A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee™

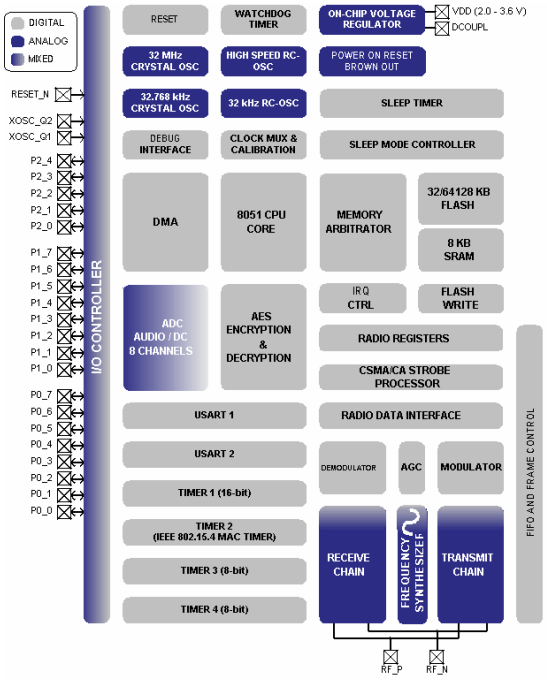
Applications

- *2.4 GHz IEEE 802.15.4 systems*
- *ZigBee™ systems*
- *Home/building automation*
- *Industrial Control and Monitoring*
- *Low power wireless sensor networks*
- *PC peripherals*
- *Set-top boxes and remote controls*
- *Consumer Electronics*

Product Description

The ***CC2430*** comes in three different versions: CC2430-F32/64/128, with 32/64/128 KB of flash memory respectively. The ***CC2430*** is a true System-on-Chip (SoC) solution specifically tailored for IEEE 802.15.4 and ZigBee™ applications. It enables ZigBee™ nodes to be built with very low total bill-of-material costs. The ***CC2430*** combines the excellent performance of the leading ***CC2420*** RF transceiver with an industry-standard enhanced 8051 MCU, 32/64/128 KB flash memory, 8 KB RAM and many other powerful features. Combined with the industry leading ZigBee™ protocol stack (Z-Stack) from Figure 8 Wireless / Chipcon, the ***CC2430*** provides the market's most competitive ZigBee™ solution.

The ***CC2430*** is highly suited for systems where ultra low power consumption is required. This is ensured by various operating modes. Short transition times between operating modes further ensure low power consumption.



This data sheet contains preliminary data, and supplementary data will be published at a later date. Chipcon reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. The product is not yet fully qualified at this point.

Key Features

- High performance and low power 8051 microcontroller core.
- 2.4 GHz IEEE 802.15.4 compliant RF transceiver (industry leading ***CC2420*** radio core).
- Excellent receiver sensitivity and robustness to interferers
- 32, 64 or 128 KB in-system programmable flash
- 8 KB RAM, 4 KB with data retention in all power modes
- Powerful DMA functionality
- Very few external components
- Only a single crystal needed for mesh network systems
- Low current consumption (RX: 27mA, TX: 25mA, microcontroller running at 32 MHz)
- Only 0.9µA current consumption in power-down mode, where external interrupts or the RTC can wake up the system
- Less than 0.6µA current consumption in stand-by mode, where external interrupts can wake up the system
- Very fast transition times from low-power modes to active mode enables ultra low average power consumption in low duty-cycle systems
- CSMA/CA hardware support.
- Wide supply voltage range (2.0V – 3.6V)
- Digital RSSI / LQI support
- Battery monitor and temperature sensor.
- 8-14 bits ADC with up to eight inputs
- AES security coprocessor
- Two powerful USARTs with support for several serial protocols.
- Watchdog timer

- One IEEE 802.15.4 MAC Timer, one general 16-bit timer and two 8-bit timers
- Hardware debug support
- 21 general I/O pins, two with 20mA sink/source capability
- Powerful and flexible development tools available
- RoHS compliant 7x7mm QLP48 package

Table Of Contents

1 ABBREVIATIONS 6

2 REFERENCES 8

3 REGISTER CONVENTIONS 9

4 FEATURES (CONTINUED FROM FRONT PAGE) 10

4.1 High-Performance and Low-Power 8051-Compatible Microcontroller 10
4.2 Up to 128 KB Non-volatile Program Memory and 2 x 4 KB Data Memory 10
4.3 Hardware AES Encryption/Decryption 10
4.4 Peripheral Features 10
4.5 Low Power 10
4.6 802.15.4 MAC hardware support 10
4.7 Integrated 2.4GHz DSSS Digital Radio 10

5 ABSOLUTE MAXIMUM RATINGS 11

6 OPERATING CONDITIONS 11

7 ELECTRICAL SPECIFICATIONS 12

7.1 General Characteristics 13
7.2 RF Receive Section 14
7.3 RF Transmit Section 15
7.4 32 MHz Crystal Oscillator 16
7.5 32.768 kHz Crystal Oscillator 16
7.6 Low Power RC Oscillator 16
7.7 High Speed RC Oscillator 17
7.8 Frequency Synthesizer Characteristics 17
7.9 Analog Temperature Sensor 18
7.10 8-14 bit ADC 18
7.11 Control AC Characteristics 19
7.12 SPI AC Characteristics 20
7.13 Debug Interface AC Characteristics 21
7.14 Port Outputs AC Characteristics 21
7.15 Timer Inputs AC Characteristics 22
7.16 DC Characteristics 22

8 PIN AND I/O PORT CONFIGURATION 23

9 CIRCUIT DESCRIPTION 25

9.1 CPU and Peripherals 26
9.2 Radio 27

10 POWER MANAGEMENT 28

11 APPLICATION CIRCUIT 29

11.1 Input / output matching 29
11.2 Bias resistors 29
11.3 Crystal 29
11.4 Voltage regulators 29
11.5 Power supply decoupling and filtering 29

12 8051 CPU 32

12.1 8051 CPU Introduction 32
12.2 Reset 32
12.3 Memory 32
12.4 SFR Registers 43
12.5 CPU Registers 46
12.6 Instruction Set Summary 48

12.7 INTERRUPTS 52
12.8 OSCILLATORS AND CLOCKS 62
12.9 DEBUG INTERFACE 62
12.10 RAM 66
12.11 FLASH MEMORY 66
12.12 MEMORY ARBITER 66
13 PERIPHERALS 69
13.1 I/O PORTS 69
13.2 DMA CONTROLLER 86
13.3 16-BIT TIMER, TIMER1 98
13.4 MAC TIMER (TIMER 2) 110
13.5 SLEEP TIMER 117
13.6 8-BIT TIMER 3 AND TIMER 4 118
13.7 ADC 127
13.8 RANDOM NUMBER GENERATOR 133
13.9 AES COPROCESSOR 135
13.10 POWER MANAGEMENT 140
13.11 POWER ON RESET AND BROWN OUT DETECTOR 144
13.12 WATCHDOG TIMER 145
13.13 USART 147
13.14 FLASH CONTROLLER 157
14 RADIO 164
14.1 IEEE 802.15.4 MODULATION FORMAT 165
14.2 COMMAND STROBES 166
14.3 RF REGISTERS 166
14.4 INTERRUPTS 166
14.5 FIFO ACCESS 169
14.6 DMA 170
14.7 RECEIVE MODE 170
14.8 RXFIFO OVERFLOW 170
14.9 TRANSMIT MODE 171
14.10 GENERAL CONTROL AND STATUS 172
14.11 DEMODULATOR, SYMBOL SYNCHRONIZER AND DATA DECISION 172
14.12 FRAME FORMAT 173
14.13 SYNCHRONIZATION HEADER 173
14.14 LENGTH FIELD 174
14.15 MAC PROTOCOL DATA UNIT 175
14.16 FRAME CHECK SEQUENCE 175
14.17 RF DATA BUFFERING 176
14.18 ADDRESS RECOGNITION 177
14.19 ACKNOWLEDGE FRAMES 178
14.20 RADIO CONTROL STATE MACHINE 179
14.21 MAC SECURITY OPERATIONS (ENCRYPTION AND AUTHENTICATION) 181
14.22 LINEAR IF AND AGC SETTINGS 181
14.23 RSSI / ENERGY DETECTION 181
14.24 LINK QUALITY INDICATION 182
14.25 CLEAR CHANNEL ASSESSMENT 182
14.26 FREQUENCY AND CHANNEL PROGRAMMING 183
14.27 VCO AND PLL SELF-CALIBRATION 183
14.28 OUTPUT POWER PROGRAMMING 183
14.29 INPUT / OUTPUT MATCHING 184
14.30 TRANSMITTER TEST MODES 184
14.31 SYSTEM CONSIDERATIONS AND GUIDELINES 186
14.32 PCB LAYOUT RECOMMENDATION 188
14.33 ANTENNA CONSIDERATIONS 189
14.34 CSMA/CA STROBE PROCESSOR 189

14.35 RADIO REGISTERS 203
15 RADIO TEST OUTPUT SIGNALS 222
16 VOLTAGE REGULATORS 223
16.1 VOLTAGE REGULATORS POWER-ON 223
17 EVALUATION SOFTWARE 223
18 REGISTER OVERVIEW 224
19 PACKAGE DESCRIPTION (QLP 48) 227
19.1 RECOMMENDED PCB LAYOUT FOR PACKAGE (QLP 48) 228
19.2 PACKAGE THERMAL PROPERTIES 228
19.3 SOLDERING INFORMATION 228
19.4 PLASTIC TUBE SPECIFICATION 228
19.5 CARRIER TAPE AND REEL SPECIFICATION 229
20 ORDERING INFORMATION 229
21 GENERAL INFORMATION 230
21.1 DOCUMENT HISTORY 230
21.2 PRODUCT STATUS DEFINITIONS 231
21.3 DISCLAIMER 232
21.4 TRADEMARKS 232
21.5 LIFE SUPPORT POLICY 232

1 Abbreviations

ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
ARIB	Association of Radio Industries and Businesses
BCD	Binary Coded Decimal
BER	Bit Error Rate
BOD	Brown Out Detector
BOM	Bill of Materials
CBC	Cipher Block Chaining
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCA	Clear Channel Assessment
CCM	Counter mode + CBC-MAC
CFB	Cipher Feedback
CFR	Code of Federal Regulations
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CSP	CSMA/CA Strobe Processor
CTR	Counter mode (encryption)
CW	Continuous Wave
DAC	Digital to Analog Converter
DC	Direct Current
DMA	Direct Memory Access
DSM	Delta Sigma Modulator
DSSS	Direct Sequence Spread Spectrum
ECB	Electronic Code Book (encryption)
EM	Evaluation Module
ESD	Electro Static Discharge
ESR	Equivalent Series Resistance
ETSI	European Telecommunications Standards Institute
EVM	Error Vector Magnitude
FCC	Federal Communications Commission
FCF	Frame Control Field
FCS	Frame Check Sequence
FFCTRL	FIFO and Frame Control
FIFO	First In First Out
HSSD	High Speed Serial Data
I/O	Input / Output
I/Q	In-phase / Quadrature-phase
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency
IOC	I/O Controller
ISM	Industrial, Scientific and Medical
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
IV	Initialization Vector
IRQ	Interrupt Request
JEDEC	Joint Electron Device Engineering Council
KB	1024 bytes
kbps	kilo bits per second
LC	Inductor-capacitor
LFSR	Linear Feedback Shift Register
LNA	Low-Noise Amplifier
LO	Local Oscillator
LQI	Link Quality Indication
LSB	Least Significant Bit / Byte
LSB	Least Significant Byte
MAC	Medium Access Control
MAC	Message Authentication Code
MCU	Microcontroller Unit
MFR	MAC Footer
MHR	MAC Header
MIC	Message Integrity Code
MISO	Master In Slave Out
MPDU	MAC Protocol Data Unit
MOSI	Master Out Slave In
MSB	Most Significant Byte
MSDU	MAC Service Data Unit
MUX	Multiplexer
NA	Not Available
NC	Not Connected
OFB	Output Feedback (encryption)
O-QPSK	Offset - Quadrature Phase Shift Keying
PA	Power Amplifier
PCB	Printed Circuit Board
PER	Packet Error Rate
PHR	PHY Header

PHY	Physical Layer
PLL	Phase Locked Loop
PM{0-3}	Power Mode 0-3
PMC	Power Management Controller
POR	Power On Reset
PSDU	PHY Service Data Unit
PWM	Pulse Width Modulator
QLP	Quad Leadless Package
RAM	Random Access Memory
RBW	Resolution Bandwidth
RC	Resistor-Capacitor
RCOSC	RC Oscillator
RF	Radio Frequency
RoHS	Restriction on Hazardous Substances
RSSI	Receive Signal Strength Indicator
RTC	Real-Time Clock
RX	Receive
SCK	Serial Clock
SFD	Start of Frame Delimiter
SFR	Special Function Register
SHR	Synchronization Header
SINAD	Signal-to-noise and distortion ratio
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
ST	Sleep Timer
T/R	Transmit / Receive
T/R	Tape and reel
TBD	To Be Decided / To Be Defined
THD	Total Harmonic Distortion
TX	Transmit
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator
VGA	Variable Gain Amplifier
WDT	Watchdog Timer
XOSC	Crystal Oscillator

2 References

[1] IEEE std. 802.15.4 - 2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)

http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf

[2] NIST FIPS Pub 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T., November 26, 2001. Available from the NIST website.

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

3 Register conventions

Each SFR register is described in a separate table. The table heading is given in the following format:

REGISTER NAME (SFR Address) - Register Description.

Each RF register is described in a separate table. The table heading is given in the following format:

REGISTER NAME (XDATA Address)

In the register descriptions, each register bit is shown with a symbol indicating the access mode of the register bit. The register values are always given in binary notation unless prefixed by '0x' which indicates hexadecimal notation.

Symbol	Access Mode
R/W	Read/write
R	Read only
R0	Read as 0
R1	Read as 1
W	Write only
W0	Write as 0
W1	Write as 1
H0	Hardware clear
H1	Hardware set

Table 1: Register bit conventions

4 Features (continued from front page)

4.1 High-Performance and Low-Power 8051-Compatible Microcontroller

- Optimized 8051 core, which typically gives 8x the performance of a standard 8051
- Dual data pointers
- In-circuit interactive debugging is supported for the IAR Embedded Workbench through a simple two-wire serial interface

4.2 Up to 128 KB Non-volatile Program Memory and 2 x 4 KB Data Memory

- 32/64/128 KB of non-volatile flash memory in-system programmable through a simple two-wire interface or by the 8051 core
- Worst-case flash memory endurance: 1000 write/erase cycles.
- Programmable read and write lock of portions of Flash memory for software security
- 4096 bytes of internal SRAM with data retention in all power modes.
- Additional 4096 bytes of internal SRAM with data retention in power modes 0 and 1.

4.3 Hardware AES Encryption/Decryption

- AES supported in hardware coprocessor

4.4 Peripheral Features

- Powerful DMA Controller
- Power On Reset/Brown-Out Detection
- Eight channel, 8-14 bit ADC
- Programmable watchdog timer
- Real time clock with 32.768 kHz crystal oscillator
- Four timers: one general 16-bit timer, two general 8-bit timers, one MAC timer
- Two programmable USARTs for master/slave SPI or UART operation
- 21 configurable general-purpose digital I/O-pins
- True random number generator

4.5 Low Power

- Four flexible power modes for reduced power consumption
- System can wake up on external interrupt or real-time counter event
- Low-power fully static CMOS design
- System clock source can be 16 MHz RC oscillator or 32 MHz crystal oscillator. The 32 MHz oscillator is used when radio is active.
- Optional clock source for ultra-low power operation can be either low-power RC oscillator or an optional 32.768 kHz crystal oscillator.

4.6 802.15.4 MAC hardware support

- Automatic preamble generator
- Synchronization word insertion/detection
- CRC-16 computation and checking over the MAC payload
- Clear Channel Assessment
- Energy detection / digital RSSI
- Link Quality Indication
- CSMA/CA Coprocessor

4.7 Integrated 2.4GHz DSSS Digital Radio

- 2.4 GHz IEEE 802.15.4 compliant RF transceiver (based on industry leading *CC2420* radio core).
- Excellent receiver sensitivity and robustness to interferers
- 250 kbps data rate, 2 MChip/s chip rate
- Complies with worldwide radio frequency regulations covered by ETSI EN 300 328 and EN 300 440 class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan).

5 Absolute Maximum Ratings

Under no circumstances must the absolute maximum ratings given in Table 2 be violated. Stress exceeding one or more of the limiting values may cause permanent damage to the device.

Parameter	Min	Max	Units	Condition
Supply voltage	–0.3	3.6	V	All supply pins must have the same voltage
Voltage on any digital pin	–0.3	VDD+0.3, max 3.6	V	
Voltage on the 1.8V pins (pin no. 22, 25-40 and 42)	–0.3	2.0	V	
Input RF level		10	dBm	
Storage temperature range	–50	150	°C	Device not programmed
Reflow soldering temperature		260	°C	According to IPC/JEDEC J-STD-020C

Table 2: Absolute Maximum Ratings



Caution! *ESD sensitive device. Precaution should be used when handling the device in order to prevent permanent damage.*

6 Operating Conditions

The operating conditions for ***CC2430*** are listed Table 3 in below.

Parameter	Min	Max	Unit	Condition
Operating ambient temperature range, T_A	-40	85	°C	
Operating supply voltage	2.0	3.6	V	The supply pins to the radio part must be driven by the on-chip regulator

Table 3: Operating Conditions

7 Electrical Specifications

T_A=25°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition
Power On Reset Voltage		1.1		V	Monitors the unregulated supply
Current Consumption					
MCU Active Mode, static		492		μA	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, dynamic		210		μA/MHz	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, highest speed		7.0		mA	MCU running at full speed (32MHz), 32MHz XOSC running. No peripherals.
MCU Active and RX Mode		27		mA	MCU running at full speed (32MHz), 32MHz XOSC running, radio in RX mode. No peripherals.
MCU Active and TX Mode, 0dBm		24.7		mA	MCU running at full speed (32MHz), 32MHz XOSC running, radio in TX mode. No peripherals.
Power mode 1		296		μA	Digital regulator on, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 2		0.9		μA	Digital regulator off, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 3		0.6		μA	No clocks. RAM retention. Power On Reset (POR) active.
Peripheral Current Consumption					Adds to the figures above if the peripheral unit is activated
Timer 1		10		μA/MHz	When enabled
Timer 2		10		μA/MHz	When enabled
Timer 3		10		μA/MHz	When enabled
Timer 4		10		μA/MHz	When enabled
Sleep Timer		0.5		μA	Including low-power RC oscillator or 32.768kHz XOSC
AES		50		μA/MHz	When encrypting/decrypting
ADC		0.9		mA	When converting
USART1 / USART2		12		μA/MHz	For each USART in use. Not including current for driving I/O pins.
DMA		30		μA/MHz	When operating, not including current for memory access
Flash write		3		mA	

Table 4: Electrical Specifications

7.1 General Characteristics

T_A=25°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Wake-Up and Timing					
Power mode 1 → power mode 0		2		µs	Digital regulator on, High Speed RCOSC and crystal oscillator off. Start-up of High Speed RCOSC.
Power mode 2 or 3 → power mode 0		54		µs	Digital regulator off, High Speed RCOSC and crystal oscillator off. Start-up of regulator and High Speed RCOSC.
Active → RX 32MHz XOSC initially OFF. Voltage regulator initially OFF		450		µs	Time from enabling radio part in power mode 0, until RX starts. Includes start-up of voltage regulator and crystal oscillator. Crystal ESR=16Ω.
Active → TX 32MHz XOSC initially OFF. Voltage regulator initially OFF		525		µs	Time from enabling radio part in power mode 0, until TX starts. Includes start-up of voltage regulator and crystal oscillator. Crystal ESR=16Ω.
Active → RX Voltage regulator initially OFF		250		µs	Time from enabling radio part in power mode 0, until RX starts. Includes start-up of voltage regulator.
Active → TX Voltage regulator initially OFF		320		µs	Time from enabling radio part in power mode 0, until TX starts. Includes start-up of voltage regulator.
Active → RX or TX			192	µs	Radio part already enabled. Time until RX or TX starts.
RX/TX turnaround			192	µs	
Radio part					
RF Frequency Range	2400		2483.5	MHz	Programmable in 1 MHz steps, 5 MHz steps for compliance with [1]
Radio bit rate		250		kbps	As defined by [1]
Radio chip rate		2.0		MChip/s	As defined by [1]

Table 5: General Characteristics

7.2 RF Receive Section

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's *CC2430* EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Receiver sensitivity		-94		dBm	PER = 1%, as specified by [1] Measured in 50 Ω single endedly through a balun. [1] requires –85 dBm
Saturation (maximum input level)	0	10		dBm	PER = 1%, as specified by [1] Measured in 50 Ω single endedly through a balun. [1] requires –20 dBm
Adjacent channel rejection + 5 MHz channel spacing		41		dB	Wanted signal 3 dB above the sensitivity level, adjacent modulated channel at +5 MHz, PER = 1 %, as specified by [1]. [1] requires 0 dB
Adjacent channel rejection - 5 MHz channel spacing		29		dB	Wanted signal 3 dB above the sensitivity level, adjacent modulated channel at -5 MHz, PER = 1 %, as specified by [1]. [1] requires 0 dB
Alternate channel rejection + 10 MHz channel spacing		54		dB	Wanted signal 3 dB above the sensitivity level, adjacent modulated channel at +10 MHz, PER = 1 %, as specified by [1] [1] requires 30 dB
Alternate channel rejection - 10 MHz channel spacing		53		dB	Wanted signal 3 dB above the sensitivity level, adjacent modulated channel at -10 MHz, PER = 1 %, as specified by [1] [1] requires 30 dB
Channel rejection ≥ + 15 MHz ≤ - 15 MHz		 53 57		 dB dB	Wanted signal @ -82 dBm. Undesired signal is an 802.15.4 modulated channel, stepped through all channels from 2405 to 2480 MHz. Signal level for PER = 1%.
Co-channel rejection		-4		dB	Wanted signal @ -82 dBm. Undesired signal is 802.15.4 modulated at the same frequency as the desired signal. Signal level for PER = 1%.

Parameter	Min	Typ	Max	Unit	Condition/Note
Blocking / Desensitization +/- 5 MHz from band edge +/- 20 MHz from band edge +/- 30 MHz from band edge +/- 50 MHz from band edge		 -29 -25 -19 -17		 dBm dBm dBm dBm	 Wanted signal 3 dB above the sensitivity level, CW jammer, PER = 1%. Measured according to EN 300 440 class 2.
Spurious emission 30 – 1000 MHz 1 – 12.75 GHz			 −57 −47	 dBm dBm	 Conducted measurement in a 50 Ω single ended load. Complies with EN 300 328, EN 300 440 class 2, FCC CFR47, Part 15 and ARIB STD-T-66
Frequency error tolerance	-300		300	kHz	Difference between centre frequency of the received RF signal and local oscillator frequency [1] requires 200 kHz
Symbol rate error tolerance		120		ppm	Difference between incoming symbol rate and the internally generated symbol rate [1] requires 80 ppm

Table 6: RF Receive Section

7.3 RF Transmit Section

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's ***CC2430*** EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Nominal output power	-3	0		dBm	Delivered to a single ended 50 Ω load through a balun. [1] requires minimum –3 dBm
Programmable output power range		24		dB	The output power is programmable in 8 steps from approximately –24 to 0 dBm.
Harmonics 2nd harmonic 3rd harmonic		 -56 -60		 dBm dBm	 Measured conducted with 1 MHz resolution bandwidth on spectrum analyser. At max output power delivered to a single ended 50 Ω load through a balun.
Spurious emission 30 - 1000 MHz 1– 12.75 GHz 1.8 – 1.9 GHz 5.15 – 5.3 GHz		 -58 -48 -58 -56		 dBm dBm dBm dBm	Maximum output power. Chipcon's ***CC2430*** EM reference design complies with EN 300 328, EN 300 440, FCC CFR47 Part 15 and ARIB STD-T-66
Error Vector Magnitude (EVM)		11		%	Measured as defined by [1] [1] requires max. 35 %
Optimum load impedance		115 + j180		Ω	Differential impedance as seen from the RF-port (RF_P and RF_N) towards the antenna.

Table 7: RF Transmit Parameters

7.4 32 MHz Crystal Oscillator

T_A=25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Crystal frequency		32		MHz	
Crystal frequency accuracy requirement	- 40		40	ppm	Including aging and temperature dependency, as specified by [1]
ESR	6	16	60	Ω	
C_0	1	1.9	7	pF	
C_L	10	13	16	pF	
Start-up time		140		µs	

Table 8: 32 MHz Crystal Oscillator Parameters

7.5 32.768 kHz Crystal Oscillator

T_A=25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Crystal frequency		32.768		kHz	
Crystal frequency accuracy requirement	–40		40	ppm	Including aging and temperature dependency, as specified by [1]
ESR		40	130	kΩ	
C_0		0.9	2.0	pF	
C_L		12	16	pF	
Start-up time			450	ms	

Table 9: 32.768 kHz Crystal Oscillator Parameters

7.6 Low Power RC Oscillator

T_A=25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Calibrated frequency		32.768		kHz	Calibrated Low Power RC Oscillator frequency is XTAL frequency multiplied by 16/15625
Frequency accuracy after calibration		±0.2		%	
Temperature coefficient		+0.4		% / °C	Frequency drift when temperature changes after calibration
Supply voltage coefficient		+3		% / V	Frequency drift when supply voltage changes after calibration
Initial calibration time		4		ms	When the Low Power RC Oscillator is enabled, calibration is continuously done in the background as long as the crystal oscillator is running.

Table 10: Low Power RC Oscillator parameters

7.7 High Speed RC Oscillator

T_A=25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Frequency		16		MHz	Calibrated High Speed RC Oscillator frequency is XTAL frequency multiplied by 1/2
Uncalibrated frequency accuracy		±18		%	Measured on Chipcon's *CC2430* EM reference design.
Calibrated frequency accuracy		±0.6	±1	%	
Start-up time			10	µs	
Temperature coefficient			-325	ppm / °C	Frequency drift when temperature changes after calibration
Supply voltage coefficient			28	ppm / mV	Frequency drift when supply voltage changes after calibration
Initial calibration time		50		µs	When the High Speed RC Oscillator is enabled, calibration is continuously done in the background as long as the crystal oscillator is running.

Table 11: High Speed RC Oscillator parameters

7.8 Frequency Synthesizer Characteristics

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's *CC2430* EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Phase noise					Unmodulated carrier
		−107		dBc/Hz	At ±1 MHz offset from carrier
		−113		dBc/Hz	At ±2 MHz offset from carrier
		−119		dBc/Hz	At ±3 MHz offset from carrier
		−121		dBc/Hz	At ±5 MHz offset from carrier
PLL lock time			192	µs	The startup time when the crystal oscillator is running and RX / TX turnaround time

Table 12: Frequency Synthesizer Parameters

7.9 Analog Temperature Sensor

T_A=25°C, VDD=2.0 V to 3.6V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Output voltage at –40°C	0.638	0.648	0.706	V	
Output voltage at 0°C	0.733	0.743	0.793	V	
Output voltage at +40°C	0.828	0.840	0.891	V	
Output voltage at +80°C	0.924	0.939	0.992	V	
Output voltage at +120°C	1.022	1.039	1.093	V	
Temperature coefficient	2.35	2.45	2.46	mV/°C	Fitted from –20°C to +80°C
Absolute error in calculated temperature	–14	–8	+14	°C	From –20°C to +80°C when assuming best fit for absolute accuracy: 0.763V at 0°C and 2.44mV / °C
Error in calculated temperature, calibrated	–2		+2	°C	From –20°C to +80°C when using 2.44mV / °C, after 1-point calibration at room temperature
Current consumption increase when enabled		0.3		mA	

Table 13: Analog Temperature Sensor Parameters

7.10 8-14 bit ADC

T_A=25°C, VDD=3.0V if nothing else stated. Differential input and differential reference.

Parameter	Min	Typ	Max	Unit	Condition/Note
Input voltage	0		AVDD	V	AVDD is voltage on AVDD_SOC pin
External reference voltage	0		AVDD	V	AVDD is voltage on AVDD_SOC pin
External reference voltage differential	0		AVDD	V	AVDD is voltage on AVDD_SOC pin
Number of bits (ENOB)	8		13	bits	The ADC is a delta-sigma. Effective resolution depends on sample rate used. Differential input signal and reference.
Offset		TBD		LSB	
Conversion time	20		132	μs	
Differential nonlinearity (DNL)		±0.3		LSB	8-bits setting
Integral nonlinearity (INL)		±0.8		LSB	8-bits setting
SINAD (sine input)		45 56 66 75		dB dB dB dB	8-bits setting 10-bits setting 12-bits setting 14-bits setting

Table 14: 8-14 bit ADC Characteristics

7.11 Control AC Characteristics

T_A= -40°C to 85°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
System clock, f_{SYSCLK} t_{SYSCLK}= 1/ f_{SYSCLK}	16		32	MHz	System clock is when 32 MHz crystal oscillator is used. System clock is 16 MHz when high speed RC oscillator is used.
RESET_N low width	2.5			ns	See item 1, Figure 1. This is the shortest pulse that is guaranteed to be recognized as a reset pin request.
Interrupt pulse width	t_{SYSCLK}			ns	See item 2, Figure 1.This is the shortest pulse that is guaranteed to be recognized as an interrupt request. In PM2/3 the internal synchronizers are bypassed so this requirement does not apply in PM2/3.

Table 15: Control Inputs AC Characteristics

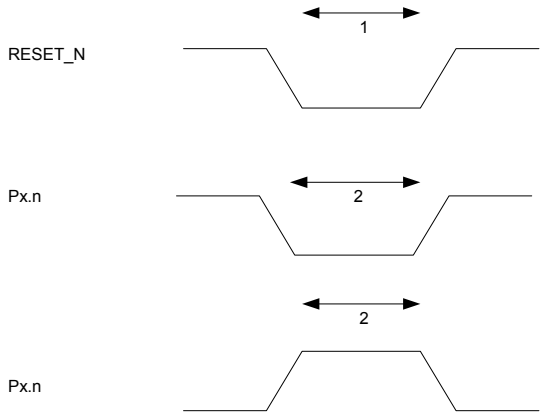


Figure 1: Control Inputs AC Characteristics

7.12 SPI AC Characteristics

T_A= -40°C to 85°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
SCK period		See section 13.13.3		ns	Master. See item 1 Figure 2
SCK duty cycle		50%			Master.
SSN low to SCK	$2*t_{SYSCLK}$				See item 5 Figure 2
SCK to SSN high	30			ns	See item 6 Figure 2
MISO setup	10			ns	Master. See item 2 Figure 2
MISO hold	10			ns	Master. See item 3 Figure 2
SCK to MOSI			25	ns	Master. See item 4 Figure 2, load = 10 pF
SCK period	100			ns	Slave. See item 1 Figure 2
SCK duty cycle		50%			Slave.
MOSI setup	10			ns	Slave. See item 2 Figure 2
MOSI hold	10			ns	Slave. See item 3 Figure 2
SCK to MISO			25	ns	Slave. See item 4 Figure 2, load = 10 pF

Table 16: SPI AC Characteristics

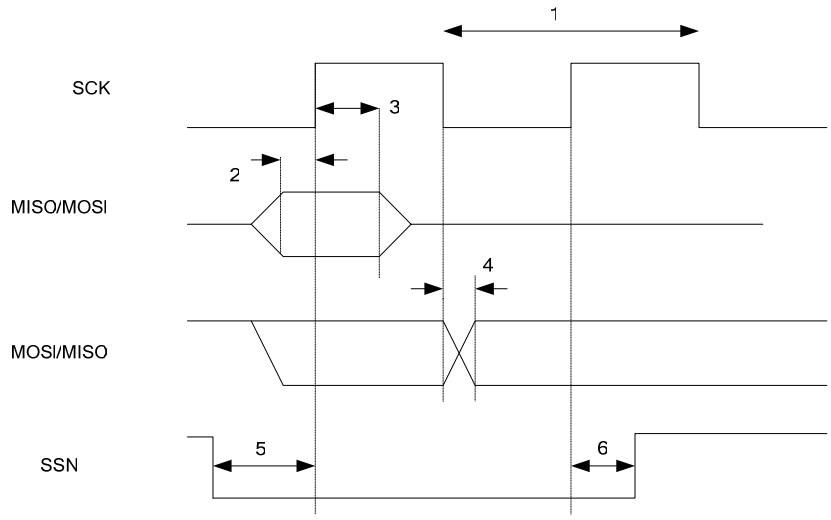


Figure 2: SPI AC Characteristics

7.13 Debug Interface AC Characteristics

T_A= -40°C to 85°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Debug clock period	31.25			ns	See item 1 Figure 3
Debug data setup	5			ns	See item 2 Figure 3
Debug data hold	5			ns	See item 3 Figure 3
Clock to data delay			10	ns	See item 4 Figure 3, load = 10 pF
RESET_N inactive after P2_2 rising	10			ns	See item 5 Figure 3

Table 17: Debug Interface AC Characteristics

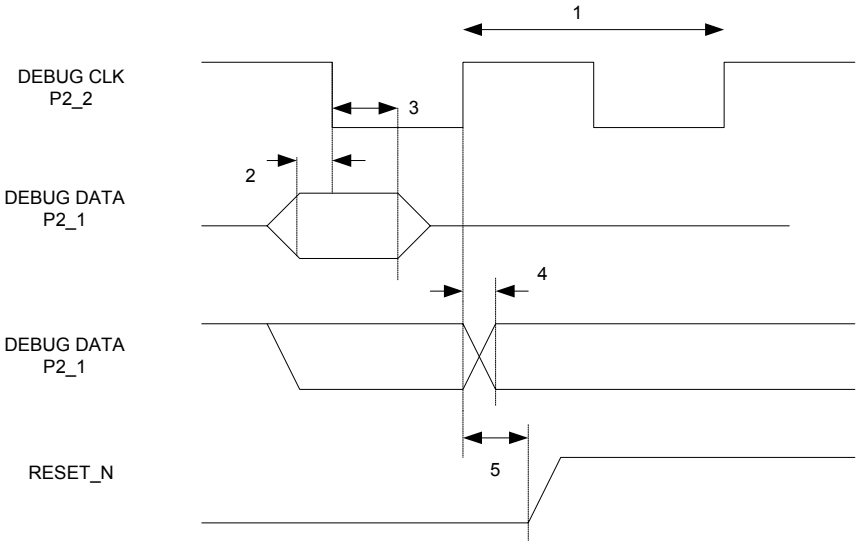


Figure 3: Debug Interface AC Characteristics

7.14 Port Outputs AC Characteristics

T_A= -40°C to 85°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
P0, P1, P2 Port output pins, rise and fall time		10		ns	Load = 10 pF Timing is with respect to 10% VDD and 90% VDD levels.

Table 18: Port Outputs AC Characteristics

7.15 Timer Inputs AC Characteristics

T_A= -40°C to 85°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Input capture pulse width	t_{SYSCLK}			ns	Synchronizers determine the shortest input pulse that can be recognized. The synchronizers operate at the current system clock rate

Table 19: Timer Inputs AC Characteristics

7.16 DC Characteristics

The DC Characteristics of ***CC2430*** are listed in Table 20 below.

T_A=25°C, VDD=3.0V if nothing else stated.

Digital Inputs/Outputs	Min	Typ	Max	Unit	Condition
Logic "0" input voltage	0	0.7	0.9	V	
Logic "1" input voltage	VDD-0.7	VDD	VDD	V	
Logic "0" output voltage	0	0	0.25	V	For up to 4mA output current on all pins except P1_0 and P1_1 which are up to 20 mA
Logic "1" output voltage	VDD-0.25	VDD	VDD	V	For up to 4mA output current on all pins except P1_0 and P1_1 which are up to 20 mA
Logic "0" input current	NA	–1	–1	μA	Input equals 0V
Logic "1" input current	NA	1	1	μA	Input equals VDD
I/O pin pull-up and pull-down resistor	17	20	23	kΩ	

Table 20: DC Characteristics

8 Pin and I/O Port Configuration

The *CC2430* pinout is shown in Figure 4 and Table 21. See section 13.1 for details on the configuration of digital I/O ports.

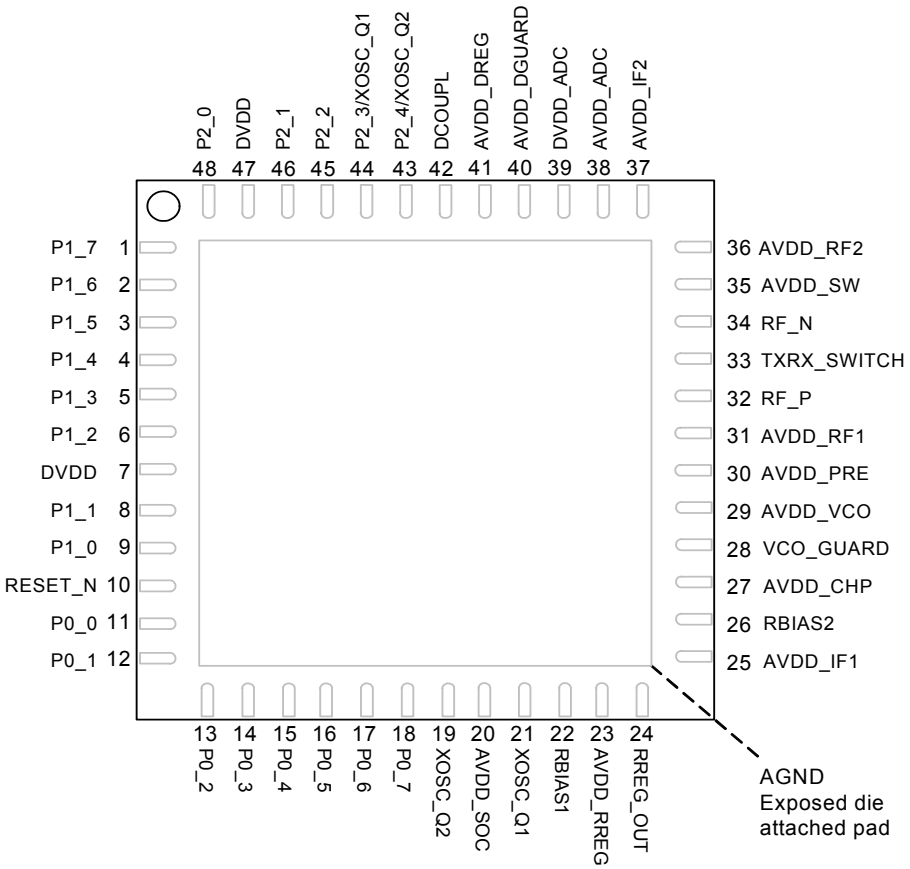


Figure 4: Pinout top view

Note: The exposed die attach pad **must** be connected to a solid ground plane as this is the ground connection for the chip.

Pin	Pin name	Pin type	Description
-	GND	Ground	The exposed die attach pad must be connected to a solid ground plane
1	P1_7	Digital I/O	Port 1.7
2	P1_6	Digital I/O	Port 1.6
3	P1_5	Digital I/O	Port 1.5
4	P1_4	Digital I/O	Port 1.4
5	P1_3	Digital I/O	Port 1.3
6	P1_2	Digital I/O	Port 1.2
7	DVDD	Power (Digital)	2.0V-3.6V digital power supply for digital I/O
8	P1_1	Digital I/O	Port 1.1 – 20 mA drive capability
9	P1_0	Digital I/O	Port 1.0 – 20 mA drive capability
10	RESET_N	Digital input	Reset, active low
11	P0_0	Digital I/O	Port 0.0
12	P0_1	Digital I/O	Port 0.1
13	P0_2	Digital I/O	Port 0.2
14	P0_3	Digital I/O	Port 0.3
15	P0_4	Digital I/O	Port 0.4
16	P0_5	Digital I/O	Port 0.5
17	P0_6	Digital I/O	Port 0.6
18	P0_7	Digital I/O	Port 0.7
19	XOSC_Q2	Analog I/O	32 MHz crystal oscillator pin 2
20	AVDD_SOC	Power (Analog)	2.0V-3.6V analog power supply connection
21	XOSC_Q1	Analog I/O	32 MHz crystal oscillator pin 1, or external clock input
22	RBIAS1	Analog I/O	External precision bias resistor for reference current
23	AVDD_RREG	Power (Analog)	2.0V-3.6V analog power supply connection
24	RREG_OUT	Power output	1.8V Voltage regulator power supply output. Only intended for supplying the analog 1.8V part (power supply for pins 25, 27-31, 35-40).
25	AVDD_IF1	Power (Analog)	1.8V Power supply for the receiver band pass filter, analog test module, global bias and first part of the VGA
26	RBIAS2	Analog output	External precision resistor, 43 kΩ, ±1 %
27	AVDD_CHP	Power (Analog)	1.8V Power supply for phase detector, charge pump and first part of loop filter
28	VCO_GUARD	Power (Analog)	Connection of guard ring for VCO (to AVDD) shielding
29	AVDD_VCO	Power (Analog)	1.8V Power supply for VCO and last part of PLL loop filter
30	AVDD_PRE	Power (Analog)	1.8V Power supply for Prescaler, Div-2 and LO buffers
31	AVDD_RF1	Power (Analog)	1.8V Power supply for LNA, front-end bias and PA
32	RF_P	RF I/O	Positive RF input signal to LNA during RX. Positive RF output signal from PA during TX
33	TXRX_SWITCH	Power (Analog)	Regulated supply voltage for PA
34	RF_N	RF I/O	Negative RF input signal to LNA during RX Negative RF output signal from PA during TX
35	AVDD_SW	Power (Analog)	1.8V Power supply for LNA / PA switch
36	AVDD_RF2	Power (Analog)	1.8V Power supply for receive and transmit mixers
37	AVDD_IF2	Power (Analog)	1.8V Power supply for transmit low pass filter and last stages of VGA
38	AVDD_ADC	Power (Analog)	1.8V Power supply for analog parts of ADCs and DACs
39	DVDD_ADC	Power (Digital)	1.8V Power supply for digital parts of ADCs
40	AVDD_DGUARD	Power (Digital)	Power supply connection for digital noise isolation
41	AVDD_DREG	Power (Digital)	2.0V-3.6V digital power supply for digital core voltage regulator
42	DCOUPL	Power (Digital)	1.8V digital power supply decoupling. Do not use for supplying external circuits.
43	P2_4/XOSC_Q2	Digital I/O	Port 2.4/32.768 kHz XOSC
44	P2_3/XOSC_Q1	Digital I/O	Port 2.3/32.768 kHz XOSC
45	P2_2	Digital I/O	Port 2.2
46	P2_1	Digital I/O	Port 2.1
47	DVDD	Power (Digital)	2.0V-3.6V digital power supply for digital I/O
48	P2_0	Digital I/O	Port 2.0

Table 21: Pinout overview

9 Circuit Description

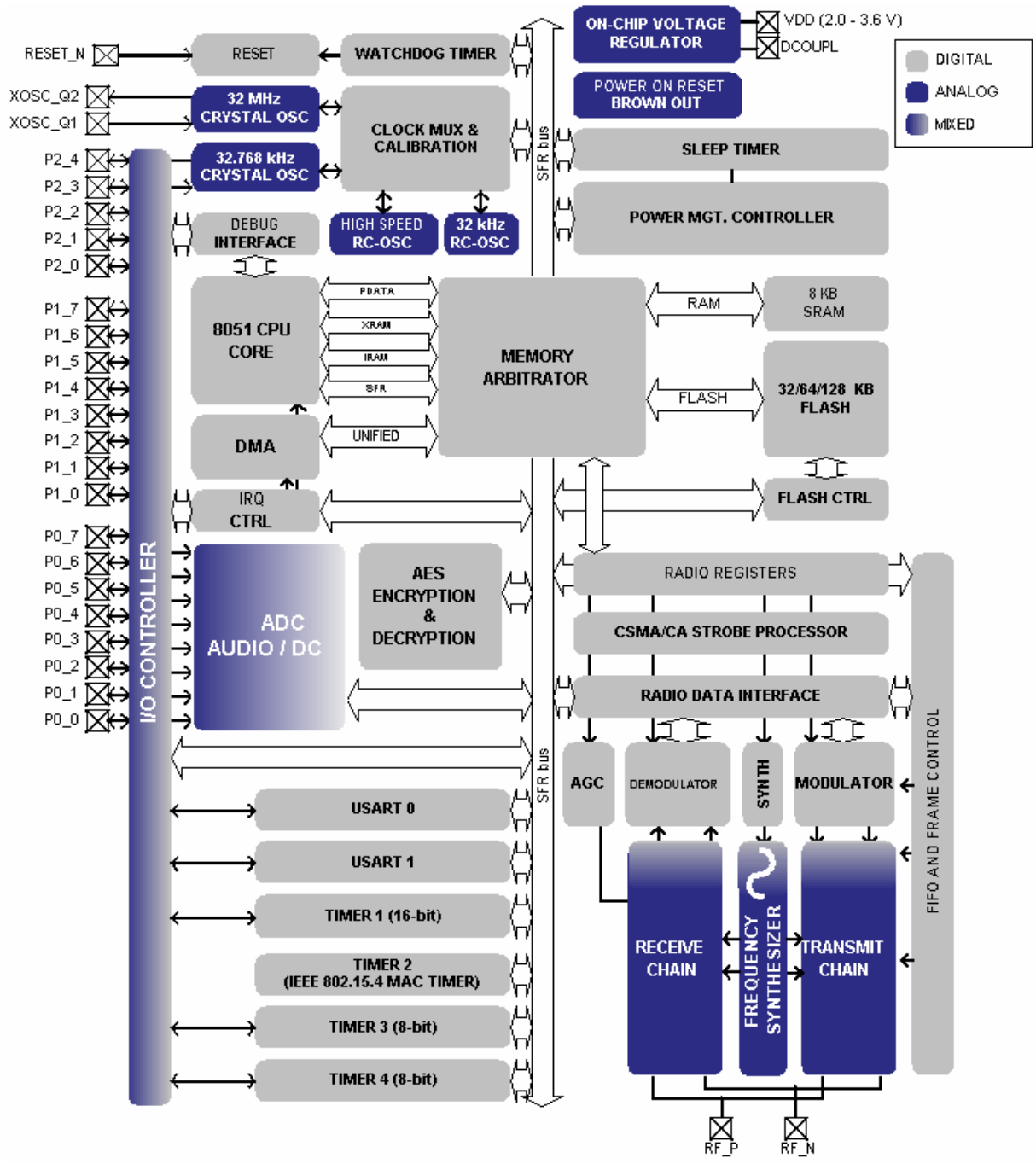


Figure 5: *CC2430* Block Diagram

A block diagram of ***CC2430*** is shown in Figure 5. The modules can be roughly divided into one of three categories: CPU-related modules, radio-related modules and modules related to power, test and clock distribution. In the following subsections, a short description of each module that appears in Figure 5 is given.

9.1 CPU and Peripherals

The **8051 CPU core** is a single-cycle 8051-compatible core. It has three different memory access buses (SFR, DATA and CODE/XDATA), a debug interface and an 18-input extended interrupt unit. See section 12 for details on the CPU.

The **memory crossbar/arbitrator** is at the heart of the system as it connects the CPU and DMA controller with the physical memories and all peripherals through the SFR bus. The memory arbitrator has four memory access points, access at which can map to one of three physical memories: an 8 KB SRAM, flash memory or RF and SFR registers. The memory arbitrator is responsible for performing arbitration and sequencing between simultaneous memory accesses to the same physical memory.

The **SFR bus** is drawn conceptually in the block diagram as a common bus that connects all hardware peripherals to the memory arbitrator. The SFR bus in the block diagram also provides access to the radio registers in the radio register bank even though these are indeed mapped into XDATA memory space.

The **8 KB SRAM** maps to the DATA memory space and to part of the XDATA memory spaces. 4 KB of the 8 KB SRAM is an ultra-low-power SRAM that retains its contents even when the digital part is powered off (power modes 2 and 3). The rest of the SRAM loses its contents when the digital part is powered off.

The **32/64/128 KB flash block** provides in-circuit programmable non-volatile program memory for the device and maps into the CODE and XDATA memory spaces. Table 22 shows the available devices in the ***CC2430*** family. The available devices differ only in flash memory size. Writing to the flash block is performed through a **flash controller** that allows page-wise (2048 byte) erasure and byte-wise reprogramming. See section 13.14 for details on the flash controller.

A versatile five-channel **DMA controller** is available in the system and accesses memory using a unified memory space (XDATA) and thus has access to all physical memories. Each channel is configured (trigger, priority, transfer mode, addressing mode, source and destination pointers, and transfer count) with DMA descriptors anywhere in memory. Many of the hardware peripherals rely on the DMA controller for efficient operation (AES core, flash write controller, USARTs, Timers, ADC interface) by performing data transfers between a single SFR address and flash/SRAM. See section 13.2 for details.

The **interrupt controller** services a total of 18 interrupt sources, divided into six *interrupt groups,* each of which is associated with one of four interrupt priorities. An interrupt request is serviced even if the device is in a sleep mode (power modes 1-3) by bringing the ***CC2430*** back to active mode (power mode 0).

The **debug interface** implements a proprietary two-wire serial interface that is used for in-circuit debugging. Through this debug interface it is possible to perform an erasure of the entire flash memory, control which oscillators are enabled, stop and start execution of the user program, execute supplied instructions on the 8051 core, set code breakpoints, and single step through instructions in the code. Using these techniques it is possible to elegantly perform in-circuit debugging and external flash programming. See section 12.9 for details.

The **I/O-controller** is responsible for all general-purpose I/O pins. The CPU can configure whether peripheral modules control certain pins or whether they are under software control, and if so whether each pin is configured as an input or output and if a pull-up or pull-down resistor in the pad is connected. Each peripheral that connects to the I/O-pins can choose between two different I/O pin locations to ensure flexibility in various applications. See section 13.1 for details.

The **sleep timer** is an ultra-low power timer that counts 32.768 kHz crystal oscillator or 32.768 kHz RC oscillator periods. The sleep timer runs continuously in all operating modes except power mode 3. Typical uses for it is as a real-time counter that runs regardless of operating mode (except power mode 3) or as a wakeup timer to get out of power mode 1 or 2. See section 13.5 for details.

A built-in **watchdog timer** allows the ***CC2430*** to reset itself in case the firmware hangs. When enabled by software, the watchdog timer must be cleared periodically, otherwise it will reset the device when it times out. See section 13.12 for details.

Timer 1 is a 16-bit timer with timer/counter/PWM functionality. It has a programmable prescaler, a 16-bit period value

and three individually programmable counter/capture channels each with a 16-bit compare value. Each of the counter/capture channels can be used as PWM outputs or to capture the timing of edges on input signals. See section 13.3 for details.

Timer 2 (MAC timer) is specially designed for supporting an IEEE 802.15.4 MAC or other time-slotted protocols in software. The timer has a configurable timer period and an 8-bit overflow counter that can be used to keep track of the number of periods that have transpired. There is also a 16-bit capture register used to record the exact time at which a start of frame delimiter is received/transmitted or the exact time of which transmission ends, as well as a 16-bit output compare register that can produce various command strobes (start RX, start TX, etc) at specific times to the radio modules. See section 13.4 for details.

Timers 3 and 4 are 8-bit timers with timer/counter/PWM functionality. They have a programmable prescaler, an 8-bit period value and one programmable counter/capture channel with a 8-bit compare value. Each of the counter/capture channels can be used as PWM outputs or to capture the timing of edges on input signals. See section 13.6 for details.

USART 0 and 1 are each configurable as either an SPI master/slave or a UART. They provide double buffering on both RX and TX and hardware flow-control and are thus well suited to high-throughput full-duplex applications. Each has its own high-precision baud-rate generator thus leaving the ordinary timers free for other uses. When configured as an SPI slave they sample the input signal using SCK directly instead of some over-sampling scheme and are thus well-suited to high data rates. See section 13.13 for details.

The **AES encryption/decryption core** allows the user to encrypt and decrypt data using the AES algorithm with 128-bit keys. The core is able to support the AES operations required by IEEE 802.15.4 MAC security, the ZigBee™ network layer and the application layer. See section 13.9 for details.

The **ADC** supports 8 to 14 bits of resolution in a 30 kHz to 4 kHz bandwidth respectively. DC and audio conversion with up to 8 input channels (Port 0) is possible. The inputs can be selected as single ended or differential. The reference voltage can be internal, AVDD, or a single ended or differential external signal. The ADC also has a temperature sensor input channel. The ADC can automate the process of periodic sampling or conversion over a sequence of channels. See Section 13.7 for details.

9.2 Radio

CC2430 features an IEEE 802.15.4 compliant radio based on the leading ***CC2420*** transceiver. See Section 14 for details.

Device	Flash
CC2430-F32	32 KB
CC2430-F64	64 KB
CC2430-F128	128 KB

Table 22: CC2430 Flash Memory Options

10 Power Management

The *CC2430* has four major power modes, called PM0, PM1, PM2 and PM3. PM0 is the active mode while PM3 has the lowest power consumption. The power modes are shown in Table 23 together with voltage regulator and oscillator options.

Power Mode	High speed oscillator	Low-speed oscillator	Voltage regulator (digital)
Configuration	A None B 32 MHz XOSC C HS RCOSC D Both	A None B 32.768 kHz RCOSC C 32.768 kHz XOSC	A Off B On
PM0	B, C, D	B, C	B
PM1	A	B, C	B
PM2	A	B, C	A
PM3	A	A	A

Table 23: Power Modes

PM0 : The full functional mode. The voltage regulator to the digital core is on and either the HS-RCOSC or the 32 MHz XOSC or both are running. Either the 32.768 kHz RCOSC or the 32.768 kHz XOSC is running.

PM1 : The voltage regulator to the digital part is on. Neither the 32 MHz XOSC nor the HS-RCOSC are running. Either the 32.768 kHz RCOSC or the 32.768 kHz XOSC is running. The system will go to PM0 on reset or an external interrupt or when the sleep timer expires.

PM2 : The voltage regulator to the digital core is turned off. Neither the 32 MHz XOSC nor the HS-RCOSC are running. Either the 32.768 kHz RCOSC or the 32.768 kHz XOSC is running. The system will go to PM0 on reset or an external interrupt or when the sleep timer expires.

PM3 : The voltage regulator to the digital core is turned off. None of the oscillators are running. The system will go to PM0 on reset or an external interrupt.

Refer to section 13.10 on page 140 for a detailed description of power management.

11 Application Circuit

Few external components are required for the operation of ***CC2430***. A typical application circuit is shown in Figure 6. Typical values and description of external components are shown in Table 24

11.1 Input / output matching

The RF input/output is high impedance and differential. The optimum differential load for the RF port is 115+j180 Ω.

When using an unbalanced antenna such as a monopole, a balun should be used in order to optimize performance. The balun can be implemented using low-cost discrete inductors and capacitors. The recommended balun shown, consists of C341, L341, L321 and L331 together with a PCB microstrip transmission line (λ/2-dipole), and will match the RF input/output to 50 Ω. An internal T/R switch circuit is used to switch between the LNA and the PA. See Input/output matching section on page 184 for more details.

If a balanced antenna such as a folded dipole is used, the balun can be omitted. If the antenna also provides a DC path from `TXRX_SWITCH` pin to the RF pins, inductors are not needed for DC bias.

Figure 6 shows a suggested application circuit using a differential antenna. The antenna type is a standard folded dipole. The dipole has a virtual ground point; hence bias is provided without degradation in antenna performance. Also refer to the section Antenna Considerations on page 189.

11.2 Bias resistors

The bias resistors are R221 and R261. The bias resistor R221 is used to set an accurate bias current for the 32 MHz crystal oscillator.

11.3 Crystal

An external 32 MHz crystal, XTAL1, with two loading capacitors (C191 and C211) is used for the 32 MHz crystal oscillator. See page 16 for details.

XTAL2 is an optional 32.768 kHz crystal. Mesh networks can be implemented without the 32.768 kHz crystal.

11.4 Voltage regulators

The on chip voltage regulators supply all 1.8 V power supply pins and internal power supplies. C241 and C421 are required for stability of the regulators. A series resistor may be used to comply with the ESR requirement.

11.5 Power supply decoupling and filtering

Proper power supply decoupling must be used for optimum performance. The placement and size of the decoupling capacitors and the power supply filtering are very important to achieve the best performance in an application. Chipcon provides a compact reference design that should be followed very closely. Refer to the section PCB Layout Recommendation on page 188.

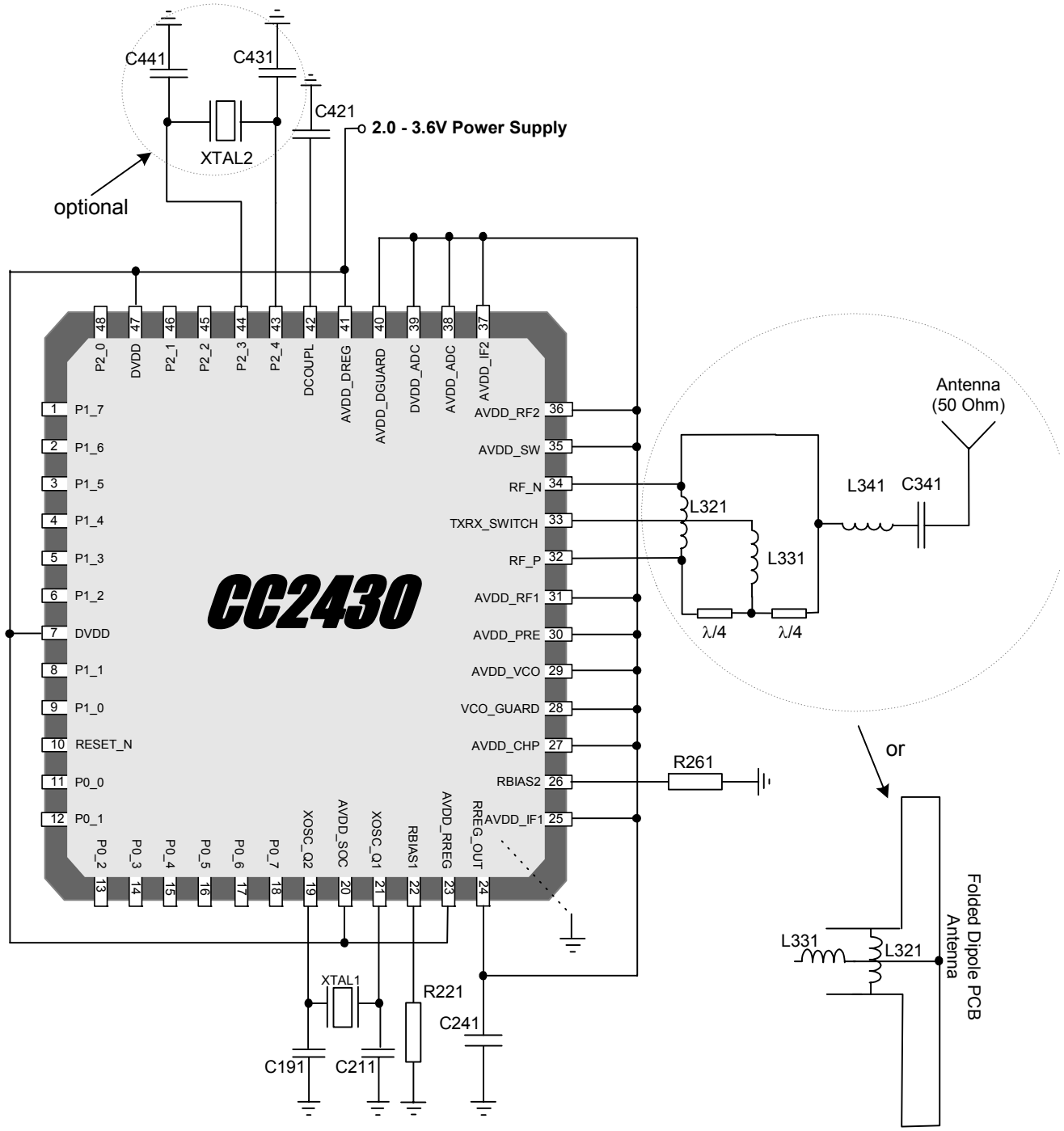


Figure 6: CC2430 Application Circuit. (Digital I/O and ADC interface not connected). Decoupling capacitors not shown.

