UNIVERSITY OF CALIFORNIA AT BERKELEY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

CHECKPOINT 3
Wireless Transceiver

1.0 MOTIVATION

The wireless, radio-frequency (RF) communication link provided by the Chipcon CC2420 allows your device to communicate with other 802.15.4 devices in the area. The 802.15.4 protocol is an official IEEE standard designed for low data-rate personal area networks (PANs). It operates on the same unlicensed 2.4 GHz band as 802.11 (wireless LAN). Sixteen channels are available, each of which supports a maximum data rate of 250 kbps.

The protocol has found its way into such applications as industrial sensors and "smart-home" devices. Many of these devices use Zigbee, a protocol built on the foundation of 802.15.4. While the CC2420 is Zigbee ready, we will not be sending packets that conform to the standard. For additional reading, refer to: Wikipedia: Zigbee and Wikipedia: 802.15.4.

This checkpoint will allow your project to communicate with the completed project of any other group in the class. Interoperation makes this checkpoint more challenging than all the others, as the digital designer will have to follow a strict set of guide-lines in order to send and receive data.

2.0 INTRODUCTION

This checkpoint will introduce you to several key ideas in digital design:

1. General interfacing with a third-party chip.
2. Serial interfaces.
3. Data buffering.
4. Communication protocols.

This checkpoint documentation explains the basics of using the CC2420. *However, you will not be able to complete the checkpoint without referring to the datasheet.*

2.1 CC2420 Interface

The CC2420 has 33 16-bit configuration registers, 15 command strobe registers, an 128-byte transmit (TX) RAM, an 128-byte receive (RX) RAM, and an 112-byte security RAM. The TX and RX RAM can be accessed by address or accessed through two 2-bit registers, in which case the memory acts as FIFO buffers. We will *not* be writing or reading any data from the security RAM and will not be accessing the TX and RX RAM as memory, only as FIFOs.

Interfacings to the registers occur over a serial peripheral interface (SPI), also referred to as a 3-wire interface, described below. In additional using the SPI wires, it is also necessary to observe the signal on the FIFO, FIFOP, SFD, and CCA wires and to drive the VREG_EN and RF_RESET_ wires.
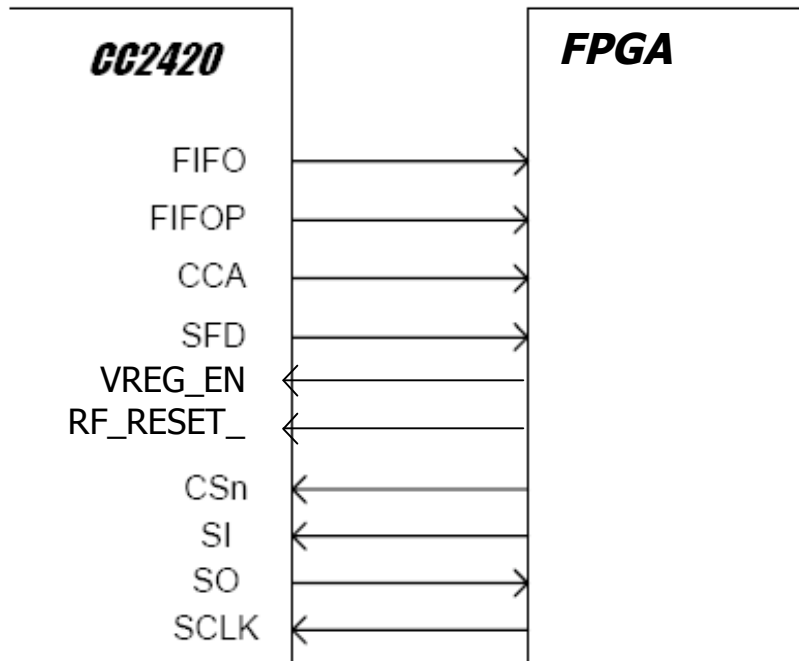
2.1.1 Serial Interface: SPI

A digital system can have numerous devices that are controlled by a single controller. In order to decrease cost and complexity, it is desirable to interface to those devices serially over a single bus.

