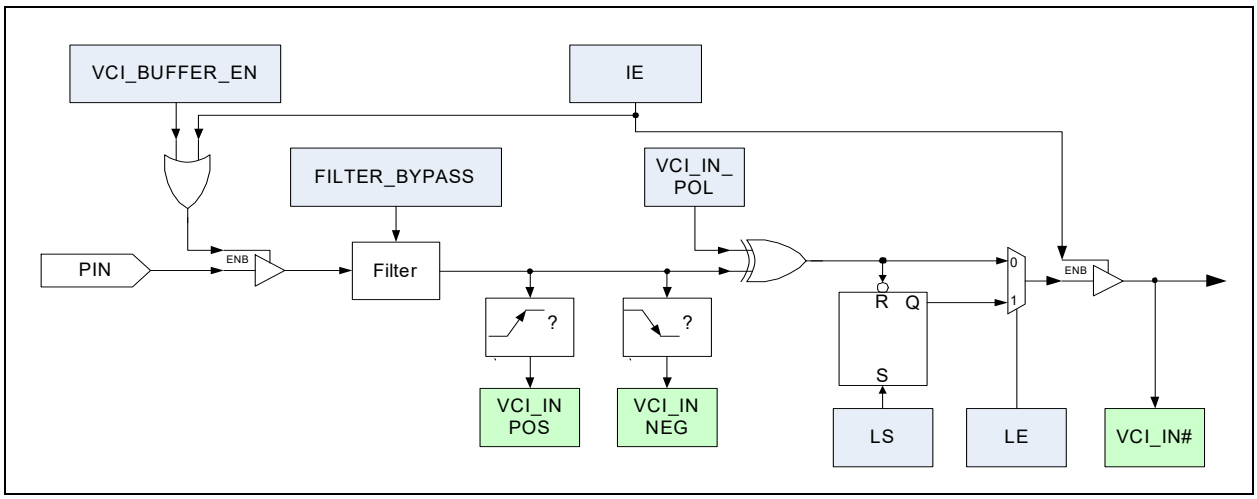
The following figures illustrate the VBAT-Power Control Interface logic:

FIGURE 42-2: VCI_OUT BLOCK DIAGRAM



The VCI_INx# Logic in the block diagram is illustrated in the following figure:

FIGURE 42-3: VBAT-POWERED CONTROL INTERFACE BLOCK DIAGRAM



42.8.1 INPUT POLARITY

The VCI_INx# pins have an optional polarity inversion. The inversion takes place after any input filtering and before the VCI_INx# signals are latched in the VCI_INx# status bits in the VCI Register. Edge detection occurs before the polarity inversion. The inversion is controlled by battery-backed configuration bits in the VCI Polarity Register.

42.8.2 EDGE EVENT STATUS

Each VCI_INx# input pin is associated with two register bits used to record edge transitions on the pins. The edge detection takes place after any input filtering, before polarity control and occurs even if the VCI_INx# input is not enabled as part of the VCI_OUT logic (the corresponding control bit in the VCI Input Enable Register is '0') or if the state of the

© 2020 Microchip Technology Inc.

VCI_INx# input is not latched (the corresponding control bit in the Latch Enable Register is '0'). One bit is set whenever there is a high-to-low transition on the VCI_INx# pin (the VCI Negedge Detect Register) and the other bit is set whenever there is a low-to-high transition on the VCI_INx# pin (the VCI Posedge Detect Register).

In order to minimize power drain on the VBAT circuit, the edge detection logic operates only when the input buffer for a VCI_INx# pin is enabled. The input buffer is enabled either when the VCI_INx# pin is configured to determine the VCI_OUT pin, as controlled by the VCI_IN[1:0]# field of the VCI Register, or when the input buffer is explicitly enabled in the VCI Input Enable Register. When the pins are not enabled transitions on the pins are ignored.

The VCI_OVRD_IN input also has an Input Buffer Enable and an Input Enable bit associated with VCI_OUT. However, the VCI_OVRD_IN input does not have any filtering, latching, input edge detection or polarity control.

42.8.3 VCI PIN MULTIPLEXING

Each of the VCI inputs, as well as VCI_OUT, are multiplexed with standard VTR_CORE-powered GPIOs. When VTR_CORE power is off, the mux control is disabled and the pin always reverts to the VCI function. The VCI_INx# function should be disabled in the VCI Input Enable Register VCI Buffer Enable Register and for any pin that is intended to be used as a GPIO rather than a VCI_INx#, so that VCI_OUT is not affected by the state of the pin.The VCI_OVRD_IN function should similarly be disabled if the pin is to be used as a GPIO.

42.8.4 POWER ON INHIBIT TIMER

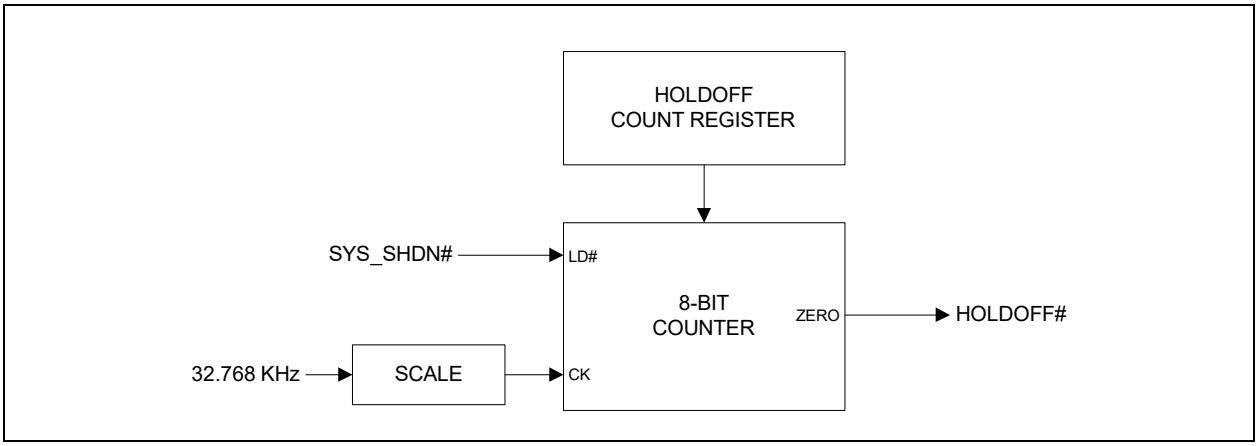
The Power On Inhibit Timer prevents the VBAT-Powered Control Interface VCI_OUT pin from being asserted for a programmable time period after the SYS_SHDN# pin asserted. This holdoff time can be used to give a system the opportunity to cool down after a thermal shutdown before allowing a user to attempt to turn the system on. While the Inhibit Timer is asserted, the VCI_OUT pin remains de-asserted and is unaffected by the VCI, Week Alarm and RTC interfaces.

The holdoff time is configured using the Holdoff Count Register. By setting the Holdoff Count Register to 0 the Inhibit Timer is disabled. When disabled, the HOLDOFF# signal is de-asserted and no counting takes place.

The HOLDOFF# output is asserted within one 32.768KHz clock cycle from the time SYS_SHDN# is asserted.

The following figure illustrates the operation of the Inhibit Timer:

FIGURE 42-4: POWER ON INHIBIT TIMER



The SCALE function reduces the 32.768KHz clock to 8Hz, so that the 8-bit counter counts intervals of 125ms. The following table shows some of examples of the effect of several settings of the Holdoff Count Register:

TABLE 42-3: HOLDOFF TIMING EXAMPLES

Holdoff Count Register	**Holdoff Time (SEC)**
0	Disabled (default)
1	0.125
5	0.625
10	1.25
15	1.875

TABLE 42-3: HOLDOFF TIMING EXAMPLES (CONTINUED)

Holdoff Count Register	Holdoff Time (SEC)
100	12.5
150	18.75
200	25
255	31.875

42.8.5 APPLICATION EXAMPLE

For this example, a mobile platform configures the VBAT-Powered Control Interface (VCI) as follows:

- VCI_IN0# is wired to a power button on the mobile platform
- VCI_IN1# is wired to a power button on a dock
- The VCI_OUT pin is connected to the regulator that sources the VTR power rail, the rail which powers the EC

The VCI can be used in a system as follows:

1. In the initial condition, there is no power on either the VTR or VBAT power rails. All registers in the VCI are in an indeterminate state
2. A coin cell battery is installed, causing a RESET_VBAT. All registers in the interface are forced to their default conditions. The VCI_OUT pin is driven by hardware, input filters on the VCI_INx# pins are enabled, the VCI_INx# pins are all active low, all VCI inputs are enabled and all edge and status latches are in their non-asserted state
3. The power button on VCI_IN0# is pushed. This causes VCI_OUT to be asserted, powering the VTR rail. This causes the EC to boot and start executing EC firmware
4. The EC changes the VCI configuration so that firmware controls the VCI_OUT pin, and sets the output control so that VCI_OUT is driven high. With this change, the power button can be released without removing the EC power rail.
5. EC firmware re-configures the VCI logic so that the VCI_INx# input latches are enabled. This means that subsequent presses of the power button do not have to be held until EC firmware switches the VCI logic to firmware control
6. During this phase the VCI_OUT pin is driven by the firmware-controlled state bit and the VCI input pins are ignored. However, the EC can monitor the state of the pins, or generate inputs when their state changes
7. At some later point, EC firmware must enter a long-term power-down state.
   - Firmware configures the Week Timer for a Week Alarm once every 8 hours. This will turn on the EC power rail three times a day and enable the EC to perform low frequency housekeeping tasks even in its lowest-power state
   - Firmware de-asserts VCI_OUT. This action kills power to the EC and automatically returns control of the VCI_OUT pin to hardware.
   - The EC will remain in its lowest-power state until a power pin is pushed, AC power is connected, or the Sub-Week Alarm is active

© 2020 Microchip Technology Inc.

42.9 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the VBAT-Powered Control Interface Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 42-4: REGISTER SUMMARY

EC Offset	Register Name
00h	VCI Register
04h	Latch Enable Register
08h	Latch Resets Register
0Ch	VCI Input Enable Register
10h	Holdoff Count Register
14h	VCI Polarity Register
18h	VCI Posedge Detect Register
1Ch	VCI Negedge Detect Register
20h	VCI Buffer Enable Register

42.9.1 VCI REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:19	Reserved	RES	-	-
18	MCHP reserved Should be set to zero		0	RESET_VBAT
18	SYSPWR_PRES/ VCI_IN3# Select 1= SYSPWR_PRES selected. 0= VCI_IN3# is selected.	R/W	0	RESET_VBAT
17	RTC_ALRM If enabled by RTC_ALRM_LE, this bit is set to '1' if the RTC Alarm signal is asserted. It is reset by writes to RTC_ALRM_LS.	R	0	RESET_VBAT
16	WEEK_ALRM If enabled by WEEK_ALRM_LE, this bit is set to '1' if the Week Alarm signal is asserted. It is reset by writes to WEEK_ALRM_LS.	R	0	RESET_VBAT
15:13	Reserved	RES	-	-
12	FILTERS_BYPASS The Filters Bypass bit is used to enable and disable the input filters on the VCI_INx# pins. See Section 47.17, "VBAT-Powered Control Interface Timing". 1=Filters disabled 0=Filters enabled (default)	R/W	0	RESET_VBAT
Note 1:	The VCI_IN[3:0]# and VCI_OVRD_INbits default to the state of their respective input pins. The VCI_OUT bit is determined by the VCI hardware circuit			

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
11	FW_EXT This bit controls selecting between the external VBAT-Powered Control Interface inputs, or the VCI_FW_CNTRL bit output to control the VCI_OUT pin. 1=VCI_OUT is determined by the VCI_FW_CNTRL field, when VTR_CORE is active **Note:** 0=VCI_OUT is determined by the external inputs.	R/W	0	RESET_SYS and RESET_VBAT
10	VCI_FW_CNTRL This bit can allow EC firmware to control the state of the VCI_OUT pin. For example, when VTR_PWRGD is asserted and the FW_EXT bit is '1', clearing the VCI_FW_CNTRL bit de-asserts the active high VCI_OUT pin. BIOS must set this bit to '1' prior to setting the FW_EXT bit to '1' on power up, in order to avoid glitches on the VCI_OUT pin.	R/W	0	RESET_SYS and RESET_VBAT
9	VCI_OUT This bit provides the current status of the VCI_OUT pin.	R	See Note 1	–
8	VCI_OVRD_IN This bit provides the current status of the VCI_OVRD_IN pin.	R	-	–
8	Reserved	RES	-	-
7	VCI_OUT/GPIO Select If this bit is 1 the signal is powered by VBAT and the output pin will be driven by VCI_OUT logic. If this bit is 0 the signal is powered by VTRx and the output pin will be driven according to GPIO Pin Control Registers. 1= VBAT Powered 0= VTR Powered	R/W	1	RESET_SYS and RESET_VBAT
6:4	Reserved	RES	-	-
3:0	VCI_IN[3:0]# These bits provide the latched state of the associated VCI_INx# pin, if latching is enabled or the current state of the pin if latching is not enabled. In both cases, the value is determined after the action of the VCI Polarity Register.	R	See Note 1	
Note 1: The VCI_IN[3:0]# and VCI_OVRD_INbits default to the state of their respective input pins. The VCI_OUT bit is determined by the VCI hardware circuit				

© 2020 Microchip Technology Inc.

42.9.2 LATCH ENABLE REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:18	Reserved	RES	-	-
17	RTC_ALRM_LE Latch enable for the RTC Power-Up signal. 1=Enabled. Assertions of the RTC Alarm are held until the latch is reset by writing the correspondingLS[3:0] bit 0=Not Enabled. The RTC Alarm signal is not latched but passed directly to the VCI_OUT logic	R/W	0h	RESET_VBAT
16	WEEK_ALRM_LE Latch enable for the Week Alarm Power-Up signal. 1=Enabled. Assertions of the Week Alarm are held until the latch is reset by writing the correspondingLS[3:0] bit 0=Not Enabled. The Week Alarm signal is not latched but passed directly to the VCI_OUT logic	R/W	0h	RESET_VBAT
15:4	Reserved	RES	-	-
3:0	LE[3:0] Latching Enables. Latching occurs after the Polarity configuration, so a VCI_INx# pin is asserted when it is ‘0’ if VCI_IN_POL[3:0] is ‘0’, and asserted when it is ‘1 ‘if VCI_IN_POL[3:0] is ‘1’. For each LE[x] bit in the field: 1=Enabled. Assertions of the VCI_INx# pin are held until the latch is reset by writing the corresponding LS[3:0] bit 0=Not Enabled. The VCI_INx# signal is not latched but passed directly to the VCI_OUT logic	R/W	30h	RESET_VBAT

42.9.3 LATCH RESETS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:18	Reserved	RES	-	-
17	RTC_ALRM_LS RTC Alarm Latch Reset. When this bit is written with a ‘1’, the RTC Alarm Event latch is reset The RTC Alarm input to the latch has priority over the Reset input Reads of this register are undefined.	W	-	–
16	WEEK_ALRM_LS Week Alarm Latch Reset. When this bit is written with a ‘1’, the Week Alarm Event latch is reset The Week Alarm input to the latch has priority over the Reset input Reads of this register are undefined.	W	-	–
15:4	Reserved	RES	-	-

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3:0	LS[3:0] Latch Resets. When a Latch Resets bit (LS[x]) is written with a '1', the corresponding VCI_IN*x*# latch is de-asserted ('1'). The VCI_IN*x*# input to the latch has priority over the Latch Reset input, so firmware cannot reset the latch while the VCI_INx# pin is asserted. Firmware should sample the state of the pin in the VCI Register before attempting to reset the latch. As noted in the Latch Enable Register, the assertion level is determined by the VCI_IN_POL[3:0] bit. Reads of this register are undefined.	W	–	–

42.9.4 VCI INPUT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	IE[3:0] Input Enables for VCI_INx# signals. After changing the input enable for a VCI input, firmware should reset the input latch and clear any potential interrupt that may have been triggered by the input, as changing the enable may cause the internal status to change. For each IE[x] bit in the field: 1=Enabled. The corresponding VCI_INx# input is not gated and toggling the pin will affect the VCI_OUT pin 0=Not Enabled. The corresponding VCI_INx# input does not affect the VCI_OUT pin, even if the input is '0.' Unless the corresponding bit in the VCI Buffer Enable Register is 1, latches are not asserted, even if the VCI_INx# pin is low, during a VBAT power transition	R/W	7h	RESET_VBAT

42.9.5 HOLDOFF COUNT REGISTER

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	-
7:0	HOLDOFF_TIME These bits determine the period of time the VCI_OUT logic is inhibited from re-asserting VCI_OUT after a SYS_SHDN# event. FFh-01h=The Power On Inhibit Holdoff Time is set to a period between 125ms and 31.875 seconds. See Table 42-3 for examples 0=The Power On Inhibit function is disabled	RW	0	RESET_VBAT

© 2020 Microchip Technology Inc.

42.9.6 VCI POLARITY REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	VCI_IN_POL[3:0] These bits determine the polarity of the VCI_INx input signals: For each VCI_IN_POL[x] bit in the field: 1=Active High. The value on the pins is inverted before use 0=Active Low (default)	RW	0	RESET_VBAT

42.9.7 VCI POSEDGE DETECT REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:7	Reserved	RES	-	-
3:0	VCI_IN_POS[3:0] These bits record a low to high transition on the VCI_INx# pins. A “1” indicates a transition occurred. For each VCI_IN_POS[x] bit in the field: 1=Positive Edge Detected 0=No edge detected	R/WC	0	RESET_VBAT

42.9.8 VCI NEGEDGE DETECT REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	VCI_IN_NEG[3:0] These bits record a high to low transition on the VCI_INx# pins. A “1” indicates a transition occurred. For each VCI_IN_NEG[x] bit in the field: 1=Negative Edge Detected 0=No edge detected	R/WC	0	RESET_VBAT

42.9.9 VCI BUFFER ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3:0	VCI_BUFFER_EN[3:0] Input Buffer enable. After changing the buffer enable for a VCI input, firmware should reset the input latch and clear any potential interrupt that may have been triggered by the input, as changing the buffer may cause the internal status to change. This register has no effect when VTR_CORE is powered. When VTR_CORE is on, the input buffers are enabled only by the IE[3:0] bit. For each VCI_BUFFER_EN[x] bit in the field: 1=VCI_INx# input buffer enabled independent of the IE[3:0] bit. The edge detection latches for this input are always enabled 0=VCI_INx# input buffer enabled by the IE[3:0] bit. The edge detection latches are only enabled when the IE[3:0] bit is '1' (default)	RW	0	RESET_VBAT

© 2020 Microchip Technology Inc.