Component	Description	Single Ended 50Ω Output	Differential Antenna
C191, C211	32 MHz crystal load capacitor	22 pF, 5%, NP0, 0402	22 pF, 5%, NP0, 0402
C241, C421	Load capacitance for power supply voltage regulators	220 nF, 10%, 0402	220 nF, 10%, 0402
C341	DC block to antenna and match	5.6 pF, +/- 0.25pF, NP0, 0402	Not used
C431, C441	32.768 kHz crystal load capacitor (if low-frequency crystal is needed in application)	15 pF, 5%, NP0, 0402	15 pF, 5%, NP0, 0402
L321	Discrete balun and match	8.2 nH, 5%, Monolithic/multilayer, 0402	27 nH, 5%, Monolithic/multilayer, 0402
L331	Discrete balun and match	22 nH, 5%, Monolithic/multilayer, 0402	12 nH, 5%, Monolithic/multilayer, 0402
L341	Discrete balun and match	1.8 nH, 5%, Monolithic/multilayer, 0402	Not used
R221	Precision resistor for current reference generator to system-on-chip part	56 kΩ, 1%, 0402	56 kΩ, 1%, 0402
R261	Precision resistor for current reference generator to RF part	43 kΩ, 1%, 0402	43 kΩ, 1%, 0402
XTAL1	32 MHz Crystal	32 MHz crystal, ESR < 60 Ω	32 MHz crystal, ESR < 60 Ω
XTAL2	Optional 32.768 kHz watch crystal (if low-frequency crystal is needed in application)	32.768 kHz crystal, Epson MC 306.	32.768 kHz crystal, Epson MC 306.

Table 24: Overview of external components (excluding supply decoupling capacitors)

12 8051 CPU

This section describes the 8051 CPU core, with interrupts, memory and instruction set.

12.1 8051 CPU Introduction

The ***CC2430*** includes an 8-bit CPU core which is an enhanced version of the industry standard 8051 core.

The enhanced 8051 core uses the standard 8051 instruction set. Instructions execute faster than the standard 8051 due to the following:

- One clock per instruction cycle is used as opposed to 12 clocks per instruction cycle in the standard 8051.
- Wasted bus states are eliminated.

Since an instruction cycle is aligned with memory fetch when possible, most of the single byte instructions are performed in a single clock cycle. In addition to the speed improvement, the enhanced 8051 core also includes architectural enhancements:

- A second data pointer.
- Extended 18-source interrupt unit

The 8051 core is object code compatible with the industry standard 8051 microcontroller. That is, object code compiled with an industry standard 8051 compiler or assembler executes on the 8051 core and is functionally equivalent. However, because the 8051 core uses a different instruction timing than many other 8051 variants, existing code with timing loops may require modification. Also because the peripheral units such as timers and serial ports differ from those on a other 8051 cores, code which includes instructions using the peripheral units SFRs will not work correctly.

12.2 Reset

The ***CC2430*** has three reset sources. The following events generate a reset:

- Forcing RESET_N input pin low
- A power-on reset condition
- Watchdog timer reset condition

The initial conditions after a reset are as follows:

- I/O pins are configured as inputs with pull-up
- CPU program counter is loaded with 0x0000 and program execution starts at this address
- All peripheral registers are initialized to their reset values (refer to register descriptions)
- Watchdog timer is disabled

12.3 Memory

The 8051 CPU has four different memory spaces:

CODE. A 16-bit read-only memory space for program memory.

DATA. An 8-bit read/write data memory space, which can be directly or indirectly accessed by a single CPU instruction. The lower 128 bytes of the DATA memory space can be addressed either directly or indirectly, the upper 128 bytes only indirectly.

XDATA. A 16-bit read/write data memory space access to which usually requires 4-5 CPU instruction cycles. Access to XDATA memory is also slower in hardware than DATA access as the CODE and XDATA memory spaces share a common bus on the CPU core and instruction pre-fetch from CODE can thus not be performed in parallel with XDATA accesses.

SFR. A 7-bit read/write register memory space which can be directly accessed by a single CPU instruction. For SFR registers whose address is divisible by eight, each bit is also individually addressable.

The four different memory spaces are distinct in the 8051 architecture, but are partly overlapping in the ***CC2430*** to ease DMA transfers and hardware debugger operation.

How the different memory spaces are mapped onto the three physical memories (flash program memory, 8 KB SRAM and hardware registers) is described in sections 12.3.1 and 12.3.2.

12.3.1 Memory Map

This section gives an overview of the memory map.

The memory map differs from the standard 8051 memory map in two important aspects, as described below.

First, in order to allow the DMA controller access to all physical memory and thus allow DMA transfers between the different 8051 memory spaces, parts of SFR and CODE memory space is mapped into the XDATA memory space.

Secondly, the CODE memory space mapping can be selected so that all physical memory is mapped to CODE space, by using a *unified mapping* of the CODE memory space.

Details about mapping of all 8051 memory spaces are given in the next section.

The memory map showing how the different physical memories are mapped into the CPU memory spaces is given in the figures on the following pages for each flash memory size option.

Note that for CODE memory space, the two possible memory maps are shown; unified and non-unified mapping.

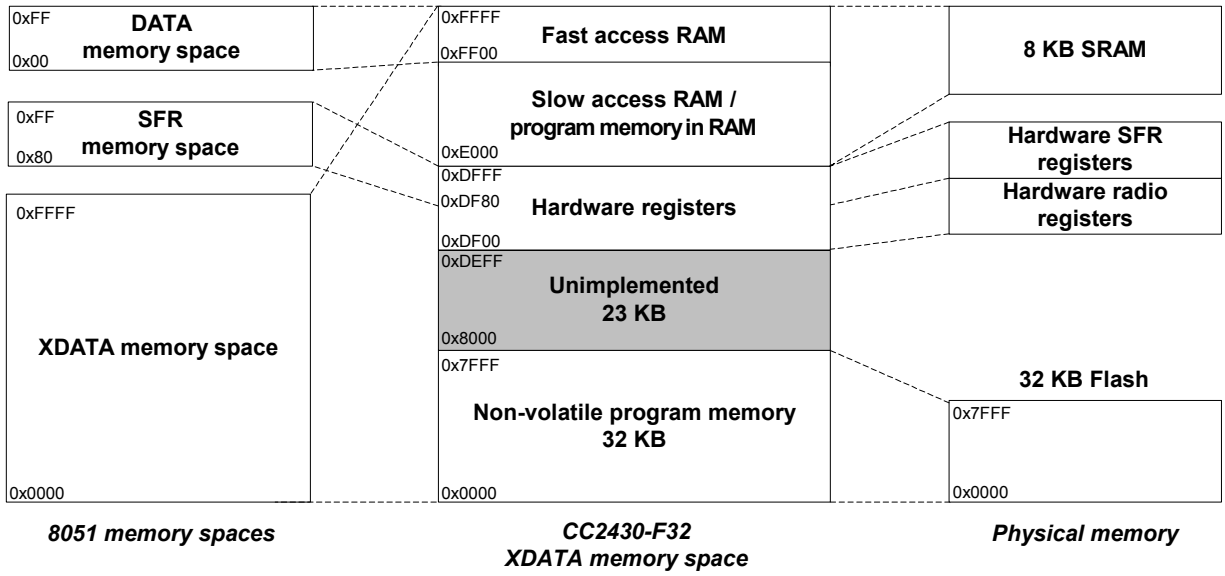


Figure 7: CC2430-F32 XDATA memory space

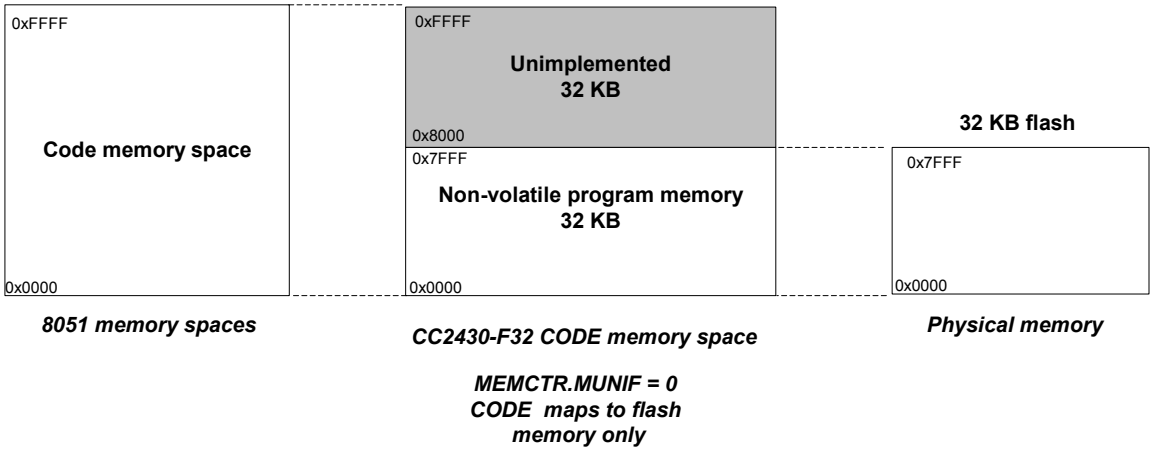


Figure 8: CC2430-F32 Non-unified mapping of CODE Space

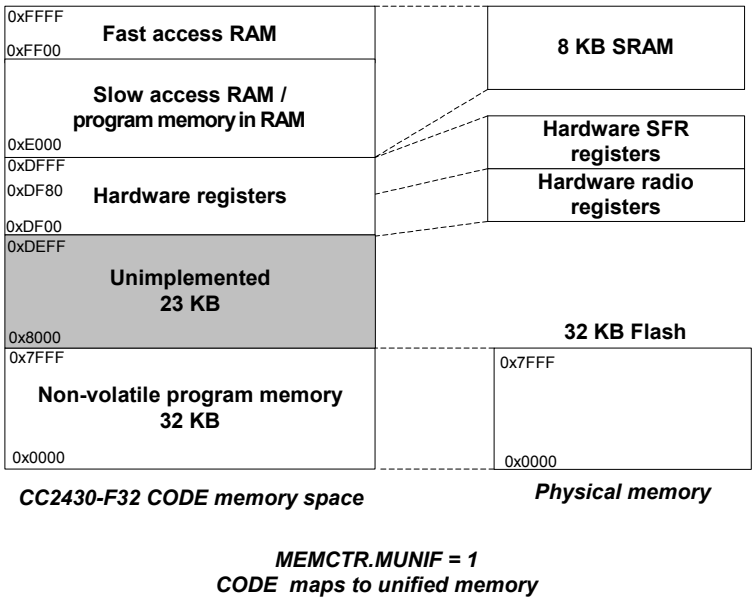


Figure 9: CC2430-F32 Unified mapping of CODE space

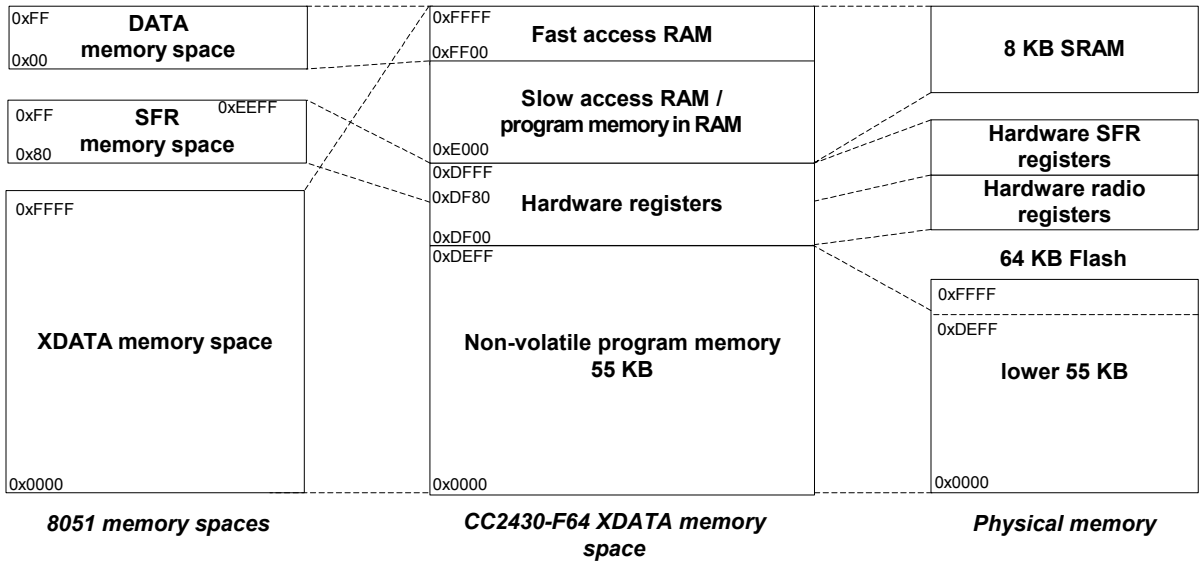


Figure 10: CC2430-F64 XDATA memory space

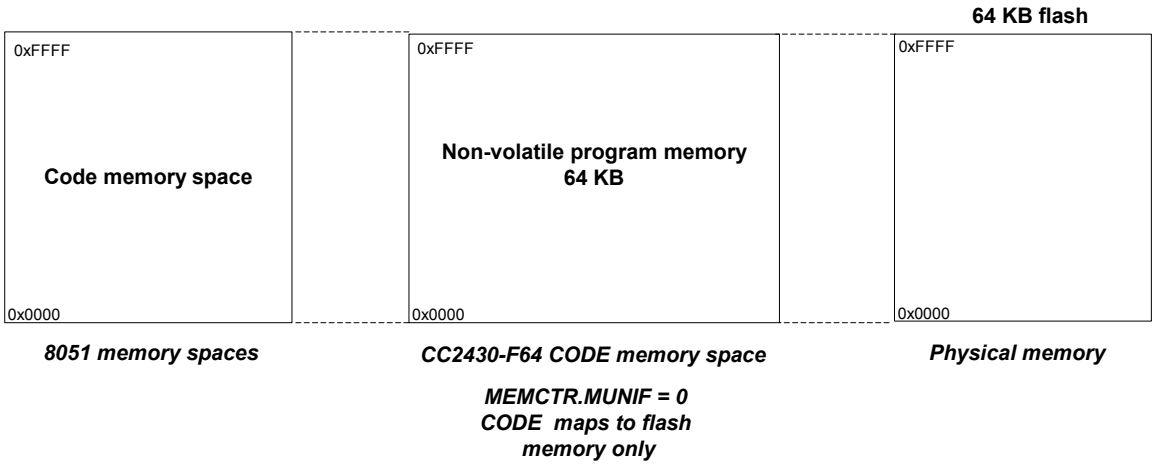


Figure 11: CC2430-F64 Non-unified mapping of CODE Space

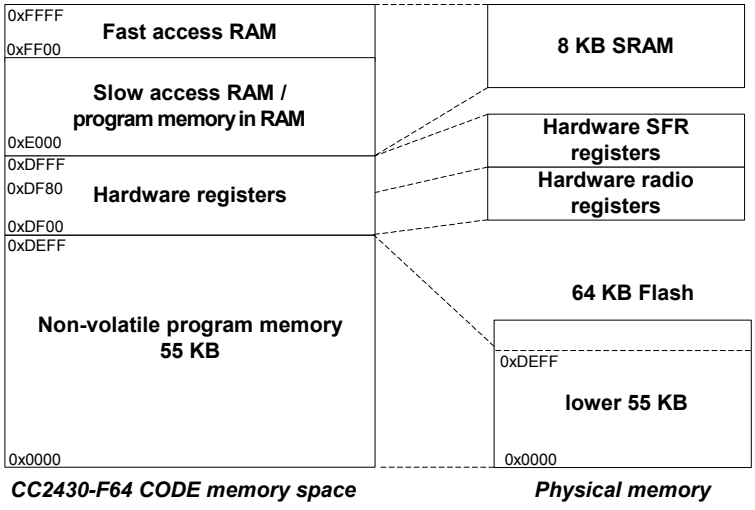


Figure 12: CC2430-F64 Unified mapping of CODE space

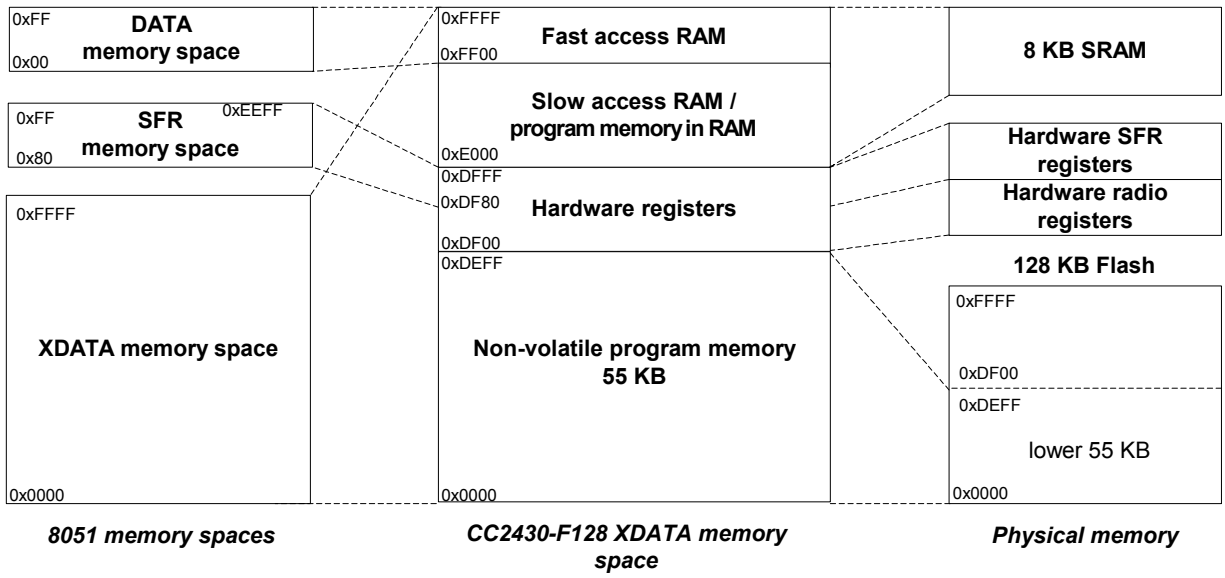


Figure 13: CC2430-F128 XDATA memory space

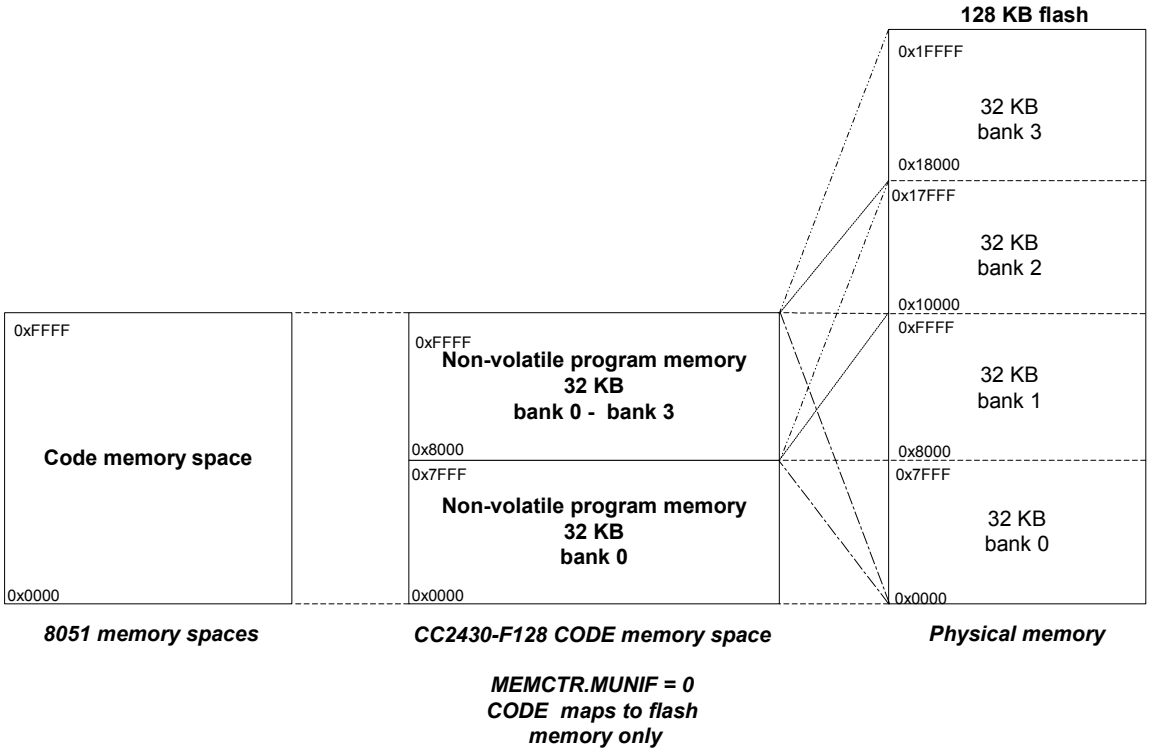


Figure 14: CC2430-F128 Non-unified mapping of CODE Space

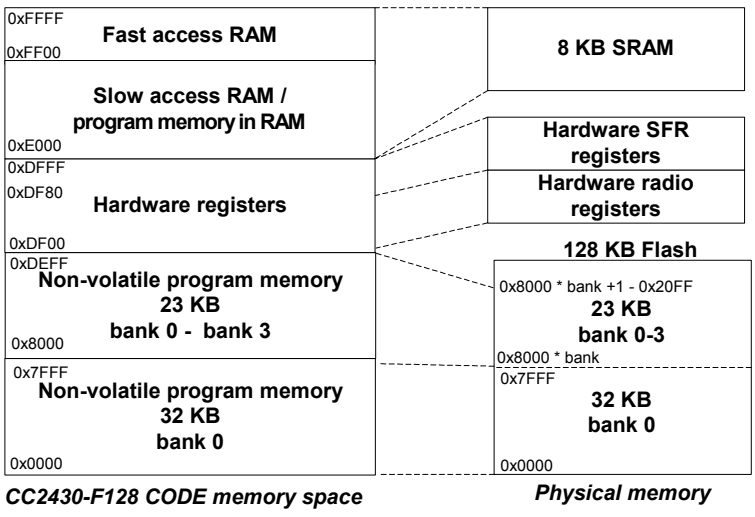


Figure 15: CC2430-F128 Unified mapping of CODE space

12.3.2 Memory Space

This section describes the details of each CPU memory space.

XDATA memory space. The XDATA memory map is given for each flash memory option in Figure 7, Figure 10 and Figure 13. For the devices with flash size above 32 KB, the lower 55 KB of the flash program memory is mapped into the address range 0x0000-0xDEFF. For the 32 KB flash size option, the 32 KB flash memory is mapped to 0x0000-0x7FFF in XDATA memory space. Access to unimplemented areas shown as shaded in the memory map gives an undefined result.

For all devices, the 8 KB SRAM is mapped into address range 0xE000-0xFFFF, and the SFR registers into address range 0xDF80-0xDFFF. This allows the DMA controller and the CPU access to all the physical memories in a single unified address space.

One of the ramifications of this mapping is that the first address of usable SRAM starts at address 0xE000 instead of 0x0000, and therefore compilers/assemblers must take this into consideration.

In low-power modes PM2-3, with the lowest power consumption, the upper 4 KB of SRAM i.e. the memory locations in XDATA address range 0xF000-0xFFFF will retain their contents. Refer to section 13.10 on page 140 for a detailed description of power modes and SRAM data retention.

CODE memory space. The CODE memory space uses either a *unified* or *non-unified* mapping to the physical memories as shown in Figure 8 - Figure 9, Figure 11 - Figure 12 and Figure 14- Figure 15. The unified mapping of the CODE memory space is similar to the XDATA mapping. Note that there is the exception that SFR registers internal to the CPU can not be accessed (see section 12.4 on page 43).

With flash memory sizes above 32 KB, the lower 55 KB of flash memory is mapped to CODE memory space when unified mapping is used. This is similar to the XDATA memory space.

The 8 KB SRAM is included in the CODE address space to allow program execution out of the SRAM.

Note: in order to use the unified memory mapping within CODE memory space, the SFR register bit `MEMCTR.MUNIF` *must be 1.*

For devices with flash memory size of 128 KB (CC2430-F128), a memory banking scheme is used for the CODE memory space. Since the physical memory size is 128 KB, the upper 32 KB area of CODE memory space is mapped to one out of the four 32 KB physical banks of flash memory through the flash bank select bits as shown in the non-unified CODE memory map. The flash bank select bits reside in the SFR register bits `MEMCTR.FMAP` (see section 12.12 on page 66). When unified CODE memory space mapping is used, the CODE memory is mapped to the flash memory in a similar manner to non-unified mapping, using memory banking, however 23 KB of the selected bank is available, as shown in the memory map.

Access to unimplemented areas shown as shaded in the memory map gives an undefined result.

DATA memory space. The 8-bit address range of DATA memory is mapped into the upper 256 bytes of the 8 KB SRAM. This area is also accessible through the CODE and XDATA memory spaces at the address range 0xFF00-0xFFFF.

SFR memory space. The 128 entry hardware register area is accessed through this memory space. The SFR registers are also accessible through the XDATA/DMA address space at the address range 0xDF80-0xDFFF. Some CPU-specific SFR registers reside inside the CPU core and can only be accessed using the SFR memory space and not through the duplicate mapping into XDATA memory space.

12.3.3 Data Pointers

The CC2430 has two data pointers, DPTR0 and DPTR1 to accelerate the movement of data blocks to/from memory. The data pointers are generally used to access CODE or XDATA space e.g.

```
MOVC A,@A+DPTR

MOV A,@DPTR.
```

The data pointer select bit, bit 0 in the Data Pointer Select register DPS, chooses which data pointer shall be the active one during execution of an instruction that uses the data pointer, e.g. in one of the above instructions.

The data pointers are two bytes wide consisting of the following SFRs:

- DPTR0 – DPH0:DPL0
- DPTR1 – DPH1:DPL1

DPH0 (0x83) – Data Pointer 0 High Byte

Bit	Name	Reset	R/W	Description
7:0	DPH0[7:0]	0	R/W	Data pointer 0, high byte

DPL0 (0x82) – Data Pointer 0 Low Byte

Bit	Name	Reset	R/W	Description
7:0	DPL0[7:0]	0	R/W	Data pointer 0, low byte

DPH1 (0x85) – Data Pointer 1 High Byte

Bit	Name	Reset	R/W	Description
7:0	DPH1[7:0]	0	R/W	Data pointer 1, high byte

DPL1 (0x84) – Data Pointer 1 Low Byte

Bit	Name	Reset	R/W	Description
7:0	DPL1[7:0]	0	R/W	Data pointer 1, low byte

DPS (0x92) – Data Pointer Select

Bit	Name	Reset	R/W	Description
7:1	-	0x00	R0	Not used
0	DPS	0	R/W	Data pointer select. Selects active data pointer. 0 : DPTR0 1 : DPTR1

12.3.4 XDATA Memory Access

The CC2430 provides an additional SFR register MPAGE. This register is used during instructions MOVX A,@Ri and MOVX @Ri,A. MPAGE gives the 8 most significant address bits, while the register Ri gives the 8 least significant bits.

In some 8051 implementations, this type of XDATA access is performed using P2 to give the most significant address bits. Existing software may therefore have to be adapted to make use of MPAGE instead of P2.

MPAGE (0x93) – Memory Page Select

Bit	Name	Reset	R/W	Description
7:0	MPAGE[7:0]	0x00	R/W	Memory page, high-order bits of address in MOVX instruction

12.4 SFR Registers

The Special Function Registers (SFRs) control several of the features of the 8051 CPU core and/or peripherals. Many of the 8051 core SFRs are identical to the standard 8051 SFRs. However, there are additional SFRs that control features that are not available in the standard 8051. The additional SFRs are used to interface with the peripheral units and RF transceiver.

Table 25 shows the address to all SFRs in ***CC2430***. The 8051 internal SFRs are shown with grey background, while the other SFRs are the SFRs specific to ***CC2430***.

Note : all internal SFRs (shown with grey background in Table 25), can only be accessed through SFR space as these registers are not mapped into XDATA space.

Table 26 lists the additional SFRs that are not standard 8051 peripheral SFRs or CPU-internal SFRs. The additional SFRs are described in the relevant sections for each peripheral function.

	8 bytes								
80	P0	SP	DPL0	DPH0	DPL1	DPH1	U0CSR	PCON	87
88	TCON	P0IFG	P1IFG	P2IFG	PICTL	P1IEN	-	P0INP	8F
90	P1	RFIM	DPS	MPAGE	T2CMP	ST0	ST1	ST2	97
98	S0CON	-	IEN2	S1CON	T2PEROF0	T2PEROF1	T2PEROF2	-	9F
A0	P2	T2OF0	T2OF1	T2OF2	T2CAPLPL	T2CAPHPH	T2TLD	T2THD	A7
A8	IEN0	IP0	-	FWT	FADDRL	FADDRH	FCTL	FWDATA	AF
B0	-	ENCDI	ENCDO	ENCCS	ADCCON1	ADCCON2	ADCCON3	-	B7
B8	IEN1	IP1	ADCL	ADCH	RNDL	RNDH	SLEEP	-	BF
C0	IRCON	U0DBUF	U0BAUD	T2CNF	U0UCR	U0GCR	CLKCON	MEMCTR	C7
C8	-	WDCTL	T3CNT	T3CTL	T3CCTL0	T3CC0	T3CCTL1	T3CC1	CF
D0	PSW	DMAIRQ	DMA1CFGL	DMA1CFGH	DMA0CFGL	DMA0CFGH	DMAARM	DMAREQ	D7
D8	TIMIF	RFD	T1CC0L	T1CC0H	T1CC1L	T1CC1H	T1CC2L	T1CC2H	DF
E0	ACC	RFST	T1CNTL	T1CNTH	T1CTL	T1CCTL0	T1CCTL1	T1CCTL2	E7
E8	IRCON2	RFIF	T4CNT	T4CTL	T4CCTL0	T4CC0	T4CCTL1	T4CC1	EF
F0	B	PERCFG	ADCCFG	P0SEL	P1SEL	P2SEL	P1INP	P2INP	F7
F8	U1CSR	U1DBUF	U1BAUD	U1UCR	U1GCR	P0DIR	P1DIR	P2DIR	FF

Table 25: SFR address overview

Table 26: CC2430 specific SFR overview

Register name	SFR Address	Module	Description
ADCCON1	0xB4	ADC	ADC Control 1
ADCCON2	0xB5	ADC	ADC Control 2
ADCCON3	0xB6	ADC	ADC Control 3
ADCL	0xBA	ADC	ADC Data Low
ADCH	0xBB	ADC	ADC Data High
RNDL	0xBC	ADC	Random Number Generator Data Low

Register name	SFR Address	Module	Description
RNDH	0xBD	ADC	Random Number Generator Data High
ENCDI	0xB1	AES	Encryption/Decryption Input Data
ENCDO	0xB2	AES	Encryption/Decryption Output Data
ENCCS	0xB3	AES	Encryption/Decryption Control and Status
DMAIRQ	0xD1	DMA	DMA Interrupt Flag
DMA1CFGL	0xD2	DMA	DMA Channel 1-4 Configuration Address Low
DMA1CFGH	0xD3	DMA	DMA Channel 1-4 Configuration Address High
DMA0CFGL	0xD4	DMA	DMA Channel 0 Configuration Address Low
DMA0CFGH	0xD5	DMA	DMA Channel 0 Configuration Address High
DMAARM	0xD6	DMA	DMA Channel Armed
DMAREQ	0xD7	DMA	DMA Channel Start Request and Status
FWT	0xAB	FLASH	Flash Write Timing
FADDRL	0xAC	FLASH	Flash Address Low
FADDRH	0xAD	FLASH	Flash Address High
FCTL	0xAE	FLASH	Flash Control
FWDATA	0xAF	FLASH	Flash Write Data
P0IFG	0x89	IOC	Port 0 Interrupt Status Flag
P1IFG	0x8A	IOC	Port 1 Interrupt Status Flag
P2IFG	0x8B	IOC	Port 2 Interrupt Status Flag
PICTL	0x8C	IOC	Port Pins Interrupt Mask and Edge
P1IEN	0x8D	IOC	Port 1 Interrupt Mask
P0INP	0x8F	IOC	Port 0 Input Mode
PERCFG	0xF1	IOC	Peripheral I/O Control
ADCCFG	0xF2	IOC	ADC Input Configuration
P0SEL	0xF3	IOC	Port 0 Function Select
P1SEL	0xF4	IOC	Port 1 Function Select
P2SEL	0xF5	IOC	Port 2 Function Select
P1INP	0xF6	IOC	Port 1 Input Mode
P2INP	0xF7	IOC	Port 2 Input Mode
P0DIR	0xFD	IOC	Port 0 Direction
P1DIR	0xFE	IOC	Port 1 Direction
P2DIR	0xFF	IOC	Port 2 Direction
MEMCTR	0xC7	MEMORY	Memory System Control
RFIM	0x91	RF	RF Interrupt Mask
RFD	0xD9	RF	RF Data
RFST	0xE1	RF	RF Command Strobe
RFIF	0xE9	RF	RF Interrupt flags
ST0	0x95	ST	Sleep Timer 0
ST1	0x96	ST	Sleep Timer 1
ST2	0x97	ST	Sleep Timer 2

Register name	SFR Address	Module	Description
SLEEP	0xBE	PMC	Sleep Mode Control
CLKCON	0xC6	PMC	Clock Control
T1CC0L	0xDA	Timer1	Timer 1 Channel 0 Capture/Compare Value Low
T1CC0H	0xDB	Timer1	Timer 1 Channel 0 Capture/Compare Value High
T1CC1L	0xDC	Timer1	Timer 1 Channel 1 Capture/Compare Value Low
T1CC1H	0xDD	Timer1	Timer 1 Channel 1 Capture/Compare Value High
T1CC2L	0xDE	Timer1	Timer 1 Channel 2 Capture/Compare Value Low
T1CC2H	0xDF	Timer1	Timer 1 Channel 2 Capture/Compare Value High
T1CNTL	0xE2	Timer1	Timer 1 Counter Low
T1CNTH	0xE3	Timer1	Timer 1 Counter High
T1CTL	0xE4	Timer1	Timer 1 Control and Status
T1CCTL0	0xE5	Timer1	Timer 1 Channel 0 Capture/Compare Control
T1CCTL1	0xE6	Timer1	Timer 1 Channel 1 Capture/Compare Control
T1CCTL2	0xE7	Timer1	Timer 1 Channel 2 Capture/Compare Control
T2CMP	0x94	Timer2	Timer 2 Compare Value
T2PEROF0	0x9C	Timer2	Timer 2 Overflow Count Compare 0
T2PEROF1	0x9D	Timer2	Timer 2 Overflow Count Compare 1
T2PEROF2	0x9E	Timer2	Timer 2 Overflow Count Compare 2
T2OF0	0xA1	Timer2	Timer 2 Overflow Count 0
T2OF1	0xA2	Timer2	Timer 2 Overflow Count 1
T2OF2	0xA3	Timer2	Timer 2 Overflow Count 2
T2CAPLPL	0xA4	Timer2	Timer 2 Timer Period Low
T2CAPHPH	0xA5	Timer2	Timer 2 Timer Period High
T2TLD	0xA6	Timer2	Timer 2 Timer Value Low
T2THD	0xA7	Timer2	Timer 2 Timer Value High
T2CNF	0xC3	Timer2	Timer 2 Configuration
T3CNT	0xCA	Timer3	Timer 3 Counter
T3CTL	0xCB	Timer3	Timer 3 Control
T3CCTL0	0xCC	Timer3	Timer 3 Channel 0 Capture/Compare Control
T3CC0	0xCD	Timer3	Timer 3 Channel 0 Capture/Compare Value
T3CCTL1	0xCE	Timer3	Timer 3 Channel 1 Capture/Compare Control
T3CC1	0xCF	Timer3	Timer 3 Channel 1 Capture/Compare Value
T4CNT	0xEA	Timer4	Timer 4 Counter
T4CTL	0xEB	Timer4	Timer 4 Control
T4CCTL0	0xEC	Timer4	Timer 4 Channel 0 Capture/Compare Control
T4CC0	0xED	Timer4	Timer 4 Channel 0 Capture/Compare Value
T4CCTL1	0xEE	Timer4	Timer 4 Channel 1 Capture/Compare Control
T4CC1	0xEF	Timer4	Timer 4 Channel 1 Capture/Compare Value
TIMIF	0xD8	TMINT	Timers 1/3/4 Joint Interrupt Mask/Flags
U0CSR	0x86	USART0	USART 0 Control and Status

Register name	SFR Address	Module	Description
U0DBUF	0xC1	USART0	USART 0 Receive/Transmit Data Buffer
U0BAUD	0xC2	USART0	USART 0 Baud Rate Control
U0UCR	0xC4	USART0	USART 0 UART Control
U0GCR	0xC5	USART0	USART 0 Generic Control
U1CSR	0xF8	USART1	USART 1 Control and Status
U1DBUF	0xF9	USART1	USART 1 Receive/Transmit Data Buffer
U1BAUD	0xFA	USART1	USART 1 Baud Rate Control
U1UCR	0xFB	USART1	USART 1 UART Control
U1GCR	0xFC	USART1	USART 1 Generic Control
WDCTL	0xC9	WDT	Watchdog Timer Control

12.5 CPU Registers

This section describes the internal registers found in the CPU.

12.5.1 Registers R0-R7

The ***CC2430*** provides four register banks of eight registers each. These register banks are mapped in the DATA memory space at addresses 0x00-0x07, 0x08-0x0F, 0x10-0x17 and 0x18-0x1F. Each register bank contains the eight 8-bit register `R0-R7`. The register bank to be used is selected through the Program Status Word `PSW.RS[1:0]`.

12.5.2 Program Status Word

The Program Status Word (PSW) contains several bits that show the current state of the CPU. The Program Status Word is accessible as an SFR and it is bit-addressable. `PSW` is shown below and contains the Carry flag, Auxiliary Carry flag for BCD operations, Register Select bits, Overflow flag and Parity flag. Two bits in `PSW` are uncommitted and can be used as user-defined status flags

.

PSW (0xD0) – Program Status Word

Bit	Name	Reset	R/W	Description
7	CY	0	R/W	Carry flag. Set to 1 when the last arithmetic operation resulted in a carry (during addition) or borrow (during subtraction), otherwise cleared to 0 by all arithmetic operations.
6	AC	0	R/W	Auxiliary carry flag for BCD operations. Set to 1 when the last arithmetic operation resulted in a carry into (during addition) or borrow from (during subtraction) the high order nibble, otherwise cleared to 0 by all arithmetic operations.
5	F0	0	R/W	User-defined, bit-addressable
4:3	RS[1:0]	00	R/W	Register bank select bits. Selects which set of R7-R0 registers to use from four possible register banks in DATA space. 00 Bank 0, 0x00 – 0x07 01 Bank 1, 0x08 – 0x0F 10 Bank 2, 0x10 – 0x17 11 Bank 3, 0x18 – 0x1F
2	OV	0	R/W	Overflow flag, set by arithmetic operations. Set to 1 when the last arithmetic operation resulted in a carry (addition), borrow (subtraction), or overflow (multiply or divide). Otherwise, the bit is cleared to 0 by all arithmetic operations.
1	F1	0	R/W	User-defined, bit-addressable
0	P	0	R/W	Parity flag, parity of accumulator set by hardware to 1 if it contains an odd number of 1's, otherwise it is cleared to 0

12.5.3 Accumulator

ACC is the accumulator. This is the source and destination of most arithmetic instructions, data transfers and other instructions. The mnemonic for the accumulator (in instructions involving the accumulator) refers to A instead of ACC.

ACC (0xE0) – Accumulator

Bit	Name	Reset	R/W	Description
7:0	ACC[7:0]	0x00	R/W	Accumulator

12.5.4 B Register

The B register is used as the second 8-bit argument during execution of multiply and divide instructions. When not used for these purposes it may be used as a scratch-pad register to hold temporary data.

B (0xF0) – B Register

Bit	Name	Reset	R/W	Description
7:0	B[7:0]	0x00	R/W	B register. Used in MUL/DIV instructions.

12.5.5 Stack Pointer

The stack resides in DATA memory space and grows upwards. The PUSH instruction first increments the Stack Pointer (SP) and then copies the byte into the stack. The Stack Pointer is initialized to 0x07 after a reset and it is incremented once to start from location 0x08 which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location not used for data storage.

SP (0x81) – Stack Pointer

Bit	Name	Reset	R/W	Description
7:0	SP[7:0]	0x07	R/W	Stack Pointer

12.6 Instruction Set Summary

The 8051 instruction set is summarized in Table 27. All mnemonics copyrighted © Intel Corporation, 1980.

The following conventions are used in the instruction set summary:

- Rn – Register R7-R0 of the currently selected register bank.
- direct – 8-bit internal data location's address. This can be DATA area (0x00 – 0x7F) or SFR area (0x80 – 0xFF).
- @Ri – 8-bit internal data location, DATA area (0x00 – 0xFF) addressed indirectly through register R1 or R0.
- #data – 8-bit constant included in instruction.
- #data16 – 16-bit constant included in instruction.
- addr16 – 16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64 KB CODE memory space.
- addr11 – 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 KB page of program memory as the first byte of the following instruction.
- rel – Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
- bit – direct addressed bit in DATA area or SFR.

The instructions that affect CPU flag settings located in PSW are listed in Table 28 on page 52. Note that operations on the PSW register or bits in PSW will also affect the flag settings.

Table 27: Instruction Set Summary

Mnemonic	Description	Hex Opcode	Bytes	Cycles
Arithmetic operations				
ADD A,Rn	Add register to accumulator	28-2F	1	1
ADD A,direct	Add direct byte to accumulator	25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	26-27	1	2
ADD A,#data	Add immediate data to accumulator	24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	38-3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	36-37	1	2
ADDC A,#data	Add immediate data to A with carry flag	34	2	2
SUBB A,Rn	Subtract register from A with borrow	98-9F	1	1
SUBB A,direct	Subtract direct byte from A with borrow	95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	96-97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	94	2	2
INC A	Increment accumulator	04	1	1
INC Rn	Increment register	08-0F	1	2
INC direct	Increment direct byte	05	2	3
INC @Ri	Increment indirect RAM	06-07	1	3
INC DPTR	Increment data pointer	A3	1	1
DEC A	Decrement accumulator	14	1	1
DEC Rn	Decrement register	18-1F	1	2
DEC direct	Decrement direct byte	15	2	3
DEC @Ri	Decrement indirect RAM	16-17	1	3
MUL AB	Multiply A and B	A4	1	5
DIV	Divide A by B	84	1	5
DA A	Decimal adjust accumulator	D4	1	1
Logical operations				
ANL A,Rn	AND register to accumulator	58-5F	1	1
ANL A,direct	AND direct byte to accumulator	55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	56-57	1	2
ANL A,#data	AND immediate data to accumulator	54	2	2
ANL direct,A	AND accumulator to direct byte	52	2	3
ANL direct,#data	AND immediate data to direct byte	53	3	4
ORL A,Rn	OR register to accumulator	48-4F	1	1
ORL A,direct	OR direct byte to accumulator	45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	46-47	1	2
ORL A,#data	OR immediate data to accumulator	44	2	2
ORL direct,A	OR accumulator to direct byte	42	2	3

Mnemonic	Description	Hex Opcode	Bytes	Cycles
ORL direct,#data	OR immediate data to direct byte	43	3	4
XRL A,Rn	Exclusive OR register to accumulator	68-6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	66-67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	63	3	4
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate accumulator left	23	1	1
RLC A	Rotate accumulator left through carry	33	1	1
RR A	Rotate accumulator right	03	1	1
RRC A	Rotate accumulator right through carry	13	1	1
SWAP A	Swap nibbles within the accumulator	C4	1	1
Data transfers				
MOV A,Rn	Move register to accumulator	E8-EF	1	1
MOV A,direct	Move direct byte to accumulator	E5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	E6-E7	1	2
MOV A,#data	Move immediate data to accumulator	74	2	2
MOV Rn,A	Move accumulator to register	F8-FF	1	2
MOV Rn,direct	Move direct byte to register	A8-AF	2	4
MOV Rn,#data	Move immediate data to register	78-7F	2	2
MOV direct,A	Move accumulator to direct byte	F5	2	3
MOV direct,Rn	Move register to direct byte	88-8F	2	3
MOV direct1,direct2	Move direct byte to direct byte	85	3	4
MOV direct,@Ri	Move indirect RAM to direct byte	86-87	2	4
MOV direct,#data	Move immediate data to direct byte	75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	F6-F7	1	3
MOV @Ri,direct	Move direct byte to indirect RAM	A6-A7	2	5
MOV @Ri,#data	Move immediate data to indirect RAM	76-77	2	3
MOV DPTR,#data16	Load data pointer with a 16-bit constant	90	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOVC A,@A+PC	Move code byte relative to PC to accumulator	83	1	3
MOVX A,@Ri	Move external RAM (8-bit address) to A	E2-E3	1	3-10
MOVX A,@DPTR	Move external RAM (16-bit address) to A	E0	1	3-10
MOVX @Ri,A	Move A to external RAM (8-bit address)	F2-F3	1	4-11
MOVX @DPTR,A	Move A to external RAM (16-bit address)	F0	1	4-11
PUSH direct	Push direct byte onto stack	C0	2	4
POP direct	Pop direct byte from stack	D0	2	3
XCH A,Rn	Exchange register with accumulator	C8-CF	1	2

Mnemonic	Description	Hex Opcode	Bytes	Cycles
XCH A,direct	Exchange direct byte with accumulator	C5	2	3
XCH A,@Ri	Exchange indirect RAM with accumulator	C6-C7	1	3
XCHD A,@Ri	Exchange low-order nibble indirect. RAM with A	D6-D7	1	3
Program branching				
ACALL addr11	Absolute subroutine call	xxx11	2	6
LCALL addr16	Long subroutine call	12	3	6
RET	Return from subroutine	22	1	4
RETI	Return from interrupt	32	1	4
AJMP addr11	Absolute jump	xxx01	2	3
LJMP addr16	Long jump	02	3	4
SJMP rel	Short jump (relative address)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
JZ rel	Jump if accumulator is zero	60	2	3
JNZ rel	Jump if accumulator is not zero	70	2	3
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit,rel	Jump if direct bit is set	20	3	4
JNB bit,rel	Jump if direct bit is not set	30	3	4
JBC bit,direct rel	Jump if direct bit is set and clear bit	10	3	4
CJNE A,direct rel	Compare direct byte to A and jump if not equal	B5	3	4
CJNE A,#data rel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn,#data rel	Compare immediate to reg. and jump if not equal	B8-BF	3	4
CJNE @Ri,#data rel	Compare immediate to indirect and jump if not equal	B6-B7	3	4
DJNZ Rn,rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct,rel	Decrement direct byte and jump if not zero	D5	3	4
NOP	No operation	00	1	1
Boolean variable operations				
CLR C	Clear carry flag	C3	1	1
CLR bit	Clear direct bit	C2	2	3
SETB C	Set carry flag	D3	1	1
SETB bit	Set direct bit	D2	2	3
CPL C	Complement carry flag	B3	1	1
CPL bit	Complement direct bit	B2	2	3
ANL C,bit	AND direct bit to carry flag	82	2	2
ANL C,/bit	AND complement of direct bit to carry	B0	2	2
ORL C,bit	OR direct bit to carry flag	72	2	2
ORL C,/bit	OR complement of direct bit to carry	A0	2	2
MOV C,bit	Move direct bit to carry flag	A2	2	2
MOV bit,C	Move carry flag to direct bit	92	2	3

Instruction	CY	OV	AC
ADD	x	x	x
ADDC	x	x	x
SUBB	x	x	x
MUL	0	x	-
DIV	0	x	-
DA	x	-	-
RRC	x	-	-
RLC	x	-	-
SETB C	1	-	-
CLR C	x	-	-
CPL C	x	-	-
ANL C,bit	x	-	-
ANL C,/bit	x	-	-
ORL C,bit	x	-	-
ORL C,/bit	x	-	-
MOV C,bit	x	-	-
CJNE	x	-	-

"0"=set to 0, "1"=set to 1, "x"=set to 0/1, "-"=not affected

Table 28: Instructions that affect flag settings

12.7 Interrupts

The CPU has 18 interrupt sources. Each source has its own request flag located in a set of Interrupt Flag SFR registers. Each interrupt requested by the corresponding flag can be individually enabled or disabled. The definitions of the interrupt sources and the interrupt vectors are given in Table 29.

The interrupts are grouped into a set of priority level groups with selectable priority levels.

The interrupt enable registers are described in section 12.7.1 and the interrupt priority settings are described in section 12.7.3 on page 60.

12.7.1 Interrupt Masking

Each interrupt can be individually enabled or disabled by the interrupt enable bits in the Interrupt Enable SFRs IEN0, IEN1 and IEN2. The Interrupt Enable SFRs are described below and summarized in Table 29.

Note that some peripherals have several events that can generate the interrupt request associated with that peripheral. This applies to Port 0, Port 1, Port 2, DMA, Timer 1, Timer 3 , Timer 4 and Radio. These peripherals have interrupt mask bits for each internal interrupt source in the corresponding SFR registers.

In order to use any of the interrupts in the ***CC2430***, the following steps must be taken

1. Enable global interrupt by setting the EA bit in IEN0 to 1
2. Set the corresponding individual, interrupt enable bit in the IEN0, IEN1 or IEN2 registers to 1.
3. Set individual interrupt enable bit in the peripherals SFR register, if any.
4. Begin the interrupt service routine at the corresponding vector address of that interrupt. See Table 29 for addresses.

Interrupt number	Description	Interrupt name	Interrupt Vector	Interrupt Mask	Interrupt Flag
0	RF TX FIFO underflow and RX FIFO overflow.	RFERR	03h	IEN0.RFERRIE	TCON.RFERRIF
1	ADC end of conversion	ADC	0Bh	IEN0.ADCIE	TCON.ADCIF
2	USART0 RX complete	URX0	13h	IEN0.URX0IE	TCON.URX0IF
3	USART1 RX complete	URX1	1Bh	IEN0.URX1IE	TCON.URX1IF
4	AES encryption/decryption complete	ENC	23h	IEN0.ENCIE	S0CON.ENCIF
5	Sleep Timer compare	ST	2Bh	IEN0.STIE	IRCON.STIF
6	Port 2 inputs	P2INT	33h	IEN2.P2IE	IRCON2.P2IF
7	USART0 TX complete	UTX0	3Bh	IEN2.UTX0IE	IRCON2.UTX0IF
8	DMA transfer complete	DMA	43h	IEN1.DMAIE	IRCON.DMAIF
9	Timer 1 (16-bit) capture/compare/overflow	T1	4Bh	IEN1.T1IE	IRCON.T1IF
10	Timer 2 (MAC Timer)	T2	53h	IEN1.T2IE	IRCON.T2IF
11	Timer 3 (8-bit) capture/compare/overflow	T3	5Bh	IEN1.T3IE	IRCON.T3IF
12	Timer 4 (8-bit) capture/compare/overflow	T4	63h	IEN1.T4IE	IRCON.T4IF
13	Port 0 inputs	P0INT	6Bh	IEN1.P0IE	IRCON.P0IF
14	USART1 TX complete	UTX1	73h	IEN2.UTX1IE	IRCON2.UTX1IF
15	Port 1 inputs	P1INT	7Bh	IEN2.P1IE	IRCON2.P1IF
16	RF general interrupts	RF	83h	IEN2.RFIE	S1CON.RFIF
17	Watchdog overflow in timer mode	WDT	8Bh	IEN2.WDTIE	IRCON2.WDTIF

Table 29: Interrupts Overview

IEN0 (0xA8) – Interrupt Enable 0

Bit	Name	Reset	R/W	Description
7	EA	0	R/W	Disables all interrupts. 0 No interrupt will be acknowledged 1 Each interrupt source is individually enabled or disabled by setting its corresponding enable bit
6	–	0	R0	Not used. Read as 0
5	STIE	0	R/W	STIE – Sleep Timer interrupt enable 0 Interrupt disabled 1 Interrupt enabled
4	ENCIE	0	R/W	ENCIE – AES encryption/decryption interrupt enable 0 Interrupt disabled 1 Interrupt enabled
3	URX1IE	0	R/W	URX1IE – USART1 RX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
2	URX0IE	0	R/W	URX0IE - USART0 RX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
1	ADCIE	0	R/W	ADCIE – ADC interrupt enable 0 Interrupt disabled 1 Interrupt enabled
0	RFERRIE	0	R/W	RFERRIE – RF TX/RX FIFO interrupt enable 0 Interrupt disabled 1 Interrupt enabled

IEN1 (0xB8) – Interrupt Enable 1

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used. Read as 0
5	P0IE	0	R/W	P0IE – Port 0 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
4	T4IE	0	R/W	T4IE - Timer 4 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
3	T3IE	0	R/W	T3IE - Timer 3 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
2	T2IE	0	R/W	T2IE – Timer 2 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
1	T1IE	0	R/W	T1IE – Timer 1 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
0	DMAIE	0	R/W	DMAIE – DMA transfer interrupt enable 0 Interrupt disabled 1 Interrupt enabled

IEN2 (0x9A) – Interrupt Enable 2

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used. Read as 0
5	WDTIE	0	R/W	WDTIE – Watchdog timer interrupt enable 0 Interrupt disabled 1 Interrupt enabled
4	P1IE	0	R/W	P1IE– Port 1 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
3	UTX1IE	0	R/W	UTX1IE – USART1 TX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
2	UTX0IE	0	R/W	UTX0IE - USART0 TX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
1	P2IE	0	R/W	P2IE – Port 2 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
0	RFIE	0	R/W	RFIE – RF general interrupt enable 0 Interrupt disabled 1 Interrupt enabled

12.7.2 Interrupt Processing

When an interrupt occurs, the CPU will vector to the interrupt vector address as shown in Table 29. Once an interrupt service has begun, it can be interrupted only by a higher priority interrupt. The interrupt service is terminated by a RETI (return from interrupt instruction). When an RETI is performed, the CPU will return to the instruction that would have been next when the interrupt occurred.

When the interrupt condition occurs, the CPU will also indicate this by setting an interrupt flag bit in the interrupt flag registers. This bit is set regardless of whether the interrupt is enabled or disabled. If the interrupt is enabled when an interrupt flag is set, then on the next instruction cycle the interrupt will be acknowledged by hardware forcing an LCALL to the appropriate vector address.

Interrupt response will require a varying amount of time depending on the state of the CPU when the interrupt occurs. If the CPU is performing an interrupt service with equal or greater priority, the new interrupt will be pending until it becomes the interrupt with highest priority. In other cases, the response time depends on current instruction. The fastest possible response to an interrupt is seven machine cycles. This includes one machine cycle for detecting the interrupt and six cycles to perform the LCALL.

TCON (0x88) – Interrupt Flags

Bit	Name	Reset	R/W	Description
7	URX1IF	0	R/W H0	URX1IF – USART1 RX interrupt flag. Set to 1 when USART1 RX interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
6	–	0	R/W	Not used
5	ADCIF	0	R/W H0	ADCIF – ADC interrupt flag. Set to 1 when ADC interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
4	–	0	R/W	Not used
3	URX0IF	0	R/W H0	URX0IF – USART0 RX interrupt flag. Set to 1 when USART0 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
2	IT1	1	R/W	Reserved. Must always be set to 1.
1	RFERRIF	0	R/W H0	RFERR – RF TX/RX FIFO interrupt flag. Set to 1 when RFERR interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
0	IT0	1	R/W	Reserved. Must always be set to 1.

S0CON (0x98) – Interrupt Flags 2

Bit	Name	Reset	R/W	Description
7:2	–	0x00	R/W	Not used
1	ENCIF_1	0	R/W	ENCIF – AES interrupt. ENC has two interrupt flags, ENCIF_1 and ENCIF_0. Setting one of these flags will request interrupt service. Both flags are set when the AES co-processor requests the interrupt. 0 Interrupt not pending 1 Interrupt pending
0	ENCIF_0	0	R/W	ENCIF – AES interrupt. ENC has two interrupt flags, ENCIF_1 and ENCIF_0. Setting one of these flags will request interrupt service. Both flags are set when the AES co-processor requests the interrupt. 0 Interrupt not pending 1 Interrupt pending

S1CON (0x9B) – Interrupt Flags 3

Bit	Name	Reset	R/W	Description
7:2	–	0x00	R/W	Not used
1	RFIF_1	0	R/W	RFIF – RF general interrupt. RF has two interrupt flags, RFIF_1 and RFIF_0. Setting one of these flags will request interrupt service. Both flags are set when the radio requests the interrupt. 0 Interrupt not pending 1 Interrupt pending
0	RFIF_0	0	R/W	RFIF – RF general interrupt. RF has two interrupt flags, RFIF_1 and RFIF_0. Setting one of these flags will request interrupt service. Both flags are set when the radio requests the interrupt. 0 Interrupt not pending 1 Interrupt pending

IRCON (0xC0) – Interrupt Flags 4

Bit	Name	Reset	R/W	Description
7	STIF	0	R/W	STIF – Sleep timer interrupt flag 0 Interrupt not pending 1 Interrupt pending
6	–	0	R/W	Not used
5	P0IF	0	R/W	P0IF – Port 0 interrupt flag 0 Interrupt not pending 1 Interrupt pending
4	T4IF	0	R/W H0	T4IF – Timer 4 interrupt flag. Set to 1 when Timer 4 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
3	T3IF	0	R/W H0	T3IF – Timer 3 interrupt flag. Set to 1 when Timer 3 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
2	T2IF	0	R/W H0	T2IF – Timer 2 interrupt flag. Set to 1 when Timer 2 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
1	T1IF	0	R/W H0	T1IF – Timer 1 interrupt flag. Set to 1 when Timer 1 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
0	DMAIF	0	R/W	DMAIF – DMA complete interrupt flag. 0 Interrupt not pending 1 Interrupt pending

IRCON2 (0xE8) – Interrupt Flags 5

Bit	Name	Reset	R/W	Description
7:5	–	00	R/W	Not used
4	WDTIF	0	R/W	WDTIF – Watchdog timer interrupt flag. 0 Interrupt not pending 1 Interrupt pending
3	P1IF	0	R/W	P1IF – Port 1 interrupt flag. 0 Interrupt not pending 1 Interrupt pending
2	UTX1IF	0	R/W	UTX1IF – USART1 TX interrupt flag. 0 Interrupt not pending 1 Interrupt pending
1	UTX0IF	0	R/W	UTX0IF – USART0 TX interrupt flag. 0 Interrupt not pending 1 Interrupt pending
0	P2IF	0	R/W	P2IF – Port2 interrupt flag. 0 Interrupt not pending 1 Interrupt pending

12.7.3 Interrupt Priority

The interrupts are grouped into six interrupt priority groups and the priority for each group is set by the registers IP0 and IP1. In order to assign a higher priority to an interrupt, i.e. to its interrupt group, the corresponding bits in IP0 and IP1 must be set as shown in Table 30 on page 61.

