**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**APPLICATIONS**

• **IEEE 802.15.4 systems**

• **ZigBee® systems**

• **Industrial monitoring and control**

• **Home and building automation**

• **Automatic Meter Reading**

• **Low-power wireless sensor networks** • **Set-top boxes and remote controls** • **Consumer electronics**

**KEY FEATURES**

• **State-of-the-art selectivity/co-existence Adjacent channel rejection: 49 dB**

**Alternate channel rejection: 54 dB**

• **Excellent link budget (103dB)**

**400 m Line-of-sight range**

• **Extended temp range (-40 to +125°C)** • **Wide supply range: 1.8 V – 3.8 V**

• **Extensive IEEE 802.15.4 MAC hardware support to offload the microcontroller** • **AES-128 security module**

• **CC2420 interface compatibility mode**

**Low Power**

• **RX (receiving frame, -50 dBm) 18.5 mA** • **TX 33.6 mA @ +5 dBm**

• **TX 25.8 mA @ 0 dBm**

• **<1**∝**A in power down**

**General**

• **Clock output for single crystal systems** • **RoHS compliant 5 x 5 mm QFN28 (RHD) package**

**DESCRIPTION**

**The CC2520 is TI's second generation ZigBee® / IEEE 802.15.4 RF transceiver for the 2.4 GHz unlicensed ISM band. This chip enables industrial grade applications by offering state-of-the-art**

**Radio**

• **IEEE 802.15.4 compliant DSSS baseband modem with 250 kbps data rate**

• **Excellent receiver sensitivity (-98 dBm)** • **Programmable output power up to +5 dBm** • **RF frequency range 2394-2507 MHz** • **Suitable for systems targeting compliance with worldwide radio frequency**

**regulations: ETSI EN 300 328 and EN 300 440 class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan)**

**Microcontroller Support**

• **Digital RSSI/LQI support**

• **Automatic clear channel assessment for CSMA/CA**

• **Automatic CRC**

• **768 bytes RAM for flexible buffering and security processing**

• **Fully supported MAC security**

• **4 wire SPI**

• **6 configurable IO pins**

• **Interrupt generator**

• **Frame filtering and processing engine** • **Random number generator**

**Development Tools**

• **Reference design**

• **IEEE 802.15.4 MAC software**

• **ZigBee® stack software**

• **Fully equipped development kit** • **Packet sniffer support in hardware**

**QFN28 (RHD) PACKAGE**

**TOP VIEW**

D

R

A

**selectivity/co-existence, excellent link budget, operation up to 125°C and low voltage operation.**

**In addition, the CC2520 provides extensive hardware support for frame handling, data buffering, burst transmissions, data encryption, data authentication, clear channel assessment,**

SO

SI

CSn

K

L

C

S

8

2

1

2

3

L P

U

OCD

7 2

N E

GE

RV

6 2

\_

n T

E

SE

R

5 2

UG

D

DVA

4 2

\_

S

A

I

B

R

3

2

4

D

D

V

A

2

2

21

20

19

NC

AVDD1 RF\_N

**link quality indication and frame timing information. These features reduce the load on the host controller.**

GPIO5 GPIO4 GPIO3 GPIO2

4 5 6 7

CC2520

18 17 16 15

NC

RF\_P

AVDD2 NC

**In a typical system, the CC2520 will be used together with a microcontroller and a few additional passive components.**

89

1

D

O

D

I

V

P

D

G

0 1

0

OI

PG

1 1

5

D

DVA

2 1

2 Q

M23

CS

OX

\_

3 1

1 Q

M23

CS

OX

\_

4 1

3

D

DVA

AGND

exposed die attached pad

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers threto appear at the end of this datasheet.

ZigBee® is a registered trademark owned by ZigBee Alliance, Inc.

Copyright © 2007, Texas Instruments Incorporated

**WWW.TI.COM** 1

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**TABLE OF CONTENTS**

1 Abbreviations ...............................................................................................................................5 2 References...................................................................................................................................7 3 Features.......................................................................................................................................8 4 Absolute Maximum Ratings .......................................................................................................10 5 Electrical Characteristics............................................................................................................11

5.1 Recommended Operating Conditions ............................................................................11 5.2 DC Characteristics .........................................................................................................11 5.3 Wake-Up and Timing .....................................................................................................11 5.4 Current Consumptions ...................................................................................................11 5.5 Receive Parameters.......................................................................................................12 5.6 Frequency Synthesizer Parameters...............................................................................12

5.6.1 Transmit Parameters..................................................................................................12 5.7 RSSI/CCA Parameters...................................................................................................13 5.8 FREQEST Parameters...................................................................................................13 5.9 Typical Performance Curves..........................................................................................14 5.10 Low-Current Mode RX....................................................................................................19

5.10.1 Low-Current RX Mode Parameters ............................................................................19 5.11 Optional Temperature Compensation of TX...................................................................20 5.11.1 Using the Temperature Sensor ..................................................................................21 6 Crystal Specific Parameters.......................................................................................................22 6.1 Crystal Requirements.....................................................................................................22 6.2 On-chip Crystal Frequency Tuning.................................................................................22 7 Pinout.........................................................................................................................................23 8 Functional Introduction...............................................................................................................25 8.1 Integrated 2.4 GHz IEEE 802.15.4 Compliant Radio .....................................................25 8.2 Comparison to CC2420..................................................................................................25 8.3 Block Diagram................................................................................................................26 9 Application Circuit ......................................................................................................................29 9.1 Input / Output Matching..................................................................................................29 9.2 Bias Resistor ..................................................................................................................30 9.3 Crystal ............................................................................................................................30 9.4 Digital Voltage Regulator................................................................................................30 9.5 Power Supply Decoupling and Filtering .........................................................................30 9.6 Board Layout Guidelines................................................................................................30 9.7 Antenna Considerations.................................................................................................31 9.8 Choosing the Most Suitable Interconnection with a Microcontroller...............................31 9.9 Interfacing CC2520 and MSP430F2618 ........................................................................31 10 Serial Peripheral Interface (SPI) ................................................................................................33 10.1 CSn ................................................................................................................................33 10.2 SCLK..............................................................................................................................33 10.3 SI....................................................................................................................................33 10.4 SO ..................................................................................................................................34 10.5 SPI Timing Requirements ..............................................................................................34 11 GPIO ..........................................................................................................................................35 11.1 Reset Configuration of GPIO Pins..................................................................................35 11.2 GPIO as Input ................................................................................................................35 11.3 GPIO as Output..............................................................................................................36 11.4 Switching Direction on GPIO..........................................................................................36 11.5 GPIO Configuration........................................................................................................36 12 Power Modes.............................................................................................................................40 12.1 Switching Between Power Modes..................................................................................40 12.2 Power Up Sequence Using RESETn (recommended)...................................................41

2 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

12.3 Power Up With SRES ....................................................................................................41 13 Instruction Set............................................................................................................................43 13.1 Definitions ......................................................................................................................43 13.2 Instruction Descriptions..................................................................................................43 13.3 Instruction Set Summary................................................................................................51 13.4 Status Byte.....................................................................................................................53 13.5 Command Strobes .........................................................................................................53 13.6 Command Strobe Buffer ................................................................................................53 14 Exceptions .................................................................................................................................55 14.1 Exceptions on GPIO Pins...............................................................................................56 14.2 Predefined Exception Channels.....................................................................................56 14.3 Binding Exceptions to Instructions (command strobes) .................................................57 15 Memory Map..............................................................................................................................59 15.1 FREG .............................................................................................................................60 15.2 SREG .............................................................................................................................60 15.3 TX FIFO .........................................................................................................................60 15.4 RX FIFO .........................................................................................................................60 15.5 MEM...............................................................................................................................60 15.6 Frame Filtering and Source Matching Memory Map ......................................................60 16 Frequency and Channel Programming ......................................................................................62 17 IEEE 802.15.4-2006 Modulation Format....................................................................................63 18 IEEE 802.15.4-2006 Frame Format...........................................................................................65 18.1 PHY Layer......................................................................................................................65 18.2 MAC Layer .....................................................................................................................65 19 Transmit Mode...........................................................................................................................67 19.1 TX Control ......................................................................................................................67 19.2 TX State Timing .............................................................................................................67 19.3 TX FIFO Access.............................................................................................................67 19.3.1 Retransmission...........................................................................................................68 19.3.2 Error Conditions .........................................................................................................68 19.4 TX Flow Diagram ...........................................................................................................69 19.5 Frame Processing ..........................................................................................................70 19.5.1 Synchronization Header.............................................................................................70 19.5.2 Frame Length Field ....................................................................................................70 19.5.3 Frame Check Sequence.............................................................................................70 19.6 Exceptions......................................................................................................................71 19.7 Clear Channel Assessment............................................................................................71 19.8 Output Power Programming...........................................................................................71 19.9 Tips And Tricks ..............................................................................................................72 20 Receive Mode............................................................................................................................73 20.1 RX Control......................................................................................................................73 20.2 RX State Timing .............................................................................................................73 20.3 Frame Processing ..........................................................................................................73 20.3.1 Synchronization Header And Frame Length Fields....................................................74 20.3.2 Frame Filtering ...........................................................................................................74 20.3.3 Source Address Matching ..........................................................................................77 20.3.4 Frame Check Sequence.............................................................................................80 20.3.5 Acknowledgement Transmission................................................................................81 20.4 RX FIFO Access ............................................................................................................82 20.4.1 Using the FIFO and FIFOP Signals............................................................................82 20.4.2 Error Conditions .........................................................................................................83 20.5 RSSI...............................................................................................................................83 20.6 Link Quality Indication ....................................................................................................84

**WWW.TI.COM** 3

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

21 Radio Control State Machine .....................................................................................................85 22 Crystal Oscillator........................................................................................................................87 23 External Clock Output ................................................................................................................88 24 Random Number Generation.....................................................................................................89 25 Memory Management Instructions.............................................................................................91

25.1 RXBUFMOV...................................................................................................................92 25.2 TXBUFCP ......................................................................................................................92 25.3 MEMCP..........................................................................................................................92 25.4 MEMCPR .......................................................................................................................92 25.5 MEMXCP .......................................................................................................................92

26 Security Instructions...................................................................................................................93 26.1 Decoding of the Flags Field in CC2520..........................................................................93 26.2 INC .................................................................................................................................94 26.3 ECB................................................................................................................................94 26.4 ECBO .............................................................................................................................95 26.5 ECBX .............................................................................................................................95 26.6 CTR / UCTR...................................................................................................................96 26.7 CBC-MAC ......................................................................................................................97 26.8 CCM / UCCM .................................................................................................................97

26.8.1 Inputs to the CCM and UCCM Instructions ................................................................97 26.9 Examples from IEEE802.15.4-2006...............................................................................98 26.9.1 Authentication Only Using CCM\* ...............................................................................99 26.9.2 Encryption Only Using CCM\* .....................................................................................99 26.9.3 Combination of Encryption and Authentication Using CCM\*....................................100 27 Packet Sniffing.........................................................................................................................101 28 Registers..................................................................................................................................102 28.1 Register Settings Update .............................................................................................103 28.2 Register Access Modes................................................................................................103 28.3 Register Descriptions ...................................................................................................105 29 Datasheet Revision History......................................................................................................126 30 Packaging Information .............................................................................................................127 30.1 Mechanical Data ..........................................................................................................128

4 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**1 Abbreviations**

AAF Anti Aliasing Filter

ACK Acknowledge

ADC Analog to Digital Converter

ADI Analog-Digital Interface

AES Advanced Encryption Standard

AGC Automatic Gain Control

AM Active Mode

ARIB Association of Radio Industries and Businesses BER Bit Error Rate

BIST Built In Self Test

CBC-MAC Cipher Block Chaining Message Authentication Code CCA Clear Channel Assessment

CCM Counter mode + CBC-MAC

CDM Charged Device Model

CFR Code of Federal Regulations

CHP Charge Pump

CMOS Complementary Metal Oxide Semiconductor CRC Cyclic Redundancy Check

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance CTR Counter mode (encryption)

CW Continuous Wave

DAC Digital to Analog Converter

DC Direct Current

DPU Data Processing Unit

DSSS Direct Sequence Spread Spectrum

ECB Electronic Code Book (mode of AES operation) ESD Electro Static Discharge

ESR Equivalent Series Resistance

ETSI European Telecommunications Standards Institute EU European Union

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

FS Frequency Synthesizer

FSM Finite State Machine

GPIO General Purpose Input/Output

HBM Human Body Model

HSSD High Speed Serial Debug

I/O Input / Output

I/Q In-phase / Quadrature-phase

IEEE Institute of Electrical and Electronics Engineers IF Intermediate Frequency

ISM Industrial, Scientific and Medical

ITU-T International Telecommunication Union –

Telecommunication Standardization Sector

kbps kilo bits per second

LB Loop Back

LF Loop Filter

LNA Low-Noise Amplifier

LO Local Oscillator

LPF Low Pass Filter

LPM Low-Power Mode

**WWW.TI.COM** 5

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

LQI Link Quality Indication

LSB Least Significant Bit / Byte

LUT Look-Up Table

MAC Medium Access Control

MCU Micro Controller Unit

MFR MAC Footer

MHR MAC Header

MIC Message Integrity Code

MISO Master In Slave Out

MM Machine Model

MOSI Master Out Slave In

MPDU MAC Protocol Data Unit

MSB Most significant Bit / Byte

MSDU MAC Service Data Unit

NA Not Available

NC Not Connected

O-QPSK Offset - Quadrature Phase Shift Keying

PA Power Amplifier

PAN Personal Area Network

PCB Printed Circuit Board

PD Power Down, Phase Detector

PER Packet Error Rate

PHR PHY Header

PHY Physical Layer

PLL Phase Locked Loop

PQFP Plastic Quad FlatPack

PSDU PHY Service Data Unit

PUE Pull-Up Enable

QLP Quad Leadless Package

RAM Random Access Memory

RBW Resolution BandWidth

RF Radio Frequency

RHD Not actually an acronym. This is the package name used in TI. RISC Reduced Instruction Set Computer

RoHS Restriction of Hazardous Substances Directive ROM Read Only Memory

RSSI Received Signal Strength Indicator

RX Receive

SFD Start of Frame Delimiter

SHR Synchronization Header

SI Serial In

SO Serial Out

SPI Serial Peripheral Interface

S-PQFP Plastic Quad Flat Pack

T/R Transmit / Receive

TBD To Be Decided / To Be Defined

TX Transmit

UI User Interface

VCO Voltage Controlled Oscillator

VGA Variable Gain Amplifier

XOSC Crystal Oscillator

LR Low Rate

NaN Not any Number

6 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**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] IEEE std. 802.15.4 - 2006: 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/802.15.4-2006.pdf

[3] CC2420 datasheet

http://www.ti.com/lit/pdf/swrs041

[4] 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.

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

[5] CC2520 reference designs

http://focus.ti.com/docs/prod/folders/print/cc2520.html#applicationnotes

[6] CC2520 Errata note

http://www.ti.com/lit/pdf/swrz024

[7] CC2520 Product folder

http://focus.ti.com/docs/prod/folders/print/cc2520.html

[8] NIST software package for randomness testing:

http://csrc.nist.gov/rng/

[9] The diehard software package for randomness testing:

http://stat.fsu.edu/~geo/diehard.html

[10] MSP430F2618 Product folder

http://focus.ti.com/docs/prod/folders/print/msp430f2618.html

[11] 2.4 GHz Inverted F Antenna

http://www.ti.com/lit/pdf/swru120

[12] Antenna selection guide

http://www.ti.com/lit/pdf/swra161

**WWW.TI.COM** 7

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**3 Features**

**2394-2507MHz transceiver**

• DSSS transceiver

• 250kbps data rate, 2 MChip/s chip rate

• O-QPSK with half sine pulse shaping modulation

• Very low current consumption

RX (receiving frame, -50 dBm): 18.5 mA

RX (waiting for frame): 22.3 mA

TX (+5 dBm output power): 33.6 mA

TX (0 dBm output power): 25.8 mA

• Three flexible power modes for reduced power consumption • Low power fully static CMOS design

• Very good sensitivity (-98dBm)

• High adjacent channel rejection (49 dB)

• High alternate channel rejection (54 dB)

• On chip VCO, LNA, PA and filters.

• Low supply voltage (1.8 - 3.8 V)

• Programmable output power up to +5 dBm

• I/Q direct conversion transceiver

**Small Size**

• QFN 28 (RHD) package, 5 x 5 mm

• Very few external components

o minimized number of passives

o Only reference crystal needed

• Clock output for other ICs to limit the number of crystals needed in a system • No external filters needed.

**Easy and Flexible User Interface**

• 4-wire SPI

• Serial clock up to 8 MHz

• 6 GPIO pins with full flexibility

• Interrupt generator

• Full control of automatic responses to different events

• Embedded packet sniffer mode

• CC2420 compatibility mode

**Data Processing Unit For Advanced Data Handling**

• Spacious (768 byte) on-chip RAM allows powerful on-chip frame processing • 128 byte transmit data FIFO

• 128 byte receive data FIFO

• Full read and write access to RAM

• 128 bit AES

**IEEE 802.15.4 MAC Hardware Support**

• Automatic preamble generator

• Synchronization word insertion and detection

• CRC-16 computation and verification over the MAC payload • Frame filtering

• Automatic ACK and setting of the pending-bit

• Clear Channel Assessment (CCA)

• Energy detection / RSSI

• Link Quality Indication (LQI)

• Fully automatic MAC security (CTR, CBC-MAC, CCM)

8 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**Development Tools**

• See product folder [7]

**Suited For Use in Systems That Target Compliance to the Following Standards** • IEEE 802.15.4 PHY

• ETSI EN 300 328

• ETSI EN 300 440 class 2

• FCC CFR47 part 15

• ARIB STD-T66

**WWW.TI.COM** 9

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**4 Absolute Maximum Ratings**

over operating free-air temperature range unless otherwise noted (1)

| **PARAMETER** | **LIMITS** | **UNIT** |
| --- | --- | --- |
| Supply voltage (2)  Voltage on any digital pin  Voltage on 1.8 V pins  Input RF level  Storage temperature range  Reflow soldering temperature  ESD HBM  ESD CDM  ESD MM | -0.3 to 3.9  -0.3 to VDD + 0.3 (Max 3.9) -0.3 to 2.0  +10  -50 to 150  260  800  500  100 | V  V  V  dBm  °C  °C  V  V  V |

1) Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

2) All voltage values are with respect to network ground terminal.

This device has limited built-in gate protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

10 **WWW.TI.COM**

**5 Electrical Characteristics**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

Note that these characteristics are only valid when using the recommended register settings presented in section 28.1.