43.0 VBAT-POWERED RAM

43.1 Overview

The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while the main power rail is operational, and will retain its values powered by battery power while the main rail is unpowered.

43.2 References

No references have been cited for this feature.

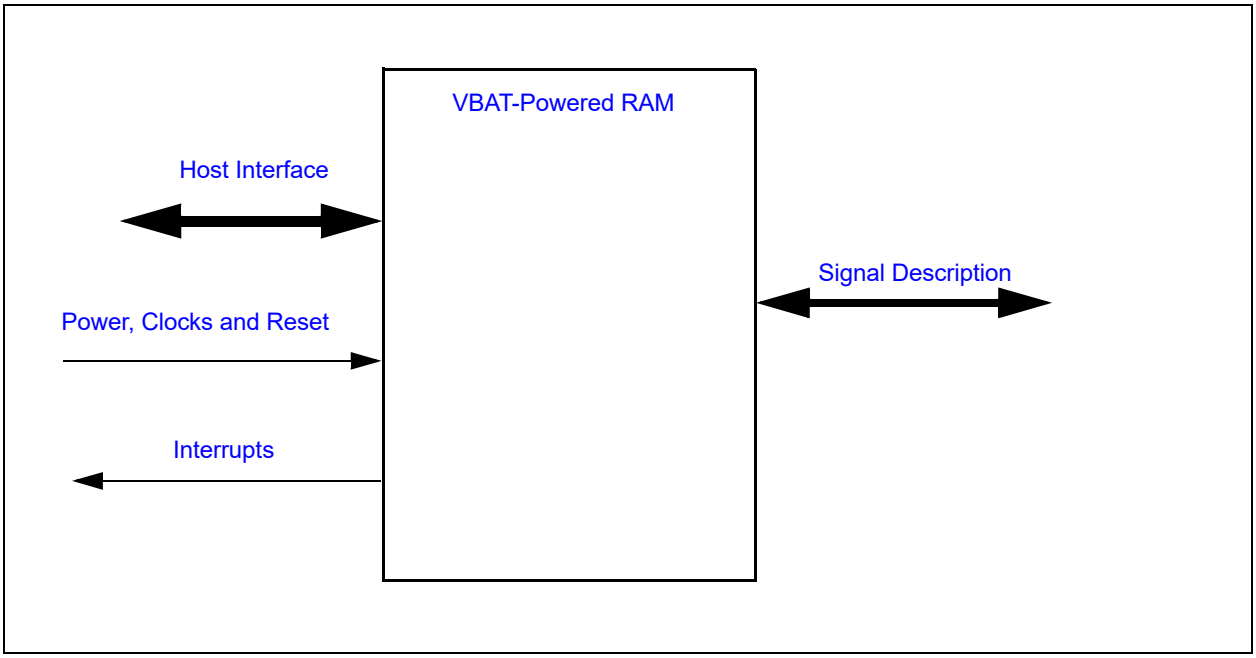
43.3 Terminology

There is no terminology defined for this section.

43.4 Interface

This block is designed to be accessed internally via a registered host interface.

FIGURE 43-1: I/O DIAGRAM OF BLOCK



43.5 Signal Description

There are no external signals for this block.

43.6 Host Interface

The contents of the VBAT RAM are accessible only by the Embedded Controller (EC).

43.7 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

43.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The main power well used when the VBAT RAM is accessed by the EC.
VBAT	The power well used to retain memory state while the main power rail is unpowered.

43.7.2 CLOCK INPUTS

No special clocks are required for this block.

43.7.3 RESETS

Name	Description
RESET_VBAT	This signal resets all the registers and logic in this block to their default state.

43.8 Interrupts

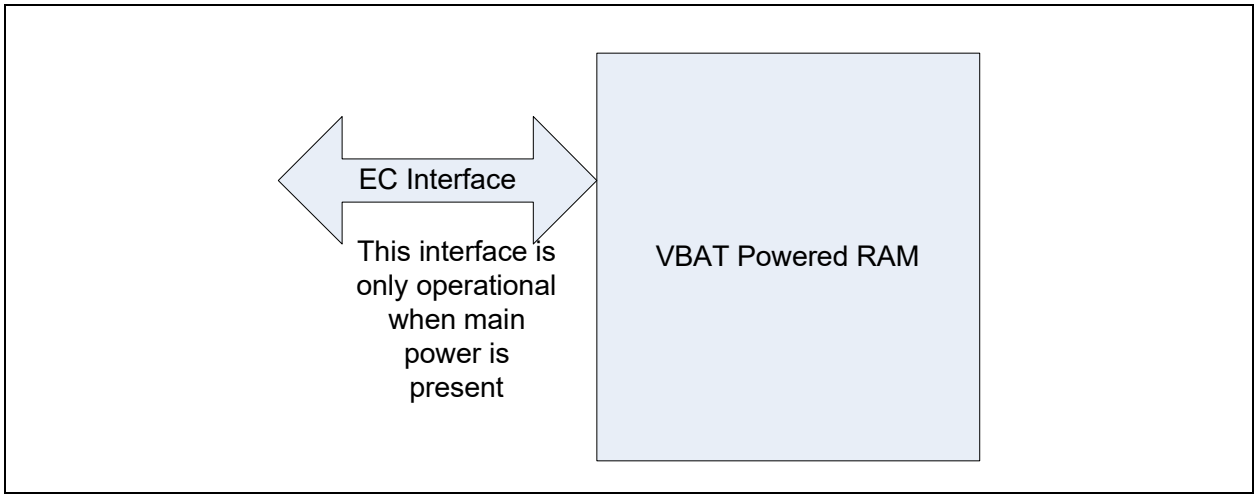
This block does not generate any interrupts.

43.9 Low Power Modes

The VBAT-Powered RAM automatically enters a low power mode whenever it is not being accessed by the EC. There is no chip-level Sleep Enable input.

43.10 Description

FIGURE 43-2: VBAT RAM BLOCK DIAGRAM



The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while VTR_CORE is powered, and will retain its values powered by VBAT while VTR_CORE is unpowered. The RAM is organized as a 16 words x 32-bit wide for a total of 64 bytes.

The contents of the VBAT RAM is indeterminate after a RESET_VBAT.

Note:	Any secret customer information stored on chip in VBAT memory must be encrypted for best security practices

© 2020 Microchip Technology Inc.

44.0 VBAT REGISTER BANK

44.1 Introduction

This chapter defines a bank of registers powered by VBAT.

44.2 Interface

This block is designed to be accessed internally by the EC via the register interface.

44.3 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

44.3.1 POWER DOMAINS

Name	Description
VBAT	The VBAT Register Bank are all implemented on this single power domain.

44.3.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

44.3.3 RESETS

Name	Description
RESET_VBAT	This reset signal, which is an input to this block, resets all the logic and registers to their initial default state.

44.4 Interrupts

This block does not generate any interrupt events.

44.5 Low Power Modes

The VBAT Register Bank is designed to always operate in the lowest power consumption state.

44.6 Description

The VBAT Register Bank block is a block implemented for aggregating miscellaneous battery-backed registers required the host and by the Embedded Controller (EC) Subsystem that are not unique to a block implemented in the EC subsystem.

44.7 EC Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the VBAT Register Bank Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 44-1: REGISTER SUMMARY

Offset	Register Name
00h	Power-Fail and Reset Status Register
04h	TEST
08h	Clock Enable Register
0Ch	SHDN Pin Disable Register
10h	TEST
14h	TEST

TABLE 44-1: REGISTER SUMMARY (CONTINUED)

Offset	Register Name
1Ch	32kHz Trim Control Register
20h	Monotonic Counter Register
24h	Counter HiWord Register
2Ch	TEST

44.7.1 POWER-FAIL AND RESET STATUS REGISTER

The Power-Fail and Reset Status Register collects and retains the VBAT RST and WDT event status when VTR_CORE is unpowered.

Address	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	VBAT_RST The VBAT_RST bit is set to '1' by hardware when a RESET_VBAT is detected. This is the register default value. To clear VBAT RST EC firmware must write a '1' to this bit; writing a '0' to VBAT RST has no affect.	R/WC	1	RESET_VBAT
6	SYSRESETREQ This bit is set to '1b' if a RESET_SYS was triggered by an ARM SYSRESETREQ event. This bit is cleared to '0b' when written with a '1b'; writes of a '0b' have no effect.	R/WC	-	-
5	WDT This bit is set to '1b' if a RESET_SYS was triggered by a Watchdog Timer event. This bit is cleared to '0b' when written with a '1b'; writes of a '0b' have no effect.	R/WC	0	RESET_VBAT
4	RESETI This bit is set to '1b' if a RESET_SYS was triggered by a low signal on the nRESET_IN input pin. This bit is cleared to '0b' when written with a '1b'; writes of a '0b' have no effect.	R/WC	0	RESET_VBAT
3	TEST	R/WC	0	RESET_VBAT
2	SOFT_SYS_RESET Status This bit is set to '1b' if a was triggered by an assertion of the SOFT-_SYS_RESET bit in the System Reset Register. This bit is cleared to '0b' when written with a '1b'; writes of a '0b' have no effect.	R/WC	0	RESET_VBAT
1	Reserved	RES	-	-
0	Reserved	RES	-	-

© 2020 Microchip Technology Inc.

44.7.2 CLOCK ENABLE REGISTER

Address	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
3	XOSEL This bit selects between a single-ended clock source for the crystal oscillator or an external parallel crystal. 1=The crystal oscillator is driven by a single-ended 32KHz clock source connected to the XTAL2 pin 0=The crystal oscillator requires a 32KHz parallel resonant crystal connected between the XTAL1 and XTAL2 pins	R/W	0b	RESET_VBAT
2	32KHZ_SOURCE This field determines the source for the 32KHz Clock internal clock source. If set to '1b', this bit will only take effect if an active clock has been detected on the crystal pins. Once the 32KHz source has been switched, activity detection on the crystal no longer functions. Therefore, if the crystal oscillator uses a single-ended input, once started that input must not stop while this bit is '1b'. 1=Crystal Oscillator. The selection between a singled-ended input or a resonant crystal is determined by XOSEL in this register 0=Silicon Oscillator	R/W	0b	RESET_VBAT
1	EXT_32K This bit selects the source for the 32KHz clock domain. 1=The 32KHZ_IN VTR-powered pin is used as a source for the 32KHz clock domain. If an activity detector does not detect a clock on the selected source, the 32KHz Clock internal clock source is automatically selected 0=The 32KHz Clock source is used as the source for the 32KHz clock domain	R/W	0b	RESET_VBAT
0	Test	R	0b	-

44.7.3 SHDN PIN DISABLE REGISTER

Address	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	SYS_SHDN_DISABLE Used to disable SYS_SHDN#. 1=Disable SYS_SHDN# 0=SYS_SHDN# active	R/W / R	0b	RESET_VBAT

If SYS_SHDN_DISABLE bit is set, then SYS_SHDN# in VCI Firmware Override Register will have no impact on the VCI_OUT Signal.

44.7.4 32KHZ TRIM CONTROL REGISTER

Address	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
2	VBAT_OSC32KHz_Override_Reset Override 32KHz Oscillator IP Port Reset	R/W	0h	RESET_VBAT
1	VBAT_OSC32KHz_Override_Enable Override 32KHz Oscillator IP Port Enable	R/W	0h	RESET_VBAT
0	VBAT_OSC32KHz_Override Enables override mode in 32KHz Oscillator IP.	R/W	0h	RESET_VBAT

44.7.5 MONOTONIC COUNTER REGISTER

Address	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	MONOTTONIC_COUNTER Read-only register that increments by 1 every time it is read. It is reset to 0 on a VBAT Power On Reset.	R	0b	RESET_VBAT

44.7.6 COUNTER HIWORD REGISTER

Address	24h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	COUNTER_HIWORD Thirty-two bit read/write register. If software sets this register to an incrementing value, based on an external non-volatile store, this register may be combined with the Monotonic Counter Register to form a 64-bit monotonic counter.	R/W	0b	RESET_VBAT

© 2020 Microchip Technology Inc.

45.0 EC SUBSYSTEM REGISTERS

45.1 Introduction

This chapter defines a bank of registers associated with the EC Subsystem.

45.2 References

None

45.3 Interface

This block is designed to be accessed internally by the EC via the register interface.

45.4 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

45.4.1 POWER DOMAINS

Name	Description
VTR_CORE	The logic and registers implemented in this block are powered by this power well.

45.4.2 CLOCK INPUTS

This block does not require any special clock inputs. All register accesses are synchronized to the host clock.

45.4.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state, except WDT Event Count Register.
RESET_SYS_nWDT	This signal resets the WDT Event Count Register register. This reset is not asserted on a WDT Event.
RESET_VTR	This reset signal is asserted only on VTR_CORE power on.

45.5 Interrupts

This block does not generate any interrupt events.

45.6 Low Power Modes

The EC Subsystem Registers may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry. When this block is commanded to sleep it will still allow read/write access to the registers.

45.7 Description

The EC Subsystem Registers block is a block implemented for aggregating miscellaneous registers required by the Embedded Controller (EC) Subsystem that are not unique to a block implemented in the EC subsystem.

45.8 EC-Only Registers

Registers for this block are shown in the following summary table. Addresses for each register are determined by adding the offset to the Base Address for the EC Subsystem Registers Block in the Block Overview and Base Address Table in Section 3.0, "Device Inventory".

TABLE 45-1: REGISTER SUMMARY

Offset	Register Name
00h	Reserved
04h	AHB Error Address Register
08h	TEST
0Ch	TEST
10h	TEST
14h	AHB Error Control Register
18h	Interrupt Control Register
1Ch	ETM TRACE Enable Register
20h	Debug Enable Register
28h	WDT Event Count Register
2Ch	TEST
30h	TEST
34h	TEST
38h	Reserved
3Ch	TEST
40h	PECI Disable Register
44h	TEST
48h	TEST
50h	VCI Firmware Override Register
54h	TEST
5Ch	TEST
60h	TEST
64h	GPIO Bank Power Register
68h	TEST
6Ch	TEST
90h	Virtual Wire Source Configuration Register
94h	Comparator Control Register
98h	Comparator Sleep Control Register
100h	TEST

45.8.1 AHB ERROR ADDRESS REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	AHB_ERR_ADDR In priority order: 1. AHB address is registered when an AHB error occurs on the processors AHB master port and the register value was already 0. This way only the first address to generate an exception is captured. 2. The processor can clear this register by writing any 32-bit value to this register.	R/WZC	0h	RESET_SYS

© 2020 Microchip Technology Inc.

45.8.2 AHB ERROR CONTROL REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-
1	TEST	R/W	0h	RESET_SYS
0	AHB_ERROR_DISABLE 1=EC memory exceptions are disabled 0=EC memory exceptions are enabled	R/W	0h	RESET_SYS

45.8.3 INTERRUPT CONTROL REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	NVIC_EN This bit enables Alternate NVIC IRQ's Vectors. The Alternate NVIC Vectors provides each interrupt event with a dedicated (direct) NVIC vector. 1=Alternate NVIC vectors enabled 0=Alternate NVIC vectors disabled	R/W	1b	RESET_SYS

45.8.4 ETM TRACE ENABLE REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	TRACE_EN This bit enables the ARM TRACE debug port (ETM/ITM). The Trace Debug pins are forced to the TRACE functions. 1=ARM TRACE port enabled 0=ARM TRACE port disabled	R/W	0b	RESET_SYS

© 2020 Microchip Technology Inc.

45.8.5 DEBUG ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:5	Reserved	RES	-	-
4	BOUNDARY SCAN PORT ENABLE 1= Enable Boundary scan port enable The Boundary Scan Tap controller is accessible via JTAG Port 0= Disable Boundary scan port enable The Boundary scan Tap controller is not accessible via JTAG Port. If JTAG_STRAP pin is sampled low, this bit cannot enable Boundary scan mode. Please refer to TAP Controller Select Strap Option for usage of this bit.	R/W	0h	RESET_SYS
3	DEBUG_PU_EN If this bit is set to '1b' internal pull-up resistors are automatically enabled on the appropriate debugging port wires whenever the debug port is enabled (the DEBUG_EN bit in this register is '1b' and the JTAG_RST# pin is high). The setting of DEBUG_PIN_CFG determines which pins have pull-ups enabled when the debug port is enabled.	R/W	0h	RESET_SYS
2:1	DEBUG_PIN_CFG This field determines which pins are affected by the JTAG_RST# debug enable pin. 3=Reserved 2=The pins associated with the JTAG TCK and TMS switch to the debug interface when JTAG_RST# is de-asserted high. The pins associated with TDI and TDO remain controlled by the associated GPIO. This setting should be used when the ARM Serial Wire Debug (SWD) is required for debugging and the Serial Wire Viewer is not required 1=The pins associated with the JTAG TCK, TMS and TDO switch to the debug interface when JTAG_RST# is de-asserted high. The pin associated with TDI remains controlled by the associated GPIO. This setting should be used when the ARM Serial Wire Debug (SWD) and Serial Wire Viewer (SWV) are both required for debugging 0=All four pins associated with JTAG (TCK, TMS, TDI and TDO) switch to the debug interface when JTAG_RST# is de-asserted high. This setting should be used when the JTAG TAP controller is required for debugging	R/W	0h	RESET_SYS
0	DEBUG_EN This bit enables the JTAG/SWD debug port. 1=JTAG/SWD port enabled. A high on JTAG_RST# enables JTAG or SWD, as determined by SWD_EN 0=JTAG/SWD port disabled. JTAG/SWD cannot be enabled (the JTAG_RST# pin is ignored and the JTAG signals remain in their non-JTAG state) If JTAG_RST# pin is sampled low, this bit cannot enable JTAG mode.	R/W	0b	RESET_SYS

© 2020 Microchip Technology Inc.

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
Note:	Boot ROM updates this register value on exit. Refer to the Boot ROM document for details.			

45.8.6 WDT EVENT COUNT REGISTER

Offset	28h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:4	Reserved	RES	-	-
3:0	WDT_EVENT_COUNT This field is cleared to 0 on a reset triggered by the main power on reset, but not on a reset triggered by the Watchdog Timer. This field needs to be written by application to indicate the number of times a WDT fired before loading a good EC code image. Note 1	R/W	0b	RESET_SYS_n-WDT
Note 1:	The recommended procedure is to first clear the WDT Status Register followed by incrementing the WDT_EVENT_COUNT.			