The interrupt priority groups with assigned interrupt sources are shown in Table 31. Each group is assigned one of four priority levels. While an interrupt service request is in progress, it cannot be interrupted by a lower or same level interrupt.

In the case when interrupt requests of the same priority level are received simultaneously, the polling sequence shown in Table 32 is used to resolve the priority of each request.

IP1 (0xB9) – Interrupt Priority 1

Bit	Name	Reset	R/W	Description
7:6	–	00	R/W	Not used.
5	IP1_5	0	R/W	Interrupt group 5, priority control bit 1, refer to Table 30
4	IP1_4	0	R/W	Interrupt group 4, priority control bit 1, refer to Table 30
3	IP1_3	0	R/W	Interrupt group 3, priority control bit 1, refer to Table 30
2	IP1_2	0	R/W	Interrupt group 2, priority control bit 1, refer to Table 30
1	IP1_1	0	R/W	Interrupt group 1, priority control bit 1, refer to Table 30
0	IP1_0	0	R/W	Interrupt group 0, priority control bit 1, refer to Table 30

IP0 (0xA9) – Interrupt Priority 0

Bit	Name	Reset	R/W	Description
7:6	–	00	R/W	Not used.
5	IP0_5	0	R/W	Interrupt group 5, priority control bit 0, refer to Table 30
4	IP0_4	0	R/W	Interrupt group 4, priority control bit 0, refer to Table 30
3	IP0_3	0	R/W	Interrupt group 3, priority control bit 0, refer to Table 30
2	IP0_2	0	R/W	Interrupt group 2, priority control bit 0, refer to Table 30
1	IP0_1	0	R/W	Interrupt group 1, priority control bit 0, refer to Table 30
0	IP0_0	0	R/W	Interrupt group 0, priority control bit 0, refer to Table 30

IP1_x	IP0_x	Priority Level
0	0	0 – lowest
0	1	1
1	0	2
1	1	3 – highest

Table 30: Priority Level Setting

Group	Interrupts		
IP0	RFERR	RF	DMA
IP1	ADC	P2INT	T1
IP2	URX0	UTX0	T2
IP3	URX1	UTX1	T3
IP4	ENC	P1INT	T4
IP5	ST	WDT	P0INT

Table 31: Interrupt Priority Groups

Interrupt number	Interrupt name	
0	RFERR	Polling sequence ↓
16	RF	
8	DMA	
1	ADC	
9	T1	
2	URX0	
10	T2	
3	URX1	
11	T3	
4	ENC	
12	T4	
5	ST	
13	P0INT	
6	P2INT	
7	UTX0	
14	UTX1	
15	P1INT	
17	WDT	

Table 32: Interrupt Polling Sequence

12.8 Oscillators and clocks

The ***CC2430*** has one internal system clock. The source for the system clock can be either a 16 MHz high-frequency RC oscillator or a 32 MHz crystal oscillator. Clock control is performed using the CLKCON SFR register described in section 13.10.

The system clock also feeds all 8051 peripherals (as described in section 6).

The choice of oscillator allows a trade-off between high-accuracy in the case of the crystal oscillator and low power consumption when the high-frequency RC oscillator is used. Note that operation of the RF transceiver requires that the crystal oscillator is used.

12.9 Debug Interface

The ***CC2430*** includes a debug interface that provides a two-wire interface to an on-chip debug module. The debug interface allows programming of the on-chip flash and it provides access to memory and register contents and debug features such as breakpoints, single-stepping and register modification.

The debug interface uses the I/O pins P2_1 as Debug Data and P2_2 as Debug Clock during Debug mode. These I/O pins can be used as general purpose I/O only while the device is not in Debug mode. Thus the debug interface does not interfere with any peripheral I/O pins.

12.9.1 Debug Mode

Debug mode is entered by forcing two rising edge transitions on pin P2_2 (Debug Clock) while the RESET_N input is held low.

While in Debug mode pin P2_1 is the Debug Data bi-directional pin and P2_2 is the Debug Clock input pin.

12.9.2 Debug Communication

The debug interface uses an SPI-like two-wire interface consisting of the Debug Data (P2_1) and Debug Clock (P2_2) pins. Data is driven on the bi-directional Debug Data pin at the positive edge of Debug Clock and data is sampled on the negative edge of this clock.

Debug commands are sent by an external host and consist of 1 to 4 output bytes (including command byte) from the host and an optional input byte read by the host. Figure 16 shows a timing diagram of data on the debug interface.

The first byte of the debug command is a command byte and is encoded as follows:

- bits 7 to 3 : instruction code
- bits 2 : return input byte to host byte when high
- bits 1 to 0 : number of output bytes from host following instruction code byte

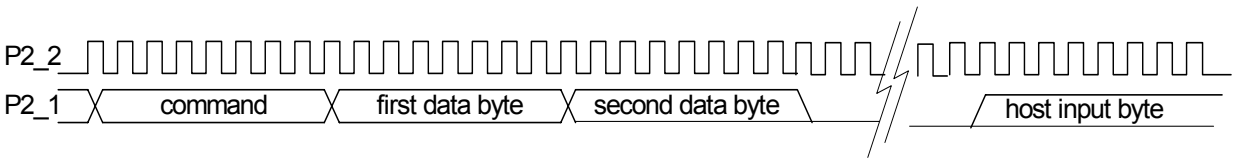


Figure 16: Debug interface timing diagram

12.9.3 Debug Commands

The debug commands are shown in Table 33. Some of the debug commands are described in further detail in the following sections.

12.9.4 Debug Lock Bit

For software code security the Debug Interface may be locked. When the Debug Lock bit , DBGLOCK, is set (see section 13.14.3) all debug commands except CHIP_ERASE, READ_STATUS and GET_CHIP_ID are disabled and will not function. The status of the Debug Lock bit can read using the READ_STATUS command (see section 12.9.6). Note that when the Debug Lock bit has been set, and before reset of the device, the Debug Lock value returned by READ_STATUS will only represent the actual Debug Lock value once either a HALT, RESUME, DEBUG_INSTR or STEP command has been attempted executed.

The CHIP_ERASE command is used to clear the Debug Lock bit.

12.9.5 Debug Configuration

The commands WR_CONFIG and RD_CONFIG are used to access the debug configuration data byte. The format and description of this configuration data is shown in Table 34.

12.9.6 Debug Status

A Debug status byte is read using the READ_STATUS command. The format and description of this debug status is shown in Table 35.

The READ_STATUS command is used e.g. for polling the status of flash chip erase after a CHIP_ERASE command or oscillator stable status required for debug commands HALT, RESUME, DEBUG_INSTR, STEP_REPLACE and STEP_INSTR.

Command	Instruction code	Description
CHIP_ERASE	0001 0x00	Perform flash chip erase (mass erase) and clear lock bits. If any other command, except READ_STATUS, is issued, then the use of CHIP_ERASE is disabled.
WR_CONFIG	0001 1x01	Write configuration data. Refer to Table 34
RD_CONFIG	0010 0100	Read configuration data. Returns value set by WR_CONFIG command.
GET_PC	0010 1000	Return value of 16-bit program counter. Returns 2 bytes regardless of value of bit 2 in instruction code
READ_STATUS	0011 0x00	Read status byte. Refer to Table 35
SET_HW_BRKPNT	0011 1x11	Set hardware breakpoint
HALT	0100 0100	Halt CPU operation
RESUME	0100 1100	Resume CPU operation. The CPU must be in halted state for this command to be run.
DEBUG_INSTR	0101 01xx	Run debug instruction. The supplied instruction will be executed by the CPU without incrementing the program counter. The CPU must be in halted state for this command to be run.
STEP_INSTR	0101 1100	Step CPU instruction. The CPU will execute the next instruction from program memory and increment the program counter after execution. The CPU must be in halted state for this command to be run.
STEP_REPLACE	0110 01xx	Step and replace CPU instruction. The supplied instruction will be executed by the CPU instead of the next instruction in program memory. The program counter will be incremented after execution. The CPU must be in halted state for this command to be run.
GET_CHIP_ID	0110 1000	Return value of 16-bit chip ID and version number. Returns 2 bytes regardless of value of bit 2 of instruction code

Table 33: Debug Commands

Bit	Name	Description
7-4	–	Not used
3	`timers_off`	Disable timers. Disable timer operation 0 Do not disable timers 1 Disable timers
2	`DMA_pause`	DMA pause 0 Enable DMA transfers 1 Pause all DMA transfers
1	`timer_suspend`	Suspend timers. Timer operation is suspended for debug instructions and if a step instruction is a branch. If not suspended these instructions would result an extra timer count during the clock cycle in which the branch is executed 0 Do not suspend timers 1 Suspend timers
0	`sel_flash_info_page`	Select flash information page 0 Select flash main page 1 Select flash information page

Table 34: Debug Configuration

Bit	Name	Description
7	chip_erase_done	Flash chip erase done 0 Chip erase in progress 1 Chip erase done
6	pcon_idle	PCON idle 0 CPU is running 1 CPU is idle (clock gated)
5	cpu_halted	CPU halted 0 CPU running 1 CPU halted
4	power_mode_0	Power Mode 0 0 Power Mode 1-3 selected 1 Power Mode 0 selected
3	halt_status	Halt status. Returns cause of last CPU halt 0 CPU was halted by HALT debug command 1 CPU was halted by software or hardware breakpoint
2	debug_locked	Debug locked. Returns value of DBGLOCK bit 0 Debug interface is not locked 1 Debug interface is locked
1	oscillator_stable	Oscillators stable. This bit represents the status of the CLKCON.XSOC_STB and CLKCON.HFRC_STB register bits. 0 Oscillators not stable 1 Oscillators stable
0	stack_overflow	Stack overflow. This bit indicates when the CPU writes to DATA memory space at address 0xFF which is possibly a stack overflow 0 No stack overflow 1 Stack overflow

Table 35: Debug Status

12.9.7 Hardware Breakpoints

The debug command SET_HW_BRKPNT is used to set a hardware breakpoint. The ***CC2430*** supports up to four hardware breakpoints. When a hardware breakpoint is enabled it will compare the CPU address bus with the breakpoint.. When a match occurs, the CPU is halted.

When issuing the SET_HW_BRKPNT, the external host must supply three data bytes that define the hardware breakpoint. The hardware breakpoint itself consists of 18 bits while three bits are used for control purposes. The format of the three data bytes for the SET_HW_BRKPNT command is as follows.

The first data byte consists of the following:

- bits 7-5 : unused
- bits 4-3 : breakpoint number; 0-3
- bit 2 : 1=enable, 0=disable
- bits 1-0 : Memory bank bits. Bits 17-16 of hardware breakpoint.

The second data byte consists of bits 15-8 of the hardware breakpoint.

The third data byte consists of bits 7-0 of the hardware breakpoint.

12.9.8 Flash Programming

Programming of the on-chip flash is performed via the debug interface. The external host must initially send instructions using the DEBUG_INSTR debug command to perform the flash programming with the Flash Controller as described in section 13.14 on page 157.

12.10 RAM

The ***CC2430*** contains static RAM. At power-on the contents of RAM is undefined. The RAM size is 8 KB in total. The upper 4 KB of the RAM (XDATA memory locations 0xF000-0xFFFF) retains data in all power modes (see exception below). The remaining lower 4 KB (XDATA memory locations 0xE000-0xEFFF) will loose its contents in PM2 and PM3 and contains undefined data when returning to PM0.

The memory locations 0xFD58-0xFEFF consisting of 424 bytes in XDATA memory space do not retain data when PM2/3 is entered.

12.11 Flash Memory

The on-chip flash memory consists of 32768, 655536 or 131072 bytes. The flash memory is primarily intended to hold program code. The flash memory has the following features:

- Flash page erase time: 20 ms
- Flash chip (mass) erase time: 20 ms
- Flash write time (4 bytes): 20 µs
- Data retention[1]:100 years
- Program/erase endurance: 1,000 cycles

[1] At room temperature

The flash memory consists of the Flash Main Page which is where the CPU reads program code and data. The flash memory also contains a Flash Information Page which contains the Flash Lock Bits. The Flash Information Page and hence the Lock Bits is only accessed by first selecting this page through the Debug Interface. The Flash Controller (see section 13.14) is used to write and erase the contents of the flash memory.

When the CPU reads instructions from flash memory, it fetches the next instruction through a cache. The instruction cache is provided mainly to reduce power consumption by reducing the amount of time the flash memory itself is accessed. The use of the instruction cache may be disabled with the `MEMCTR.CACHDIS` register bit.

12.12 Memory Arbiter

The ***CC2430*** includes a memory arbiter which handles CPU and DMA access to all memory space.

The control register `MEMCTR` is used to control various aspects of the memory sub-system. The `MEMCTR` register is described below.

`MEMCTR.MUNIF` controls unified mapping of CODE memory space as shown in

Figure 13 on page 38. Unified mapping is required when the CPU is to execute program stored in XDATA.

For the 128 KB flash version (CC2430-F128), `MEMCTR.FMAP1:0` controls mapping of physical banks of the 128 KB flash to the program address region 0x8000-0xFFFF in CODE memory space as shown in Figure 14 on 23.

MEMCTR (0xC7) – Memory Arbiter Control CC2430-F128

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	MUNIF	0	R/W	Unified memory mapping. When unified mapping is enabled, all physical memories are mapped into the CODE memory space as far as possible, when uniform mapping is disabled only flash memory is mapped to CODE space 0 Disable unified mapping 1 Enable unified mapping
5:4	FMAP[1:0]	01	R/W	Flash bank map. Controls which of the four 32 KB flash memory banks to map to program address 0x8000 – 0xFFFF in CODE memory space. 00 Map program address 0x8000 – 0xFFFF to physical memory address 0x00000 – 0x07FFF 01 Map program address 0x8000 – 0xFFFF to physical memory address 0x08000– 0x0FFFF 10 Map program address 0x8000 – 0xFFFF to physical memory address 0x10000 – 0x17FFF 11 Map program address 0x8000 – 0xFFFF to physical memory address 0x18000 – 0x1FFFF
3:2	–	00	R0	Not used
1	CACHDIS	0	R/W	Flash cache disable. Invalidates contents of instruction cache and forces all instruction read accesses to read straight from flash memory. Disabling will increase power consumption and is provided for debug purposes. 0 Cache enabled 1 Cache disabled
0	–	1	R/W	Reserved. Always set to 1.[2]

[2] Reserved bits must always be set to the specified value. Failure to follow this will result in indeterminate behaviour.

MEMCTR (0xC7) – Memory Arbiter Control CC2430-F32/64

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	MUNIF	0	R/W	Unified memory mapping. When unified mapping is enabled, all physical memories are mapped into the CODE memory space as far as possible, when uniform mapping is disabled only flash memory is mapped to CODE space 0 Disable unified mapping 1 Enable unified mapping
5:2	–	0000	R/W	Reserved. Always set to 0000.[3]
1	CACHDIS	0	R/W	Flash cache disable. Invalidates contents of instruction cache and forces all instruction read accesses to read straight from flash memory. Disabling will increase power consumption and is provided for debug purposes. 0 Cache enabled 1 Cache disabled
0	–	1	R/W	Reserved. Always set to 1. [3]

[3] Reserved bits must always be set to the specified value. Failure to follow this will result in indeterminate behaviour.

13 Peripherals

In the following sub-sections each ***CC2430*** peripheral is described in detail.

The ***CC2430*** has four timers. These timers all run on the tick frequency given by the Power Management Controller register CLKCON.TICKSPD.

13.1 I/O ports

The ***CC2430*** has 21 digital input/output pins that can be configured as general purpose digital I/O or as peripheral I/O signals connected to the ADC, Timers or USART peripherals. The usage of the I/O ports is fully configurable from user software through a set of configuration registers.

The I/O ports have the following key features:

- 21 digital input/output pins
- General purpose I/O or peripheral I/O
- Pull-up or pull-down capability on inputs
- External interrupt capability

The external interrupt capability is available on all 21 I/O pins. Thus external devices may generate interrupts if required. The external interrupt feature can also be used to wake up from sleep modes.

13.1.1 General Purpose I/O

When used as general purpose I/O, the pins are organized as three 8-bit ports, ports 0-2, denoted P0, P1 and P2. P0 and P1 are complete 8-bit wide ports while P2 has only five usable bits. All ports are both bit- and byte addressable through the SFR registers P0, P1 and P2. Each port pin can individually be set to operate as a general purpose I/O or as a peripheral I/O.

The output drive strength is 4 mA on all outputs, except for the two high-drive outputs, P1_0 and P1_1, which each have 20 mA output drive strength.

To use a port as a general purpose I/O pin the pin must first be configured. The registers PxSEL where x is the port number 0-2 are used to configure each pin in a port as either a general purpose I/O pin or as a peripheral I/O signal. By default, after a reset, all digital input/output pins are configured as general-purpose I/O pins.

By default all general-purpose I/O pins are configured as inputs. To change the direction of a port pin, at any time, the registers PxDIR are used to set each port pin to be either an input or an output. Thus by setting the appropriate bit within PxDIR, to 1 the corresponding pin becomes an output.

When reading the port registers P0, P1 and P2, the logic values on the input pins are returned regardless of the pin configuration. This does not apply during the execution of read-modify-write instructions. The read-modify-write instructions when operating on a port registers are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a port register P0, P1 or P2. For these read-modify-write instructions, the value of the register, not the value on the pin, is read, modified, and written back to the port register.

When used as an input, the general purpose I/O port pins can be configured to have a pull-up, pull-down or tri-state mode of operation. By default, after a reset, inputs are configured as inputs with pull-up. To deselect the pull-up or pull-down function on an input the appropriate bit within the PxINP must be set to 1. The I/O port pins P1_0 and P1_1 do not have pull-up/pull-down capability.

In power modes PM2 and PM3 the I/O pins retain the I/O mode and output value (if applicable) that was set when PM2/3 was entered.

13.1.2 General Purpose I/O Interrupts

General purpose I/O pins configured as inputs can be used to generate interrupts. The interrupts can be configured to trigger on either a rising or falling edge of the external signal. Each of the P0, P1 and P2 ports have separate interrupt enable bits common for all bits within the port located in the IEN1-2 registers as follows:

- IEN1.P0IE : P0 interrupt enable
- IEN2.P1IE : P1 interrupt enable
- IEN2.P2IE : P2 interrupt enable

In addition to these common interrupt enables, the bits within each port have interrupt enables located in I/O port SFR registers. Each bit within P1 has an individual interrupt enable. In P0 the low-order nibble and the high-order nibble have their individual interrupt enables. For the P2_0 – P2_4 inputs there is a common interrupt enable.

When an interrupt condition occurs on one of the general purpose I/O pins, the corresponding interrupt status flag in the P0-P2 interrupt flag registers, P0IFG , P1IFG or P2IFG will be set to 1. The interrupt status flag is set regardless of whether the pin has its interrupt enable set. When an interrupt is serviced the interrupt status flag is cleared by writing to a 0 to that flag.

Note that when clearing the PxIFG interrupt status flags, only one active flag should be cleared at a time. Failure to do this may result in generation of false interrupt requests.

The I/O SFR registers used for interrupts are described in section 13.1.9 on page 73. The registers are summarized below:

- P1IEN : P1 interrupt enables
- PICTL : P0/P2 interrupt enables and P0-2 edge configuration
- P0IFG : P0 interrupt flags
- P1IFG : P1 interrupt flags
- P2IFG : P2 interrupt flags

13.1.3 General Purpose I/O DMA

When used as general purpose I/O pins, the P0 and P1 ports are each associated with one DMA trigger. These DMA triggers are IOC_0 for P0 and IOC_1 for P1 as shown in Table 37 on page 92.

The IOC_0 or IOC_1 DMA trigger is activated when an input transition occurs on one of the P0 or P1 pins respectively. Note input transitions on pins configured as general purpose I/O inputs only will produce the DMA trigger.

13.1.4 Peripheral I/O

This section describes how the digital input/output pins are configured as peripheral I/Os. For each peripheral unit that can interface with an external system through the digital input/output pins, a description of how peripheral I/Os are configured is given in the following sub-sections.

In general, setting the appropriate PxSEL bits to 1 is required to select peripheral I/O function on a digital I/O pin.

Note that peripheral units have two alternative locations for their I/O pins, refer to Table 36.

Table 36: Peripheral I/O Pin Mapping

Periphery / Function	P0								P1								P2				
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	4	3	2	1	0
ADC	A7	A6	A5	A4	A3	A2	A1	A0													
USART0 SPI			C	SS	M0	MI															
Alt. 2											MO	MI	C	SS							
USART0 UART			RT	CT	TX	RX															
Alt. 2											TX	RX	RT	CT							
USART1 SPI			MI	M0	C	SS															
Alt. 2									MI	M0	C	SS									
USART1 UART			RX	TX	RT	CT															
Alt. 2									RX	TX	RT	CT									
TIMER1				2	1	0															
Alt. 2														0	1	2					
TIMER3												1	0								
Alt. 2									1	0											
TIMER4															1	0					
Alt. 2																		1			0
32.768 kHz XOSC																	Q2	Q1			
DEBUG																			DC	DD	

13.1.4.1 USART0

The SFR register bit `PERCFG.U0CFG` selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the USART0 signals are shown as follows:

UART:

- RX : RXDATA
- TX : TXDATA
- RT : RTS
- CT : CTS

SPI:

- MI : MISO
- MO : MOSI
- C : SCK
- SS : SSN

`P2DIR.PRIP0` selects the order of precedence when assigning several peripherals to port 0. When set to 00, USART0 has precedence. Note that if UART mode is selected and hardware flow control is disabled, USART1 or timer 1 will have precedence to use ports P0_4 and P0_5.

`P2SEL.PRI3P1` and `P2SEL.PRI0P1` select the order of precedence when assigning several peripherals to port 1. USART0 has precedence when both are set to 0. Note that if UART mode is selected and hardware flow control is disabled, timer 1 or timer 3 will have precedence to use ports P1_2 and P1_3.

13.1.4.2 USART1

The SFR register bit `PERCFG.S1CFG` selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the USART1 signals are shown as follows:

- RX : RXDATA
- TX : TXDATA

- RT : RTS
- CT : CTS

SPI:

- MI : MISO
- MO : MOSI
- C : SCK
- SS : SSN

P2DIR.PRIP0 selects the order of precedence when assigning several peripherals to port 0. When set to 01, USART1 has precedence. Note that if UART mode is selected and hardware flow control is disabled, USART0 or timer 1 will have precedence to use ports P0_2 and P0_3.

P2SEL.PRI3P1 and P2SEL.PRI2P1 select the order of precedence when assigning several peripherals to port 1. USART1 has precedence when the former is set to 1 and the latter is set to 0. Note that if UART mode is selected and hardware flow control is disabled, USART0 or timer 3 will have precedence to use ports P2_4 and P2_5.

13.1.4.3 Timer 1

PERCFG.T1CFG selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the Timer 1 signals are shown as the following:

- 0 : Channel 0 capture/compare pin
- 1 : Channel 1 capture/compare pin
- 2 : Channel 2 capture/compare pin

P2DIR.PRIP0 selects the order of precedence when assigning several peripherals to port 0. When set to 10 or 11 the timer 1 channels have precedence.

P2SEL.PRI1P1 and P2SEL.PRI0P1 select the order of precedence when assigning several peripherals to port 1. The timer 1 channels have precedence when the former is set low and the latter is set high.

13.1.4.4 Timer 3

PERCFG.T3CFG selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the Timer 3 signals are shown as the following:

- 0 : Channel 0 capture/compare pin
- 1 : Channel 1 capture/compare pin

P2SEL.PRI2P1 selects the order of precedence when assigning several peripherals to port 1. The timer 3 channels have precedence when the bit is set.

13.1.4.5 Timer 4

PERCFG.T4CFG selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the Timer 4 signals are shown as the following:

- 0 : Channel 0 capture/compare pin
- 1 : Channel 1 capture/compare pin

P2SEL.PRI1P1 selects the order of precedence when assigning several peripherals to port 1. The timer 4 channels have precedence when the bit is set.

13.1.5 ADC

When using the ADC in an application, Port 0 pins must be configured as ADC inputs. Up to eight ADC inputs can be used. To configure a Port 0 pin to be used as an ADC input the corresponding bit in the ADCCFG register must be set to 1. The default values in this register select the Port 0 pins as non-ADC input i.e. digital input/outputs.

The settings in the ADCCFG register override the settings in P0SEL.

The ADC can be configured to use the general-purpose I/O pin P2_0 as an external trigger to start conversions. P2_0 must be configured as a general-purpose I/O in input mode, when being used for ADC external trigger.

Refer to section 13.7 on page 127 for a detailed description of use of the ADC.

13.1.6 Debug interface

Ports P2_1 and P2_2 are used for debug data and clock signals, respectively. These are shown as DD (debug data) and DC (debug clock) in Table 36. When the debug interface is in use, P2DIR should select these pins as inputs. The state of P2SEL is overridden by the debug interface. Also, the direction is overridden when the chip changes the direction to supply the external host with data.

13.1.7 32.768 kHz XOSC input

Ports P2_3 and P2_4 are used to connect an external 32.768 kHz crystal. These port pins will be used by the 32.768 kHz crystal oscillator when CLKCON.OSC32K is low, regardless of register settings. The port pins will be set in analog mode when CLKCON.OSC32K is low.

13.1.8 Unused I/O pins

Unused I/O pins should have a defined level and not be left floating. One way to do this is to leave the pin unconnected and configure the pin as a general purpose I/O input with pull-up resistor. This is also the state of all pins during reset. Alternatively the pin can be configured as a general purpose I/O output. In both cases the pin should not be connected directly to VDD or GND in order to avoid excessive power consumption.

13.1.9 I/O registers

The registers for the I/O ports are described in this section. The registers are:

- P0 Port 0
- P1 Port 1
- P2 Port 2
- PERCFG Peripheral control register
- ADCCFG ADC input configuration register
- P0SEL Port 0 function select register
- P1SEL Port 1 function select register
- P2SEL Port 2 function select register
- P0DIR Port 0 direction register
- P1DIR Port 1 direction register
- P2DIR Port 2 direction register
- P0INP Port 0 input mode register
- P1INP Port 1 input mode register
- P2INP Port 2 input mode register
- P0IFG Port 0 interrupt status flag register
- P1IFG Port 1 interrupt status flag register
- P2IFG Port 2 interrupt status flag register
- PICTL Interrupt mask and edge register
- P1IEN Port 1 interrupt mask register

P0 (0x80) – Port 0

Bit	Name	Reset	R/W	Description
7:0	P0[7:0]	0xFF	R/W	Port 0. General purpose I/O port. Bit-addressable.

P1 (0x90) – Port 1

Bit	Name	Reset	R/W	Description
7:0	P1[7:0]	0xFF	R/W	Port 1. General purpose I/O port. Bit-addressable.

P2 (0xA0) – Port 2

Bit	Name	Reset	R/W	Description
7:5	-	000	R0	Not used
4:0	P2[4:0]	0x1F	R/W	Port 2. General purpose I/O port. Bit-addressable.

PERCFG (0xF1) – Peripheral Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	T1CFG	0	R/W	Timer 1 I/O location 0 Alternative 1 location 1 Alternative 2 location
5	T3CFG	0	R/W	Timer 3 I/O location 0 Alternative 1 location 1 Alternative 2 location
4	T4CFG	0	R/W	Timer 4 I/O location 0 Alternative 1 location 1 Alternative 2 location
3:2	–	00	R0	Not used
1	U1CFG	0	R/W	USART1 I/O location 0 Alternative 1 location 1 Alternative 2 location
0	U0CFG	0	R/W	USART0 I/O location 0 Alternative 1 location 1 Alternative 2 location

ADCCFG (0xF2) – ADC Input Configuration

Bit	Name	Reset	R/W	Description
7:0	ADCCFG[7:0]	0x00	R/W	ADC input configuration. ADCCFG[7:0] select P0_7 - P0_0 as ADC inputs AIN7 – AIN0 0 ADC input disabled 1 ADC input enabled

P0SEL (0xF3) – Port 0 Function Select

Bit	Name	Reset	R/W	Description
7	SELP0_7	0	R/W	P0_7 function select 0 General purpose I/O 1 Peripheral function
6	SELP0_6	0	R/W	P0_6 function select 0 General purpose I/O 1 Peripheral function
5	SELP0_5	0	R/W	P0_5 function select 0 General purpose I/O 1 Peripheral function
4	SELP0_4	0	R/W	P0_4 function select 0 General purpose I/O 1 Peripheral function
3	SELP0_3	0	R/W	P0_3 function select 0 General purpose I/O 1 Peripheral function
2	SELP0_2	0	R/W	P0_2 function select 0 General purpose I/O 1 Peripheral function
1	SELP0_1	0	R/W	P0_1 function select 0 General purpose I/O 1 Peripheral function
0	SELP0_0	0	R/W	P0_0 function select 0 General purpose I/O 1 Peripheral function

P1SEL (0xF4) – Port 1 Function Select

Bit	Name	Reset	R/W	Description
7	SELP1_7	0	R/W	P1_7 function select 0 General purpose I/O 1 Peripheral function
6	SELP1_6	0	R/W	P1_6 function select 0 General purpose I/O 1 Peripheral function
5	SELP1_5	0	R/W	P1_5 function select 0 General purpose I/O 1 Peripheral function
4	SELP1_4	0	R/W	P1_4 function select 0 General purpose I/O 1 Peripheral function
3	SELP1_3	0	R/W	P1_3 function select 0 General purpose I/O 1 Peripheral function
2	SELP1_2	0	R/W	P1_2 function select 0 General purpose I/O 1 Peripheral function
1	SELP1_1	0	R/W	P1_1 function select 0 General purpose I/O 1 Peripheral function
0	SELP1_0	0	R/W	P1_0 function select 0 General purpose I/O 1 Peripheral function

P2SEL (0xF5) – Port 2 Function Select

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	PRI3P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns USART0 and USART1 to the same pins. 0 USART0 has priority 1 USART1 has priority
5	PRI2P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns USART1 and timer 3 to the same pins. 0 USART1 has priority 1 Timer 3 has priority
4	PRI1P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns timer 1 and timer 4 to the same pins. 0 Timer 1 has priority 1 Timer 4 has priority
3	PRI0P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns USART0 and timer 1 to the same pins. 0 USART0 has priority 1 Timer 1 has priority
2	SELP2_4	0	R/W	P2_4 function select 0 General purpose I/O 1 Peripheral function
1	SELP2_3	0	R/W	P2_3 function select 0 General purpose I/O 1 Peripheral function
0	SELP2_0	0	R/W	P2_0 function select 0 General purpose I/O 1 Peripheral function

P0DIR (0xFD) – Port 0 Direction

Bit	Name	Reset	R/W	Description
7	DIRP0_7	0	R/W	P0_7 I/O direction 0 Input 1 Output
6	DIRP0_6	0	R/W	P0_6 I/O direction 0 Input 1 Output
5	DIRP0_5	0	R/W	P0_5 I/O direction 0 Input 1 Output
4	DIRP0_4	0	R/W	P0_4 I/O direction 0 Input 1 Output
3	DIRP0_3	0	R/W	P0_3 I/O direction 0 Input 1 Output
2	DIRP0_2	0	R/W	P0_2 I/O direction 0 Input 1 Output
1	DIRP0_1	0	R/W	P0_1 I/O direction 0 Input 1 Output
0	DIRP0_0	0	R/W	P0_0 I/O direction 0 Input 1 Output

P1DIR (0xFE) – Port 1 Direction

Bit	Name	Reset	R/W	Description
7	DIRP1_7	0	R/W	P1_7 I/O direction 0 Input 1 Output
6	DIRP1_6	0	R/W	P1_6 I/O direction 0 Input 1 Output
5	DIRP1_5	0	R/W	P1_5 I/O direction 0 Input 1 Output
4	DIRP1_4	0	R/W	P1_4 I/O direction 0 Input 1 Output
3	DIRP1_3	0	R/W	P1_3 I/O direction 0 Input 1 Output
2	DIRP1_2	0	R/W	P1_2 I/O direction 0 Input 1 Output
1	DIRP1_1	0	R/W	P1_1 I/O direction 0 Input 1 Output
0	DIRP1_0	0	R/W	P1_0 I/O direction 0 Input 1 Output

P2DIR (0xFF) – Port 2 Direction

Bit	Name	Reset	R/W	Description
7:6	PRIP0[1:0]	0	R/W	Port 0 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns several peripherals to the same pins 00 USART0 – USART1 01 USART1 – USART0 10 Timer 1 channels 0 and 1 – USART1 11 Timer 1 channel 2 – USART0
5	–	0	R0	Not used
4	DIRP2_4	0	R/W	P2_4 I/O direction 0 Input 1 Output
3	DIRP2_3	0	R/W	P2_3 I/O direction 0 Input 1 Output
2	DIRP2_2	0	R/W	P2_2 I/O direction 0 Input 1 Output
1	DIRP2_1	0	R/W	P2_1 I/O direction 0 Input 1 Output
0	DIRP2_0	0	R/W	P2_0 I/O direction 0 Input 1 Output

P0INP (0x8F) – Port 0 Input Mode

Bit	Name	Reset	R/W	Description
7	MDP0_7	0	R/W	P0_7 I/O input mode 0 Pull-up / pull-down 1 Tristate
6	MDP0_6	0	R/W	P0_6 I/O input mode 0 Pull-up / pull-down 1 Tristate
5	MDP0_5	0	R/W	P0_5 I/O input mode 0 Pull-up / pull-down 1 Tristate
4	MDP0_4	0	R/W	P0_4 I/O input mode 0 Pull-up / pull-down 1 Tristate
3	MDP0_3	0	R/W	P0_3 I/O input mode 0 Pull-up / pull-down 1 Tristate
2	MDP0_2	0	R/W	P0_2 I/O input mode 0 Pull-up / pull-down 1 Tristate
1	MDP0_1	0	R/W	P0_1 I/O input mode 0 Pull-up / pull-down 1 Tristate
0	MDP0_0	0	R/W	P0_0 I/O input mode 0 Pull-up / pull-down 1 Tristate

P1INP (0xF6) – Port 1 Input Mode

Bit	Name	Reset	R/W	Description
7	MDP1_7	0	R/W	P1_7 I/O input mode 0 Pull-up / pull-down 1 Tristate
6	MDP1_6	0	R/W	P1_6 I/O input mode 0 Pull-up / pull-down 1 Tristate
5	MDP1_5	0	R/W	P1_5 I/O input mode 0 Pull-up / pull-down 1 Tristate
4	MDP1_4	0	R/W	P1_4 I/O input mode 0 Pull-up / pull-down 1 Tristate
3	MDP1_3	0	R/W	P1_3 I/O input mode 0 Pull-up / pull-down 1 Tristate
2	MDP1_2	0	R/W	P1_2 I/O input mode 0 Pull-up / pull-down 1 Tristate
1:0	–	00	R0	Not used

P2INP (0xF7) – Port 2 Input Mode

Bit	Name	Reset	R/W	Description
7	PDUP2	0	R/W	Port 2 pull-up/down select. Selects function for all Port 2 pins configured as pull-up/pull-down inputs. 0 Pull-up 1 Pull-down
6	PDUP1	0	R/W	Port 1 pull-up/down select. Selects function for all Port 1 pins configured as pull-up/pull-down inputs. 0 Pull-up 1 Pull-down
5	PDUP0	0	R/W	Port 0 pull-up/down select. Selects function for all Port 0 pins configured as pull-up/pull-down inputs. 0 Pull-up 1 Pull-down
4	MDP2_4	0	R/W	P2_4 I/O input mode 0 Pull-up / pull-down 1 Tristate
3	MDP2_3	0	R/W	P2_3 I/O input mode 0 Pull-up / pull-down 1 Tristate
2	MDP2_2	0	R/W	P2_2 I/O input mode 0 Pull-up / pull-down 1 Tristate
1	MDP2_1	0	R/W	P2_1 I/O input mode 0 Pull-up / pull-down 1 Tristate
0	MDP2_0	0	R/W	P2_0 I/O input mode 0 Pull-up / pull-down 1 Tristate

P0IFG (0x89) – Port 0 Interrupt Status Flag

Bit	Name	Reset	R/W	Description
7:0	P0IF[7:0]	0x00	R/W0	Port 0, inputs 7 to 0 interrupt status flags. When an input port pin has an interrupt request pending, the corresponding flag bit will be set.

P1IFG (0x8A) – Port 1 Interrupt Status Flag

Bit	Name	Reset	R/W	Description
7:0	P1IF[7:0]	0x00	R/W0	Port 1, inputs 7 to 0 interrupt status flags. When an input port pin has an interrupt request pending, the corresponding flag bit will be set.

P2IFG (0x8B) – Port 2 Interrupt Status Flag

Bit	Name	Reset	R/W	Description
7:5	–	000	R0	Not used.
4:0	P2IF[4:0]	0x00	R/W0	Port 2, inputs 4 to 0 interrupt status flags. When an input port pin has an interrupt request pending, the corresponding flag bit will be set.

PICTL (0x8C) – Port Interrupt Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	PADSC	0	R/W	Strength control for port pads in output mode. Selects output drive capability to account for low I/O supply voltage on pin DVDD. 0 Minimum drive capability 1 Maximum drive capability
5	P2IEN	0	R/W	Port 2, inputs 4 to 0 interrupt enable. This bit enables interrupt requests for the port 2 inputs 4 to 0. 0 Interrupts are disabled 1 Interrupts are enabled
4	P0IENH	0	R/W	Port 0, inputs 7 to 4 interrupt enable. This bit enables interrupt requests for the port 0 inputs 7 to 4. 0 Interrupts are disabled 1 Interrupts are enabled
3	P0IENL	0	R/W	Port 0, inputs 3 to 0 interrupt enable. This bit enables interrupt requests for the port 0 inputs 3 to 0. 0 Interrupts are disabled 1 Interrupts are enabled
2	P2ICON	0	R/W	Port 2, inputs 4 to 0 interrupt configuration. This bit selects the interrupt request condition for all port 2 inputs 0 Rising edge on input gives interrupt 1 Falling edge on input gives interrupt
1	P1ICON	0	R/W	Port 1, inputs 7 to 0 interrupt configuration. This bit selects the interrupt request condition for all port 1 inputs 0 Rising edge on input gives interrupt 1 Falling edge on input gives interrupt
0	P0ICON	0	R/W	Port 0, inputs 7 to 0 interrupt configuration. This bit selects the interrupt request condition for all port 0 inputs 0 Rising edge on input gives interrupt 1 Falling edge on input gives interrupt

P1IEN (0x8D) – Port 1 Interrupt Mask

Bit	Name	Reset	R/W	Description
7	P1_7IEN	0	R/W	Port P1_7 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
6	P1_6IEN	0	R/W	Port P1_6 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
5	P1_5IEN	0	R/W	Port P1_5 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
4	P1_4IEN	0	R/W	Port P1_4 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
3	P1_3IEN	0	R/W	Port P1_3 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
2	P1_2IEN	0	R/W	Port P1_2 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
1	P1_1IEN	0	R/W	Port P1_1 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
0	P1_0IEN	0	R/W	Port P1_0 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled

13.2 DMA Controller

The *CC2430* includes a direct memory access (DMA) controller, which can be used to relieve the 8051 CPU core of handling data movement operations thus achieving high overall performance with good power efficiency. The DMA controller can move data from a peripheral unit such as ADC or RF transceiver to memory with minimum CPU intervention.

The DMA controller coordinates all DMA transfers, ensuring that DMA requests are prioritized appropriately relative to each other and CPU memory access. The DMA controller contains a number of programmable DMA channels for memory-memory data movement.

The DMA controller controls data transfers over the entire address range in XDATA memory space. Since most of the SFR registers are mapped into the DMA memory space, these flexible DMA channels can be used to unburden the CPU in innovative ways, e.g. feed a USART with data from memory or periodically transfer samples between ADC and memory, etc. Use of the DMA can also reduce system power consumption by keeping the CPU in a low-power mode without having to wake up to move data to or from a peripheral unit. Note that section 12.4 describes which SFR registers that are not mapped into XDATA memory space.

The main features of the DMA controller are as follows:

- Five independent DMA channels
- Three configurable levels of DMA channel priority
- 31 configurable transfer trigger events
- Independent control of source and destination address
- Single, block and repeated transfer modes
- Supports length field in transfer data setting variable transfer length
- Can operate in either word-size or byte-size mode

13.2.1 DMA Operation

There are five DMA channels available in the DMA controller numbered channel 0 to channel 4. Each DMA channel can move data from one place within the DMA memory space to another i.e. between XDATA locations.

In order to use a DMA channel it must first be configured as described in sections 13.2.2 and 13.2.3. Figure 17 shows the DMA state diagram.

Once a DMA channel has been configured it must be armed before any transfers are allowed to be initiated. A DMA channel is armed by setting the appropriate bit in the DMA Channel Arm register `DMAARM`.

When a DMA channel is armed a transfer will begin when the configured DMA trigger event occurs. There are 31 possible DMA trigger events, e.g. UART transfer, Timer overflow etc. The trigger event to be used by a DMA channel is set by the DMA channel configuration. The DMA trigger events are listed in Table 37.

In addition to starting a DMA transfer through the DMA trigger events, the user software may force a DMA transfer to begin by setting the corresponding `DMAREQ` bit.

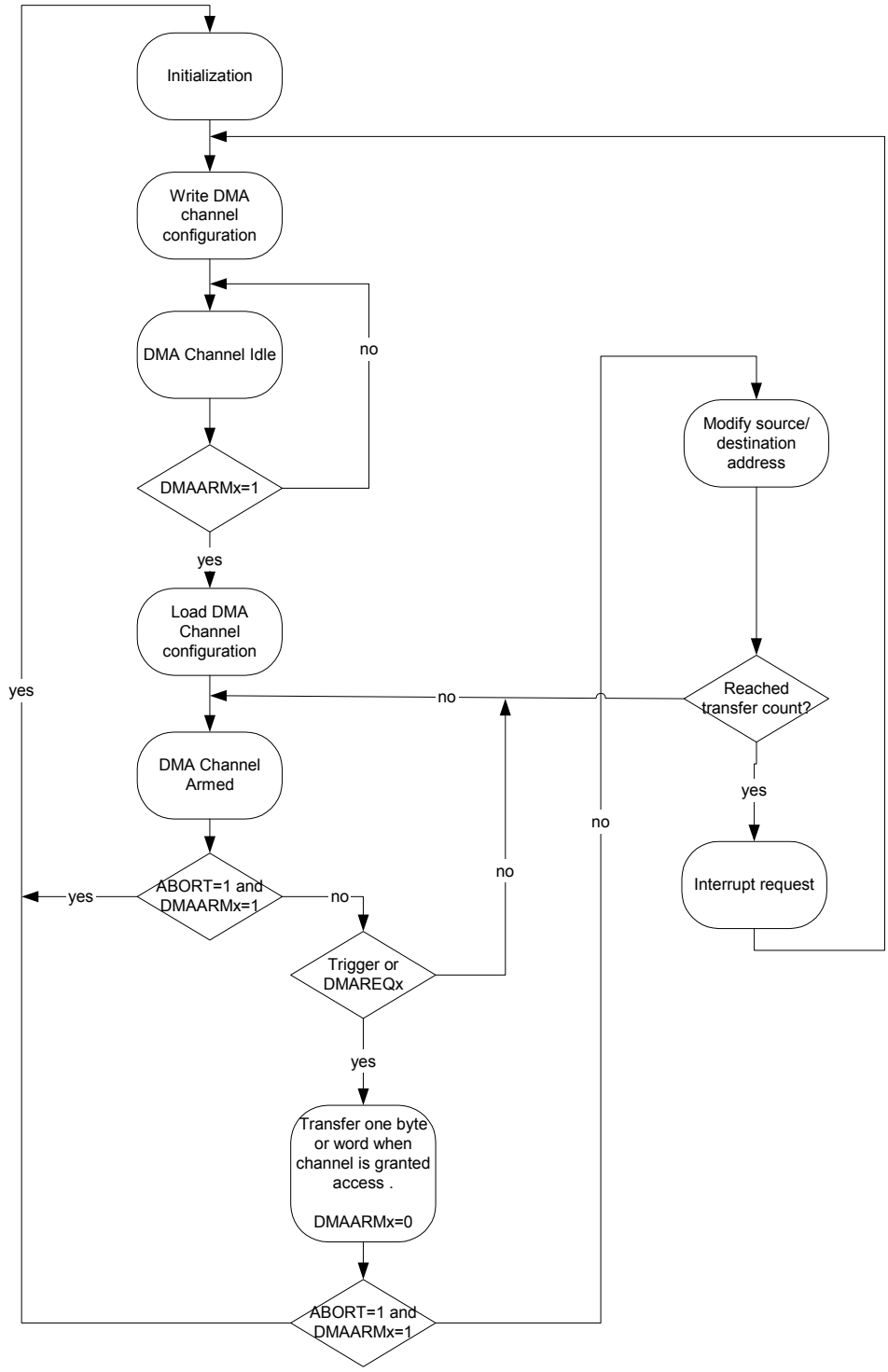


Figure 17: DMA Operation

13.2.2 DMA Configuration Parameters

Setup and control of the DMA operation is performed by the user software. This section describes the parameters which must be configured before a DMA channel can be used. Section 13.2.3 on page 90 describes how the parameters are set up in software and passed to the DMA controller.

The behavior of each of the five DMA channels is configured with the following parameters:

Source address. The first address from which the DMA channel should read data.

Destination address. The first address to which the DMA channel should write the data read from the source address. The user must ensure that the destination is writable.

Transfer count. The number of transfers to perform before rearming or disarming the DMA channel and alerting the CPU with an interrupt request. The length can be defined in the configuration or it can be defined as described next as VLEN setting.

VLEN setting. The DMA channel is capable of variable length transfers using the first byte or word to set the transfer length. When doing this, various options regarding how to count number of bytes to transfer are available.

Priority. The priority of the DMA transfers for the DMA channel in respect to the CPU and other DMA channels and access ports.

Trigger event. All DMA transfers are initiated by so-called DMA trigger events. This trigger either starts a DMA block transfer or a single DMA transfer. In addition to the configured trigger, a DMA channel can always be triggered by setting its designated `DMAREQ.DMAREQx` flag. The DMA trigger sources are described in Table 37 on page 92.

Source and Destination Increment. The source and destination addresses can be controlled to increment or decrement or not change, in order to give good flexibility for various types of transfers.

Transfer mode. The transfer mode determines whether the transfer should be a single transfer or a block transfer, or repeated versions of these.

Byte or word transfers. Determines whether each DMA transfer should be 8-bit (byte) or 16-bit (word).

Interrupt Mask. An interrupt request is generated upon completion of the DMA transfer. The interrupt mask bit controls if the interrupt generation is enabled or disabled.

M8: Decide whether to use seven or eight bits of length byte for transfer length. Only applicable when doing byte transfers.

A detailed description of all configuration parameters are given in the following sections.

13.2.2.1 Source Address

The address in XDATA memory where the DMA channel shall start to read data.

13.2.2.2 Destination Address

The first address to which the DMA channel should write the data read from the source address. The user must ensure that the destination is writable.

13.2.2.3 Transfer Count

The number of bytes/words needed to be transferred for the DMA transfer to be complete. When the transfer count is reached, the DMA controller rearms or disarms the DMA channel and alerts the CPU with an interrupt request. The transfer count can be defined in the configuration or it can be defined as a variable length described in the next section.

13.2.2.4 VLEN Setting

The DMA channel is capable of using the first byte or word (for word, bits 12:0 are used) in source data as the transfer length. This allows variable length transfers. When using variable length transfer, various options regarding how to count number of bytes to transfer is given. In any case, the transfer count (LEN) setting is used as maximum transfer count. Note that the M8 bit (see page 90) is only used when byte size transfers are chosen.

Options which can be set with VLEN are the following:

1. Transfer number of bytes/words commanded by first byte/word + 1 (transfers the length byte/word, and then as many bytes/words as dictated by length byte/word)
2. Transfer number of bytes/words commanded by first byte/word
3. Transfer number of bytes/words commanded by first byte/word + 2 (transfers the length byte/word, and then as many bytes/words as dictated by length byte/word + 1)
4. Transfer number of bytes/words commanded by first byte/word + 3 (transfers the length byte/word, and then as many bytes/words as dictated by length byte/word + 2)

Figure 18 shows the VLEN options.

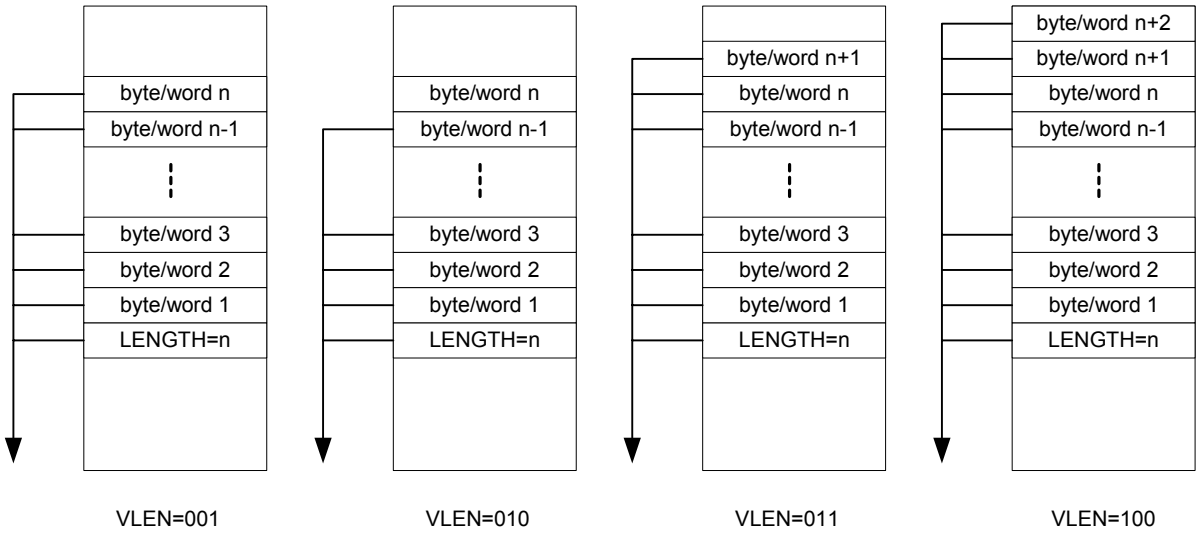


Figure 18: Variable Length (VLEN) Transfer Options

13.2.2.5 Trigger Event

Each DMA channel can be set up to sense on a single trigger. This field determines which trigger the DMA channel shall sense.

13.2.2.6 Source and Destination Increment

When the DMA channel is armed or rearmed the source and destination addresses are transferred to internal address pointers. The possibilities for address increment are :

- *Increment by zero.* The address pointer shall remain fixed after each transfer.
- *Increment by one.* The address pointer shall increment one count after each transfer.
- *Increment by two.* The address pointer shall increment two counts after each transfer.
- *Decrement by one.* The address pointer shall decrement one count after each transfer.

13.2.2.7 DMA Transfer Mode

The transfer mode determines how the DMA channel behaves when it starts transferring data. There are four transfer modes described below:

Single. On a trigger a single DMA transfer occurs and the DMA channel awaits the next trigger. After the number of transfers specified by the transfer count, are completed, the CPU is notified and the DMA channel is disarmed.

Block. On a trigger the number of DMA transfers specified by the transfer count is performed as quickly as possible, after which the CPU is notified and the DMA channel is disarmed.

Repeated single. On a trigger a single DMA transfer occurs and the DMA channel awaits the next trigger. After the number of transfers specified by the transfer count are completed, the CPU is notified and the DMA channel is rearmed.

Repeated block. On a trigger the number of DMA transfers specified by the transfer count is performed as quickly as possible, after which the CPU is notified and the DMA channel is rearmed.

13.2.2.8 DMA Priority

A DMA priority is associated with each DMA access port and is configurable for each DMA channel. The DMA priority is used to

determine the winner in the case of multiple simultaneous internal memory requests, and whether the DMA memory access should have priority or not over a simultaneous CPU memory access. In case of an internal tie, a round-robin scheme is used to ensure access for all. There are three levels of DMA priority:

High. Highest internal priority. DMA access will always prevail over CPU access.

Normal. Second highest internal priority. Guarantees that DMA access prevails over CPU on at least every second try.

Low. Lowest internal priority. DMA access will always defer to a CPU access.

13.2.2.9 Byte or Word transfers

Determines whether 8-bit (byte) or 16-bit (word) are done.

13.2.2.10 Interrupt mask

Upon completing a DMA transfer, the channel can generate an interrupt to the processor. This bit will mask the interrupt.

13.2.2.11 Mode 8 setting

This field determines whether to use 7 or 8 bits of length byte for transfer length. Only applicable when doing byte transfers.

13.2.3 DMA Configuration Setup

The DMA channel parameters such as address mode, transfer mode and priority described in the previous section have to be configured before a DMA channel can be armed and activated. The parameters are not configured directly through SFR registers, but instead they are written in a special DMA configuration data structure in memory. Each DMA channel in use requires its own DMA configuration data structure. The DMA configuration data structure consists of eight bytes and is described in section 13.2.6 on page 91. A DMA configuration data structure may reside at any location decided upon by the user software, and the address location is passed to the DMA controller through a set of SFRs DMAxCFGH:DMAxCFGL, Once a channel has been armed, the DMA controller will read the configuration data structure for that channel, given by the address in DMAxCFGH:DMAxCFGL.

It is important to note that the method for specifying the start address for the DMA configuration data structure differs between DMA channel 0 and DMA channels 1-4 as follows:

DMA0CFGH:DMA0CFGL gives the start address for DMA channel 0 configuration data structure.

DMA1CFGH:DMA1CFGL gives the start address for DMA channel 1 configuration data structure followed by channel 2-4 configuration data structures.

Thus the DMA controller expects the DMA configuration data structures for DMA channels 1-4 to lie in a contiguous area in memory starting at the address held in DMA1CFGH:DMA1CFGL and consisting of 32 bytes.

13.2.4 Stopping DMA Transfers

Ongoing DMA transfer or armed DMA channels will be aborted using the DMAARM register to disarm the DMA channel.

One or more DMA channels are aborted by writing the following to the DMAARM register.

- Writing a 1 to DMAARM.ABORT, and at the same time,
- Select which DMA channels to abort by setting the corresponding, DMAARM.DMAARMx bits.

An example of DMA channel arm and disarm is shown in Figure 19.

```
MOV DMAARM, #0x03   ; arm DMA channel 0 and 1

MOV DMAARM, #0x81   ; disarm DMA channel 0,
                    ; channel 1 is still armed
```

Figure 19: DMA arm/disarm example

13.2.5 DMA Interrupts

Each DMA channel can be configured to generate an interrupt to the CPU upon completing a DMA transfer. This is accomplished with the IRQMASK bit in the channel configuration. The corresponding interrupt flag in the DMAIRQ SFR register will be set when the interrupt is generated.

Regardless of the IRQMASK bit in the channel configuration, the interrupt flag will be set upon DMA channel complete. Thus software should always check (and clear) this register when rearming a channel with a changed IRQMASK setting. Failure to do so could generate an interrupt based on the stored interrupt flag.

13.2.6 DMA Configuration Data Structure

For each DMA channel, the DMA configuration data structure consists of eight bytes. The configuration data structure is described in Table 38.