**5.1 Recommended Operating Conditions**

| **PARAMETER** | **MIN NOM MAX** | **UNIT** |
| --- | --- | --- |
| Operating supply voltage  Ambient temperature | 1.8 3.8 -40 125 | V  °C |

**5.2 DC Characteristics**

TA =25°C, VDD=3.0 V, fc=2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Logic "1" input voltage Logic "0" input voltage Input pad hysteresis  Logic "0" input current Logic "1" input current | Valid for all pads (both GPIOs and fixed-input pads)  Valid for all pads (both GPIOs and fixed-input pads)  Only for fixed-input pads like RESET\_N, CSn etc  Input equals 0V  Input equals VDD | 80%  30%  0.5  -25 25 -25 25 | of VDD  of VDD  V  nA  nA |

**5.3 Wake-Up and Timing**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **COMMENTS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| LPM2 ⮴ AM time  LPM1 ⮴ AM time  AM ⮴ RX time  AM ⮴ TX time  RX/TX turnaround time TX/RX turnaround time Radio bit rate  Radio chip rate | Internal regulator startup time + XOSC startup time  XOSC startup time | 0.3  0.2  192  192  192  192  250  2.0 | ms  ms  ∝s  ∝s  ∝s  ∝s  kbps  MChip/s |

**5.4 Current Consumptions**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Receive current | Wait for sync  TA=-40 to 125°C, VDD=1.8 to 3.8 V, fc =2394 to 2507 MHz Wait for sync, Low-current RX setting  Receving frame, -50 dBm input level | 22.3 24.8  26.3  18.8  18.5 | mA  mA  mA  mA |
| Transmit current | 0 dBm setting  +5 dBm setting  TA=-40 to 125°C, VDD=1.8 to 3.8 V, fc =2394 to 2507 MHz | 25.8 28.8  33.6 37.2  37.5 | mA  mA  mA |
| Active Mode current | XOSC on, digital regulator on.  TA=-40 to 125°C, VDD=1.8 to 3.8 V, fc =2394 to 2507 MHz | 1.6 1.9  2.6 | mA  mA |

**WWW.TI.COM** 11

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| LPM1 current | XOSC off, digital regulator on. State retention.  TA=-40 to 125°C, VDD=1.8 to 3.8 V, fc =2394 to 2507 MHz | 175 250  1000 | ∝A  ∝A |
| LPM2 current | XOSC off, digital regulator off. No state retention.  TA=-40 to 125°C, VDD=1.8 to 3.8 V, fc =2394 to 2507 MHz | 30 120  4.5 | nA  ∝A |

**5.5 Receive Parameters**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Receiver sensitivity | [2] requires -85 dBm  TA=-40 to 125°C, VDD=1.8 to 3.8V, fc =2394 to 2507 MHz | -99 -98 -95 -88 | dBm  dBm |
| Saturation | [2] requires -20 dBm | 6 | dBm |
| Interferer Rejection | Wanted signal 3 dB above the sensitivity level, 802.15.4 modulated interferer at 802.15.4 channels:  ±5 MHz from wanted signal. [2] requires 0 dB  ±10 MHz from wanted signal. [2] requires 30 dB  ±20MHz or above. Wanted signal at -82dBm. | 49  54  55 | dB  dB  dB |
| Maximum Spurious  Emission  Conducted measurement in a 50Ω single ended load. Complies with EN 300 328, EN 300 440 class 2, FCC CFR47, Part 15 and ARIB STD-T-66 | 30 – 1000 MHz  1 – 12.75 GHz | < -80  -56 | dBm  dBm |
| Frequency error  tolerance | Input level is 3 dB above sensitivity level. | +/-400 | kHz |
| IIP3 |  | -24 | dBm |

**5.6 Frequency Synthesizer Parameters**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Phase noise.  Unmodulated carrier | At ±1 MHz offset from carrier  At ±2 MHz offset from carrier  At ±5 MHz offset from carrier | -111  -118  -128 | dBc/Hz  dBc/Hz  dBc/Hz |
| RF Frequency range | Programmable in 1 MHz steps. Use 5 MHz steps for compliance with [2]. | 2394 2507 | MHz |

**5.6.1 Transmit Parameters**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Output power  Note: to reduce the output power variation over  temperature, it is suggested that different settings are used at different  temperatures. The on-chip temperature sensor can be used for this purpose.  Please see section 5.11 for more information. | 0 dBm setting  +5 dBm setting  TA=-40 to 85°C, VDD=2.0 to 3.8 V, fc =2394 to 2507 MHz TA=-40 to 85°C , VDD=1.8 to 3.8 V, fc =2394 to 2507 MHz TA=-40 to 125°C, VDD=2.0 to 3.8 V, fc =2394 to 2507 MHz TA=-40 to 125°C, VDD=1.8 to 3.8 V, fc =2394 to 2507 MHz | -3 1 5 2 5 7  -3 8 -4 8 -6 8 -9 8 | dBm  dBm  dBm  dBm  dBm  dBm |

12 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Largest spurious  emission at maximum output power.  Texas Instruments CC2520 EM reference design  complies with EN 300 328, EN 300 440, FCC CFR47 Part 15 and ARIB STDT-66.  Transmit on 2480 MHz  under FCC at +5 dBm is supported by duty-cycling, or by reducing output  power.  The peak conducted  spurious emission might violate ETSI and FCC  restricted band limits at frequencies below 1GHz. All radiated spurious  emissions are within the limits of ETSI/FCC/ARIB. Applications that must pass conducted requirements are suggested to use a  simple 50 Ω high pass filter between matching network and RF connector. | 25 MHz – 1 GHz (outside restricted bands)  25 MHz – 1 GHz (within FCC restricted bands)  47-74, 87.5-118, 174-230, 470-862 MHz (ETSI restricted bands) 1800 MHz-1900 MHz (ETSI restricted band)  5150 MHz-5300 MHz (ETSI restricted band)  At 2483.5 MHz and above (FCC restricted band)  fc=2480 MHz, +5 dBm  fc=2480 MHz, 0 dBm  At 2·RF and 3·RF (FCC restricted band) | -40  -53  -42  -56  -54  -37  -41  -54 | dBm  dBm  dBm  dBm  dBm  dBm  dBm  dBm |
| Error Vector Magnitude (EVM) | [2] requires max. 35%. Measured as defined by [2].  +5 dBm setting. fc =IEEE 802.15.4 channels  0 dBm setting. fc =IEEE 802.15.4 channels | 6  2 | %  % |

**5.7 RSSI/CCA Parameters**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **COMMENTS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| RSSI range |  | 100 | dB |
| RSSI/CCA accuracy |  | +/-4 | dB |
| RSSI/CCA offset | Real RSSI = Register value - offset | 76 | dB |
| LSB value |  | 1 | dB |

**5.8 FREQEST Parameters**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **COMMENTS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| FREQEST range |  | +/-300 | kHz |
| FREQEST accuracy |  | +/-10 | kHz |
| FREQEST offset | Real frequency offset = FREQEST value - offset | 64 | kHz |
| LSB value |  | 7.8 | kHz |

**WWW.TI.COM** 13

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**5.9 Typical Performance Curves**

TA =25°C, VDD=3.0 V, fc =2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

)

m

B

d

(

Y

TI

VI

TI

S

N

ES

SENSITIVITY VS TEMPERATURE

-92

-94

)

m

B

d

(

Y

-96

T

I

V

I

T

I

S

N

E

-98

S

-100

SENSITIVITY VS EVM

-90.0

-92.0

-94.0

-96.0

-98.0

-100.0

)

m

B

d

(

Y

TI

VI

TI

S

NES

-40 10 60 110

TEMPERATURE (ºC)

SENSITIVITY VS SUPPLY VOLTAGE

-90

-92

)

m

B

-94

d

(

Y

T

I

V

I

-96

T

I

S

N

E

S

-98

-100

0 % 10 % 20 % 30 % 40 % 50 % 60 % ERROR VECTOR MAGNITUDE (% RMS)

SENSITIVITY VS CARRIER FREQUENCY OFFSET

0.0

-40.0

-80.0

-120.0

)

m

B

d

(

Y

TI

VI

TI

S

NE

S

1.8 2.3 2.8 3.3 3.8

VOLTAGE (V)

SENSITIVITY VS CARRIER FREQUENCY

-94

8

)

4

m

-96

B

d

(

R

E

0

W

O

P

-98

T

U

P

-4

T

U

O

-100

-8

-1000 -500 0 500 1000 FREQUENCY OFFSET (kHz)

OUTPUT POWER VS TEMPERATURE

5dBm (0xF7)

0dBm (0x32)

2394 2414 2434 2454 2474 2494 FREQUENCY (MHz)

-40 10 60 110 TEMPERATURE (ºC)

14 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

OUTPUT POWER VS SUPPLY VOLTAGE

6

1.9

AM CURRENT VS TEMPERATURE

)5dBm (0xF7) 4

m

B

d

(

R

E

) A

m

(

T

1.8 1.7

W

O

P

T

U

PT

UO

2

N

E

R

R

U

0dBm (0x32)

C

1.6

0

1.5

-2

1.8 2.3 2.8 3.3 3.8 VOLTAGE (V)

TX (5dBm setting, 0xF7) CURRENT VS TEMPERATURE

-40 10 60 110 TEMPERATURE (ºC)

LPM1 CURRENT VS TEMPERATURE

) A

m

(

T

NE

RR

UC

) A

m

(

T

NE

R

RU

C

400

35

300

)

34

A

u

(

T

200

N

E

R

R

U

33

C

100

0

32

-40 10 60 110

TEMPERATURE (ºC)

RX CURRENT VS TEMPERATURE

2

25

1.6

24

)

1.2

A

u

(

T

23

N

E

0.8

R

R

U

C

22

0.4

0

21

-40 10 60 110 TEMPERATURE (ºC)

LPM2 CURRENT VS TEMPERATURE

-40 10 60 110 TEMPERATURE (ºC)

-40 10 60 110 TEMPERATURE (ºC)

**WWW.TI.COM** 15

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

TX (+5dBm SETTING, 0xF7) CURRENT VS SUPPLY VOLTAGE

33.5

300

33

200

)

)

A

A

m

u

(

(

32.5

T

T

N

N

E

E

R

R

R

R

100

U

U

C

C

32

31.5

0

1.8 2.3 2.8 3.3 3.8

VOLTAGE (V)

RX CURRENT VS SUPPLY VOLTAGE

100

22.8

22.4

70

)

)

A

A

n

(

m

(

T

22

T

N

N

E

E

R

R

R

40

R

U

U

C

C

21.6

10

21.2

1.8 2.3 2.8 3.3 3.8

VOLTAGE (V)

AM CURRENT VS SUPPLY VOLTAGE

24

1.8

21

1.7

)

)

A

A

m

u

(

(

T

T

N

N

E

E

R

R

R

R

18

1.6

U

U

C

C

15

1.5

LPM1 CURRENT VS SUPPLY VOLTAGE

1.8 2.3 2.8 3.3 3.8 VOLTAGE (V)

LPM2 CURRENT VS SUPPLY VOLTAGE

1.8 2.3 2.8 3.3 3.8 VOLTAGE (V)

RX CURRENT VS INPUT LEVEL

1.8 2.3 2.8 3.3 3.8 VOLTAGE (V)

-100 -80 -60 -40 -20 0 INPUT LEVEL (dBm)

16 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

)

B

d

(

N

OI

T

C

EJ

E

R

RE

R

EF

R

ET

NI

INTERFERER REJECTION (802.15.4 INTERFERER) VS

INTERFERER FREQUENCY. CARRIER AT -82dBm/2440MHz.

75

80

)

B

d

(

60

50

N

O

I

T

C

E

J

40

25

E

R

R

E

R

E

20

0

F

R

E

T

N

I

0

-25

INTERFERER REJECTION VS 802.11g CARRIER AT -82dBm/2440MHz

)

B

d

(

R

CA

2400 2420 2440 2460 2480 INTERFERER FREQUENCY (MHz)

ADJACENT CHANNEL REJECTION (802.15.4 INTERFERER)

VS CARRIER LEVEL

55

50

)

B

d

(

N

45

O

I

T

C

E

J

40

E

R

R

E

35

R

E

F

R

E

T

30

N

I

-95 -90 -85 -80 -75 -70 -65 -60

CARRIER LEVEL (dBm)

INTERFERER REJECTION VS 802.11g

CARRIER AT -82dBm/2405MHz

2412 2422 2432 2442 2452 2462 2472 2482 INTERFERER FREQUENCY (MHz)

INTERFERER REJECTION VS 802.11g

CARRIER AT -82dBm/2480MHz

80

60

40

20

0

2412 2422 2432 2442 2452 2462 2472 2482 INTERFERER FREQUENCY (MHz)

FALSE PACKET RATE AND SENSITIVITY

80

1000

)

B

d

(

60

N

O

100

I

T

**.**

C

N

I

E

J

40

M

E

R

R

10

E

R

P

E

S

R

T

E

20

E

F

K

R

1

C

E

A

T

P

N

I

E

0

S

L

0.1

A

2412 2422 2432 2442 2452 2462 2472 2482

F

INTERFERER FREQUENCY (MHz)

0.01

vs CORRELATION THRESHOLD

False packets/min

Sensitivity

0x0B 0x0F 0x13 0x17 CORRELATION THRESHOLD (MDMCTRL1)

-91 -92 -93 -94 -95 -96 -97 -98

)

m

B

d

(

Y

TI

VI

TI

S

N

ES

**WWW.TI.COM** 17

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

) z

H

k

(

TS

E

QE

RF

FREQEST VS ACTUAL OFFSET FREQUENCY

500

400

)

300

V

(

R

O

200

S

N

E

100

S

E

R

0

U

T

A

R

-100

E

P

M

-200

E

T

-300

TEMPERATURE SENSOR OUTPUT VS SUPPLY VOLTAGE (TEMPERATURE = 25ºC)

0.820

0.810

0.800

0.790

0.780

-500 -300 -100 100 300 500

ACTUAL FREQUENCY OFFSET (kHz)

OFFSET CORRECTED RSSI VS INPUT LEVEL

112

0

)

l

)

a

108

-20

m

m

i

B

c

d

e

(

d

I

(

S

-40

E

104

S

U

R

L

D

A

E

V

-60

T

N

100

C

E

O

I

R

T

R

-80

A

O

L

E

C

96

R

T

R

E

-100

S

O

F

C

F

92

O

1.8 2.3 2.8 3.3 3.8 VOLTAGE (V)

CORRELATION VALUE VS ERROR VECTOR

MAGNITUDE OF INPUT SIGNAL

-120

-100 -80 -60 -40 -20 0 INPUT LEVEL (dBm)

TEMPERATURE SENSOR OUTPUT VS TEMPERATURE (SUPPLY VOLTAGE = 3V)

1.100

)

1.000

V

(

E

G

A

T

0.900

L

O

V

R

0.800

O

S

N

E

S

0.700

P

M

E

T

0.600

-40 10 60 110 TEMPERATURE (ºC)

0 % 10 % 20 % 30 % 40 % 50 % 60 % 70 % EVM (% RMS)

18 **WWW.TI.COM**

**5.10 Low-Current Mode RX**

Applications that spend more time waiting for an input

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

INTERFERER REJECTION (802.15.4 INTERFERER) VS

CARRIER LEVEL WHEN USING RX\_LOCUR

60

signal than actually receiving it, might benefit from using

the special low-current RX mode. This mode draws less

)

B

current at the expense of sensitivity.

d

(

N

40

O

I

T

Note that when using this mode, neither RSSI nor CCA

C

E

J

E

is valid. This means that these settings can not be used

R

R

in conjunction with STXONCCA, for instance. Also note

E

20

R

E

that the interferer rejection will drop at stronger input

F

R

signal levels compared to when using the regular

E

T

N

I

recommended settings.

0

-87 -78 -69 -60 -51

CARRIER LEVEL (dBm)

**Important: The low-current RX mode is only valid from -40 to 85ºC !**

**5.10.1 Low-Current RX Mode Parameters**

TA =25°C, VDD=3.0 V, fc=2440 MHz if nothing else stated. All parameters measured on Texas Instruments’ CC2520 EM 2.1 reference design with 50 Ω load.

| **PARAMETER** | **CONDITIONS** | **MIN** | **TYP** | **MAX** | **UNIT** |
| --- | --- | --- | --- | --- | --- |
| RX current | Wait for sync | 18.8 | | | mA |
| Sensitivity | [2] requires -85 dBm | -90 | | | dBm |
| Interferer Rejection | Wanted signal 3 dB above the sensitivity level, 802.15.4 modulated interferer at 802.15.4 channels:  ±5 MHz from wanted signal. [2] requires 0 dB  ±10 MHz from wanted signal. [2] requires 30 dB  ±20MHz or above. | 52  54  55 | | | dB  dB  dB |

**Table 1: Low-current RX mode. Use in addition to regular recommended settings.**

| **Register** | **Setting (hex)** | **Comment** |
| --- | --- | --- |
| RXCTRL | 33 | Reduces sensitivity and current consumption |
| FSCTRL | 12 | Reduces current consumption and valid temperature range |
| AGCCTRL2 | EB | Reduces sensitivity and current consumption |

**WWW.TI.COM** 19

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**5.11 Optional Temperature Compensation of TX**

Using the on-chip temperature sensor (or any other sensor), it is possible to adapt the settings to the actual temperature. This will reduce the variation in output power over temperature, which in the range -40ºC to 125ºC can be significant.

For this purpose, a TX setting only suited for high-temperature operation has been found (F7125deg). This setting should only be used above 70 degrees, but will significantly reduce the drop in output power at high temperatures.

**Table 2: F7125deg setting, only suited for high temperature operation (only changes from recommended settings shown)**

| **Register** | **Setting (hex)** | **Comment** |
| --- | --- | --- |
| TXCTRL | 94 | Increased output power at high temperatures. |
| FSCTRL | 7B | Increased output power at high temperatures. |

**Table 3: Suggested TXPOWER register settings for different temperatures**

| **Temperature** | **-40** | **-30** | **-20** | **-10** | **10** | **30** | **50** | **70** | **90** | **110** | **125** | **ºC** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Recommended Setting | 13 | 13 | AB | AB | F2 | F7 | F7 | F7125deg | F7125deg | F7125deg | F7125deg | - |
| Typical Output Power | 3.6 | 3.3 | 4.3 | 4.1 | 4.2 | 4.3 | 3.5 | 3.6 | 2.7 | 1.9 | 1.1 | dBm |

)

m

B

d

(

RE

W

O

P

T

U

PT

UO

8.0 4.0 0.0 -4.0

TYPICAL OUTPUT POWER WITH AND WITHOUT

TEMPERATURE COMPENSATION

)

m

B

d

(

R

With compensation

E

W

O

P

T

U

Without compensation

P

T

(+5dBm setting)

U

O

-40 10 60 110 TEMPERATURE (ºC)

8.0

4.0

0.0

-4.0

-8.0

-12.0

MINIMUM OUTPUT POWER WITH AND WITHOUT TEMPERATURE COMPENSATION

With

compensation

Without compensation

(+5dBm setting)

-40 10 60 110 TEMPERATURE (ºC)

20 **WWW.TI.COM**

**5.11.1 Using the Temperature Sensor**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

The on-chip temperature sensor can be accessed via the GPIO0 and GPIO1 pins by following this procedure: • Configure GPIO0 and GPIO1 as inputs by writing 0x80 to the GPIOCTRL0 and GPIOCTRL1 registers. • Enable analog output functionality for these two pins by setting GPIOCTRL.GPIO\_ACTRL=’1’. • Select temperature sensor output by writing 0x01 to the ATEST register. This will make GPIO1 output GND and GPIO0 will output a voltage proportional to the temperature.

• Use an ADC in the microcontroller to measure the output voltage on GPIO0 and then calculate the temperature.