45.8.7 PECI DISABLE REGISTER

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	-
0	PECI_DISABLE This bit reduces leakage current through the CPU voltage reference pin if PECI or SB-TSI are not used. 1=The VREF_VTT function is disabled, independent of the mux setting of the GPIO that shares the pin. The GPIO that shares the pin is not disabled 0=The VREF_VTT pin is enabled	R/W	0b	RESET_SYS

45.8.8 VCI FIRMWARE OVERRIDE REGISTER

Offset	50h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:1	Reserved	RES	-	–
0	SYS_SHDN# If a thermal event occurs, the firmware asserts the SYS_SHDN# bit which can de-assert the VCI_OUT pin that is associated with the VBAT-Powered Control Interface. Please refer to VCI_OUT Block Diagram for VCI_OUT logic.	R/W	1	RESET_SYS

45.8.9 GPIO BANK POWER REGISTER

Offset	64h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	Reserved	RES	-	–
7	GPIO Bank Power Lock 0 = VTR_LEVEL bits[2:0] and GPIO Bank Power Lock bit are R/W 1 = VTR_LEVEL bits[2:0] and GPIO Bank Power Lock bit are Read Only This bit cannot be cleared once it is set to ‘1’. Writing zero has no effect.	Bit[7]=0 R/W Bit[7]=1 RO	0h	RESET_SYS
6:3	Reserved	RES	-	–
2	VTR_LEVEL3 Voltage level on VTR3 power rail.This bit is set by software 1=VTR3 is powered by 1.8V 0=VTR3 is powered by 3.3V	see Bit[7]	0h	RESET_SYS
1	VTR_LEVEL2 Voltage level on VTR2 power rail. Software will set this bit based on the VTR2_STAP pin. 1=VTR2 is powered by 1.8V 0=VTR2 is powered by 3.3V	see Bit[7]	0h	RESET_SYS
0	TEST This is a TEST bit and should be always programmed to 0b for proper functioning of the device.	RW	0h	RESET_SYS

Note: The Boot ROM reads the VTR_LEVEL2VTR_LEVEL3 values from the SPI Flash Header and writes the VTR_LEVEL2,VTR_LEVEL3bits. If the SPI Flash load fails, the Boot ROM clears all VTR_LEVEL2,VTR_LEVEL3 bits.

© 2020 Microchip Technology Inc.

45.8.10 VIRTUAL WIRE SOURCE CONFIGURATION REGISTER

Offset	90h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	-	-
1:0	VWIRE_SOURCE VWIRE_SOURCE [2] should always be programmed to 1b. VWIRE_SOURCE [1] 0 = The hardware source MBX_Host_SMI affects the state of the SMI# (SRC1) bit of the SMVW02 register. 1 = The hardware source MBX_Host_SMI does not affect the SMI# (SRC1) bit of the SMVW02 register. **Note:** Firmware can always write to the SRC1 bit of the SMVW02 register. VWIRE_SOURCE [0] 0=The hardware source EC_SCI# affects the state of the SCI# (SRC0) bit of the SMVW02 register. 1= The hardware source EC_SCI# does not affect the SCI# (SRC0) bit of the SMVW02 register. **Note:** Firmware can always write to the SRC0 bit of the SMVW02 register.	RW	7h	RESET_SYS

45.8.11 COMPARATOR CONTROL REGISTER

Offset	94h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:5	Reserved	RES	-	-
4	Comparator 1 Enable 1= Enable Comparator 1 operation 0= Disable Comparator 1 operation	RW	0h	RESET_SYS
3	Reserved	RW	0h	RESET_SYS
2	Comparator 0 Configuration Locked 1= Configuration locked.Bits[2:0] are read only 0= Configuration not locked.Bits[2:0] are read write	R/W1X Note 2	CMP_STRAP0 pin = 1 then default= 1 All other configura-tions default= 0	RESET_SYS
1	Reserved	RES	0h	RESET_SYS

Offset	94h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	Comparator 0 Enable 1= Enable Comparator 0 operation 0= Disable Comparator 0 operation	RW or RO Note 1	CMP_STRAP0 pin = 1 then default= 1 All other configura-tions default= 0	RESET_SYS
Note 1: These bits become read only by writing bit 2 Comparator 0 Configuration Locked bit. **2:** If CMP_STRAP0 pin = 1, then Boot ROM writes this bit. Once this bit is written, this bit becomes read only.				

45.8.12 COMPARATOR SLEEP CONTROL REGISTER

Offset	98h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:2	Reserved	RES	-	-
1	Comparator 1 Deep Sleep Enable 0 = Comparator Deep Sleep Disable 1 = Comparator Deep Sleep Enable	R/W	0h	RESET_SYS
0	Comparator 0 Deep Sleep Enable 0 = Comparator Deep Sleep Disable 1 = Comparator Deep Sleep Enable	R/W or RO Note 1	0h	RESET_SYS
Note: Comparator Deep Sleep Enable must be set when the Comparator is enabled				

© 2020 Microchip Technology Inc.

46.0 SECURITY FEATURES

46.1 Overview

This device includes a set of components that can support a high level of system security. Hardware support is provided for:

- Authentication, using public key algorithms
- Integrity, using Secure Hash Algorithms (SHA)
- Privacy, using symmetric encryption (Advanced Encryption Standard, AES)
- Entropy, using a true Random Number Generator

46.2 References

- American National Standards Institute, “Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography”, X9.63-2011, December 2011
- American National Standards Institute, “Public Key Cryptography for the Financial Servic3es Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)”, X9.62-2005, November 2005
- International Standards Organization, “Information Technology - Security techniques - Cryptographic techniques based on elliptic curves -- Part 2: Digital Signatures”, ISO/IEC 15946-2, December 2002
- National Institute of Standards and Technology, “Secure Hash Standard (SHS)”, FIPS Pub 180-4, March 2012
- National Institute of Standards and Technology, “Digital Signature Standard (DSS)”, FIPS Pub 186-3, June 2009
- National Institute of Standards and Technology, “Advanced Encryption Standard (AES)”, FIPS Pub 197, November 2001
- National Institute of Standards and Technology, “Recommendation for Block Cipher Modes of Operation”, FIPS SP 800-38A, 2001
- RSA Laboratories, “PKCS#1 v2.2: RSA Cryptography Standard”, October 2012

46.3 Terminology

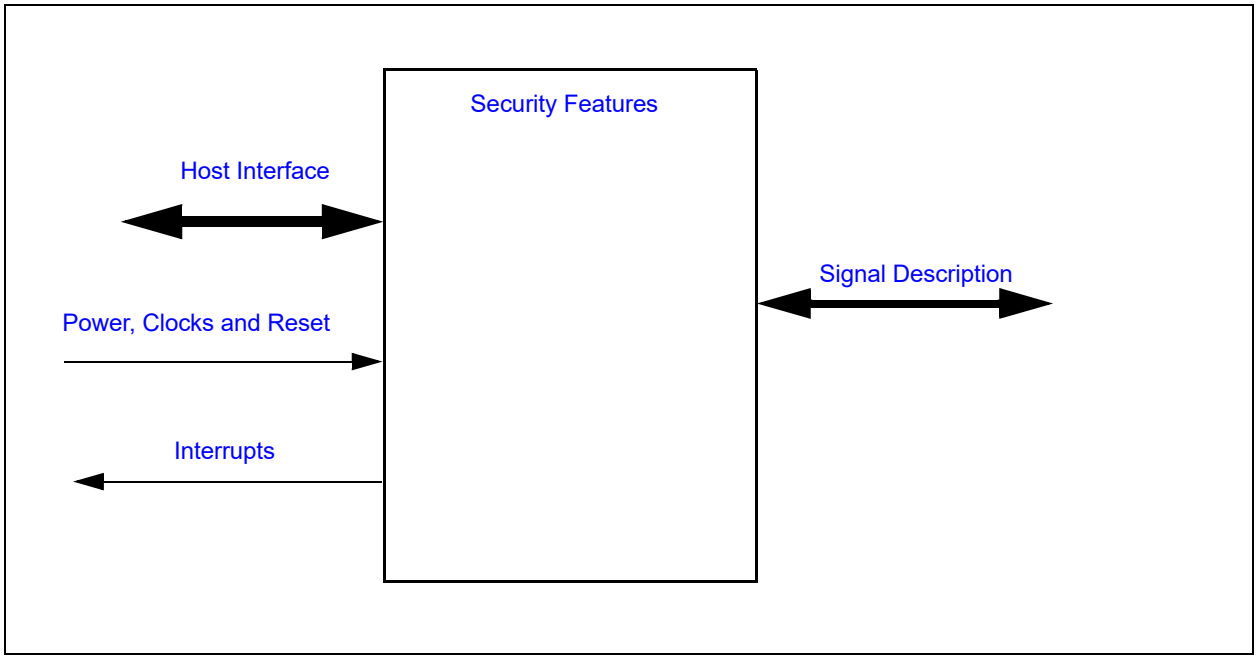
There is no terminology defined for this section.

46.4 Interface

This block is designed to be accessed internally via a registered host interface.

© 2020 Microchip Technology Inc.

FIGURE 46-1: I/O DIAGRAM OF BLOCK



46.5 Signal Description

There are no external signals for this block.

46.6 Host Interface

Registers for the cryptographic hardware are accessible by the EC.

46.7 Power, Clocks and Reset

46.7.1 POWER DOMAINS

Name	Description
VTR_CORE	The main power well used when the VBAT RAM is accessed by the EC.

46.7.2 CLOCK INPUTS

No special clocks are required for this block.

46.7.3 RESETS

Name	Description
RESET_SYS	This signal resets all the registers and logic in this block to their default state.

© 2020 Microchip Technology Inc.

46.8 Interrupts

This section defines the Interrupt Sources generated from this block.

Source	Description
Public Key Engine	
PKE_ERROR	Public Key Engine core error detected
PKE_END	Public Key Engine completed processing
Symmetric Encryption	
AES	Symmetric Encryption block completed processing
Cryptographic Hashing	
HASH	HASH
Random Number Generator	
RNG	Random Number Generator filled its FIFO

46.9 Low Power Modes

The Security Features may be put into a low power state by the chip's Power, Clocks, and Reset (PCR) circuitry.

46.10 Description

The security hardware incorporates the following functions:

46.10.1 SYMMETRIC ENCRYPTION/DECRYPTION

Standard AES encryption and decryption, with key sizes of 128 bits, 192 bits and 256 bits, are supported with a hardware accelerator. AES modes that can be configured include Electronic Code Block (ECB), Cipher Block Chaining (CBC), Counter Mode (CTR), Output Feedback (OFB) and Cipher Feedback (CFB).

46.10.2 CRYPTOGRAPHIC HASHING

Standard SHA hash algorithms, including SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512, are supported by hardware.

46.10.3 PUBLIC KEY CRYPTOGRAPHIC ENGINE

A large variety of public key algorithms are supported directly in hardware. These include:

- RSA encryption and decryption, with key sizes of 1024 bits, 2048 bits, 3072 bits and 4096 bits
- Elliptic Curve point multiply, with all standard NIST curves, using either binary fields or prime fields
- Elliptic Curve point multiply with Curve25519
- The Elliptic Curve Digital Signature Algorithm (ECDSA), using all supported NIST curves
- The Elliptic Curve Korean Certificate-based Digital Signature Algorithm (EC-KCDSA), using all supported NIST curves
- The Edwards-curve Digital Signature Algorithm (EdDSA), using Curve25519
- Miller-Rabin primality testing

The Public Key Engine includes a 24KB cryptographic SRAM, which can be accessed by the EC when the engine is not in operation. With its private SRAM memory, the Public Key Engine can process public key operations independently of the EC.

46.10.4 TRUE RANDOM NUMBER GENERATOR

A true Random Number Generator, which includes a 1K bit FIFO for pre-calculation of random bits.

© 2020 Microchip Technology Inc.

46.10.5 MONOTONIC COUNTER

The Monotonic Counter is defined in Section 44.7.5, "Monotonic Counter Register". The counter automatically increments every time it is accessed, as long as VBAT power is maintained. If it is necessary to maintain a monotonic counter across VBAT power cycles, the Counter HiWord Register can be combined with the Monotonic Counter Register to form a 64-bit monotonic counter. Firmware would be responsible for updating the Counter HiWord on a VBAT POR. The HiWord could be maintained in a non-volatile source, such as the EEPROM or an external SPI Flash.

46.10.6 CRYPTOGRAPHIC API

The Boot ROM includes an API for direct software access to cryptographic functions. API functions for Hashing and AES include a DMA interface, so the operations can function on large blocks of SRAM with a single call.

46.11 Registers

TABLE 46-1: CRYPTOGRAPHIC SRAM

Block Instance	Start Address	End Address	Size
Cryptographic SRAM	4010_0000h	4010_5FFF	24KB

© 2020 Microchip Technology Inc.

47.0 OTP BLOCK

47.1 Introduction

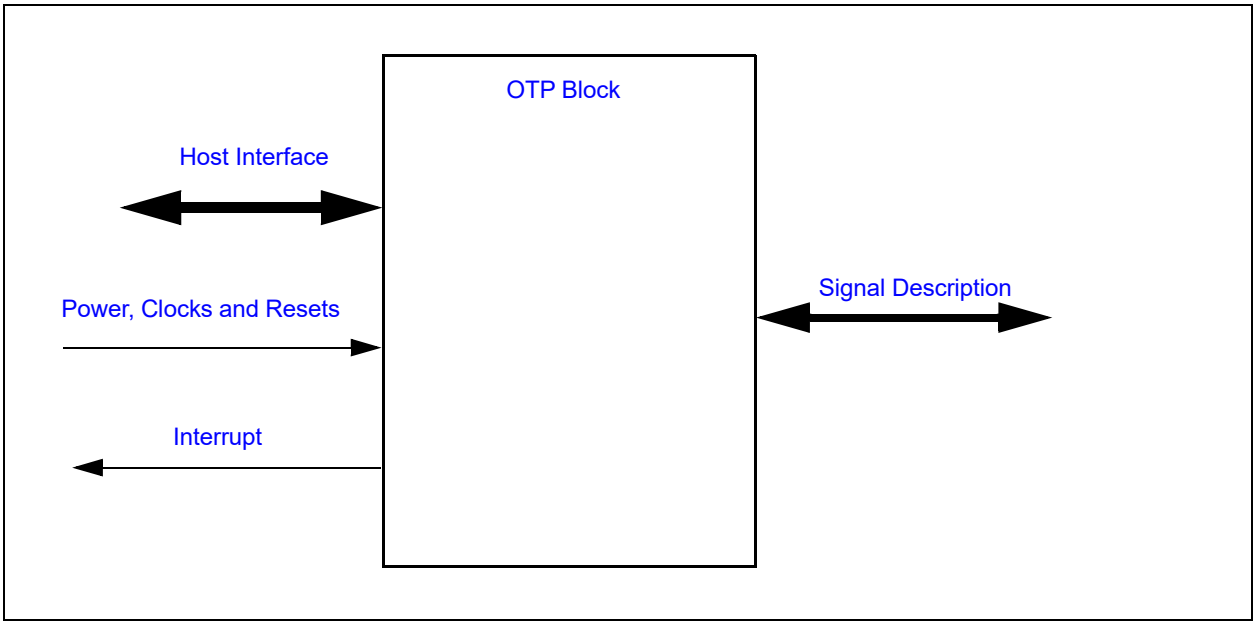
The OTP Block provides a means of programming and accessing a block of One Time Programmable memory.

47.2 Terminology

None.

47.3 Interface

FIGURE 47-1: OTP BLOCK INTERFACE DIAGRAM



47.4 Signal Description

There are no external signals from this block

47.5 Host Interface

The registers defined for the OTP Block are accessible by the EC.

47.6 Interrupt Interface

TABLE 47-1: INTERRUPT SIGNALS

Source	Description
OTP_READY	The OTP_READY interrupt will be generated whenever an OTP command is completed.

47.7 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

47.7.1 POWER DOMAINS

TABLE 47-2: POWER SOURCES

Name	Description
VTR_CORE	This power well sources all of the registers and logic in this block, except where noted.
VTR	This is the IO voltage for the block.

47.7.2 CLOCKS

This section describes all the clocks in the block, including those that are derived from the I/O Interface as well as the ones that are derived or generated internally.

TABLE 47-3: CLOCKS

Name	Description
48MHz	This clock signal drives selected logic (e.g., counters).

47.7.3 RESETS

TABLE 47-4: RESET SIGNALS

Name	Description
RESET_SYS	This reset signal resets all of the registers and logic in this block.

47.8 Low Power Modes

The OTP always comes up in low power mode and stays in that state unless the firmware needs to use it.

47.9 Description

The OTP Block has a capacity of 8 K bits arranged as 1K x 8 bits.

Note:	Any secret customer information stored on chip in OTP memory must be encrypted for best security practices.

47.10 OTP Memory Map

Please refer to Boot ROM document for this information.

TABLE 47-5: REGISTER SUMMARY

Offset	Register Name
44h	OTP Write Lock Register
48h	OTP Read Lock Register

© 2020 Microchip Technology Inc.

47.10.1 OTP WRITE LOCK REGISTER

Offset	44h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	OTP_WRLOCK When any of these bits are set, the corresponding 32 byte range in the OTP is not writable.	R/W1S	0h	RESET_SYS

47.10.2 OTP READ LOCK REGISTER

Offset	48h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:0	OTP_RDLOCK When any of these bits are set, the corresponding 32 byte range in the OTP is not readable.	R/W1S	0h	RESET_SYS

Note 1: OTP Memory can be locked by writing to OTP bytes 1012 - 1019. Boot ROM will then lock the region on every Boot preventing the code that is loaded from accessing this memory location.

2: Application FW can write to the above lock registers and lock the memory region preventing other code loaded from accessing the locked region. This is useful in multistage boot loaders

48.0 TEST MECHANISMS

48.1 ARM Test Functions

Test mechanisms for the ARM are described in Section 5.0, "ARM M4 Based Embedded Controller".

48.2 JTAG Boundary Scan

Note:	Boundary Scan operates in 4-wire JTAG mode only. This is not supported by 2-wire SWD.

JTAG Boundary Scan includes registers and functionality as defined in IEEE 1149.1 and the MEC152x BSDL file. Functionality implemented beyond the standard definition is summarized in Table 48-2. The MEC152x Boundary Scan JTAG ID is shown in Table 1-1.

Note:	Must wait a minimum of 35ms after a POR to accurately read the Boundary Scan JTAG ID. Reading the JTAG ID too soon may return a Boundary Scan JTAG ID of 00000000h. This is not a valid ID value.

48.2.1 TAP CONTROLLER SELECT STRAP OPTION

The TAP Controller Select Strap Option determines the JTAG slave that is selected when JTAG_RST# is not asserted. The state of the TAP Controller Select Strap Option pin, defined in the Pin Configuration chapter (JTAG_STRAP), is sampled by hardware at POR according to the Slave Select Timing as defined in Section 50.20, "JTAG Interface Timing" The same JTAG port is used for accessing boundary scan and ARM TAP Controller based on the JTAG_STRAP pin.

illustrates the selection of the Tap controller over the JTAG port

By default the ARM Tap controller defaults to 3-Pin mode and may be configured to 4- pin or 2-pin mode.