DMA Trigger number	DMA Trigger name	Functional unit	Description
0	NONE	DMA	No trigger, setting DMAREQ.DMAREQx bit starts transfer
1	PREV	DMA	DMA channel is triggered by completion of previous channel
2	T1_CH0	Timer 1	Timer 1, compare, channel 0
3	T1_CH1	Timer 1	Timer 1, compare, channel 1
4	T1_CH2	Timer 1	Timer 1, compare, channel 2
5	T2_COMP	Timer 2	Timer 2, compare
6	T2_OVFL	Timer 2	Timer 2, overflow
7	T3_CH0	Timer 3	Timer 3, compare, channel 0
8	T3_CH1	Timer 3	Timer 3, compare, channel 1
9	T4_CH0	Timer 4	Timer 4, compare, channel 0
10	T4_CH1	Timer 4	Timer 4, compare, channel 1
11	ST	Sleep Timer	Sleep Timer compare
12	IOC_0	IO Controller	Port 0 I/O pin input transition
13	IOC_1	IO Controller	Port 1 I/O pin input transition
14	URX0	USART0	USART0 RX complete
15	UTX0	USART0	USART0 TX complete
16	URX1	USART1	USART1 RX complete
17	UTX1	USART1	USART1 TX complete
18	FLASH	Flash controller	Flash data write complete
19	RADIO	Radio	RF packet byte received/transmit
20	ADC_CHALL	ADC	ADC end of a conversion in a sequence, sample ready
21	ADC_CH11	ADC	ADC end of conversion channel 0 in sequence, sample ready
22	ADC_CH21	ADC	ADC end of conversion channel 1 in sequence, sample ready
23	ADC_CH32	ADC	ADC end of conversion channel 2 in sequence, sample ready
24	ADC_CH42	ADC	ADC end of conversion channel 3 in sequence, sample ready
25	ADC_CH53	ADC	ADC end of conversion channel 4 in sequence, sample ready
26	ADC_CH63	ADC	ADC end of conversion channel 5 in sequence, sample ready
27	ADC_CH74	ADC	ADC end of conversion channel 6 in sequence, sample ready
28	ADC_CH84	ADC	ADC end of conversion channel 7 in sequence, sample ready
29	ENC_DW	AES	AES encryption processor requests download input data
30	ENC_UP	AES	AES encryption processor requests upload output data

Table 37: DMA Trigger Sources

Table 38: DMA Configuration Data Structure

Byte Offset	Bit	Name	Description
0	7:0	SRCADDR[15:8]	The DMA channel source address, high
1	7:0	SRCADDR[7:0]	The DMA channel source address, low
2	7:0	DESTADDR[15:8]	The DMA channel destination address, high. Note that flash memory is not directly writeable.
3	7:0	DESTADDR[7:0]	The DMA channel destination address, low. Note that flash memory is not directly writeable.
4	7:5	VLEN[2:0]	Variable length transfer mode. In word mode, bits 12:0 of the first word is considered as the transfer length. 000/111 Use LEN for transfer count 001 Transfer the number of bytes/words specified by first byte/word + 1 (up to a maximum specified by LEN). Thus transfer count excludes length byte/word 010 Transfer the number of bytes/words specified by first byte/word (up to a maximum specified by LEN). Thus transfer count includes length byte/word. 011 Transfer the number of bytes/words specified by first byte/word + 2 (up to a maximum specified by LEN). 100 Transfer the number of bytes/words specified by first byte/word + 3 (up to a maximum specified by LEN). 101 reserved 110 reserved
4	4:0	LEN[12:8]	The DMA channel transfer count. Used as maximum allowable length when VLEN = 000/111. The DMA channel counts in words when in WORDSIZE mode, and in bytes otherwise.
5	7:0	LEN[7:0]	The DMA channel transfer count. Used as maximum allowable length when VLEN = 000/111. The DMA channel counts in words when in WORDSIZE mode, and in bytes otherwise.
6	7	WORDSIZE	Selects whether each DMA transfer shall be 8-bit (0) or 16-bit (1).
6	6:5	TMODE[1:0]	The DMA channel transfer mode: 00 : Single 01 : Block 10 : Repeated single 11 : Repeated block
6	4:0	TRIG[4:0]	Select DMA trigger to use 00000 : No trigger (writing to DMAREQ is only trigger) 00001 : The previous DMA channel finished 00010 – 11111 : Selects one of the triggers shown in Table 37. The trigger is selected in the order shown in the table.
7	7:6	SRCINC[1:0]	Source address increment mode (after each transfer): 00 : 0 bytes/words 01 : 1 bytes/words 10 : 2 bytes/words 11 : -1 bytes/words
7	5:4	DESTINC[1:0]	Destination address increment mode (after each transfer): 00 : 0 bytes/words 01 : 1 bytes/words 10 : 2 bytes/words 11 : -1 bytes/words

Byte Offset	Bit	Name	Description
7	3	IRQMASK	Interrupt Mask for this channel. 0 : Disable interrupt generation 1 : Enable interrupt generation upon DMA channel done
7	2	M8	Mode of 8th bit for VLEN transfer length; only applicable when WORDSIZE=0. 0 : Use all 8 bits for transfer count 1 : Use 7 LSB for transfer count
7	1:0	PRIORITY[1:0]	The DMA channel priority: 00 : Low, CPU has priority. 01 : Guaranteed, DMA at least every second try. 10 : High, DMA has priority 11 : Highest, DMA has priority. Reserved for DMA port access.

13.2.7 DMA registers

This section describes the SFR registers associated with the DMA Controller

DMAARM (0xD6) – DMA Channel Arm

Bit	Name	Reset	R/W	Description
7	ABORT	0	R0/W	DMA abort. This bit is used to stop ongoing DMA transfers. Writing a 1 to this bit will abort all channels which are selected by setting the corresponding DMAARM bit to 1 0 : Normal operation 1 : Abort channels all selected channels
6:5	-	00	R/W	Not used
4	DMAARM4	0	R/W	DMA arm channel 4 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
3	DMAARM3	0	R/W	DMA arm channel 3 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
2	DMAARM2	0	R/W	DMA arm channel 2 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
1	DMAARM1	0	R/W	DMA arm channel 1 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
0	DMAARM0	0	R/W	DMA arm channel 0 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.

DMAREQ (0xD7) – DMA Channel Start Request and Status

Bit	Name	Reset	R/W	Description
7:5	-	000	R0	Not used
4	DMAREQ4	0	R/W1 H0	DMA transfer request, channel 4 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
3	DMAREQ3	0	R/W1 H0	DMA transfer request, channel 3 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
2	DMAREQ2	0	R/W1 H0	DMA transfer request, channel 2 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
1	DMAREQ1	0	R/W1 H0	DMA transfer request, channel 1 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
0	DMAREQ0	0	R/W1 H0	DMA transfer request, channel 0 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.

DMA0CFGH (0xD5) – DMA Channel 0 Configuration Address High Byte

Bit	Name	Reset	R/W	Description
7:0	DMA0CFG[15:8]	0x00	R/W	The DMA channel 0 configuration address, high order

DMA0CFGL (0xD4) – DMA Channel 0 Configuration Address Low Byte

Bit	Name	Reset	R/W	Description

Bit	Name	Reset	R/W	Description
7:0	DMA0CFG[7:0]	0x00	R/W	The DMA channel 0 configuration address, low order

DMA1CFGH (0xD3) – DMA Channel 1-4 Configuration Address High Byte

Bit	Name	Reset	R/W	Description
7:0	DMA1CFG[15:8]	0x00	R/W	The DMA channel 1-4 configuration address, high order

DMA1CFGL (0xD2) – DMA Channel 1-4 Configuration Address Low Byte

Bit	Name	Reset	R/W	Description
7:0	DMA1CFG[7:0]	0x00	R/W	The DMA channel 1-4 configuration address, low order

DMAIRQ (0xD1) – DMA Interrupt Flag

Bit	Name	Reset	R/W	Description
7:5	-	000	R/W0	Not used
4	DMAIF4	0	R/W0	DMA channel 4 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
3	DMAIF3	0	R/W0	DMA channel 3 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
2	DMAIF2	0	R/W0	DMA channel 2 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
1	DMAIF1	0	R/W0	DMA channel 1 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
0	DMAIF0	0	R/W0	DMA channel 0 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending

13.3 16-bit Timer, Timer1

Timer 1 is an independent 16-bit timer which supports typical timer/counter functions such as input capture, output compare and PWM functions. The timer has three independent capture/compare channels. The timer uses one I/O pin per channel. The timer is used for a wide range of control and measurement applications and the availability of up/down count mode with three channels will for example allow implementation of motor control applications.

The features of Timer 1 are as follows:

- Three capture/compare channels
- Rising, falling or any edge input capture
- Set, clear or toggle output compare
- Free-running, modulo or up/down counter operation
- Clock prescaler for divide by 1, 8, 32 or 128
- Interrupt request generated on each capture/compare and terminal count
- Capture triggered by radio
- DMA trigger function

13.3.1 16-bit Timer Counter

The timer consists of a 16-bit counter that increments or decrements at each active clock edge. The period of the active clock edges is defined by the register bits CLKCON.TICKSPD which sets the global division of the system clock giving a variable clock tick frequency from 0.25 MHz to 32 MHz. This is further divided in Timer 1 by the prescaler value set by T1CTL.DIV. This prescaler value can be from 1 to 128. Thus the lowest clock frequency used by Timer 1 is 1953.125 Hz and the highest is 32 MHz when the 32 MHz crystal oscillator is used as system clock source. When the 16 MHz RC oscillator is used as system clock source then the highest clock frequency used by Timer 1 is 16 MHz.

The counter operates as either a free-running counter, a modulo counter or as an up/down counter for use in centre-aligned PWM.

It is possible to read the 16-bit counter value through the two 8-bit SFRs; T1CNTH and T1CNTL, containing the high-order byte and low-order byte respectively. When the T1CNTL is read, the high-order byte of the counter at that instant is buffered in T1CNTH so that the high-order byte can be read from T1CNTH. Thus T1CNTL shall always be read first before reading T1CNTH.

All write accesses to the T1CNTL register will reset the 16-bit counter.

The counter produces an interrupt request when the terminal count value (overflow) is reached. It is possible to clear and halt the counter with T1CTL control register settings. The counter is started when a value other than 00 is written to T1CTL.MODE. If 00 is written to T1CTL.MODE the counter halts at its present value.

13.3.2 Timer 1 Operation

In general, the control register T1CTL is used to control the timer operation. The various modes of operation are described below.

13.3.3 Free-running Mode

In the free-running mode of operation the counter starts from 0x0000 and increments at each active clock edge. When the counter reaches 0xFFFF the counter is loaded with 0x0000 and continues incrementing its value as shown in Figure 20. When the terminal count value 0xFFFF is reached, the flag T1CTL.OVFIF is set. An interrupt request is generated if the corresponding interrupt mask bit TIMIF.OVFIM is set. The free-running mode can be used to generate independent time intervals and output signal frequencies.

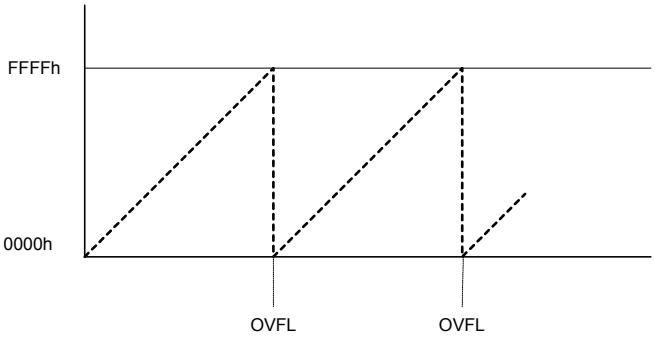


Figure 20: Free-running mode

13.3.4 Modulo Mode

When the timer operates in modulo mode the 16-bit counter starts at 0x0000 and increments at each active clock edge. When the counter reaches the terminal count value held in registers T1CC0H:T1CC0L, the counter is reset to 0x0000 and continues to increment. The flag T1CTL.OVFIF is set when the terminal count value is reached. An interrupt request is generated if the corresponding interrupt mask bit TIMIF.OVFIM is set. The modulo mode can be used for applications where a period other then 0xFFFF is required. The counter operation is shown in Figure 21.

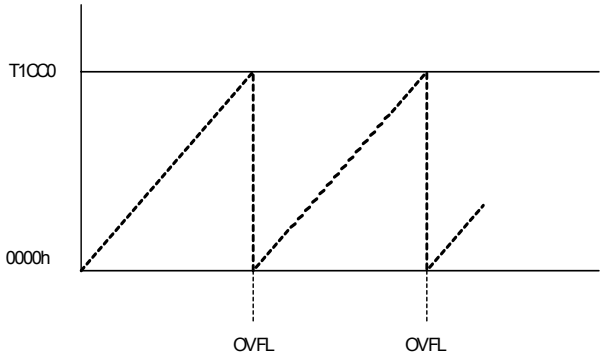


Figure 21: Modulo mode

13.3.5 Up/down Mode

In the up/down timer mode, the counter repeatedly starts from 0x0000 and counts up until the value held in T1CC0H:T1CC0L is reached and then the counter counts down until 0x0000 is reached as shown in Figure 22. This timer mode is used when symmetrical output pulses are required with a period other than 0xFFFF, and therefore allows implementation of centre-aligned PWM output applications. The flag T1CTL.OVFIF is not set when the terminal count value is reached in the up/down mode.

Clearing the counter by writing to T1CNTL will also reset the count direction to the count up from 0x0000 mode.

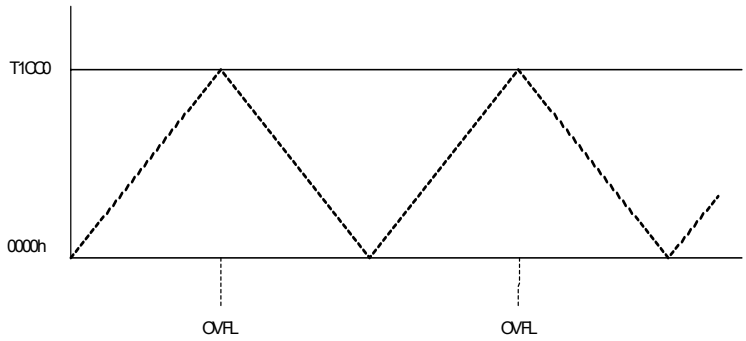


Figure 22 : Up/down mode

13.3.6 Channel Mode Control

The channel mode is set with each channel's control and status register T1CCTLn. The settings include input capture and output compare modes.

13.3.7 Input Capture Mode

When a channel is configured as an input capture channel, the I/O pin associated with that channel, is configured as an input. After the timer has been started, a rising edge, falling edge or any edge on the input pin will trigger a capture of the 16-bit counter contents into the associated capture register. Thus the timer is able to capture the time when an external event takes place.

Note: before an I/O pin can be used by the timer, the required I/O pin must be configured as a Timer 1 peripheral pin as described in section 13.1.3 on page 70 .

The channel input pin is synchronized to the internal system clock. Thus pulses on the input pin must have a minimum duration greater than the system clock period.

The contents of the 16-bit capture register is read out from registers T1CCnH:T1CCnL.

When the capture takes place the interrupt flag for the channel is set. This bit is T1CTL.CH0IF for channel 0, T1CTL.CH1IF for channel 1, and T1CTL.CH2IF for channel 2. An interrupt request is generated if the corresponding interrupt mask bit on T1CCTL0.IM, T1CCTL1.IM, or T1CCTL2.IM, respectively, is set.

13.3.8 RF Event Capture

Each timer channel may be configured so that an RF interrupt event will trigger a capture instead of the normal input pin capture. This function is selected with the register bit T1CCTLx.CPSEL, which selects to use either the input pin or the RF interrupt as capture event. When RF is selected as capture input, the interrupt source(s) enabled by RFIM (see section 14.4.1 on page 166) will trigger a capture. In this way the timer can be used to capture a value when e.g. a start of frame delimiter (SFD) is detected.

13.3.9 Output Compare Mode

In output compare mode the I/O pin associated with a channel is set as an output. After the timer has been started, the contents of the counter is compared with the contents of the channel compare register T1CCnH:T1CCnL. If the compare register equals the counter contents, the output pin is set, reset or toggled according to the compare output mode setting of T1CCTLn.CMP. Note that all edges on output pins are glitch-free when operating in a given output compare mode. Writing to the compare register T1CCnL is buffered so that a value written to T1CCnL does not take effect until the corresponding high order register, T1CCnH is written. For output compare modes 1-3, a new value written to the compare register T1CCnH:T1CCnL takes effect after the registers have been written. For other output compare modes the new value written to the compare register take effect when the timer reaches 0x0000.

Note that channel 0 has fewer output compare modes because T1CC0H:T1CC0L has a special function in modes 6 and 7, meaning these modes would not be useful for channel 0.

When a compare occurs, the interrupt flag for the channel is set. This bit is T1CTL.CH0IF for channel 0, T1CTL.CH1IF for channel 1,

and T1CTL.CH2IF for channel 2. An interrupt request is generated if the corresponding interrupt mask bit on T1CCTL0.IM, T1CCTL1.IM, or T1CCTL2.IM, respectively, is set.

Examples of output compare modes in various timer modes are given in the following figures.

Edge-aligned PWM output signals can be generated using the timer modulo mode and channels 1 and 2 in output compare mode 6 or 7 as shown in Figure 23. The period of the PWM signal is determined by the setting T1CC0 and the duty cycle for the channel output is determined by T1CCn. The timer free-running mode may also be used. In this case CLKCON.TICKSPD and the prescaler divider value T1CTL.DIV set the period of the PWM signal. The polarity of the PWM signal is determined by whether output compare mode 6 or 7 is used. PWM output signals can also be generated using output compare modes 4 and 5 as shown in the same figure, or by using modulo mode as shown in Figure 24. Using output compare mode 4 and 5 is preferred for simple PWM.

Centre-aligned PWM outputs can be generated when the timer up/down mode is selected. The channel output compare mode 4 or 5 is selected depending on required polarity of the PWM signal. The period of the PWM signal is determined by T1CC0 and the duty cycle for the channel output is determined by T1CCn.

The centre-aligned PWM mode is required by certain types of motor drive applications and typically less noise is produced than the edge-aligned PWM mode because the I/O pin transitions are not lined up on the same clock edge.

In some types of applications, a defined delay or dead time is required between outputs. Typically this is required for outputs driving an H-bridge configuration to avoid uncontrolled cross-conduction in one side of the H-bridge. The delay or dead-time can be obtained in the PWM outputs by using T1CCn as shown in the following:

Assuming that channel 1 and channel 2 are used to drive the outputs using timer up/down mode and the channels use output compare modes 4 and 5 respectively, then the timer period (in Timer 1 clock periods) is:

T_P = T1CC0 x 2

and the dead time, i.e. the time when both outputs are low, (in Timer 1 clock periods) is given by:

T_D = T1CC1 – T1CC2

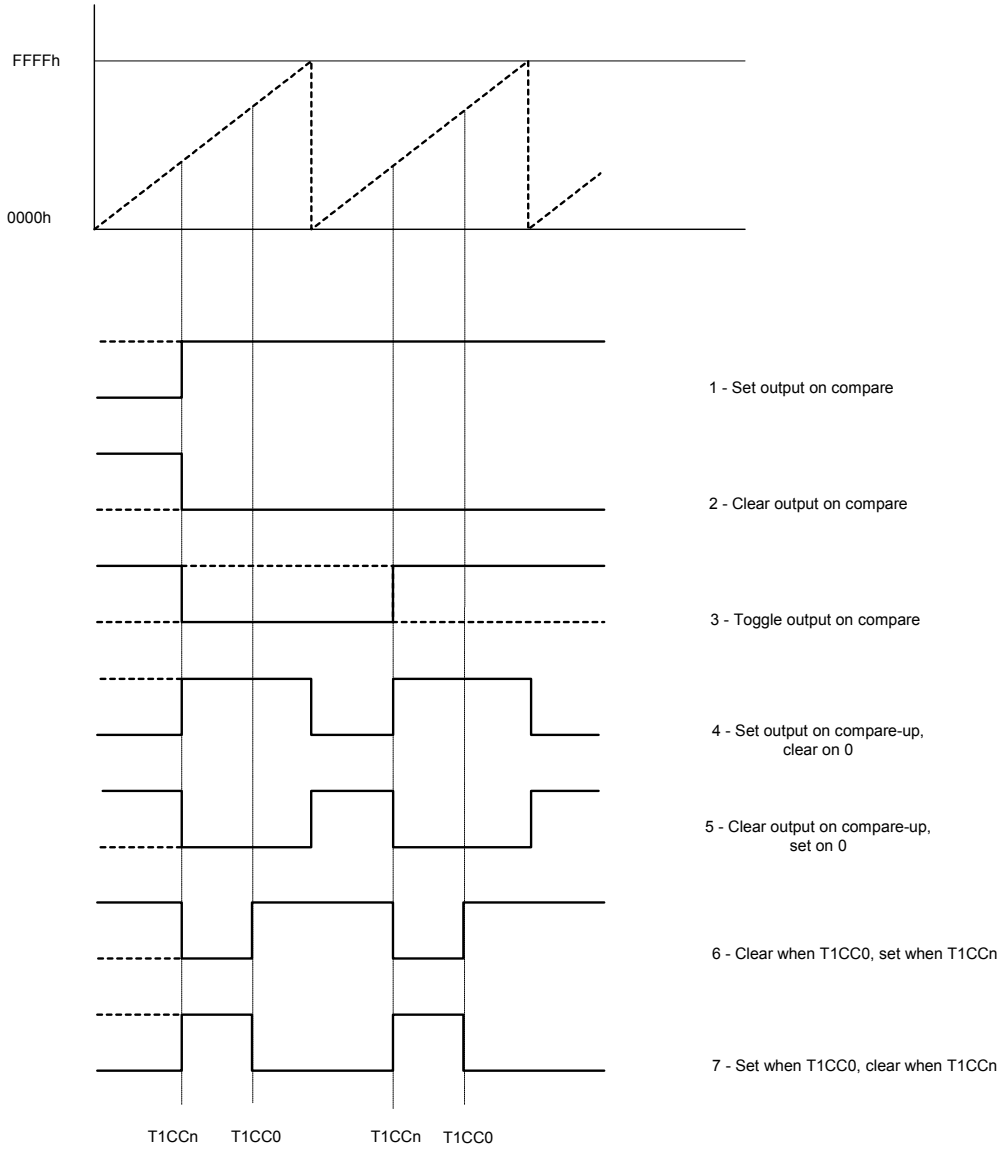


Figure 23: Output compare modes, timer free-running mode

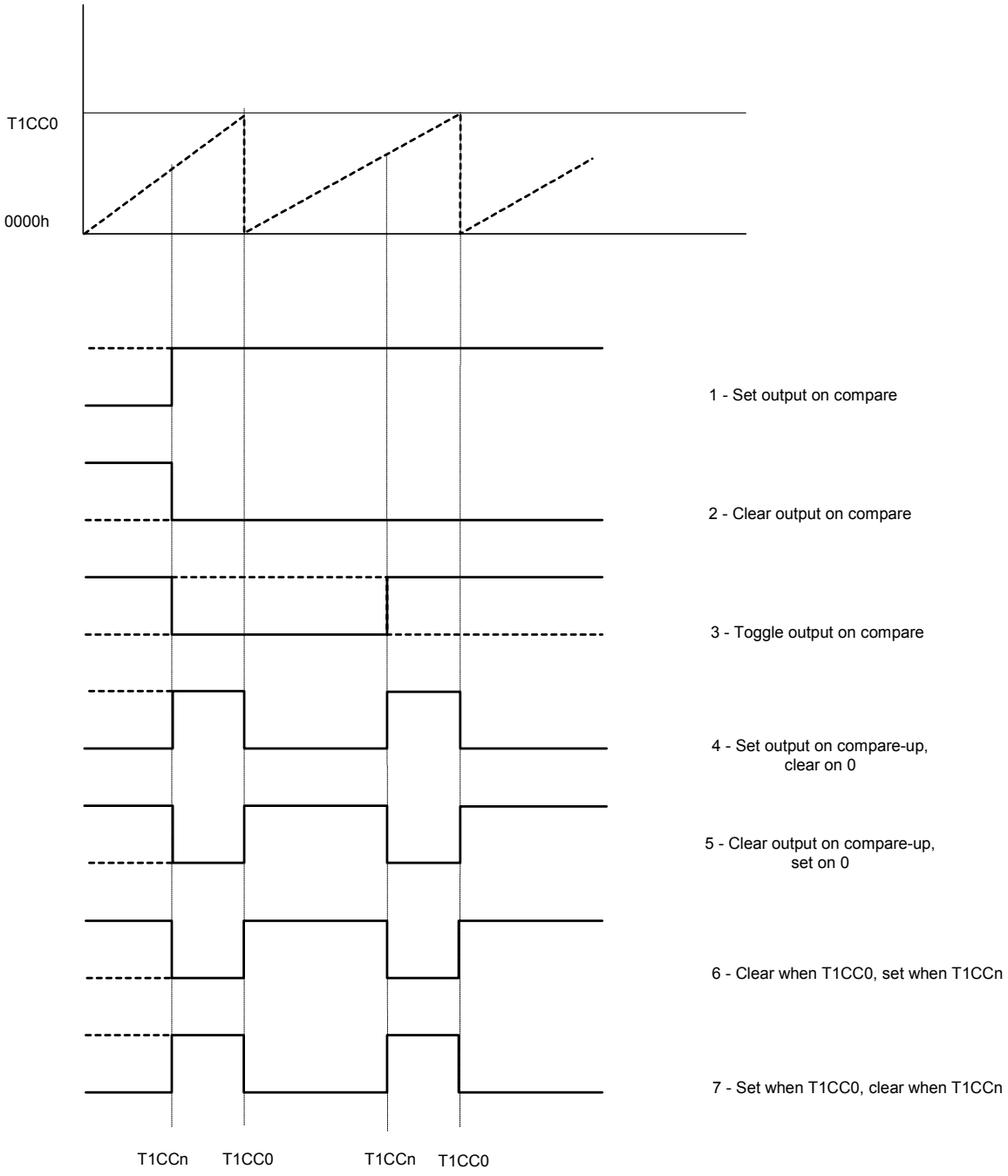


Figure 24: Output compare modes, timer modulo mode

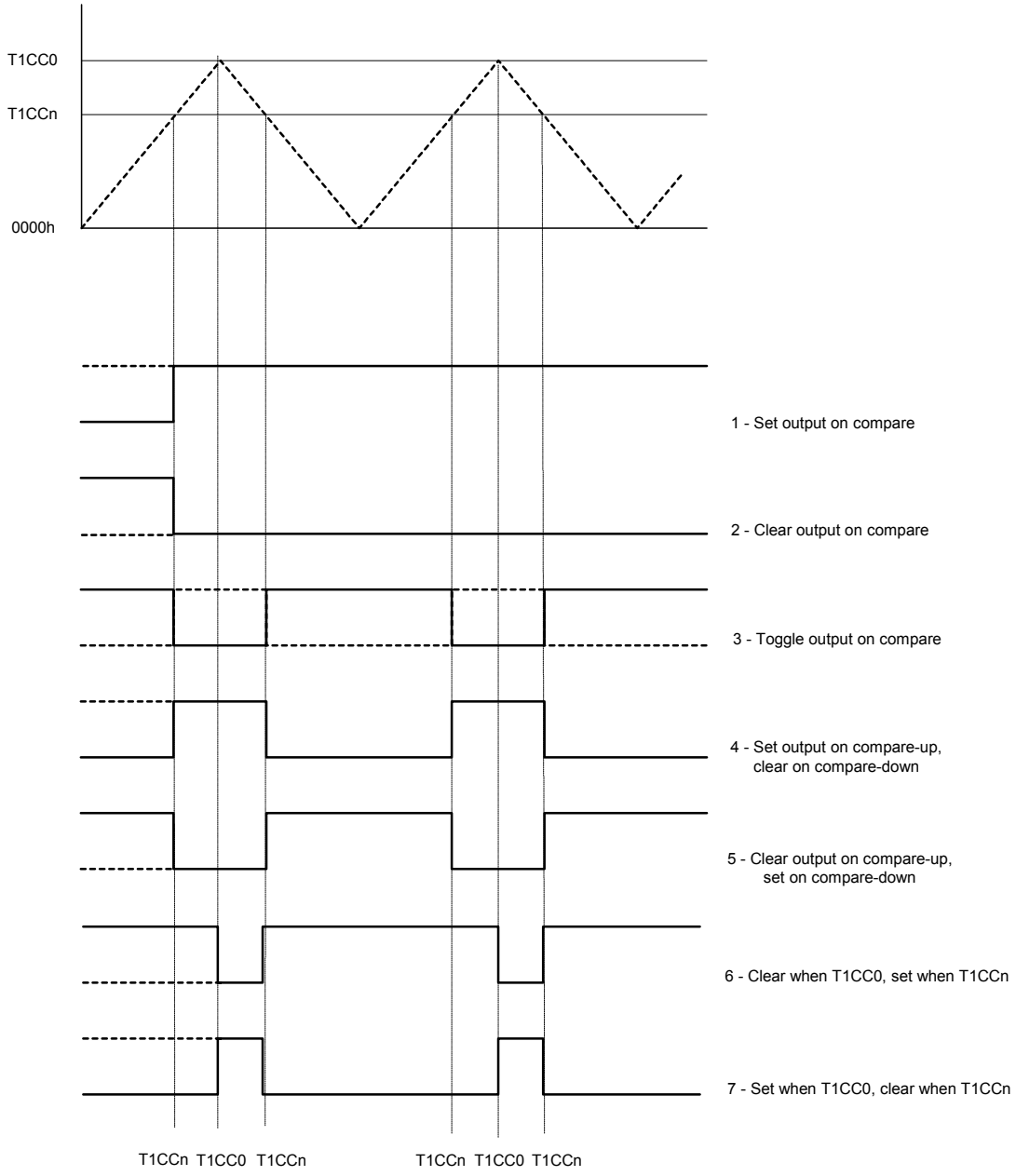


Figure 25: Output modes, timer up/down mode

13.3.10 Timer 1 Interrupts

There is one interrupt vector assigned to the timer. An interrupt request is generated when one of the following timer events occur:

- Counter reaches terminal count value.
- Input capture event.
- Output compare event

The register bits `T1CTL.OVFIF`, `T1CTL.CH0IF`, `T1CTL.CH1IF`, and `T1CTL.CH2IF` contains the interrupt flags for the terminal count value event, and the three channel compare/capture events, respectively. An interrupt request is only generated when the corresponding interrupt mask bit is set. The interrupt mask bits are `T1CCTL0.IM`, `T1CCTL1.IM`, `T1CCTL2.IM` and `TIMIF.OVFIM`. If there are other pending

interrupts, the corresponding interrupt flag must be cleared by software before a new interrupt request is generated. Also, enabling an interrupt mask bit will generate a new interrupt request if the corresponding interrupt flag is set.

13.3.11 Timer 1 DMA Triggers

There are three DMA triggers associated with Timer 1. These are DMA triggers T1_CH0, T1_CH1 and T1_CH2 which are generated on timer compare events as follows:

- T1_CH0 – channel 0 compare
- T1_CH1 – channel 1 compare
- T1_CH2 – channel 2 compare

13.3.12 Timer 1 Registers

This section describes the Timer 1 registers which consist of the following registers:

- `T1CNTH` – Timer 1 Count High
- `T1CNTL` – Timer 1 Count Low
- `T1CTL` – Timer 1 Control and Status
- `T1CCTLx` – Timer 1 Channel x Capture/Compare Control
- `T1CCxH` – Timer 1 Channel x Capture/Compare Value High
- `T1CCxL` – Timer 1 Channel x Capture/Compare Value Low

T1CNTH (0xE3) – Timer 1 Counter High

Bit	Name	Reset	R/W	Description
7:0	CNT[15:8]	0x00	R	Timer count high order byte. Contains the high byte of the 16-bit timer counter buffered at the time T1CNTL is read.

T1CNTL (0xE2) – Timer 1 Counter Low

Bit	Name	Reset	R/W	Description
7:0	CNT[7:0]	0x00	R/W	Timer count low order byte. Contains the low byte of the 16-bit timer counter. Writing anything to this register results in the counter being cleared to 0x0000.

T1CTL (0xE4) – Timer 1 Control and Status

Bit	Name	Reset	R/W	Description
7	CH2IF	0	R/W0	Timer 1 channel 2 interrupt flag. Set when the channel 2 interrupt condition occurs. Writing a 1 has no effect.
6	CH1IF	0	R/W0	Timer 1 channel 1 interrupt flag. Set when the channel 1 interrupt condition occurs. Writing a 1 has no effect.
5	CH0IF	0	R/W0	Timer 1 channel 0 interrupt flag. Set when the channel 0 interrupt condition occurs. Writing a 1 has no effect.
4	OVFIF	0	R/W0	Timer 1 counter overflow interrupt flag. Set when the counter reaches the terminal count value in free-running or modulo mode. Writing a 1 has no effect.
3:2	DIV[1:0]	00	R/W	Prescaler divider value. Generates the active clock edge used to update the counter as follows: 00 Tick frequency/1 01 Tick frequency/8 10 Tick frequency/32 11 Tick frequency/128
1:0	MODE[1:0]	00	R/W	Timer 1 mode select. The timer operating mode is selected as follows: 00 Operation is suspended 01 Free-running, repeatedly count from 0x0000 to 0xFFFF 10 Modulo, repeatedly count from 0x0000 to T1CC0 11 Up/down, repeatedly count from 0x0000 to T1CC0 and from T1CC0 down to 0x0000

T1CCTL0 (0xE5) – Timer 1 Channel 0 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	CPSEL	0	R/W	Capture select. Timer 1 channel 0 captures on RF interrupt from RF transceiver or capture input pin. 0 Use normal capture input 1 Use RF interrupt from RF transceiver for capture
6	IM	1	R/W	Channel 0 interrupt mask. Enables interrupt request when set.
5:3	CMP[2:0]	000	R/W	Channel 0 compare mode select. Selects action on output when timer value equals compare value in T1CC0 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Not used 110 Not used 111 Not used
2	MODE	0	R/W	Mode. Select Timer 1 channel 0 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 0 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T1CC0H (0xDB) – Timer 1 Channel 0 Capture/Compare Value High

Bit	Name	Reset	R/W	Description
7:0	T1CC0[15:8]	0x00	R/W	Timer 1 channel 0 capture/compare value, high order byte

T1CC0L (0xDA) – Timer 1 Channel 0 Capture/Compare Value Low

Bit	Name	Reset	R/W	Description
7:0	T1CC0[7:0]	0x00	R/W	Timer 1 channel 0 capture/compare value, low order byte

T1CCTL1 (0xE6) – Timer 1 Channel 1 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	CPSEL	0	R/W	Capture select. Timer 1 channel 1 captures on RF interrupt from RF transceiver or capture input pin 0 Use normal capture input 1 Use RF interrupt from RF transceiver for capture
6	IM	1	R/W	Channel 1 interrupt mask. Enables interrupt request when set.
5:3	CMP[2:0]	000	R/W	Channel 1 compare mode select. Selects action on output when timer value equals compare value in T1CC1 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Clear when equal T1CC0, set when equal T1CC1 110 Set when equal T1CC0, clear when equal T1CC1 111 Not used
2	MODE	0	R/W	Mode. Select Timer 1 channel 1 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 1 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T1CC1H (0xDD) – Timer 1 Channel 1 Capture/Compare Value High

Bit	Name	Reset	R/W	Description
7:0	T1CC1[15:8]	0x00	R/W	Timer 1 channel 1 capture/compare value, high order byte

T1CC1L (0xDC) – Timer 1 Channel 1 Capture/Compare Value Low

Bit	Name	Reset	R/W	Description
7:0	T1CC1[7:0]	0x00	R/W	Timer 1 channel 1 capture/compare value, low order byte

T1CCTL2 (0xE7) – Timer 1 Channel 2 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	CPSEL	0	R/W	Capture select. Timer 1 channel 2 captures on RF interrupt from RF transceiver or capture input pin 0 Use normal capture input 1 Use RF interrupt from RF transceiver for capture
6	IM	1	R/W	Channel 2 interrupt mask. Enables interrupt request when set.
5:3	CMP[2:0]	000	R/W	Channel 2 compare mode select. Selects action on output when timer value equals compare value in T1CC2 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Clear when equal T1CC0, set when equal T1CC2 110 Set when equal T1CC0, clear when equal T1CC2 111 Not used
2	MODE	0	R/W	Mode. Select Timer 1 channel 2 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 2 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T1CC2H (0xDF) – Timer 1 Channel 2 Capture/Compare Value High

Bit	Name	Reset	R/W	Description
7:0	T1CC2[15:8]	0x00	R/W	Timer 1 channel 2 capture/compare value, high order byte

T1CC2L (0xDE) – Timer 1 Channel 2 Capture/Compare Value Low

Bit	Name	Reset	R/W	Description
7:0	T1CC2[7:0]	0x00	R/W	Timer 1 channel 2 capture/compare value, low order byte

The TIMIF.OVFIM register bit resides in the TIMIF register, which is described together with timer 3 and timer 4

13.4 MAC Timer (Timer 2)

The MAC Timer is mainly used to provide timing for 802.15.4 CSMA-CA algorithms and for general timekeeping in the 802.15.4 MAC layer. When the MAC Timer is used together with the Sleep Timer described in section 13.5, the timing function is provided even when the system enters low-power modes.

The main features of the MAC Timer are the following:

- 16-bit timer up-counter providing symbol/frame period: 16µs/320µs
- Adjustable period with accuracy 31.25 ns
- 8-bit timer compare function
- 20-bit overflow count
- 20-bit overflow count compare function
- Start of Frame Delimiter capture function.
- Timer start/stop synchronous with 32.768 kHz clock and timekeeping maintained by Sleep Timer.
- Interrupts generated on compare and overflow
- DMA trigger capability

13.4.1 Timer Operation

This section describes the operation of the timer.

13.4.1.1 General

After a reset the timer is in the timer IDLE mode where it is stopped. The timer starts running when `T2CNF.RUN` is set to 1. The timer will then enter the timer RUN mode. The entry is either immediate or it is performed synchronous with the 32.768 kHz clock. See section 13.4.4 for a description of the synchronous start and stop mode.

Once the timer is running in RUN mode, it can be stopped by writing a 0 to `T2CNF.RUN`. The timer will then enter the timer IDLE mode. The stopping of the timer is performed either immediately or it is performed synchronous with the 32.768 kHz clock

13.4.1.2 Up Counter

The MAC Timer contains a 16-bit timer, which increments during each clock cycle.

13.4.1.3 Timer overflow

When the timer is about to *count* to a value that is equal to or greater than the timer period set by registers `T2CAPHPH:T2CAPLPL`, a timer overflow occurs. When the timer overflow occurs, the timer value is set to the difference between the value it is about to count to and the timer period during the next clock cycle. If the overflow interrupt mask bit `T2PEROF2.PERIM` is 1, an interrupt request is generated. The interrupt flag bit `T2CNF.PERIF` is set to 1 regardless of the interrupt mask value.

13.4.1.4 Timer delta increment

The timer period may be adjusted once during a timer period by writing a timer delta value. When a timer delta value is written to the registers `T2THD:T2TLD`, the 16-bit timer halts at its current value and a delta counter starts counting. The delta counter starts counting from the delta value written, down to zero. Once the delta counter reaches zero, the 16-bit timer starts counting again.

The delta counter decrements by the same rate as the timer. When the delta counter has reached zero it will not start counting again until the delta value is written once again. In this way a timer period may be increased by the delta value in order to make adjustments to the timer overflow events over time.

13.4.1.5 Timer Compare

A timer compare occurs when the timer is about to *count* to a value that is equal or greater than the 8-bit compare value held in the `T2CMP` register. Note that the compare value is only 8 bits so the compare is made between the compare value and either the most significant byte or the least significant byte of the timer.

The selection of which part of the timer is to be compared is set by the T2CNF.CMSEL bit.

When a timer compare occurs the interrupt flag T2CNF.CMPIF is set to 1. An interrupt request is also generated if the interrupt mask T2PEROF2.CMPIM is set to 1.

13.4.1.6 Capture Input

The MAC timer has a timer capture function which captures at the time when the start of frame delimiter (SFD) status in the radio goes high. Refer to sections 14.6 and 14.9 starting on page 170 for a description of the SFD.

When the capture event occurs the current timer value will be captured into the capture register. The capture value can be read from the registers T2CAPHPH:T2CAPLPL. The value of the overflow count is also captured (see section 13.4.1.7) at the time of the capture event and can be read from the registers T2PEROF2:T2PEROF1:T2PEROF0.

13.4.1.7 Overflow count

At each timer overflow, the 20-bit overflow counter is incremented by 1. The overflow counter value is read through the SFR registers T2OF2:T2OF1:T2OF0. Note that the register contents in T2OF2:T2OF1 is latched when T2OF0 is read, meaning that T2OF0 must always be read first.

Overflow count update

The overflow count value may be updated by writing to the registers T2OF2:T2OF1:T2OF0 when the timer is IDLE.

Overflow count increment selection

The increment value for the overflow counter can be set *once* by writing to the T2OF2:T2OF1:T2OF0 registers when the timer is in the RUN state. The value written to these registers will be added to the normal increment of 1 at the time of the next overflow count increment i.e. at the next timer overflow. The overflow count increment will return to 1 at the following increment.

13.4.1.8 Overflow count compare

A compare value may be set for the overflow counter. The compare value is set by writing to T2PEROF2:T2PEROF1:T2PEROF0.
When the overflow count value is equal or greater than the set compare value an overflow compare event occurs. If the overflow compare interrupt mask bit T2PEROF2.OFCMPIM is 1, an interrupt request is generated. The interrupt flag bit T2CNF.OFCMPIF is set to 1 regardless of the interrupt mask value.

13.4.2 Interrupts

The Timer has three individually maskable interrupt sources. These are the following:

- Timer overflow
- Timer compare
- Overflow count compare

The interrupt flags are given in the T2CNF registers. The interrupt flag bits are set only by hardware and may be cleared only by writing to the SFR register.

Each interrupt source may be masked by the mask bits in the T2PEROF2 register. An interrupt is generated when the corresponding mask bit is set, otherwise the interrupt will not be generated. The interrupt flag bit is set, however disregarding the state of the interrupt mask bit.

13.4.3 DMA Triggers

Timer 2 can generate two DMA triggers – T2_COMP and T2_OVFL which are activated as follows:

- T2_COMP: Timer 2 compare event
- T2_OVFL: Timer 2 overflow event

13.4.4 Timer start/stop synchronization

This section describes the synchronized timer start and stop.

13.4.4.1 General

The Timer can be started and stopped synchronously with the 32.768 kHz clock rising edge. Note this event is derived from a 32.768 kHz clock signal, but is synchronous with the 32 MHz system

clock and thus has a period approximately equal the 32.768 kHz clock period.

At the time of a synchronous start the timer is reloaded with new calculated values for the timer and overflow count such that it appears that the timer has not been stopped.

13.4.4.2 Timer synchronous stop

After the timer has started running, i.e. entered timer RUN mode it is stopped synchronously by writing 0 to `T2CNF.RUN` when `T2CNF.SYNC` is 1. After `T2CNF.RUN` has been set to 0, the timer will continue running until the 32.768 kHz clock rising edge is sampled as 1. When this occurs the timer is stopped and the current Sleep timer value is stored.

13.4.4.3 Timer synchronous start

When the timer is in the IDLE mode it is started synchronously by writing 1 to `T2CNF.RUN` when `T2CNF.SYNC` is 1. After `T2CNF.RUN` has been set to 1, the timer will remain in the IDLE mode until the 32.768 kHz clock rising edge is detected. When this occurs the timer will first calculate new values for the 16-bit timer value and for the 20-bit timer overflow count, based on the current and stored Sleep timer values and the current 16-bit timer values. The new MAC Timer and overflow count values are loaded into the timer and the timer enters the RUN mode. This synchronous start process takes 75 clock cycles from the time when the 32.768 kHz clock rising edge is sampled high. The synchronous start and stop function requires that the system clock frequency is selected to be 32 MHz. If the 16 MHz clock is selected, there will be an offset added to the new calculated value.

The method for calculating the new MAC Timer value and overflow count value is given below. Due to the fact that the MAC Timer clock and Sleep timer clocks are asynchronous with a non-integer clock ratio there will be an error of maximum ±1 in calculated timer value compared to the ideal timer value.

Calculation of new timer value and overflow count value

$$N_c = CurrentSleepTimerValue$$

$$N_s = StoredSleepTimerValue$$

$$K_{ck} = ClockRatio = 976.5625\ ^{4}$$

$$stw = SleepTimerWidth = 24$$

$$P = Timer2Period$$

$$O_c = CurrentOverflowCountValue$$

$$T_c = CurrentTimerValue$$

$$T_{OH} = Overhead = 75$$

$$N_t = N_c - N_s$$

$$N_t \le 0 \Rightarrow N_d = 2^{stw} + N_t; N_t > 0 \Rightarrow N_d = N_t$$

$C = N_d \cdot K_{ck} + T_C + T_{OH}$ (Rounded to nearest integer value)

$$T = C \bmod P$$

$$O = \frac{(C - T)}{P} + O_C$$

$$Timer2Value = T$$

$$Timer2OverflowCount = O$$

13.4.5 Timer 2 Registers

The SFR registers associated with Timer 2 are listed in this section. These registers are the following:

- `T2CNF` – Timer 2 Configuration
- `T2HD` – Timer 2 Count/Delta High
- `T2LD` – Timer 2 Count/Delta Low
- `T2CMP` – Timer 2 Compare
- `T2OF2` – Timer 2 Overflow Count 2
- `T2OF1` – Timer 2 Overflow Count 1
- `T2OF0` – Timer 2 Overflow Count 0

[4] Clock ratio of MAC Timer clock frequency (32 MHz) and Sleep timer clock frequency (32.768 kHz)

- T2CAPHPH – Timer 2 Capture/Period High
- T2CAPLPL – Timer 2 Capture/Period Low
- T2PEROF2 – Timer 2 Overflow Compare/Capture 2
- T2PEROF1 – Timer 2 Overflow Compare/Capture 1
- T2PEROF0 – Timer 2 Overflow Compare/Capture 0

T2CNF (0xC3) – Timer 2 Configuration

Bit	Name	Reset	R/W	Description
7	CMPIF	0	R/W0	Timer compare interrupt flag. This bit is set to 1 when a timer compare event occurs. Cleared by software only. Writing a 1 to this bit has no effect.
6	PERIF	0	R/W0	Overflow interrupt flag. This bit is set to 1 when a period event occurs. Cleared by software only. Writing a 1 to this bit has no effect.
5	OFCMPIF	0	R/W	Overflow compare interrupt flag. This bit is set to 1 when a overflow compare occurs. Cleared by software only. Writing a 1 to this bit has no effect.
4	–	0	R0	Not used. Read as 0
3	CMSEL	0	R/W	Timer compare source select. 0 Compare with 16-bit Timer bits [15:8] 1 Compare with 16-bit Timer bits [7:0]
2	–	0	R/W	Reserved. Always set to 0
1	SYNC	1	R/W	Enable synchronized start and stop. 0 start and stop of timer is immediate 1 start and stop of timer is synchronized with 32.768 kHz edge and new timer values are reloaded.
0	RUN	0	R/W	Start timer. Writing this bit shall start or stop the timer. When reading this bit the current state of the timer is returned. 0 stop timer (IDLE state) 1 start timer (RUN state)

T2THD (0xA7) – Timer 2 Timer Value High Byte

Bit	Name	Reset	R/W	Description
7:0	THD[7:0]	0x00	R/W	The value read from this register is the high-order byte of the timer value. The high-order byte read is from timer value at the last instant when T2TLD was read. The value written to this register while the timer is running is the high-order byte of the timer delta counter value. The low-order byte of this value is the value last written to T2TLD. The timer will halt for delta clock cycles. The value written to this register while the timer is idle will be written to the high-order byte of the timer.

T2TLD (0xA6) – Timer 2 Timer Value Low Byte

Bit	Name	Reset	R/W	Description
7:0	TLD[7:0]	0x00	R/W	The value read from this register is the low-order byte of the timer value. The value written to this register while the timer is running is the low-order byte of the timer delta counter value. The timer will halt for delta clock cycles. The value written to T2TLD will not take effect until T2THD is written. The value written to this register while the timer is idle will be written to the low-order byte of the timer.

T2CMP (0x94) – Timer 2 Compare Value

Bit	Name	Reset	R/W	Description
7:0	CMP[7:0]	0x00	R/W	Timer Compare value. A timer compare occurs when the compare source selected by T2CNF.CMSEL equals the value held in CMP.

T2OF2 (0xA3) – Timer 2 Overflow Count 2

Bit	Name	Reset	R/W	Description
7:4	-	0000	R0	Not used, read as 0
3:0	OF2[3:0]	0x00	R/W	Overflow count. High bits T2OF[19:16]. T2OF is incremented by 1 each time the timer overflows i.e. timer counts to a value greater or equal to period. When reading this register, the value read is the value latched when T2OF0 was read. Writing to this register when the timer is in IDLE or RUN states will force the overflow count to be set to the value written to T2OF2:T2OF1:T2OF0. If the count would otherwise be incremented by 1 when this register is written then 1 is added to the value written.

T2OF1 (0xA2) – Timer 2 Overflow Count 1

Bit	Name	Reset	R/W	Description
7:0	OF1[7:0]	0x00	R/W	Overflow count. Middle bits T2OF[15:8]. T2OF is incremented by 1 each time the timer overflows i.e. timer counts to a value greater or equal to period. When reading this register, the value read is the value latched when T2OF0 was read. Writing to this register when the timer is in IDLE or RUN states will force the overflow count to be set to the value written to T2OF2:T2OF1:T2OF0. If the count would otherwise be incremented by 1 when this register is written then 1 is added to the value written. The value written will not take effect until T2OF2 is written.

T2OF0 (0xA1) – Timer 2 Overflow Count 0

Bit	Name	Reset	R/W	Description
7:0	OF0[7:0]	0x00	R/W	Overflow count. Low bits T2OF[7:0]. T2OF is incremented by 1 each time the timer overflows i.e. timer counts to a value greater or equal to period. Writing to this register when the timer is in IDLE or RUN states will force the overflow count to be set to the value written to T2OF2:T2OF1:T2OF0. If the count would otherwise be incremented by 1 when this register is written then 1 is added to the value written. The value written will not take effect until T2OF2 is written.

T2CAPHPH (0xA5) – Timer 2 Period High Byte

Bit	Name	Reset	R/W	Description
7:0	CAPHPH[7:0]	0xFF	R/	Capture value high/timer period high. Writing this register sets the high order bits [15:8] of the timer period. Reading this register gives the high order bits [15:8] of the timer value at the last capture event.

T2CAPLPL (0xA4) – Timer 2 Period Low Byte

Bit	Name	Reset	R/W	Description
7:0	CAPLPL[7:0]	0xFF	R/W	Capture value low/timer period low. Writing this register sets the low order bits [7:0] of the timer period. Reading this register gives the low order bits [7:0] of the timer value at the last capture event.

T2PEROF2 (0x9E) – Timer 2 Overflow Capture/Compare 2

Bit	Name	Reset	R/W	Description
7	CMPIM	0	R/W	Compare interrupt mask. 0: No interrupt is generated on compare event 1: Interrupt is generated on compare event.
6	PERIM	0	R/W	Overflow interrupt mask 0: No interrupt is generated on timer overflow 1: Interrupt is generated on timer overflow
5	OFCMPIM	0	R/W	Overflow count compare interrupt mask 0: No interrupt is generated on overflow count compare 1: Interrupt is generated on overflow count compare
4	-	0	R0	Not used, read as 0
3:0	PEROF2[3:0]	0000	R/W	Overflow count capture/Overflow count compare value. Writing these bits set the high bits [19:16] of the overflow count compare value. Reading these bits returns the high bits [19:16] of the overflow count value at the time of the last capture event.

T2PEROF1 (0x9D) – Timer 2 Overflow Capture/Compare 1

Bit	Name	Reset	R/W	Description
7:0	PEROF1[7:0]	0x00	R/W	Overflow count capture /Overflow count compare value. Writing these bits set the middle bits [15:8] of the overflow count compare value. Reading these bits returns the middle bits [15:8] of the overflow count value at the time of the last capture event.

T2PEROF0 (0x9C) – Timer 2 Overflow Capture/Compare 0

Bit	Name	Reset	R/W	Description
7:0	PEROF0[7:0]	0x00	R/W	Overflow count capture /Overflow count compare value. Writing these bits set the low bits [7:0] of the overflow count compare value. Reading these bits returns the low bits [7:0] of the overflow count value at the time of the last capture event.

13.5 Sleep Timer

The Sleep timer is used to set the period between when the system enters and exits low-power sleep modes.

The Sleep timer is also used to maintain timing in Timer 2 (MAC Timer) when entering a low-power sleep mode.

The main features of the Sleep timer are the following:

- 24-bit timer up-counter operating at 32.768 kHz clock
- 24-bit compare
- Low-power mode operation in PM2
- Interrupt and DMA trigger

13.5.1 Timer Operation

This section describes the operation of the timer.

13.5.1.1 General

The Sleep timer is a 24-bit timer running on the 32.768 kHz clock. The timer starts running immediately after a reset and continues to run uninterrupted. The current value of the timer can be read from the SFR registers ST2:ST1:ST0.

13.5.1.2 Timer Compare

A timer compare occurs when the timer value is equal to the 24-bit compare value. The compare value is set by writing to the registers ST2:ST1:ST0. When a timer compare occurs the interrupt flag STIF (interrupt 5) is asserted.

The interrupt enable bit for the ST interrupt is IEN0.STIE and the interrupt flag is IRCON.STIF.

When operating in power modes PM1 or PM2, the Sleep timer will be running and the Sleep timer compare event is used to wake up the device and return to active operation in PM0.

The default value of the compare value after reset is 0xFFFFFF.

The Sleep timer compare can also be used as a DMA trigger (DMA trigger 9 in Table 37).

ST2 (0x97) – Sleep Timer 2

Bit	Name	Reset	R/W	Description
7:0	ST2[7:0]	0x00	R/W	Sleep timer count/compare value. When read, this register returns the high bits [23:16] of the sleep timer count. When writing this register sets the high bits [23:16] of the compare value. The value read is latched at the time of reading register ST0. The value written is latched when ST0 is written.

ST1 (0x96) – Sleep Timer 1

Bit	Name	Reset	R/W	Description
7:0	ST1[7:0]	0x00	R/W	Sleep timer count/compare value. When read, this register returns the middle bits [15:8] of the sleep timer count. When writing this register sets the middle bits [15:8] of the compare value. The value read is latched at the time of reading register ST0. The value written is latched when ST0 is written.

ST0 (0x95) – Sleep Timer 0

Bit	Name	Reset	R/W	Description
7:0	ST0[7:0]	0x00	R/W	Sleep timer count/compare value. When read, this register returns the low bits [7:0] of the sleep timer count. When writing this register sets the low bits [7:0] of the compare value.

13.6 8-bit Timer 3 and Timer 4

Timer 3 and 4 are 8-bit timers which support typical input capture and output compare operations using two capture/compare channels each. The timer allows general purpose timer and waveform generation functions.

Features of Timer 3/4 are as follows:

- Dual channel operation
- Rising, falling or any edge input compare
- Set, clear or toggle output compare
- Clock prescaler for divide by 1, 2, 4, 8, 16, 32, 64, 128
- Interrupt request generated on each capture/compare and terminal count event
- DMA trigger function

13.6.1 8-bit Timer Counter

All timer functions are based on the main 8-bit counter found in Timer 3/4. The counter increments or decrements at each active clock edge. The period of the active clock edges is defined by the register bits CLKCON.TICKSPD which is further divided by the prescaler value set by TxCTL.DIV (where x refers to the timer number, 3 or 4). The counter operates as either a free-running counter, a down counter, a modulo counter or as an up/down counter.

It is possible to read the 8-bit counter value through the SFR TxCNT where x refers to the timer number, 3 or 4.

The possibility to clear and halt the counter is given with TxCTL control register settings. The counter is started when a 1 is written to TxCTL.START. If a 0 is written to TxCTL.START the counter halts at its present value.

13.6.2 Timer 3/4 Mode Control

In general the control register TxCTL is used to control the timer operation.

13.6.2.1 Free-running Mode

In the free-running mode of operation the counter starts from 0x00 and increments at each active clock edge. When the counter reaches 0xFF the counter is loaded with 0x00 and continues incrementing its value. When the terminal count value 0xFF is reached (i.e. an overflow occurs), the interrupt flag TIMIF.TxOVFIF is set. If the corresponding interrupt mask bit TxCTL.OVFIM is set, an interrupt request is generated. The free-running mode can be used to generate independent time intervals and output signal frequencies.

13.6.2.2 Down mode

In the down mode, after the timer has been started, the counter is loaded with the contents in TxCC. The counter then counts down to 0x00. The flag TIMIF.TxOVFIF is set when 0x00 is reached. If the corresponding interrupt mask bit TxCTL.OVFIM is set, an interrupt request is generated. The timer down mode can generally be used in applications where an event timeout interval is required.

13.6.2.3 Modulo Mode

When the timer operates in modulo mode the 8-bit counter starts at 0x00 and increments at each active clock edge. When the counter reaches the terminal count value held in register TxCC the counter is reset to 0x00 and continues to increment. The flag TIMIF.TxOVFIF is set when on this event. If the corresponding interrupt mask bit TxCTL.OVFIM is set, an interrupt request is generated. The modulo mode can be used for applications where a period other than 0xFF is required.

13.6.2.4 Up/down Mode

In the up/down timer mode, the counter repeatedly starts from 0x00 and counts up until the value held in TxCC is reached and then the counter counts down until 0x00 is reached. This timer mode is used when symmetrical output pulses are required with a period other than 0xFF, and therefore allows implementation of centre-aligned PWM output applications.

Clearing the counter by writing to TxCTL.CLR will also reset the count direction to the count up from 0x00 mode.

13.6.3 Channel Mode Control

The channel modes for each channel; 0 and 1, are set by the control and status registers

TxCCTLn where n is the channel number, 0 or 1. The settings include input capture and output compare modes.

13.6.4 Input Capture Mode

When the channel is configured as an input capture channel, the I/O pin associated with that channel is configured as an input. After the timer has been started, either a rising edge, falling edge or any edge on the input pin triggers a capture of the 8-bit counter contents into the associated capture register. Thus the timer is able to capture the time when an external event takes place.

The channel input pins are synchronized to the internal system clock. Thus pulses on the input pins must have a minimum duration greater than the system clock period.

Note: before an I/O pin can be used by the timer, the required I/O pin must be configured as a Timer 3/4 peripheral pin as described in sections 13.1.4.4 and 13.1.4.5 .

The content of the 8-bit capture registers is read out from registers TxCCn.

When a capture takes place the interrupt flag corresponding to the actual channel is set. This is TIMIF.TxCHnIF. An interrupt request is generated if the corresponding interrupt mask bit TxCCTLn.IM is set.

13.6.5 Output Compare Mode

In output compare mode the I/O pin associated with a channel shall be set to an output. After the timer has been started, the contents of the counter is compared with the contents of the channel compare register TxCC0n. If the compare register equals the counter contents, the output pin is set, reset or toggled according to the compare output mode setting of TxCCTL.CMP1:0. Note that all edges on output pins are glitch-free when operating in a given compare output mode.

For simple PWM use, output compare modes 4 and 5 are preferred.

Writing to the compare register TxCC0 does not take effect on the output compare value until the counter value is 0x00. Writing to the compare register TxCC1 takes effect immediately.

When a compare occurs the interrupt flag corresponding to the actual channel is set. This is TIMIF.TxCHnIF. An interrupt request is generated if the corresponding interrupt mask bit TxCCTLn.IM is set.

13.6.6 Timer 3 and 4 interrupts

There is one interrupt vector assigned to each of the timers. These are T3 (interrupt 11) and T4 (interrupt 12). An interrupt request is generated when one of the following timer events occur:

- Counter reaches terminal count value.
- Input capture event.
- Output compare event

The SFR register TIMIF contains all interrupt flags for Timer 3 and Timer 4. The register bits TIMIF.TxOVFIF and TIMIF.TxCHnIF, contains the interrupt flags for the two terminal count value events and the four channel compare/capture events, respectively. An interrupt request is only generated when the corresponding interrupt mask bit is set. If there are other pending interrupts, the corresponding interrupt flag must be cleared by the CPU before a new interrupt request can be generated. Also, enabling an interrupt mask bit will generate a new interrupt request if the corresponding interrupt flag is set.

13.6.7 Timer 3 and Timer 4 DMA triggers

There are two DMA triggers associated with Timer 3 and two DMA triggers associated with Timer 4. These are the following:

- T3_CH0 : Timer 3 channel 0 compare
- T3_CH1 : Timer 3 channel 1 compare
- T4_CH0 : Timer 4 channel 0 compare
- T4_CH0 : Timer 4 channel 1 compare

Refer to section 13.2 on page 86 for a description on use of DMA channels.

13.6.8 Timer 3 and 4 registers

T3CNT (0xCA) – Timer 3 Counter

Bit	Name	Reset	R/W	Description
7:0	CNT[7:0]	0x00	R	Timer count byte. Contains the current value of the 8-bit counter.