The output from the temperature sensor is shown in graph form in section 5.9, but as a basis for calculating the temperature, the following numbers can be used:

Tc=-40 – 125°C, VDD=1.8 – 3.8 V

| **Parameter** | **Min** | **Typ** | **Max** | **Unit** |
| --- | --- | --- | --- | --- |
| Temp sensor voltage at 25°C |  | 0.8 |  | V |
| Temp. sens. output vs temperature |  | 25 |  | mV/10°C |
| Temp. sens. output vs supply voltage |  | 6 |  | mV/V |
| Temp. sens accuracy no calibration (at fixed voltage) |  | +/-12 |  | °C |
| Temp, sens. accuracy with 1-point calibration (at fixed voltage) |  | +/-1 |  | °C |

**WWW.TI.COM** 21

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**6 Crystal Specific Parameters**

**6.1 Crystal Requirements**

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Crystal frequency |  | 32 | MHz |
| Crystal frequency accuracy  requirement | Including initial tolerance, aging and  temperature dependency, as specified by [2]. Can be relaxed using on-chip crystal tuning (see below). | - 40 40 | ppm |
| ESR |  | 60 | Ohm |
| C0 |  | 7 | pF |
| CL |  | 16 | pF |

**6.2 On-chip Crystal Frequency Tuning**

| **PARAMETER** | **CONDITIONS** | **MIN TYP MAX** | **UNIT** |
| --- | --- | --- | --- |
| Crystal tuning range (Ctune) | Only adding capacitance is possible | 7 | pF |
| Crystal tuning step size |  | 0.4 | pF |
| Crystal tuning drift | In % of applied tuning | +/-10 | % |
| CRYSTAL TUNING USING CC2520 EM 2.1 REFERENCE DESIGN (NX3225DA, CL = 16 pF) : | | | |
| Start-up time | NDK crystal NX3225DA, CL=16 pF | 0.2 | ms |
| Crystal tuning step size | 3 | ppm |
| Crystal tuning range | -45 | ppm |
| CRYSTAL TUNING USING OTHER CRYSTALS, ALL NUMBERS ARE ESTIMATES : | | | |
| Start-up time | NDK crystal NX4025DA, CL=13 pF | 0.2 | ms |
| Crystal tuning step size | 8 | ppm |
| Crystal tuning range | -120 | ppm |
| Start-up time | NDK crystal NX5032SA, CL=10 pF | 0.1 | ms |
| Crystal tuning step size | 10 | ppm |
| Crystal tuning range | -160 | ppm |

See section 22 for further details on using the crystal oscillator.

22 **WWW.TI.COM**

**7 Pinout**

N

L

n

E

P

\_

T

U

E

G

K

O

S

L

E

E

C

C

R

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

D

R

A

U

G

4

\_

S

D

D

A

I

D

D

B

V

V

SO

SI

CSn

S

8

2

1

2

3

D

7 2

V

6 2

R

5 2

A

4 2

R

3 2

A

2

2

21

20

19

NC

AVDD1 RF\_N

GPIO5 GPIO4 GPIO3 GPIO2

4 5 6 7

CC2520

18 17 16 15

NC

RF\_P

AVDD2 NC

89

1

D

O

D

I

V

P

D

G

0 1

0

OI

PG

1 1

5 D

DV

A

2 1

2 Q

M23

CS

OX

\_

3 1

1 Q

M23

CS

OX

\_

4 1

3 D

DV

A

AGND

exposed die attached pad

**Figure 1: Pinout of CC2520 (top view)**

**Table 4: CC2520 Pinout**

| **Signal** | **Pin #** | **Type** | **Description** |
| --- | --- | --- | --- |
| **SPI** | | | |
| SCLK | 28 | I | SPI interface: Serial Clock. Maximum 8 MHz |
| SO | 1 | O | SPI interface: Serial Out |
| SI | 2 | I | SPI interface: Serial In |
| CSn | 3 | I | SPI interface: Chip Select, active low |
| **General Purpose digital I/O** | | | |
| GPIO0 | 10 | IO | General purpose digital I/O |
| GPIO1 | 9 | IO | General purpose digital I/O |
| GPIO2 | 7 | IO | General purpose digital I/O |
| GPIO3 | 6 | IO | General purpose digital I/O |
| GPIO4 | 5 | IO | General purpose digital I/O |
| GPIO5 | 4 | IO | General purpose digital I/O |
| **Misc** | | | |
| RESETn | 25 | I | External reset pin, active low |
| VREG\_EN | 26 | I | When high, digital voltage regulator is active. |
| NC | 15,  18, 21 |  | Not Connected. |

**WWW.TI.COM** 23

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **Signal** | **Pin #** | **Type** | **Description** |
| --- | --- | --- | --- |
| **Analog** | | | |
| RBIAS | 23 | Analog IO | External precision bias resistor for reference current. 56 kΩ, ±1% |
| RF\_N | 19 | RF IO | Negative RF input signal to LNA in receive mode  Negative RF output signal from PA in transmit mode |
| RF\_P | 17 | RF IO | Positive RF input signal to LNA in receive mode  Positive RF output signal from PA in transmit mode |
| XOSC32M\_Q1 | 13 | Analog IO | Crystal oscillator pin 1 |
| XOSC32M\_Q2 | 12 | Analog IO | Crystal oscillator pin 2 |
| **Power/ground** | | | |
| AVDD | 11,  14,  16,  20, 22 | Power  (Analog) | 1.8 V to 3.8 V analog power supply connections |
| AVDD\_GUARD | 24 | Power  (Analog) | Power supply connection for digital noise isolation and digital voltage regulator. |
| DCOUPL | 27 | Power  (Digital)  O | 1.6 V to 2.0 V digital power supply output for decoupling.  Note: this pin can not be used to supply any external devices. |
| DVDD | 8 | Power  (Digital) | 1.8 V to 3.8 V digital power supply for digital pads. |
| AGND | Die  pad | Ground  (Analog) |  |

24 **WWW.TI.COM**

**8 Functional Introduction**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**8.1 Integrated 2.4 GHz IEEE 802.15.4 Compliant Radio**

CC2520 features a Direct Conversion Transceiver operating in the 2.4 GHz band with excellent receiver sensitivity and robustness to interferers. The CC2520 radio complies with the IEEE 802.15.4 PHY specification. The radio has 250 kbps data rate, 2 Mchip/s chip rate, and is suitable for systems targeting compliance 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).

**8.2 Comparison to CC2420**

CC2520 represents significant improvement over the CC2420 features and performance. A comparison is given in the table below.

**Table 5: Comparison of CC2420 and CC2520**

| **Feature** | **CC2420** | **CC2520** |
| --- | --- | --- |
| Standard | IEEE 802.15.4-2003 | IEEE 802.15.4-2006 |
| Maximum output power | 0 dB | +5 dB |
| Typical sensitivity | -95 dBm | -98 dBm |
| General clock output | No | Yes, configurable frequency 1-16MHz |
| User interface | Command strobes and configuration registers. All user control goes through the SPI. | Instruction set (which includes the command strobes as a subset) and configuration registers. Command strobes may be triggered by GPIO pins, which gives excellent timing control. Improved status information. |
| Register access | Possible without crystal oscillator running. | Only possible when crystal oscillator is running. |
| Digital inputs | No Schmitt triggers | Schmitt triggers on all digital inputs. |
| Digital outputs | Fixed configuration | Highly flexible and configurable |
| Start up | Manual start of XOSC | XOSC starts automatically after reset (by reset\_n pin). Manual start of XOSC after SRES instruction. |
| Crystal frequency | 16 MHz | 32 MHz |
| Packet sniffing | No hardware support | Hardware support for non-intrusive sniffing of both transmitted and received frames. |
| Maximum SPI clock speed | 10 MHz | 8 MHz |
| RAM size | 364 byte | 768 byte |
| Operating voltage | 2.1 – 3.6 V | 1.8 – 3.8 V |
| Maximum operating temperature | 85°C | 125°C |
| Security | Limited flexibility | Highly flexible security instructions. More RAM available allows more flexible  processing. |
| Package | QLP-48, 7x7 mm | QFN 28 (RHD), 5x5 mm |
| RF frequency range | 2400-2483.5 MHz | 2394-2507 MHz |

**WWW.TI.COM** 25

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**8.3 Block Diagram** SO

K L

CS

L

P

U

O

C

D

Vreg

N E

GE

RV

n

\_

T

E

S

E

R

Clock/

reset

S

A

I

B

R

BIAS

SI

CSn

SPI

Frame

filtering and

source

matching

FSM Synthesizer

Instruction decoder

Exception

r

e

ll

o

r

tn

o

c

s

u

B

C

G

A

AES

DPU

Demod ADI

Modulator

RF\_core

ADI

GPIO5 GPIO4 GPIO3

controller IO

C 

D

A

RAM

C 

D

A

AAF

D

A

C

PS

D

A

C

LPF

GPIO2

t 

s

e

t

A

REF 

DIV XOSC

RX MIX LNA

FS

TX MIX PA

RF\_N RF\_P

1

OI

PG

0

OI

PG

2 Q

M2

3

CS

OX

\_

1 Q

M2

3

CS

OX

\_

**Figure 2: CC2520 block diagram**

CC2520 is typically controlled by a microcontroller connected to the **SPI** and some GPIOs. The microcontroller will send instructions to CC2520 and it is the responsibility of the **instruction decoder** to execute the instructions or pass them on to other modules.

The execution of an instruction or external events (e.g. reception of a frame) may result in one or more exceptions. The exceptions provide a very flexible mechanism for automating tasks. They can for instance be used to trigger execution of other instructions or they can be routed out to GPIO pins and used as interrupt signals to the microcontroller. The **exception controller** is responsible for handling of the exceptions.

26 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

The microcontroller will typically be connected to one or more of the GPIO pins. The function of each pin is independently controlled by the **IO** module based on register settings. It is possible to observe a large number of internal signals on the GPIO pins. The GPIO pins can also be configured as inputs and used to trigger the execution of certain instructions. This would typically be used when the microcontroller needs to precisely control the timing of an instruction.

The **RAM** module contains memory which is used for receive and transmit FIFOs (in fixed address ranges) and temporary storage for other data. There are separate instructions for general memory access and FIFO access.

The **data processing unit (DPU)** is responsible for execution of the more advanced instructions. The DPU includes an AES core, which is used while executing the security instructions. Memory management (copying, incrementing etc.) is also performed by the DPU.

The **Clock/Reset** module generates the internal clocks and reset signals.

The **RF core** contains several submodules that support and control the analog radio modules.

The **FSM** submodule controls the RF transceiver state, the transmitter and receiver FIFOs and most of the dynamically controlled analog signals such as power up / down of analog modules. The FSM is used to provide the correct sequencing of events (such as performing an FS calibration before enabling the receiver). Also, it provides step by step processing of incoming frames from the demodulator: reading the frame length, counting the number of bytes received, checks the FCS, and finally, optionally handles automatic transmission of ACK frames after successful frame reception. It performs similar tasks in TX including performing an optional CCA before transmission and automatically going to RX after the end of transmission to receive an ACK frame. Finally, the FSM controls the transfer of data between modulator/demodulator and the TXFIFO/RXFIFO in RAM.

The **modulator** transforms raw data into I/Q signals to the transmitter DAC. This is done in compliance with the IEEE 802.15.4 standard.

The **demodulator** is responsible for retrieving the sent data from the received signal.

The amplitude information from the demodulator is used by the **automatic gain control (AGC)**. The AGC adjusts the gain of the analog LNA so that the signal level within the receiver is approximately constant..

The **frame filtering and source matching** supports the FSM in RF\_core by performing all operations needed in order to do frame filtering and source address matching, as defined by IEEE 802.15.4.

The **xosc** module interfaces the crystal which is connected to the XOSC32M\_Q1 and XOSC32M\_Q2 pins. The xosc module generates a clock for the digital part and RF system, and implements the programmable crystal frequency tuning.

The **BIAS** module generates voltage and current references. It relies on a high precision (1%) 56kΩ external resistor which is shown in the application circuit in Figure 3.

The **TX DACs** convert the digital baseband signal to analog signals.

After LPF the signal is fed to the **TXMIX** module, which is an up-converting complex mixer. The **PA** amplifies the RF signal up to a maximum of ~5dBm during TX.

The **LNA** amplifies the received RF signal. The gain is controlled by the digital AGC module so that optimum sensitivity and interferer rejection is achieved.

The **RXMIX** module is a complex down-mixer that converts the RF signal to a baseband signal. A passive **anti-aliasing filter (AAF)** low pass filters the signal after down mixing.

The low pass filtered I and Q signals and digitized by the **ADC.**

**WWW.TI.COM** 27

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

The **frequency synthesizer (FS)** generates the carrier wave for the RF signal.

The **voltage regulator (Vreg)** provides a 1.8V supply voltage to the digital core. It contains a current limiter, which is enabled for currents above ~32mA.

28 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**9 Application Circuit**

Very few external components are required for the operation of CC2520. A typical application circuit is shown in Figure 4. Note that it does not show how the board layout should be done. The board layout will greatly influence the RF performance of CC2520.

This section is meant as an introduction only. For further details, see the reference design, which includes complete board layouts and bill of materials with manufacturer and part numbers. The reference design can be downloaded from the CC2520 product folder [7].

Note that decoupling capacitors are not shown in the figure below. See the reference design for complete bill of materials.



e

c

a

fr

e

t

n

i

l

a

ti

gi

D

8 2

K L

C

S

D 

DV

D

7 2

L P

U

OC

D

1

OI

PG

6 2

N E

GE

RV



0

OI

PG

\_

5 2

n T

E

SE

R



5 D

D

VA

4 2

D

RA

UG

D

DVA

2 Q

M23

CS

OX

\_

\_

3 2

S

AI

BR

1 Q

M23

CS

OX

\_

2 2

4 D

D

VA

3 D

D

VA

89

0 1

1 1

2 1

3 1

4 1



**Figure 3: Typical application circuit with transmission line balun for single-ended operation**

See the antenna selection guide [12] for further details on other compact and low-cost alternatives.

**9.1 Input / Output Matching**

The RF input/output is high impedance and differential.

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 only or in combination with transmission lines replacing the discrete inductors.

Figure 4 shows the balun implemented in a two-layer reference design. It consists of three transmission lines (L1, L2 and L3) and the discrete components C191, C171, C192, C173 and C174. The circuit will present the optimum RF termination to CC2520 with a 50Ω load on the antenna connection.

**WWW.TI.COM** 29

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

0

25

2

CC

1

9

1

C

1

7

1

C

C172

SMA 

connector

R201

2

9

1

C

C173 C174

PCB

antenna

**Figure 4: Actual board layout of the RF section of the reference design (rev 2.1).**

**9.2 Bias Resistor**

The bias resistor R231 is used to set an accurate bias current. A high precision (±1%) 56kΩ resistor should be used.

**9.3 Crystal**

An external 32MHz crystal with two loading capacitors (C121 and C131) is used for the crystal oscillator. It is possible to feed a single-ended signal to the XOSC32M\_Q1 pin and thus not use a crystal.

**9.4 Digital Voltage Regulator**

The on chip voltage regulator supplies 1.8 V to the digital part of CC2520. C271 is a decoupling capacitor for the voltage regulator. Note that this should not be used to provide power to other IC’s.

**9.5 Power Supply Decoupling and Filtering**

Proper power supply decoupling must be used for optimum performance. This is shown as a lumped capacitor C1 in Figure 4. The placement and size of the decoupling capacitors and the power supply filtering are very important to achieve the best performance in an application. TI provides a compact reference design that should be followed very closely.

**9.6 Board Layout Guidelines**

It is highly recommended to copy the board layout from the reference design [5].

• It is recommended to use star topology for the power supplies to CC2520.

• The power supply decoupling capacitor C1 is a lumped component. On the actual board layout there should be separate decoupling capacitors as close to each of the power pins as possible. • The balun is highly layout sensitive. The inductors in Figure 4 are actually transmission lines embedded in the PCB and their values must be adapted according to the board layout. The values of the capacitors C192, C172, C173 and C174 must also be adapted to the actual board layout. • The GPIO pins can be configured to use internal pull-up resistors. They are not enabled after a reset or in LPM2. Remember to take the default GPIO configuration into consideration when connecting these signals, because there will be some time before the MCU is able to change the configuration. In LPM2 GPIO5 (which is configured as an input) should be connected to either

30 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

ground or VDD. The other GPIO pins should be grounded or high impedance. Failing to do this, will result in significantly higher current consumption than necessary.

• The SO pin is configured as an input when CSn is high or the device is in reset or LPM2. This makes it possible to connect multiple SPI slaves to one SPI master. This pin should not be left floating when in LPM2, as this will draw more current than necessary. If the voltage level can not be controlled in any other way, use a 1MOhm pull-down resistor.

• The crystal input lines should be routed as far away from each other as practically possible. • The NC pins can be left floating.

• Glitches on the digital inputs may create serious issues in a system design. The digital input pads have Schmitt-triggers to help make them less sensitive to glitches, but the board layout should still avoid routing the digital input lines close to other noisy signals.

**9.7 Antenna Considerations**

The reference design contains two antenna options. As default, the SMA connector is connected to the balun through a 0Ω resistor. This resistor can be soldered off and rotated 90° clockwise in order to connect to the PCB antenna, which is a planar inverted F antenna (PIFA).

Note that all testing and characterization has been done using the SMA connector. The PCB antenna has only been functionally tested by establishing a link between two EMs. In our experiment, the PCB antenna gave approximately the same range as when using an antenna connected to the SMA connector.

Please refer to the antenna selection guide [12] and the Inverted F antenna app note [11] for further details.

**9.8 Choosing the Most Suitable Interconnection with a Microcontroller**

• Connect the 4 SPI signals; CSn, SCLK, SI and SO to the microcontroller.

These signals are required in order to configure CC2520 and exchange data with it. • Connect RESETn to the microcontroller. Using the RESETn signal is the recommended way to reset CC2520 for instance after powering up. If saving a pin is critical, the RESETn pin can be connected to VDD. The CC2520 can still be reset with the SRES command strobe. This will also require a manual start of the crystal oscillator by issuing a SXOSCON command strobe. • Connecting VREG\_EN to the microcontroller will make it possible to put CC2520 into LPM2 to save power. VREG\_EN may be connected to VDD and thus always leave the regulator on. If power saving is not important in the target application, this may be an acceptable way of saving a pin. • Connecting one or more of the GPIOs to the microcontroller is optional.

The number of GPIOs to connect depends on the application. Connecting more GPIOs to the microcontroller generally gives more flexibility and less SPI traffic because it reduces the need to keep reconfiguring the GPIOs for different uses.

• If CC2520 will be providing clock to the microcontroller, GPIO0 should be connected to the clock input of the microcontroller. After reset, GPIO0 will output a 1MHz clock signal with 50/50 duty cycle.

The digital IO of CC2520 is described in more detail from section 12.

**9.9 Interfacing CC2520 and MSP430F2618**

The MSP430F2618 is well suited for use with the CC2520. The suggested interfacing of these two chips is given in Table 5. The interconnections shown in Table 6 are exactly the same as is used in the CC2520 development kit [5].

**WWW.TI.COM** 31

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**Table 6: Interconnection of MSP430F2618 and CC2520**

| **CC2520** | **MSP430F2618** |
| --- | --- |
| VREG\_EN | P01.0/TACLK/CAOUT |
| RESETn | P05.7/TBOUTH/SVSOUT |
| SCLK | P05.3/UCB1CLK/UCA1STE |
| SO | P05.2/UCB1SOMI/UCB1SCL |
| SI | P05.1/UCB1SIMO/UCB1SDA |
| CSn | P05.0/UCB1STE/UCA1CLK |
| GPIO0 | P01.3/TA2 |
| GPIO1 | P01.5/TA0 |
| GPIO2 | P01.6/TA1 |
| GPIO3 | P01.1/TA0/BSLTX |
| GPIO4 | P01.2/TA1 |
| GPIO5 | P01.7/TA2 |

A simplified drawing of the interconnection of MSP430F2618 and CC2520 is shown in Figure 8. For further details on the MSP430F2618, please refer to [10].

RESETn

VREG\_EN

4

SPI

6

GPIO

**Figure 5: Interconnection of MSP430F2618 and CC2520**

32 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**10 Serial Peripheral Interface (SPI)**

The SPI provides an interface for giving instructions to the CC2520 and transferring data between CC2520 and a microcontroller. The CC2520 4-wire slave interface consists of three input signals (CSn, SCLK and SI) and one output signal (SO).

In section 15 all instructions available via the SPI interface are listed and described. The instructions are byte oriented and required bytes sent over the interface to CC2520 vary from 1 and up. To transfer one byte CSn must be pulled low and SCLK must complete 8 periods starting with a positive edge. There are no requirements to maximum period for SCLK or that it needs to be continuous. As long as CSn is held low, SCLK can be halted at any time and started again when desired.