TABLE 48-1: TAP CONTROLLER SELECTION

JTAG_STRAP Pin Value *at Power On*	**JTAG_RST#**	BOUNDARY SCAN PORT ENABLE **Bit**	DEBUG_EN **(JTAG Debug Disable) Bit**	**Description**
0	0	X	x	Boundary scan/ ARM Tap controller cannot be accessed through JTAG port.
0	1	X	0	Hardware does not enable Boundary scan functionality. BOUNDARY SCAN PORT ENABLE bit has no effect. ARM TAP controller cannot be accessed.
0	1	X	1	Hardware does not enable Boundary scan functionality. BOUNDARY SCAN PORT ENABLE bit has no effect. ARM TAP controller can be accessed.
1	0	x	x	Hardware does not enable Boundary Scan functionality Boundary Scan TAP controller or ARM TAP controller is not accessible via JTAG port.
1	1	0	x	Hardware enables Boundary Scan functionality, but Boundary Scan Port is disabled. Boundary Scan TAP controller or ARM TAP controller is not accessible via JTAG port.

© 2020 Microchip Technology Inc.

TABLE 48-1: TAP CONTROLLER SELECTION (CONTINUED)

JTAG_STRAP Pin Value *at Power On*	JTAG_RST#	BOUNDARY SCAN PORT ENABLE Bit	DEBUG_EN (JTAG Debug Disable) Bit	Description
1	1	1	x	Hardware enables Boundary Scan functionality and Boundary Scan Port is enabled. Boundary Scan TAP controller is accessible via JTAG port ARM TAP controller is not accessible via JTAG port.

TABLE 48-2: EXTENDED BOUNDARY SCAN FUNCTIONALITY

Bits	Function	Description
12, 14	TAP Controller Select Strap Option Override	When the Strap Option Override is '1,' the strap option is overridden to select the debug TAP Controller until the next time the strap is sampled. To set Strap Override Function, write 0X1FFFFD to the TAP controller instruction register, then write 0x5000 to the TAP controller data register. Note that the instruction register is 18 bits long; the data register is 16 bits long.

49.0 ELECTRICAL SPECIFICATIONS

49.1 Maximum Ratings*

*Stresses exceeding those listed could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

Note:	When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

49.1.1 ABSOLUTE MAXIMUM THERMAL RATINGS

Parameter	Maximum Limits
Operating Temperature Range	-40^{o}C to +85^{o}C Industrial
Storage Temperature Range	-55^{o} to +150^{o}C
Lead Temperature Range	Refer to JEDEC Spec J-STD-020B

49.1.2 ABSOLUTE MAXIMUM SUPPLY VOLTAGE RATINGS

Symbol	Parameter	Maximum Limits
VBAT	3.0V Battery Backup Power Supply with respect to ground	-0.3V to +3.63V
VTR_REG	Main Regulator Power Supply with respect to ground	-0.3V to +3.63V
VTR_ANALOG	3.3V Analog Power Supply with respect to ground	-0.3V to +3.63V
VTR1	3.3V Power Supply with respect to ground	-0.3V to +3.63V
VTR2	3.3V or 1.8V Power Supply with respect to ground	-0.3V to +3.63V
VTR3	3.3V or 1.8V Power Supply with respect to ground	-0.3V to +3.63V
VCC	3.3V Main Power Supply with respect to ground (Connected to VCC_PWRGD pin)	-0.3V to +3.63V

49.1.3 ABSOLUTE MAXIMUM I/O VOLTAGE RATINGS

Parameter	Maximum Limits
Voltage on any Digital Pin with respect to ground	Determined by Power Supply of I/O Buffer and Pad Type

© 2020 Microchip Technology Inc.

49.2 Operational Specifications

49.2.1 POWER SUPPLY OPERATIONAL CHARACTERISTICS

TABLE 49-1: POWER SUPPLY OPERATING CONDITIONS

Symbol	Parameter	MIN	TYP	MAX	Units
VBAT	Battery Backup Power Supply	2.0	3.0	3.465	V
VTR_REG	Main Regulator Power Supply	1.71	3.0	3.465	V
VTR_ANALOG	Analog Power Supply	3.135	3.3	3.465	V
VTRx	3.3V Power Supply	3.135	3.3	3.465	V
	1.8V Power Supply	1.71	1.80	1.89	V

Note: The specification for the VTRx supplies are +/- 5%.

49.2.2 AC ELECTRICAL SPECIFICATIONS

The AC Electrical Specifications for the clock input time are defined in Section 50.4, "Clocking AC Timing Characteristics". The clock rise and fall times use the standard input thresholds of 0.8V and 2.0V unless otherwise specified and the capacitive values listed in this section.

49.2.3 CAPACITIVE LOADING SPECIFICATIONS

The following table defines the maximum capacitive load validated for the buffer characteristics listed in Table 49-3, "DC Electrical Characteristics" and the AC characteristics defined in Section 50.4, "Clocking AC Timing Characteristics".

CAPACITANCE T_A = 25°C; fc = 1MHz; V_{cc} = 3.3 VDC

Note: All output pins, except pin under test, tied to AC ground.

TABLE 49-2: MAXIMUM CAPACITIVE LOADING

Parameter	Symbol	Limits			Unit	Notes
		MIN	TYP	MAX		
Input Capacitance of PECI_IO	C_{IN}			10	pF	
Output Load Capacitance supported by PECI_IO	C_{OUT}			10	pF	
Input Capacitance (all other input pins)	C_{IN}			10	pF	Note 1
Output Capacitance (all other output pins)	C_{OUT}			20	pF	Note 2

Note 1: All input buffers can be characterized by this capacitance unless otherwise specified.

2: All output buffers can be characterized by this capacitance unless otherwise specified.

© 2020 Microchip Technology Inc.

49.2.4 DC ELECTRICAL CHARACTERISTICS FOR I/O BUFFERS

TABLE 49-3: DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PIO Type Buffer						
All PIO Buffers Pull-up Resistor @3.3V @1.8V	R_{PU}	 34 35	 52 60	 95 105	 KΩ	Internal PU selected via the GPIO Pin Control Register.
All PIO Buffers Pull-down Resistor @3.3V @1.8V	R_{PD}	 38 36	 63 63	 127 118	 KΩ	Internal PD selected via the GPIO Pin Control Register.
PIO DRIVE_STRENGTH = 00b DRIVE_STRENGTH = 01b DRIVE_STRENGTH = 10b DRIVE_STRENGTH = 11b	 _ _ _ _				 _ _ _ _	The drive strength is determined by programming bits[5:4] of the Pin Control Register 2 Same characteristics as an IO-2 mA. Same characteristics as an IO-4 mA. Same characteristics as an IO-8 mA. Same characteristics as an IO-12 mA.
I Type Input Buffer Low Input Level High Input Level Schmitt Trigger Hysteresis	 V_{ILI} V_{IHI} V_{HYS}		 0.7x VTR 400	 0.3x VTR	 V V mV	TTL Compatible Schmitt Trigger Input
O-2 mA Type Buffer Low Output Level High Output Level	 V_{OL} V_{OH}	 VTR-0.4		 0.4	 V V	 I_{OL} = 2 mA (min) I_{OH} = -2 mA (min)
IO-2 mA Type Buffer	_				_	Same characteristics as an I and an O-2mA.
OD-2 mA Type Buffer Low Output Level	 V_{OL}			 0.4	 V	 I_{OL} = 2 mA (min)

© 2020 Microchip Technology Inc.

TABLE 49-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
IOD-2 mA Type Buffer	_				_	Same characteristics as an I and an OD-2mA.
O-4 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 4 mA (min)
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -4 mA (min)
IO-4 mA Type Buffer	_				_	Same characteristics as an I and an O-4mA.
OD-4 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 4 mA (min)
IOD-4 mA Type Buffer	_				_	Same characteristics as an I and an OD-4mA.
O-8 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA (min)
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -8 mA (min)
IO-8 mA Type Buffer	_				_	Same characteristics as an I and an O-8mA.
OD-8 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA (min)
IOD-8 mA Type Buffer	_				_	Same characteristics as an I and an OD-8mA.
O-12 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA (min)
High Output Level	V_{OH}	VTR-0.4			V	I_{OH} = -12mA (min)
IO-12 mA Type Buffer	_				_	Same characteristics as an I and an O-12mA.
OD-12 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA (min)
IOD-12 mA Type Buffer	_				_	Same characteristics as an I and an OD-12mA.

TABLE 49-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
I_AN Type Buffer						
I_AN Type Buffer (Analog Input Buffer)	I_AN	Voltage range on pins: -0.3V to +3.63V			V	These buffers are not 5V tolerant buffers and they are not back-drive protected
PECI Type Buffer						
VREF_VTT						Connects to CPU Voltage pin (Processor dependent)
PECI Bus Voltage	V_{BUS}	0.95		1.26	V	
SBTSI Bus Voltage	V_{BUS}	1.28		1.9	V	
Input current	IDC			100	µA	
Input Low Current	ILEAK	-10		+10	µA	This buffer is not 5V tolerant This buffer is not backdrive protected.
PECI_IO						All input and output voltages are a function of Vref, which is connected to CPU_VREF input.
Input voltage range	VIn	-0.3		+Vref 0.3	V	See PECI Specification.
Hysteresis	VHYS	0.1 ×Vref	0.2×V ref		V	
Low Input Level	VIL			0.275 ×Vref	V	
High Input Level	VIH	0.725 ×Vref			V	
Low Output Level	VOL			0.25× Vref	V	0.5mA < IOL < 1mA
High Output Level	VOH	0.75 ×Vref			V	IOH = -6mA
Tolerance				3.63	V	This buffer is not 5V tolerant This buffer is not backdrive protected.

© 2020 Microchip Technology Inc.

TABLE 49-3: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
Crystal Oscillator						
XTAL1 (OCLK)	The MEC152x crystal oscillator design requires a 32.768 KHz parallel resonant crystal with load caps in the range 4-18pF. Refer to “Application Note PCB Layout Guide for MEC152x” for more information.					
XTAL2 (ICLK)						
Low Input Level	V_{ILI}			0.4	V	
High Input Level	V_{ILH}	2.0			V	VIN = 0 to VTR
ADC Reference Pins						
ADC_VREF						
Voltage (Option A)	V		VTR		V	Connect to same power supply as VTR
Voltage (Option B)	V	2.97	3.0	3.03	V	
Input Impedance	R_{REF}		75		KΩ	
Input Low Current	ILEAK	-0.05		+0.05	µA	This buffer is not 5V tolerant This buffer is not backdrive protected.

Note: Tolerance for the pins are not 5VT Unless the pin chapter explicitly indicates specific pin has “Over-voltage protection” feature.

49.2.4.1 Pin Leakage

Leakage characteristics for all digital I/O pins is shown in the following Pin Leakage table, unless otherwise specified. Two exceptions are pins with Over-voltage protection and Backdrive protection. Leakage characteristics for Over-Voltage protected pins and Backdrive protected pins are shown in the two sub-sections following the Pin Leakage table.

TABLE 49-4: PIN LEAKAGE (VTR=3.3V + 5%; VTR = 1.8V +5%)

(T_A = -40°C to +85°C)						
Leakage Current	I_{IL}			+/-2	µA	VIN=0V to VTR

OVER-VOLTAGE PROTECTION TOLERANCE

Note: 5V tolerant pins have both backdrive protection and over-voltage protection.

All the I/O buffers that do not have “Over-voltage Protection” are can only tolerate up to +/-10% I/O operation (or +1.98V when powered by 1.8V, or 3.63V when powered by 3.3V).

Functional pins that have “Over-voltage Protection” can tolerate up to 3.63V when powered by 1.8V, or 5.5V when powered by 3.3V. These pins are also backdrive protected. Backdrive Protection characteristics are shown in the following table:

TABLE 49-5: 5V TOLERANT LEAKAGE CURRENTS (VTR=3.3-5%)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
(T_A = -40°C to +85°C)						
Three-State Input Leakage Current for 5V Tolerant Pins	I_{IL}	-1		2	µA	

Note: These measurements are done without an external pull-up.

TABLE 49-6: 3.6V TOLERANT LEAKAGE CURRENTS (VTR = 1.8V-5%)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
(T_A = -40°C to +85°C)						
Three-State Input Leakage Current for Under-Voltage Tolerant Pins	I_{IL}	-2		9	µA	

Note: This measurements are done without an external pull-up.

BACKDRIVE PROTECTION

TABLE 49-7: BACKDRIVE PROTECTION LEAKAGE CURRENTS (VTR=0V)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
(T_A = -40°C to +85°C)						
Input Leakage	I_{IL}	-1		2	µA	VIN@3.47V
Input Leakage	I_{IL}	-2		9	µA	VIN@5.0V

49.2.5 ADC ELECTRICAL CHARACTERISTICS

TABLE 49-8: ADC CHARACTERISTICS

Symbol	Parameter	MIN	TYP	MAX	Units	Comments
VTR_ ANALOG	Analog Supply Voltage (powered by VTR)	3.135	3.3	3.465	V	
V_{RNG}	Input Voltage Range	0		VREF _ADC	V	Range of VREF_ADC input to ADC ground

© 2020 Microchip Technology Inc.

TABLE 49-8: ADC CHARACTERISTICS (CONTINUED)

Symbol	Parameter	MIN	TYP	MAX	Units	Comments
RES	Resolution	–	–	10/12	Bits	Guaranteed Monotonic
ACC	Absolute Accuracy	–	2	4	LSB	
DNL	Differential Non Linearity, DNL	-1	–	+1	LSB	Guaranteed Monotonic
INL	Integral Non Linearity, INL	-3.0	–	+3	LSB	Guaranteed Monotonic
EGAIN	Gain Error, EGAIN	-2	–	2	LSB	
EOFFSET	Offset Error, EOFFSET	-2	–	2	LSB	
CONV	Conversion Time		1.125		μS/channel	
II	Input Impedance	4	4.5	5.3	MΩ	

49.2.6 COMPARATOR ELECTRICAL CHARACTERISTICS

TABLE 49-9: AC AND DC CHARACTERISTICS: COMPARATOR

CHARACTERISTICS		Standard Operating Conditions (unless otherwise noted)				
Symbol	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
V_{IN}	Input Voltage Range	0	-	VTR	V	
V_{HYST}	Input Hysteresis Voltage	15	30	45	mV	
CMRR	Common mode rejection ratio	44			dB	
T_{RESP}	Large signal response time	-	100	160	ns	VCM = VDD/2 100 mV step
TS_{RESP}	Small signal response time	-	160	320	ns	VCM = VDD/2 100 mV step
T_{ON}	Comparator Enable to Valid Output	-	-	0.1	μS	Note 49-1

Note 49-1 To prevent getting glitches on the comparator output, it is recommended to enable the comparator and wait for the output to be valid and stable before configuring the pin for the CMP_VOUTx function.

49.2.7 THERMAL CHARACTERISTICS

TABLE 49-10: THERMAL OPERATING CONDITIONS

Rating	Symbol	MIN	TYP	MAX	Unit
Consumer Temperature Devices					
Operating Junction Temperature Range	T_J	0	—	125	°C
Operating Ambient Temperature Range - Industrial	T_A	-40	—	+85	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - S\ I_{OH})$ I/O Pin Power Dissipation: $I/O = S\ (\{V_{DD} - V_{OH}\} \times I_{OH}) + S\ (V_{OL} \times I_{OL})$	P_D	69.3 (P_{INT} + $P_{I/O}$)			mW
Maximum Allowed Power Dissipation	P_{DMAX}	(T_J[a] – T_A)/θ_{JA}			W

a. Tj Max value is at ambient of 70°C

49.3 Power Consumption

TABLE 49-12: VTR SUPPLY CURRENT, I_VTR

VCC	VTR	48 MHz Clock	VTR1			VTR2			VTR3			Units	Comments
			Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)		
On	On	48MHz	0.15	0.16	0.17	0.01	0.02	0.02	0.01	0.012	0.012	mA	Full On
On	On	PLL Gated	0.15	0.16	0.17	0.01	0.02	0.02	0.01	0.012	0.012	mA	Light Sleep
On	On	PLL Off	0.04	0.05	0.06	0.01	0.02	0.02	0.01	0.012	0.012	mA	Heavy Sleep

TABLE 49-13: VTR SUPPLY CURRENT, I_VTR

VCC	VTR	48 MHz Clock	VTR_REG			VTR_PLL			VTR_ANALOG			Units	Comments
			Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)		
On	On	48MHz	7.5	11.5	13.5	0.10	0.11	0.11	1.2	1.5	1.5	mA	Full On
On	On	PLL Gated	2.0	5.5	7.0	0.10	0.11	0.11	0.15	0.15	0.17	mA	Light Sleep
On	On	PLL Off	0.2	3.5	5.25	0.01	0.01	0.01	0.15	0.15	0.17	mA	Heavy Sleep

Note 1: Full On is defined as follows: The processor is not sleeping, the PLL is powered and the following blocks are Active: ADC, EC Subsystem, Hibernation Timer, Interrupt Controller, PWM, TFDP, Basic Timers, JTAG, RTC. The following blocks are Idle: PECI, eSPI.

2: The sleep states are defined in the System Sleep Control Register in the Power, Clocks and Resets Chapter

3: In order to achieve the lowest leakage current when both PECI and SB TSI are not used, set the VREF_VTT Disable bit to 1.

4: In order to achieve the lowest leakage current when the VREF_VTT power domain is not required, ground the VREF_VTT pin.

5: All values are taken with no eSPI traffic and ADC disabled.

6: VCC on/off is determined by VCC_PWRGD pin

© 2020 Microchip Technology Inc.

TABLE 49-14: ADDITIONAL VTR SUPPLY CURRENT WITH VARIOUS BLOCKS ENABLED

VCC	VTR	48 MHz Clock	VTR1			VTR_ANALOG			Units	Comments
			Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)		
On/Off	On	48MHz	0.015	0.015	0.015	0.01	0.01	0.01	mA	Comparator 0 enabled
On/Off	On	48MHz	0.015	0.015	0.015	0.01	0.01	0.01	mA	Comparator 1 enabled
On/Off	On	48MHz				0.45	0.50	0.50	mA	ADC enabled

Note 1: The values in this table are added to the values in VTR Supply Current, I_VTR excluding the sleep states.