T3CTL (0xCB) – Timer 3 Control

Bit	Name	Reset	R/W	Description
7:5	DIV[2:0]	00	R/W	Prescaler divider value. Generates the active clock edge used to clock the timer from CLKCON.TICKSPD as follows: 000 Tick frequency /1 001 Tick frequency /2 010 Tick frequency /4 011 Tick frequency /8 100 Tick frequency /16 101 Tick frequency /32 110 Tick frequency /64 111 Tick frequency /128
4	START	0	R/W	Start timer. Normal operation when set, suspended when cleared
3	OVFIM	1	R/W0	Overflow interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
2	CLR	0	R0/W1	Clear counter. Writing high resets counter to 0x00
1:0	MODE[1:0]	00	R/W	Timer 3 mode. Select the mode as follows: 00 Free running, repeatedly count from 0x00 to 0xFF 01 Down, count from T3CC0 to 0x00 10 Modulo, repeatedly count from 0x00 to T3CC0 11 Up/down, repeatedly count from 0x00 to T3CC0 and down to 0x00

T3CCTL0 (0xCC) – Timer 3 Channel 0 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 0 interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
5:3	CMP[7:0]	000	R/W	Channel 0 compare output mode select. Specified action on output when timer value equals compare value in T3CC0 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on 0xFF 110 Clear output on compare, set on 0x00 111 Not used
2	MODE	0	R/W	Mode. Select Timer 3 channel 0 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP	00	R/W	Channel 0 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T3CC0 (0xCD) – Timer 3 Channel 0 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 0

T3CCTL1 (0xCE) – Timer 3 Channel 1 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 1 interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
5:3	CMP[2:0]	000	R/W	Channel 1 compare output mode select. Specified action on output when timer value equals compare value in T3CC1 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on T3CC0 110 Clear output on compare, set on T3CC0 111 Not used
2	MODE	0	R/W	Mode. Select Timer 3 channel 1 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 1 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T3CC1 (0xCF) – Timer 3 Channel 1 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 1

T4CNT (0xEA) – Timer 4 Counter

Bit	Name	Reset	R/W	Description
7:0	CNT[7:0]	0x00	R	Timer count byte. Contains the current value of the 8-bit counter.

T4CTL (0xEB) – Timer 4 Control

Bit	Name	Reset	R/W	Description
7:5	DIV[2:0]	00	R/W	Prescaler divider value. Generates the active clock edge used to clock the timer from CLKCON.TICKSPD as follows: 000 Tick frequency /1 001 Tick frequency /2 010 Tick frequency /4 011 Tick frequency /8 100 Tick frequency /16 101 Tick frequency /32 110 Tick frequency /64 111 Tick frequency /128
4	START	0	R/W	Start timer. Normal operation when set, suspended when cleared
3	OVFIM	1	R/W0	Overflow interrupt mask
2	CLR	0	R0/W1	Clear counter. Writing high resets counter to 0x00
1:0	MODE[1:0]	00	R/W	Timer 4 mode. Select the mode as follows: 00 Free running, repeatedly count from 0x00 to 0xFF 01 Down, count from T4CC0 to 0x00 10 Modulo, repeatedly count from 0x00 to T4CC0 11 Up/down, repeatedly count from 0x00 to T4CC0 and down to 0x00

T4CCTL0 (0xEC) – Timer 4 Channel 0 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 0 interrupt mask
5:3	CMP[2:0]	000	R/W	Channel 0 compare output mode select. Specified action on output when timer value equals compare value in T4CC0 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on 0x00 110 Clear output on compare, set on 0x00 111 Not used
2	MODE	0	R/W	Mode. Select Timer 4 channel 0 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 0 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T4CC0 (0xED) – Timer 4 Channel 0 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 0

T4CCTL1 (0xEE) – Timer 4 Channel 1 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 1 interrupt mask
5:3	CMP[2:0]	000	R/W	Channel 1 compare output mode select. Specified action on output when timer value equals compare value in T4CC1 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on T4CC0 110 Clear output on compare, set on T4CC0 111 Not used
2	MODE	0	R/W	Mode. Select Timer 4 channel 1 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 1 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T4CC1 (0xEF) – Timer 4 Channel 1 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 1

TIMIF (0xD8) – Timers 1/3/4 Interrupt Mask/Flag

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	OVFIM	1	R/W	Timer 1 overflow interrupt mask
5	T4CH1IF	0	R/W0	Timer 4 channel 1 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending
4	T4CH0IF	0	R/W0	Timer 4 channel 0 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending
3	T4OVFIF	0	R/W0	Timer 4 overflow interrupt flag 0 : no interrupt is pending 1 : interrupt is pending
2	T3CH1IF	0	R/W0	Timer 3 channel 1 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending
1	T3CH0IF	0	R/W0	Timer 3 channel 0 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending
0	T3OVFIF	0	R/W0	Timer 3 overflow interrupt flag 0 : no interrupt is pending 1 : interrupt is pending

13.7 ADC

13.7.1 ADC Introduction

The ADC supports up to 14-bit analog-to-digital conversion. The ADC includes an analog multiplexer with up to eight individually configurable channels, reference voltage generator and conversion results written to memory through DMA. Several modes of operation are available.

The main features of the ADC are as follows:

- Selectable decimation rates which also sets the resolution (8 to 13 bits).
- Eight individual input channels, single-ended or differential
- Reference voltage selectable as internal, external single ended, external differential or AVDD_SOC.
- Interrupt request generation
- DMA triggers at end of conversions
- Temperature sensor input
- Battery measurement capability

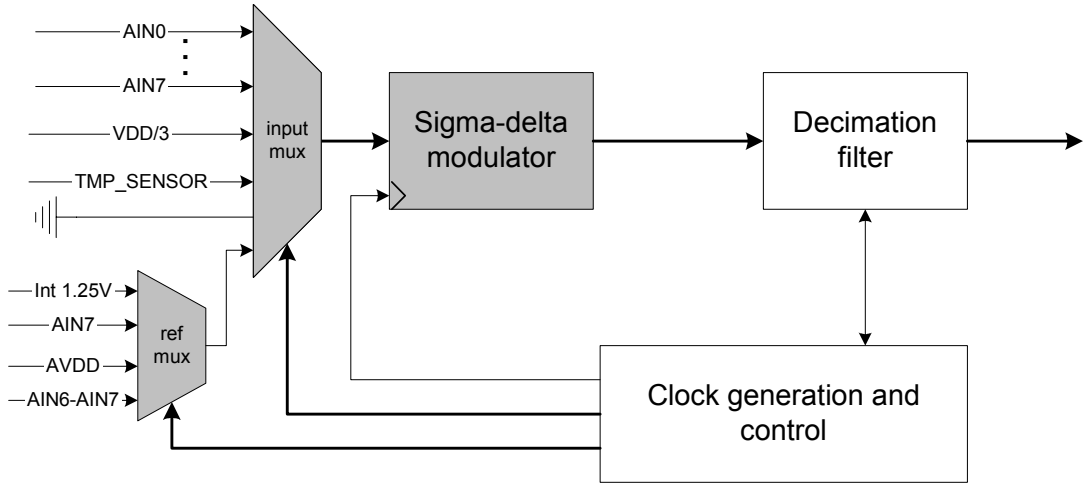


Figure 26: ADC block diagram.

13.7.2 ADC Operation

This section describes the general setup and operation of the ADC and describes the usage of the ADC control and status registers accessed by the CPU.

13.7.2.1 ADC Core

The ADC includes an ADC capable of converting an analog input into a digital representation with up to 13 bits resolution. The ADC uses a selectable positive reference voltage.

13.7.2.2 ADC Inputs

The signals on the `P0` port pins can be used as ADC inputs. In the following these port pin will be referred to as the AIN0-AIN7 pins. The input pins AIN0-AIN7 are connected to the ADC. The ADC can be set up to automatically perform a sequence of conversions and optionally perform an extra conversion from any channel when the sequence is completed.

It is possible to configure the inputs as single-ended or differential inputs. In the case where differential inputs are selected,

the differential inputs consist of the input pairs AIN0-1, AIN2-3, AIN4-5 and AIN6-7.

In addition to the input pins AIN0-AIN7, the output of an on-chip temperature sensor can be selected as an input to the ADC for temperature measurements.

It is also possible to select a voltage corresponding to AVDD_SOC/3 as an ADC input. This input allows the implementation of e.g. a battery monitor in applications where this feature is required.

13.7.2.3 ADC conversion sequences

The ADC will perform a sequence of conversions, and move the results to memory (through DMA) without any interaction from the CPU.

The ADCCON2.SCH register bits are used to define an ADC conversion sequence, from the ADC inputs. A conversion sequence will contain a conversion from each channel from 0 up to and including the channel number programmed in ADCCON2.SCH when ADCCON2.SCH is set to a value less than 8.

The single-ended inputs AIN0 to AIN7 are represented by channel numbers 0 to 7 in ADCCON2.SCH. Channel numbers 8 to 11 represent the differential inputs consisting of AIN0-AIN1, AIN2-AIN3, AIN4-AIN5 and AIN6-AIN7. Channel numbers 12 to 15 represent GND, internal voltage reference, temperature sensor and AVDD_SOC/3, respectively.

When ADCCON2.SCH is set to a value between 8 and 12, the sequence will start at channel 8. For even higher settings, only single conversions are performed. In addition to this sequence of conversions, the ADC can be programmed to perform a single conversion from any channel as soon as the sequence has completed. This is called an extra conversion and is controlled with the ADCCON3 register.

The conversion sequence can also be influenced with the ADCCFG register (see section 13.1.5 on page 72). The eight analog inputs to the ADC comes from IO pins, which are not necessarily programmed to be analog inputs. If a channel should normally be part of a sequence, but the corresponding analog input is disabled in the ADCCFG, then that channel will be skipped. For channels 8 to 12, both input pins must be enabled.

13.7.2.4 ADC Operating Modes

This section describes the operating modes and initialization of conversions.

The ADC has three control registers: ADCCON1, ADCCON2 and ADCCON3. These registers are used to configure the ADC and to report status.

The ADCCON1.EOC bit is a status bit that is set high when a conversion ends and cleared when ADCH is read.

The ADCCON1.ST bit is used to start a sequence of conversions. A sequence will start when this bit is set high, ADCCON1.STSEL="11" and no conversion is currently running. When the sequence is completed, this bit is automatically cleared.

The ADCCON1.STSEL bits select which event that will start a new sequence of conversions. The options which can be selected are rising edge on external pin P2_0, end of previous sequence, a Timer 1 channel 0 compare event or ADCCON1.ST='1'.

The ADCCON2 register controls how the sequence of conversions is performed.

ADCCON2.SREF is used to select the reference voltage. The reference voltage should only be changed when no conversion is running.

The ADCCON2.SDIV bits select the decimation rate (and thereby also the resolution and time required to complete a conversion and sample rate). The decimation rate should only be changed when no conversion is running.

The last channel of a sequence is selected with the ADCCON2.SCH bits.

The ADCCON3 register controls the channel number, reference voltage and decimation rate for the extra conversion. The extra conversion will take place immediately after the ADCCON3 register is updated. The coding of the register bits is exactly as for ADCCON2.

13.7.2.5 ADC Conversion Results

The digital conversion result is represented in two's complement form. For 13-bit resolution the digital conversion result is 8191 when the analog input is equal to the VREF, and the conversion result is -8192 when the analog input is equal to –VREF, where VREF is the selected positive voltage reference.

The digital conversion result is available when ADCCON1.EOC is set to 1, in ADCH and ADCL.

When the ADCCON2.SCH bits are read, they will indicate which channel the conversion result in ADCL and ADCH apply to.

13.7.2.6 ADC Reference Voltage

The positive reference voltage for analog-to-digital conversions is selectable as either an internally generated 1.25V voltage, the AVDD_SOC pin, the external voltage applied to the AIN7 input pin or the differential voltage applied to the AIN6-AIN7 inputs.

It is possible to select the reference voltage as the input to the ADC in order to perform a conversion of the reference voltage e.g. for calibration purposes. Similarly, it is possible to select the ground terminal GND as an input.

13.7.2.7 ADC Conversion Timing

The ADC runs on the 32 MHz system clock, which is divided by 8 to give a 4 MHz clock. Both the delta sigma modulator and decimation filter use the 4 MHz clock for their calculations.

The time required to perform a conversion depends on the selected decimation rate. When the decimation rate is set to for instance 128, the decimation filter uses exactly 128 of the 4 MHz clock periods to calculate the result. When a conversion is started, the input multiplexer is allowed 16 4 MHz clock cycles to settle in case the channel has been changed since the previous conversion. The 16 clock cycles settling time applies to all decimation rates. Thus in general, the conversion time is given by:

Tconv = (decimation rate + 16) x 0.25 µs.

13.7.2.8 ADC Interrupts

The ADC will generate an interrupt when an extra conversion has completed. An interrupt is not generated when a conversion from the sequence is completed.

13.7.2.9 ADC DMA Triggers

The ADC will generate a DMA trigger every time a conversion from the sequence has completed. When an extra conversion completes, no DMA trigger is generated.

There is one DMA trigger for each of the eight channels defined by the first eight possible settings for ADCCON2.SCH . The DMA trigger is active when a new sample is ready from the conversion for the channel. The DMA triggers are named ADC_CHx in Table 37 on page 92.

In addition there is one DMA trigger, ADC_CHALL, which is active when new data is ready from any of the channels in the ADC conversion sequence.

13.7.2.10 ADC Registers

This section describes the ADC registers.

ADCL (0xBA) – ADC Data Low

Bit	Name	Reset	R/W	Description
7:2	ADC[5:0]	0x00	R	Least significant part of ADC conversion result.
1:0	-	00	R0	Not used. Always read as 0

ADCH (0xBB) – ADC Data High

Bit	Name	Reset	R/W	Description
7:0	ADC[13:6]	0x00	R	Most significant part of ADC conversion result.

ADCCON1 (0xB4) – ADC Control 1

Bit	Name	Reset	R/W	Description
7	EOC	0	R H0	End of conversion Cleared when both ADCH and ADCL has been read. If a new conversion is completed before the previous data has been read, the EOC bit will remain high. 0 conversion not complete 1 conversion completed
6	ST	0	R/W1	Start conversion. Read as 1 until conversion has completed 0 no conversion in progress 1 start a conversion sequence if ADCCON1.STSEL = “11” and no sequence is running.
5:4	STSEL[1:0]	11	R/W	Start select. Selects which event that will start a new conversion sequence. 00 External trigger on P2_0 pin. 01 Full speed. Do not wait for triggers. 10 Timer 1 channel 0 compare event 11 ADCCON1.ST = 1
3:2	RCTRL[1:0]	00	R/W	Controls the 16 bit random number generator. When written 01, the setting will automatically return to 00 when operation has completed. 00 Normal operation. (13x unrolling) 01 Clock the LFSR once (no unrolling). 10 Reserved 11 Stopped. Random number generator is turned off.
1:0	-	11	R/W	Reserved. Always set to 11.

ADCCON2 (0xB5) – ADC Control 2

Bit	Name	Reset	R/W	Description
7:6	SREF[1:0]	00	R/W	Selects reference voltage used for the sequence of conversions 00 Internal 1.25V reference 01 External reference on AIN7 pin 10 AVDD_SOC pin 11 External reference on AIN6-AIN7 differential input
5:4	SDIV[1:0]	01	R/W	Sets the decimation rate for channels included in the sequence of conversions. The decimation rate also determines the resolution and time required to complete a conversion. 00 64 dec rate (8 bits resolution) 01 128 dec rate (10 bits resolution) 10 256 dec rate (12 bits resolution) 11 512 dec rate (13 bits resolution)
3:0	SCH[3:0]	00	R/W	Sequence Channel Select. Selects the end of the sequence. A sequence can either be from AIN0 to AIN7 (SCH<=7) or from the differential input AIN0-AIN1 to AIN6-AIN7 (8<=SCH<=11). For other settings, only single conversions are performed. When read, these bits will indicate the channel number of current conversion result. 0000 AIN0 0001 AIN1 0010 AIN2 0011 AIN3 0100 AIN4 0101 AIN5 0110 AIN6 0111 AIN7 1000 AIN0-AIN1 1001 AIN2-AIN3 1010 AIN4-AIN5 1011 AIN6-AIN7 1100 GND 1101 Positive voltage reference 1110 Temperature sensor 1111 VDD/3

ADCCON3 (0xB6) – ADC Control 3

Bit	Name	Reset	R/W	Description
7:6	EREF[1:0]	00	R/W	Selects reference voltage used for the extra conversion 00 Internal 1.25V reference 01 External reference on AIN7 pin 10 AVDD_SOC pin 11 External reference on AIN6-AIN7 differential input
5:4	EDIV[1:0]	00	R/W	Sets the decimation rate used for the extra conversion. The decimation rate also determines the resolution and time required to complete the conversion. 00 64 dec rate (8 bits resolution) 01 128 dec rate (10 bits resolution) 10 256 dec rate (12 bits resolution) 11 512 dec rate (13 bits resolution)
3:0	ECH[3:0]	0000	R/W	Extra channel select. Selects the channel number of the extra conversion that is carried out after a conversion sequence has ended. This bit field must be written for an extra conversion to be performed. If the ADC is not running, writing to these bits will trigger an immediate single conversion from the selected extra channel. The bits are automatically cleared when the extra conversion has finished. 0000 AIN0 0001 AIN1 0010 AIN2 0011 AIN3 0100 AIN4 0101 AIN5 0110 AIN6 0111 AIN7 1000 AIN0-AIN1 1001 AIN2-AIN3 1010 AIN4-AIN5 1011 AIN6-AIN7 1100 GND 1101 Positive voltage reference 1110 Temperature sensor 1111 VDD/3

13.8 Random Number Generator

13.8.1 Introduction

The random number generator has the following features.

- Generate pseudo-random bytes which can be read by the CPU or used directly by the Command Strobe Processor.
- Calculate CRC16 of bytes that are written to RNDH.
- Seeded by value written to RNDL.

The random number generator is a 16-bit Linear Feedback Shift Register (LFSR) with polynomial $X^{16} + X^{15} + X^2 + 1$ (i.e. CRC16). It uses different levels of unrolling depending on the operation it performs. The basic version (no unrolling) is shown below.

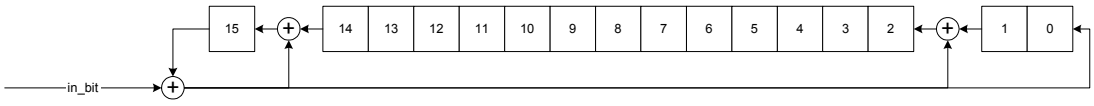


Figure 27: Basic structure of the Random Number Generator

The random number generator is turned off when ADCCON1.RCTRL=”11”.

13.8.2 Random Number Generator Operation

The operation of the random number generator is controlled through a combination of the ADCCON1.RCTRL bits and input signals from other modules. The current value of the 16-bit shift register in the LFSR can be read from the RNDH and RNDL registers.

13.8.2.1 Semi random sequence generation

The default operation (ADCCON1.RCTRL=”00”) is to clock the LFSR once (13x unrolling) each time the Command Strobe Processor reads the random value. This leads to the availability of a fresh pseudo-random byte from the LSB end of the LFSR.

Another way to update the LFSR is to set ADCCON1.RCTRL=”01”. This will clock the LFSR once (no unrolling) and the ADCCON1.RCTRL bits will automatically be cleared when the operation has completed.

13.8.2.2 Seeding

The LFSR can be seeded by writing to the RNDL register twice. Each time the RNDL register is written, the 8 LSB of the LFSR is copied to the 8 MSB and the 8 LSBs are replaced with the new data byte that was written to RNDL.

When a true random value is required, the LFSR can be seeded by writing RNDL with random values from the ADC in the RF receive path. To use this seeding method, the radio must first be powered on by enabling the voltage regulator as described in section 16.1. The radio should be placed in infinite TX state, to avoid possible sync detect in RX state. The random values from the ADC are read from the RF registers ADCTSTH and ADCTSTL (see page 216). The values read are used as the seed values to be written to the RNDL register as described above.

13.8.2.3 CRC16

The LFSR can also be used to calculate the CRC value of a sequence of bytes. Writing to the RNDH register will trigger a CRC calculation. The new byte is processed from the MSB end and an 8x

unrolling is used, so that a new byte can be written to RNDH every clock cycle.

Note that the LFSR must be properly seeded by writing to RNDL, before the CRC calculations start. Usually the seed value should be 0x0000 or 0xFFFF.

13.8.3 Random Number Generator Registers

This section describes the Random Number Generator registers.

RNDL (0xBC) – Random Number Generator Data Low Byte

Bit	Name	Reset	R/W	Description
[7:0]	RNDL[7:0]	0xFF	R/W	Random value/seed or CRC result, low byte When used for random number generation writing this register twice will seed the random number generator. Writing to this register copies the 8 LSBs of the LFSR to the 8 MSBs and replaces the 8 LSBs with the data value written. The value returned when reading from this register is the 8 LSBs of the LSFR. When used for random number generation, reading this register returns the 8 LSBs of the random number. When used for CRC calculations, reading this register returns the 8 LSBs of the CRC result.

RNDH (0xBD) – Random Number Generator Data High Byte

Bit	Name	Reset	R/W	Description
[7:0]	RNDH[7:0]	0xFF	R/W	Random value or CRC result/input data, high byte When written, a CRC16 calculation will be triggered, and the data value written is processed starting with the MSB bit. The value returned when reading from this register is the 8 MSBs of the LSFR. When used for random number generation, reading this register returns the 8 MSBs of the random number. When used for CRC calculations, reading this register returns the 8 MSBs of the CRC result.

13.9 AES Coprocessor

The CC2430 data encryption is performed using a dedicated coprocessor which supports the Advanced Encryption Standard, AES. The coprocessor allows encryption/decryption to be performed with minimal CPU usage.

The coprocessor has the following features:

- Supports all security suites in IEEE 802.15.4
- ECB, CBC, CFB, OFB, CTR and CBC-MAC modes.
- Hardware support for CCM mode
- 128-bits key and IV/Nonce
- DMA transfer trigger capability

13.9.1 AES Operation

To encrypt a message, the following procedure must be followed:

- Load key
- Load initialization vector (IV)
- Download and upload data for encryption/decryption.

The AES coprocessor works on blocks of 128 bits. A block of data is loaded into the coprocessor, encryption is performed and the result must be read out before the next block can be processed. Before each block load, a dedicated start command must be sent to the coprocessor.

13.9.2 Key and IV

Before a key or IV/nonce load starts, an appropriate load key or IV/nonce command must be issued to the coprocessor. When loading the IV it is important to also set the correct mode.

A key load or IV load operation aborts any processing that could be running.

The key, once loaded, stays valid until a key reload takes place.

The IV must be downloaded before the beginning of each message (not block).

Both key and IV values are cleared by a reset of the CC2430.

13.9.3 Padding of input data

The AES coprocessor works on blocks of 128 bits. If the last block contains less than 128 bits, it must be padded with zeros when written to the coprocessor.

13.9.4 Interface to CPU

The CPU communicates with the coprocessor using three SFR registers:

- ENCCS, Encryption control and status register
- ENCDI, Encryption input register
- ENCDO, Encryption output register

Read/write to the status register is done directly by the CPU, while access to the input/output registers must be performed using direct memory access (DMA).

Two DMA channels must be used, one for input data and one for output data. The DMA channels must be initialized before a start command is written to the ENCCS. Writing a start command generates a DMA trigger and the transfer is started. After each block is processed, an interrupt is generated. The interrupt is used to issue a new start command to the ENCCS.

13.9.5 Modes of operation

ECB and CBC modes are performed as described in section 13.9.1

When using CFB, OFB and CTR mode, the 128 bits blocks are divided into four 32 bit blocks. 32 bits are loaded into the AES coprocessor and the resulting 32 bits are read out. This continues until all 128 bits have been encrypted. The only time one has to consider this is if data is loaded/read directly using the CPU. When using DMA, this is handled automatically by the DMA triggers generated by the AES coprocessor.

Both encryption and decryption are performed similarly.

The CBC-MAC mode is a variant of the CBC mode. When performing CBC-MAC, data is downloaded to the coprocessor one 128 bits block at a time, except for the last block. Before the last block is loaded,

the mode must be changed to CBC. The last block is then downloaded and the block uploaded will be the MAC value.

CCM is a combination of CBC-MAC and CTR. Parts of the CCM must therefore be done in software. The following section gives a short explanation of the necessary steps to be done.

13.9.5.1 CBC-MAC

When performing CBC-MAC encryption, data is downloaded to the coprocessor in CBC-MAC mode one block at a time, except for the last block. Before the last block is loaded, the mode is changed to CBC. The last block is downloaded and the block uploaded is the message MAC.

CBC-MAC decryption is similar to encryption. The message MAC uploaded must be compared with the MAC to be verified.

13.9.5.2 CCM mode

To encrypt a message under CCM mode, the following sequence can be conducted (key is already loaded):

Message Authentication Phase

This phase takes place during steps 1-6 shown in the following.

(1) The software loads the IV with zeros.

(2) The software creates the block B0. The layout of block B0 is shown in Figure 28.

	Name B0					Designation First block for authentication in CCM mode										
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Name	Flag	NONCE									L_M					

Figure 28: Message Authentication Phase Block 0

There is no restriction on the NONCE value. L_M is the message length in bytes.

For 802.15.4 the NONCE is 13 bytes and L_M is 2 bytes.

The content of the Authentication Flag byte is described in Figure 29.

L is set to 6 in this example. So, L-1 is set to 5. M and A_Data can be set to any value.

	Name FLAG/B0		Designation Authentication Flag Field for CCM mode					
Bit	7	6	5	4	3	2	1	0
Name	Reserved	A_Data	(M-2)/2			L-1		
Value	0	x	x	x	x	1	0	1

Figure 29: Authentication Flag Byte

(3) If some Additional Authentication Data (denoted a below) is needed (that is A_Data =1), the software creates the A_Data length field, called L(a) by :

- **(3a)** If l(a)=0, (that is A_Data =0), then L(a) is the empty string. We note l(a) the length of a in octets.
- **(3b)** If $0 < l(a) < 2^{16} - 2^{8}$, then L(a) is the 2-octets encoding of l(a).

The Additional Authentication Data is appended to the A_Data length field L(a). The Additional Authentication Blocks is padded with zeros until the last Additional Authentication Block is full. There is no restriction on the length of a.

AUTH-DATA = L(a) + Authentication Data + (zero padding)

(**4)** The last block of the message is padded with zeros until full (that is if its length is not a multiple of 128).

(5) The software concatenates the block B0, the Additional Authentication Blocks if any, and the message;

Input message = B0 + AUTH-DATA + Message + (zero padding of message)

(6) Once the input message authentication by CBC-MAC is finished, the software leaves the uploaded buffer contents unchanged (M=16), or keeps only the buffer's higher M bytes unchanged, while setting the lower bits to 0 (M != 16).

The result is called T.

Message Encryption

(7) The software creates the key stream block A0. Note that L=6, with the current example of the CTR generation. The content is shown in Figure 30.

Note that any value but zero works for the CTR value.

The content of the Encryption Flag byte is described in Figure 31.

	Name A0					Designation First CTR value for CCM mode										
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Name	Flag	NONCE											CTR			

Figure 30: Message Encryption Phase Block 0

	Name FLAG/A0		Designation Encryption Flag Field for CCM mode					
Bit	7	6	5	4	3	2	1	0
Name	Reserved		-			L-1		
Value	0	0	0	0	0	1	0	1

Figure 31: Encryption Flag Byte

Message Encryption (cont.)

(**8)** The software loads A0 by selecting a Load IV/Nonce command. To do so, it sets Mode to CFB or OFB at the same time it selects the Load IV/Nonce command.

(9) The software calls a CFB or an OFB encryption on the authenticated data T. The uploaded buffer contents stay unchanged (M=16), or only its first M bytes stay unchanged, the others being set to 0 (M-16). The result is U, which will be used later.

(**10)** The software calls a CTR mode encryption right now on the still padded message blocks. It does not have to reload the IV/CTR.

(11) The encrypted authentication data U is appended to the encrypted message. This gives the final result, c.

Result c = encrypted message(m) + U

***Message Decrypt*ion**

CCM Mode decryption

In the coprocessor, the automatic generation of CTR works on 32 bits, therefore the maximum length of a message is 128 x 2^{32} bits, that is 2^{36} bytes, which can be written in a six-bit word. So, the value L is set to 6. To decrypt a CCM mode processed message, the following sequence can be conducted (key is already loaded):

Message Parsing Phase

(1) The software parses the message by separating the M rightmost octets, namely U, and the other octets, namely string C.

(2) C is padded with zeros until it can fill an integer number of 128-bit blocks;

(3) U is padded with zeros until it can fill a 128-bit block.

(4) The software creates the key stream block A0. It is done the same way as for CCM encryption.

(5) The software loads A0 by selecting a Load IV/Nonce command. To do so, it sets Mode to CFB or OFB at the same time as it selects the IV load.

(6) The software calls a CFB or an OFB encryption on the encrypted authenticated data U. The uploaded buffer contents stay unchanged (M=16), or only its first M bytes stay unchanged, the others being set to 0 (M!=16). The result is T.

(**7)** The software calls a CTR mode decryption right now on the encrypted message blocks C. It does not have to reload the IV/CTR.

Reference Authentication tag generation

This phase is identical to the Authentication Phase of CCM encryption. The only difference is that the result is named MACTag (instead of T).

Message Authentication checking Phase

The software compares T with MACTag.

13.9.6 Sharing the AES coprocessor between layers

The AES coprocessor is a common resource shared by all layers. The AES coprocessor can only be used by one instance one at a time. It is therefore necessary to implement some kind of software semaphore to allocate and de-allocate the resource.

13.9.7 AES Interrupts

The AES interrupt, ENC, is produced when encryption or decryption of a block is completed. The interrupt enable bit is `IEN0.ENCIE` and the interrupt flag is `S0CON.ENCIF`.

13.9.8 AES DMA Triggers

There are two DMA triggers associated with the AES coprocessor. These are ENC_DW which is active when input data needs to be downloaded to the `ENCDI` register, and ENC_UP which is active when output data needs to be uploaded from the `ENCDO` register.

The `ENCDI` and `ENCDO` registers should be set as destination and source locations for DMA channels used to transfer data to or from the AES coprocessor.

13.9.9 AES Registers

The AES coprocessor registers have the layout shown in this section.

ENCCS (0xB3) – Encryption Control and Status

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used, always read as 0
6:4	MODE[2:0]	000	R/W	Encryption/decryption mode 000 CBC 001 CFB 010 OFB 011 CTR 100 ECB 101 CBC MAC 110 Not used 111 Not used
3	RDY	1	R	Encryption/decryption ready status 0 Encryption/decryption in progress 1 Encryption/decryption is completed
2:1	CMD[1:0]	0	R/W	Command to be performed when a 1 is written to ST. 00 encrypt block 01 decrypt block 10 load key 11 load IV/nonce
0	ST	0	R/W1 H0	Start processing command set by CMD. Must be issued for each command or 128 bits block of data. Cleared by hardware

ENCDI (0xB1) – Encryption Input Data

Bit	Name	Reset	R/W	Description
7:0	DIN[7:0]	0x00	R/W	Encryption input data

ENCDO (0xB2) – Encryption Output Data

Bit	Name	Reset	R/W	Description
7:0	DOUT[7:0]	0x00	R/W	Encryption output data

13.10 Power Management

This section describes the Power Management Controller. The Power Management Controller controls the use of power modes and clock control to achieve low-power operation.

13.10.1 Power Management Introduction

The ***CC2430*** uses different operating modes, or power modes, to allow low-power operation. Ultra-low-power operation is obtained by turning off power supply to modules to avoid static (leakage) power consumption and also by using clock gating to reduce dynamic power consumption.

The various operating modes are enumerated and are be designated as power modes (PMx). The power modes are:

- PM0
  Clock oscillators on, voltage regulator on
- PM1
  32.768 kHz oscillators on, voltage regulator on
- PM2
  32.768 kHz oscillators on, voltage regulator off
- PM3
  All clock oscillators off, voltage regulator off

Note: the voltage regulator above refers to the digital regulator. The analog voltage regulator must be disabled separately through the RF register `RFPWR`.

13.10.1.1 PM0

PM0 is the full functional mode of operation where the CPU, peripherals and RF transceiver are active. The voltage regulator is turned on.

PM0 is used for normal operation.

13.10.1.2 PM1

In PM1, the high-speed oscillators are powered down. The voltage regulator and the 32.768 kHz oscillators are on. When PM1 is entered, a power down sequence is run. When the device is taken out of PM1 to PM0, the high-speed oscillators are started. The device will run on the high speed RC oscillator until the high speed XOSC has settled and been selected.

PM1 is used when the expected time until a wakeup event is relatively short since PM1 uses a fast power down/up sequence.

13.10.1.3 PM2

PM2 has the second lowest power consumption. In PM2 the power-on reset, external interrupts, 32.768 kHz oscillator and sleep timer peripherals are active. I/O pins retain the I/O mode and output value set before entering PM2. All other internal circuits are powered down. The voltage regulator is also turned off. When PM2 is entered, a power down sequence is run.

PM2 is used when it is relatively long until the expected time for wakeup event, since the power up/down sequence is relatively long. PM2 is typically entered when using the sleep timer as the wakeup event.

13.10.1.4 PM3

PM3 is used to achieve the operating mode with the lowest power consumption. In PM3 all internal circuits that are powered from the voltage regulator are turned off. The internal voltage regulator and all oscillators are also turned off.

Power-on reset and external I/O port interrupts are the only functions that are operating in this mode. I/O pins retain the I/O mode and output value set before entering PM3. Only a reset or external interrupt condition will wake the device up and place it into PM0. The contents of RAM and registers are preserved in this mode. PM3 uses the same power down/up sequence as PM2.

PM3 is used to achieve ultra low power consumption when waiting for an external event.

13.10.2 Power Management Control

The required power mode is selected by the `MODE` bits in the `SLEEP` control register. Setting the SFR register

PCON.IDLE bit after setting the MODE bits, enters the selected sleep mode.

An enabled interrupt from port pins or sleep timer or a power-on reset will wake the device from other power modes and bring it into PM0 by resetting the MODE bits.

13.10.3 System clock

The system clock is derived from the selected system clock source, which is the high-speed (32 MHz) crystal oscillator or the high-speed RC oscillator. The CLKCON.OSC bit selects the source of the system clock. Note that to use the RF transceiver the high speed crystal oscillator must be selected and stable.

Note that each time the CLKCON.OSC bit is altered, then the CLKCON.CLKSPD bit must also be altered in order for the system to know the frequency of the system clock.

When the SLEEP.XOSC_STB is 1, the 32 MHz crystal oscillator is stable and can be used as the source for the system clock.

The oscillator not selected as the system clock source, will be set in power-down mode by setting SLEEP.OSC_PD to 1. Thus the high-speed RC oscillator may be turned off when the 32 MHz crystal oscillator has been selected as system clock source and vice versa. When SLEEP.OSC_PD is 0, both oscillators are powered up and running.

13.10.4 High-speed oscillators

Two high speed oscillators are present in the device. The high-speed crystal oscillator startup time may be too long for some applications, therefore the device can run on the high-speed RC oscillator until crystal oscillator is stable. The high-speed RC oscillator consumes less power than the crystal oscillator, but since it is not as accurate as the crystal oscillator it can not be used for RF transceiver operation.

13.10.5 32.768 kHz oscillators

Two 32.768 kHz oscillators are present in the device. By default the low power RC oscillator is enabled. The RC oscillator consumes less power, but is less accurate than the 32.768 kHz crystal oscillator. Refer to Table 9 and Table 10 on page 16 for characteristics of these oscillators. The 32.768 kHz oscillators are used to drive the 32.768 kHz clock to the Sleep Timer and Watchdog Timer. The selection of which oscillator source for the 32.768 kHz clock is performed with the CLKCON.OSC32K register bit.

The CLKCON.OSC32K register bit must only be changed while using the high-speed RC oscillator as the system clock source. When the high speed crystal oscillator is running the 32.768 kHz RC oscillator is continuously calibrated, and switching 32.768 kHz oscillator is not supported in this case.

13.10.6 Timer Tick generation

The power management controller generates a tick or enable signal for the peripheral timers, thus acting as a prescaler for the timers. This is a global clock division for Timer 1, Timer 3 and Timer 4. The tick speed is programmed from 0.25 to 32 MHz in the CLKCON.TICKSPD register.

13.10.7 Data Retention

In power modes PM2 and PM3 parts of SRAM will retain its contents. The content of internal registers is also retained in PM2/3.

The XDATA memory locations 0xF000-0xFFFF (4096 bytes) retains data in PM2/3. Please note one exception as given below.

The XDATA memory locations 0xE000-0xEFFF (4096 bytes) and the area 0xFD58-0xFEFF (424 bytes) will lose all data when PM2/3 is entered. These locations will contain undefined data when PM0 is re-entered.

The registers which retain their contents are the CPU registers, peripheral registers and RF registers therefore switching to the low-power modes PM2/3 appears transparent to software. The RF TXFIFO/RXFIFO contents is not retained when entering PM2/3.

13.10.8 Power Management Registers

This section describes the Power Management registers.

PCON (0x87) – Power Mode Control

Bit	Name	Reset	R/W	Description
7:2	–	0x00	R/W	Not used.
1	–	0	R0	Not used, always read as 0.
0	IDLE	0	R0/W H0	Power mode control. Writing a 1 to this bit forces CC2430 to enter the power mode set by SLEEP.MODE. This bit is always read as 0

SLEEP (0xBE) – Sleep Mode Control

Bit	Name	Reset	R/W	Description
7	-	0	R0	Unused
6	XOSC_STB	0	R	XOSC stable status: 0 – XOSC is not powered up or not yet stable 1 – XOSC is powered up and stable
5	HFRC_STB	0	R	HF RCOSC stable status: 0 – HF RCOSC is not powered up or not yet stable 1 – HF RCOSC is powered up and stable
4:3	RST[1:0]	XX	R	Status bit indicating the cause of the last reset. If there are multiple resets, the register will only contain the last event. 00 – Power-on reset 01 – External reset 10 – Watchdog timer reset
2	OSC_PD	1	R/W H0	XOSC and HF RCOSC power down setting. The bit is cleared if the CLKCON.OSC bit is toggled. Also, if there is a calibration in progress and the CPU attempts to set the bit, the bit will be updated at the end of calibration: 0 – Both oscillators powered up 1 – Oscillator not selected by CLKCON.OSC bit powered down
1:0	MODE[1:0]	00	R/W	Sleep mode setting: 00 – Power mode 0 01 – Power mode 1 10 – Power mode 2 11 – Power mode 3

CLKCON (0xC6) – Clock Control

Bit	Name	Reset	R/W	Description
7	OSC32K	1	R/W	32 kHz clock oscillator select. The 16 MHz high speed RC oscillator must be selected as system clock source when this bit is to be changed. 0 – 32 kHz crystal oscillator 1 – 32 kHz RC oscillator
6	OSC	1	R/W	System clock oscillator select: 0 – 32 MHz crystal oscillator 1 – 16 MHz high speed RC oscillator This setting will only take effect when the selected oscillator is powered up and stable. If the selected oscillator is not powered up, then writing this bit will power it up.
5:3	TICKSPD[2:0]	001	R/W	Timer ticks output setting, can not be higher than system clock setting given by OSC bit setting 000 – System clock 001 – System clock / 2 010 – System clock / 4 011 – System clock / 8 100 – System clock / 16 101 – System clock / 32 110 – System clock / 64 111 – System clock / 128
2:1	–	00	R/W	Reserved. Must always be set to 00.
0	CLKSPD	1	R/W	Clock Speed. Sets system clock frequency together with the OSC bit setting 0 – 32 MHz 1 – 16 MHz This bit must always follow the value of the OSC bit, which selects which oscillator to use.

13.11 Power On Reset and Brown Out Detector

The *CC2430* includes a Power On Reset (POR) and Brown Out Detector (BOD) in order to protect the memory contents during supply voltage variations and provide correct initialization during power-on.

When power is initially applied to the *CC2430* the Power On Reset (POR) and Brown Out Detector (BOD) will hold the device in reset state until the supply voltage reaches above the Power On Reset and Brown Out voltages as defined in Table 4 on page 12.

Figure 32 shows the POR/BOD operation with the 1.8V (typical) regulated supply voltage together with the active low reset signals BOD_RESET and POR_RESET shown in the bottom of the figure.

The cause of the last reset can read from the register bits `SLEEP.RST`.

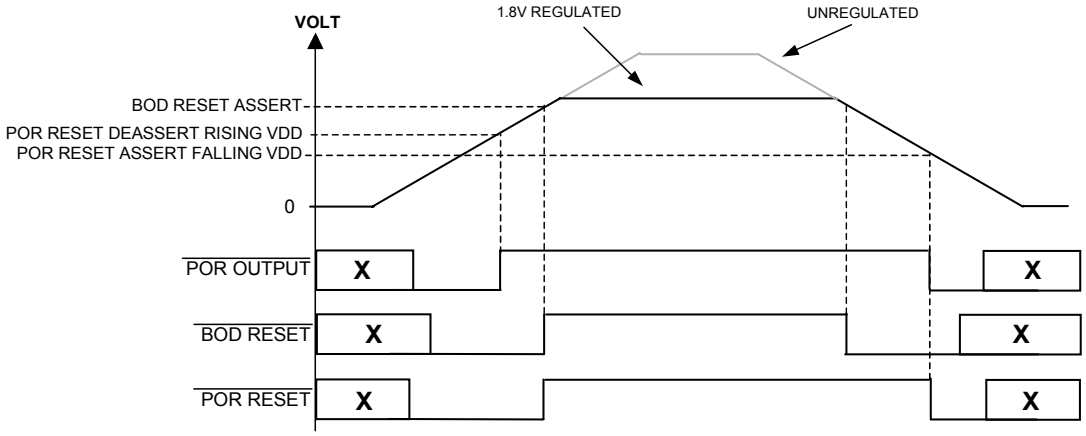


Figure 32 : Power On Reset and Brown Out Detector Operation

13.12 Watchdog Timer

The watchdog timer (WDT) is intended as a recovery method in situations where the CPU may be subjected to a software upset. The WDT shall reset the system when software fails to clear the WDT within a selected time interval. The watchdog can be used in applications that are subject to electrical noise, power glitches, electrostatic discharge etc., or where high reliability is required. If the watchdog function is not needed in an application, it is possible to configure the watchdog timer to be used as an interval timer that can be used to generate interrupts at selected time intervals.

The features of the watchdog timer are as follows:

- Four selectable timer intervals
- Watchdog mode
- Timer mode
- Interrupt request generation in timer mode
- Clock independent from system clock

The WDT is configured as either a watchdog timer or as a timer for general-purpose use. The operation of the WDT module is controlled by the WDCTL register. The watchdog timer consists of an 15-bit counter clocked by the 32.768 kHz oscillator clock. Note that the contents of the 15-bit counter is not user-accessible.

13.12.1 Watchdog mode

The watchdog timer is disabled after a system reset. To set the WDT in watchdog mode the WDCTL.MODE bit is set to 0. The watchdog timer counter starts incrementing when the enable bit WDCTL.EN is set to 1. When the timer is enabled in watchdog mode it is not possible to disable the timer. Therefore, writing a 0 to WDCTL.EN has no effect if a 1 was already written to this bit when WDCTL.MODE was 0.

The WDT operates with a watchdog timer clock frequency of 32.768 kHz. This clock frequency gives time-out periods equal to 1.9 ms, 15.625 ms, 0.25 s and 1 s corresponding to the count value settings 64, 512, 8192 and 32768 respectively.

If the counter reaches the selected timer interval value, the watchdog timer generates a reset signal for the system. If a watchdog clear sequence is performed before the counter reaches the selected timer interval value, the counter is reset to 0x0000 and continues incrementing its value. The watchdog clear sequence consists of writing 0xA to WDCTL.CLR[3:0] followed by writing 0x5 to the same register bits within one half of a watchdog clock period. If this complete sequence is not performed, the watchdog timer generates a reset signal for the system. Note that as long as a correct watchdog clear sequence begins within the selected timer interval, the counter is reset when the complete sequence has been received.

When the watchdog timer has been enabled in watchdog mode, it is not possible to change the mode by writing to the WDCTL.MODE bit. The timer interval value can be changed by writing to the WDCTL.INT[1:0] bits.

Note that it is recommended that user software clears the watchdog timer at the same time as the timer interval value is changed, in order to avoid an unwanted watchdog reset.

In watchdog mode, the WDT does not produce an interrupt request.

13.12.2 Timer mode

To set the WDT in normal timer mode, the WDCTL.MODE bit is set to 1. When register bit WDCTL.EN is set to 1, the timer is started and the counter starts incrementing. When the counter reaches the selected interval value, the timer will produce an interrupt request.

In timer mode, it is possible to clear the timer contents by writing a 1 to WDCTL.CLR[0]. When the timer is cleared the contents of the counter is set to 0x0000. Writing a 0 to the enable bit WDCTL.EN stops the timer and writing 1 restarts the timer from 0x0000.

The timer interval is set by the WDCTL.INT[1:0] bits. In timer mode, a reset will not be produced when the timer interval has been reached.

13.12.3 Watchdog Timer Example

Figure 33 shows an example of periodical clearing of an active watchdog timer.

```
; clear watchdog timer
MOV    WDCTL,#ABh
MOV    WDCTL,#5Bh
```

Figure 33: WDT Example

13.12.4 Watchdog Timer Register

This section describes the register, WDCTL, for the Watchdog Timer.

WDCTL (0xC9) – Watchdog Timer Control

Bit	Name	Reset	R/W	Description
7:4	CLR[3:0]	0000	R/W	Clear timer. When 0xA followed by 0x5 is written to these bits, the timer is loaded with 0x0. Note the timer will only be cleared when 0x5 is written within 0.5 watchdog clock period after 0xA was written. Writing to these bits when EN is 0 have no effect. These bits are always be read as 0000.
3	EN	0	R/W	Enable timer. When a 1 is written to this bit the timer is enabled and starts incrementing. Writing a 0 to this bit in timer mode stops the timer. Writing a 0 to this bit in watchdog mode has no effect. 0 Timer disabled (stop timer) 1 Timer enabled
2	MODE	0	R/W	Mode select. This bit selects the watchdog timer mode. 0 Watchdog mode 1 Timer mode
1:0	INT[1:0]	00	R/W	Timer interval select. These bits select the timer interval defined as a given number of 32.768 kHz oscillator periods. 00 clock period x 32768 (typical 1 s) 01 clock period x 8192 (typical 0.25 s) 10 clock period x 512 (typical 15.625 ms) 11 clock period x 64 (typical 1.9 ms)

13.13 USART

USART0 and USART1 are serial communications interfaces that can be operated separately in either asynchronous UART mode or in synchronous SPI mode. The two USARTs have identical function, and are assigned to separate I/O pins. Refer to section 13.1 for I/O configuration.

13.13.1 UART mode

For asynchronous serial interfaces, the UART mode is provided. In the UART mode the interface uses a two-wire or four-wire interface consisting of the pins RXD, TXD and optionally RTS and CTS. The UART mode of operation includes the following features:

- 8 or 9 data bits
- Odd, even or no parity
- Configurable start and stop bit level
- Configurable LSB or MSB first transfer
- Independent receive and transmit interrupts
- Independent receive and transmit DMA triggers
- Parity and framing error status

The UART mode provides full duplex asynchronous transfers, and the synchronization of bits in the receiver does not interfere with the transmit function. A UART byte transfer consists of a start bit, eight data bits, an optional ninth data or parity bit, and one or two stop bits. Note that the data transferred is referred to as a byte, although the data can actually consist of eight or nine bits.

The UART operation is controlled by the USART Control and Status registers, `UxCSR` and the UART Control register `UxUCR` where x is the USART number, 0 or 1.

The UART mode is selected when `UxCSR.MODE` is set to 1.

13.13.1.1 UART Transmit

A UART transmission is initiated when the USART Receive/transmit Data Buffer, `UxDBUF` register is written. The byte is transmitted on TXDx output pin. The `UxDBUF` register is double-buffered.

The `UxCSR.ACTIVE` bit goes high when the byte transmission starts and low when it ends. When the transmission ends, the `UxCSR.TX_BYTE` bit is set to 1. An interrupt request is generated when the `UxDBUF` register is ready to accept new transmit data. This happens immediately after the transmission has been started, hence a new data byte value can be loaded into the data buffer while the byte is being transmitted.

13.13.1.2 UART Receive

Data reception on the UART is initiated when a 1 is written to the `UxCSR.RE` bit. The UART will then search for a valid start bit on the RXDx input pin and set the `UxCSR.ACTIVE` bit high. When a valid start bit has been detected the received byte is shifted into the receive register. The `UxCSR.RX_BYTE` bit is set and a receive interrupt is generated when the operation has completed. At the same time `UxCSR.ACTIVE` will go low.

The received data byte is available through the `UxDBUF` register. When `UxDBUF` is read, `UxCSR.RX_BYTE` is cleared by hardware.

13.13.1.3 UART Hardware Flow Control

Hardware flow control is enabled when the `UxUCR.FLOW` bit is set to 1. The *RTS* output will then be driven low when the receive register is empty and reception is enabled. Transmission of a byte will not occur before the CTS input goes low.

13.13.1.4 UART Character Format

If the `BIT9` and `PARITY` bits in register `UxUCR` are set high, parity generation and detection is enabled. The parity is computed and transmitted as the ninth bit, and during reception, the parity is computed and compared to the received ninth bit. If there is a parity error, the `UxCSR.ERR` bit is set high. This bit is cleared when `UxCSR` is read.

The number of stop bits to be transmitted is set to one or two bits determined by the register bit UxUCR.SPB. The receiver will always check for one stop bit. If the first stop bit received during reception is not at the expected stop bit level, a framing error is signaled by setting register bit UxCSR.FE high. UxCSR.FE is cleared when UxCSR is read. The receiver will check both stop bits when UxUCR.SPB is set.

13.13.2 SPI Mode

This section describes the SPI mode of operation for synchronous communication. In SPI mode, the USART communicates with an external system through a 3-wire or 4-wire interface. The interface consists of the pins MOSI, MISO, SCK and SS_N. Refer to section 13.1 for description of how the USART pins are assigned to the I/O pins.

The SPI mode includes the following features:

- 3-wire (master) and 4-wire SPI interface
- Master and slave modes
- Configurable SCK polarity and phase
- Configurable LSB or MSB first transfer

The SPI mode is selected when UxCSR.MODE is set to 0.

In SPI mode, the USART can be configured to operate either as an SPI master or as an SPI slave by writing the UxCSR.SLAVE bit.

13.13.2.1 SPI Master Operation

An SPI byte transfer in master mode is initiated when the UxDBUF register is written. The USART generates the SCK serial clock using the baud rate generator (see section 13.13.3) and shifts the provided byte from the transmit register onto the MOSI output. At the same time the receive register shifts in the received byte from the MISO input pin.

The UxCSR.ACTIVE bit goes high when the transfer starts and low when the transfer ends. When the transfer ends, the UxCSR.RX_BYTE and UxCSR.TX_BYTE bits are set to 1. A receive interrupt is generated when new received data is ready in the UxDBUF USART Receive/Transmit Data register.

The polarity and clock phase of the serial clock SCK is selected by UxGCR.CPOL and UxGCR.CPHA. The order of the byte transfer is selected by the UxGCR.ORDER bit.

At the end of the transfer, the received data byte is available for reading from the UxDBUF.

A transmit interrupt is generated when the unit is ready to accept another data byte for transmission. Since UxDBUF is double-buffered, this happens just after the transmission has been initiated.

13.13.2.2 SPI Slave Operation

An SPI byte transfer in slave mode is controlled by the external system. The data on the MOSI input is shifted into the receive register controlled by the serial clock SCK which is an input in slave mode. At the same time the byte in the transmit register is shifted out onto the MISO output.

The UxCSR.ACTIVE bit goes high when the transfer starts and low when the transfer ends. Then the UxCSR.RX_BYTE and UxCSR.TX_BYTE bits are set and a receive interrupt is generated.

The expected polarity and clock phase of SCK is selected by UxGCR.CPOL and UxGCR.CPHA. The expected order of the byte transfer is selected by the UxGCR.ORDER bit.

At the end of the transfer, the received data byte is available for reading from UxDBUF

The transmit interrupt is generated at the start of the operation.

13.13.3 SSN Slave Select Pin

When the USART is operating in SPI mode, configured as an SPI slave, the Slave Select (SSN) pin is an input to the SPI. When SSN is held low, the SPI slave is active and receives data on the MOSI input and outputs data on the MISO output. When SSN is held high, the SPI slave is inactive and will not receive data.

In SPI master mode, the SSN pin is not used. When the USART operates as an SPI master and a slave select signal is needed by an external SPI slave device, then a general purpose I/O pin should be used to implement the slave select signal function in software.

13.13.4 Baud Rate Generation

An internal baud rate generator sets the UART baud rate when operating in UART mode and the SPI master clock frequency when operating in SPI mode.

The `UxBAUD.BAUD_M[7:0]` and `UxGCR.BAUD_E[4:0]` registers define the baud rate used for UART transfers and the rate of the serial clock for SPI transfers. The baud rate is given by the following equation:

$$Baudrate = \frac{(256 + BAUD_M) * 2^{BAUD_E}}{2^{28}} * F$$

where F is the system clock frequency, 16 MHz or 32 MHz.

The register values required for standard baud rates are shown in Table 39 for a typical system clock set to 32 MHz. The table also gives the difference in actual baud rate to standard baud rate value as a percentage error.

The maximum baud rate for UART mode is F/16 when BAUD_E is 16 and BAUD_M is 0, and where F is the system clock frequency.

The maximum baud rate for SPI mode and thus SCK frequency, is F/2 when BAUD_E is 19 and BAUD_M is 0. Setting higher baud rates than this will give erroneous results.

Baud rate (bps)	UxBAUD.BAUD_M	UxGCR.BAUD_E	Error (%)
2400	59	6	0.14
4800	59	7	0.14
9600	59	8	0.14
14400	216	8	0.03
19200	59	9	0.14
28800	216	9	0.03
38400	59	10	0.14
57600	216	10	0.03
76800	59	11	0.14
115200	216	11	0.03
230400	216	12	0.03

Table 39: Commonly used baud rate settings for 32 MHz system clock

13.13.5 USART flushing

The current operation can be aborted by setting the `UxUCR.FLUSH` register bit. This event will immediately stop the current operation and clear all data buffers.

13.13.6 USART Interrupts

Each USART has two interrupts. These are the RX complete interrupt (URXx) and the TX complete interrupt (UTXx).

The USART interrupt enable bits are found in the `IEN0` and `IEN2` registers. The interrupt flags are located in the `TCON` and `IRCON2` registers. Refer to section 12.7 on page 52 for details of these registers. The

interrupt enables and flags are summarized below.

Interrupt enables:

- USART0 RX : IEN0.URX0IE
- USART1 RX : IEN0.URX1IE
- USART0 TX : IEN2.UTX0IE
- USART1 TX : IEN2.UTX1IE

Interrupt flags:

- USART0 RX : TCON.URX0IF
- USART1 RX : TCON.URX1IF
- USART0 TX : IRCON2.UTX0IF
- USART1 TX : IRCON2.UTX1IF

13.13.7 USART DMA Triggers

There are two DMA triggers associated with each USART. The DMA triggers are activated by RX complete and TX complete events i.e. the same events as the USART interrupt requests. A DMA channel can be configured using a USART Receive/transmit buffer, UxDBUF, as source or destination address.

Refer to Table 37 on page 92 for an overview of the DMA triggers.

13.13.8 USART Registers

The registers for the USART are described in this section. For each USART there are five registers consisting of the following (x refers to USART number i.e. 0 or 1):

- UxCSR USART x Control and Status
- UxUCR USART x UART Control
- UxGCR USART x Generic Control
- UxDBUF USART x Receive/Transmit data buffer
- UxBAUD USART x Baud Rate Control

U0CSR (0x86) – USART 0 Control and Status

Bit	Name	Reset	R/W	Description
7	MODE	0	R/W	USART mode select 0 SPI mode 1 UART mode
6	RE	0	R/W	UART receiver enable 0 Receiver disabled 1 Receiver enabled
5	SLAVE	0	R/W	SPI master or slave mode select 0 SPI master 1 SPI slave
4	FE	0	R/W0	UART framing error status 0 No framing error detected 1 Byte received with incorrect stop bit level
3	ERR	0	R/W0	UART parity error status 0 No parity error detected 1 Byte received with parity error
2	RX_BYTE	0	R/W0	Receive byte status 0 No byte received 1 Received byte ready
1	TX_BYTE	0	R/W0	Transmit byte status 0 Byte not transmitted 1 Last byte written to Data Buffer register transmitted
0	ACTIVE	0	R	USART transmit/receive active status 0 USART idle 1 USART busy in transmit or receive mode

U0UCR (0xC4) – USART 0 UART Control

Bit	Name	Reset	R/W	Description
7	FLUSH	0	R0/W1	Flush unit. When set, this event will immediately stop the current operation and return the unit to idle state.
6	FLOW	0	R/W	UART hardware flow enable. Selects use of hardware flow control with RTS and CTS pins 0 Flow control disabled 1 Flow control enabled
5	D9	0	R/W	UART data bit 9 contents. This value is used when 9 bit transfer is enabled. When parity is disabled, the value written to D9 is transmitted as the bit 9 when 9 bit data is enabled. If parity is enabled then this bit sets the parity level as follows. 0 Odd parity 1 Even parity
4	BIT9	0	R/W	UART 9-bit data enable. When this bit is 1, data is 9 bits and the content of data bit 9 is given by D9 and PARITY. 0 8 bits transfer 1 9 bits transfer
3	PARITY	0	R/W	UART parity enable. 0 Parity disabled 1 Parity enabled
2	SPB	0	R/W	UART number of stop bits. Selects the number of stop bits to transmit 0 1 stop bit 1 2 stop bits
1	STOP	1	R/W	UART stop bit level 0 Low stop bit 1 High stop bit
0	START	0	R/W	UART start bit level. The polarity of the idle line is assumed the opposite of the selected start bit level. 0 Low start bit 1 High start bit

U0GCR (0xC5) – USART 0 Generic Control

Bit	Name	Reset	R/W	Description
7	CPOL	0	R/W	SPI clock polarity 0 Negative clock polarity 1 Positive clock polarity
6	CPHA	0	R/W	SPI clock phase 0 Data is output on *MOSI* when *SCK* goes from **CPOL** inverted to **CPOL**, and data input is sampled on *MISO* when *SCK* goes from **CPOL** to **CPOL** inverted. 1 Data is output on *MOSI* when *SCK* goes from **CPOL** to **CPOL** inverted, and data input is sampled on *MISO* when *SCK* goes from **CPOL** inverted to **CPOL.**
5	ORDER	0	R/W	Bit order for transfers 0 LSB first 1 MSB first
4:0	BAUD_E[4:0]	0x00	R/W	Baud rate exponent value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

U0DBUF (0xC1) – USART 0 Receive/Transmit Data Buffer

Bit	Name	Reset	R/W	Description
7:0	DATA[7:0]	0x00	R/W	USART receive and transmit data. When writing this register the data written is written to the internal, transmit data register. When reading this register, the data from the internal read data register is read.