**10.1 CSn**

CSn is an input enable signal for the SPI and is controlled by the external MCU. The CSn signal is used as an asynchronous active high reset to the SPI module.

CSn must be held low during all SPI operations and must also be held low for more than two periods of XOSC before the first positive edge of SCLK and more than two periods of XOSC after the last negative edge of SCLK.

When CSn is high it must be held high for at least 2 periods of XOSC.

CSn can be held low between SPI operations in the case where the last instruction completed has a constant number of bytes, but this will result in unnecessary power consumption since parts of the instruction controller will then be running.

The instructions that have a constant number of bytes can be found in the instruction summary table in section 15.3. I.e. SRXON (1 byte) and RXMASKAND (3 bytes) has constant number of bytes and REGRD (2 bytes or more) has user controlled number of bytes indicated in the table by three dots (…) in the byte column after the last required byte of the instruction command (Byte 3 for REGRD).

Instructions that have user controlled number of bytes are ended by rising CSn.

Status is output as the first byte on SO during the first byte of all instructions. When instructions are transferred consecutively without rising CSn between them, the status byte on SO may not contain the correct current status. However, the status will be updated for the second byte of an instruction so i.e RXMASKAND which outputs status also during the second instruction byte will then output the correct status during the second byte.

When pulling CSn low after power-up, SO outputs the internal XOSC stable signal combinatorically, so no edge on SCLK is necessary to find the XOSC stable status. In any case where CSn is pulled low and SO is low it means that XOSC is still not stable and thus there is no clock in the digital part. The maximum time from power up to XOSC should be stable is described in section 5.3.

**10.2 SCLK**

SCLK is controlled by an external MCU and is an input clock to CC2520. SCLK is asynchronous to the internal XOSC clock in CC2520. The maximum SCLK frequency is 8 MHz. There is no minimum frequency requirement.

**10.3 SI**

SI is the serial data input from the microcontroller to CC2520. Data shall be sent with MSB first (bit 7 in each byte of instruction commands).

Data should be set up on the negative edge of SCLK and will be clocked into CC2520 by the next positive edge of SCLK.

**WWW.TI.COM** 33

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**10.4 SO**

SO is serial data out from CC2520 to an external MCU. Data is clocked out on the negative edge of SCLK, so the SO signal should be sampled on the following rising edge of SCLK. MSB (bit 7 in register definitions) will be clocked out first.

SO is configured as an input when CSn is high or RESETn is low. Note that the SO pin should not be left floating while in LPM1 or LPM2, as this will result in higher current consumption than necessary.

**10.5 SPI Timing Requirements**

SCLK I CSn I

**tcsckh**

tcsnh tcscks

**tsis**

tsclk

tsih

**tsclkh**

tsclkl

SI I SO O

1 0 7 6 5 4 tsod

0 7 7 6 5 4 **Figure 6: SPI timing relationships**

The following table and figure shows required timing relations between an external microcontroller and the SPI interface on CC2520.

**Table 7: SPI timing requirements**

| **PARAMETER** | **DESCRIPTION** | **MIN** | **TYP** | **MAX** | **UNIT** |
| --- | --- | --- | --- | --- | --- |
| tcscks | CSn to SCLK setup time | 62.5 |  |  | ns |
| tcsckh | SCLK to CSn hold time | 62.5 |  |  | ns |
| tcsnh | CSn high | 62.5 |  |  | ns |
| tsclk | SCLK period | 125 |  |  | ns |
| tsclkh | SCLK high time | 62.5 |  |  | ns |
| tsclkl | SCLK low time | 62.5 |  |  | ns |
| tsis | SI to SCLK setup time | 31 |  |  | ns |
| tsih | SI to SCLK hold time | 31 |  |  | ns |
| tsod | SCLK to SO delay |  |  | 31 | ns |

34 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**11 GPIO**

CC2520 has 6 GPIO pins that can be individually configured as inputs, outputs and activate pull-up resistors. Each GPIO has an associated register, GPIOCTRLn, where the MSB configure the pin to either input or output. The GPIOCTRL register control pull-up for each individual GPIO pin, extra drive strength for all pins and analog function for pin 0 and 1. See section 30 for details about test functionality and observability through GPIO.

Note that GPIO5, which is configured as an input in LPM2, should be tied either to ground or VDD when entering LPM2. If GPIO5 (or any other input) is left floating, the current consumption will be unpredictable.

**11.1 Reset Configuration of GPIO Pins**

The reset setting for GPIO pins are as shown in the table below. This is also the configuration that is used when the device is in LPM2. If a different GPIO setup is required, the GPIOs have to be re-configured every time CC2520 has been in LPM2.

This particular reset configuration was selected so that CC2520 looks as much like CC2420 as possible.

**Table 8: GPIO reset state**

| **GPIO pin** | **Dir** | **Value** | **Pull**  **up** | **Extra**  **drive** | **Polarity** | **Signal** | **GPIOCTRLn**  **value (hex)** | **Description** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Out | 0 | No | No | Positive | clock | 0x00 | 1MHz clock signal with 50/50 duty cycle. |
| 1 | Out | 0 | No | No | Positive | fifo | 0x27 | High when one or more bytes are in the RX FIFO. Low during RX FIFO overflow. |
| 2 | Out | 0 | No | No | Positive | fifop | 0x28 | High when the number of bytes in the RX FIFO exceeds the programmable threshold or at least one complete frame is in the RX FIFO. Also high during RX FIFO overflow. |
| 3 | Out | 0 | No | No | Positive | cca | 0x29 | Clear channel assessment. See FSMSTAT1 register for details on how to configure the behavior of this signal. |
| 4 | Out | 0 | No | No | Positive | sfd | 0x2A | Pin is high when SFD has been received or transmitted. Cleared when leaving RX/TX respectively. |
| 5 | In | Tie to  ground or VDD | No | No | Positive |  | 0x90 | No function |

**11.2 GPIO as Input**

When configured as input, the GPIO pin can be used to trigger one of 16 different command strobes (See section 15) as shown in the GPIO configuration table in section 12.6. These command strobes are a subset of all the SPI instructions available. The command strobe is triggered by applying a rising or falling edge to the GPIO pin depending on the setting in the GPIOPOLARITY register. Which command strobe the pin triggers is set by the 7 LSBs in GPIOCTRLn.

**Example:** Set up GPIO2 to run SACK instruction on rising edge.

• Set GPIOPOLARITY[2] to ‘1’. GPIO pin 2 set to rising edge active.

• Set GPOICTRL2[7:0] to “1000 0101” . GPIO pin 2 is now an input and connected to the SACK instruction.

**WWW.TI.COM** 35

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**11.3 GPIO as Output**

When a GPIO pin is configured as an output, the signal corresponding to the CTRLn setting in GPIOCTRLn register (CTRLn values are shown in Table 8 in section 12.6). The polarity of the pin is set in the GPIOPOLARITY register.

**Example**: Set up GPIO3 to output sniff\_data with active high level indication.

• Set GPIOPOLARITY[3] to ‘1’’. GPIO pin 3 set to active high level indication.

• Set GPIOCTRL3[7:0] to “0011 0010”. GPIO pin 3 is now an output and outputs sniff\_data.

**11.4 Switching Direction on GPIO**

When switching from output to input, care must be taken so that command strobes are not triggered unintentionally. Changing GPIOn to a command strobe triggering input (one of the first 16 entries in Table 8) needs to be done using the following procedure to avoid changing direction while the pin is high:

1. Write 0x7E to GPIOCTRLn to make it output a constant 0.

2. Drive a ‘0’ from the microcontroller to the GPIO pin.

3. Write for instance 0x88 to GPIOCTRLn to change to input that triggers the STXON command strobe.

**11.5 GPIO Configuration**

Table 8 summarizes the signals that are available as output on any GPIO pin. The CTRLn column shows the configuration value that needs to be written to any one of the GPIOCTRL0-GPIOCTRL5 registers in order to get the described functionality. The IN column in Table 8 shows which command strobe that will be executed if the GPIO is configured as input and an edge (with the correct polarity) is applied. The OUT column shows the name of the internal signal that is observable on the pin if the GPIO is configured as an output.

36 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**Table 9: GPIO configuration**

| **CTRLn**  **(hex)** | **IN**  **(Command strobes)** | **OUT** | **Description of OUT signal** |
| --- | --- | --- | --- |
| 0x00 | SIBUFEX | Clock | Clock signal. Programmable frequency from 1MHz to 16MHz |
| 0x01 | SRXMASKBITCLR | RF\_IDLE | RF\_IDLE exception. See Table 14: Exceptions summary for details. |
| 0x02 | SRXMASKBITSET | TX\_FRM\_DONE | TX\_FRM\_DONE exception. See Table 14: Exceptions summary for details. |
| 0x03 | SRXON | TX\_ACK\_DONE | TX\_ACK\_DONE exception. See Table 14: Exceptions summary for details. |
| 0x04 | SSAMPLECCA | TX\_UNDERFLOW | TX\_UNDERFLOW exception. See Table 14: Exceptions summary for details. |
| 0x05 | SACK | TX\_OVERFLOW | TX\_OVERFLOW exception. See Table 14: Exceptions summary for details. |
| 0x06 | SACKPEND | RX\_UNDERFLOW | RX\_UNDERFLOW exception. See Table 14: Exceptions summary for details. |
| 0x07 | SNACK | RX\_OVERFLOW | RX\_OVERFLOW exception. See Table 14: Exceptions summary for details. |
| 0x08 | STXON | RXENABLE\_ZERO | RXENABLE\_ZERO exception. See Table 14: Exceptions summary for details. |
| 0x09 | STXONCCA | RX\_FRM\_DONE | RX\_FRM\_DONE exception. See Table 14: Exceptions summary for details. |
| 0x0A | SFLUSHRX | RX\_FRM\_ACCEPTED | RX\_FRM\_ACCEPTED exception. See Table 14: Exceptions summary for details. |
| 0x0B | SFLUSHTX | SRC\_MATCH\_DONE | SRC\_MATCH\_DONE exception. See Table 14: Exceptions summary for details. |
| 0x0C | SRXFIFOPOP | SRC\_MATCH\_FOUND | SRC\_MATCH\_FOUND exception. See Table 14: Exceptions summary for details. |
| 0x0D | STXCAL | FIFOP | FIFOP exception. See Table 14: Exceptions summary for details. |
| 0x0E | SRFOFF | SFD | SFD exception. See Table 14: Exceptions summary for details. |
| 0x0F | SXOSCOFF | DPU\_DONE\_L | DPU\_DONE\_L exception. See Table 14: Exceptions summary for details. |
| 0x10 |  | DPU\_DONE\_H | DPU\_DONE\_H exception. See Table 14: Exceptions summary for details. |
| 0x11 |  | MEMADDR\_ERROR | MEMADDR\_ERROR exception. See Table 14: Exceptions summary for details. |
| 0x12 |  | USAGE\_ERROR | USAGE\_ERROR exception. See Table 14: Exceptions summary for details. |
| 0x13 |  | OPERAND\_ERROR | OPERAND\_ERROR exception. See Table 14: Exceptions summary for details. |
| 0x14 |  | SPI\_ERROR | SPI\_ERROR exception. See Table 14: Exceptions summary for details. |
| 0x15 |  | RF\_NO\_LOCK | RF\_NO\_LOCK exception. See Table 14: Exceptions summary for details. |
| 0x16 |  | RX\_FRM\_ABORTED | RX\_FRM\_ABORTED exception. See Table 14: Exceptions summary for details. |
| 0x17 |  | RXBUFMOV\_TIMEOUT | RXBUFMOV\_TIMEOUT exception. See Table 14: Exceptions summary for details. |
| 0x18 |  | UNUSED | UNUSED exception. See Table 14: Exceptions summary for details. |
| ... |  | Reserved |  |
| 0x21 |  | Exception channel A | Pin is high when one or more of the exception flags in collection A are active. It is configurable which exceptions to include in collection A. |

**WWW.TI.COM** 37

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **CTRLn**  **(hex)** | **IN**  **(Command strobes)** | **OUT** | **Description of OUT signal** |
| --- | --- | --- | --- |
| 0x22 |  | Exception channel B | Pin is high when one or more of the exception flags in collection B are active. It is configurable which exceptions to include in collection B. |
| 0x23 |  | Complementary exception channel A | Pin is high when one or more exception flags not in collection A are active. |
| 0x24 |  | Complementary exception channel B | Pin is high when one or more exception flags not in collection B are active. |
| 0x25 |  | Predefined exception channel for RX related errors. | Predefined exception channel. High when one or more of the following exception flags are active: RX\_UNDERFLOW, RX\_OVERFLOW, RX\_FRM\_ABORTED and  RXBUFMOV\_TIMEOUT |
| 0x26 |  | Predefined exception channel for general error conditions. | High when one or more of the following exception flags are active: MEMADDR\_ERROR, USAGE\_ERROR, OPERAND\_ERROR and SPI\_ERROR. |
| 0x27 |  | fifo | Pin is high when one or more bytes are in the RXFIFO. Low during RXFIFO overflow. |
| 0x28 |  | fifop | Pin is high when the number of bytes in the RXFIFO exceeds the programmable threshold or at least one complete frame is in the RXFIFO. Also high during RXFIFO overflow. Not to be confused with the FIFOP exception. |
| 0x29 |  | cca | Clear channel assessment. See FSMSTAT1 register for details on how to configure the behavior of this signal. |
| 0x2A |  | sfd | Pin is high when a SFD has been received or transmitted. Cleared when leaving RX/TX respectively. Not to be confused with the SFD exception. |
| 0x2B |  | lock | Pin is high when frequency synthesizer is in lock. |
| 0x2C |  | rssi\_valid | Pin is high when the RSSI value has been updated at least once since RX was started. Cleared when leaving RX. |
| 0x2D |  | sampled\_cca | A sampled version of the CCA bit from demodulator. The value is updated whenever a SSAMPLECCA or STXONCCA strobe is issued. |
| 0x2E |  | rand\_i | Random data output from the I channel of the receiver. Updated at 8MHz. |
| 0x2F |  | rand\_q | Random data output from the Q channel of the receiver. Updated at 8MHz |
| 0x30 |  | rand\_xor\_ i\_q | XOR between I and Q random outputs. Updated at 8MHz |
| 0x31 |  | sniff\_clk | 250kHz clock for packet sniffer data. |
| 0x32 |  | sniff\_data | Data from packet sniffer. Sample data on rising edges of sniff\_clk. |
| 0x33 |  | mod\_serial\_clk | 250kHz serial data clock from modulator. |
| 0x34 |  | mod\_serial\_data | Serial data from modulator. Sample data on rising edges of mod\_serial\_clk. |
| ... |  | Reserved |  |
| 0x43 |  | rx\_active | Indicates that FFCTRL is in one of the RX states. Active high.  Note: This signal might have glitches, because it has no output flip-flop and is based on the current state register of the FFCTRL FSM. |
| 0x44 |  | tx\_active | Indicates that FFCTRL is in one of the TX states. Active high.  Note: This signal might have glitches, because it has no output flip-flop and is based on the current state register of the FFCTRL FSM. |
| ... |  | Reserved |  |
| 0x5E |  | dpu\_core\_activepri(0) | High when the DPU is busy processing a low priority thread. |
| 0x5F |  | dpu\_core\_activepri(1) | High when the DPU is busy processing a high priority thread. |

38 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **CTRLn**  **(hex)** | **IN**  **(Command strobes)** | **OUT** | **Description of OUT signal** |
| --- | --- | --- | --- |
| ... |  | Reserved |  |
| 0x62 |  | dpu\_state\_l\_active | High when low priority thread is pending or active. |
| 0x63 |  | dpu\_state\_h\_active | High when high priority thread is pending or active. |
| … |  | Reserved |  |
| 0x7E |  | ‘0’ | Constant value |
| 0x7F |  | ‘1’ | Constant value |

**WWW.TI.COM** 39

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**12 Power Modes**

CC2520 has three power modes as described below. In all these power modes the supply voltage is applied to the circuit.

In **Low Power Mode 2** (LPM2) the digital voltage regulator is turned off (VREG\_EN=0) and no clocks are running. No data is retained. All analog modules are in power down state.

In **Low Power Mode 1** (LPM1) the digital voltage regulator is on (VREG\_EN=1), but no clocks are running. Data is retained. The power down signals to the analog modules are controlled by the digital part.

In **Active mode** the digital voltage regulator is on (VREG\_EN=1) and the crystal oscillator clock is running. The power down signals to the analog modules are controlled by the digital part.

**12.1 Switching Between Power Modes**

When the device has been in LPM2, all register content is lost. To bring the device up to active mode, a reset is required or the device will be in an unknown state. The reset can be applied either by setting the RESETn pin low, or issuing a reset instruction (SRES) over the SPI. It is recommended that the RESETn method is used, because it will give a controlled start and automatic start of the crystal oscillator.

Before entering LPM2, it is strongly recommended that the device is reset. This way, the configuration will always be the same when the power to the digital part is removed, and it is less likely that there will be issues with current spikes or other side effects of the power being removed.

**LPM2**

Set VREG\_EN=0

Set GPIO5=0

Set RESETn=0

Set CSn=1

SRES

Set RESETn=1 Set VREG\_EN=1

Set RESETn=0 Set VREG\_EN=1

Wait until regulator has stabilized. Use a timeout.

Set RESETn=1 Set CSn=0

Wait until regulator has stabilized. Use a timeout.

SRES

SXOSCON

SNOP

Set CSn=0

**LPM1**

SXOSCON

SNOP

SXOSCOFF

(Radio must be idle)

Set CSn=0 and wait until SO=1

Wait until SO=1

Set CSn=1

**Active mode**

**Active mode**

Set CSn=1

**Figure 7: Procedures for switching between power modes.**

40 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**12.2 Power Up Sequence Using RESETn (recommended)**

When the RESETn pin is used it must be held low until the internal regulator has stabilized. This typically takes 0.1 ms. When the RESETn pin is set high, the crystal oscillator (in the CC2520 reference design) uses typically 0.2 ms to start. See section 6 for crystal specific parameters.

The GPIO pins are configured according to Table 8: GPIO reset state when power is applied to the chip and RESETn is held low.

VDD I

VREG\_EN I

Tdres

RESETn I

CSn O

SCLK I

SI I

GPIO [5..0] IO

Txr

Internal XOSC O

SO O

**12.3 Power Up With SRES**

XOSC stable and running

**Figure 8: Power up sequence using RESETn**

If one prefers to use the SRES command strobe to reset the device after powering up, the CSn signal must be set low and SRES must be issued after the internal regulator has stabilized. Until the SRES command strobe has been issued, the chip will be in an unknown state. Note that this means it could theoretically for instance be transmitting.

The time from power is applied to the XOSC has started depends on the clock frequency used on the SPI (max 8MHz) and the startup time for the crystal.

Note that the crystal oscillator does not necessarily start automatically when the SRES command strobe is issued. That means one also has to issue an SXOSCON command strobe to be sure that the oscillator starts. Unlike the RESETn pin, the SRES command strobe will not influence the state of the crystal oscillator, so if the oscillator accidentally comes up in the “off” state, issuing a SRES will not make it start.

VDD I

VREG\_EN I

RESETn I

Tdres

CSn O

SCLK I

SI I GPIO [5..0] IO

Internal XOSC O SO O

SRES B0 SRES B1 SXOSCON SNOP

Txr

XOSC stable and running

STATUS STATUS STATUS STATUS

**Figure 9: Power up sequence using SRES**

**WWW.TI.COM** 41

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**Table 10: Start-up Timing**

| **Name** | **Description** | **Time** |
| --- | --- | --- |
| Tdres | Time required after VREG\_EN is activated until RESETn is released or CSn is set low. | ≥ 0.1 ms |
| Txr | Time for internal XOSC to stabilize after RESETn is released or SXOSCON strobe is issued. | 0.2 ms (crystal dependent) |

42 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**13 Instruction Set**

The CC2520 has a comprehensive instruction set. The instructions are transferred to CC2520 via the SPI, and can consist of one or more bytes. The first byte contains the unique op-code and the following bytes are parameters needed to execute the selected instruction. In the following sections, every instruction and parameter is described in detail.