TABLE 49-15: ADDITIONAL VTR SUPPLY CURRENT WITH OTHER BLOCKS ENABLED

VCC	VTR	48 MHz Clock	VTR_REG			VTR3			Units	Comments
			Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)	Typical (3.3V, 25C)	Max (3.45V, 70C)	Max (3.45V, 85C)		
On/Off	On	48MHz	0.3	0.3	0.3	0.01	0.01	0.01	mA	eSPI Enabled (eSPI Clock at 66MHz)

Note 1: The values in this table are added to the values in VTR Supply Current, I_VTR excluding the sleep states.

TABLE 49-16: VBAT SUPPLY CURRENT, I_VBAT (VBAT=3.3V)

VCC	VTR	48 MHz PLL	Typical (3.3V, 25^0 C)	Max (3.3V, 25^0 C)	Units	Comments
Off	Off	Off	14	20	uA	Internal 32kHz oscillator - supplied by coin cell
Off	On	Off	65	70	uA	Internal 32kHz oscillator - add to VTR power well that supplies this current through the diode or is connected to the VBAT pin. This is not from the coin cell.
Off	Off	Off	14.5	20	uA	32kHz crystal oscillator
Off	Off	Off	14	20	uA	External 32kHz clock on XTAL2 pin -Running
Off	Off	Off	14	20	uA	External 32kHz clock on XTAL2 pin -Low

50.0 TIMING DIAGRAMS

50.1 Power-up and Power-down Timing

FIGURE 50-1: VTR/VBAT POWER-UP TIMING

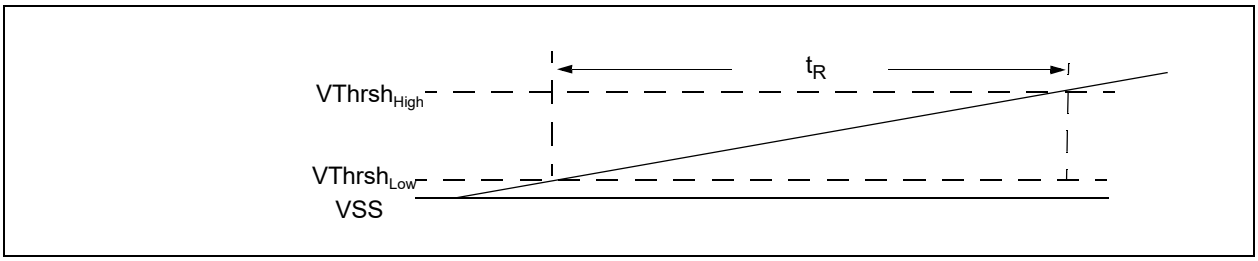


FIGURE 50-2: VTR RESET AND POWER-DOWN

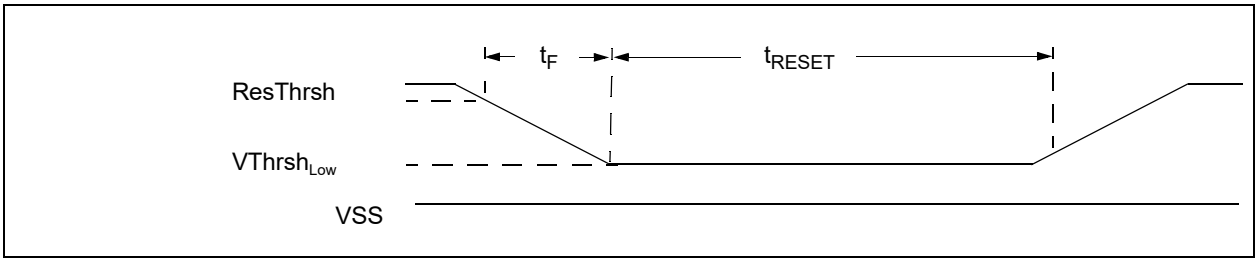


TABLE 50-1: VTR/VBAT TIMING PARAMETERS

Symbol	Parameter	MIN	TYP	MAX	Units	Notes
t_F	VTR Fall time	30			µs	1
	VBAT Fall time	30			µs	
t_R	VTR Rise time	0.050		20	ms	1
	VBAT Rise time	0.100		20	ms	
t_{RESET}	Minimum Reset Time	1			µs	
$VThrsh_{Low}$	VTR Low Voltage Threshold	0.1 × VTR			V	1
	VBAT Low Voltage Threshold	0.1 × VBAT			V	
$VThrsh_{High}$	VTR High Voltage Threshold			0.9 × VTR	V	1
	VBAT High Voltage Threshold			0.9 × VBAT	V	
ResThrsh	VTR Reset Threshold	0.5	1.8	2.7	V	1
	VBAT Reset Threshold	0.5	1.25	1.9	V	

Note 1: VTR applies to both VTR_REG and VTR_ANALOG

© 2020 Microchip Technology Inc.

50.2 Power Sequencing

FIGURE 50-3: POWER RAIL SEQUENCING

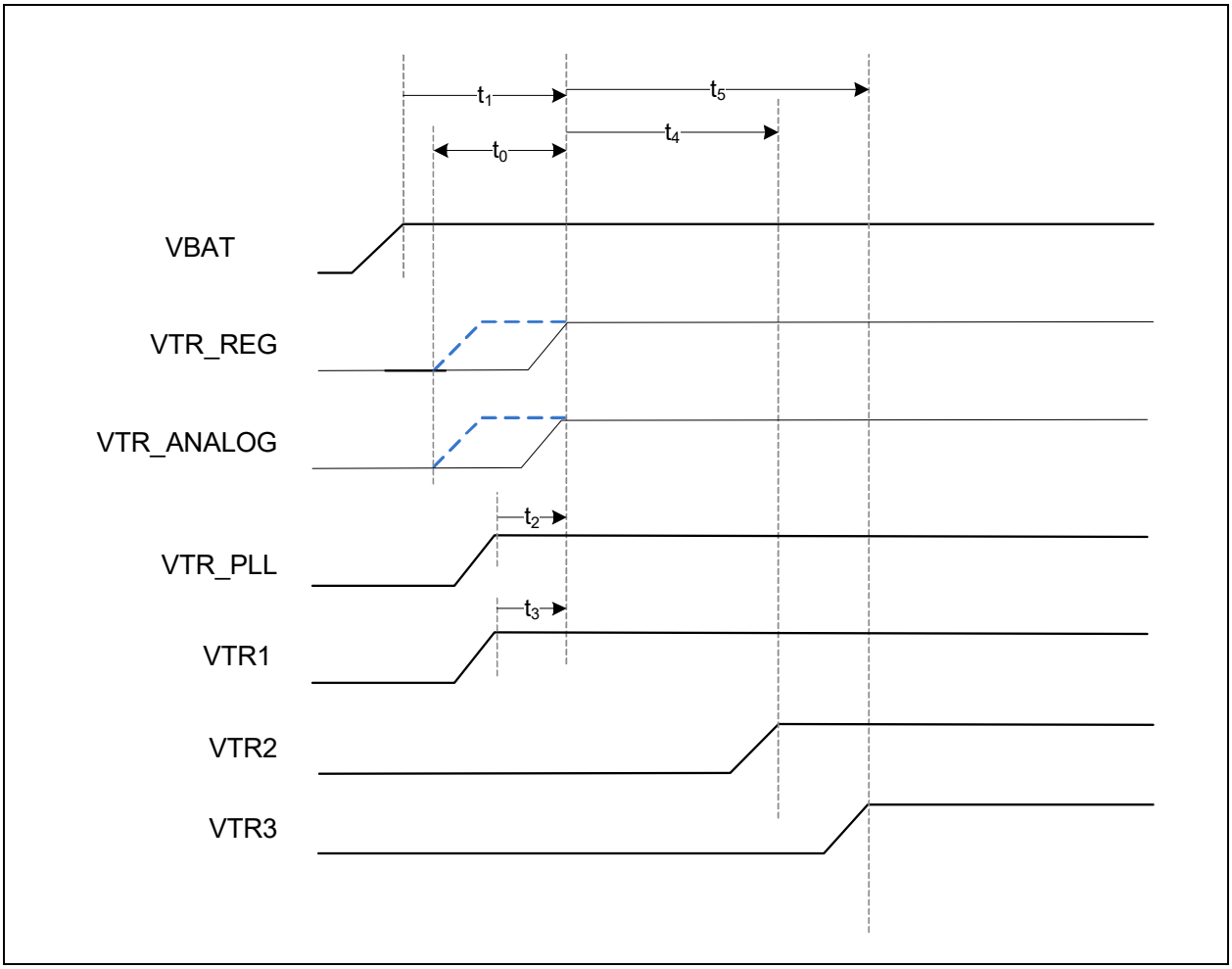


TABLE 50-2: POWER SEQUENCINGPARAMETERS

Symbol	Parameter	Min	Typ	Max	Units	Notes
t_0	VTR_ANALOG above minimum operating threshold to VTR_REG above minimum operating threshold	0		1	ms	1, 5
	VTR_REG above minimum operating threshold to VTR_ANALOG above minimum operating threshold	0		1	ms	
t_1	VBAT above minimum operating threshold to VTR_ANALOG and VTR_REG are both above minimum operating thresholds	0			µs	2

TABLE 50-2: POWER SEQUENCINGPARAMETERS (CONTINUED)

Symbol	Parameter	Min	Typ	Max	Units	Notes
t_2	VTR_PLL above minimum operating threshold to VTR_ANALOG above minimum operating threshold			0	ms	3, 5
t_3	VTR_ANALOG and VTR_REG are both above minimum operating thresholds to VTR1 above minimum operating threshold.	0		1	ms	6 3, 5
t_4	VTR_ANALOG and VTR_REG are both above minimum operating thresholds to VTR2 above minimum operating threshold. VTR2 at1.8V(nom) or 3.3V(nom)	0		1	ms	6, 3, 5
t_5	FOR ESPI BOOT; VTR3=1.8V VTR_ANALOG and VTR_REG are both above minimum operating thresholds to VTR3 above minimum operating threshold.	0		30	sec	6 5
t_5	FOR NON-ESPI BOOT; VTR3=1.8 or 3.3V VTR_ANALOG and VTR_REG are both above minimum operating thresholds to VTR3 above minimum operating threshold.	0			ms	6 5

Note 1: VTR_ANALOG and VTR_REG may ramp in either order

2: VBAT must rise no later than VTR_ANALOG and VTR_REG. This relationship is guaranteed by the recommended battery circuit in the layout guidelines.

3: The SHD_CS# pin, which is powered by VTR2, must be powered before the Boot ROM samples this pin.

4: If booting over eSPI, the EC boot ROM code monitors GPIO227/SHD_IO2, which is a VTR2 signal, to determine that VTR3 is active. The maximum time is the time after which the code abandons the boot.

5: Minimum operating threshold values for Power Rails are defined in Table 49-1, "Power Supply Operating Conditions".

6: VTR3 may be either 1.8V or 3.3V, software must program the GPIO Bank Power register for VTR3 pins before any of the VTR3 powered pins are used.

© 2020 Microchip Technology Inc.

50.3 nRESET_IN Timing

FIGURE 50-4: NRESET_IN TIMING

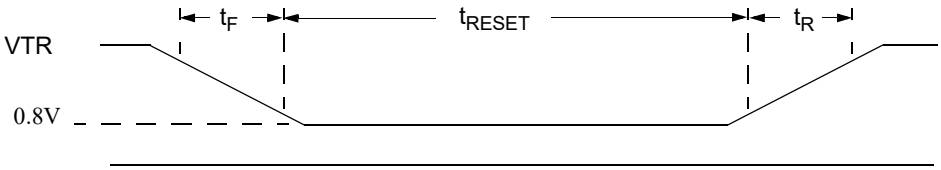


TABLE 50-3: RESETI# TIMING PARAMETERS

Symbol	Parameter	Limits		Units	Comments
		MIN	MAX		
t_F	nRESET_IN Fall time	0	1	ms	
t_R	nRESET_IN Rise time	0	1	ms	
t_{RESET}	Minimum Reset Time	1		µs	Note 1
Note 1: The nRESET_IN input pin can tolerate glitches of no more than 50ns.					

50.4 Clocking AC Timing Characteristics

FIGURE 50-5: CLOCK TIMING DIAGRAM

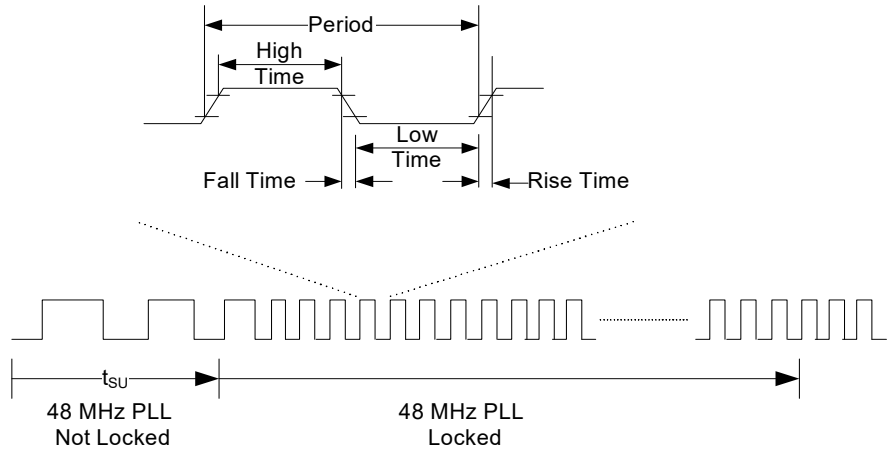


TABLE 50-4: CLOCK TIMING PARAMETERS

Clock	Symbol	Parameters	MIN	TYP	MAX	Units
48 MHz PLL	t_{SU}	Start-up accuracy from power-on-reset and waking from Heavy Sleep (Note 6)	-	-	3	ms
	-	Operating Frequency (locked to 32KHz single-ended input) (Note 1)	47.5	48	48.5	MHz
	-	Operating Frequency (locked to 32KHz Silicon Oscillator) (Note 1)	46.56	48	49.44	MHz
	CCJ	Cycle to Cycle Jitter(Note 2)	-200		200	ps
	t_{DO}	Output Duty Cycle	45	-	55	%
32MHz Ring Oscillator	-	Operating Frequency	16	-	48	MHz

Note 1: The 48MHz PLL is frequency accuracy is computed by adding +/-1% to the accuracy of the 32kHz reference clock.

2: The Cycle to Cycle Jitter of the 48MHz PLL is +/-200ps based on an ideal 32kHz clock source. The actual jitter on the 48MHz clock generated is computed by adding the clock jitter of the 32kHz reference clock to the 48MHz PLL jitter (e.g., 32kHz jitter +/- 200ps).

3: See the PCB Layout guide for design requirements and recommended 32.768 kHz Crystal Oscillators.

4: An external single-ended 32KHz clock is required to have an accuracy of +/- 100 ppm.

5: The external single-ended 32KHz clock source may be connected to either the XTAL2 pin or 32KHZ_IN pin.

6: PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset

© 2020 Microchip Technology Inc.

TABLE 50-4: CLOCK TIMING PARAMETERS (CONTINUED)

Clock	Symbol	Parameters	MIN	TYP	MAX	Units
32.768 kHz Crystal Oscillator (Note 3)	-	Operating Frequency	-	32.768	-	kHz
32KHz Silicon Oscillator	-	Operating Frequency	32.112	32.768	33.424	kHz
	-	Start-up delay from 0k Hz to Operating Frequency			150	us
32KHz single-ended input (Note 5)	-	Operating Frequency	-	32.768	-	kHz
	-	Period	(Note 4)	30.52	(Note 4)	µs
	-	High Time	10			us
	-	Low Time	10			us
	-	Fall Time	-	-	1	us
	-	Rise Time	-	-	1	us

Note 1: The 48MHz PLL is frequency accuracy is computed by adding +/-1% to the accuracy of the 32kHz reference clock.

2: The Cycle to Cycle Jitter of the 48MHz PLL is +/-200ps based on an ideal 32kHz clock source. The actual jitter on the 48MHz clock generated is computed by adding the clock jitter of the 32kHz reference clock to the 48MHz PLL jitter (e.g., 32kHz jitter +/- 200ps).

3: See the PCB Layout guide for design requirements and recommended 32.768 kHz Crystal Oscillators.

4: An external single-ended 32KHz clock is required to have an accuracy of +/- 100 ppm.

5: The external single-ended 32KHz clock source may be connected to either the XTAL2 pin or 32KHZ_IN pin.

6: PLL is started, either from waking from the Heavy Sleep mode, or after a Power On Reset

50.5 GPIO Timings

FIGURE 50-6: GPIO TIMING

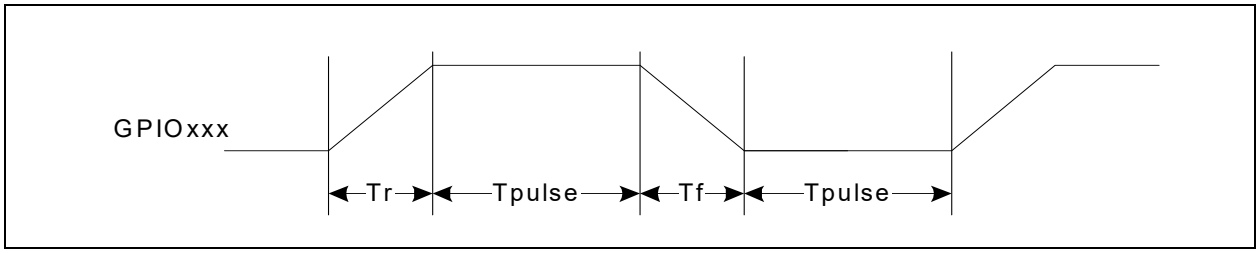


TABLE 50-5: GPIO TIMING PARAMETERS

Symbol	Parameter	MIN	TYP	MAX	Unit	Notes
t_R	GPIO Rise Time (push-pull)	0.54		1.31	ns	1
t_F	GPIO Fall Time	0.52		1.27	ns	
t_R	GPIO Rise Time (push-pull)	0.58		1.46	ns	2
t_F	GPIO Fall Time	0.62		1.48	ns	
t_R	GPIO Rise Time (push-pull)	0.80		2.00	ns	3
t_F	GPIO Fall Time	0.80		1.96	ns	
t_R	GPIO Rise Time (push-pull)	1.02		2.46	ns	4
t_F	GPIO Fall Time	1.07		2.51	ns	
t_{pulse}	GPIO Pulse Width	60			ns	

Note 1: Pad configured for 2ma, CL=2pF

2: Pad configured for 4ma, CL=5pF

3: Pad configured for 8ma, CL=10pF

4: Pad configured for 12ma, CL=20pF

© 2020 Microchip Technology Inc.