U0BAUD (0xC2) – USART 0 Baud Rate Control

Bit	Name	Reset	R/W	Description
7:0	BAUD_M[7:0]	0x00	R/W	Baud rate mantissa value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

U1CSR (0xF8) – USART 1 Control and Status

Bit	Name	Reset	R/W	Description
7	MODE	0	R/W	USART mode select 0 SPI mode 1 UART mode
6	RE	0	R/W	UART receiver enable 0 Receiver disabled 1 Receiver enabled
5	SLAVE	0	R/W	SPI master or slave mode select 0 SPI master 1 SPI slave
4	FE	0	R/W0	UART framing error status 0 No framing error detected 1 Byte received with incorrect stop bit level
3	ERR	0	R/W0	UART parity error status 0 No parity error detected 1 Byte received with parity error
2	RX_BYTE	0	R/W0	Receive byte status 0 No byte received 1 Received byte ready
1	TX_BYTE	0	R/W0	Transmit byte status 0 Byte not transmitted 1 Last byte written to Data Buffer register transmitted
0	ACTIVE	0	R	USART transmit/receive active status 0 USART idle 1 USART busy in transmit or receive mode

U1UCR (0xFB) – USART 1 UART Control

Bit	Name	Reset	R/W	Description
7	FLUSH	0	R0/W1	Flush unit. When set, this event will immediately stop the current operation and return the unit to idle state.
6	FLOW	0	R/W	UART hardware flow enable. Selects use of hardware flow control with RTS and CTS pins 0 Flow control disabled 1 Flow control enabled
5	D9	0	R/W	UART data bit 9 contents. This value is used 9 bit transfer is enabled. When parity is disabled, the value written to D9 is transmitted as the bit 9 when 9 bit data is enabled. If parity is enabled then this bit sets the parity level as follows. 0 Odd parity 1 Even parity
4	BIT9	0	R/W	UART 9-bit data enable. When this bit is 1, data is 9 bits and the content of data bit 9 is given by D9 and PARITY. 0 8 bits transfer 1 9 bits transfer
3	PARITY	0	R/W	UART parity enable. 0 Parity disabled 1 Parity enabled
2	SPB	0	R/W	UART number of stop bits. Selects the number of stop bits to transmit 0 1 stop bit 1 2 stop bits
1	STOP	1	R/W	UART stop bit level 0 Low stop bit 1 High stop bit
0	START	0	R/W	UART start bit level. The polarity of the idle line is assumed the opposite of the selected start bit level. 0 Low start bit 1 High start bit

U1GCR (0xFC) – USART 1 Generic Control

Bit	Name	Reset	R/W	Description
7	CPOL	0	R/W	SPI clock polarity 0 Negative clock polarity 1 Positive clock polarity
6	CPHA	0	R/W	SPI clock phase 0 Data is output on *MOSI* when *SCK* goes from **CPOL** inverted to **CPOL**, and data input is sampled on *MISO* when *SCK* goes from **CPOL** to **CPOL** inverted. 1 Data is output on *MOSI* when *SCK* goes from **CPOL** to **CPOL** inverted, and data input is sampled on *MISO* when *SCK* goes from **CPOL** inverted to **CPOL.**
5	ORDER	0	R/W	Bit order for transfers 0 LSB first 1 MSB first
4:0	BAUD_E[4:0]	0x00	R/W	Baud rate exponent value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

U1DBUF (0xF9) – USART 1 Receive/Transmit Data Buffer

Bit	Name	Reset	R/W	Description
7:0	DATA[7:0]	0x00	R/W	USART receive and transmit data. When writing this register the data written is written to the internal, transmit data register. When reading this register, the data from the internal read data register is read.

U1BAUD (0xFA) – USART 1 Baud Rate Control

Bit	Name	Reset	R/W	Description
7:0	BAUD_M[7:0]	0x00	R/W	Baud rate mantissa value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

13.14 Flash Controller

The *CC2430* contains 32, 64 or 128 KB flash memory for storage of program code. The flash memory is programmable from the user software.

The Flash Controller handles writing and erasing the embedded flash memory. The embedded flash memory consists of 64 pages of 2048 bytes each (CC2430-F128). The flash memory is byte-addressable from the CPU and 32-bit word-programmable.

The flash controller has the following features:

- 32-bit word programmable
- Page erase
- Lock bits for write-protection and code security
- Flash erase timing 20 ms
- Flash write timing (4 bytes) 20 µs
- Auto power-down during low-frequency CPU clock read access

13.14.1 Flash Write

Data is written to the flash memory by using a program command initiated by writing the Flash Control register, FCTL. Flash write operations can program any number of locations in the flash memory at a time – it is however important to make sure the pages to be written are erased first.

A write operation is performed using one out of two methods;

- Through DMA transfer
- Through CPU SFR access.

The DMA transfer method is the preferred way to write to the flash memory.

A write operation is initiated by writing a 1 to FCTL.WRITE. The address to start writing at, is given by FADDRH:FADDRL. During each single, write operation FCTL.SWBSY is set high. During a write, operation the data written to the FWDATA register is forwarded to the flash memory. The flash memory is 32-bit word-programmable, meaning data is written as 32-bit words. Therefore, the actual writing to flash memory takes place each time four bytes have been written to FWDATA.

The CPU will not be able to access the flash, e.g. to read program code, while a flash write operation is in progress. Therefore the program code executing the flash write must be executed from RAM, meaning that the program code must reside in the area 0xE000 to 0xFF00 in CODE memory space.

When a flash write operation is executed from RAM, the CPU continues to execute code from the next instruction after the write to FWDATA, which initiated the flash write operation.

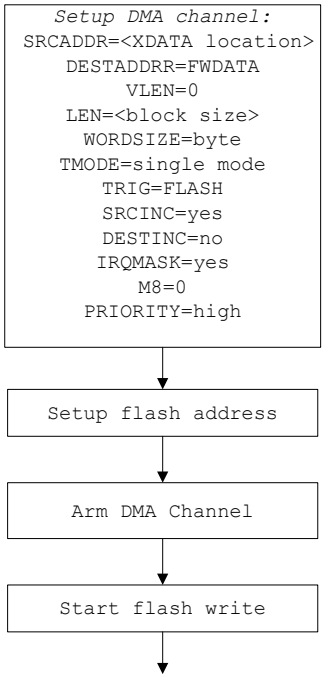
The FCTL.SWBSY bit must be 0 before accessing the flash after a flash write, otherwise an access violation occurs. This also means that FCTL.SWBSY must be 0 before program execution can continue at a location in flash memory.

13.14.1.1 DMA Flash Write

When using DMA write operations, the data to be written into flash is stored in Data/XDATA memory. A DMA channel is configured to read the data to be written from memory and write this data to the Flash Write Data register, FWDATA, with the DMA trigger event FLASH enabled. Thus the Flash Controller will trigger a DMA transfer when the Flash Write Data register, FWDATA, is ready to receive new data. The DMA channel should be configured to perform a block to fixed, single mode, byte size transfers.

When the DMA channel is armed, starting a flash write will trigger the first DMA transfer.

Figure 34 shows an example how a DMA channel is configured and how a DMA transfer is initiated to write a block of data from a location in XDATA to flash memory.




```
; Write a consecutive block of data from XDATA to consecutive locations in
; flash memory using DMA
; Assumes 32 MHz system clock is used
;
      MOV    DPTR,#DMACFG         ;load data pointer with address for DMA
                                  ;channel configuration and
                                  ;start writing DMA configuration
      MOV    A,#SRC_HI            ;source data high address
      MOVX   @DPTR,A              ;
      INC    DPTR                 ;
      MOV    A,#SRC_LO            ;source data low address
      MOVX   @DPTR,A              ;
      INC    DPTR                 ;
      MOV    A,#0DFh              ;destination high address = HIGH(X_FWDATA)
      MOVX   @DPTR,A              ;
      INC    DPTR                 ;
      MOV    A,#0AFh              ;destination low address = LOW(X_FWDATA)
      MOVX   @DPTR,A              ;
      INC    DPTR                 ;
      MOV    A,#BLK_LEN           ;block length
      MOVX   @DPTR,A              ;
      INC    DPTR                 ;
      MOV    A,#012h              ;8 bits, single mode, use FLASH trigger
      MOVX   @DPTR,A              ;
      INC    DPTR                 ;
      MOV    A,#042h              ;increment source by 1, don't increment
      MOVX   @DPTR,A              ;destination, mask interrupt, high DMA
                                  ;priority
      MOV    DMA0CFGL,#DMACFG_LO  ;setup start address for current DMA
      MOV    DMA0CFGH,#DMACFG_HI  ;configuration
      MOV    DMAARM,#01h          ;arm DMA channel 0
      MOV    FADDRH,#00h          ;setup flash address high
      MOV    FADDRL,#01h          ;setup flash address low
      MOV    FWT,#2Ah             ;setup flash timing
      MOV    FCTL,#02h            ;start flash page write => trigger DMA
      .
      .
```

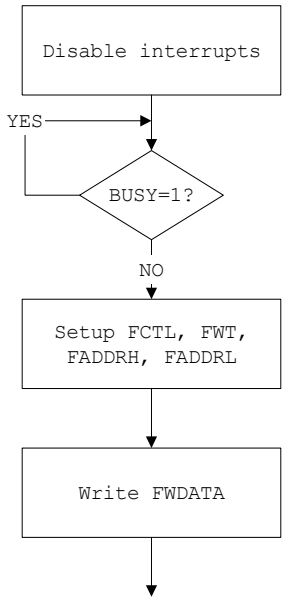
Figure 34: Flash write using DMA

13.14.1.2 CPU Flash Write

The CPU can also write directly to the flash. The CPU writes data to the Flash Write Data register, FWDATA. The flash memory is written each time four bytes have been written to FWDATA. The CPU can poll the FCTL.SWBSY status to determine when the flash is ready for four more bytes to be written to FWDATA

Performing flash write from XDATA

The steps required to start a flash write operation from XDATA are shown in Figure 35 on page 159.




```
; Write 32-bit word from XDATA
; Assumes 32 MHz system clock is used
;
       CLR    EA                  ;mask interrupts
C1:    MOV    A,FCTL              ;wait until flash controller is ready
       JB     ACC.7,C1
       MOV    FADDRH,#00h         ;setup flash address high
       MOV    FADDRL,#01h         ;setup flash address low
       MOV    FWT,#2Ah            ;setup flash timing
       MOV    FCTL,#02h           ;set flash page write
       MOV    FWDATA,#12h         ;first byte
       MOV    FWDATA,#34h         ;second byte
       MOV    FWDATA,#56h         ;third byte
       MOV    FWDATA,#78h         ;fourth byte, initiates write
```

Figure 35: Flash write performed from XDATA

13.14.2 Flash Page Erase

A page erase is initiated by setting FCTL.ERASE to 1. The page addressed by FADDRH[6:1] is erased when a page erase is initiated. Note that if a page erase is initiated simultaneously with a page write, i.e. FCTL.WRITE is set to 1, the page erase will be performed before the page write operation. The FCTL.BUSY bit can be polled to see when the page erase has completed.

Note: if flash erase operations are performed from within flash memory and the watchdog timer is enabled, a watchdog timer interval must be selected that is longer than 20 ms, the duration of the flash erase operation, so that the CPU will manage to clear the watchdog timer.

Performing flash erase from flash memory.

The steps required to perform a flash page erase from within flash memory are outlined in Figure 36.

Note that, while executing program code from within flash memory, when a flash erase or write operation is initiated, program execution will resume from the next instruction when the flash controller has completed the operation.

```
; Erase page in flash memory
; Assumes 32 MHz system clock is used
;
       CLR    EA                  ;mask interrupts
C1:    MOV    A,FCTL              ;wait until flash controller is ready
       JB     ACC.7,C1
       MOV    FADDRH,#00h         ;setup flash address high
       MOV    FWT,#2Ah            ;setup flash timing
       MOV    FCTL,#01h           ;erase page
       RET                        ;continues here when flash is ready
```

Figure 36: Flash page erase performed from flash memory

13.14.3 Flash Lock Protection

For software protection purposes a set of lock protection bits can be written once after each chip erase has been performed. The lock protect bits can only be written through the Debug Interface. There are three kinds of lock protect bits as described in this section. The flash lock bits reside at location 0x000 in the Flash Information page as described in section 12.11.

The LSIZE[2:0] lock protect bits are used to define a section of the flash memory which is write protected. The size of the write protected area can be set by the LSIZE[2:0] lock protect bits in sizes of eight steps from 0 to 128 KB.

The second type of lock protect bits is BBLOCK, which is used to lock the boot sector page (page 0 ranging from address 0 to 0x07FF). When BBLOCK is set to 0, the boot sector page is locked.

The third type of lock protect bit is DBGLOCK, which is used to disable hardware debug support through the Debug Interface. When DBGLOCK is set to 0, all debug commands are disabled.

The lock protect bits are written as a normal flash write to FWDATA, but the Debug Interface needs to select the Flash Information Page first instead of the Flash Main Page which is the default setting. The Information Page is selected through the Debug Configuration which is written through the Debug Interface only. Refer to section 12.9 on page 62 for details on how the Flash Information Page is selected using the Debug Interface.

Table 40 defines the byte containing the flash lock protection bits. Note that this is not an SFR register, but instead the byte stored at location 0x000 in Flash Information Page.

Table 40: Flash Lock Protection Bits Definition

Bit	Name	Description
7:5	-	Reserved, write as 0
4	BBLOCK	Boot Block Lock 0 Page 0 is write protected 1 Page 0 is writeable, unless LSIZE is 000
3:1	LSIZE[2:0]	Lock Size. Sets the size of the upper Flash area which is write-protected. Byte sizes and page number are listed below 000 128k bytes (All pages) CC2430-F128 only 001 64k bytes (page 32 - 63) CC2430-F64/128 only 010 32k bytes (page 48 - 63) 011 16k bytes (page 56 - 63) 100 8k bytes (page 60 - 63) 101 4k bytes (page 62 - 63) 110 2k bytes (page 63) 111 0k bytes (no pages)
0	DBGLOCK	Debug lock bit 0 Disable debug commands 1 Enable debug commands

13.14.4 Flash Write Timing

The Flash Controller contains a timing generator, which controls the timing sequence of flash write and erase operations. The timing generator uses the information set in the Flash Write Timing register, FWT.FWT[5:0], to set the internal timing. FWT.FWT[5:0] must be set to a value according to the currently selected CPU clock frequency.

The value set in the FWT.FWT[5:0] shall be set according to the CPU clock frequency by the following equation.

$$FWT = \frac{21000 * F_{CPU}}{16 * 10^9}$$

F_{CPU} is the CPU clock frequency. The initial value held in FWT.FWT[5:0] after a reset is 0x2A which corresponds to 32 MHz CPU clock frequency.

The FWT values for the 16 MHz and 32 MHz CPU clock frequencies are given in Table 41.

CPU clock frequency (MHz)	FWT
16	0x15
32	0x2A

Table 41: Flash timing (FWT) values

13.14.5 Flash DMA trigger

The Flash DMA trigger is activated when flash data written to the FWDATA register has been written to the specified location in the flash memory, thus indicating that the flash controller is ready to accept new data to be written to FWDATA.

13.14.6 Flash Controller Registers

The Flash Controller registers are described in this section.

FCTL (0xAE) – Flash Control

Bit	Name	Reset	R/W	Description
7	BUSY	0	R	Indicates that write or erase is in operation
6	SWBSY	0	R	Indicates that single write is busy; avoid writing to FWDATA register while this is true
5	–	0	R/W	Not used.
4	CONTRD	0	R/W	Continuous read enable mode 0 Avoid wasting power; turn on read enables to flash only when needed 1 Enable continuous read enables to flash when read is to be done. Reduces internal switching of read enables, but greatly increases power consumption.
3:2		0	R/W	Not used.
1	WRITE	0	R0/W	Page Write. Start writing page given by FADDRH:FADDRL. If ERASE is set to 1, a page erase is performed before the write.
0	ERASE	0	R0/W	Page Erase. Erase page that is given by FADDRH[6:1]

FWDATA (0xAF) – Flash Write Data

Bit	Name	Reset	R/W	Description
7:0	FWDATA[7:0]	0x00	R/W	Flash write data. Data written to FWDATA is written to flash when FCTL.WRITE is set to 1.

FADDRH (0xAD) – Flash Address High Byte

Bit	Name	Reset	R/W	Description
7	-	0	R/W	Not used
6:0	FADDRH[6:0]	0x00	R/W	High byte of flash address Bits 6:1 will select page to access, while bit 0 is MSB of row access.

FADDRL (0xAC) – Flash Address Low Byte

Bit	Name	Reset	R/W	Description
7:0	FADDRL[7:0]	0x00	R/W	Low byte of flash address Bit 0 of FADDRH and bits 7:6 will select which row to write to, while bits 5:0 will select which location to write to.

FWT (0xAB) – Flash Write Timing

Bit	Name	Reset	R/W	Description
7:6	–	00	R/W	Not used
5:0	FWT[5:0]	0x2A	R/W	Flash Write Timing. Controls flash timing generator.

14 Radio

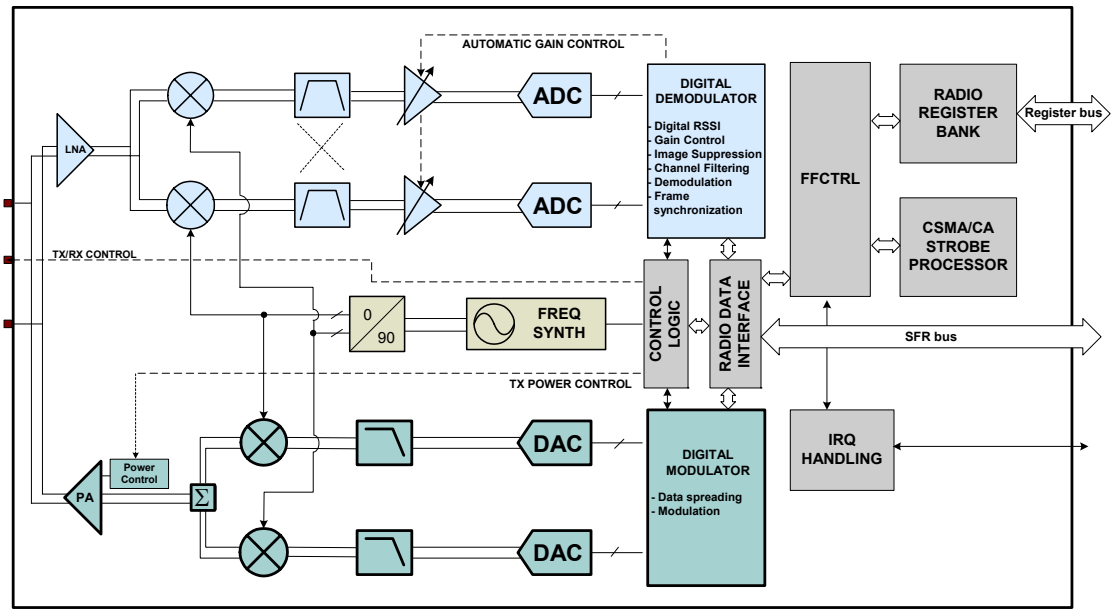


Figure 37: CC2430 Radio Module

A simplified block diagram of the IEEE 802.15.4 compliant radio inside ***CC2430*** is shown in Figure 37. The radio core is based on the industry leading ***CC2420*** RF transceiver.

CC2430 features a low-IF receiver. The received RF signal is amplified by the low-noise amplifier (LNA) and down-converted in quadrature (I and Q) to the intermediate frequency (IF). At IF (2 MHz), the complex I/Q signal is filtered and amplified, and then digitized by the ADCs. Automatic gain control, final channel filtering, de-spreading, symbol correlation and byte synchronization are performed digitally.

An interrupt indicates that a start of frame delimiter has been detected. ***CC2430*** buffers the received data in a 128 byte receive FIFO. The user may read the FIFO through an SFR interface. It is recommended to use direct memory access (DMA) to move data between memory and the FIFO.

CRC is verified in hardware. RSSI and correlation values are appended to the frame. Clear channel assessment, CCA, is available through an interrupt in receive mode.

The ***CC2430*** transmitter is based on direct up-conversion. The data is buffered in a 128 byte transmit FIFO (separate from the receive FIFO). The preamble and start of frame delimiter are generated in hardware. Each symbol (4 bits) is spread using the IEEE 802.15.4 spreading sequence to 32 chips and output to the digital-to-analog converters (DACs).

An analog low pass filter passes the signal to the quadrature (I and Q) up-conversion mixers. The RF signal is amplified in the power amplifier (PA) and fed to the antenna.

The internal T/R switch circuitry makes the antenna interface and matching easy. The RF connection is differential. A balun may be used for single-ended antennas. The biasing of the PA and LNA is done by connecting `TXRX_SWITCH` to `RF_P` and `RF_N` through an external DC path.

The frequency synthesizer includes a completely on-chip LC VCO and a 90 degrees phase splitter for generating the I and Q LO signals to the down-conversion mixers in receive mode and up-conversion mixers in transmit mode. The VCO operates in the frequency range 4800 –

4966 MHz, and the frequency is divided by two when split into I and Q signals.

The digital baseband includes support for frame handling, address recognition, data buffering, CSMA-CA strobe processor and MAC security.

An on-chip voltage regulator delivers the regulated 1.8 V supply voltage.

14.1 IEEE 802.15.4 Modulation Format

This section is meant as an introduction to the 2.4 GHz direct sequence spread spectrum (DSSS) RF modulation format defined in IEEE 802.15.4. For a complete description, please refer to [1].

The modulation and spreading functions are illustrated at block level in Figure 38 [1]. Each byte is divided into two symbols, 4 bits each. The least significant symbol is transmitted first. For multi-byte fields, the least significant byte is transmitted first.

Each symbol is mapped to one out of 16 pseudo-random sequences, 32 chips each. The symbol to chip mapping is shown in Table 42. The chip sequence is then transmitted at 2 MChips/s, with the least significant chip (C_0) transmitted first for each symbol.

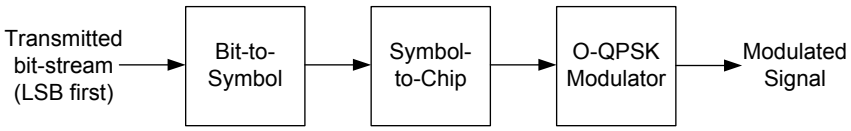


Figure 38: Modulation and spreading functions [1]

Symbol	Chip sequence (C_0, C_1, C_2, … , C_{31})
0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 1 0 1 1 0 1
8	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
11	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

Table 42: IEEE 802.15.4 symbol-to-chip mapping [1]

The modulation format is Offset – Quadrature Phase Shift Keying (O-QPSK) with half-sine chip shaping. This is equivalent to MSK modulation. Each chip is shaped as a half-sine, transmitted alternately in the I and Q channels with one half chip period offset. This is illustrated for the zero-symbol in Figure 39.

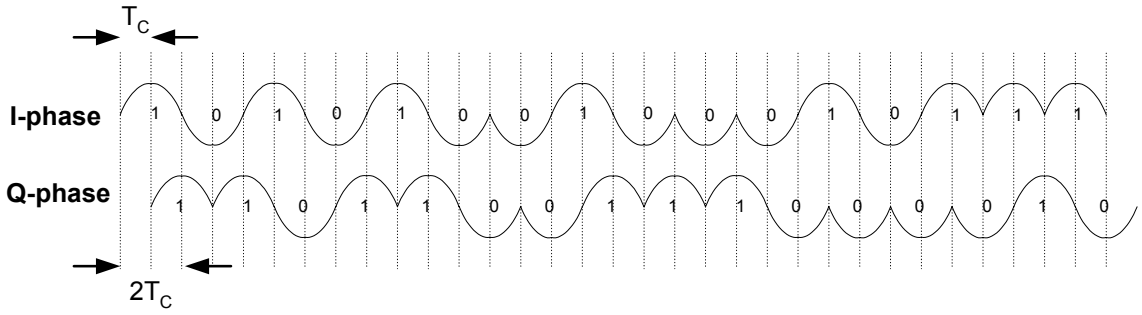


Figure 39: I / Q Phases when transmitting a zero-symbol chip sequence, T_C = 0.5 µs

14.2 Command strobes

The CPU uses a set of *command strobes* to control operation of the radio in ***CC2430***.

Command strobes may be viewed as single byte instructions which each control some function of the radio. These command strobes must be used to enable the frequency synthesizer, enable receive mode, enable transmit mode and other functions.

A total of nine command strobes are defined for the radio and these can be written individually to the radio or they can be given in a sequence together with a set of dedicated software instructions making up a simple program. All command strobes from the CPU to the radio pass through the CSMA-CA/Command Strobe Processor (CSP). Detailed description about the CSP and how command strobes are used is given in section 14.34 on page 189.

14.3 RF Registers

The operation of the radio is configured through a set of RF registers. These RF registers are mapped to XDATA memory space as shown in

Figure 13 on page 38.

The RF registers also provide status information from the radio.

The RF registers control/status bits are referred to where appropriate in the following sections while section 14.35 on page 203 gives a full description of all RF registers.

14.4 Interrupts

The radio is associated with two interrupt vectors on the CPU. These are the RFERR interrupt (interrupt 0) and the RF interrupt (interrupt 12) with the following functions

- RFERR : TXFIFO underflow, RXFIFO overflow
- RF : all other RF interrupts given by `RFIF` interrupt flags

The RF interrupt vector combines the interrupts in RFIF shown on page 168. Note that these RF interrupts are rising-edge triggered. Thus an interrupt is generated when e.g. the SFD status flag goes from 0 to 1.

The RF interrupt can also be used to trigger a timer capture in Timer 1.

The RFIF interrupt flags are described in the next section.

14.4.1 Interrupt registers

Two of the main interrupt control SFR registers are used to enable the RF and RFERR interrupts. These are the following:

- RFERR : `IEN0.RFERRIE`
- RF : `IEN2.RFIE`

Two main interrupt flag SFR registers hold the RF and RFERR interrupt flags. These are the following:

- RFERR : `TCON.RFERR`
- RF : `S1CON.RFIF`

Refer to section 12.7 on page 52 for details about the interrupts.

The RF interrupt is the combined interrupt from eight different sources in the radio. Two SFR registers are used for setting the eight individual RFIF radio interrupt flags and interrupt enables. These are the RFIF and RFIM registers.

The interrupt flags in SFR register RFIF show the status for each interrupt source for the RF interrupt vector.

The interrupt enable bits in RFIM are used to disable individual interrupt sources for the RF interrupt vector. Note that masking an interrupt source in RFIM does not affect the update of the status in the RFIF register.

Due to the use of the individual interrupt masks in RFIM, and the main interrupt mask for the RF interrupt given by IEN2.RFIE there is two-layered masking of this interrupt. Special attention needs to be taken when processing this type of interrupt as described below.

To clear the RF interrupt, S1CON.RFIF and the interrupt flag in RFIF need to be cleared. The order and method of doing this is shown in Figure 40. Note that S1CON is cleared after RFIF, otherwise S1CON.RFIF could be set once again due to the same interrupt.

```
MOV    RFIF,#00h          ;clear all interrupt flags
MOV    S1CON,#00h         ;clear main interrupt flags
MOV    RFIM,RFIM          ;set interrupt mask
```

Figure 40: Clearing RF Interrupt

RFIF (0xE9) – RF Interrupt Flags

Bit	Name	Reset	R/W	Description
7	IRQ_RREG_ON	0	R/W0	Voltage regulator for radio has been turned on 0 No interrupt pending 1 Interrupt pending
6	IRQ_TXDONE	0	R/W0	TX completed with packet sent 0 No interrupt pending 1 Interrupt pending
5	IRQ_FIFOP	0	R/W0	Number of bytes in RXFIFO is above threshold set by IOCFG0.FIFOP_THR 0 No interrupt pending 1 Interrupt pending
4	IRQ_SFD	0	R/W0	Start of frame delimiter (SFD) has been detected 0 No interrupt pending 1 Interrupt pending
3	IRQ_CCA	0	R/W0	Clear channel assessment (CCA) indicates that channel is clear 0 No interrupt pending 1 Interrupt pending
2	IRQ_CSP_WT	0	R/W0	CSMA-CA/strobe processor (CSP) wait condition is true 0 No interrupt pending 1 Interrupt pending
1	IRQ_CSP_STOP	0	R/W0	CSMA-CA/strobe processor (CSP) program execution stopped 0 No interrupt pending 1 Interrupt pending
0	IRQ_CSP_INT	0	R/W0	CSMA-CA/strobe processor (CSP) INT instruction executed 0 No interrupt pending 1 Interrupt pending

RFIM (0x91) – RF Interrupt Mask

Bit	Name	Reset	R/W	Description
7	IM_RREG_PD	0	R/W	Voltage regulator for radio has been turned on 0 Interrupt disabled 1 Interrupt enabled
6	IM_TXDONE	0	R/W	TX completed with packet sent 0 Interrupt disabled 1 Interrupt enabled
5	IM_FIFOP	0	R/W	Number of bytes in RXFIFO is above threshold set by IOCFG0.FIFOP_THR 0 Interrupt disabled 1 Interrupt enabled
4	IM_SFD	0	R/W	Start of frame delimiter (SFD) has been detected 0 Interrupt disabled 1 Interrupt enabled
3	IM_CCA	0	R/W	Clear channel assessment (CCA) indicates that channel is clear 0 Interrupt disabled 1 Interrupt enabled
2	IM_CSP_WT	0	R/W	CSMA-CA/strobe processor (CSP) wait condition is true 0 Interrupt disabled 1 Interrupt enabled
1	IM_CSP_STOP	0	R/W	CSMA-CA/strobe processor (CSP) program execution stopped 0 Interrupt disabled 1 Interrupt enabled
0	IM_CSP_INT	0	R/W	CSMA-CA/strobe processor (CSP) INT instruction executed 0 Interrupt disabled 1 Interrupt enabled

14.5 FIFO access

The TXFIFO and RXFIFO may be accessed through the SFR register RFD (0xD9).

Data is written to the TXFIFO when writing to the RFD register. Data is read from the he RXFIFO when the RFD register is read.

The RF register bits RFSTATUS.FIFO and RFSTATUS.FIFOP provide information on the data in the receive FIFO, as described in section 14.6 on page 170. Note that the RFSTATUS.FIFO and RFSTATUS.FIFOP only apply to the RXFIFO.

The TXFIFO may be flushed by issuing a SFLUSHTX command strobe. Similarly, a SFLUSHRX command strobe will flush the receive FIFO.

RFD (0xD9) – RF Data

Bit	Name	Reset	R/W	Description
7:0	RFD[7:0]	0x00	R/W	Data written to the register is written to the TXFIFO. When reading this register, data from the RXFIFO is read

14.6 DMA

It is possible, and in most cases recommended, to use direct memory access (DMA) to move data between memory and the radio. The DMA controller is described in section 13.2. Refer to this section for a detailed description on how to setup and use DMA transfers.

To support the DMA controller there is one DMA trigger associated with the radio, this is the RADIO DMA trigger (DMA trigger 19). The RADIO DMA trigger is activated by two events. The first event to cause a RADIO DMA trigger, is when the first data is present in the RXFIFO, i.e. when the RXFIFO goes from the empty state to the non-empty state. The second event that causes a RADIO DMA trigger, is when data is read from the RXFIFO (through RFD SFR register) and there is still more data available in the RXFIFO.

14.7 Receive mode

In receive mode, the interrupt flag RFIF.IRQ_SFD goes high and the RF interrupt is requested after the start of frame delimiter (SFD) field has been completely received. If address recognition is disabled or is successful, the RFSTATUS.SFD bit goes low again only after the last byte of the MPDU has been received. If the received frame fails address recognition, the RFSTATUS.SFD bit goes low immediately. This is illustrated in Figure 41.

The RFSTATUS.FIFO bit is high when there is one or more data bytes in the RXFIFO. The first byte to be stored in the RXFIFO is the length field of the received frame, i.e. the RFSTATUS.FIFO bit is set high when the length field is written to the RXFIFO. The RFSTATUS.FIFO bit then remains high until the RXFIFO is empty.

The RF register RXFIFOCNT contains the number of bytes present in the RXFIFO.

The RFSTATUS.FIFOP bit is high when the number of unread bytes in the RXFIFO exceeds the threshold programmed into IOCFG0.FIFOP_THR. When address recognition is enabled the RFSTATUS.FIFOP bit will not go high until the incoming frame passes address recognition, even if the number of bytes in the RXFIFO exceeds the programmed threshold.

The RFSTATUS.FIFOP bit will also go high when the last byte of a new packet is received, even if the threshold is not exceeded. If so the RFSTATUS.FIFOP bit will go back to low once one byte has been read out of the RXFIFO.

When address recognition is enabled, data should not be read out of the RXFIFO before the address is completely received, since the frame may be automatically flushed by ***CC2430*** if it fails address recognition. This may be handled by using the RFSTATUS.FIFOP bit, since this bit does not go high until the frame passes address recognition.

Figure 42 shows an example of status bit activity when reading a packet from the RXFIFO. In this example, the packet size is 8 bytes, IOCFG0.FIFOP_THR = 3 and MODEMCTRL0.AUTOCRC is set. The length will be 8 bytes, RSSI will contain the average RSSI level during receiving of the packet and FCS/corr contain information of FCS check result and the correlation levels.

14.8 RXFIFO overflow

The RXFIFO can only contain a maximum of 128 bytes at a given time. This may be divided between multiple frames, as long as the total number of bytes is 128 or less. If an overflow occurs in the RXFIFO, this is

signaled to the CPU by asserting the RFERR interrupt when enabled. In addition the radio will set RFSTATUS.FIFO bit low while the RFSTATUS.FIFOP bit is high. Data already in the RXFIFO will not be affected by the overflow, i.e. frames already received may be read out.

A SFLUSHRX command strobe is required after a RXFIFO overflow to enable reception of new data. Note that the SFLUSHRX command strobe should be issued twice to ensure that the RFSTATUS.SFD bit goes back to its idle state.

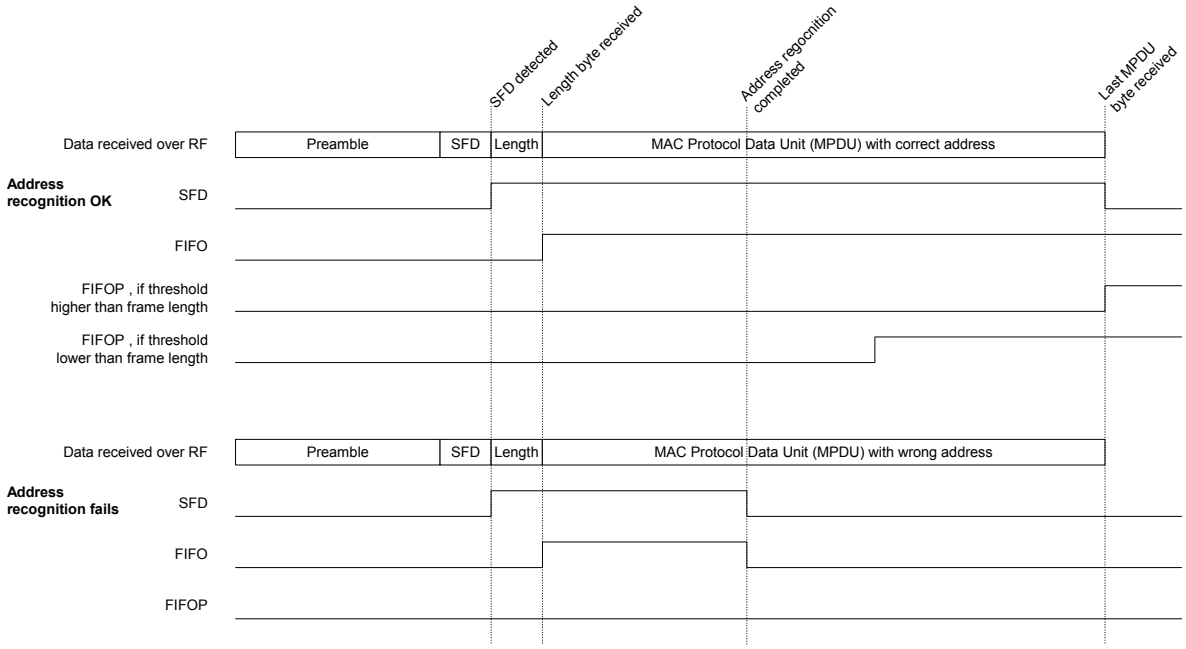


Figure 41: SFD, FIFO and FIFOP activity examples during receive

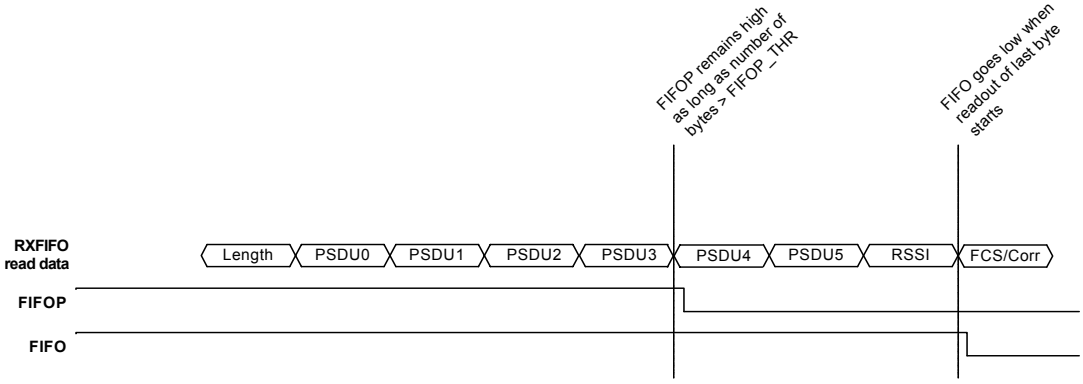


Figure 42: Example of status activity when reading RXFIFO.

14.9 Transmit mode

During transmit, the RFSTATUS.FIFO and RFSTATUS.FIFOP bits are still only related to the RXFIFO. The RFSTATUS.SFD bit is however active during transmission of a data frame, as shown in

Figure 43.

The RFIF.IRQ_SFD interrupt flag goes high and the RF interrupt is requested when the SFD field has been completely transmitted. It goes low again when the complete MPDU (as defined by the length field) has been transmitted or if an underflow is detected. The interrupt RFERR is asserted when enabled. See section 14.17.1 on page 176 for more information on TXFIFO underflow.

As can be seen from comparing Figure 41 and

Figure 43, the RFSTATUS.SFD bit behaves very similarly during reception and transmission of a data frame. If the RFSTATUS.SFD bits of the transmitter and the receiver are compared during the transmission of a data frame, a small delay of approximately 2 µs can be seen because of bandwidth limitations in both the transmitter and the receiver.

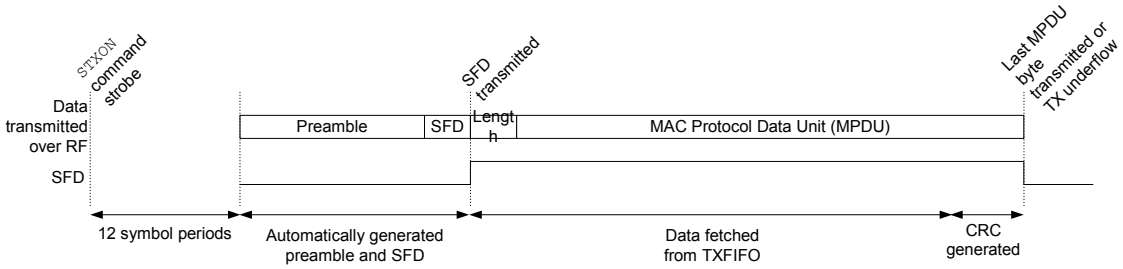


Figure 43: SFD status activity example during transmit

14.10 General control and status

In receive mode, the RFIF.IRQ_FIFOP interrupt flag and RF interrupt request can be used to interrupt the CPU when a threshold has been exceeded or a complete frame has been received.

In receive mode, the RFSTATUS.FIFO bit can be used to detect if there is data at all in the receive FIFO.

The RFIF.IRQ_SFD interrupt flag can be used to extract the timing information of transmitted and received data frames. The RFIF.IRQ_SFD bit will go high when a start of frame delimiter has been completely detected / transmitted.

For debug purposes, the RFSTATUS.SFD, RFSTATUS.FIFO, RFSTATUS.FIFOP and RFSTATUS.CCA bits can be output onto P1.7 – P1.4 I/O pins to monitor the status of these signals as selected by the IOCFG0, IOCFG1 and IOCFG2 register.

The polarity of these signals given on the debug outputs can also be controlled by the IOCFG0-2 registers, if needed.

14.11 Demodulator, Symbol Synchronizer and Data Decision

The block diagram for the ***CC2430*** demodulator is shown in Figure 44. Channel filtering and frequency offset compensation is performed digitally. The signal level in the channel is estimated to generate the RSSI level (see the RSSI / Energy Detection section on page 181 for more information). Data filtering is also included for enhanced performance.

With the ±40 ppm frequency accuracy requirement from [1], a compliant receiver must be able to compensate for up to 80 ppm or 200 kHz. The ***CC2430*** demodulator tolerates up to 300 kHz offset without

significant degradation of the receiver performance.

Soft decision is used at the chip level, i.e. the demodulator does not make a decision for each chip, only for each received symbol. De-spreading is performed using over-sampling symbol correlators. Symbol synchronization is achieved by a continuous start of frame delimiter (SFD) search.

When an SFD is detected, data is written to the RXFIFO and may be read out by the CPU at a lower bit rate than the 250 kbps generated by the receiver.

The ***CC2430*** demodulator also handles symbol rate errors in excess of 120 ppm without performance degradation. Resynchronization is performed continuously to adjust for error in the incoming symbol rate.

The RF register `MDMCTRL1H.CORR_THR` control bits should be written to 20 to set the threshold for detecting IEEE 802.15.4 start of frame delimiters.

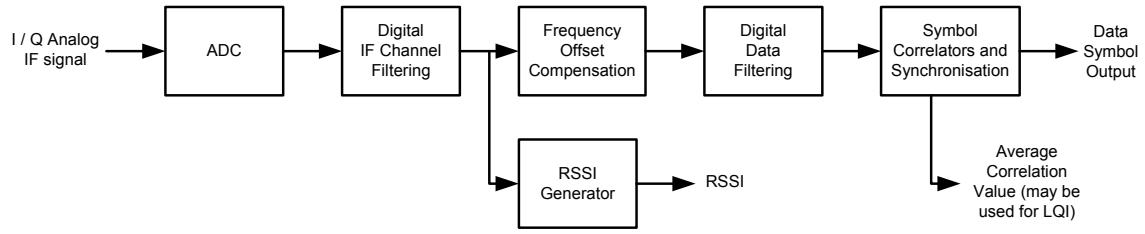


Figure 44: Demodulator Simplified Block Diagram

14.12 Frame Format

CC2430 has hardware support for parts of the IEEE 802.15.4 frame format. This section gives a brief summary to the IEEE 802.15.4 frame format, and describes how ***CC2430*** is set up to comply with this.

Figure 45 [1] shows a schematic view of the IEEE 802.15.4 frame format. Similar figures describing specific frame formats (data frames, beacon frames, acknowledgment frames and MAC command frames) are included in [1].

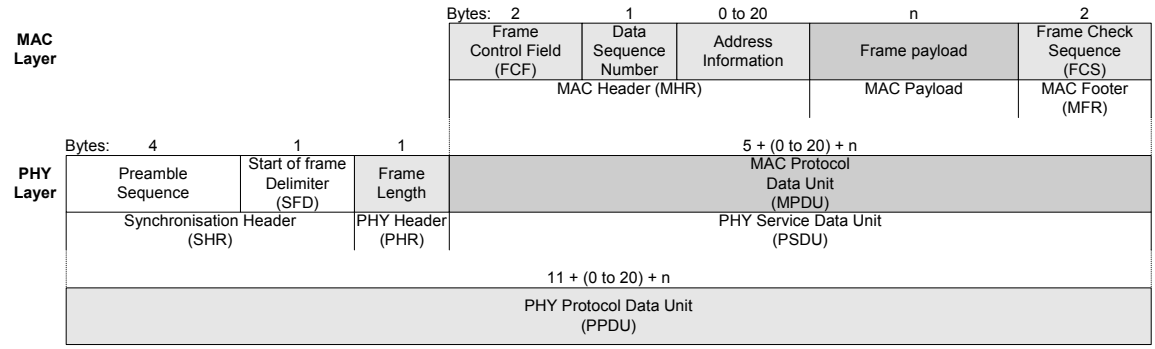


Figure 45: Schematic view of the IEEE 802.15.4 Frame Format [1]

14.13 Synchronization header

The synchronization header (SHR) consists of the preamble sequence followed by the start of frame delimiter (SFD). In [1], the preamble sequence is defined to be four bytes of 0x00. The SFD is one byte, set to 0xA7.

In ***CC2430***, the preamble length and SFD is configurable. The default values are

compliant with [1]. Changing these values will make the system non-compliant to IEEE 802.15.4.

A synchronization header is always transmitted first in all transmit modes.

The preamble sequence length can be set with RF register bit MDMCTRL0L.PREAMBLE_LENGTH, while the SFD is programmed in the SYNCWORDH:SYNCWORDL registers. SYNCWORDH:SYNCWORDL is two bytes long, which gives the user some extra flexibility as described below. Figure 46 shows how the *CC2430* synchronization header relates to the IEEE 802.15.4 specification.

The programmable preamble length only applies to transmission, it does not affect receive mode. The preamble length should not be set shorter than the default value. Note that 2 of the 8 zero-symbols in the preamble sequence required by [1] are included in the SYNCWORDH:SYNCWORDL registers so that the *CC2430* preamble sequence is only 6 symbols long for compliance with [1]. Two additional zero symbols in SYNCWORDH:SYNCWORDL make *CC2430* compliant with [1].

In reception, *CC2430* synchronizes to received zero-symbols and searches for the SFD sequence defined by the SYNCWORDH:SYNCWORDL registers. The least significant symbols in SYNCWORDH:SYNCWORDL set to 0xF will be ignored, while symbols different from 0xF will be required for synchronization. The default setting of 0xA70F thereby requires one additional zero-symbol for synchronization. This will reduce the number of false frames detected due to noise.

In receive mode *CC2430* uses the preamble sequence for symbol synchronization and frequency offset adjustments. The SFD is used for byte synchronization, and is not part of the data stored in the receive buffer (RXFIFO).

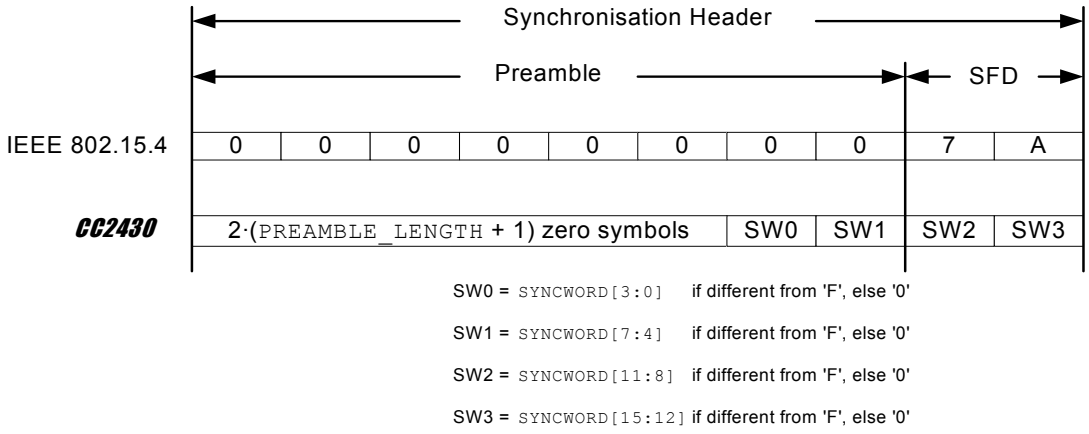


Figure 46: Transmitted Synchronization Header

14.14 Length field

The frame length field shown in Figure 45 defines the number of bytes in the MPDU. Note that the length field does not include the length field itself. It does however include the FCS (Frame Check Sequence), even if this is inserted automatically by *CC2430* hardware.

The length field is 7 bits and has a maximum value of 127. The most significant bit in the length field is reserved [1], and should be set to zero.

CC2430 uses the length field both for transmission and reception, so this field must always be included. In transmit mode, the length field is used for underflow detection, as described in the FIFO access section on page 169.

14.15 MAC protocol data unit

The FCF, data sequence number and address information follows the length field as shown in Figure 45. Together with the MAC data payload and Frame Check Sequence, they form the MAC Protocol Data Unit (MPDU).

The format of the FCF is shown in Figure 47. Please refer to [1] for details.

There is no hardware support for the data sequence number, this field must be inserted and verified by software.

CC2430 includes hardware address recognition, as described in the Address Recognition section on page 177.

Bits: 0-2	3	4	5	6	7-9	10-11	12-13	14-15
Frame Type	Security Enabled	Frame Pending	Acknowledge request	Intra PAN	Reserved	Destination addressing mode	Reserved	Source addressing mode

Figure 47: Format of the Frame Control Field (FCF) [1]

14.16 Frame check sequence

A 2-byte frame check sequence (FCS) follows the last MAC payload byte as shown in Figure 45. The FCS is calculated over the MPDU, i.e. the length field is not part of the FCS. This field is automatically generated and verified by hardware when the RF register `MDMCTRL0L.AUTOCRC` control bit is set. It is recommended to always have this enabled, except possibly for debug purposes. If cleared, CRC generation and verification must be performed by software.

The FCS polynomial is [1]:

$$x^{16} + x^{12} + x^5 + 1$$

The *CC2430* hardware implementation is shown in Figure 48. Please refer to [1] for further details.

In transmit mode the FCS is appended at the correct position defined by the length field. The FCS is not written to the TXFIFO, but stored in a separate 16-bit register.

In receive mode the FCS is verified by hardware. The user is normally only interested in the correctness of the FCS, not the FCS sequence itself. The FCS sequence itself is therefore not written to the RXFIFO during receive.

Instead, when `MDMCTRL0L.AUTOCRC` is set the two FCS bytes are replaced by the RSSI value, average correlation value (used for LQI) and CRC OK/not OK. This is illustrated in Figure 49.

The first FCS byte is replaced by the 8-bit RSSI value. See the RSSI section on page 181 for details.

The seven least significant bits in the last FCS byte are replaced by the average correlation value of the 8 first symbols of the received PHY header (length field) and PHY Service Data Unit (PSDU). This correlation value may be used as a basis for calculating the LQI. See the Link Quality Indication section on page 182 for details.

The most significant bit in the last byte of each frame is set high if the CRC of the received frame is correct and low otherwise.

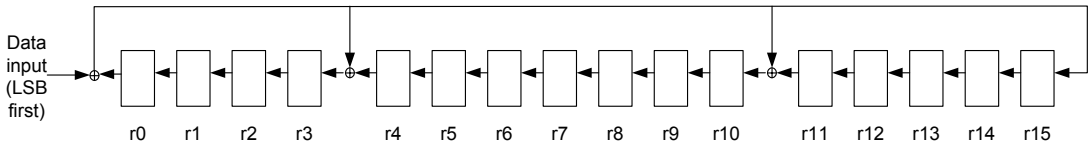


Figure 48: *CC2430* Frame Check Sequence (FCS) hardware implementation [1]

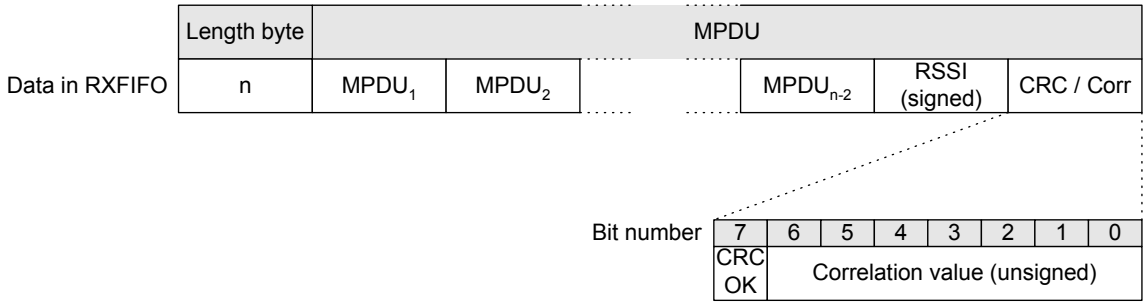


Figure 49: Data in RXFIFO when MDMCTRL0L.AUTOCRC is set

14.17 RF Data Buffering

CC2430 can be configured for different transmit and receive modes, as set in the MDMCTRL1L.TX_MODE and MDMCTRL1L.RX_MODE control bits. Buffered mode (mode 0) will be used for normal operation of *CC2430*, while other modes are available for test purposes.

14.17.1 Buffered transmit mode

In buffered transmit mode (TX_MODE 0), the 128 byte TXFIFO, located in *CC2430* RAM, is used to buffer data before transmission. A synchronization header is automatically inserted before the length field during transmission. The length field must always be the first byte written to the transmit buffer for all frames.

Writing one or multiple bytes to the TXFIFO is described in the FIFO access section on page 169. A DMA transfer can be configured to write transmit data to the TXFIFO.

Transmission is enabled by issuing a STXON or STXONCCA command strobe. See the Radio control state machine section on page 179 for an illustration of how the transmit command strobes affect the state of *CC2430*. The STXONCCA strobe is ignored if the channel is busy. See section 14.25 on page 182 for details on CCA.

The preamble sequence is started 12 symbol periods after the transmit command strobe. After the programmable start of frame delimiter has been transmitted, data is fetched from the TXFIFO.

The TXFIFO can only contain one data frame at a given time.

After complete transmission of a data frame, the TXFIFO is automatically refilled with the last transmitted frame. Issuing a new STXON or STXONCCA command strobe will then cause *CC2430* to retransmit the last frame.

Writing to the TXFIFO after a frame has been transmitted will cause the TXFIFO to be automatically flushed before the new byte is written. The only exception is if a TXFIFO underflow has occurred, when a SFLUSHTX command strobe is required.

14.17.2 Buffered receive mode

In buffered receive mode (RX_MODE 0), the 128 byte RXFIFO, located in *CC2430* RAM, is used to buffer data received by the demodulator. Accessing data in the RXFIFO is described in the FIFO access section on page 169.

The RF interrupt generated by RFSTATUS.FIFOP and also the RFSTATUS.FIFO and RFSTATUS.FIFOP register bits are used to assist the CPU in supervising the RXFIFO. Please note that these status bits are only related to the RXFIFO, even if ***CC2430*** is in transmit mode.

A DMA transfer should be used to read data from the RXFIFO. In this case a DMA channel can be setup to use the RADIO DMA trigger (see DMA triggers on page 92) to initiate a DMA transfer using the RFD register as the DMA source.

Multiple data frames may be in the RXFIFO simultaneously, as long as the total number of bytes does not exceed 128.

See the RXFIFO overflow section on page 170 for details on how a RXFIFO overflow is detected and signaled.

14.18 Address Recognition

CC2430 includes hardware support for address recognition, as specified in [1]. Hardware address recognition may be enabled or disabled using the MDMCTRL0H.ADDR_DECODE control bit. Address recognition uses the following RF registers

- IEEE_ADDR7-IEEE_ADDR0
- PANIDH:PANIDL
- SHORTADDRH:SHORTADDRL.

Address recognition is based on the following requirements, listed from section 7.5.6.2 in [1]:

- The frame type subfield shall not contain an illegal frame type
- If the frame type indicates that the frame is a beacon frame, the source PAN identifier shall match macPANId unless macPANId is equal to 0xFFFF, in which case the beacon frame shall be accepted regardless of the source PAN identifier.
- If a destination PAN identifier is included in the frame, it shall match macPANId or shall be the broadcast PAN identifier (0xFFFF).
- If a short destination address is included in the frame, it shall match either macShortAddress or the broadcast address (0xFFFF). Otherwise if an extended destination address is included in the frame, it shall match aExtendedAddress.
- If only source addressing fields are included in a data or MAC command frame, the frame shall only be accepted if the device is a PAN coordinator and the source PAN identifier matches macPANId.

If any of the above requirements are not satisfied and address recognition is enabled, ***CC2430*** will disregard the incoming frame and flush the data from the RXFIFO. Only data from the rejected frame is flushed, data from previously accepted frames may still be in the RXFIFO.

Incoming frames are first subject to frame type filtering according to the setting of the MDMCTRL0H.FRAMET_FILT register bit.

Following the required frame type filtering, incoming frames with reserved frame types (FCF frame type subfield is 4, 5, 6 or 7) are however accepted if the RESERVED_FRAME_MODE control bit in the RF register MDMCTRL0H is set. In this case, no further address recognition is performed on these frames. This option is included for future expansions of the IEEE 802.15.4 standard.

If a frame is rejected, ***CC2430*** will only start searching for a new frame after the rejected frame has been completely received (as defined by the length field) to avoid detecting false SFDs within the frame.

The MDMCTRL0.PAN_COORDINATOR control bit must be correctly set, since parts of the address recognition procedure requires knowledge about whether the current device is a PAN coordinator or not.

14.19 Acknowledge Frames

CC2430 includes hardware support for transmitting acknowledge frames, as specified in [1]. Figure 50 shows the format of the acknowledge frame.

If MDMCTRL0L.AUTOACK is enabled, an acknowledge frame is transmitted for all incoming frames accepted by the address recognition with the acknowledge request flag set and a valid CRC. AUTOACK therefore does not make sense unless also ADDR_DECODE and AUTOCRC are enabled. The sequence number is copied from the incoming frame.

Bytes:	4	1	1	2	1	2
	Preamble Sequence	Start of Frame Delimiter (SFD)	Frame Length	Frame Control Field (FCF)	Data Sequence Number	Frame Check Sequence (FCS)
	Synchronisation Header (SHR)		PHY Header (PHR)	MAC Header (MHR)		MAC Footer (MFR)

Figure 50: Acknowledge frame format [1]

Two command strobes, SACK and SACKPEND are defined to transmit acknowledge frames with the frame pending field cleared or set, respectively. The acknowledge frame is only transmitted if the CRC is valid.

For systems using beacons, there is an additional timing requirement that the acknowledge frame transmission may be started on the first backoff-slot boundary (20 symbol periods) at least 12 symbol periods after the last symbol of the incoming frame. When the RF register control bit MDMCTRL1H.SLOTTED_ACK is set to 1, the acknowledge frame is transmitted between 12 and 30 symbol periods after the incoming frame. The timing is defined such that there is an integer number of 20-symbol period backoff-slots between the incoming packet SFD and the transmitted acknowledge frame SFD. This timing is also illustrated in Figure 51.

Using SACKPEND will set the pending data flag for automatically transmitted acknowledge frames using AUTOACK. The pending flag will then be set also for future acknowledge frames, until a SACK command strobe is issued. The pending data flag that is transmitted will be logically OR'ed with the value of FSMTC1.PENDING_OR. Thus the pending flag can be set high using this register control bit.

Acknowledge frames may be manually transmitted using normal data transmission if desired.