**13.1 Definitions**

• All parameters and data are transferred over the SPI with their most significant bit first and their least significant bit last.

• For instructions that read data from CC2520, the data byte will replace the status byte on the SO pin.

• Address parameters point to the least significant byte in a block of data. The address A+1 contains the next but least significant byte and so on.

• When CC2520 automatically increments addresses, it will wrap around when incrementing beyond the highest possible address (0xFFF).

• An instruction is ended by either sending the complete instruction (for finite instructions) or raising CSn (For infinite instructions, indicated by “...” in the instructions summary).

• Once an instruction is ended a new instruction can be started.

• If an instruction is ended before it is complete or if the instruction is not recognized, an OPERAND\_ERROR exception is raised.

• If the user sets parameter bits explicitly marked as ‘0’ in instruction summary table to ‘1’ an OPERAND\_ERROR exception is raised.

• When an instruction is aborted an error exception is raised and the SPI interface ceases to receive further data until CSn has been set high then low again. The instruction that was aborted may have made changes to memory contents before it was aborted.

• If the SPI interface is reset (by pulling CSn high) in the middle of an SPI byte transfer (i.e. not between bytes) an SPI\_ERROR exception is raised.

**13.2 Instruction Descriptions**

The codes shown below are used in the descriptions of the instructions. They represent bits selectable by the user. A sequence of bits thus represented by the same letter, even when spanning multiple bytes represents a word with a width equal to the number of repeated letters and with MSB the leftmost bit in the first byte transferred with this encoding. Such words may be represented in the text as a capital letter of the encoding letter in which case they shall be interpreted as a positive integer encoded by the bits represented in the encoding by the same letter only in lower-case.

Note that the bits that refer to one such integer need not be continuous in the encoding. So the encoding aaaaeeee aaaaaaaa eeeeeeee represents two 12 bit words transferred in three bytes with the most significant bits of each word transferred in the first byte.

**Table 11: Codes used in instruction set description**

| **Code** | **Description** |
| --- | --- |
| a, e, k, n | Address data |
| b | Bit address |
| i | Instruction |
| d | Data |
| s | Status byte |
| p | Priority |
| m | Security parameter |
| c, f | Count |
| - | Don’t care |

**WWW.TI.COM** 43

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**Table 12: CC2520 instruction set**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| **Peripheral instructions** | | | | |
| IBUFLD | i[7:0] | s[7:0] | Load instruction into instruction buffer. The instruction buffer holds a single instruction 1 byte long. The instruction to be loaded, I, is held and shall be parsed as a normal instruction when SIBUFEX is executed as if those bytes had just been transferred to the SPI interface.  Once the instruction held in the instruction buffer is executed it is replaced by SNOP. |  |
| SIBUFEX  Command strobe |  | s[7:0] | Execute the instruction stored in the instruction buffer as though those bytes had been transferred on the SPI interface.  A USAGE\_ERROR exception is raised if the instructions stored are not valid for use with the instruction buffer.  The executed instruction may raise any exception it normally can. | USAGE\_ERROR  Special. |
| SSAMPLECCA  Command strobe |  | s[7:0] | Sample the value of the CCA status signal, and store in status register. |  |
| SNOP |  | s[7:0] | No Operation (has no other effect than reading out status-bits) |  |
| SXOSCON |  | s[7:0] | Turn on the crystal oscillator. If this instruction is executed when the XOSC is already on, the instruction has no effect.  This instruction can only be run as the first instruction after CSn has been pulled low.  Must be immediately followed by a SNOP instruction in order to terminate properly. | OPERAND\_ERROR |
| STXCAL  Command strobe |  | s[7:0] | Enable and calibrate frequency synthesizer for TX; Go from RX / TX to a wait state where only the synthesizer is running. For test purposes only.  If a frame is currently being received a  RX\_FRM\_ABORTED exception is raised. | RX\_FRM\_ABORTED |
| SRXON  Command strobe |  | s[7:0] | Enable RX.  If a frame is currently being received a  RX\_FRM\_ABORTED exception is raised. | RX\_FRM\_ABORTED |
| STXON  Command strobe |  | s[7:0] | Enable TX after calibration (if not already performed)  If a frame is currently being received a  RX\_FRM\_ABORTED exception is raised. | RX\_FRM\_ABORTED |
| STXONCCA  Command strobe |  | s[7:0] | If CCA indicates a clear channel:  Enable calibration, then TX.  else  do nothing  Also sample the value of the CCA status signal, and store in status register. |  |
| SRFOFF  Command strobe |  | s[7:0] | Disable RX/TX and frequency synthesizer.  If RX, TX and frequency synthesizer is already off a USAGE\_ERROR exception is raised and the instruction has no effect.  If a frame is currently being received a  RX\_FRM\_ABORTED exception is raised. | USAGE\_ERROR  RX\_FRM\_ABORTED |

44 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| SXOSCOFF  Command strobe |  | s[7:0] | Turn off the crystal oscillator.  If the RF section is not idle a USAGE\_ERROR is generated.  If a frame is currently being received a  RX\_FRM\_ABORTED exception is raised. | USAGE\_ERROR  RX\_FRM\_ABORTED |
| SFLUSHRX  Command strobe |  | s[7:0] | Flush the RX FIFO and reset the demodulator.  If a frame is currently being received a  RX\_FRM\_ABORTED exception is raised. | RX\_FRM\_ABORTED |
| SFLUSHTX  Command strobe |  | s[7:0] | Flush the TX FIFO |  |
| SACK  Command strobe |  | s[7:0] | Send acknowledgement frame, with the frame pendig subfield cleared, following reception of the current frame.  Raises USAGE\_ERROR exception if a frame is currently not being received. In this case no ACK frame is sent. | USAGE\_ERROR |
| SACKPEND  Command strobe |  | s[7:0] | Send acknowledgement frame, with the frame pendig subfield set, following reception of the current frame.  Raises USAGE\_ERROR exception if a frame is currently not being received. In this case no ACK frame is sent. | USAGE\_ERROR |
| SNACK  Command strobe |  | s[7:0] | Do not send an acknowledgement frame to the currently received frame, even if the rfr\_autoack is set.  Raises USAGE\_ERROR exception if a frame is currently not being received. In this case no ACK frame is sent. | USAGE\_ERROR |
| SRXMASKBITSET Command strobe |  | s[7:0] | Set bit 13 in the RXMASK. |  |
| SRXMASKBITCLR Command strobe |  | s[7:0] | Clear bit 13 in the RXMASK.  Raises RXENABLE\_ZERO exception if this causes the RXENABE registers to be zero. | RXENABLE\_ZERO |
| RXMASKOR | d[15:0] | s[7:0] | Perform bitwise OR between RX enable mask and D. |  |
| RXMASKAND | d[15:0] | s[7:0] | Perform bitwise AND between RX enable mask and D. Raises RXENABLE\_ZERO exception if this causes the RXENABLE registers to be zero. | RXENABLE\_ZERO |
| **Data IO** | | | | |
| BSET | a[7:3]  b[2:0] | s[7:0] | Set a single bit. Writes 1 to bit B in address A. This is done without affecting the value of, or triggering side effects of other bits at the same address. Only the address range [0, 31] is accessible with this instruction. | MEMADDR\_ERROR |
| BCLR | a[7:3]  b[2:0] | s[7:0] | Clear a single bit. Writes 0 to bit B in address A. This is done without affecting the value of, or triggering side effects of other bits at the same address. Only the address range [0, 31] is accessible with this instruction. | MEMADDR\_ERROR |
| MEMRD | a[11:0] | s[7:0]  d[7:0]  ... | Read memory. The n’th byte of data D is read from address (A+n). Note that when an address with LSB=0 is read the content of the corresponding address with LSB=1 is buffered. If that address is read immediately after within the same MEMRD instruction, the buffered copy is read. In this way a read of a complete 16 bit word is performed as an atomic operation. | MEMADDR\_ERROR |
| MEMWR | a[11:0] d[7:0] ... | s[7:0]  d[7:0]  ... | Write memory. The n’th byte of data D input with the instruction is written to address (A+n).  In addition, the n’th byte of data D output from the instruction is the unaltered data read from the memory location (A+n). | MEMADDR\_ERROR |
| REGRD | a[5:0] | s[7:0]  d[7:0]  ... | Same functionality as MEMRD, except the operation can only be started from addresses below 0x40. | MEMADDR\_ERROR |

**WWW.TI.COM** 45

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| REGWR | a[5:0]  d[7:0] ... | s[7:0]  d[7:0]  ... | Same functionality as MEMWR, except the operation can only be started from addresses below 0x40. | MEMADDR\_ERROR |
| MEMXWR | a[11:0] d[7:0] ... | s[7:0]  d[7:0]  ... | XOR memory. Writes the bitwise XOR of the n’t data byte D following the instruction and the current contents of address (A+n) to memory location (A+n).  In addition, the n’th byte of data D output from the instruction is the unaltered data read from the memory location (A+n). | MEMADDR\_ERROR |
| RXBUF |  | s[7:0]  d[7:0]  ... | Read the oldest byte in the RX FIFO. At the first data transfer the oldest byte in the RX FIFO is read and removed from the RX FIFO. This operation is repeated for subsequent SPI transfers.  If this instruction is performed when the RX FIFO is empty, an RX\_UNDERFLOW exception is raised.  Note: Do not execute RXBUF while RXBUFMOV is in progress. It could result in loss of data. | RX\_UNDERFLOW |
| RXBUFCP | a[11:0] | s[7:0]  c[7:0]  d[7:0]  ... | This instruction functions as RXBUF except it also copies the data bytes read from the RX FIFO to the memory location starting at address A.  The second byte transferred is the number of bytes, C, currently in the RX FIFO.  Note: Do not execute RXBUFCP while RXBUFMOV is in progress. It could result in loss of data. | RX\_UNDERFLOW |
| TXBUF | d[7:0] ... | s[7:0]  c[7:0] | Write to the end of TX FIFO. Data bytes transferred after the opcode are appended to the end of TX FIFO.  The SPI interface will output the number of bytes, C, in TX FIFO before the currently transferred byte has been entered. I.e. 0x00 is returned when transferring the first byte to TX FIFO.  If this instruction is performed when the TX FIFO is full, a TX\_OVERFLOW exception is raised. | TX\_OVERFLOW |
| RANDOM | ... | s[7:0]  d[7:0]  ... | Read randomly generated bytes D, generated from noise in the receiver chain. |  |
| **Data management instructions** | | | | |
| RXBUFMOV | p  a[11:0] c[7:0] | s[7:0]  c[7:0] | Moves the C oldest bytes from the RX FIFO to the memory location starting at address A.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  An RXBUFMOV\_TIMEOUT exception is raised if the RX FIFO empties before the instruction is completed, as defined by DPUCON.RXTIM. The remaining bytes to be moved is available in status register. Note that running RXBUFMOV on high priority with DPUCON.RXTIM=’1’ will block execution of other DPU instructions while a frame is beeing received, which is more than 4ms for a 128 byte frame.  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low). | RXBUFMOV\_TIMEOUT OPERAND\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H  MEMADDR\_ERROR |

46 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| TXBUFCP | p  a[11:0] c[7:0] | s[7:0]  c[7:0] | Copy C bytes of data starting from the memory location starting at address AT Tto the end of TXBUF.  The SPI interface will output the number of bytes, C, in TXBUF.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  If TXBUF fills before the operation is completed a TX\_OVERFLOW exception is raised. The remaining bytes to be moved is available in status register.  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low). | TX\_OVERFLOW  OPERAND\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H  MEMADDR\_ERROR |
| MEMCP | p  c[7:0]  a[11:0] e[11:0] | s[7:0] | Copy data from one memory block to another. Copies the block of C bytes of data from the memory location starting at address A to the memory location starting at address E.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low). | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| MEMCPR | p  c[7:0]  a[11:0] e[11:0] | s[7:0] | Copy data from one memory block to another, and revert endianess. Copies the block of C bytes of data from the memory location starting at address A to the memory location starting at address E, while reverting the endianess of the data block. I.e., data from memory location (A+n) is written to memory location (E+C-1-n).  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low). | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| MEMXCP | p  c[7:0]  a[11:0] e[11:0] | s[7:0] | XOR one memory block with another memory block. The input to the instruction are two memory blocks, both of size C bytes, starting at address A and E respectively. The output is the bitwise XOR of the two memory blocks, written to the memory location starting at address E.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low). | MEMADDR\_ERROR DPU\_DONE\_L  DPU\_DONE\_H  USAGE\_ERROR |

**WWW.TI.COM** 47

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| **Security instructions** | | | | |
| INC | p  c[1:0]  a[11:0] | s[7:0] | Increment the 2C byte word with least significant byte at address A. The nth least significant byte beyond the least is located at address (A+n). (n’ < n means n’ has less significance than n)  C shall be in the range 0 through 2, i.e. 1, 2 or 4 bytes are incremented. If C equals 3, a USAGE\_ERROR exception shall be raised.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not.  The instruction will always access 4 bytes regardless of the C parameter, so a MEMADDR\_ERROR exception is raised if A > 0x3FC. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| ECB | p  k[7:0]  c[3:0]  a[11:0] e[11:0] | s[7:0] | ECB encryption. Encrypt one 16 byte block of plaintext consisting of (16-C) bytes of data read from memory, starting at address A, concatenated with C zero-bytes, using the key stored at address (16⋅K) and storing the output at address E.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  The output is 16 AES-128 encrypted bytes.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| ECBO | p  k[7:0]  c[3:0]  a[11:0] | s[7:0] | ECB encryption. Encrypt one 16 byte block of plaintext consisting of (16-C) bytes of data read from memory, starting at address A, concatenated with C zero-bytes, using the key stored at address (16⋅K) and storing the output by overwriting the data from address A.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  The output is 16 AES-128 encrypted bytes.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| ECBX | p  k[7:0]  c[3:0]  a[11:0] e[11:0] | s[7:0] | ECB encryption and XOR. As ECB except each ciphertext byte is bitwise XOR’ed with the existing byte in the destination address E before it is written to that address. | USAGE\_ERROR  MEMADDR\_ERROR DPU\_DONE\_L  DPU\_DONE\_H |

48 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| CTR | p  k[7:0]  c[6:0]  n[7:0] a[11:0] e[11:0] | s[7:0] | Encryption instruction using counter mode encryption. Process C bytes of plaintext with a starting address at A using the key stored at address (16⋅K), the counter stored at address (16⋅N) storing the output starting at address E.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  If the destination address E provided in the instruction equals zero, the destination address E is set equal to A, thereby replacing the plaintext directly with the ciphertext (or vice versa, in the case of an UCTR instruction).  The output is C encrypted bytes processed according to 802.15.4 standard for security level 4 (CTR).  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| UCTR | p  k[7:0]  c[6:0]  n[7:0] a[11:0] e[11:0] | s[7:0] | Decryption instruction using CTR mode decryption.  This instruction is the same as CTR, since counter mode encryption and decryption are symmetrical operations. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| CBCMAC | p  k[7:0]  c[6:0]  a[11:0] e[11:0] m[2:0] | s[7:0] | Authentication instruction using CBC-MAC security. Process C bytes of plaintext starting at address A, using the key stored at address (16⋅K)T, Tstoring the output starting at address E.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  If the destination address E provided in the instruction equals zero, the destination address E is set equal to (A + C), thereby writing the output directly following the plaintext input data.  The output is 4, 8, or 16 bytes of integrity code for instructions M[1:0] equals 1, 2, or 3 respectively. For M[1:0]=0, no integrity code output is generated.  If M[2]=0, the plaintext data to be authenticated is automatically prefixed with C, as used in IEEE 802.15.4- 2003.  If M[2]=1, the plaintext data is not prefixed with C. This mode can be used for backwards compatibility with existing systems.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H  OPERAND\_ERROR |

**WWW.TI.COM** 49

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| UCBCMAC | p  k[7:0]  c[6:0]  a[11:0] m[2:0] | s[7:0] | Reverse authentication instruction using CBC-MAC security. Process C bytes of plaintext starting at address A, using the key stored at address (16⋅K)  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  The instruction generates 4, 8, or 16 bytes of integrity code (for M[1:0] equals 1, 2 or 3 respectively) and compares them to the received integrity code at address (A + C). The result (pass / fail) is stored in the AUTHSH / AUTHSL status bits for high / low priority security operations respectively. For M[1:0]=0, no integrity code checking is performed and the result will always be ‘pass’.  If M[2]=0, the plaintext data to be authenticated is automatically prefixed with C, as used in IEEE 802.15.4- 2003.  If M[2]=1, the plaintext data is not prefixed with C. This mode can be used for backwards compatibility with existing systems.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| CCM | p  k[7:0]  c[6:0]  n[7:0] a[11:0] e[11:0] f[6:0]  m[1:0] | s[7:0] | Encryption and authentication instruction using CCM / CCM\* security. Authenticate F bytes of plaintext starting at address A. Authenticate and encrypt C bytes starting at address (A+F). Use the key stored at address (16⋅K), the counter starting at address (16⋅N) and storing the output starting at address E.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  If the destination address E provided in the instruction equals zero, the destination address E is set equal to (A+F), thereby replacing the last C bytes of plaintext with the ciphertext and the integrity code.  The output is C encrypted bytes followed by 0, 4, 8 or 16 bytes of integrity code for M equals 0, 1, 2 or 3 respectively.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A USAGE\_ERROR exception is also raised if ((C+F) > 128).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |

50 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **OPCODE** | **Inputs** | **Outputs** | **Description** | **Possible exceptions** |
| --- | --- | --- | --- | --- |
| UCCM | p  k[7:0]  c[6:0]  n[7:0] a[11:0] e[11:0] f[6:0]  m[1:0] | s[7:0] | Decryption and reverse authentication instruction using CCM / CCM\* security. Authenticate F bytes of plaintext with a starting at address A. Decrypt and authenticate C  bytes starting at address (A+F). Use the key stored at address (16⋅K)T, Tthe counter stored at address (16⋅N), storing the output starting at address E.  The priority of the instruction is defined by P, which is either low (if P=0) or high (if P=1).  The output is C plaintext bytes. Note that for the authentication part of the instruction to succed, these bytes should be written back to the address the ciphertext was read from (A+F). This can easily be done by setting E=0x000.  In addition, the instruction generates 0, 4, 8, or 16 bytes of encrypted integrity code (for M equals 0, 1, 2 or 3 respectively) and compares them to the stored integrity code at address (A+C+F). The result (pass / fail) is stored in the AUTHSH / AUTHSL status bits for high / low priority security operations respectively.  A USAGE\_ERROR exception is raised if an instruction is already active with the requested priority level (high or low).  A USAGE\_ERROR exception is also raised if ((C+F) > 128).  A DPU\_DONE\_L or DPU\_DONE\_H exception is raised when the operation completes, depending on the priority of the instruction. This happens regardless of whether the operation was successful or not. | MEMADDR\_ERROR USAGE\_ERROR  DPU\_DONE\_L  DPU\_DONE\_H |
| **Other** | | | | |
| ABORT | c[1:0] | s[7:0] | Abort ongoing data management or security instruction.  c[1]=1: Abort high priority data management or security instructions  c[0]=1: Abort low priority data management or security instructions  c[1]=0: Don’t abort high priority data management or security instructions  c[0]=0: Don’t abort low priority data management or security instructions  Once a class of instructions is aborted, the ongoing instruction is immediately ended leaving the device state as it is at that time. Any pending data management instructions are flushed. |  |
| SRES |  | s[7:0] | Reset the device except the SPI interface.  This instruction can only be run as the first instruction after CSn has been pulled low. |  |

**13.3 Instruction Set Summary**

A summary of the CC2520 instruction set with op-codes is shown in the table below.