50.6 Boot from SPI Flash Timing

Refer to MEC152x Boot ROM document for the sequence and timing

50.7 Boot from eSPI Timing

Refer to MEC152x Boot ROM document for the sequence and timing

50.8 VCC_PWRGD Timing

FIGURE 50-7: VCC_PWRGD TIMING

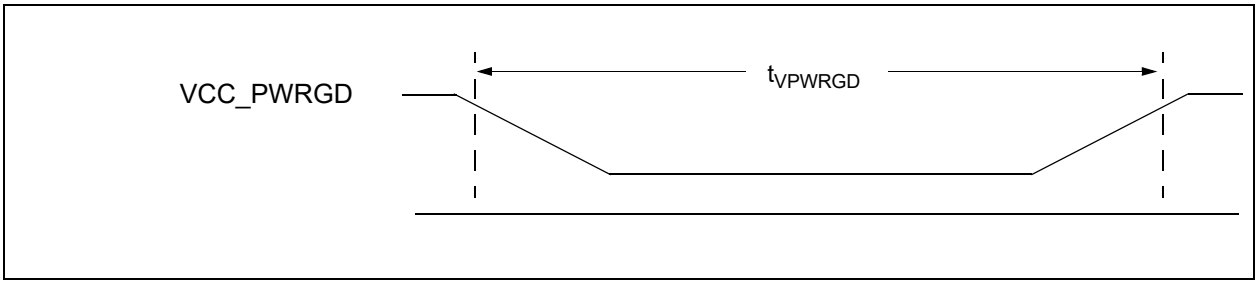


TABLE 50-6: VCC_PWRGD POWER TIMING PARAMETERS

Symbol	Parameter	Limits		Units	Notes
		MIN	MAX		
t_{VPWRGD}	VCC_PWRGD Pulse Width	31		ns	

© 2020 Microchip Technology Inc.

50.9 Serial Port (UART) Data Timing

FIGURE 50-8: SERIAL PORT DATA

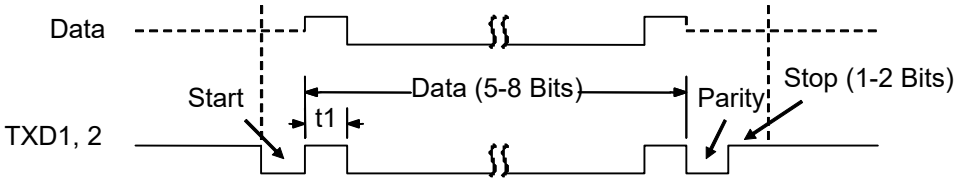


TABLE 50-7: SERIAL PORT DATA PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Serial Port Data Bit Time		t_{BR} (Note 1)		nsec

Note 1: tBR is 1/Baud Rate. The Baud Rate is programmed through the Baud_Rate_Divisor bits located in the Programmable Baud Rate Generator registers. The selectable baud rates are listed in Table 16-8, "UART Baud Rates using Clock Source 1.8432MHz" and Table 16-9, "UART Baud Rates using Clock Source 48MHz"Some of the baud rates have some percentage of error because the clock does not divide evenly. This error can be determined from the values in these baud rate tables.

50.10 PECI Interface

TABLE 50-8: PECI RECEIVE/SEND DATA TIMING (NOT AVAILABLE IN PUBLIC FROM INTEL)

Name	Description	MIN	MAX	Units	Notes
t_{BIT}	Bit time (overall time evident on PECI pin) Bit time driven by an originator	0.495 0.495	500 250	µsec µsec	Note 1
t_{H1}	High level time for logic 1	0.6	0.8	t_{BIT}	Note 2
t_{H0}	High level time for logic 0	0.2	0.4	t_{BIT}	
t_{PECIR}	Rise time (measured from V_{OL} to $V_{IH,min}$, $Vtt_{(nom)}$–5%)	-	30 + (5 x #nodes)	ns	Note 3
t_{PECIF}	Fall time (measured from V_{OH} to $V_{IL,max}$, $Vtt_{(nom)}$+5%)	-	(30 x #nodes)	ns	Note 3

Note 1: The originator must drive a more restrictive time to allow for quantized sampling errors by a client yet still attain the minimum time less than 500 µsec. t_{BIT} limits apply equally to t_{BIT-A} and t_{BIT-M}. The MEC152x is designed to support 2 MHz, or a 500ns bit time. See the PECI 3.0 specification from Intel Corp. for further details.

2: The minimum and maximum bit times are relative to t_{BIT} defined in the Timing Negotiation pulse. See the PECI 3.0 specification from Intel Corp. for further details.

3: “#nodes” is the number of nodes on the PECI bus; host and client nodes are counted as one each. Extended trace lengths may appear as extra nodes. Refer also to Table 27-2, "PECI Routing Guidelines". See the PECI 3.0 specification from Intel Corp. for further details.

Note: At the time of the release of Revision 0.3 (11-07-12) of this document, the relevant Intel specs (e.g., those listed below) still do not provide these parameters nor do they include the PECI Receive/Send Data Timing figure.

- Ibex Peak Chipset External Design Specification (EDS) Revision 1.0 September 2008 Document Number: 401376
- Patsburg Chipset External Design Specification (EDS) Revision 1.5 June 2011 Document Number: 450911

© 2020 Microchip Technology Inc.

50.11 8042 Emulation CPU_Reset Timing

FIGURE 50-9: KBRST TIMING

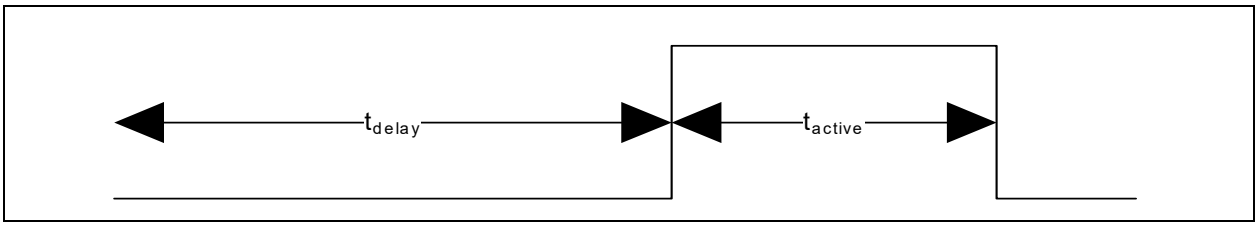


TABLE 50-9: KBRST TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{delay}	Delay prior to active pulse	14	15	15.5	µs
t_{active}	Active pulse width	6	8	8.5	µs

The KBRST pin is the CPU_RESET signal described in Section 11.10.2, "CPU_RESET Hardware Speed-Up"

50.12 Keyboard Scan Matrix Timing

TABLE 50-10: ACTIVE PRE DRIVE MODE TIMING

Parameter	Symbol	Value			Units	Notes
		MIN	TYP	MAX		
Active Predrive Mode	$t_{PREDRIVE}$		41.7		ns	

Note: The TYP value is based on two 48 MHz PLL clocks. The MIN and MAX values are dependent on the accuracy of the 48 MHz PLL.

© 2020 Microchip Technology Inc.

50.13 PS/2 Timing

FIGURE 50-10: PS/2 TRANSMIT TIMING

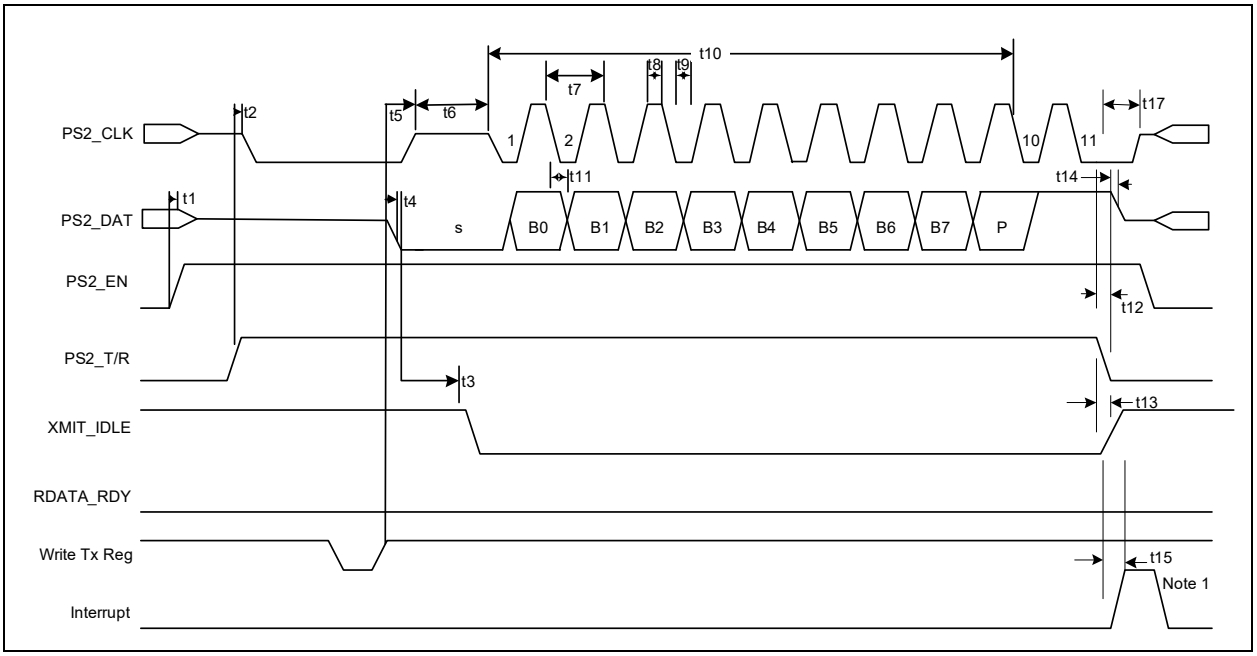


TABLE 50-11: PS/2 CHANNEL TRANSMISSION TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	PS2_T/R bit set to CLK driven low preparing the PS/2 Channel for data transmission.				
t3	CLK line floated to XMIT_IDLE bit deasserted.			1.7	
t4	Trailing edge of WR to Transmit Register to DATA line driven low.	45		90	
t5	Trailing edge of EC WR of Transmit Register to CLK line floated.	90		130	ns
t6	Initiation of Start of Transmit cycle by the PS/2 channel controller to the auxiliary peripheral's responding by latching the Start bit and driving the CLK line low.	0.002		25.003	ms
t7	Period of CLK	60		302	µs
t8	Duration of CLK high (active)	30		151	
t9	Duration of CLK low (inactive)				

TABLE 50-11: PS/2 CHANNEL TRANSMISSION TIMING PARAMETERS (CONTINUED)

Name	Description	MIN	TYP	MAX	Units
t10	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t11	DATA output by MEC152x following the falling edge of CLK. The auxiliary peripheral device samples DATA following the rising edge of CLK.			1.0	µs
t12	Rising edge following the 11th falling clock edge to PS_T/R bit driven low.	3.5		7.1	µs
t13	Trailing edge of PS_T/R to XMIT_IDLE bit asserted.			500	ns
t14	DATA released to high-Z following the PS2_T/R bit going low.				
t15	XMIT_IDLE bit driven high to interrupt generated.				
t17	Trailing edge of CLK is held low prior to going high-Z				

© 2020 Microchip Technology Inc.

FIGURE 50-11: PS/2 RECEIVE TIMING

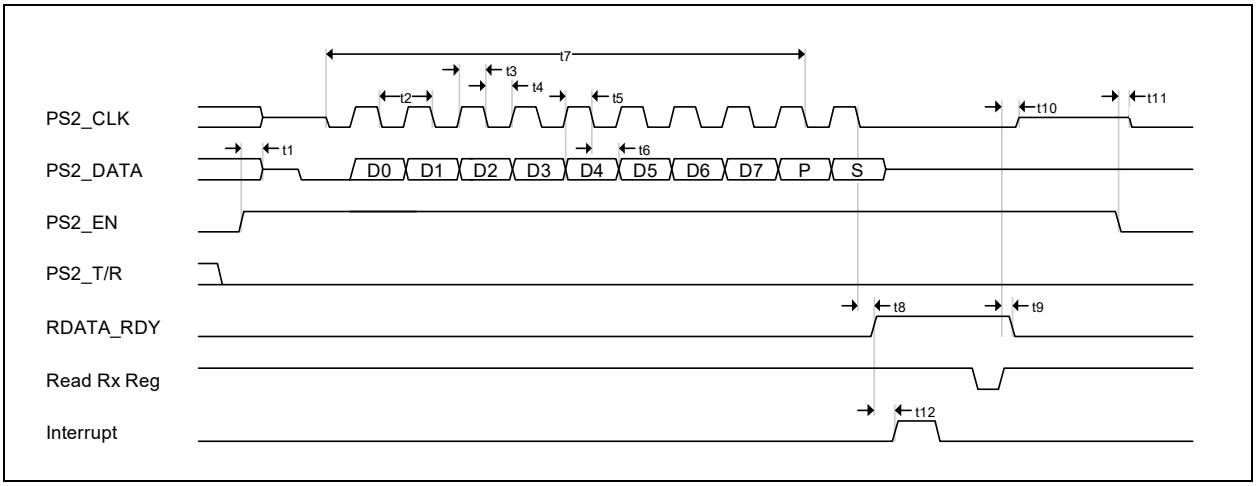


TABLE 50-12: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	Period of CLK	60		302	µs
t3	Duration of CLK high (active)	30		151	
t4	Duration of CLK low (inactive)				
t5	DATA setup time to falling edge of CLK. MEC152x samples the data line on the falling CLK edge.	1			
t6	DATA hold time from falling edge of CLK. MEC152x samples the data line on the falling CLK edge.	2			
t7	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t8	Falling edge of 11th CLK to RDATA_RDY asserted.			1.6	µs

TABLE 50-12: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS (CONTINUED)

Name	Description	MIN	TYP	MAX	Units
t9	Trailing edge of the EC's RD signal of the Receive Register to RDATA_RDY bit de-asserted.			500	ns
t10	Trailing edge of the EC's RD signal of the Receive Register to the CLK line released to high-Z.				
t11	PS2_CLK is "Low" and PS2_DATA is "Hi-Z" when PS2_EN is de-asserted.				
t12	RDATA_RDY asserted an interrupt is generated.				

© 2020 Microchip Technology Inc.

50.14 PWM Timing

FIGURE 50-12: PWM OUTPUT TIMING

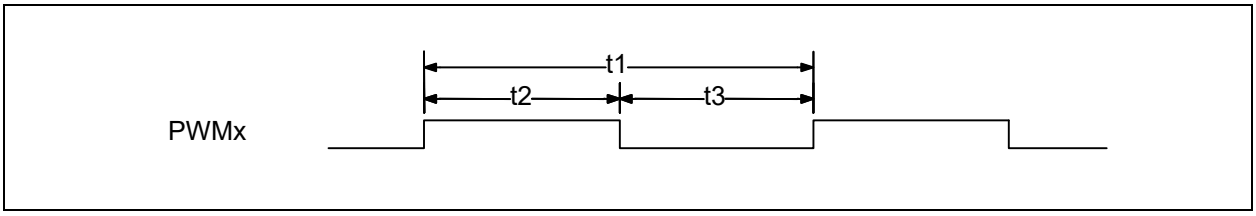


TABLE 50-13: PWM TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Period	42ns		23.3sec	
t_f	Frequency	0.04Hz		24MHz	
t2	High Time	0		11.65	sec
t3	Low Time	0		11.65	sec
t_d	Duty cycle	0		100	%

50.15 Fan Tachometer Timing

FIGURE 50-13: FAN TACHOMETER INPUT TIMING

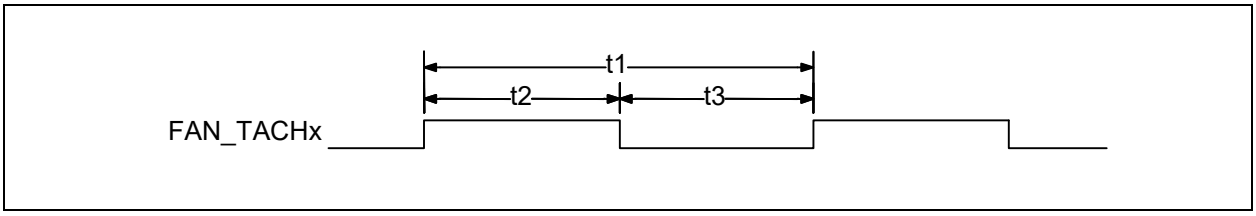


TABLE 50-14: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time	100			µsec
t2	Pulse High Time	20			
t3	Pulse Low Time	20			
Note:	t_{TACH} is the clock used for the tachometer counter. It is 30.52 * prescaler, where the prescaler is programmed in the Fan Tachometer Timebase Prescaler register.				

© 2020 Microchip Technology Inc.