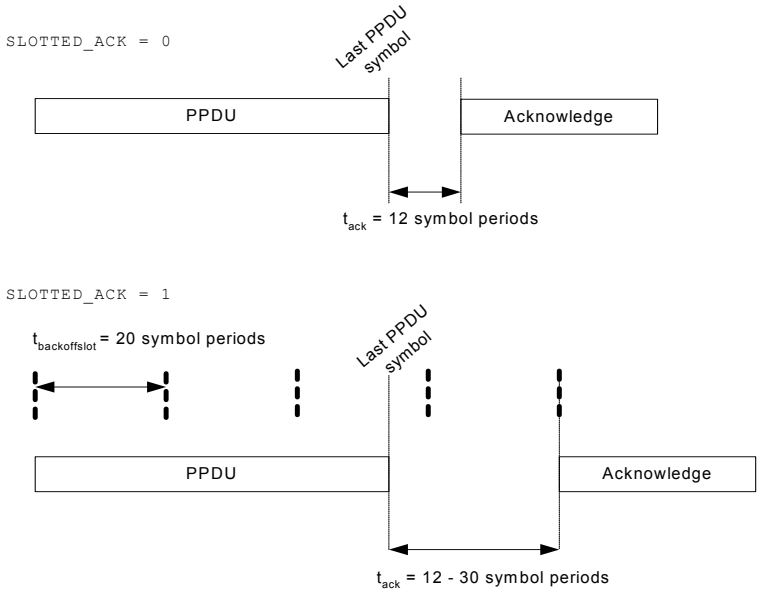


Figure 51: Acknowledge frame timing

14.20 Radio control state machine

CC2430 has a built-in state machine that is used to switch between different operation states (modes). The change of state is done either by using command strobes or by internal events such as SFD detected in receive mode.

The radio control state machine states are shown in Figure 52. The numbers in brackets refer to the state number readable in the FSMSTATE status register. Reading the FSMSTATE status register is primarily for test / debug purposes. The figure assumes that the device is already placed in the PM0 power mode.

Before using the radio in either RX or TX mode, the voltage regulator and crystal oscillator must be turned on and become stable. The voltage regulator and crystal oscillator startup times are given in the section 7.4 on page 16.

The voltage regulator for the radio is enabled by setting the RF register bit RFPWR.RREG_RADIO_PD high. The interrupt flag RFIF.IRQ_RREG_ON is set to 1 when the voltage regulator has powered-up.

The crystal oscillator is controlled through the Power Management Controller. The SLEEP.XOSC_STB bit indicates whether the oscillator is running and stable or not (see page 142). This SFR register can be polled when waiting for the oscillator to start.

For test purposes, the frequency synthesizer (FS) can also be manually calibrated and started by using the STXCALN or ISTXCALN command strobe (see section 0 and 0). This will not start a transmission before a STXON command strobe is issued. This is not shown in Figure 52.

Enabling transmission is done by issuing a STXON or STXONCCA command strobe.

Turning off RF can be accomplished by using the SRFOFF command strobe.

After bringing the *CC2430* up to Power Mode 0 (PM0) from a low-power mode e.g. Power Mode 3 (PM3), all RF registers will retain their values thus placing the chip ready to operate at the correct frequency and mode. Due to the very fast start-up time, *CC2430* can remain in a low-power mode until a transmission session is requested.

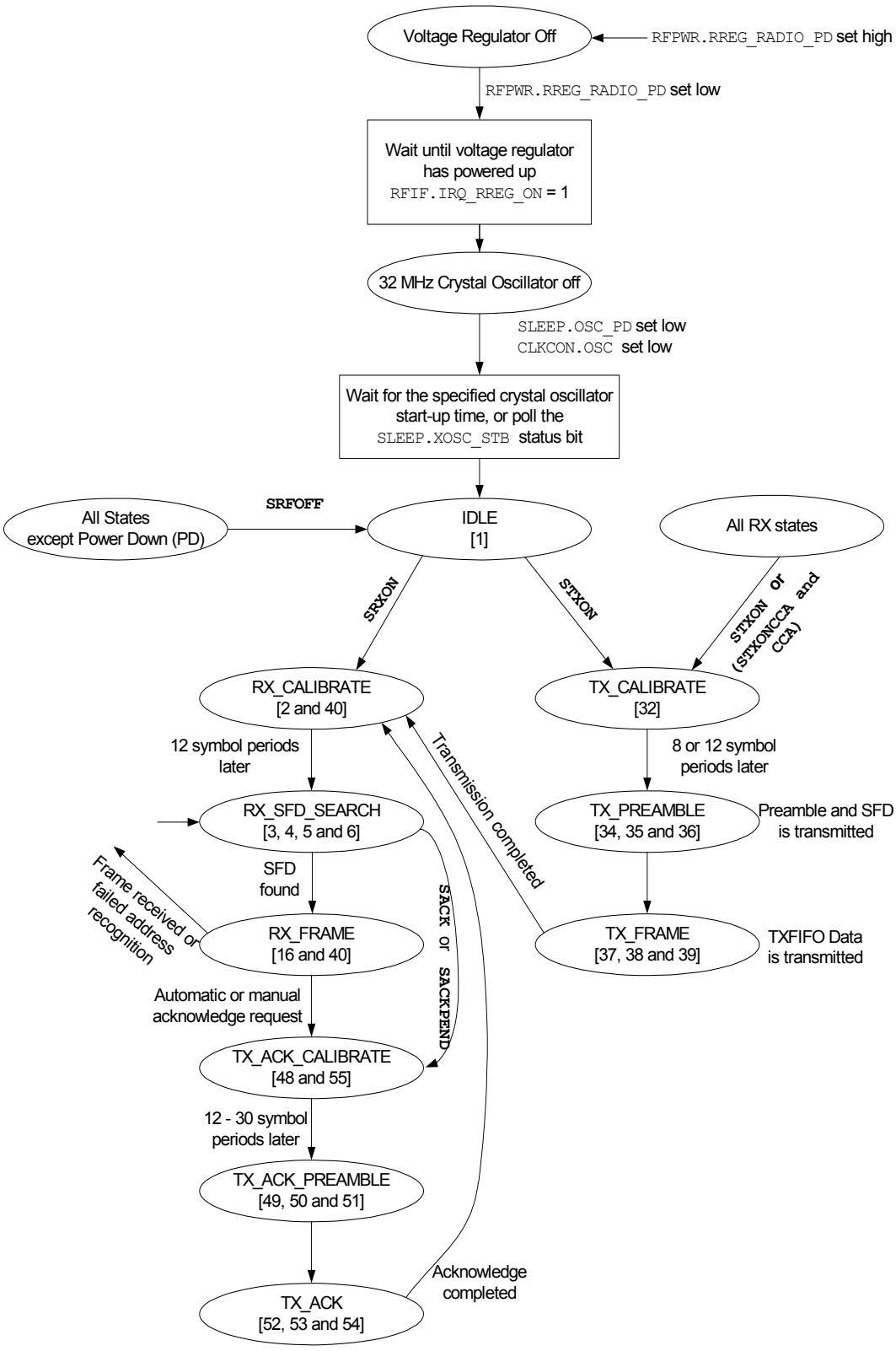


Figure 52: Radio control states

14.21 MAC Security Operations (Encryption and Authentication)

CC2430 features hardware IEEE 802.15.4 MAC security operations. Refer to section 13.9 on page 135 for a description of the AES encryption unit.

14.22 Linear IF and AGC Settings

CC2430 is based on a linear IF chain where the signal amplification is done in an analog VGA (variable gain amplifier). The gain of the VGA is digitally controlled.

The AGC (Automatic Gain Control) loop ensures that the ADC operates inside its dynamic range by using an analog/digital feedback loop.

The AGC characteristics are set through the AGCCTRLL:AGCCTRLH, registers. The reset values should be used for all AGC control registers.

14.23 RSSI / Energy Detection

CC2430 has a built-in RSSI (Received Signal Strength Indicator) giving a digital value that can be read form the 8 bit, signed 2's complement RSSIL.RSSI_VAL register bits.

The RSSI value is always averaged over 8 symbol periods (128 µs), in accordance with [1].

The RSSI register value RSSI.RSSI_VAL can be referred to the power P at the RF pins by using the following equations:

P = RSSI_VAL + RSSI_OFFSET [dBm]

where the RSSI_OFFSET is found empirically during system development from the front end gain. RSSI_OFFSET is approximately –45. E.g. if reading a value of –20 from the RSSI register, the RF input power is approximately –65 dBm.

A typical plot of the RSSI_VAL reading as function of input power is shown in Figure 53. It can be seen from the figure that the RSSI reading from CC2430 is very linear and has a dynamic range of about 100 dB.

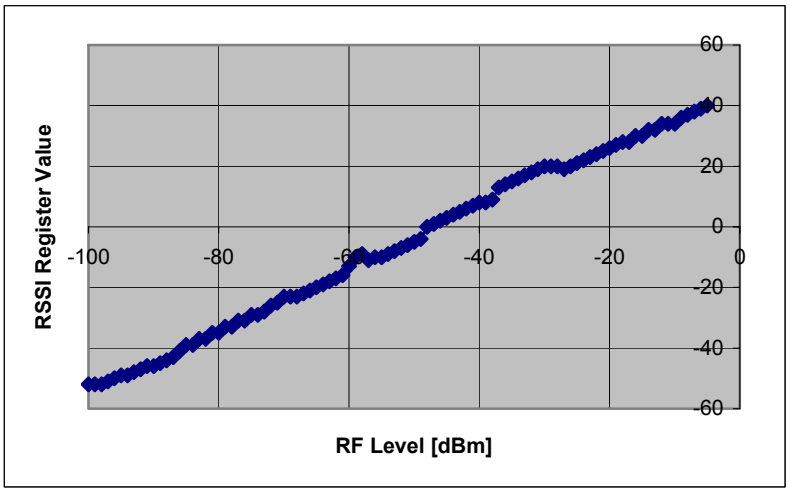


Figure 53: Typical RSSI value vs. input power

14.24 Link Quality Indication

The link quality indication (LQI) measurement is a characterization of the strength and/or quality of a received packet, as defined by [1].

The RSSI value described in the previous section may be used by the MAC software to produce the LQI value. The LQI value is required by [1] to be limited to the range 0 through 255, with at least eight unique values. Software is responsible for generating the appropriate scaling of the LQI value for the given application.

Using the RSSI value directly to calculate the LQI value has the disadvantage that e.g. a narrowband interferer inside the channel bandwidth will increase the LQI value although it actually reduces the true link quality. ***CC2430*** therefore also provides an average correlation value for each incoming packet, based on the eight first symbols following the SFD. This unsigned 7-bit value can be looked upon as a measurement of the "chip error rate," although ***CC2430*** does not do chip decision.

As described in the Frame check sequence section on page 175, the average correlation value for the eight first symbols is appended to each received frame together with the RSSI and CRC OK/not OK when MDMCTRL0L.AUTOCRC is set. A correlation value of approx. 110 indicates a maximum quality frame while a value of approx. 50 is typically the lowest quality frames detectable by ***CC2430***.

Software must convert the correlation value to the range 0-255 defined by [1], e.g. by calculating:

$$LQI = (CORR - a) \cdot b$$

limited to the range 0-255, where *a* and *b* are found empirically based on PER measurements as a function of the correlation value.

A combination of RSSI and correlation values may also be used to generate the LQI value.

14.25 Clear Channel Assessment

The clear channel assessment signal is based on the measured RSSI value and a programmable threshold. The clear channel assessment function is used to implement the CSMA-CA functionality specified in [1]. CCA is valid when the receiver has been enabled for at least 8 symbol periods.

Carrier sense threshold level is programmed by RSSI.CCA_THR. The threshold value can be programmed in steps of 1 dB. A CCA hysteresis can also be programmed in the MDMCTRL0H.CCA_HYST control bits.

All three CCA modes specified by [1] are implemented in ***CC2430***. These are set in MDMCTRL0L.CCA_MODE, as can be seen in the register description. The different modes are:

00 Reserved

01 Clear channel when received energy is below threshold.

10 Clear channel when not receiving valid IEEE 802.15.4 data.

11 Clear channel when energy is below threshold and not receiving valid IEEE 802.15.4 data

Clear channel assessment is available on the RFSTATUS.CCA RF register bit. RFSTATUS.CCA is active high. This register bit will also set the interrupt flag RFIF.IRQ_CCA.

Implementing CSMA-CA may easiest be done by using the STXONCCA command strobe given by the CSMA-CA/strobe processor, as shown in the Radio control state machine section on page 179. Transmission will then only start if the channel is clear. The TX_ACTIVE status bit in the RFSTATUS RF register may be used to detect the result of the CCA.

14.26 Frequency and Channel Programming

The operating frequency is set by programming the 10 bit frequency word located in FSCTRLH.FREQ[9:8] and FSCTRLL.FREQ[7:0]. The operating frequency F_C in MHz is given by:

F_C = 2048 + FREQ[9:0] MHz

where FREQ[9:0] is the value given by FSCTRLH.FREQ[9:8]:FSCTRLL.FREQ[7:0]

In receive mode the actual LO frequency is F_C – 2 MHz, since a 2 MHz IF is used. Direct conversion is used for transmission, so here the LO frequency equals F_C. The 2 MHz IF is automatically set by ***CC2430***, so the frequency programming is equal for RX and TX.

IEEE 802.15.4 specifies 16 channels within the 2.4 GHz band, numbered 11 through 26. The RF frequency of channel k is given by [1] :

F_C = 2405 + 5 (k-11) MHz, k=11, 12, ..., 26

For operation in channel k, the FSCTRLH.FREQ:FSCTRLL.FREQ register should therefore be set to:

FSCTRLH.FREQ:FSCTRLL.FREQ = 357 + 5 (k-11)

14.27 VCO and PLL Self-Calibration

14.27.1 VCO

The VCO is completely integrated and operates at 4800 – 4966 MHz. The VCO frequency is divided by 2 to generate frequencies in the desired band (2400-2483.5 MHz).

14.27.2 PLL self-calibration

The VCO's characteristics will vary with temperature, changes in supply voltages, and the desired operating frequency.

In order to ensure reliable operation the VCO's bias current and tuning range are automatically calibrated every time the RX mode or TX mode is enabled, i.e. in the RX_CALIBRATE, TX_CALIBRATE and TX_ACK_CALIBRATE control states in Figure 52 on page 180.

14.28 Output Power Programming

The RF output power of the device is programmable and is controlled by the TXCTRLL.PA_LEVEL RF register. Table 43 shows the output power for different settings, including the complete programming of the TXCTRLL control register and the current consumption in the radio itself.

PA_LEVEL	TXCTRLL register	Output Power [dBm]	Current consumption [mA]
31	0xFF	0	17.4
27	0xFB	-1	16.5
23	0xF7	-3	15.2
19	0xF3	-5	13.9
15	0xEF	-7	12.5
11	0xEB	-10	11.2
7	0xE7	-15	9.9
3	0xE3	-25	8.5

Table 43: Output power settings

14.29 Input / Output Matching

The RF input / output is differential (RF_N and RF_P). In addition there is supply switch output pin (TXRX_SWITCH) that must have an external DC path to RF_N and RF_P.

In RX mode the TXRX_SWITCH pin is at ground and will bias the LNA. In TX mode the TXRX_SWITCH pin is at supply rail voltage and will properly bias the internal PA.

The RF output and DC bias can be done using different topologies. Some are shown in Figure 6 on page 30.

Component values are given in Table 24 on page 31. If a differential antenna is implemented, no balun is required.

If a single ended output is required (for a single ended connector or a single ended antenna), a balun should be used for optimum performance.

14.30 Transmitter Test Modes

CC2430 can be set into different transmit test modes for performance evaluation. The test mode descriptions in the following sections requires that the chip is first reset, the crystal oscillator is selected using the CLKCON register and that the crystal oscillator has stabilized.

14.30.1 Unmodulated carrier

An unmodulated carrier may be transmitted by setting MDMCTRL1L.TX_MODE to 2, writing 0x1800 to the DACTSTH:DACTSTL registers and issue a STXON command strobe. The transmitter is then enabled while the transmitter I/Q DACs are overridden to static values. An un-modulated carrier will then be available on the RF output pins.

A plot of the single carrier output spectrum from CC2430 is shown in Figure 54 below.

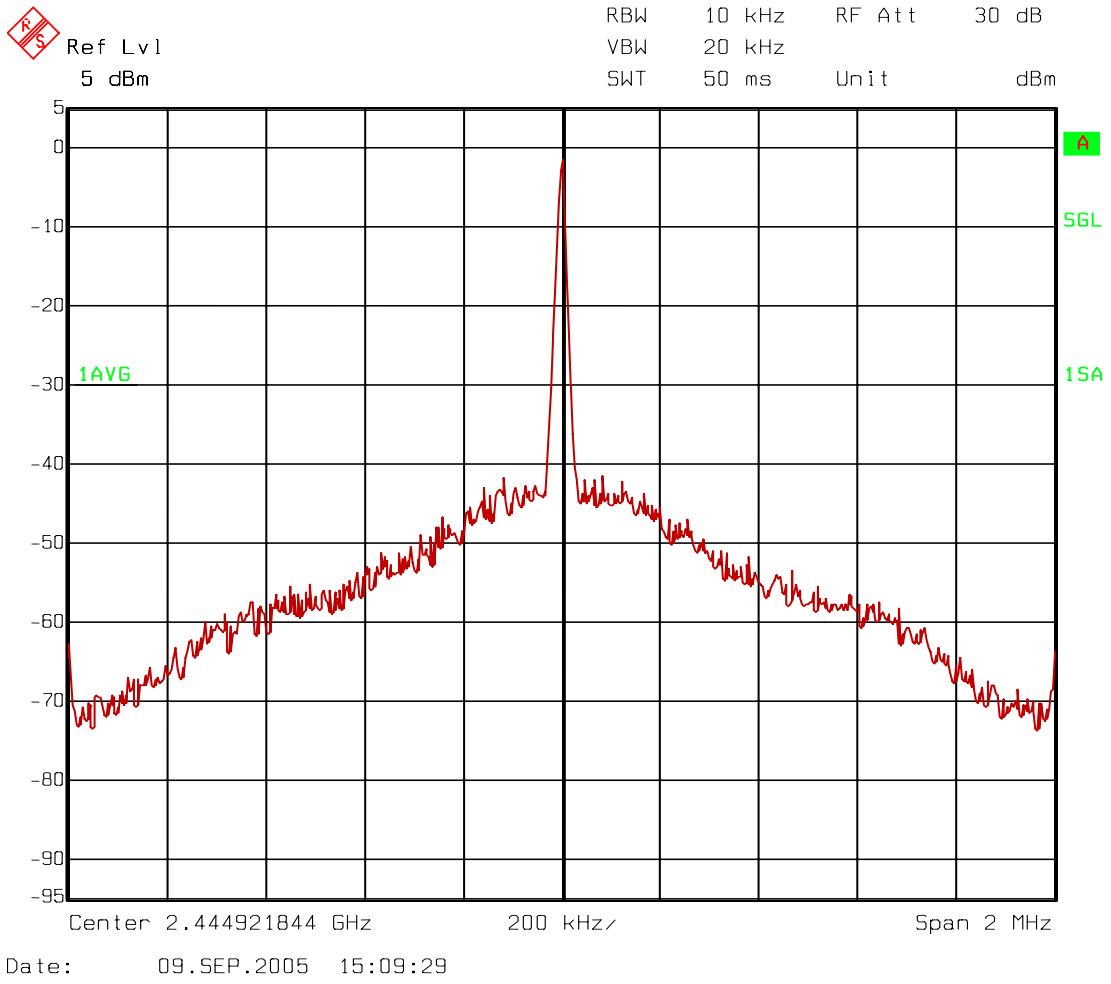


Figure 54: Single carrier output

14.30.2 Modulated spectrum

The ***CC2430*** has a built-in test pattern generator that can generate a pseudo random sequence using the CRC generator. This is enabled by setting MDMCTRL1L.TX_MODE to 3 and issuing a STXON command strobe. The modulated spectrum is then available on the RF pins. The low byte of the CRC word is transmitted and the CRC is updated with 0xFF for each new byte. The length of the transmitted data sequence is 65535 bits. The transmitted data-sequence is then:

[synchronization header] [0x00, 0x78, 0xb8, 0x4b, 0x99, 0xc3, 0xe9, …]

Since a synchronization header (preamble and SFD) is transmitted in all TX modes, this test mode may also be used to transmit a known pseudorandom bit sequence for bit error testing. Please note that ***CC2430*** requires symbol synchronization, not only bit synchronization, for correct reception. Packet error rate is therefore a better measurement for the true RF performance.

Another option to generate a modulated spectrum is to fill the TXFIFO with pseudo-random data and set MDMCTRL1L.TX_MODE to 2. ***CC2430*** will then transmit data from the FIFO disregarding a TXFIFO underflow. The length of the transmitted data sequence is then 1024 bits (128 bytes).

A plot of the modulated spectrum from ***CC2430*** is shown in Figure 55. Note that to find the output power from the modulated spectrum, the RBW must be set to 3 MHz or higher.

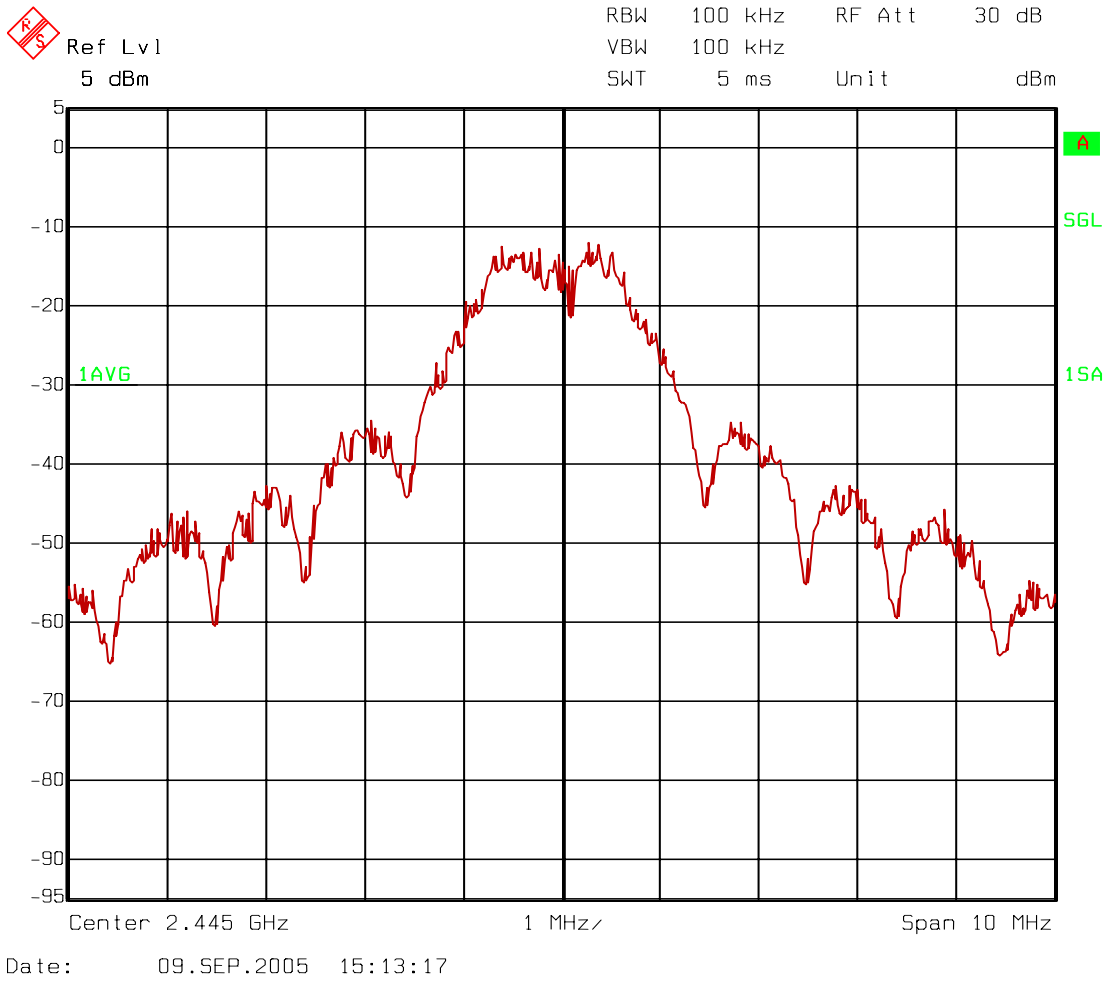


Figure 55: Modulated spectrum plot

14.31 System Considerations and Guidelines

14.31.1 SRD regulations

International regulations and national laws regulate the use of radio receivers and transmitters. SRDs (Short Range Devices) for license free operation are allowed to operate in the 2.4 GHz band worldwide. The most important regulations are ETSI EN 300 328 and EN 300 440 (Europe), FCC CFR-47 part 15.247 and 15.249 (USA), and ARIB STD-T66 (Japan).

14.31.2 Frequency hopping and multi-channel systems

The 2.4 GHz band is shared by many systems both in industrial, office and home environments. ***CC2430*** uses direct sequence spread spectrum (DSSS) as defined by [1] to spread the output power, thereby making the communication link more robust even in a noisy environment.

With ***CC2430*** it is also possible to combine both DSSS and FHSS (frequency hopping spread spectrum) in a proprietary non-IEEE 802.15.4 system. This is achieved by reprogramming the operating frequency (see the Frequency and Channel Programming section on page 183) before enabling RX or TX. A frequency synchronization scheme must then be implemented within the proprietary MAC layer to make the transmitter and receiver operate on the same RF channel.

14.31.3 Data burst transmissions

The data buffering in ***CC2430*** lets the user have a lower data rate link between the CPU and the radio module than the RF bit rate of 250 kbps. This allows the CPU to

buffer data at its own speed, reducing the workload and timing requirements. DMA transfers may be used to efficiently move data to and from the radio FIFOs.

The relatively high data rate of **CC2430** also reduces the average power consumption compared to the 868 / 915 MHz bands defined by [1], where only 20 / 40 kbps are available. **CC2430** may be powered up a smaller portion of the time, so that the average power consumption is reduced for a given amount of data to be transferred.

14.31.4 Crystal accuracy and drift

A crystal accuracy of ±40 ppm is required for compliance with IEEE 802.15.4 [1]. This accuracy must also take ageing and temperature drift into consideration.

A crystal with low temperature drift and low aging could be used without further compensation. A trimmer capacitor in the crystal oscillator circuit (in parallel with C191 in Figure 6) could be used to set the initial frequency accurately.

For non-IEEE 802.15.4 systems, the robust demodulator in **CC2430** allows up to 120 ppm total frequency offset between the transmitter and receiver. This could e.g. relax the accuracy requirement to 60 ppm for each of the devices.

Optionally in a star network topology, the FFD could be equipped with a more accurate crystal thereby relaxing the requirement on the RFD. This can make sense in systems where the RFDs ship in higher volumes than the FFDs.

14.31.5 Communication robustness

CC2430 provides very good adjacent, alternate and co channel rejection, image frequency suppression and blocking properties. The **CC2430** performance is significantly better than the requirements imposed by [1]. These are highly important parameters for reliable operation in the 2.4 GHz band, since an increasing number of devices/systems are using this license free frequency band.

14.31.6 Communication security

The hardware encryption and authentication operations in **CC2430** enable secure communication, which is required for many applications. Security operations require a lot of data processing, which is costly in an 8-bit microcontroller system. The hardware support within **CC2430** enables a high level of security with minimum CPU processing requirements.

14.31.7 Low cost systems

As the **CC2430** provides 250 kbps multi-channel performance without any external filters, a very low cost system can be made (e.g. two layer PCB with single-sided component mounting).

A differential antenna will eliminate the need for a balun, and the DC biasing can be achieved in the antenna topology.

14.31.8 Battery operated systems

In low power applications, the **CC2430** should be placed in the low-power modes PM2 or PM3 when not being active. Ultra low power consumption may be achieved since the voltage regulators are turned off.

14.31.9 BER / PER measurements

CC2430 includes test modes where data is received infinitely and output to pins. The required test modes are selected with the RF register bits `MDMCTRL1L.TX_MODE[1:0]` and `MDMCTRL1L.RX_MODE[1:0]`. These modes may be used for Bit Error Rate (BER) measurements. However, the following precautions must be taken to perform such a measurement:

- A preamble and SFD sequence must be used, even if pseudo random data is transmitted, since receiving the DSSS modulated signal requires *symbol* synchronization, not *bit* synchronization like e.g. in 2FSK systems. The `SYNCWORDH:SYNCWORDL` may be set to another value to fit to the measurement setup if necessary.
- The data transmitted over air must be spread according to [1] and the description on page 165. This means that the transmitter used during measurements must be able to do spreading of the bit

data to chip data. Remember that the *chip* sequence transmitted by the test setup is not the same as the *bit* sequence, which is output by ***CC2430***.

- When operating at or below the sensitivity limit, ***CC2430*** may lose symbol synchronization in infinite receive mode. A new SFD and restart of the receiver may be required to re-gain synchronization.

In an IEEE 802.15.4 system, all communication is based on packets. The sensitivity limit specified by [1] is based on Packet Error Rate (PER) measurements instead of BER. This is a more realistic measurement of the true RF performance since it mirrors the way the actual system operates.

Chipcon recommends performing PER measurements instead of BER measurements to evaluate the performance of IEEE 802.15.4 systems. To do PER measurements, the following may be used as a guideline:

- A valid preamble, SFD and length field must be used for each packet.
- The PSDU (see Figure 45 on page 173) length should be 20 bytes for sensitivity measurements as specified by [1].
- The sensitivity limit specified by [1] is the RF level resulting in a 1% PER. The packet sample space for a given measurement must then be >> 100 to have a sufficiently large sample space. E.g. at least 1000 packets should be used to measure the sensitivity.
- The data transmitted over air must be spread according to [1] and the description on page 165. Pre-generated packets may be used, although [1] requires that the PER is averaged over random PSDU data.
- The ***CC2430*** receive FIFO may be used to buffer data received during PER measurements, since it is able to buffer up to 128 bytes.
- The `MDMCTRL1H.CORR_THR` control register should be set to 20, as described in the Demodulator, Symbol Synchronizer and Data Decision section.

The simplest way of making a PER measurement will be to use another ***CC2430*** as the reference transmitter. However, this makes it difficult to measure the exact receiver performance.

Using a signal generator, this may either be set up as O-QPSK with half-sine shaping or as MSK. If using O-QPSK, the phases must be selected according to [1]. If using MSK, the chip sequence must be modified such that the modulated MSK signal has the same phase shifts as the O-QPSK sequence previously defined.

For a desired symbol sequence s_0, s_1, … , s_{n-1} of length n symbols, the desired chip sequence c_0, c_1, c_2, …, c_{32n-1} of length 32n is found using table lookup from Table 42 on page 165. It can be seen from comparing the phase shifts of the O-QPSK signal with the frequency of a MSK signal that the MSK chip sequence is generated as:

(c_0 xnor c_1), (c_1 xor c_2), (c_2 xnor c_3), … , (c_{32n-1} xor c_{32n}) where c_{32n} may be arbitrarily selected.

14.32 PCB Layout Recommendation

A two layer PCB is highly recommended.

In Chipcon's reference design, the top layer is used for signal routing, and the open areas are filled with metallization connected to ground using several vias. The area under the chip is used for grounding and must be well connected to the ground plane with several vias.

The ground pins should be connected to ground as close as possible to the package pin using individual vias. The de-coupling capacitors should also be placed as close as possible to the supply pins and connected to the ground plane by separate vias. Supply power filtering is very important.

The external components should be as small as possible (0402 is recommended) and surface mount devices must be used.

If using any external high-speed digital devices, caution should be used when placing these in order to avoid interference with the RF circuitry.

A Development Kit, CC2430DK, with a fully assembled Evaluation Module is available. It is strongly advised that this reference layout is followed very closely in order to obtain the best performance.

The schematic, BOM and layout Gerber files for the reference designs are all available from the Chipcon website.

14.33 Antenna Considerations

CC2430 can be used together with various types of antennas. A differential antenna like a dipole would be the easiest to interface not needing a balun (balanced to un-balanced transformation network).

The length of the λ/2-dipole antenna is given by:

$$L = 14250 / f$$

where f is in MHz, giving the length in cm. An antenna for 2450 MHz should be 5.8 cm. Each arm is therefore 2.9 cm.

Other commonly used antennas for short-range communication are monopole, helical and loop antennas. The single-ended monopole and helical would require a balun network between the differential output and the antenna.

Monopole antennas are resonant antennas with a length corresponding to one quarter of the electrical wavelength (λ/4). They are very easy to design and can be implemented simply as a “piece of wire” or even integrated into the PCB.

The length of the λ/4-monopole antenna is given by:

$$L = 7125 / f$$

where f is in MHz, giving the length in cm. An antenna for 2450 MHz should be 2.9 cm.

Non-resonant monopole antennas shorter than λ/4 can also be used, but at the expense of range. In size and cost critical applications such an antenna may very well be integrated into the PCB.

Enclosing the antenna in high dielectric constant material reduces the overall size of the antenna. Many vendors offer such antennas intended for PCB mounting.

Helical antennas can be thought of as a combination of a monopole and a loop antenna. They are a good compromise in size critical applications. Helical antennas tend to be more difficult to optimize than the simple monopole.

Loop antennas are easy to integrate into the PCB, but are less effective due to difficult impedance matching because of their very low radiation resistance.

For low power applications the differential antenna is recommended giving the best range and because of its simplicity.

The antenna should be connected as close as possible to the IC. If the antenna is located away from the RF pins the antenna should be matched to the feeding transmission line (50 Ω).

14.34 CSMA/CA Strobe Processor

The Command Strobe/CSMA-CA Processor (CSP) provides the control interface between the CPU and the Radio module in the *CC2430*.

The CSP interfaces with the CPU through the SFR register `RFST` and the RF registers `CSPX`, `CSPY`, `CSPZ`, `CSPT` and `CSPCTRL`. The CSP produces interrupt requests to the CPU. In addition the CSP interfaces with the MAC Timer by observing MAC Timer overflow events.

The CSP allows the CPU to issue command strobes to the radio thus controlling the operation of the radio.

The CSP has two modes of operation as follows, which are described below.

- Immediate Command Strobe execution.
- Program execution

Immediate Command Strobes are written as an Immediate Command Strobe instruction to the CSP which are issued instantly to the Radio module. The Immediate Command Strobe instruction is also used only to control the CSP. The Immediate Command Strobe instructions are described in section 1.1.1.

Program execution mode means that the CSP executes a sequence of instructions, from a program memory or instruction memory, thus constituting a short user-defined program. The available instructions are from a set of 14 instructions. The instruction set is defined in section 1.1.1. The required program is first loaded into the CSP by the CPU, then the CPU instructs the CSP to start executing the program.

The program execution mode together with the MAC Timer allows the CSP to automate CSMA-CA algorithms and thus act as a co-processor for the CPU.

The operation of the CSP is described in detail in the following sections. The command strobes and other instructions supported by the CSP are given in section 14.34.8 on page 194. Example programs for the CSP are shown in section 14.34.9 on page 201.

RFST (0xE1) – RF CSMA-CA / Strobe Processor

Bit	Name	Reset	R/W	Description
7:0	INSTR[7:0]	0xC0	R/W	Data written to this register will be written to the CSP instruction memory. Reading this register will return the CSP instruction currently being executed.

14.34.1 Instruction Memory

The CSP executes single byte program instructions which are read from a 24 byte instruction memory. The instruction memory is written to sequentially through the SFR register RFST. An instruction write pointer is maintained within the CSP to hold the location within the instruction memory where the next instruction written to RFST will be stored. Following a reset the write pointer is reset to location 0. During each RFST register write, the write pointer will be incremented by 1 until the end of memory is reached when the write pointer will stop incrementing. The first instruction written to RFST will be stored in location 0, the location where program execution starts. Thus a complete 24 instruction program is written to the instruction memory by writing each instruction in the desired order to the RFST register.

The write pointer may be reset to 0 by writing the immediate command strobe instruction ISSTOP. In addition the write pointer will be reset to 0 when the command strobe SSTOP is executed in a program.

Following a reset, the instruction memory is filled with SNOP (No Operation) instructions (opcode value 0xC0).

While the CSP is executing a program, there shall be no attempts to write instructions to the instruction memory by writing to RFST. Failure to observe this rule can lead to incorrect program execution and corrupt instruction memory contents. However, Immediate Command Strobe instructions may be written to RFST (see section 14.34.3).

14.34.2 Data Registers

The CSP has three data registers CSPT, CSPX, CSPY and CSPZ, which are read/write accessible for the CPU as RF registers. These registers are read or modified by some instructions, thus allowing the CPU to set parameters to be used by a CSP program or allowing the CPU to read CSP program status.

The CSPT data register is not modified by any instruction. The CSPT data register is used to set a MAC Timer overflow compare value. Once program execution has started on the CSP, the contents of this register is decremented by 1 each time the MAC timer overflows. When CSPT

reaches zero, program execution is halted and the interrupt IRQ_CSP_STOP is asserted. The CSPT register will not be decremented if the CPU writes 0xFF to this register.

Note: If the CSPT register compare function is not used, this register must be set to 0xFF before the program execution is started.

14.34.3 Program Execution

After the instruction memory has been filled, program execution is started by writing the immediate command strobe instruction ISSTART to the RFST register. The program execution will continue until either the instruction at last location has been executed, the CSPT data register contents is zero, a SSTOP instruction has been executed, an immediate ISSTOP instruction is written to RFST or until a SKIP instruction returns a location beyond the last location in the instruction memory.

Immediate Command Strobe instructions may be written to RFST while a program is being executed. In this case the Immediate instruction will bypass the instruction in the instruction memory, which will be completed once the Immediate instruction has been completed.

During program execution, reading RFST will return the current instruction being executed. An exception to this is the execution of immediate command strobes, during which RFST will return C0h.

14.34.4 Interrupt Requests

The CSP has three interrupts flags which can produce the RF interrupt vector. These are the following:

- IRQ_CSP_STOP: asserted when the processor has executed the last instruction in memory and when the processor stops due to a SSTOP or ISSTOP instruction or CSPT register equal zero.
- IRQ_CSP_WT: asserted when the processor continues executing the next instruction after a WAIT W or WAITX instruction.
- IRQ_CSP_INT: asserted when the processor executes an INT instruction.

14.34.5 Random Number Instruction

There will be a delay in the update of the random number used by the RANDXY instruction. Therefore if an instruction, RANDXY, that uses this value is issued immediately after a previous RANDXY instruction, the random value read may be the same in both cases.

14.34.6 Running CSP Programs

The basic flow for loading and running a program on the CSP is shown in Figure 56.

When program execution stops due to end of program the current program remains in program memory so that the same program can be run again by starting execution once again with the ISSTART command. However, when program execution is stopped by the SSTOP or ISTOP instruction, the program memory will be cleared.

.

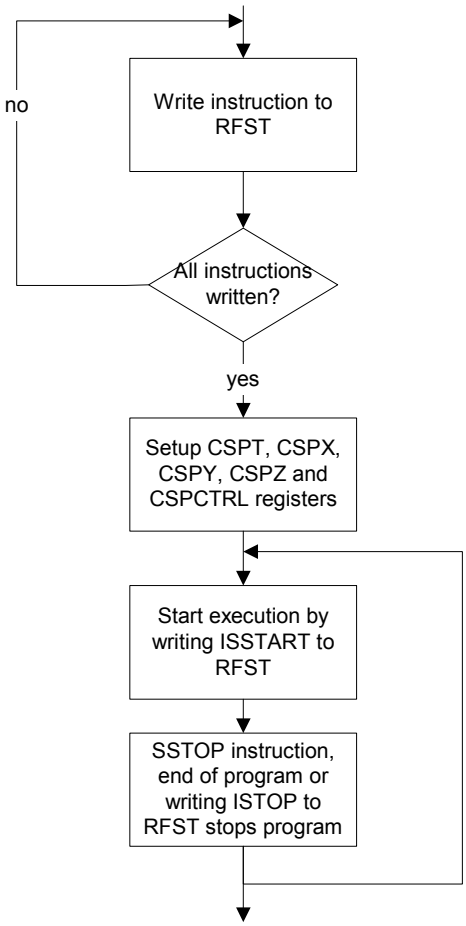


Figure 56: Running a CSP program

14.34.7 Instruction Set Summary

This section gives an overview of the instruction set. This is intended as a summary and definition of instruction opcodes. Refer to section 14.34.8 for a description of each instruction.

Each instruction consists of one byte which is written to the `RFST` register to be stored in the instruction memory.

The Immediate Strobe instructions (ISxxx) are not used in a program. When these instructions are written to the `RFST` register, they are executed immediately. If the CSP is already executing a program the current instruction will be delayed until the Immediate Strobe instruction has completed.

For undefined opcodes, the behavior of the CSP is defined as a No Operation Strobe Command (SNOP).

Table 44: Instruction Set Summary

Mnemonic	Opcode Bit number								Description
	7	6	5	4	3	2	1	0	
SKIP C,S	0	S			N	C			Skip S instructions when condition (C xor N) is true[5]
WAIT W	1	0	0	W					Wait for MAC Timer overflow count equal W
WEVENT	1	0	1	1	1	0	0	0	Wait until MAC Timer first compare
WAITX	1	0	1	1	1	0	1	1	Wait for MAC Timer overflow count equal CSPX
LABEL	1	0	1	1	1	0	1	0	Label next instruction as loop start
RPT	1	0	1	0	N	C			Repeat from start of loop if condition (C xor N) is true
INT	1	0	1	1	1	0	0	1	Assert interrupt
INCY	1	0	1	1	1	1	0	1	Increment CSPY
INCMAXY	1	0	1	1	0	M			Increment CSPY not greater than M
DECY	1	0	1	1	1	1	1	0	Decrement CSPY
DECZ	1	0	1	1	1	1	1	1	Decrement CSPZ
RANDXY	1	0	1	1	1	1	0	0	Load CSPX with CSPY bit random value.
Sxxx	1	1	0	STRB					Command strobe instructions[6]
ISxxx	1	1	1	STRB					Immediate strobe instructions[7]

[5] Refer to section 14.34.8 for full description of each instruction

[6] The Command strobe instruction is divided into eleven sub-instructions as defined by the STRB field. See sections 14.34.8.13 to 14.34.8.23 for a description.

[7] The Immediate strobe instruction is divided into eleven sub-instructions as defined by the STRB field. See sections 0 to 14.34.8.34 for a description.

14.34.8 Instruction Set Definition

There are 14 basic instruction types. Furthermore the Command Strobe and Immediate Strobe instructions can each be divided into eleven sub-instructions giving an effective number of 34 different instructions. The following sub-sections describe each instruction.

Note: the following definitions are used in this section

PC = CSP program counter

X = RF register CSPX

Y = RF register CSPY

Z = RF register CSPZ

T = RF register CSPT

14.34.8.1 DECZ

Function: Decrement Z

Description: The Z register is decremented by 1. An original value of 0x00 will underflow to 0x0FF.

Operation: Z := Z-1

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	1	1	1

14.34.8.2 DECY

Function: Decrement Y

Description: The Y register is decremented by 1. An original value of 0x00 will underflow to 0x0FF.

Operation: Y := Y-1

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	1	1	0

14.34.8.3 INCY

Function: Increment Y

Description: The Y register is incremented by 1. An original value of 0x0FF will overflow to 0x00.

Operation: Y := Y+1

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	1	0	1

14.34.8.4 INCMAXY

Function: Increment Y not greater than M.

Description: The Y register is incremented by 1 if the result is less than M otherwise Y register is loaded with value M.

Operation: Y:= min(Y+1, M)

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	0	M		

14.34.8.5 RANDXY

Function: Load random value into X

Description: The [Y] LSB bits of X register are loaded with random value. Note that if two RANDXY instructions are issued immediately after each other the same random value will be used in both cases.

Operation: X[Y-1:0]:= RNG_DOUT[Y-1:0], X[7:Y] := 0

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	1	0	0

14.34.8.6 INT

Function: Interrupt.

Description: The interrupt IRQ_CSP_INT is asserted when this instruction is executed.

Operation: IRQ_CSP_INT=1

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	0	0	1

14.34.8.7 WAITX

Function: Wait for X MAC Timer overflows

Description: Wait until MAC Timer overflows the number of times equal to register X. The contents of register X is decremented each time a MAC Timer overflow is detected. Program execution continues with the next instruction and the interrupt flag IRQ_CSP_WT is asserted when the wait condition is true.

Operation:

X := X-1 when MAC Timer overflow = true

PC := PC while number of MAC Timer overflow = true < X

PC := PC+1 when number of MAC Timer overflow = true = X

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	0	1	1

14.34.8.8 WAIT W

Function: Wait for W MAC Timer overflows

Description: Wait until MAC Timer overflows number of times equal to value W. If W=0 the instruction will wait for 32 overflows. Program execution continues with the next instruction and the interrupt flag IRQ_CSP_WT is asserted when the wait condition is true.

Operation:

PC := PC while number of MAC Timer overflow = true < W

PC := PC+1 when number of MAC Timer overflow = true = W

Opcode:

7	6	5	4	3	2	1	0
1	0	0	W				

14.34.8.9 WEVENT

Function: Wait until MAC Timer compare

Description: Wait until next MAC Timer compare. Program execution continues with the next instruction when the wait condition is true.

Operation:

PC := PC while MAC Timer compare = false

PC := PC+1 when MAC Timer compare = true

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	0	0	0

14.34.8.10 LABEL

Function: Set loop label

Description: Sets next instruction as start of loop. If the current instruction is the last instruction in the instruction memory then the current PC is set as start of loop. Only one level of loops is supported.

Operation: LABEL := PC+1

Opcode:

7	6	5	4	3	2	1	0
1	0	1	1	1	0	1	0

14.34.8.11 RPT C

Function: Conditional repeat

Description: If condition C is true then jump to instruction defined by last LABEL instruction, i.e. jump to start of loop. If the condition is false or if a LABEL instruction has not been executed, then execution will continue from next instruction. The condition C may be negated by setting N=1 and is described in the table below.

Condition code C	Description	Function
000	CCA is true	CCA=1
001	Receiving packet	SFD =1
010	CPU control true	CSPCTRL.CPU_CTRL=1
011	End of instruction memory	PC=23
100	Register X=0	X=0
101	Register Y=0	Y=0
110	Register Z=0	Z=0
111	Not used	-

Operation:

PC := LABEL when (C xor N)=true

PC := PC+1 when (C xor N) =false or LABEL=not set

Opcode:

7	6	5	4	3	2	1	0
1	0	1	0	N	C		

14.34.8.12 SKIP S, C

Function: Conditional skip instruction

Description: If condition C is true then skip S instructions. The condition C may be negated (N=1) and is described in the table below (note same conditions as RPT C instruction).

Condition code C	Description	Function
000	CCA is true	CCA=1
001	Receiving packet	SFD =1
010	CPU control true	CSPCTRL.CPU_CTRL =1
011	End of instruction memory	PC=23
100	Register X=0	X=0
101	Register Y=0	Y=0
110	Register Z=0	Z=0
111	Not used	-

Operation:

PC := PC+S+1 when (C xor N)=true

PC := PC+1 when (C xor N)=false

Opcode:

7	6	5	4	3	2	1	0
0	S			N	C		

14.34.8.13 STOP

Function: Stop program execution

Description: The SSTOP instruction stops the CSP program execution. The instruction memory is cleared, any loop start location set by the LABEL instruction is invalidated and the IRQ_CSP_STOP interrupt flag is asserted.

Operation: Stop execution, PC := 0, write pointer := 0

Opcode:

7	6	5	4	3	2	1	0
1	1	0	1	1	1	1	1

14.34.8.14 SNOP

Function: No Operation

Description: Operation continues at the next instruction.

Operation: PC := PC+1

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0

14.34.8.15 STXCALN

Function: Enable and calibrate frequency synthesizer for TX.

Note this instruction is for test purposes only (see section 14.20).

Description: The STXCALN instruction enables and calibrate frequency synthesizer for TX. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: STXCALN

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1

14.34.8.16 SRXON

Function: Enable and calibrate frequency synthesizer for RX

Description: The SRXON instruction asserts the output FFCTL_SRXON_STRB to enable and calibrate frequency synthesizer for RX. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SRXON

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	0	1	0

14.34.8.17 STXON

Function: Enable TX after calibration

Description: The STXON instruction enables TX after calibration. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: STXON

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	0	1	1

14.34.8.18 STXONCCA

Function: Enable calibration and TX if CCA indicates a clear channel

Description: The STXONCCA instruction enables TX after calibration if CCA indicates a clear channel. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: STXONCCA

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	0

14.34.8.19 SRFOFF

Function: Disable RX/TX and frequency synthesizer.

Description: The SRFOFF instruction asserts disables RX/TX and the frequency synthesizer. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SRFOFF

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	1

14.34.8.20 SFLUSHRX

Function: Flush RXFIFO buffer and reset demodulator

Description: The SFLUSHRX instruction flushes the RXFIFO buffer and resets the demodulator. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SFLUSHRX

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	1	1	0

14.34.8.21 SFLUSHTX

Function: Flush TXFIFO buffer

Description: The SFLUSHTX instruction flushes the TXFIFO buffer. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SFLUSHTX

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	0	1	1	1

14.34.8.22 SACK

Function: Send acknowledge frame with pending field cleared

Description: The SACK instruction sends an acknowledge frame. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SACK

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	1	0	0	0

14.34.8.23 SACKPEND

Function: Send acknowledge frame with pending field set

Description: The SACKPEND instruction sends an acknowledge frame with pending field set. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SACKPEND

Opcode:

7	6	5	4	3	2	1	0
1	1	0	0	1	0	0	1

14.34.8.24 ISSTOP

Function: Stop program execution

Description: The ISSTOP instruction stops the CSP program execution. The instruction memory is cleared, any loop start location set be the LABEL instruction is invalidated and the IRQ_CSP_STOP interrupt flag is asserted.

Operation: Stop execution

Opcode:

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

14.34.8.25 ISSTART

Function: Start program execution

Description: The ISSTART instruction starts the CSP program execution from first instruction written to instruction memory.

Operation: PC := 0, start execution

Opcode:

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	0

14.34.8.26 ISTXCALN

Function: Enable and calibrate frequency synthesizer for TX.

Note this instruction is for test purposes only (see section 14.20).

Description: The ISTXCALN instruction immediately enables and calibrates frequency synthesizer for TX. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: STXCALN

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	1

14.34.8.27 ISRXON

Function: Enable and calibrate frequency synthesizer for RX

Description: The ISRXON instruction immediately enables and calibrates frequency synthesizer for RX. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SRXON

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	0	0	1	0

14.34.8.28 ISTXON

Function: Enable TX after calibration

Description: The ISTXON instruction immediately enables TX after calibration. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: STXON_STRB

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	0	0	1	1

14.34.8.29 ISTXONCCA

Function: Enable calibration and TX if CCA indicates a clear channel

Description: The ISTXONCCA instruction immediately enables TX after calibration if CCA indicates a clear channel. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: STXONCCA

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	0

14.34.8.30 ISRFOFF

Function: Disable RX/TX and frequency synthesizer.

Description: The ISRFOFF instruction immediately disables RX/TX and frequency synthesizer. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation:

FFCTL_SRFOFF_STRB=1

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	1

14.34.8.31 ISFLUSHRX

Function: Flush RXFIFO buffer and reset demodulator

Description: The ISFLUSHRX instruction immediately flushes the RXFIFO buffer and resets the demodulator. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SFLUSHRX

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	0	1	1	0

14.34.8.32 ISFLUSHTX

Function: Flush TXFIFO buffer

Description: The ISFLUSHTX instruction immediately flushes the TXFIFO buffer. The instruction waits for the radio to acknowledge the command before executing the next instruction.

Operation: SFLUSHTX

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	0	1	1	1

14.34.8.33 ISACK

Function: Send acknowledge frame with pending field cleared

Description: The ISACK instruction immediately sends an acknowledge frame. The instruction waits for the radio to receive and interpret the command before executing the next instruction.

Operation: SACK_

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	1	0	0	0

14.34.8.34 ISACKPEND

Function: Send acknowledge frame with pending field set

Description: The ISACKPEND instruction immediately sends an acknowledge frame with pending field set. The instruction waits for the radio to receive and interpret the command before executing the next instruction.

Operation: SACKPEND

Opcode:

7	6	5	4	3	2	1	0
1	1	1	0	1	0	0	1

14.34.9 Example programs

This section shows two example programs for the CSP.

The first example in Figure 57 on page 202 shows how a slotted CSMA-CA algorithm as defined by IEEE 802.15.4 can be implemented.

The second example in Figure 58 shows how a non-slotted CSMA-CA algorithm can be implemented on the CSP.

```
0xba,    // LABEL
0xbb,    //     WAITX              Delay for random backoffs
0x22,    //     SKIP 2, C2         Turn on RX?
0xc2,    //         SRXON          Yes, RX on
0xb8,    //         WEVENT         Wait for RX to be stable
0x58,    //     SKIP 5, !C0        Is CCA = TRUE?
0xb8,    //         WEVENT         CCA = TRUE, CW = CW - 1
0x38,    //         SKIPC 3, !C0   Is CCA = TRUE?
0xc3,    //             STXON      Turn on TX
0xb9,    //             INT        Yes, signal success to CPU
0xdf,    //             SSTOP      CSMA completed successfully, stop processing
0x12,    //     SKIP 1, C2         Turn off RX to preserve power?
0xc5,    //         SRFOFF         Yes, RX off
0xb5,    //     INCMAXY 5          BE = min(BE+1, aMaxBE)
0xbc,    //     RANDXY             Next delay random unit backoff periods
0xbf,    //     DECZ               NB = NB - 1
0xae,    // RPT ! C6               Continue until NB = 0 (NB > macMaxCSMABackoffs
```

Figure 57: Example Slotted CSMA-CA algorithm implementation

```
0xba,  // LABEL
0xbb,  //     WAITX              Delay for random backoffs
0x22,  //     SKIP 2, C2         Turn on RX?
0xc2,  //         SRXON          Yes, RX on
0xb8,  //         WEVENT         Wait for RX to be stable
0x38,  //     SKIP 3, !C0        Is CCA = TRUE?
0xc3,  //         STXON          Turn on TX
0xb9,  //         INT            Yes, signal success to CPU
0xdf,  //         SSTOP          CSMA completed successfully, stop processing
0x12,  //     SKIP 1, C2         Turn off RX to preserve power?
0xc5,  //         SRFOFF         Yes, RX off
0xb5,  //     INCMAXY 5          BE = min(BE+1, aMaxBE)
0xbc,  //     RANDXY             Next delay random unit backoff periods
0xbf,  //     DECZ               NB = NB - 1
0xae,  // RPT ! C6               Continue until NB = 0 (NB > macMaxCSMABackoffs)
```

Figure 58: Example Non-slotted CSMA-CA algorithm implementation

14.35 Radio Registers

This section describes all RF registers used for control and status for the radio. The RF registers reside in XDATA memory space. Table 45 gives an overview of register addresses while the remaining tables in this section describe each register. Refer also to section 3 for Register conventions.

Table 45 : Overview of RF registers

XDATA Address	Register name	Description
0xDF00-0xDF01	-	Reserved
0xDF02	MDMCTRL0H	Modem Control 0, high
0xDF03	MDMCTRL0L	Modem Control 0, low
0xDF04	MDMCTRL1H	Modem Control 1, high
0xDF05	MDMCTRL1L	Modem Control 1, low
0xDF06	RSSIH	RSSI and CCA Status and Control, high
0xDF07	RSSIL	RSSI and CCA Status and Control, low
0xDF08	SYNCWORDH	Synchronisation Word Control, high
0xDF09	SYNCWORDL	Synchronisation Word Control, low
0xDF0A	TXCTRLH	Transmit Control, high
0xDF0B	TXCTRLL	Transmit Control, low
0xDF0C	RXCTRL0H	Receive Control 0, high
0xDF0D	RXCTRL0L	Receive Control 0, low
0xDF0E	RXCTRL1H	Receive Control 1, high
0xDF0F	RXCTRL1L	Receive Control 1, low
0xDF10	FSCTRLH	Frequency Synthesizer Control and Status, high
0xDF11	FSCTRLL	Frequency Synthesizer Control and Status, low
0xDF12	CSPX	CSP X Data
0xDF13	CSPY	CSP Y Data
0xDF14	CSPZ	CSP Z Data
0xDF15	CSPCTRL	CSP Control
0xDF16	CSPT	CSP T Data
0xDF17	RFPWR	RF Power Control
0xDF20	FSMTCH	Finite State Machine Time Constants, high
0xDF21	FSMTCL	Finite State Machine Time Constants, low
0xDF22	MANANDH	Manual AND Override, high
0xDF23	MANANDL	Manual AND Override, low
0xDF24	MANORH	Manual OR Override, high
0xDF25	MANORL	Manual OR Override, low
0xDF26	AGCCTRLH	AGC Control, high
0xDF27	AGCCTRLL	AGC Control, low

XDATA Address	Register name	Description
0xDF28-0xDF38	-	Reserved
0xDF39	FSMSTATE	Finite State Machine State Status
0xDF3A	ADCTSTH	ADC Test, high
0xDF3B	ADCTSTL	ADC Test, low
0xDF3C	DACTSTH	DAC Test, high
0xDF3D	DACTSTL	DAC Test, low
0xDF3E	-	Reserved
0xDF3F	-	Reserved
0xDF40	-	Reserved
0xDF41	-	Reserved
0xDF43	IEEE_ADDR0	IEEE Address 0
0xDF44	IEEE_ADDR1	IEEE Address 1
0xDF45	IEEE_ADDR2	IEEE Address 2
0xDF46	IEEE_ADDR3	IEEE Address 3
0xDF47	IEEE_ADDR4	IEEE Address 4
0xDF48	IEEE_ADDR5	IEEE Address 5
0xDF49	IEEE_ADDR6	IEEE Address 6
0xDF4A	IEEE_ADDR7	IEEE Address 7
0xDF4B	PANIDH	PAN Identifier, high
0xDF4C	PANIDL	PAN Identifier, low
0xDF4D	SHORTADDRH	Short Address, high
0xDF4E	SHORTADDRL	Short Address, low
0xDF4F	IOCFG0	I/O Configuration 0
0xDF50	IOCFG1	I/O Configuration 1
0xDF51	IOCFG2	I/O Configuration 2
0xDF52	IOCFG3	I/O Configuration 3
0xDF53	RXFIFOCNT	RX FIFO Count
0xDF54	FSMTC1	Finite State Machine Control
0xDF55-xDF5F	-	Reserved
0xDF60	CHVER	Chip Version
0xDF61	CHIPID	Chip Identification
0xDF62	RFSTATUS	RF Status

Bit	Name	Reset	R/W	Function
7:5	-	000	R/W	Reserved. Always set to 000
4	PAN_COORDINATOR	0	R/W	PAN Coordinator enable. Used for filtering packets with no destination address, as specified in section 7.5.6.2 in 802.15.4 [1] 0 : Device is not a PAN Coordinator 1 : Device is a PAN Coordinator
3	ADR_DECODE	1	R/W	Hardware Address decode enable. 0 : Address decoding is disabled 1 : Address decoding is enabled
2:0	CCA_HYST[2:0]	010	R/W	CCA Hysteresis in dB, values 0 through 7 dB

Table 46: Register MDMCTRL0H (0xDF02)

Bit	Name	Reset	R/W	Description
7:6	CCA_MODE[1:0]	11	R/W	Clear Channel Assessment mode select. 00 : Reserved 01 : CCA=1 when RSSI < CCA_THR-CCA_HYST CCA=0 when RSSI >= CCA_THR 10 : CCA=1 when not receiving a packet 11 : CCA=1 when RSSI < CCA_THR-CCA_HYST and not receiving a packet CCA=0 when RSSI >= CCA_THR or receiving a packet
5	AUTOCRC	1	R/W	In packet mode a CRC-16 (ITU-T) is calculated and is transmitted after the last data byte in TX. In RX CRC is calculated and checked for validity.
4	AUTOACK	0	R/W	If AUTOACK is set, all packets accepted by address recognition with the acknowledge request flag set and a valid CRC are ack'ed 12 symbol periods after being received.
3:0	PREAMBLE_LENGTH[3:0]	0010	R/W	The number of preamble bytes (2 zero-symbols) to be sent in TX mode prior to the SYNCWORD. The reset value of 0010 is compliant with IEEE 802.15.4, since the 4th zero byte is included in the SYNCWORD. 0000 : 1 leading zero bytes (not recommended) 0001 : 2 leading zero bytes (not recommended) 0010 : 3 leading zero bytes (IEEE 802.15.4 compliant) 0011 : 4 leading zero bytes … 1111 : 16 leading zero bytes

Table 47: Register MDMCTRL0L (0xDF03)

Bit	Name	Reset	R/W	Description
7	SLOTTED_ACK	0	R/W	SLOTTED_ACK defines the timing of automatically transmitted acknowledgment frames. 0 : The acknowledgment frame is transmitted 12 symbol periods after the incoming frame. 1 : The acknowledgment frame is transmitted between 12 and 30 symbol periods after the incoming frame. The timing is defined such that there is an integer number of 20-symbol periods between the received and the transmitted SFDs. This may be used to transmit slotted acknowledgment frames in a beacon enabled network.
6	-	0	R/W	Reserved
5	CORR_THR_SFD	1	R/W	CORR_THR_SFD defines the level at which the CORR_THR correlation threshold is used to filter out received frames. 0 : Same filtering as CC2420, should be combined with a CORR_THR of 0x14 1 : More extensive filtering is performed, which will result in less false frame detections e.g. caused by noise.
4:0	CORR_THR[4:0]	0x10	R/W	Demodulator correlator threshold value, required before SFD search.