**WWW.TI.COM** 51

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**Byte 1 Byte2 Byte3 Byte4 Byte5 Byte6 Byte7 Byte8 Byte9 Mnemonic Pin 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0**

| SNOP SI  SO  IBUFLD SI  SO  SIBUFEX SI  SO  SSAMPLECCA SI  SO  SRES SI  SO  MEMRD SI  SO  MEMWR SI  SO  RXBUF SI  SO  RXBUFCP SI  SO  RXBUFMOV SI  SO  TXBUF SI  SO  TXBUFCP SI  SO  RANDOM SI  SO  SXOSCON SI  SO  STXCAL SI  SO  SRXON SI  SO  STXON SI  SO  STXONCCA SI  SO  SRFOFF SI  SO  SXOSCOFF SI  SO  SFLUSHRX SI  SO  SFLUSHTX SI  SO  SACK SI  SO  SACKPEND SI  SO  SNACK SI  SO  SRXMASKBITSET SI  SO  SRXMASKBITCLR SI  SO  RXMASKAND SI  SO  RXMASKOR SI  SO  MEMCP SI  SO  MEMCPR SI  SO  MEMXCP SI  SO  MEMXWR SI  SO  BCLR SI  SO  BSET SI  SO  CTR / UCTR SI  SO  CBCMAC SI  SO  UCBCMAC SI  SO  CCM SI  SO  UCCM SI  SO  ECB SI  SO  ECBO SI  SO  ECBX SI  SO  INC SI  SO  ABORT SI  SO  REGRD SI  SO  REGWR SI  SO | 0 0 0 0 0 0 0 0 s s s s s s s s 0 0 0 0 0 0 1 0 i s s s s s s s s s s s s s s s s 0 0 0 0 0 0 1 1 s s s s s s s s 0 0 0 0 0 1 0 0 s s s s s s s s 0 0 0 0 1 1 1 1 - s s s s s s s s s s s s s s s s  0 1 0 0 0 0 0 0 s s s s s s s s 0 1 0 0 0 0 0 1 s s s s s s s s 0 1 0 0 0 0 1 0 s s s s s s s s 0 1 0 0 0 0 1 1 s s s s s s s s 01000100 s s s s s s s s 0 1 0 0 0 1 0 1 s s s s s s s s 0 1 0 0 0 1 1 0 s s s s s s s s 0 1 0 0 0 1 1 1 s s s s s s s s 0 1 0 0 1 0 0 0 s s s s s s s s 0 1 0 0 1 0 0 1 s s s s s s s s 0 1 0 0 1 0 1 0 s s s s s s s s 0 1 0 0 1 0 1 1 s s s s s s s s 0 1 0 0 1 1 0 0 s s s s s s s s 0 1 0 0 1 1 0 1 s s s s s s s s  s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s  s s s s s s s s s s s s s s s s | iiiiiii  -------    0 0 0 1 a a a a a a a a a a a a -  0 0 1 1 0 0 0 0 - - - - - - - - ... s s s s s s s s d d d d d d d d ...  0 1 0 0 1 1 1 0 d d d d d d d d d d d d d d d d s s s s s s s s s s s s s s s s s s s s s s s s 0 1 0 0 1 1 1 1 d d d d d d d d d d d d d d d d s s s s s s s s s s s s s s s s s s s s s s s s  0 1 0 1 1 0 0 0 a a a a a b b b    0 1 0 1 1 0 0 1 a a a a a b b b  0 1 1 0 0 1 0 p k k k k kk k k 0c c c c c c c  0 1 1 1 1 0 0 p 0 0 c c a a a a a a a a a a a a s s s s s s s s s s s s s s s s s s s s s s s s 0 1 1 1 1 1 1 1 0 0 0 0 0 0 c c    1 0 a a a a a a - - - - - - - - ... s s s s s s s s d d d d d d d d ... 1 1 a a a a a a d d d d d d d d ... s s s s s s s s d d d d d d d d ... | - - - - - - - s s s s s s s s s s s s s s s s d d d d d d d d ... 0 0 1 0 a a a a a a a a a a a a d d d d d d d d ... s s s s s s s s s s s s s s s s d d d d d d d d ...    0 0 1 1 0 0 1 p cccccccc 0000aaaaaaaaaaaa s s s s s s s s c c c c c c c c s s s s s s s s s s s s s s s s 0 0 1 1 1 0 1 0 d d d d d d d d d d d d d d d d ... s s s s s s s s c c c c c c c c s s s s s s s s ... 0 0 1 1 1 1 1 p cccccccc 0000aaaaaaaaaaaa s s s s s s s s c c c c c c c c s s s s s s s s s s s s s s s s 0 0 1 1 1 1 0 0 - - - - - - - - - - - - - - - - ... s s s s s s s s -------- d d d d d d d d ...      0 1 1 1 0 0 1 p kkkkkkkkcccc aaaaaaaaaaaa s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s | ...        0 0 1 1 1 0 0 0 0 0 0 0 a a a a a a a a a a a a - - - - - - - - ... s s s s s s s s c c c c c c c c s s s s s s s s d d d d d d d d ...                0 1 0 1 0 0 0 p cccccccc aaaaeeeeaaaaaaaaeeeeeeee s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s 0 1 0 1 0 0 1 p cccccccc aaaaeeeeaaaaaaaaeeeeeeee s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s 0 1 0 1 0 1 0 p cccccccc aaaaeeeeaaaaaaaaeeeeeeee s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s 0 1 0 1 0 1 1 0 0 0 0 0 a a a a a a a a a a a a d d d d d d d d ...  s s s s s s s s s s s s s s s s s s s s s s s s d d d d d d d d ...  0 1 1 0 0 1 1 p k k k k kk k k 0c c c c c c c 0 0 0 0 a a a a a a a a a a a a 0 0 0 0 0 mmm s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s  0 1 1 1 0 0 0 p kkkkkkkkcccc aaaaaaaaaaaa 0000eeeeeeeeeeee s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s    0 1 1 1 0 1 0 p kkkkkkkkcccc aaaaaaaaaaaa 0000eeeeeeeeeeee s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s |  | 0 1 1 0 0 0 0 p k k k k kk k k 0c c c c c c c nnnnnnnnaaaaeeeeaaaaaaaaeeeeeeee s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s a a a a e e e e a a a a a a a a e e e e e e e e 0 0 0 0 0 mmm s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s      0 1 1 0 1 0 0 p k k k k kk k k 0c c c c c c c nnnnnnnnaaaaeeeeaaaaaaaaeeeeeeee0 f f f f f f f 00 0 0 0 0 mm s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s0110101p k k k k kk k k 0c c c c c c c nnnnnnnnaaaaeeeeaaaaaaaaeeeeeeee0 f f f f f f f 00 0 0 0 0 mm s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s s |
| --- | --- | --- | --- | --- | --- | --- |

52 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**13.4 Status Byte**

All instructions sent over the SPI to CC2520 result in a status byte being output on SO when the first byte of the instruction is clocked in on SI. The status byte is latched internally when a falling edge is detected on CSn and on the last falling edge of SCLK within each byte. The latched status value is then shifted out on the following falling SCLK edges.

The SNOP instruction can be used to read the status byte without causing any side effects.

**Table 13: Status byte contents**

| **Status byte (MSB clocked out first)** | | |
| --- | --- | --- |
| **Bit no** | **Signal** | **Description** |
| 7 | XOSC stable and running | 0: XOSC off or not yet stable  1: XOSC stable and running (Digital part has clock) |
| 6 | RSSI valid | 0: RSSI value is not valid  1: RSSI value is valid |
| 5 | EXCEPTION channel A | 0: No exceptions selected in EXCMASKAn has corresponding flag in EXCFLAGn set  1: At least one exception selected in EXCMASKAn has corresponding flag EXCFLAGn set |
| 4 | EXCEPTION channel B | 0: No exceptions selected in EXCMASKBn has corresponding flag in EXCFLAGn set  1: At least one exception selected in EXCMASKBn has corresponding flag EXCFLAGn set |
| 3 | DPU H active | 0: No high priority DPU instruction is currently active. 1: A high priority DPU instruction is currently active. |
| 2 | DPU L active | 0: No low priority DPU instruction is currently active.  1: A low priority DPU instruction is currently active. |
| 1 | TX active | 0: Device is not in TX mode  1: Device is in TX mode |
| 0 | RX active | 0: Device is not in RX mode  1: Device is in RX mode |

**13.5 Command Strobes**

Most of the instructions in section 15.3 that are only one byte long are referred to as command strobes. There are two exceptions to this: SNOP and SXOSCON. SNOP is used to read the status byte without causing any side effects. SXOSCON turns on the crystal oscillator and must be run via the SPI. It is not possible to load SXOSCON into the instruction buffer using IBUFLD and then execute it using IBUFEX.

The command strobes can be executed by configuring GPIO pins as input in accordance to GPIO configuration table in section 12.6 and be triggered with a selected edge in the GPIOPOLARITY register. Thus SPI traffic can be omitted for command strobes.

There are also two channels, X and Y, for binding exceptions to the command strobes, so that CC2520 may automatically react to different internal events. This feature is described in more detail in section 16.1.

**13.6 Command Strobe Buffer**

The command strobe buffer provides another mechanism for execution of command strobes. The buffer is loaded with the help of the IBUFLD instruction sent via SPI. Once the buffer is loaded, the instruction is executed when CC2520 receives the SIBUFX strobe. The SIBUFX strobe can be triggered from any of the triggering sources (SPI, GPIO, exceptions bound to SIBUFX instruction). When the instruction in the instruction buffer has been executed, it is replaced by a SNOP instruction. If both the SIBUFEX strobe and

**WWW.TI.COM** 53

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

the IBUFLD instruction are received at the same time, the old command strobe is executed. The new strobe that the user tried to write to the buffer is lost and will never be executed.

54 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**14 Exceptions**

Exceptions in CC2520 are used to indicate that different events have occurred. Exceptions are used both for error conditions such as incorrect use of the SPI and for events that are perfectly normal and expected such as transmission of a start of frame delimiter (SFD). Exception flags are stored in status registers and can be

read over the SPI or observed on GPIO. To clear an exception flag, the user must write ‘0’ to the correct bit in the status register. If the user tries to clear an exception flag in the exact same clock period as the same exception occurs, the flag will not be cleared.

Table 14 shows a summary of the available exceptions in CC2520. The NUM column shows how the exceptions are numbered. The number correspond to the bits in the EXCFLAGn registers, and must be used when binding exceptions to instructions.

**Table 14: Exceptions summary**

| **Mnemonic** | **Num**  **(hex)** | **Description** |
| --- | --- | --- |
| RF\_IDLE | 0x00 | The main radio FSM enters its idle state from any other state. This exception is not generated when the FSM enters the idle state because of a device reset. |
| TX\_FRM\_DONE | 0x01 | TX frame successfully transmitted, which means that TX FIFO is empty and no underflow occurred. Exception is not generated when TX is aborted with SRFOFF, SRXON or STXON. |
| TX\_ACK\_DONE | 0x02 | ACK frame successfully transmitted. Exception is not generated when the acknowledge transmission is aborted with SRFOFF, SRXON or STXON. |
| TX\_UNDERFLOW | 0x03 | Underflow has occurred in the TX FIFO. TX is aborted and the TX FIFO must be flushed. |
| TX\_OVERFLOW | 0x04 | An attempt was made to write to TX FIFO while it is full. The instruction is aborted. |
| RX\_UNDERFLOW | 0x05 | An attempt has been made to read the RX FIFO without any bytes available to read. Instruction is aborted.  Note that the RX\_UNDERFLOW exception should only be used for debugging software, and should not be trusted in a RX FIFO readout routine. In some scenarios the RX\_UNDERFLOW exception will not be issued when a reading starts even when the RX\_FIFO is empty. |
| RX\_OVERFLOW | 0x06 | An attempt has been made by RF\_core to write to RX FIFO while the RX FIFO is full. The byte that was attempted written to the RX FIFO is lost. Reception of data is aborted and the FSM enters the rx\_overflow state. Recommended action is to issue a SFLUSHRX command strobe to empty the RX FIFO and restart RX. |
| RXENABLE\_ZERO | 0x07 | RX enable register has changed value to all zeros. |
| RX\_FRM\_DONE | 0x08 | A complete frame has been received. I.E the number of bytes set by the length field is received. |
| RX\_FRM\_ACCEPTED | 0x09 | When frame filtering is enabled, this exception is generated when a frame is accepted (happens immediately after receiving the fields required to determine the outcome). |
| SRC\_MATCH\_DONE | 0x0A | When source address matching is enabled, this exception is generated upon completion of source address matching. The exception is generated regardless of the result. |
| SRC\_MATCH\_FOUND | 0x0B | If a source match is found, this exception is generated immediately before SRC\_MATCH\_DONE. |
| FIFOP | 0x0C | The RX FIFO is filled up with bytes that have passed address filtering to the FIFOP threshold value defined in register, or at least one complete frame has been written to the RX FIFO. High when FFCTRL is in the rx\_overflow state. |
| SFD | 0x0D | Start of frame delimiter received when in RX or start of frame delimiter transmitted when in TX. |
| DPU\_DONE\_L | 0x0E | Low priority DPU operation completed. Will not be issued if operation fails or is aborted. |

**WWW.TI.COM** 55

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **Mnemonic** | **Num**  **(hex)** | **Description** |
| --- | --- | --- |
| DPU\_DONE\_H | 0x0F | High priority DPU operation completed. Will not be issued if operation fails or is aborted. |
| MEMADDR\_ERROR | 0x10 | An illegal address has been used for an instruction. Instruction is aborted. |
| USAGE\_ERROR | 0x11 | Instruction performed in a context that does not permit this instruction. Instruction is aborted. |
| OPERAND\_ERROR | 0x12 | Wrong format for instruction. Instruction is aborted. This will happen for a multi-byte fixed-length instruction if CSn is raised on a byte boundary but before the required number of operands has been transferred. |
| SPI\_ERROR | 0x13 | An SPI transfer was aborted by raising CSn in the middle of a byte. (I.e. not on a byte boundary) |
| RF\_NO\_LOCK | 0x14 | If no lock has been found before 256 us after entering RX this exception will go active. Also a negative edge on LOCK\_STATUS when in RX will trigger this exception. |
| RX\_FRM\_ABORTED | 0x15 | Frame reception aborted. Not issued when RX\_OVERFLOW occurs. |
| RXBUFMOV\_TIMEOUT | 0x16 | RXBUFMOV has timed out. There were not enough bytes available in the RX FIFO and the wait time set by DPUCON.RXTIMEOUT has expired. |
| UNUSED | 0x17 | Reserved |

**14.1 Exceptions on GPIO Pins**

All exception flags can be routed individually to a GPIO pin by writing the CTRLn value corresponding to the desired exception in Table 8 into the GPIOCTRLn registers.

CC2520 has two exception channels, A and B, that let the user select a collection of exceptions to combine to output on a GPIO pin. If any of the selected exceptions goes active, the GPIO pin goes active. It is also possible to output the complementary collection of exceptions of each of the two channels.

**Example:** Collect RF\_IDLE and RX\_UNDERFLOW in exception channel B and output on GPIO3.

• Write 0x22 to GPIOCTRL3. Set GPIO3 as output and select exception channel B from the GPIO configuration table in section 12.6.

• Write 0x21 to EXCMASKB0. Select RF\_IDLE and RX\_UNDERFLOW exceptions in accordance with table Exceptions overview (section 16).

• Write 0x00 to EXCMASKB1. Mask all other exceptions.

• Write 0x00 to EXCMASKB2. Mask all other exceptions.

The complementary exception channel B with the settings in the example above will include all other exceptions than RF\_IDLE and RX\_UNDERFLOW. This channel can be routed to another GPIO pin by writing 0x24 to the corresponding GPIOCTRLn register.

Exceptions linked to GPIO pins separately or as a group in a channel will be consistent with the corresponding bits in the EXCFLAGn registers. EXCFLAGn register bits that are high can only be cleared by writing zero to the bit.

**14.2 Predefined Exception Channels**

There are two predefined exception channels that can be observed on GPIO pins. They are not included in the status byte and no complementary channel is available.

The first predefined exception channel is a collection of exceptions that indicate that something has gone wrong during RX.

• RX\_UNDERFLOW

• RX\_OVERFLOW

• RX\_FRM\_ABORTED

• RXBUFMOV\_TIMEOUT

56 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

The second predefined exception channel includes exceptions that indicate general error conditions. • MEMADDR\_ERROR

• USAGE\_ERROR

• OPERAND\_ERROR

• SPI\_ERROR

Figure 10 shows how the exceptions are linked to the instruction set of CC2520. Note that there are several sources that may trigger instructions. The large or-gate illustrates that it only takes one of these sources to trigger the execution of an instruction.









s

u

b

s

n

o

i

t

p

e

c

x

E





**Figure 10: Functional details of exception handling and instruction triggering.**

**14.3 Binding Exceptions to Instructions (command strobes)**

An exception can be bound to trigger a command strobe so that a command strobe can be automatically executed when an exception occurs. There are two possible binding combinations, X and Y, defined in the registers EXCBINDXn and EXCBINDYn.

**Example**

Run SACKPEND instruction when RX\_FRM\_ACCEPTED exception is activated.

**WWW.TI.COM** 57

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

1. Write 0x06 to EXCBINDX0. This will select SACKPEND as the bound instruction from Table 8: GPIO configuration.

2. Write 0x89 to EXCBINDX1. Enables X-binding and selects RX\_FRM\_ACCEPTED as the bound exception from Table 14: Exceptions summary.

**Note**

Be aware of the offset in numbering in the tables Exceptions summary (section 16) and GPIO configuration (section 12.6) for exceptions.

It is for example possible to route the exception RF\_IDLE to a GPIO pin in the GPIOCTRLn.CTRLn register bit when the pin is set as output. In this case, the exception RF\_IDLE has the numbering 0x01 in accordance to Table 9: GPIO configuration

When RF\_IDLE is to be bound with an instruction the numbering to be used in EXCBINDX/Y1 is 0x00 in accordance to Table 14: Exceptions summary.

58 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**15 Memory Map**

The configuration registers in CC2520 are located at addresses from 0x000 to 0x07F. From 0x080 to 0x0FF there is currently a reserved area that is not used. CC2520 contains 768 bytes of physical RAM located at addresses 0x100 to 0x3FF.



**Figure 11: CC2520 memory map** 

**WWW.TI.COM** 59

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**15.1 FREG**

FREG is 128 fast access 8-bit registers that can be reached with REGRD and REGWR instructions. REGRD and REGWR instructions that begin in the FREG memory area can be continued into the SREG and wrap around at 0x07F. FREG can also be accessed with MEMRD and MEMWR instructions which require one extra byte over the SPI with respect to REGRD and REGWR.

Registers in FREG between 0x000 and 0x01F are bit wise writeable with the BCLR and BSET instructions. The registers located in FREG are described in section 28. Note that not all 128 addresses are used.

**15.2 SREG**

SREG is 128 8-bit registers that are accessible with MEMRD and MEMWR instructions. The registers located in SREG are described in section 32. Note that not all 128 addresses are used.

**15.3 TX FIFO**

The TX FIFO memory area is located at addresses 0x100 to 0x17F and is thus 128 bytes. Although this memory area is intended for the TX FIFO, it is not protected in any way, so it is still accessible with for instance the MEMWR and MEMRD instructions. Normally, only the designated instructions should be used to manipulate the contents of the TX FIFO. The TX FIFO can only contain one frame at a time. More details on the TX FIFO can be found in section 22.3.