50.16 Blinking/Breathing PWM Timing

FIGURE 50-14: BLINKING/BREATHING PWM OUTPUT TIMING

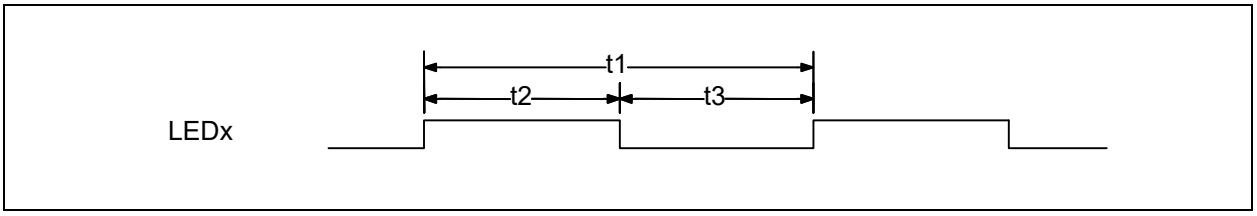


TABLE 50-15: BLINKING/BREATHING PWM TIMING PARAMETERS, BLINKING MODE

Name	Description	MIN	TYP	MAX	Units
t1	Period	7.8ms		32sec	
t_f	Frequency	0.03125		128	Hz
t2	High Time	0		16	sec
t3	Low Time	0		16	sec
t_d	Duty cycle	0		100	%

TABLE 50-16: BLINKING/BREATHING PWM TIMING PARAMETERS, GENERAL PURPOSE

Name	Description	MIN	TYP	MAX	Units
t1	Period	5.3µs		21.8ms	
t_f	Frequency	45.8Hz		187.5kHz	
t2	High Time	0		10.9	ms
t3	Low Time	0		10.9	ms
t_d	Duty cycle	0		100	%

50.17 I2C/SMBus Timing

FIGURE 50-15: I2C/SMBUS TIMING

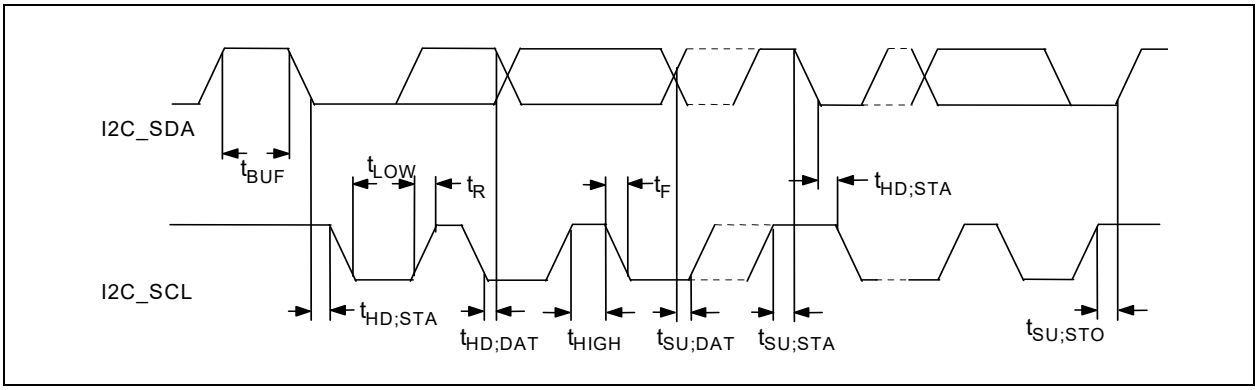


TABLE 50-17: I2C/SMBUS TIMING PARAMETERS

Symbol	Parameter	Standard-Mode		Fast-Mode		Fast-Mode Plus		Units
		MIN	MAX	MIN	MAX	MIN	MAX	
f_{SCL}	SCL Clock Frequency		100		400		1000	kHz
t_{BUF}	Bus Free Time	4.7		1.3		0.5		µs
$t_{SU;STA}$	START Condition Set-Up Time	4.7		0.6		0.26		µs
$t_{HD;STA}$	START Condition Hold Time	4.0		0.6		0.26		µs
t_{LOW}	SCL LOW Time	4.7		1.3		0.5		µs
t_{HIGH}	SCL HIGH Time	4.0		0.6		0.26		µs
t_R	SCL and SDA Rise Time		1.0		0.3		0.12	µs
t_F	SCL and SDA Fall Time		0.3		0.3		0.12	µs
$t_{SU;DAT}$	Data Set-Up Time	0.25		0.1		0.05		µs
$t_{HD;DAT}$	Data Hold Time	0		0		0		µs
$t_{SU;STO}$	STOP Condition Set-Up Time	4.0		0.6		0.26		µs

© 2020 Microchip Technology Inc.

50.18 Quad SPI Master Controller - Serial Peripheral Interface (QMSPI) Timings

FIGURE 50-16: SPI CLOCK TIMING

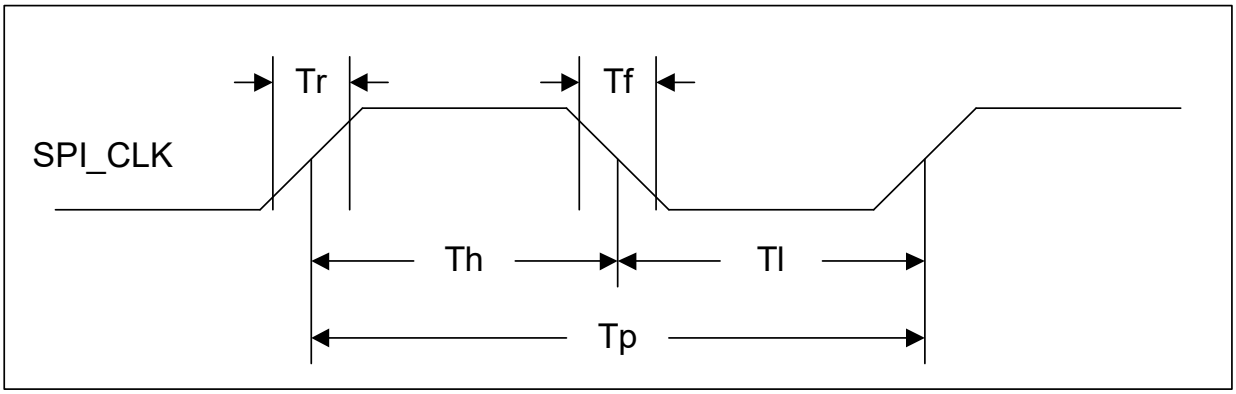


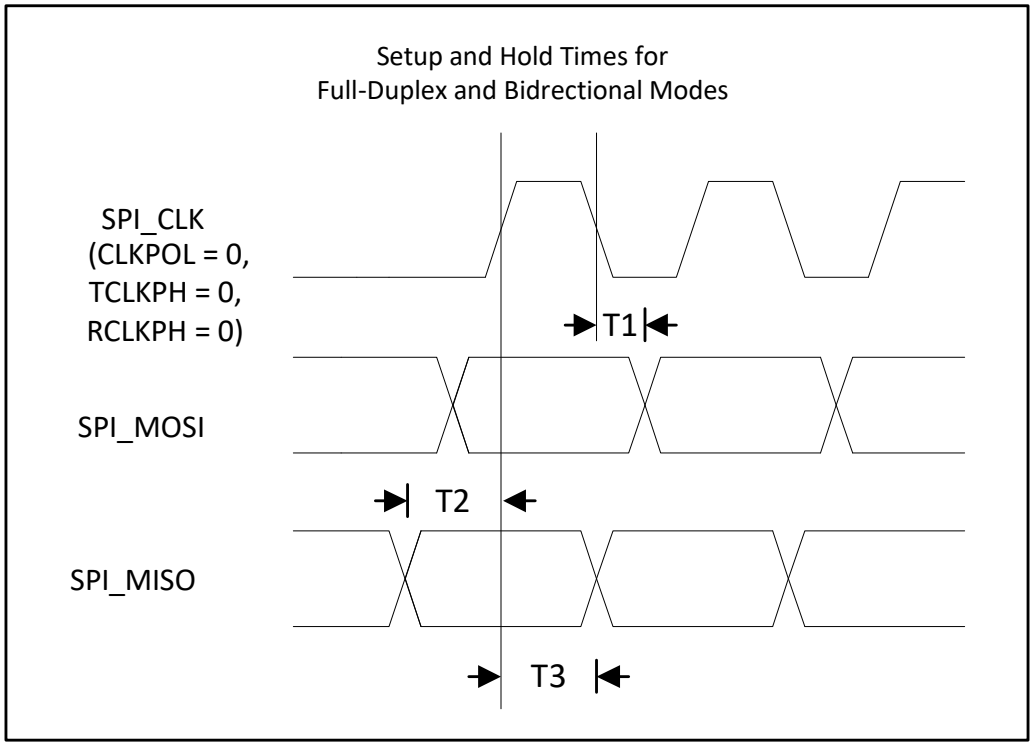
TABLE 50-18: SPI CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
Tr	SPI Clock Rise Time. Measured from 10% to 90%.			3	ns
Tf	SPI Clock Fall Time. Measured from 90% to 10%.			3	ns
Th/Tl	SPI Clock High Time/SPI Clock Low Time	40% of SPCLK Period	50% of SPCLK Period	60% of SPCLK Period	ns
Tp	SPI Clock Period – As selected by SPI Clock Generator Register	20.8		5,333	ns

Note 1: Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow.

2: SPI_IO[3:0] have the same Tr and Tf values as SPI_CLK

FIGURE 50-17: SPI SETUP AND HOLD TIMES



Note:	SPI_IO[3:0] obey the SPI_MOSI and SPI_MISO timing. In the 2-pin SPI Interface implementation, SPI_IO0 pin is the SPI Master-Out/Slave-In (MOSI) pin and the SPI_IO1 pin is the Master-In/Slave-out (MISO) pin.

TABLE 50-19: SPI SETUP AND HOLD TIMES PARAMETERS

Name	Description	MIN	TYP	MAX	Units
T1	Data Output Delay			2	ns
T2	Data IN Setup Time	5.5			ns
T3	Data IN Hold Time	0			ns
Note:	Test conditions are as follows: output load is C_L=30pF, pin drive strength setting is 4mA and slew rate setting is slow				

© 2020 Microchip Technology Inc.

50.19 Serial Debug Port Timing

FIGURE 50-18: SERIAL DEBUG PORT TIMING PARAMETERS

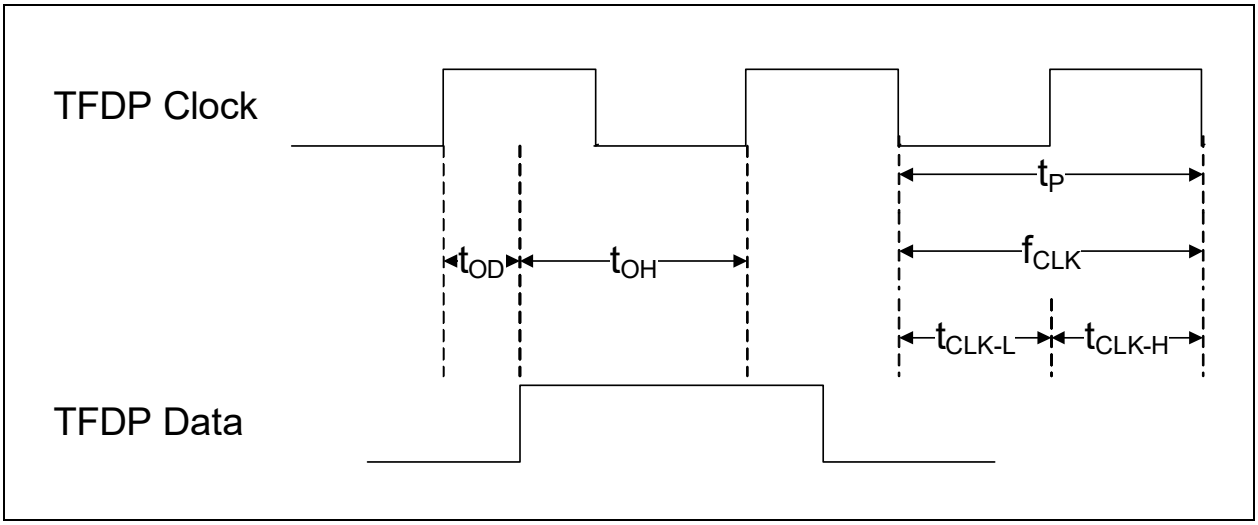


TABLE 50-20: SERIAL DEBUG PORT INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
f_{clk}	TFDP Clock frequency (see note)	2.5	-	24	MHz
t_P	TFDP Clock Period.	1/fclk			µs
t_{OD}	TFDP Data output delay after falling edge of TFDP_CLK.			5	nsec
t_{OH}	TFDP Data hold time after falling edge of TFDP Clock	$t_P - t_{OD}$			nsec
t_{CLK-L}	TFDP Clock Low Time	$t_P/2 - 3$		$t_P/2 + 3$	nsec
t_{CLK-H}	TFDP Clock high Time (see Note 1)	$t_P/2 - 3$		$t_P/2 + 3$	nsec

Note 1: When the clock divider for the embedded controller is an odd number value greater than 2h, then $t_{CLK-L} = t_{CLK-H} + 15$ ns. When the clock divider for the embedded controller is 0h, 1h, or an even number value greater than 2h, then $t_{CLK-L} = t_{CLK-H}$.

50.20 JTAG Interface Timing

FIGURE 50-19: JTAG POWER-UP & ASYNCHRONOUS RESET TIMING

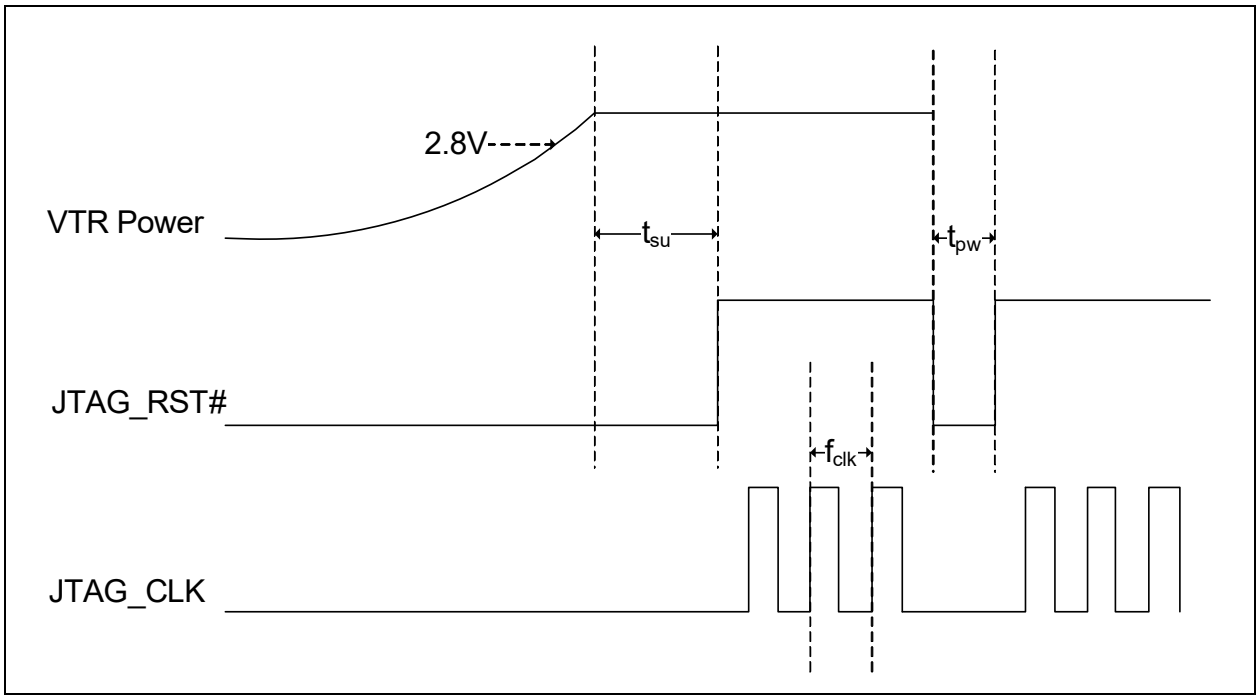
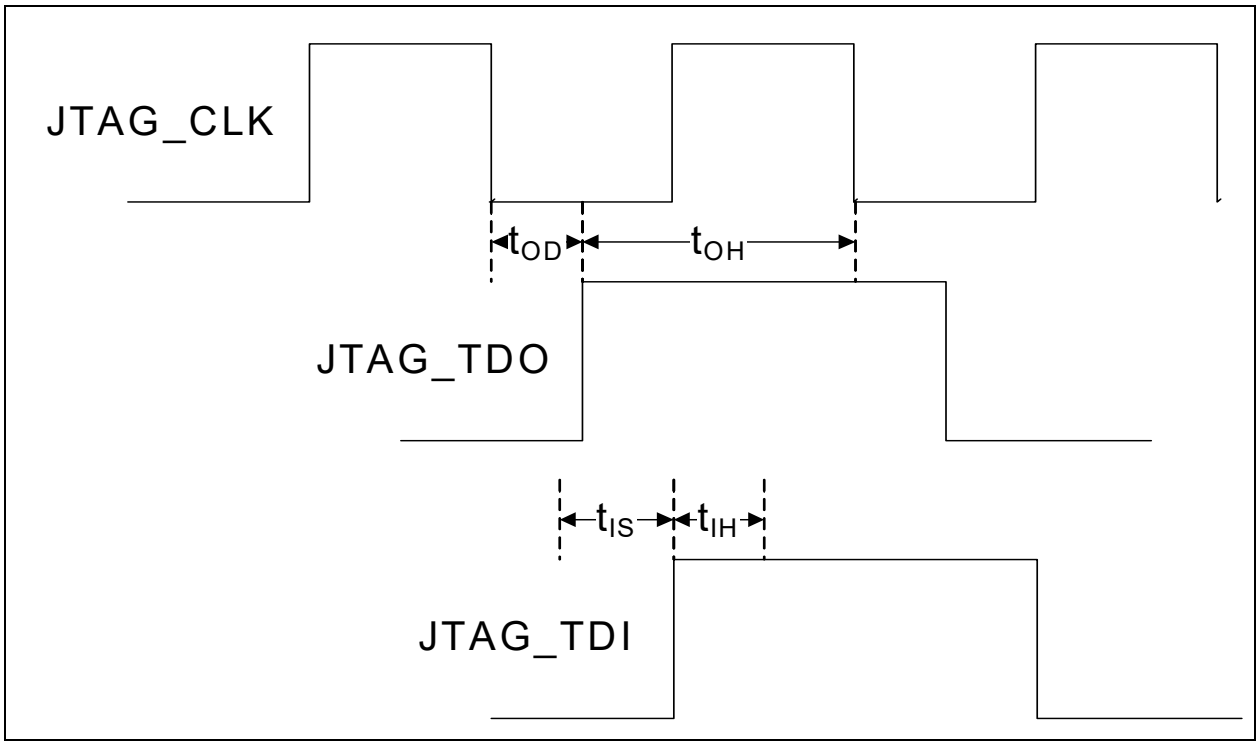


FIGURE 50-20: JTAG SETUP & HOLD PARAMETERS



© 2020 Microchip Technology Inc.