Table 48: Register MDMCTRL1H (0xDF04)

Bit	Name	Reset	R/W	Description
7:6	-	00	R0	Reserved, read as 0.
5	DEMOD_AVG_MODE	0	R/W	DC average filter behavior. 0 : Lock DC level to be removed after preamble match 1 : Continuously update DC average level.
4	MODULATION_MODE	0	R/W	Set one of two RF modulation modes for RX / TX 0 : IEEE 802.15.4 compliant mode 1 : Reversed phase, non-IEEE compliant (could be used to set up a system which will not receive 802.15.4 packets)
3:2	TX_MODE[1:0]	00	R/W	Set test modes for TX 00 : Normal operation, transmit TXFIFO 01 : Serial mode, use transmit data on serial interface, infinite transmission. 10 : TXFIFO looping ignore underflow in TXFIFO and read cyclic, infinite transmission. 11 : Send random data from CRC, infinite transmission.
1:0	RX_MODE[1:0]	00	R/W	Set test mode of RX 00 : Normal operation, use RXFIFO 01 : Receive serial mode, output received data on pins. Infinite RX. 10 : RXFIFO looping ignore overflow in RXFIFO and write cyclic, infinite reception. 11 : Reserved

Table 49: Register MDMCTRL1L (0xDF05)

Bit	Name	Reset	R/W	Description
7:0	CCA_THR[7:0]	0xE0	R/W	Clear Channel Assessment threshold value, signed number in 2's complement for comparison with the RSSI. The unit is 1 dB, offset is TBD [depends on the absolute gain of the RX chain, including external components and should be measured]. The CCA signal goes high when the received signal is below this value. The reset value is in the range of -70 dBm.

Table 50: Register RSSIH (0xDF06)

Bit	Name	Reset	R/W	Description
7:0	RSSI_VAL[7:0]	0x00	R	RSSI estimate on a logarithmic scale, signed numbern 2's complement. Unit is 1 dB, offset is TBD [depends on the absolute gain of the RX chain, including external components, and should be measured]. The RSSI value is averaged over 8 symbol periods.

Table 51: Register RSSIL (0xDF07)

Bit	Name	Reset	R/W	Description
7:0	SYNCWORD[15:8]	0xA7	R/W	Synchronization word. The SYNCWORD is processed from the least significant nibble (F at reset) to the most significant nibble (A at reset). SYNCWORD is used both during modulation (where 0xF's are replaced with 0x0's) and during demodulation (where 0xF's are not required for frame synchronization). In reception an implicit zero is required before the first symbol required by SYNCWORD. The reset value is compliant with IEEE 802.15.4.

Table 52: Register SYNCWORDH (0xDF08)

Bit	Name	Reset	R/W	Description
7:0	SYNCWORD[7:0]	0x0F	R/W	Synchronization word. The SYNCWORD is processed from the least significant nibble (F at reset) to the most significant nibble (A at reset). SYNCWORD is used both during modulation (where 0xF's are replaced with 0x0's) and during demodulation (where 0xF's are not required for frame synchronization). In reception an implicit zero is required before the first symbol required by SYNCWORD. The reset value is compliant with IEEE 802.15.4.

Table 53: Register SYNCWORDL (0xDF09)

Bit	Name	Reset	R/W	Description
7:6	TXMIXBUF_CUR[1:0]	10	R/W	TX mixer buffer bias current. 00 : 690 uA 01 : 980 uA 10 : 1.16 mA (nominal) 11 : 1.44 mA
5	TX_TURNAROUND	1	R/W	Sets the wait time after STXON before transmission is started. 0 : 8 symbol periods (128 us) 1 : 12 symbol periods (192 us)
4:3	TXMIX_CAP_ARRAY[1:0]	0	R/W	Selects varactor array settings in the transmit mixers.
2:1	TXMIX_CURRENT[1:0]	0	R/W	Transmit mixers current: 00 : 1.72 mA 01 : 1.88 mA 10 : 2.05 mA 11 : 2.21 mA
0	PA_DIFF	1	R/W	Power Amplifier (PA) output select. Selects differential or single-ended PA output. 0 : Single-ended output 1 : Differential output

Table 54: Register TXCTRLH (0xDF0A)

Bit	Name	Reset	R/W	Description
7:5	PA_CURRENT[2:0]	011	R/W	Current programming of the PA 000 : -3 current adjustment 001 : -2 current adjustment 010 : -1 current adjustment 011 : Nominal setting 100 : +1 current adjustment 101 : +2 current adjustment 110 : +3 current adjustment 111 : +4 current adjustment
4:0	PA_LEVEL[4:0]	0x1F	R/W	Output PA level. (~0 dBm)

Table 55: Register TXCTRLL (0xDF0B)

Bit	Name	Reset	R/W	Description
7:6	-	00	R0	Reserved, read as 0.
5:4	RXMIXBUF_CUR[1:0]	01	R/W	RX mixer buffer bias current. 00 : 690 uA 01 : 980 uA (nominal) 10 : 1.16 mA 11 : 1.44 mA
3:2	HIGH_LNA_GAIN[1:0]	0	R/W	Controls current in the LNA gain compensation branch in AGC High gain mode. 00 : Compensation disabled 01 : 100 µA compensation current 10 : 300 µA compensation current (Nominal) 11 : 1000 µA compensation current
1:0	MED_LNA_GAIN[1:0]	10	R/W	Controls current in the LNA gain compensation branch in AGC Med gain mode.

Table 56: Register RXCTRL0H (0xDF0C)

Bit	Name	Reset	R/W	Description
7:6	LOW_LNA_GAIN[1:0]	11	R/W	Controls current in the LNA gain compensation branch in AGC Low gain mode
5:4	HIGH_LNA_CURRENT[1:0]	10	R/W	Controls main current in the LNA in AGC High gain mode 00 : 240 µA LNA current (x2) 01 : 480 µA LNA current (x2) 10 : 640 µA LNA current (x2) 11 : 1280 µA LNA current (x2)
3:2	MED_LNA_CURRENT[1:0]	01	R/W	Controls main current in the LNA in AGC Med gain mode
1:0	LOW_LNA_CURRENT[1:0]	01	R/W	Controls main current in the LNA in AGC Low gain mode

Table 57: Register RXCTRL0L (0xDF0D)

Bit	Name	Reset	R/W	Description
7:6	-	0	R0	Reserved, read as 0.
5	RXBPF_LOCUR	1	R/W	Controls reference bias current to RX band-pass filters: 0 : 4 uA 1 : 3 uA (Default)
4	RXBPF_MIDCUR	0	R/W	Controls reference bias current to RX band-pass filters: 0 : 4 uA (Default) 1 : 3.5 uA
3	LOW_LOWGAIN	1	R/W	LNA low gain mode setting in AGC low gain mode.
2	MED_LOWGAIN	0	R/W	LNA low gain mode setting in AGC medium gain mode.
1	HIGH_HGM	1	R/W	RX Mixers high gain mode setting in AGC high gain mode.
0	MED_HGM	0	R/W	RX Mixers high gain mode setting in AGC medium gain mode.

Table 58: Register RXCTRL1H (0xDF0E)

Bit	Name	Reset	R/W	Description
7:6	LNA_CAP_ARRAY[1:0]	01	R/W	Selects varactor array setting in the LNA 00 : OFF 01 : 0.1 pF (x2) (Nominal) 10 : 0.2 pF (x2) 11 : 0.3 pF (x2)
5:4	RXMIX_TAIL[1:0]	01	R/W	Control of the receiver mixers output current. 00 : 12 µA 01 : 16 µA (Nominal) 10 : 20 µA 11 : 24 µA
3:2	RXMIX_VCM[1:0]	01	R/W	Controls VCM level in the mixer feedback loop 00 : 8 µA mixer current 01 : 12 µA mixer current (Nominal) 10 : 16 µA mixer current 11 : 20 µA mixer current
1:0	RXMIX_CURRENT[1:0]	10	R/W	Controls current in the mixer 00 : 360 µA mixer current (x2) 01 : 720 µA mixer current (x2) 10 : 900 µA mixer current (x2) (Nominal) 11 : 1260 µA mixer current (x2)

Table 59: Register RXCTRL1L (0xDF0F)

Bit	Name	Reset	R/W	Description
7:6	LOCK_THR[1:0]	01	R/W	Number of consecutive reference clock periods with successful sync windows required to indicate lock: 00 : 64 01 : 128 10 : 256 11 : 512
5	CAL_DONE	0	R	Frequency synthesizer calibration done. 0 : Calibration not performed since the last time the FS was turned on. 1 : Calibration performed since the last time the FS was turned on.
4	CAL_RUNNING	0	R	Calibration status, '1' when calibration in progress.
3	LOCK_LENGTH	0	R/W	LOCK_WINDOW pulse width: 0: 2 CLK_PRE periods 1: 4 CLK_PRE periods
2	LOCK_STATUS	0	R	PLL lock status 0 : PLL is not in lock 1 : PLL is in lock
1:0	FREQ[9:8]	01 (2405 MHz)	R/W	Frequency control word. Used directly in TX, in RX the LO frequency is automatically set 2 MHz below the RF frequency. $Frequency\ division = \frac{2048 + FREQ[9:0]}{4} \Leftrightarrow$ $f_{RF} = (2048 + FREQ[9:0])\ \text{MHz}$ $f_{LO} = (2048 + FREQ[9:0] - 2 \cdot RXEN)\ \text{MHz}$

Table 60: Register FSCTRLH (0xDF10)

Bit	Name	Reset	R/W	Description
7:0	FREQ[7:0]	0x65 (2405 MHz)	R/W	Frequency control word. Used directly in TX, in RX the LO frequency is automatically set 2 MHz below the RF frequency. $Frequency\ division = \frac{2048 + FREQ[9:0]}{4} \Leftrightarrow$ $f_{RF} = (2048 + FREQ[9:0])\ \text{MHz}$ $f_{LO} = (2048 + FREQ[9:0] - 2 \cdot RXEN)\ \text{MHz}$

Table 61: Register FSCTRLL (0xDF11)

Bit	Name	Reset	R/W	Description
7:0	CSPT	0x00	R/W	CSP T Data register. Contents is decremented each time MAC Timer overflows while CSP program is running. CSP program stops when is about to count to 0. Setting T=0xFF disables decrement function.

Table 62: Register CSPT (0xDF16)

Bit	Name	Reset	R/W	Description
7:0	CSPX	0x00	R/W	CSP X Data register. Used by CSP WAITX, RANDXY and conditional instructions

Table 63: Register CSPX (0xDF12)

Bit	Name	Reset	R/W	Description
7:0	CSPY	0x00	R/W	CSP Y Data register. Used by CSP INCY, DECY, INCMAXY, RANDXY and conditional instructions

Table 64: Register CSPY (0xDF13)

Bit	Name	Reset	R/W	Description
7:0	CSPZ	0x00	R/W	CSP Z Data register. Used by CSP DECZ and conditional instructions

Table 65: Register CSPZ (0xDF14)

Bit	Name	Reset	R/W	Description
7:1	-	0x00	R0	Reserved, read as 0
0	CPU_CTRL	0	R/W	CSP CPU control input. Used by CSP conditional instructions.

Table 66: Register CSPCTRL (0xDF15)

Bit	Name	Reset	R/W	Description
7:5	-	0	R0	Reserved, read as 0.
4	ADI_RADIO_PD	0	R	ADI_RADIO_PD is a delayed version of RREG_RADIO_PD. The delay is set by RREG_DELAY[2:0]. When ADI_RADIO_PD is 0, all analog modules in the radio are set in power down. ADI_RADIO_PD is read only.
3	RREG_RADIO_PD	1	R/W	Power down of the voltage regulator to the analog part of the radio. This signal is used to enable or disable the analog radio. 0 : Power up 1 : Power down
2:0	RREG_DELAY[2:0]	100	R/W	Delay value used in power-on for voltage regulator VREG_DELAY[2:0] Delay Units 000 0 µs 001 31 µs 010 63 µs 011 125 µs 100 250 µs 101 500 µs 110 1000 µs 111 2000 µs

Table 67: Register RFPWR (0xDF17)

Bit	Name	Reset	R/W	Description
7:5	TC_RXCHAIN2RX[2:0]	011	R/W	The time in 5 us steps between the time the RX chain is enabled and the demodulator and AGC is enabled. The RX chain is started when the band pass filter has been calibrated (after 6.5 symbol periods).
4:2	TC_SWITCH2TX[2:0]	110	R/W	The time in advance the RXTX switch is set high, before enabling TX. Unit is µs.
1:0	TC_PAON2TX[3:2]	10	R/W	The time in advance the PA is powered up before enabling TX. Unit is µs.

Table 68: Register FSMTCH (0xDF20)

Bit	Name	Reset	R/W	Description
7:6	TC_PAON2TX[1:0]	10	R/W	The time in advance the PA is powered up before enabling TX. Unit is µs.
5:3	TC_TXEND2SWITCH[2:0]	010	R/W	The time after the last chip in the packet is sent, and the rxtx switch is disabled. Unit is µs.
2:0	TC_TXEND2PAOFF[2:0]	100	R/W	The time after the last chip in the packet is sent, and the PA is set in power-down. Also the time at which the modulator is disabled. Unit is µs.

Table 69: Register FSMTCL (0xDF21)

Bit	Name	Reset	R/W	Description
7	VGA_RESET_N	1	R/W	The VGA_RESET_N signal is used to reset the peak detectors in the VGA in the RX chain.
6	BIAS_PD	1	R/W	Reserved, read as 0
5	BALUN_CTRL	1	R/W	The BALUN_CTRL signal controls whether the PA should receive its required external biasing (1) or not (0) by controlling the RX/TX output switch.
4	RXTX	1	R/W	RXTX signal: controls whether the LO buffers (0) or PA buffers (1) should be used.
3	PRE_PD	1	R/W	Powerdown of prescaler.
2	PA_N_PD	1	R/W	Powerdown of PA (negative path).
1	PA_P_PD	1	R/W	Powerdown of PA (positive path). When PA_N_PD=1 and PA_P_PD=1 the up conversion mixers are in powerdown.
0	DAC_LPF_PD	1	R/W	Powerdown of TX DACs.

Table 70: Register MANANDH (0xDF22)

Bit	Name	Reset	R/W	Description
7	-	0	R0	Reserved, read as 0
6	RXBPF_CAL_PD	1	R/W	Powerdown control of complex band pass receive filter calibration oscillator.
5	CHP_PD	1	R/W	Powerdown control of charge pump.
4	FS_PD	1	R/W	Powerdown control of VCO, I/Q generator, LO buffers.
3	ADC_PD	1	R/W	Powerdown control of the ADCs.
2	VGA_PD	1	R/W	Powerdown control of the VGA.
1	RXBPF_PD	1	R/W	Powerdown control of complex band pass receive filter.
0	LNAMIX_PD	1	R/W	Powerdown control of LNA, down conversion mixers and front-end bias.

Table 71: Register MANANDL (0xDF23)

Bit	Name	Reset	R/W	Description
7	VGA_RESET_N	0	R/W	The VGA_RESET_N signal is used to reset the peak detectors in the VGA in the RX chain.
6	BIAS_PD	0	R/W	Global Bias power down (1)
5	BALUN_CTRL	0	R/W	The BALUN_CTRL signal controls whether the PA should receive its required external biasing (1) or not (0) by controlling the RX/TX output switch.
4	RXTX	0	R/W	RXTX signal: controls whether the LO buffers (0) or PA buffers (1) should be used.
3	PRE_PD	0	R/W	Powerdown of prescaler.
2	PA_N_PD	0	R/W	Powerdown of PA (negative path).
1	PA_P_PD	0	R/W	Powerdown of PA (positive path). When PA_N_PD=1 and PA_P_PD=1 the up conversion mixers are in powerdown.
0	DAC_LPF_PD	0	R/W	Powerdown of TX DACs.

Table 72: Register MANORH (0xDF24)

Bit	Name	Reset	R/W	Description
7	-	0	R0	Reserved, read as 0
6	RXBPF_CAL_PD	0	R/W	Powerdown control of complex band pass receive filter calibration oscillator.
5	CHP_PD	0	R/W	Powerdown control of charge pump.
4	FS_PD	0	R/W	Powerdown control of VCO, I/Q generator, LO buffers.
3	ADC_PD	0	R/W	Powerdown control of the ADCs.
2	VGA_PD	0	R/W	Powerdown control of the VGA.
1	RXBPF_PD	0	R/W	Powerdown control of complex band pass receive filter.
0	LNAMIX_PD	0	R/W	Powerdown control of LNA, down conversion mixers and front-end bias.

Table 73: Register MANORL (0xDF25)

Bit	Name	Reset	R/W	Description
7	VGA_GAIN_OE	0	R/W	Use the VGA_GAIN value during RX instead of the AGC value.
6:0	VGA_GAIN[6:0]	0x7F	R/W	When written, VGA manual gain override value; when read, the currently used VGA gain setting.

Table 74: Register AGCCTRLH (0xDF26)

Bit	Name	Reset	R/W	Description
7:4	-	0	R0	Reserved, read as 0.
3:2	LNAMIX_GAINMODE_O [1:0]	00	R/W	LNA / Mixer Gain mode override setting 00 : Gain mode is set by AGC algorithm 01 : Gain mode is always low-gain 10 : Gain mode is always med-gain 11 : Gain mode is always high-gain
1:0	LNAMIX_GAINMODE[1:0]	00	R	Status bit, defining the currently selected gain mode selected by the AGC or overridden by the LNAMIX_GAINMODE_O setting.

Table 75: Register AGCCTRLL (0xDF27)

Bit	Name	Reset	R/W	Description
7:6	-	0	R0	Reserved, read as 0.
5:0	FSM_FFCTRL_STATE[5:0]	-	R	Gives the current state of the FIFO and Frame Control (FFCTRL) finite state machine.

Table 76: Register FSMSTATE (0xDF39)

Bit	Name	Reset	R/W	Function
7	ADC_CLOCK_DISABLE	0	R/W	ADC Clock Disable 0 : Clock enabled when ADC enabled 1 : Clock disabled, even if ADC is enabled
6:0	ADC_I[6:0]	-	R	Returns the current ADC I-branch value.

Table 77: Register ADCTSTH (0xDF3A)

Bit	Name	Reset	R/W	Function
7:6	-	0	R0	Reserved, read as 0.
6:0	ADC_Q[6:0]	-	R	Returns the current ADC Q-branch value.

Table 78: Register ADCTSTL (0xDF3B)

Bit	Name	Reset	R/W	Description
7	-	0	R0	Reserved, read as 0.
6:4	DAC_SRC[2:0]	000	R/W	The TX DACs data source is selected by DAC_SRC according to: 000 : Normal operation (from modulator). 001 : The DAC_I_O and DAC_Q_O override values below.- 010 : From ADC, most significant bits 011 : I/Q after digital down mix and channel filtering. 100 : Full-spectrum White Noise (from CRC) 101 : From ADC, least significant bits 110 : RSSI / Cordic Magnitude Output 111 : HSSD module. This feature will often require the DACs to be manually turned on in MANOVR and PAMTST.ATESTMOD_MODE=4.
3:0	DAC_I_O[5:2]	000	R/W	I-branch DAC override value.

Table 79: Register DACTSTH (0xDF3C)

Bit	Name	Reset	R/W	Description
7:6	DAC_I_O[1:0]	00	R/W	I-branch DAC override value.
5:0	DAC_Q_O[5:0]	0x00	R/W	Q-branch DAC override value.

Table 80: Register DACTSTL (0xDF3D)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR0[7:0]	0x00	R/W	IEEE ADDR byte 0

Table 81: Register IEEE_ADDR0 (0xDF43)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR1[7:0]	0x00	R/W	IEEE ADDR byte 1

Table 82: Register IEEE_ADDR1 (0xDF44)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR2[7:0]	0x00	R/W	IEEE ADDR byte 2

Table 83: Register IEEE_ADDR2 (0xDF45)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR3[7:0]	0x00	R/W	IEEE ADDR byte 3

Table 84: Register IEEE_ADDR3 (0xDF46)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR4[7:0]	0x00	R/W	IEEE ADDR byte 4

Table 85: Register IEEE_ADDR4 (0xDF47)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR5[7:0]	0x00	R/W	IEEE ADDR byte 5

Table 86: Register IEEE_ADDR5 (0xDF48)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR6[7:0]	0x00	R/W	IEEE ADDR byte 6

Table 87: Register IEEE_ADDR6 (0xDF49)

Bit	Name	Reset	R/W	Description
7:0	IEEE_ADDR7[7:0]	0x00	R/W	IEEE ADDR byte 7

Table 88: Register IEEE_ADDR7 (0xDF4A)

Bit	Name	Reset	R/W	Description
7:0	PANIDH[7:0]	0x00	R/W	PAN identifier high byte

Table 89: Register PANIDH (0xDF4B)

Bit	Name	Reset	R/W	Description
7:0	PANIDL[7:0]	0x00	R/W	PAN identifier low byte

Table 90: Register PANIDL (0xDF4C)

Bit	Name	Reset	R/W	Description
7:0	SHORTADDRH[7:0]	0x00	R/W	Short address high byte

Table 91: Register SHORTADDRH (0xDF4D)

Bit	Name	Reset	R/W	Description
7:0	SHORTADDRL[7:0]	0x00	R/W	Short address low byte

Table 92: Register SHORTADDRL (0xDF4E)

Bit	Name	Reset	R/W	Description
7	-	0	R0	Reserved, read as 0.
6:0	FIFOP_THR[6:0]	0x40	R/W	Sets the number of bytes in RXFIFO that is required for FIFOP to go high.

Table 93: Register IOCFG0 (0xDF4F)

Bit	Name	Reset	R/W	Description
7	-	0	R0	Reserved, read as 0.
6	OE_CCA	0	R/W	CCA is output on P1.7 when this bit is 1
5	IO_CCA_POL	0	R/W	Polarity of the IO_CCA signal. This bit is xor'ed with the internal CCA signal.
4:0	IO_CCA_SEL	000000	R/W	Multiplexer setting for the CCA signal. Must be 0x00 in order to output the CCA status.

Table 94: Register IOCFG1 (0xDF50)

Bit	Name	Reset	R/W	Description
7	-	0	R0	Reserved, read as 0.
6	OE_SFD	0	R/W	SFD is output on P1.6 when this bit is 1
5	IO_SFD_POL	0	R/W	Polarity of the IO_SFD signal. This bit is xor'ed with the internal SFD signal.
4:0	IO_SFD_SEL	000000	R/W	Multiplexer setting for the SFD signal. Must be 0x00 in order to output the SFD status

Table 95: Register IOCFG2 (0xDF51)

Bit	Name	Reset	R/W	Description
7:6	-	00	R0	Reserved, read as 0.
5:4	HSSD_SRC	00	R/W	Configures the HSSD interface. Only the first 4 settings (compared to CC2420) are used. 00 : Off 01 : Output AGC status (gain setting/peak detector status/accumulator value) 10 : Output ADC I and Q values 11 : Output I/Q after digital down mix and channel filtering
3	OE_FIFOP	0	R/W	FIFOP is output on P1.5 when this bit is 1.
2	IO_FIFOP_POL	0	R/W	Polarity of the IO_FIFOP signal. This bit is xor'ed with the internal FIFOP signal
1	OE_FIFO	0	R/W	FIFO is output on P1.4 when this bit is 1
0	IO_FIFO_POL	0	R/W	Polarity of the IO_FIFO signal. This bit is xor'ed with the internal FIFO signal

Table 96: Register IOCFG3 (0xDF52)

Bit	Name	Reset	R/W	Description
7:0	RXFIFOCNT[7:0]	0x00	R	Number of bytes in the RX FIFO

Table 97: Register RXFIFOCNT (0xDF53)

Bit	Name	Reset	R/W	Description
7:6	-	00	R0	Reserved, read as 0.
5	ABORTRX_ON_SRXON	1	R/W	Abort RX when SRXON strobe is issued 0 : Packet reception is not aborted when SRXON is issued 1 : Packet reception is aborted when SRXON is issued
4	RX_INTERRUPTED	0	R	RX interrupted by strobe command This bit is cleared when the next strobe is detected. 0 : Strobe command detected 1 : Packet reception was interrupted by strobe command
3	AUTO_TX2RX_OFF	0	R/W	Automatically go to RX after TX. Applies to both data packets and ACK packets. 0 : Automatic RX after TX 1 : No automatic RX after TX
2	RX2RX_TIME_OFF	0	R/W	Turns off the 12 symbol timeout after packet reception has ended. Active high.
1	PENDING_OR	0	R/W	This bit is OR'ed with the pending bit from FFCTRL before it goes to the modulator.
0	ACCEPT_ACKPKT.	1	R/W	Accept ACK packet control. 0 : Reject all ACK packets 1 : ACK packets are received

Table 98: Register FSMTC1 (0xDF54)

Bit	Name	Reset	R/W	Description
7:0	VERSION[7:0]	0x01	R	Chip revision number

Table 99: Register CHVER (0xDF60)

Bit	Name	Reset	R/W	Description
7:0	CHIPID[7:0]	0x85	R	Chip identification number. Always read as 0x85.

Table 100: Register CHIPID (0xDF61)

Bit	Name	Reset	R/W	Description
7:5	-	000	R0	Reserved, read as 0.
4	TX_ACTIVE	0	R	TX active indicates transmission in progress 0 : TX inactive 1 : TX active
3	FIFO	0	R	RXFIFO data available 0 : No data available in RXFIFO 1 : One or more bytes available in RXFIFO
2	FIFOP	0	R	RXFIFO threshold flag 0 : Number of bytes in RXFIFO is less or equal threshold set by IOCFG0.FIFOP_THR 1 : Number of bytes in RXFIFO is above threshold set by IOCFG0.FIFOP_THR
1	SFD	0	R	Start of Frame Delimiter status 0 : SFD inactive 1 : SFD active
0	CCA		R	Clear Channel Assessment

Table 101: Register RFSTATUS (0xDF62)

15 Radio Test Output Signals

For debug purposes, the RFSTATUS.SFD, RFSTATUS.FIFO, RFSTATUS.FIFOP and RFSTATUS.CCA bits can be output onto P1.7 – P1.4 I/O pins to monitor the status of these signals. These test output signals are selected by the IOCFG0, IOCFG1 and IOCFG2 registers.

The debug signals are output to the following I/O pins:

- P1.4 – FIFO
- P1.5 – FIFOP
- P1.6 – SFD
- P1.7 - CCA

16 Voltage Regulators

The ***CC2430*** includes two low drop-out voltage regulators. These are used to provide a 1.8 V power supply to the ***CC2430*** analog and digital power supplies.

Note: The voltage regulators should not be used to provide power to external circuits because of limited power sourcing capability and also due to noise considerations.

The analog voltage regulator input pin AVDD_RREG is to be connected to the unregulated 2.0 to 3.6 V power supply. The regulated 1.8 V voltage output to the analog parts, is available on the RREG_OUT pin. The digital regulator input pin AVDD_DREG is also to be connected to the unregulated 2.0 to 3.6 V power supply. The output of the digital regulator is connected internally within the ***CC2430*** to the digital power supply.

The voltage regulators require external components as described in section 11 on page 29.

16.1 Voltage Regulators Power-on

The analog voltage regulator is disabled by setting the RF register bit RFPWR.RREG_RADIO_PD to 1. When the analog voltage regulator is powered-on by clearing the RFPWR.RREG_RADIO_PD bit, there will be a delay before the regulator is enabled. This delay is programmable through the RFPWR RF register. The interrupt flag RFIF.IRQ_RREG_PD is set when the delay has expired. The delayed power-on can also be observed by polling the RF register bit RFPWR.ADI_RADIO_PD.

The digital voltage regulator is disabled when the ***CC2430*** is placed in power modes PM2 or PM3 (see section 13.10). When the voltage regulators are disabled, register and RAM contents will be retained while the unregulated 2.0 to 3.6 power supply is present.

17 Evaluation Software

Chipcon provides users of ***CC2430*** with a software program, SmartRF® Studio, which may be used for radio performance and functionality evaluation. SmartRF® Studio runs on Microsoft Windows 95/98 and Microsoft Windows NT/XP/2000. SmartRF® Studio can be downloaded from Chipcon's web page: http://www.chipcon.com

18 Register overview

DPH0 (0x83) – Data Pointer 0 High Byte 41
DPL0 (0x82) – Data Pointer 0 Low Byte 41
DPH1 (0x85) – Data Pointer 1 High Byte 41
DPL1 (0x84) – Data Pointer 1 Low Byte 41
DPS (0x92) – Data Pointer Select 41
MPAGE (0x93) – Memory Page Select 42
PSW (0xD0) – Program Status Word 47
ACC (0xE0) – Accumulator 47
B (0xF0) – B Register 47
SP (0x81) – Stack Pointer 48
IEN0 (0xA8) – Interrupt Enable 0 54
IEN1 (0xB8) – Interrupt Enable 1 55
IEN2 (0x9A) – Interrupt Enable 2 56
TCON (0x88) – Interrupt Flags 57
S0CON (0x98) – Interrupt Flags 2 58
S1CON (0x9B) – Interrupt Flags 3 58
IRCON (0xC0) – Interrupt Flags 4 59
IRCON2 (0xE8) – Interrupt Flags 5 60
IP1 (0xB9) – Interrupt Priority 1 61
IP0 (0xA9) – Interrupt Priority 0 61
MEMCTR (0xC7) – Memory Arbiter Control CC2430-F128 67
MEMCTR (0xC7) – Memory Arbiter Control CC2430-F32/64 68
P0 (0x80) – Port 0 73
P1 (0x90) – Port 1 73
P2 (0xA0) – Port 2 73
PERCFG (0xF1) – Peripheral Control 74
ADCCFG (0xF2) – ADC Input Configuration 74
P0SEL (0xF3) – Port 0 Function Select 75
P1SEL (0xF4) – Port 1 Function Select 76
P2SEL (0xF5) – Port 2 Function Select 77
P0DIR (0xFD) – Port 0 Direction 78
P1DIR (0xFE) – Port 1 Direction 79
P2DIR (0xFF) – Port 2 Direction 80
P0INP (0x8F) – Port 0 Input Mode 81
P1INP (0xF6) – Port 1 Input Mode 82
P2INP (0xF7) – Port 2 Input Mode 83
P0IFG (0x89) – Port 0 Interrupt Status Flag 83
P1IFG (0x8A) – Port 1 Interrupt Status Flag 83
P2IFG (0x8B) – Port 2 Interrupt Status Flag 84
PICTL (0x8C) – Port Interrupt Control 84
P1IEN (0x8D) – Port 1 Interrupt Mask 85
DMAARM (0xD6) – DMA Channel Arm 95
DMAREQ (0xD7) – DMA Channel Start Request and Status 96
DMA0CFGH (0xD5) – DMA Channel 0 Configuration Address High Byte 96
DMA0CFGL (0xD4) – DMA Channel 0 Configuration Address Low Byte 96
DMA1CFGH (0xD3) – DMA Channel 1-4 Configuration Address High Byte 97
DMA1CFGL (0xD2) – DMA Channel 1-4 Configuration Address Low Byte 97
DMAIRQ (0xD1) – DMA Interrupt Flag 97

T1CNTH (0xE3) – Timer 1 Counter High 106
T1CNTL (0xE2) – Timer 1 Counter Low 106
T1CTL (0xE4) – Timer 1 Control and Status 106
T1CCTL0 (0xE5) – Timer 1 Channel 0 Capture/Compare Control 107
T1CC0H (0xDB) – Timer 1 Channel 0 Capture/Compare Value High 107
T1CC0L (0xDA) – Timer 1 Channel 0 Capture/Compare Value Low 107
T1CCTL1 (0xE6) – Timer 1 Channel 1 Capture/Compare Control 108
T1CC1H (0xDD) – Timer 1 Channel 1 Capture/Compare Value High 108
T1CC1L (0xDC) – Timer 1 Channel 1 Capture/Compare Value Low 108
T1CCTL2 (0xE7) – Timer 1 Channel 2 Capture/Compare Control 109
T1CC2H (0xDF) – Timer 1 Channel 2 Capture/Compare Value High 109
T1CC2L (0xDE) – Timer 1 Channel 2 Capture/Compare Value Low 109
T2CNF (0xC3) – Timer 2 Configuration 113
T2THD (0xA7) – Timer 2 Timer Value High Byte 114
T2TLD (0xA6) – Timer 2 Timer Value Low Byte 114
T2CMP (0x94) – Timer 2 Compare Value 114
T2OF2 (0xA3) – Timer 2 Overflow Count 2 114
T2OF1 (0xA2) – Timer 2 Overflow Count 1 115
T2OF0 (0xA1) – Timer 2 Overflow Count 0 115
T2CAPHPH (0xA5) – Timer 2 Period High Byte 115
T2CAPLPL (0xA4) – Timer 2 Period Low Byte 115
T2PEROF2 (0x9E) – Timer 2 Overflow Capture/Compare 2 116
T2PEROF1 (0x9D) – Timer 2 Overflow Capture/Compare 1 116
T2PEROF0 (0x9C) – Timer 2 Overflow Capture/Compare 0 116
ST2 (0x97) – Sleep Timer 2 117
ST1 (0x96) – Sleep Timer 1 117
ST0 (0x95) – Sleep Timer 0 117
T3CNT (0xCA) – Timer 3 Counter 120
T3CTL (0xCB) – Timer 3 Control 120
T3CCTL0 (0xCC) – Timer 3 Channel 0 Capture/Compare Control 121
T3CC0 (0xCD) – Timer 3 Channel 0 Capture/Compare Value 121
T3CCTL1 (0xCE) – Timer 3 Channel 1 Capture/Compare Control 122
T3CC1 (0xCF) – Timer 3 Channel 1 Capture/Compare Value 122
T4CNT (0xEA) – Timer 4 Counter 122
T4CTL (0xEB) – Timer 4 Control 123
T4CCTL0 (0xEC) – Timer 4 Channel 0 Capture/Compare Control 124
T4CC0 (0xED) – Timer 4 Channel 0 Capture/Compare Value 124
T4CCTL1 (0xEE) – Timer 4 Channel 1 Capture/Compare Control 125
T4CC1 (0xEF) – Timer 4 Channel 1 Capture/Compare Value 125
TIMIF (0xD8) – Timers 1/3/4 Interrupt Mask/Flag 126
ADCL (0xBA) – ADC Data Low 129
ADCH (0xBB) – ADC Data High 130
ADCCON1 (0xB4) – ADC Control 1 130
ADCCON2 (0xB5) – ADC Control 2 131
ADCCON3 (0xB6) – ADC Control 3 132
RNDL (0xBC) – Random Number Generator Data Low Byte 134
RNDH (0xBD) – Random Number Generator Data High Byte 134
ENCCS (0xB3) – Encryption Control and Status 139
ENCDI (0xB1) – Encryption Input Data 139
ENCDO (0xB2) – Encryption Output Data 139
PCON (0x87) – Power Mode Control 142
SLEEP (0xBE) – Sleep Mode Control 142

CLKCON (0xC6) – Clock Control 143
WDCTL (0xC9) – Watchdog Timer Control 146
U0CSR (0x86) – USART 0 Control and Status 151
U0UCR (0xC4) – USART 0 UART Control 152
U0GCR (0xC5) – USART 0 Generic Control 153
U0DBUF (0xC1) – USART 0 Receive/Transmit Data Buffer 153
U0BAUD (0xC2) – USART 0 Baud Rate Control 153
U1CSR (0xF8) – USART 1 Control and Status 154
U1UCR (0xFB) – USART 1 UART Control 155
U1GCR (0xFC) – USART 1 Generic Control 156
U1DBUF (0xF9) – USART 1 Receive/Transmit Data Buffer 156
U1BAUD (0xFA) – USART 1 Baud Rate Control 156
FCTL (0xAE) – Flash Control 162
FWDATA (0xAF) – Flash Write Data 162
FADDRH (0xAD) – Flash Address High Byte 162
FADDRL (0xAC) – Flash Address Low Byte 162
FWT (0xAB) – Flash Write Timing 163
RFIF (0xE9) – RF Interrupt Flags 168
RFIM (0x91) – RF Interrupt Mask 169
RFD (0xD9) – RF Data 170
RFST (0xE1) – RF CSMA-CA / Strobe Processor 190

19 Package Description (QLP 48)

All dimensions are in millimeters, angles in degrees. NOTE: The ***CC2430*** is available in RoHS lead-free package only. Compliant with JEDEC MS-020.

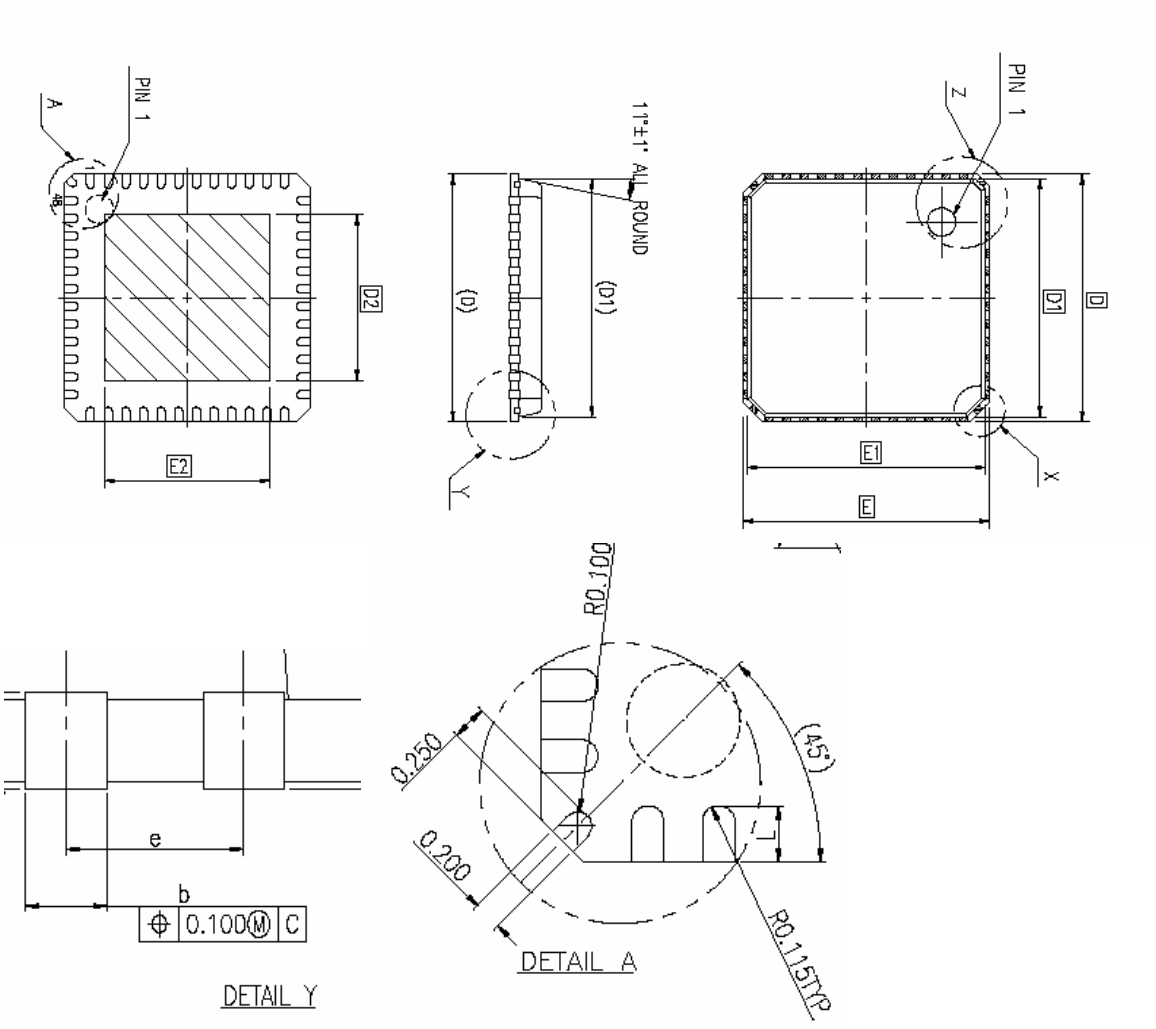


Figure 59: Package dimensions drawing

Quad Leadless Package (QLP)										
		D	D1	E	E1	e	b	L	D2	E2
QLP 48	Min	6.9	6.65	6.9	6.65		0.18	0.3	5.05	5.05
		7.0	6.75	7.0	6.75	0.5		0.4	5.10	5.10
	Max	7.1	6.85	7.1	6.85		0.30	0.5	5.15	5.15
The overall package height is 0.85 +/- 0.05 All dimensions in mm										

Table 102: Package dimensions

19.1 Recommended PCB layout for package (QLP 48)

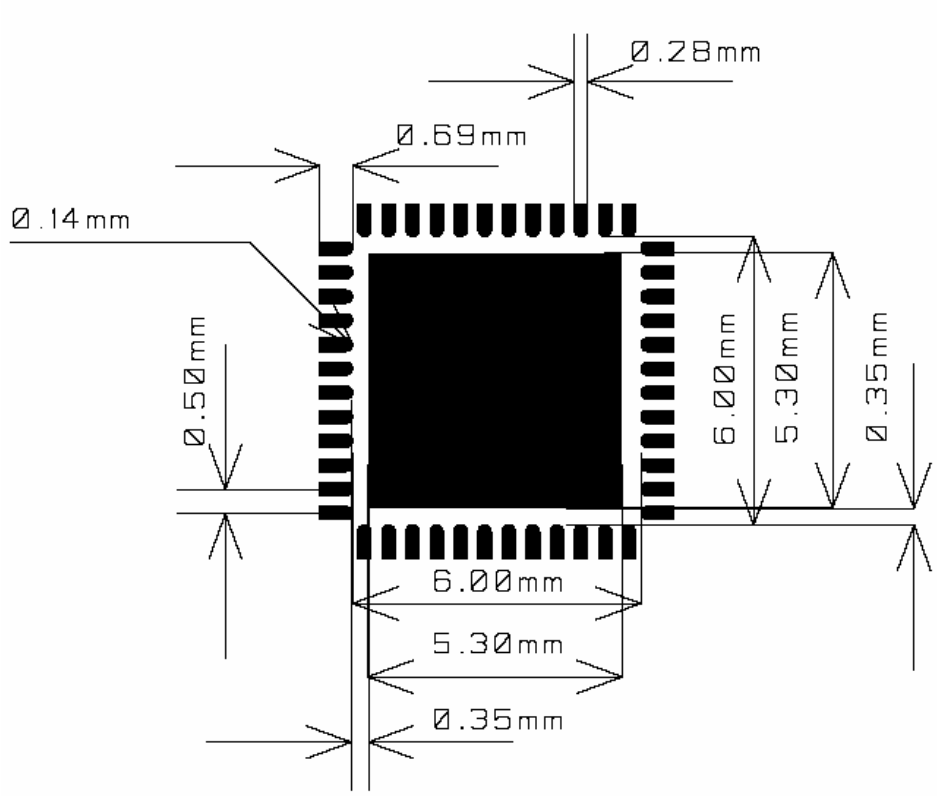


Figure 60: Recommended PCB layout for QLP 48 package

Note: The figure is an illustration only and not to scale. There are nine 14 mil diameter via holes distributed symmetrically in the ground pad under the package. See also the ***CC2430*** EM reference design.

19.2 Package thermal properties

Thermal resistance	
Air velocity [m/s]	0
Rth,j-a [K/W]	25.6

Table 103: Thermal properties of QLP 48 package

19.3 Soldering information

The recommendations for lead-free solder reflow in IPC/JEDEC J-STD-020C should be followed.

19.4 Plastic tube specification

Tube Specification				
Package	Tube Width	Tube Height	Tube Length	Units per Tube
QLP 48	8.5mm ± 0.2mm	2.2mm +0.2/–0.1mm	315mm ± 1.25mm	43

Table 104: Plastic tube specification

19.5 Carrier tape and reel specification

Carrier tape and reel is in accordance with EIA Specification 481.

Tape and Reel Specification					
Package	Tape Width	Component Pitch	Hole Pitch	Reel Diameter	Units per Reel
QLP 48	16mm	12mm	4mm	13 inches	2500

Table 105: Carrier tape and reel specification

20 Ordering Information

Ordering part number		Description	MOQ
Ti-Part	Chipcon Part		
CC2430F-128RTC	CC2430-RTB1-F128	CC2430, QLP48 package, RoHS compliant Pb-free assembly, tubes with 43 pcs per tube, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver.	43
CC2430F-128RTCR	CC2430-RTR1-F128	CC2430, QLP48 package, RoHS compliant Pb-free assembly, T&R with 2500 pcs per reel, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver.	2,500
CC2430F-128ZRTC	CC2430-RTB1-F128Z	CC2430, QLP48 package, RoHS compliant Pb-free assembly, tubes with 43 pcs per tube, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver, Including licence fee.	43
CC2430F-128ZRTCR	CC2430-RTR1-F128Z	CC2430, QLP48 package, RoHS compliant Pb-free assembly, T&R with 2500 pcs per reel, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver, Including licence fee.	2,500
CC2430F-64RTC	CC2430-RTB1-F64	CC2430, QLP48 package, RoHS compliant Pb-free assembly, tubes with 43 pcs per tube, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver.	43
CC2430F-64RTCR	CC2430-RTR1-F64	CC2430, QLP48 package, RoHS compliant Pb-free assembly, T&R with 2500 pcs per reel, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver.	2,500
CC2430F64ZRTC	CC2430-RTB1-F64Z	CC2430, QLP48 package, RoHS compliant Pb-free assembly, tubes with 43 pcs per tube, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver, Including licence fee.	43
CC2430F-64ZRTCR	CC2430-RTR1-F64Z	CC2430, QLP48 package, RoHS compliant Pb-free assembly, T&R with 2500 pcs per reel, 64 Kbytes in-system programmable flash memory, System-on-chip RF transceiver, Including licence fee.	2,500
CC2430F-32RTC	CC2430-RTB1-F32	CC2430, QLP48 package, RoHS compliant Pb-free assembly, tubes with 43 pcs per tube, 32 Kbytes in-system programmable flash memory, System-on-chip RF transceiver.	43
CC2430F-32RTCR	CC2430-RTR1-F32	CC2430, QLP48 package, RoHS compliant Pb-free assembly, T&R with 2500 pcs per reel, 32 Kbytes in-system programmable flash memory, System-on-chip RF transceiver.	2,500
CC2430F-32ZRTC	CC2430-RTB1-F32Z	CC2430, QLP48 package, RoHS compliant Pb-free assembly, tubes with 43 pcs per tube, 32 Kbytes in-system programmable flash memory, System-on-chip RF transceiver, Including licence fee.	43
CC2430F-32ZRTCR	CC2430-RTR1-F32Z	CC2430, QLP48 package, RoHS compliant Pb-free assembly, T&R with 2500 pcs per reel, 32 Kbytes in-system	2,500

		programmable flash memory, System-on-chip RF transceiver, Including licence fee.	
CC2430DK	CC2430DK	CC2430 ZigBee DK Development kit.	1
CC2430ZDK Pro	CC2430ZDK Pro	CC2430 ZigBee DK Development kit including support and training	1

MOQ = Minimum Order Quantity

T&R = tape and reel

Table 106: Ordering Information

21 General Information

21.1 Document History

Revision	Date	Description/Changes
1.03		Updated ADC characteristics (Table 14). Updated DMAREQ, DMAARM and changed section 13.2 first paragraph, section 13.2.4 and Figure 17 Registers UxBUF renamed as UxDBUF. EAL renamed as EA. Added code protection and encryption disclaimers. Clarified flash write description Updated DC characteristics (Table 20) Added information about clearing PxIFG in section 13.1.2 Removed T2CON Updated Table 45 Changed formatting of Table 25 STXCALN/ISTXCALN are intended for test only Modified Figure 37 Updated MDMCTRL0L.PREAMBLE_LENGTH Clarified use of ADC conversions triggered by ADCCON3 Clarified RNDL/RNDH descriptions, renamed Random Generator to Random Number Generator Expanded description and changed reset value for SLEEP.OSC_PD Corrected P0/P1/P2 reset values Changed reference to SXOSCON to CLKCON Changed RFSTATUS.SFD description Corrected references to Figure 42 and Figure 43 Corrected descriptions in FSMTCH/FSMTCL Corrected numbers in Table 32 Various clarifications in Timers, ADC and Power Management Changed typical start-up time for 32 MHz crystal oscillator Corrected T1CCTL1.CMP and T1CCTL2.CMP Updated address information Changed Random Number Generator seeding; modified ADCCON1.RCTRL and section 13.8.2.2 Changed descriptions for T2PEROF2, T2PEROF1 and T2PEROF0 Changed Figure 15 and section 12.3.2; unified CODE space uses memory banking Added CLKCON.CLKSPD bit, and altered descriptions in CLKCON Corrected RSSIL description Updated ordering information Added SSN description section 13.13.3 and updated Table 16 and Figure 2 Preliminary status updated
1.02	2005-10-03	Added ADC external trigger (section 13.1.5 and 13.7.2). Updated sensitivity. Corrected Application circuit
1.01	2005-09-15	Updated Table 103, Units per reel
1.0	2005-09-12	First release, preliminary

Table 107: Document History

21.2 Product Status Definitions

Data Sheet Identification	Product Status	Definition
Advance Information	Planned or Under Development	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	Engineering Samples and Pre-Production Prototypes	This data sheet contains preliminary data, and supplementary data will be published at a later date. Chipcon reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. The product is not yet fully qualified at this point.
No Identification Noted	Full Production	This data sheet contains the final specifications. Chipcon reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Obsolete	Not In Production	This data sheet contains specifications on a product that has been discontinued by Chipcon. The data sheet is printed for reference information only.

Table 108: Product Status Definitions

21.3 Disclaimer

Chipcon AS believes the information contained herein is correct and accurate at the time of this printing. However, Chipcon AS reserves the right to make changes to this product without notice. Chipcon AS does not assume any responsibility for the use of the described product; neither does it convey any license under its patent rights, or the rights of others. The latest updates are available at the Chipcon website or by contacting Chipcon directly.

As far as possible, major changes of product specifications and functionality, will be stated in product specific Errata Notes published at the Chipcon website. Customers are encouraged to sign up for the Chipcon Newsletter for the most recent updates on products and support tools.

When a product is discontinued this will be done according to Chipcon's procedure for obsolete products as described in Chipcon's Quality Manual. This includes informing about last-time-buy options. The Quality Manual can be downloaded from Chipcon's website.

Compliance with regulations is dependent on complete system performance. It is the customer's responsibility to ensure that the system complies with regulations.

The ZigBee Specification includes intellectual property rights of ZigBee Alliance member/promoter companies. Chipcon is a ZigBee Alliance Promoter. Under the ZigBee Alliance terms of use, no part of the Specification may be used by a company in the development of a product for sale without such company becoming a member of the ZigBee Alliance. Therefore, the Figure 8 Wireless Z-Stack™ may only be used for commercial purposes by ZigBee Alliance member companies. If a customer desires to use the Figure 8 Wireless Z-Stack™ or any other third party ZigBee stack together with a product described in this datasheet, the customer is responsible for complying with the applicable ZigBee Alliance policies. See http://www.zigbee.org.

This Chipcon product contains Flash memory code protection. However, Chipcon does not guarantee the security of this protection. Chipcon customers using or selling these products with program code do so at their own risk and agree to fully indemnify Chipcon AS for any damages resulting from the use or sale of such products.

Chipcon believes that the Flash memory protection used in this product is one of the most secure in the market today when used in the intended manner and under normal conditions. However, there might be methods to breach the code protection feature. Neither Chipcon nor any other semiconductor manufacturer can guarantee the security of their code protection. Code protection does not mean that we are guaranteeing the product as "unbreakable".

This Chipcon product contains hardware AES encryption. Chipcon does not guarantee the security of the key protection or the security of the encryption scheme. Chipcon customers using or selling products with AES do so at their own risk and agree to fully indemnify Chipcon AS for any damages resulting from the use or sale of such products.

It is the Chipcon customer's responsibility to ensure that sale or export/import of products including this Chipcon product with AES encryption is sold with the required export/import licenses, if necessary, and does not violate any applicable export/import and/or other trade restrictions.

21.4 Trademarks

SmartRF® is a registered trademark of Chipcon AS. SmartRF® is Chipcon's RF technology platform with RF library cells, modules and design expertise. Based on SmartRF® technology Chipcon develops standard component RF circuits as well as full custom ASICs based on customer requirements and this technology.

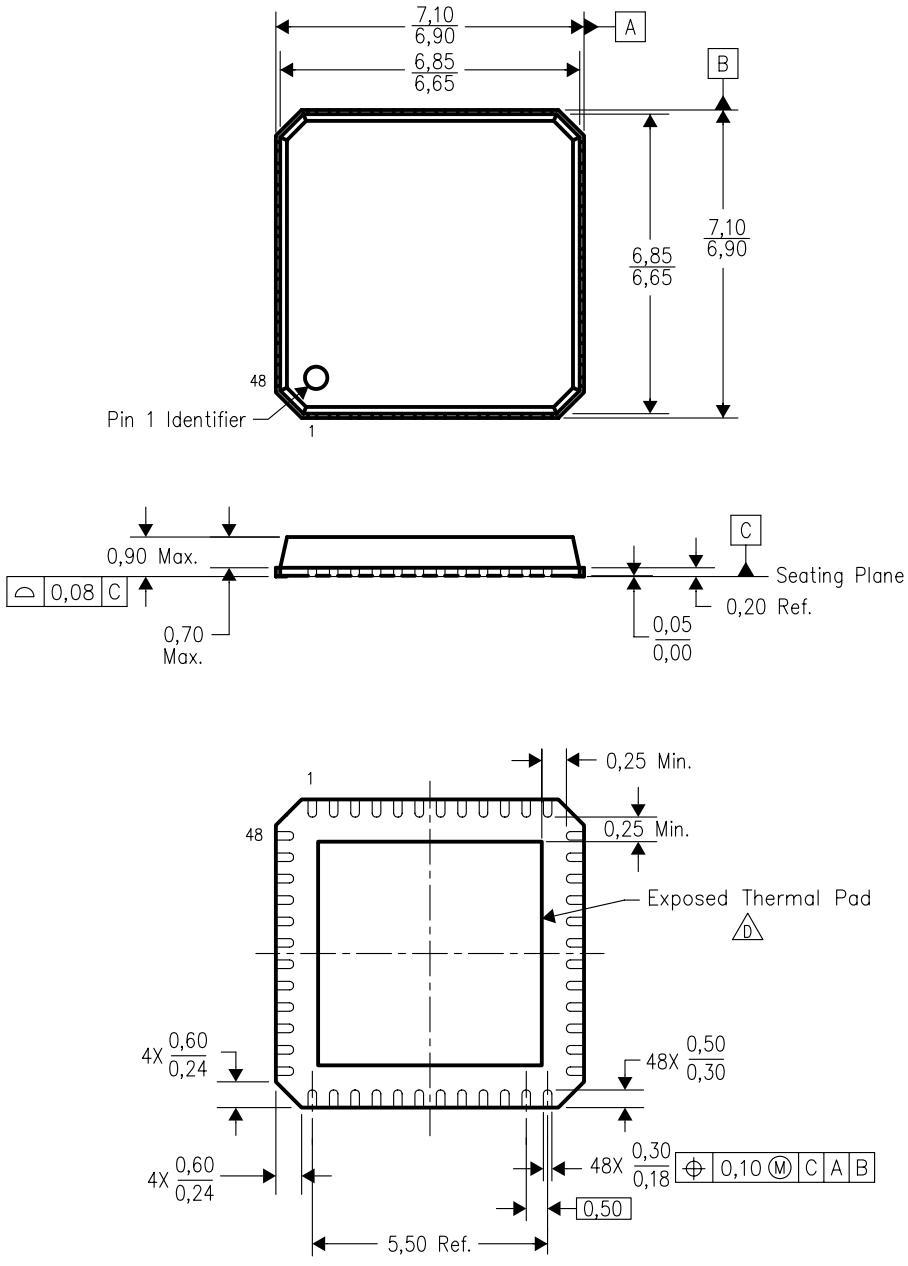
All other trademarks, registered trademarks and product names are the sole property of their respective owners.

21.5 Life Support Policy

This Chipcon product is not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Chipcon AS customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Chipcon AS for any damages resulting from any improper use or sale.

© 2006, Chipcon AS. All rights reserved.

RTC (S–PQFP–N48) PLASTIC QUAD FLATPACK



NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. QFN (Quad Flatpack No–Lead) Package configuration.
D. The package thermal pad must be soldered to the board for thermal and mechanical performance. See the Product Data Sheet for details regarding the exposed thermal pad dimensions.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2006, Texas Instruments Incorporated