**15.4 RX FIFO**

The RX FIFO memory area is located at addresses 0x180 to 0x1FF and is thus 128 bytes. Although this memory area is intended for the RX FIFO, it is not protected in any way, so it is still accessible with for instance the MEMWR and MEMRD instructions. Normally, only the designated instructions should be used to manipulate the contents of the RX FIFO. The RX FIFO can contain more than one frame at a time.

**15.5 MEM**

The MEM memory area from address 0x200 to 0x37F is 384 bytes long. The two 16-byte temporary areas CBCTEMPH and CBCTEMPL are used for CBCMAC, UCBCMAC, CCM and UCCM instructions, with high and low priority respectively. The remaining MEM area is general purpose memory.

**15.6 Frame Filtering and Source Matching Memory Map**

The frame filtering and source address matching functions use a 128-byte block of CC2520 memory to store local address information and source matching configuration and results. This memory space is described in Table 15. Values that do not fill an entire byte/word are in the least significant part of the byte/word.

**Table 15: Frame Filtering and Source Matching Memory map**

| **Address** | **REGISTER / Variable** | **Endian** | **Description** |
| --- | --- | --- | --- |
| Reserved | | | |
| 0x3F6-3FF | Temporary storage |  | Memory space used for temporary storage of variables. |
| Local address information | | | |
| 0x3F4-0x3F5 | SHORT\_ADDR | LE | The short address used during destination address filtering. |
| 0x3F2-0x3F3 | PAN\_ID | LE | The PAN ID used during destination address filtering. |
| 0x3EA-0x3F1 | EXT\_ADDR | LE | The IEEE extended address used during destination address filtering. |
| Source address matching control | | | |

60 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **Address** | **REGISTER / Variable** | | **Endian** | | **Description** |
| --- | --- | --- | --- | --- | --- |
| 0x3E9 | SRCSHORTPENDEN2 | |  | | 8 MSBs of the 24-bit mask that enables / disables automatic pending for each of the 24 short address. |
| 0x3E8 | SRCSHORTPENDEN1 | |  | |  |
| 0x3E7 | SRCSHORTPENDEN0 | |  | | 8 LSBs of the 24-bit mask that enables / disables automatic pending for each of the 24 short address. |
| 0x3E6 | SRCEXTPENDEN2 | |  | | 8 MSBs of the 24-bit mask that enables / disables automatic pending for each of the 12 extended addresses. Entry n is mapped SRCEXTPENDEN[2n]. All SRCEXTPENDEN[2n+1] bits are don't care. |
| 0x3E5 | SRCEXTPENDEN1 | |  | |  |
| 0x3E4 | SRCEXTPENDEN0 | |  | | 8 LSBs of the 24-bit mask that enables / disables automatic pending for each of the 12 extended addresses. Entry n is mapped SRCEXTPENDEN[2n]. All SRCEXTPENDEN[2n+1] bits are don't care. |
| Source address matching result | | | | | |
| 0x3E3 | SRCRESINDEX | |  | | The bit index of the least significant '1' in SRCRESMASK, or 0x3F when there is no source match.  Upon a match, bit 5 is '0' when the match is on a short address and '1' when it is on an extended address.  Upon a match, bit 6 is '1' when the conditions for automatic pending bit in acknowledgment have been met (see the description of SRCMATCH.AUTOPEND). The bit gives no indication of whether or not the acknowledgment actually is transmitted, and does not take the PENDING\_OR register bit and the SACK/SACKPEND/SNACK strobes into account. |
| 0x3E2 | SRCRESMASK2 | |  | | 24-bit mask that indicates source address match for each individual entry in the source address table.  Short address matching: When there is a match on entry panid\_n + short\_n, bit n will be set in SRCRESMASK.  Extended address matching: When there is a match on entry ext\_n, bits 2n and 2n+1 will be set in SRCRESMASK. |
| 0x3E1 | SRCRESMASK1 | |  | |
| 0x3E0 | SRCRESMASK0 | |  | |
| Source address table | | | | | |
| 0x3DE-0x3DF | short\_23 | ext\_11 | LE | LE | 2 individual short address entries (combination of 16 bit PAN ID and 16 bit short address) or 1 extended address entry. |
| 0x3DC-0x3DD | panid\_23 | LE |
| 0x3DA-0x3DB | short\_22 | LE |
| 0x3D8-0x3D9 | panid\_22 | LE |
| - - - - - | | | | | |
| 0x38E-0x38F | short\_03 | ext\_01 | LE | LE | 2 individual short address entries (combination of 16 bit PAN ID and 16 bit short address) or 1 extended address entry. |
| 0x38C-0x38D | panid\_03 | LE |
| 0x38A-0x38B | short\_02 | LE |
| 0x388-0x389 | panid\_02 | LE |
| 0x386-0x387 | short\_01 | ext\_00 | LE | LE | 2 individual short address entries (combination of 16 bit PAN ID and 16 bit short address) or 1 extended address entry. |
| 0x384-0x385 | panid\_01 | LE |
| 0x382-0x383 | short\_00 | LE |
| 0x380-0x381 | panid\_00 | LE |

**WWW.TI.COM** 61

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**16 Frequency and Channel Programming**

The carrier frequency is set by programming the 7 bit frequency word located in FREQCTRL.FREQ[6:0]. CC2520 supports carrier frequencies in the range 2394MHz to 2507MHz. The carrier frequency FC in MHz is given by FC = (2394 + FREQCTRL.FREQ[6:0]) MHz, and is programmable in 1 MHz steps.

IEEE 802.15.4-2006 specifies 16 channels within the 2.4 GHz band. They are numbered 11 through 26 and are 5 MHz apart. The RF frequency of channel k is given by [2].

*F* = 2405+ 5(*k* −11) [*MHz*] *k* ∈[11,26] *c*

For operation in channel k, the FREQCTRL.FREQ register should therefore be set to FREQCTRL.FREQ = 11 + 5 (k-11)

62 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**17 IEEE 802.15.4-2006 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 802.15.4-2006. For a complete description, please refer to the standard document [2].

The modulation and spreading functions are illustrated at block level in Figure 12. 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, except for security related fields where the most significant byte it 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 16. The chip sequence is then transmitted at 2 Mchips/s, with the least significant chip (C0) transmitted first for each symbol. The transmitted bit stream and the chip sequences are observable on GPIO pins. See Table 9 for details on how to configure the GPIO to do this.

**Figure 12: Modulation** 

**Table 16: IEEE 802.15.4-2006 symbol to chip mapping**

| **Symbol** | **Chip sequence (C0, C1, C2, … , C31)** |
| --- | --- |
| 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 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 |

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 13.

**WWW.TI.COM** 63

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

****

**Figure 13: I / Q Phases when transmitting a zero-symbol chip sequence, TC = 0.5 µs**

64 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**18 IEEE 802.15.4-2006 Frame Format**

This section gives a brief summary of the IEEE 802.15.4 frame format [2]. CC2520 has built in support for processing of parts of the frame. This is described in the following sections.

Figure 18 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 the standard document [2].

Bytes: 2 1

0 to 20

n

2

**MAC Layer**

Frame

Control Field (FCF)

Data

Sequence Number

Address

Information

Frame payload

Frame Check Sequence

(FCS)

MAC Header (MHR) MAC Payload MAC Footer

(MFR)

Bytes: 1 1 5 + (0 to 20) + n

4

**PHY Layer**

Preamble Sequence

Start of frame Delimiter

(SFD)

Frame Length

MAC Protocol Data Unit

(MPDU)

Synchronisation Header (SHR)

PHY Header (PHR)

11 + (0 to 20) + n

PHY Protocol Data Unit (PPDU)

PHY Service Data Unit (PSDU)

**Figure 14: Schematic view of the IEEE 802.15.4 Frame Format [1]**

**18.1 PHY Layer**

**Synchronization Header**

The synchronization header (SHR) consists of the preamble sequence followed by the start of frame delimiter (SFD). In the IEEE 802.15.4 specification [2], the preamble sequence is defined to be 4 bytes of 0x00. The SFD is one byte with value 0xA7.

**PHY Header**

The PHY header consists only of the frame length field. The frame length field defines the number of bytes in the MPDU. Note that the value of 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 TCC2520 Thardware.

The frame length field is 7 bits long and has a maximum value of 127. The most significant bit in the length field is reserved, and should always be set to zero.

**PHY Service Data Unit**

The PHY Service Data Unit contains the MAC Protocol Data Unit (MPDU). It is the MAC layer’s responsibility to generate/interpret the MPDU, and CC2520 has built in support for processing of some of the MPDU subfields.

**18.2 MAC Layer**

The FCF, data sequence number and address information follows the length field as shown in Figure 14. Together with the MAC data payload and Frame Check Sequence, they form the MPDU. The format of the FCF is shown in Figure 15. For full details, please refer to the IEEE 802.15.4 specification [2].

| **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 15: Format of the Frame Control Field (FCF)**

**WWW.TI.COM** 65

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**Frame Check Sequence**

A 2-byte frame check sequence (FCS) follows the last MAC payload byte as shown in Figure 14. The FCS is calculated over the MPDU, i.e. the length field is not part of the FCS.

The FCS polynomial defined in [2] is

( ) 1 16 12 5 *G x* = *x* + *x* + *x* +

CCC2520 supports automatic calculation/verification of the FCS. See sections 20.3 and 22.1.3 for details.

66 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**19 Transmit Mode**

This section describes how to control the transmitter, the integrated frame processing and how to use the TX FIFO.

**19.1 TX Control**

CC2520 has many built in features for frame processing and status reporting. Note that CC2520 provides features that make it easy for the microcontroller to have precise control of the timing of outgoing frames. This is very important in an IEEE 802.15.4/ZigBee system, because there are strict timing requirements to such systems.

Frame transmission will be started by the following actions:

• The STXON command strobe

o The SAMPLED\_CCA signal is not updated.

• The STXONCCA command strobe, provided that the CCA signal is high.

o Aborts ongoing transmission/reception and forces a TX calibration followed by transmission. o The SAMPLED\_CCA signal is updated

Clear channel assessment is described in detail in section 19.7.

Frame transmission will be aborted by the following command actions:

• The SRXON command strobe

o Aborts ongoing transmission and forces a RX calibration

• The SRFOFF command strobe

o Aborts ongoing transmission/reception and forces the FSM to the IDLE state.

• The STXON command strobe

o See above.

To enable the receiver after transmission with STXON, the FRMCTRL1.SET\_RXENMASK\_ON\_TX bit should be set. This will set bit 14 in RXENABLE when STXON is executed. When transmitting with STXONCCA, the receiver would be on before the transmission and will be turned back on afterwards (unless the RXENABLE registers have been cleared in the mean time).

**19.2 TX State Timing**

Transmission of preamble begins 192 us after the STXON or STXONCCA command strobe. This is referred to as "TX turnaround time" in [2]. There is an equal delay when returning to receive mode.

When returning to idle or receive mode, there is a 2 us delay while the modulator ramps down the signals to the DACs. The down ramping happens automatically after the complete MPDU (as defined by the length byte) has been transmitted or if TX underflow occurs. This affects:

• The SFD signal, which is stretched by 2 us.

• The radio FSM transition to the IDLE state, which is delayed by 2 us.

**19.3 TX FIFO Access**

The TX FIFO can hold 128 bytes and only one frame at a time. The frame can be buffered before or after the TX command strobe is executed, as long as it does not generate TX underflow (see the error conditions listed below).

Figure 16 illustrates what needs to be written to the TX FIFO (marked blue). Additional bytes are ignored, unless TX overflow occurs (see the error conditions listed below).

**WWW.TI.COM** 67

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

****

****

****

**Figure 16. Frame data written to the TX FIFO **

There are three ways to write to the TX FIFO:

• The TXBUF instruction transfers bytes from the microcontroller to the TX FIFO in CC2520. • The TXBUFCP instruction copies bytes from the general RAM in CC2520 into the TX FIFO. • Frame buffering always begins at the start of the TX FIFO memory. By enabling the

FRMCTRL1.IGNORE\_TX\_UNDERF bit, it is possible to MEMWR, MEMCP and other memory instructions to write the frame. Note, however, that using dedicated TXBUF and TXBUFCP instructions should be preferred.

The number of bytes in the TX FIFO is stored in the TXFIFOCNT register.

The TX FIFO can be emptied manually with the SFLUSHTX command strobe. TX underflow will occur If the FIFO is emptied during transmission.

**19.3.1 Retransmission**

In order to support simple retransmission of frames, the CC2520 does not delete TX FIFO contents as they are transmitted. After a frame has been successfully transmitted, the FIFO contents are left unchanged. To retransmit the same frame again, simply restart TX by issuing a STXON or STXONCCA command strobe.

If a different frame is to be transmitted, just write the new frame to the TX FIFO. In this case, the TX FIFO is automatically flushed before the actual writing takes place.

**19.3.2 Error Conditions**

There are two error conditions associated with the TX FIFO:

• Overflow happens when the TX FIFO is full and it is attempted to write another byte. • Underflow happens when the TX FIFO is empty and CC2520 attempts to fetch another byte for transmission.

TX overflow is indicated by the TX\_OVERFLOW exception. When this error occurs, the writing will be aborted, i.e. the data byte that caused the overflow will be lost. The error condition must be cleared with the SFLUSHTX strobe.

TX underflow is indicated by the TX\_UNDERFLOW exception. When this error occurs, the ongoing transmission is aborted. The error condition must be cleared with the SFLUSHTX strobe.

The TX\_UNDERFLOW exception can be disabled by setting the FRMCTRL1.IGNORE\_TX\_UNDERF bit. In this case, the CC2520 will continue transmitting the bytes that happen to be in the TX FIFO memory, until the number of bytes given by the first byte (i.e. the length byte) has been transmitted

68 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**19.4 TX Flow Diagram**

Figure 17 summarizes the previous sections in a flow diagram: No CSMA-CA Unslotted CSMA-CA Slotted CSMA-CA

**SSAMPLECCA**

Success? Yes

(SAMPLED\_CCA = 1)

No

Data buffering

Write a frame to the TX buffer using:

- TXBUF

- TXBUFCP

- Memory access

- A combination of these methods

**STXON STXONCCA**

(SAMPLED\_CCA = 0)

This can be done before, after or in parallel with the TX strobe.

Yes

(SAMPLED\_CCA = 1)

TX started?

No

(SAMPLED\_CCA = 0) TX is aborted by

TX buffer overfilled

TX completes?

Yes

TX\_FRM\_DONE

~~No~~

Why?

TX\_UNDERFLOW

~~SRXON,~~

STXON or SRFOFF

TX\_OVERFLOW

Frame transmitted successfully Incomplete or no frame transmission Error condition

T

I

M

E

Error condition

(left side of the flow diagram should be ignored since the TX buffer is corrupted)

Between two transmissions there can be multiple other activities such as frame reception, RX FIFO access and acknowledgment transmission (using SACK, SACKPEND or AUTOACK), or idle periods (random backoffs). This will have no side effects on the state of the TX buffer.

The placement of the SFLUSHTX strobe in the diagram shows the latest point in time where this strobe can be executed. If fewer special cases is desired, it is always possible to use the SFLUSHTX strobe and then load or reload TXBUF with the next frame to be transmitted.

Next time...

To retransmit the

To transmit a

To retransmit or transmit a

different frame...

Next time...

To (re)transmit

what is

To transmit a

To retransmit or transmit a

different frame...

current frame... Restart from the

different frame... Restart from the

currently in

the TX buffer... Restart from the

different frame...

top of the diagram

Do not write

anything to the TX

top of the diagram

Write the new frame to the TX

**SFLUSHTX SFLUSHTX** top of the diagram

If anything is

written to the TX

**SFLUSHTX**

buffer

buffer

(before, after or in parallel with the TX strobe)

Restart from the top of the diagram

Write the next frame to the TX buffer

(before, after or in parallel with the TX strobe)

buffer, it will be

appended to the

current data.

**Figure 17: TX flow**

Restart from the top of the diagram

Write the new frame to the TX buffer

(before, after or in parallel with the TX strobe)

Restart from the top of the diagram

Write the next frame to the TX buffer

(before, after or in parallel with the TX strobe)

**WWW.TI.COM** 69

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**19.5 Frame Processing**

CC2520 performs the following frame generation tasks for TX frames:

Transmitted frame

Preamble SFD LEN MHR FCS

MAC Payload

1 3

2

1. Generation and automatic transmission of the PHY Layer synchronization header which consists of the preamble and the SFD.

2. Transmission of the number of bytes specified by the frame length field.

3. Calculation of and automatic transmission of the FCS (can be disabled).

The recommended usage is to write the length field followed by MAC header and MAC payload to the TX FIFO, and let CC2520 handle the rest. Note that the length field must include the two FCS bytes even though CC2520 adds these automatically.

**19.5.1 Synchronization Header**

****

**Figure 18: Transmitted Synchronisation Header** 

TCC2520 has programmable Tpreamble length. The default value is compliant with [2] and changing the value will make the system non-compliant to IEEE 802.15.4.

The preamble sequence length is set by MDMCTRL0.PREAMBLE\_LENGTH. Figure 18 shows how the TCC2520T synchronization header relates to the IEEE 802.15.4 specification.

When the required number of preamble bytes have been transmitted, CC2520 will automatically transmit the one byte long SFD. The SFD is fixed and it is not possible to change this value from software.

**19.5.2 Frame Length Field**

When the SFD has been transmitted, the modulator in CC2520 will start to read data from the TX FIFO. It expects to find the frame length field followed by MAC header and MAC payload. The frame length field is used to determine how many bytes that is to be transmitted.

Note that the minimum frame length is 3 when AUTOCRC=’1’ and 1 when AUTOCRC=’0’.

**19.5.3 Frame Check Sequence**

When the FRMCTRL0.AUTOCRC control bit is set, the FCS field is automatically generated by CC2520 and appended to the transmitted frame at the position defined by the length field. The FCS is not written to the TXFIFO, but stored in a separate 16-bit register. It is recommended to always have AUTOCRC enabled, except possibly for debug purposes. If FRMCTRL0.AUTOCRC=’0’ then the modulator will expect to find the

70 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

FCS in the TX FIFO, so software must generate the FCS and write it to the TX FIFO along with the rest of the MPDU.

The TCC2520T hardware implementation of the FCS calculator is shown in Figure 22. Please refer to [2] for further details.

**Figure 19: CC2520 FCS hardware implementation** 

**19.6 Exceptions**

The SFD exception will be raised when the SFD field of the frame has been transmitted. At the end of the frame, the TX\_FRM\_DONE exception will be raised when the complete frame has been successfully transmitted.

Note that there is a second SFD signal available on GPIO (config value 0x2A) that should not be confused with the SFD exception.

**19.7 Clear Channel Assessment**

The clear channel assessment (CCA) status signal indicates whether the channel is available for transmission or not. The CCA function is used to implement the CSMA-CA functionality specified in the IEEE 802.15.4 specification [2]. The CCA signal is valid when the receiver has been enabled for at least 8 symbol periods. The RSSI\_VALID status signal can be used to verify this.

The CCA is based on the RSSI value and a programmable threshold. The exact behavior is configurable in the CCACTRL0 and CCACTRL1 registers.

There are two variations of the CCA signal, one that is updated at every new RSSI sample and one that is only updated on SSAMPLECCA and STXONCCA command strobes. They are both available in the FSMSTAT1 register.

Note that the CCA signal is updated 4 clock cycles (32 MHz) after the RSSI\_VALID signal has been set.