TABLE 50-21: JTAG INTERFACE TIMING PARAMETERS

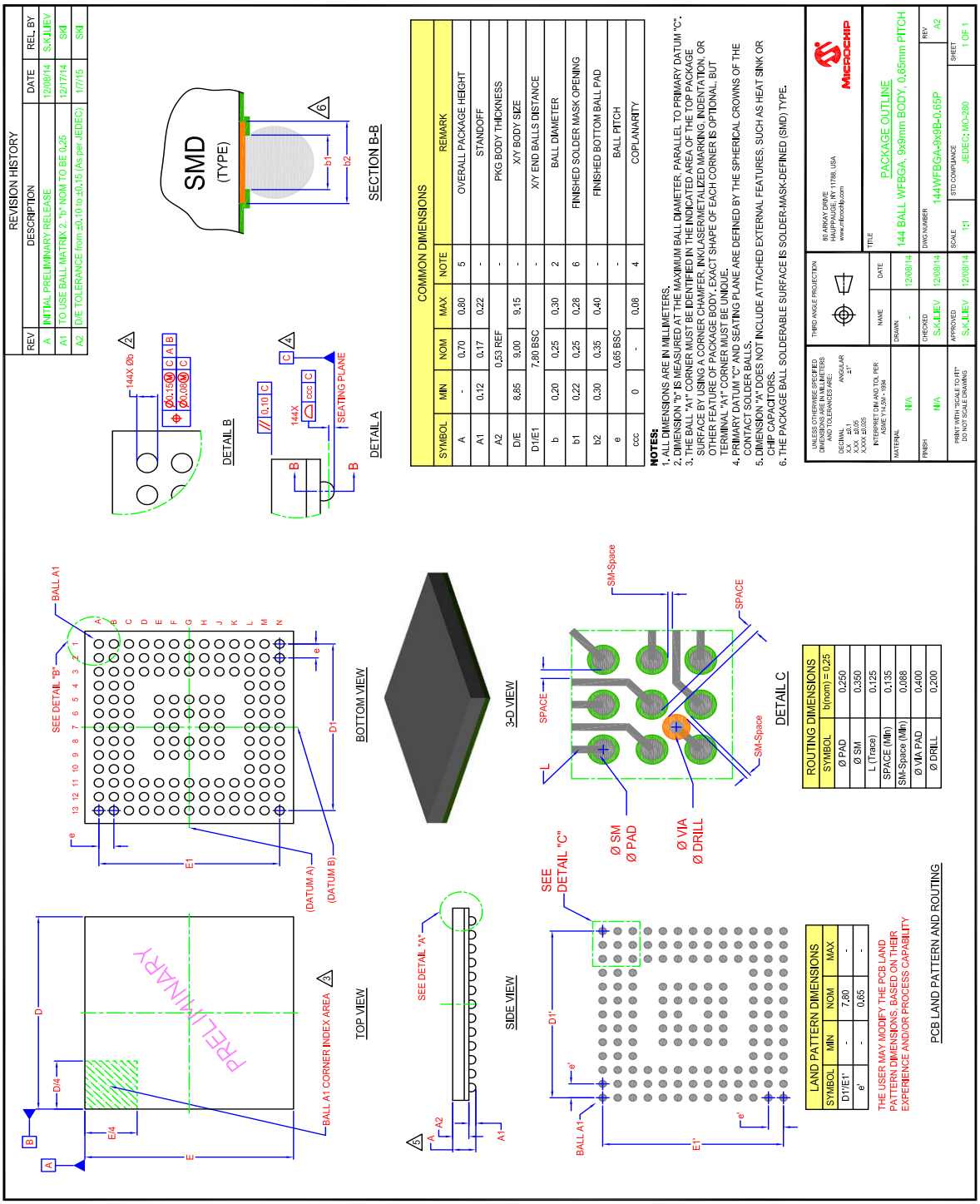
Name	Description	MIN	TYP	MAX	Units
t_{su}	JTAG_RST# de-assertion after VTR power is applied	5			ms
t_{pw}	JTAG_RST# assertion pulse width	500			nsec
f_{clk}	JTAG_CLK frequency (see note)			48	MHz
t_{OD}	TDO output delay after falling edge of TCLK.	5		10	nsec
t_{OH}	TDO hold time after falling edge of TCLK	1 TCLK - t_{OD}			nsec
t_{IS}	TDI setup time before rising edge of TCLK.	5			nsec
t_{IH}	TDI hold time after rising edge of TCLK.	5			nsec

Note: f_{clk} is the maximum frequency to access a JTAG Register.

51.0 PACKAGING INFORMATION

51.1 144 Pin WFBGA/SZ Package

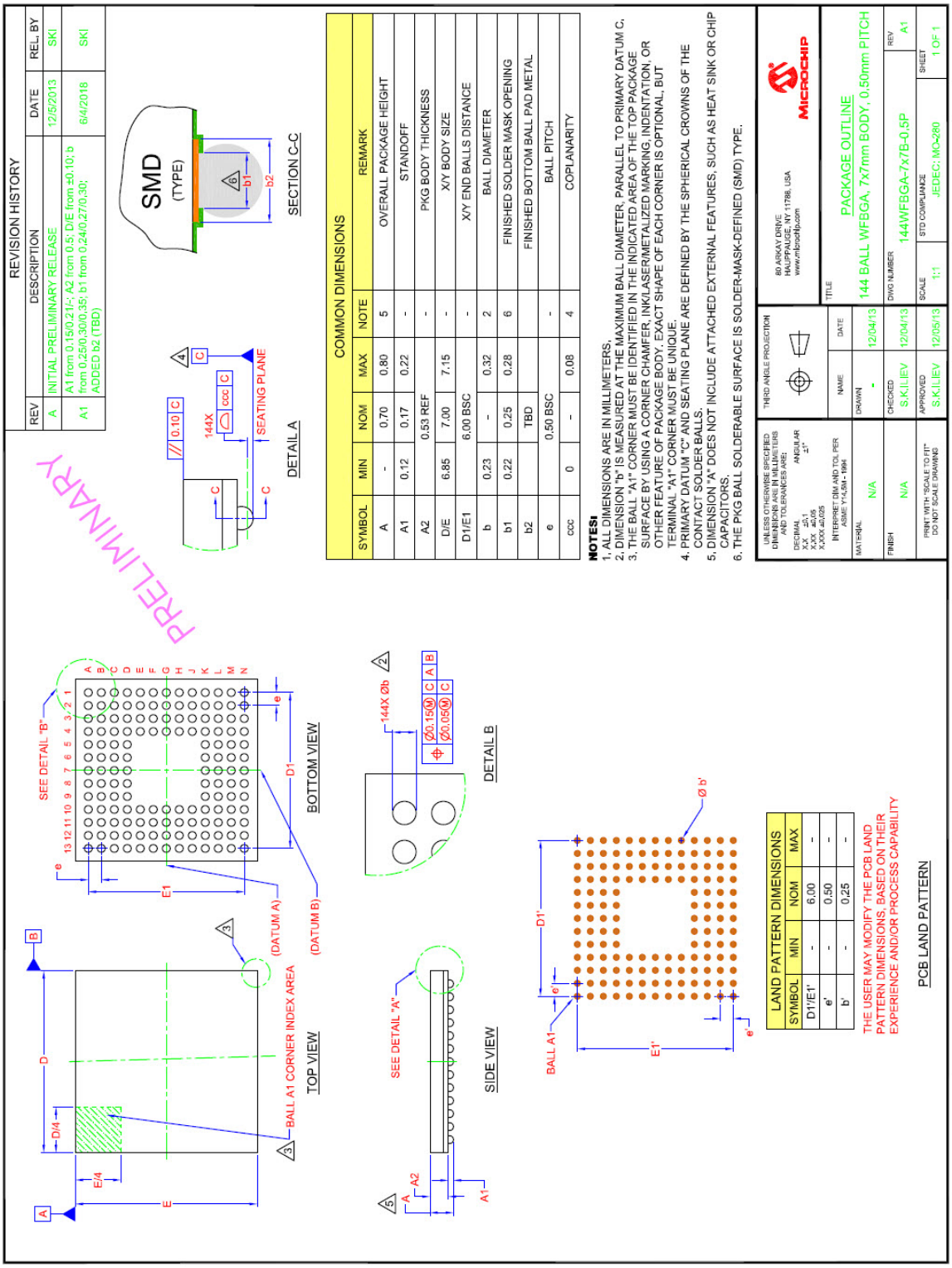
Note: For the most current package drawings, see the Microchip Packaging Specification at http://www.microchip.com/packaging.



© 2020 Microchip Technology Inc.

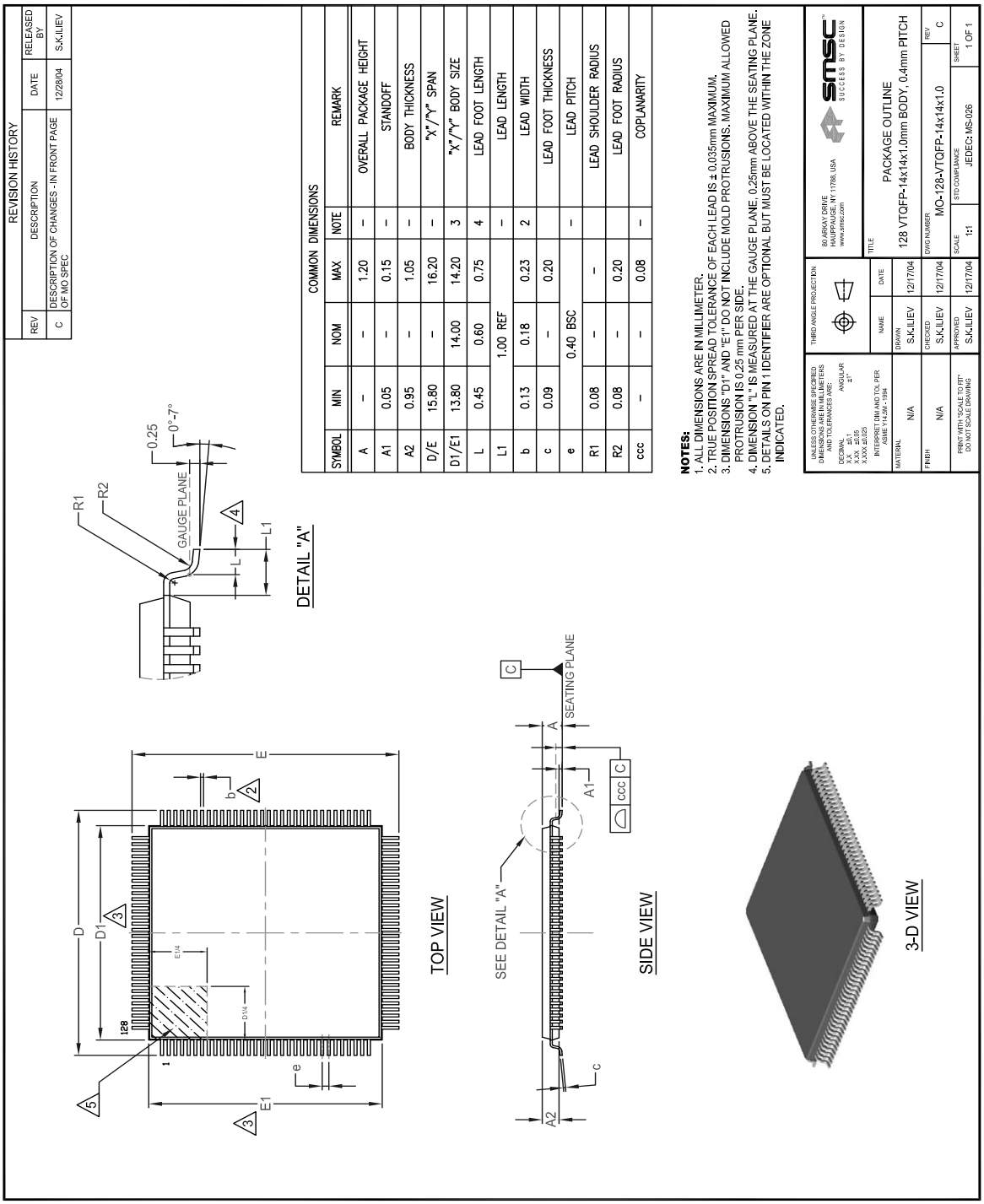
51.2 144 Pin WFBGA/3Y Package

Note: For the most current package drawings, see the Microchip Packaging Specification at http://www.microchip.com/packaging.



51.3 128 Pin VTQFP/Z2 Package

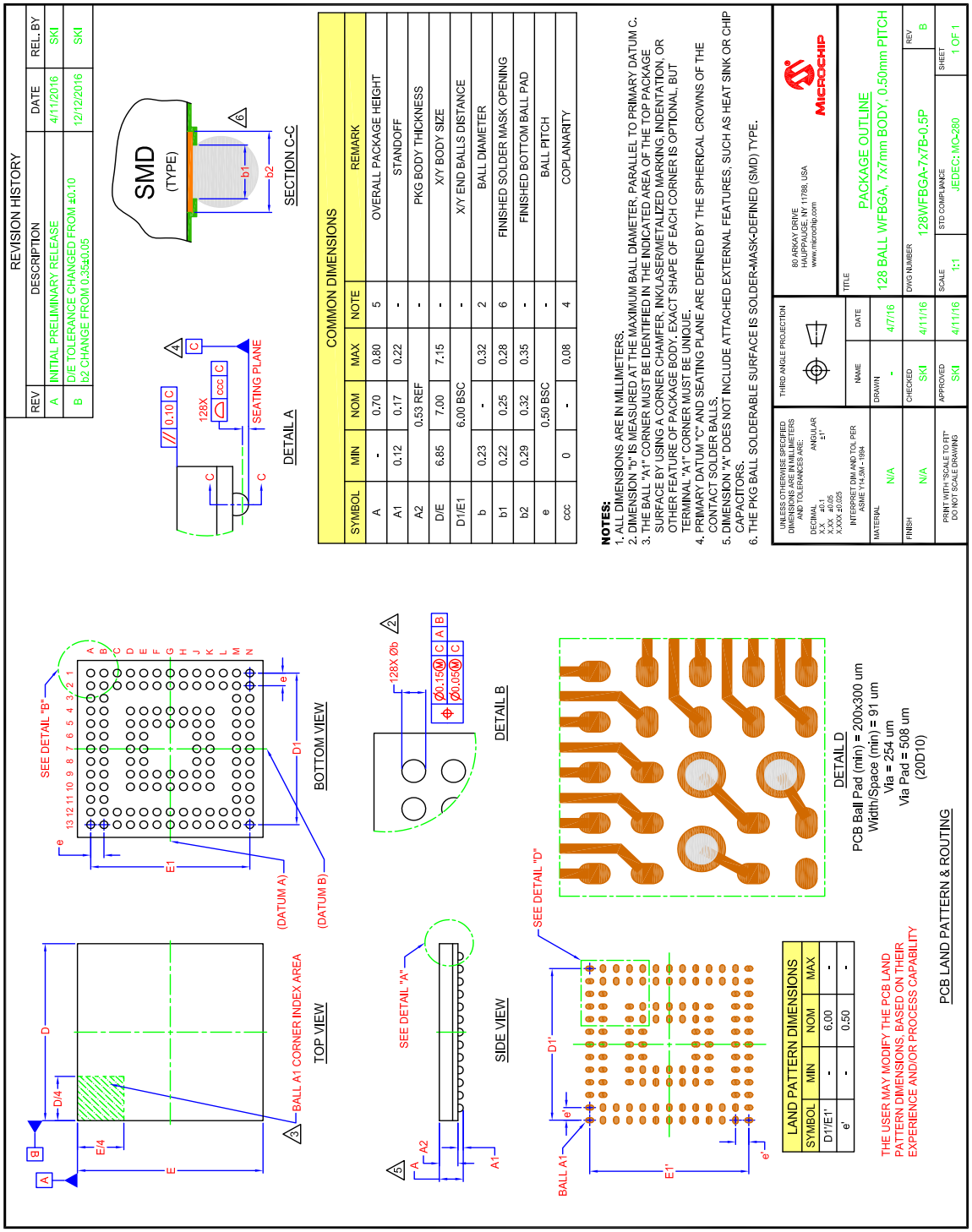
Note: For the most current package drawings, see the Microchip Packaging Specification at http://www.microchip.com/packaging.



© 2020 Microchip Technology Inc.

51.4 128 Pin WFBGA Package

Note: For the most current package drawings, see the Microchip Packaging Specification at http://www.microchip.com/packaging.



APPENDIX A: DATA SHEET REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision	Section/Figure/Entry	Correction
DS00003427D (10-20-20)	Table 50-18, "SPI Clock Timing Parameters"	SPI_10 [3:0] timing parameter added.
DS00003427C (05-04-20)	Table 2-1, "MEC152x PIN MAP"	Mention of “MEC150x” changed to “MEC152x” in table title.
DS00003427B (04-28-20)	Table 49-12, "VTR Supply Current, I_VTR" and Table 49-13, "VTR Supply Current, I_VTR"	Duplicate rows removed
	Table 49-5, "5V Tolerant Leakage Currents (VTR=3.3-5%)" and Table 49-6, "3.6V Tolerant Leakage Currents (VTR = 1.8V-5%)"	Leakage currents modified
	Product Identification System on page 568	Updated to include: “MEC1523 Embedded Controller with internal 2KB EEPROM” and “MEC1527 Embedded Controller with internal 512KB SPI-Flash”
DS00003427A (03-11-20)	Document Release	

© 2020 Microchip Technology Inc.

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at 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 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://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

Not all of the possible combinations of Device, Temperature Range and Package may be offered for sale. To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.[1]	-	X	-	XX	-	X/XXX[2]	-	[X][3]
Device		Total SRAM		Version/ Revision		Temp Range/ Package		Tape and Reel Option

Device:	MEC1521[1] MEC1523[1]	Embedded Controller for Notebook Embedded Controller with internal 2KB EEPROM
	MEC1527[1]	Embedded Controller with internal 512KB SPI-Flash
Total SRAM	H	256KB
Version/Revision:	B#	B = Version, # = Version Revision Number
Temperature Range:	I/	= -40°C to +85°C (Industrial)
Package:	Z2	128 pin VTQFP 14x14mm body, 0.40mm pitch
	TF	128 pin WFBGA[2], 7mm x 7mm body, 0.50mm pitch
	SZ	144 pin WFBGA[2], 9 mm x 9mm body, 0.65mm pitch
	3Y	144 pin WFBGA[2], 7mm x 7mm body, 0.50mm pitch
Tape and Reel Option:	Blank = Tray packaging TR = Tape and Reel[3]	

Examples:

a) MEC1521H-B0-I/Z2 = MEC1521, 256KB total SRAM, Standard ROM, ROM Version 1, 128-VTQFP 14x14mm body, Industrial grade, Tray packaging

b) MEC1527H-B0-I/SZ = MEC1527 with internal 512KB SPI flash, 256KB total SRAM, Standard ROM, ROM Version 1, 144-WFBGA 9x9mm body, Industrial grade, Tray packaging

c) MEC1523H-B0-I/3Y = MEC1523 with Internal 2KB EEPROM, 256KB total SRAM, Standard ROM,ROM Version 1, 144-WFBGA 7x7mm body, Industrial grade, tray packaging

Note 1: These products meet the halogen maximum concentration values per IEC61249-2-21.

2: All package options are RoHS compliant. For RoHS compliance and environmental information, please visit http://www.microchip.com/pagehandler/en-us/aboutus/ehs.html

3: 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

© 2020 Microchip Technology Inc.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "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.

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information 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.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. 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, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, 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.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks 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.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

All other trademarks mentioned herein are property of their respective companies.

© 2020, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 9781522469575

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

© 2020 Microchip Technology Inc.