**19.8 Output Power Programming**

The RF output power of CC2520 is controlled by the 7 bit value in the TXPOWER register. Table 17 shows the typical output power and current consumption for the recommended settings when the centre frequency is set to 2440 GHz. Note that the recommended settings are only a small subset of all the possible register

settings. Using other settings than those in Table 17 might result in very high current consumption and generally poor performance. Please refer to section 5.11 for details on the optional temperature compensated TX.

**WWW.TI.COM** 71

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**Table 17: Output power and current consumption measured on the CC2520 reference design @ +3.0 V, +25°C, fc=2.440 GHz**

| **TXPOWER**  **register (hex)** | **Typical output power (dBm)** | **Typical current**  **consumption (mA)** |
| --- | --- | --- |
| F7 | 5 | 33.6 |
| F2 | 3 | 31.3 |
| AB | 2 | 28.7 |
| 13 | 1 | 27.9 |
| 32 | 0 | 25.8 |
| 81 | -2 | 24.9 |
| 88 | -4 | 23.1 |
| 2C | -7 | 19.9 |
| 03 | -18 | 16.2 |

**19.9 Tips And Tricks**

• Trigger the STXON and STXONCCA strobes from GPIO pins. This gives the microcontroller very accurate control of the timing of the outgoing frame.

• Use a timer in the microcontroller to capture the timing of the SFD exception. This gives the microcontroller exact knowledge of when the frame was transmitted.

• Note that there is no requirement to have the complete frame in the TXFIFO before starting a transmission. Bytes may be added to the TX FIFO during transmission.

• It is possible to make CC2520 transmit non-IEEE 802.15.4 compliant frames by setting MDMTEST1.MODULATION\_MODE=’1’.

72 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

**20 Receive Mode**

This section describes how to control the receiver, integrated RX frame processing, and how use the RX FIFO.

**20.1 RX Control**

The CC2520 receiver is turned on and off with the SRXON and SRFOFF command strobes, and with the RXENABLE registers. The command strobes provide a "hard" on/off mechanism, while RXENABLE manipulation provides a "soft" on/off mechanism.

The receiver will be turned **on** by the following actions:

• The SRXON strobe:

o Sets RXENABLE[15]

o Aborts ongoing transmission/reception by forcing a transition to RX calibration.

• The STXON strobe when FRMCTRL1.SET\_RXENMASK\_ON\_TX is enabled:

o Sets RXENABLE[14]

o The receiver is enabled after transmission completes.

• Setting RXENABLE != 0x0000:

o Does not abort ongoing transmission/reception.

The receiver will be turned **off** by the following actions:

• The SRFOFF strobe:

o Clears RXENABLE[15:0]

o Aborts ongoing transmission/reception by forcing the transition to IDLE mode.

• Setting RXENABLE = 0x0000

o Does not abort ongoing transmission/reception. Once the ongoing transmission/reception is finished, the CC2520 will return to IDLE state.

There are several ways to manipulate the RXENABLE registers:

• The REGWR and MEMWR instructions

• The BSET and BCLR instructions

• The RXENABLEAND and RXENABLEOR instructions

• The SRXMASKBITSET and SRXMASKBITCLR strobes (affecting RXENABLE[13]) • The SRXON, SRFOFF and STXON strobes, including the FRMCTRL1.SET\_RXMASK\_ON\_TX setting

**20.2 RX State Timing**

The receiver is ready 192 us after RX has been enabled by one of the methods described above. This is referred to as "RX turnaround time" in [2].

When returning to receive mode after frame reception, there is by default an interval of 192 us where SFD detection is disabled. This interval can be disabled by clearing FSMCTRL.RX2RX\_TIME\_OFF.

**20.3 Frame Processing**

CC2520 integrates critical portions of the RX requirements in IEEE 802.15.4-2003 and -2006 in hardware. This reduces the microcontroller interruption rate, simplifies the software that handles frame reception, and provides the results with minimum latency.

During reception of a single frame, the CC2520 performs the following frame processing steps:



**WWW.TI.COM** 73

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

1. Detection and removal of the received PHY synchronization header (preamble and SFD), and reception of the number of bytes specified by the frame length field.

2. Frame filtering as specified by [1] and [2], section 7.5.6.2, third filtering level.

3. Matching of the source address against a table containing up to 24 short addresses or 12 extended IEEE addresses. The source address table is stored on-chip in RAM.

4. Automatic FCS checking, and attaching this result and other status values (RSSI, LQI and source match result) to received frames.

5. Automatic acknowledgment transmission with correct timing, and correct setting of the frame pending bit, based on the results from source address matching and FCS checking.

**20.3.1 Synchronization Header And Frame Length Fields**

Frame reception starts with detection of a start-of-frame delimiter (SFD), followed by the length byte, which determines when the reception is complete. The SFD signal, which is default output on GPIO4, can be connected to a timer input on a microcontroller to capture the start of received frames:



**Figure 20: SFD signal timing**

Preample and SFD are not written to the RX FIFO.

The CC2520 uses a correlator to detect the SFD. The correlation threshold value in MDMCTRL1.CORR\_THR determines how closely the received SFD must match an "ideal" SFD. The threshold must be adjusted with care:

• If set too high, CC2520 will miss lots of actual SFDs, effectively reducing the receiver sensitivity. • If set too low, CC2520 will detect lots of false SFDs. Although this does not reduce the receiver sensitivity, the effect will be similar, since false frames might overlap with SFDs of actual frames. It also increases the risk of receiving false frames with correct FCS.

In addition to SFD detection, it is also possible to require a number of valid preamble symbols (also above the correlation threshold) prior to SFD detection. Refer to the register descriptions of MDMCTRL0 and MDMCTRL1 for available options and recommended settings.

For CC2520 rev. A the default correlation threshold is too low, and must updated after reset (before RX is attempted).

**20.3.2 Frame Filtering**

The frame filtering function rejects non-intended frames as specified by [1] and [2], section 7.5.6.2, third filtering level. In addition, it provides filtering on:

• The 8 different frame types (see the FRMFILT1 register)

• The reserved bits in the frame control field (FCF)

The function is controlled by:

• The FRMFILT0 and FRMFILT1 registers

• The LOCAL\_PAN\_ID, LOCAL\_SHORT\_ADDR and LOCAL\_EXT\_ADDR values in RAM

74 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**Filtering Algorithm**

The FRMFILT0.FRM\_FILTER\_EN bit controls whether frame filtering is applied or not. When disabled, the CC2520 will accept all received frames. When enabled (which is the default setting), the CC2520 will only accept frames that fulfill all of the following requirements:

• The length byte must be equal to or higher than the “minimum frame length”, which is derived from the source- and destination address mode and PAN ID compression subfields of the FCF.

• The reserved FCF bits [9:7] and’ed together with FRMFILT0.FCF\_RESERVED\_BITMASK must equal 0b000.

• The value of the frame version subfield of the FCF cannot be higher than

FRMFILT0.MAX\_FRAME\_VERSION.

• The source and destination address modes cannot be reserved values (1).

• Destination address:

• If a destination PAN ID is included in the frame, it must match LOCAL\_PANID or must be the broadcast PAN identifier (0xFFFF).

• If a short destination address is included in the frame, it must match either LOCAL\_SHORT\_ADDR or the broadcast address (0xFFFF).

• If an extended destination address is included in the frame, it must match LOCAL\_EXT\_ADDR. • Frame type:

• Beacon frames (0) are only accepted when:

• FRMFILT1.ACCEPT\_FT0\_BEACON = 1

• Length byte >= 9

• The destination address mode is 0 (no destination address)

• The source address mode is 2 or 3 (i.e. a source address is included)

• The source PAN ID matches LOCAL\_PANID, or LOCAL\_PANID equals 0xFFFF • Data (1) frames are only accepted when:

• FRMFILT1.ACCEPT\_FT1\_DATA = 1

• Length byte >= 9

• A destination address and/or source address is included in the frame. If no destination address is included in the frame, the FRMFILT0.PAN\_COORDINATOR bit must be set and the source PAN ID must equal LOCAL\_PANID.

• Acknowledgment (2) frames are only accepted when:

• FRMFILT1.ACCEPT\_FT2\_ACK = 1

• Length byte = 5

• MAC command (3) frames are only accepted when:

• FRMFILT1.ACCEPT\_FT3\_MAC\_CMD = 1

• Length byte >= 9

• A destination address and/or source address is included in the frame. If no destination address is included in the frame, the FRMFILT0.PAN\_COORDINATOR bit must be set and the source PAN ID must equal LOCAL\_PANID for the frame to be accepted..

• Reserved frame types (4, 5, 6 and 7) are only accepted when:

• FRMFILT1.ACCEPT\_FT4TO7\_RESERVED = 1 (default is 0)

• Length byte >= 9

The following operations are performed before the filtering begins, with no effect on the frame data stored in the RX FIFO:

• Bit 7 of the length byte is masked out (don’t care).

• If FRMFILT1.MODIFY\_FT\_FILTER is unlike zero, the MSB of the frame type subfield of the FCF is either inverted or forced to 0 or 1.

**WWW.TI.COM** 75

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

If a frame is rejected, CC2520 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. Note that rejected frames can generate RX overflow if it occurs before the frame is rejected.

**Exceptions**

When frame filtering is enabled and the filtering algorithm accepts a received frame, an RX\_FRM\_ACCEPTED exception will be generated. It will not be generated if frame filtering is disabled or RX\_OVERFLOW or RX\_FRM\_ABORTED is generated before the filtering result is known.

Figure 24 illustrates the three different scenarios (not including the overflow and abort error conditions). 















**Figure 21: Filtering scenarios (exceptions generated during reception)**

The FSMSTAT1.SFD register bit will go high when start of frame delimiter is completely received and remain high until either the last byte in MPDU is received or the received frame has failed to pass address recognition and been rejected.

SFD exception can be routed to a GPIO pin alone or as a part of a group of exceptions in channel A or B. SFD exception should preferably be connected to a timer capture pin on the microcontroller to extract timing information of transmitted and received data frames. SFD exception is also stored in EXCFLAG1 register. The register bit (and possibly the GPIO pin) will go high when the start of frame delimiter has been completely received and will continue to be high until cleared by SW.

**Tips and Tricks**

The following register settings must be configured correctly:

• FRMFILT0.PAN\_COORDINATOR must be set if the device is a PAN coordinator, and cleared if not. • FRMFILT0.MAX\_FRAME\_VERSION must correspond to the supported version(s) of the IEEE 802.15.4 standard.

• The local address information must be loaded into RAM.

To completely avoid receiving frames during energy detection scanning, set FRMCTRL0.RX\_MODE = 0b11 and then (re)start RX. This will disable symbol search and thereby prevent SFD detection. To resume normal RX mode, set FRMCTRL0.RX\_MODE = 0b00 and (re)start RX.

During operation in a busy IEEE 802.15.4 environment, CC2520 will receive large numbers of non-intended acknowledgment frames. To effectively block reception of these frames, use the

FRMFILT1.ACCEPT\_FT2\_ACK bit to control when acknowledgment frames should be received:

76 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

• Set FRMFILT1.ACCEPT\_FT2\_ACK after successfully starting a transmission with acknowledgment request, and clear the bit again after the acknowledgment frame has been received, or the timeout has been reached.

• Keep the bit cleared otherwise.

It is not necessary to turn off the receiver while changing the values of the FRMFILT0/1 registers and the local address information stored in RAM. However, if the changes take place between reception of the SFD byte and the source PAN ID (i.e. between the *SFD and RX\_FRM\_ACCEPTED* exceptions), the modified values must be considered as don’t care for that particular frame (CC2520 will use either the old or the new value).

Note that it is possible to make CC2520 ignore all IEEE 802.15.4 incoming frames by setting MDMTEST1.MODULATION\_MODE=’1’.

**20.3.3 Source Address Matching**

CC2520 supports matching of the source address in received frames against a table stored in the on-chip memory. The table is 96 bytes long, and hence it can contain up to:

• 24 short addresses (2 + 2 bytes each)

• 12 IEEE extended addresses (8 bytes each).

Source address matching will only be performed when frame filtering is also enabled, and the received frame has been accepted. The function is controlled by:

• The SRCMATCH, SRCSHORTEN0, SRCSHORTEN1, SRCSHORTEN2, SRCEXTEN0, SRCEXTEN1 and SRCEXTEN2 registers

• The source address table in RAM.

**Applications**

*Automatic acknowledgment transmission with correct setting of the frame pending bit:* When using indirect frame transmission, the devices will send data requests to poll frames stored on the coordinator. To indicate whether it actually has a frame stored for the device, the coordinator must set or clear the frame pending bit in the returned acknowledgment frame. On most 8- and 16-bit MCUs, however, there is not enough time to determine this, and so the coordinator ends up setting the pending bit regardless of whether there are pending frames for the device (as required by IEEE 802.15.4 [2]). This is wasteful in terms of power consumption, because the polling device will have to keep its receiver enabled for a considerable period of time, even if there are no frames for it. By loading the destination addresses in the indirect frame queue into the source address table and enabling the AUTOPEND function, CC2520 will set the pending bit in outgoing acknowledgment frames automatically. This way the operation is no longer timing critical, as the effort done by the microcontroller is when adding or removing frames in the indirect frame queue and updating the source address table accordingly.

*Security material look-up:* To reduce the time needed to process secured frames, the source address table can be set up so the entries match the table of security keys on the microcontroller. A second level of masking on the table entries allows this application to be combined with automatic setting of the pending bit in acknowledgment frames.

*Other applications:* The two previous applications are the main targets for the source address matching function. However, for proprietary protocols that only rely on the basic IEEE 802.15.4 frame format, there are several other useful applications. For instance, by using it together with the exception binding mechanism, it is possible to create firewall functionality where only a specified set of nodes will be acknowledged.

**The Source Address Table**

The source address table begins at address 0x380 in RAM as shown in Figure 11. The space is shared between short and extended addresses, and the SRCSHORTEN0/1/2 and SRCEXTEN0/1/2 registers are used to control which entries are enabled. All values in the table are little-endian (as in the received frames).

• A *short address entry* starts with the 16-bit PAN ID followed by the 16-bit short address. These entries are stored at address 0x380 + (4 ⋅ n), where n is a number between 0 and 23.

**WWW.TI.COM** 77

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

• An *extended address entry* consists only of the 64-bit IEEE extended address. These entries are stored at address 0x380 + (8 ⋅ n), where n is a number between 0 and 11.

**Address Enable Registers**

Software is responsible for allocating table entries and for making sure that active short and extended address entries do not overlap. There are separate enable bits for short and extended addresses: • Short address entries are enabled in the SRCSHORTEN0, SRCSHORTEN1 and SRCSHORTEN2 registers. Register bit **n** corresponds to short address entry **n**.

• Extended address entries are enabled in the SRCEXTEN0, SRCEXTEN1 and SRCEXTEN2 registers. In this case register bit **2n** corresponds to extended address entry **n**. This mapping is convenient when creating a combined bit vector (of short and extended enable bits) to find unused entries. Moreover, when read, register bit **2n+1** will always have the same value as register bit **2n**, since an extended address occupies the same memory as two short address entries.

**Figure 22 - Example of enabled table entries** 

**Matching Algorithm**

The SRCMATCH.SRC\_MATCH\_EN bit controls whether source address matching is enabled or not. When enabled (which is the default setting) and a frame passes the filtering algorithm, the CC2520 will apply one of the algorithms outlined in Figure 22, depending on which type of source address is present.

The result is reported in two different forms:

• A 24-bit vector called SRCRESMASK contains a ’1’ for each enabled short entry with a match, or two ’1’s for each enabled extended entry with a match (the bit mapping is the same as for the address enable registers upon read access).

• A 7-bit value called SRCRESINDEX:

• When no source address is present in the received frame, or there is no match on the received source address:

• Bits 6:0: 0x3F

• If there is a match on the received source address:

• Bits 4:0: The index of the first entry (i.e. the one with the lowest index number) with a match, 0- 23 for short addresses or 0-11 for extended addresses.

• Bit 5: ’0’ if the match is on a short address, ’1’ if the match is on an extended address. • Bit 6: The result of the AUTOPEND function

78 **WWW.TI.COM**

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER**

**SWRS068 – DECEMBER 2007**

| **Short Source Address (mode 2)**  The received source PAN ID is called srcPanid. The received short address is called srcShort.  SRCRESMASK = 0x000000;  SRCRESINDEX = 0x3F;  for (n = 0; n < 24; n++) {  bitVector = 0x000001 << n;  if (SRCSHORTEN & bitVector) {  if ((panid[n] == srcPanid) && (short[n] == srcShort)) { SRCRESMASK |= bitVector; if (SRCRESINDEX == 0x3F) { SRCRESINDEX = n;  }  }  }  } | **Extended Source Address (mode 3)** The received extended address is called srcExt.  SRCRESMASK = 0x000000;  SRCRESINDEX = 0x3F;  for (n = 0; n < 12; n++) {  bitVector = 0x000003 << (2\*n); if (SRCEXTEN & bitVector) {  if (ext[n] == srxExt) {  SRCRESMASK |= bitVector; if (SRCRESINDEX == 0x3F) { SRCRESINDEX = n | 0x20; }  }  }  } |
| --- | --- |

**Figure 23 - Matching algorithm for short and extended addresses**

SRCRESMASK and SRCRESINDEX are written to CC2520 memory as soon as the result is available. SRCRESINDEX is also appended to received frames if the FRMCTRL0.AUTOCRC and FRMCTRL0.APPEND\_DATA\_MODE bits have been set. The value then replaces the 7-bit LQI value of the 16-bit status word.

**Exceptions**

When source address matching is enabled and the matching algorithm completes, a *SRC\_MATCH\_DONE* exception will be generated, regardless of the result. If a match is found, a *SRC\_MATCH\_FOUND* exception will also be generated, immediately before *SRC\_MATCH\_DONE*.

Figure 24 illustrates the timing of these exceptions:



















**Figure 24 - Exceptions generated by source address matching**

**WWW.TI.COM** 79

**CC2520 DATASHEET**

**2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER SWRS068 – DECEMBER 2007**

**Tips and Tricks**

• The source address table can be modified safely during frame reception. If one address replaces another while the receiver is active, the corresponding enable bit should be turned off during the modification. This will prevent CC2520 from using a combination of old and new values, because it will only consider entries that are enabled throughout the whole source matching process.

The following measures can be taken to avoid that the next received frame overwrites the results from source address matching:

• Use the appended SRCRESINDEX result instead of the value written to RAM (this is the recommended approach).

• Read the results from RAM before RX\_FRM\_ACCEPTED occurs in the next received frame. For the shortest frame type this will happen after the sequence number, so the total available time (absolute worst-case with a small safety margin) becomes:

*16 µs (required preamble) + 32 µs (SFD) + 128 µs (4 bytes) = 176 µs*

• To increase the available time, clear the FSMCTRL.RX2RX\_TIME\_OFF bit. This will add another *192 µs*, for a total of *368 µs*. This will also reduce the risk of RX overflow.

**20.3.4 Frame Check Sequence**

In receive mode the FCS is verified by hardware if FRMCTRL0.AUTOCRC is enabled. 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 RX FIFO during receive. Instead, when FRMCTRL0.AUTOCRC is set the two FCS bytes are replaced by other more useful values. Which values that are substituted for the FCS sequence is configurable in the FRMCTRL0 register.





**Figure 25: Data in RX FIFO for different settings.** 

Field descriptions:

• The *RSSI* value is measured over the first 8 symbols following the SFD.

• The *CRC\_OK* bit indicates whether the FCS is correct (1) or incorrect (0). When incorrect, software is responsible for discarding the frame.

• The *correlation value* is the average correlation value over the 8 first symbols following the SFD. • *SRCRESINDEX* is the same value that is written to RAM after completion of source address matching.

Calculation of the LQI value used by IEEE 802.15.4 is described in section 20.5.

80 **WWW.TI.COM**