The CC2420 uses the SPI interface, which has 4 wires: SCLK, CS_, SI, and SO. The CS_ (chip select) is an active low signal, which means that the signal is active when the wire is driven to low logic level. For comparison, the typical RESET signal is active high, since it is active when driven to high logic level. Use of the interface is as follows:

1. Drive CS_ low, in order to indicate to the CC2420 that a new SPI session has started.
2. After the chip has been "selected," begin driving SCLK clock signal. SCLK does *not* have to be driven at a constant frequency and *can* have a variable duty cycle. At each positive edge of SCLK, the CC2420 samples the data on SI. On each negative edge of SCLK, the CC2420 will change the data on SO if it is operating in a mode in which data is being output.
3. At the conclusion of the session, stop driving SCLK and bring CS_ back high.
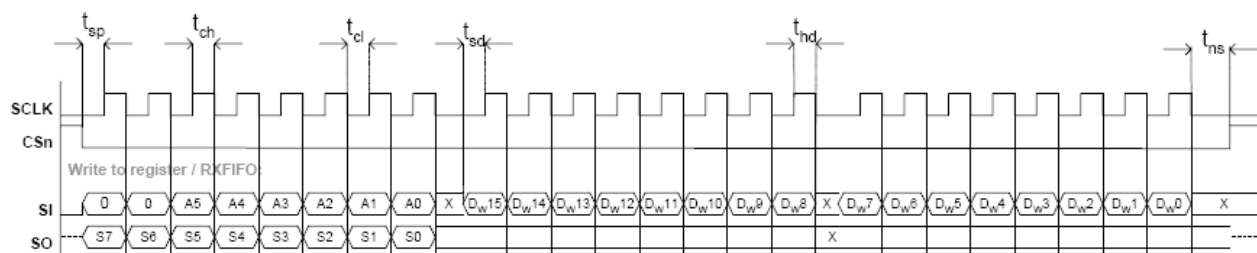
The diagram above summarizes a "write to register / RXFIFO" SPI session. It is *imperative* to meet all the timing constraints summarized in the table below. Details of what to send and receive in each SPI session follow.

Parameter	Symbol	Min	Max	Units	Conditions
SCLK, clock frequency	F_{SCLK}		10	MHz	
SCLK low pulse duration	t_{cl}	25		ns	The minimum time SCLK must be low.
SCLK high pulse duration	t_{ch}	25		ns	The minimum time SCLK must be high.
CSn setup time	t_{sp}	25		ns	The minimum time CSn must be low before the first *positive* edge of SCLK.
CSn hold time	t_{ns}	25		ns	The minimum time CSn must be held low after the last *negative* edge of SCLK.
SI setup time	t_{sd}	25		ns	The minimum time data on SI must be ready before the *positive* edge of SCLK.
SI hold time	t_{hd}	25		ns	The minimum time data must be held at SI, after the *positive* edge of SCLK.
Rise time	t_{rise}		100	ns	The maximum rise time for SCLK and CSn
Fall time	t_{fall}		100	ns	The maximum fall time for SCLK and CSn

The first 8 bits sent at the beginning of each SPI session *always* has the following format:

Sent First ←	Bit Position	→ Sent Later
7	6	5:0
1 = RAM access (not used) 0 = register access	1 = read 0 = write	Address of register. Refer to p. 60 of the datasheet.

At the same time that these first 8 bits are sent, the transceiver replies on the SO line with the 8 status bits. For more information, refer to p. 27 of the datasheet. While it is not necessary to read the status bits, it may be useful for debugging purposes. The bits that follow the first byte vary in length, depending on the register access that is being performed. Refer to the register section for details.

2.1.2 Chipcon Registers and FIFOs
The configuration, command strobe, and FIFO access registers are accessed via the SPI interface. Refer to the SPI section above on the general usage of the serial interface; the specifics on what data to output on the SI line and read from the SO line are explained in each section below.

2.1.2.1 Configuration Registers
The configuration registers contain information about the operation mode of the CC2420. A full list of registers begins on p. 61 of the datasheet.

Register	Address	Bit(s) of Interest	Purpose
MDMCTRL0	0x11	11	The CC2420 supports automatic address recognition, which requires the use of standard Zigbee frames. For more information, refer to p. 39 of the datasheet. You will have to implement address checking on the FPGA. You must set bit 11 to 1'b0.
FSCTRL	0x18	9:0	The CC2420 supports 16 different communication channels, which have a center frequency ranging from 2405 – 2480 MHz. For more information, refer to p. 49 of the datasheet. You must set bits 9:0 to $357 + 5(k-11)$, where the channel number k ranges from 11 to 26.
IOCFG0	0x1C	6:0	Sets the number of bytes that RX FIFO needs to contain before FIFOP will go high. You will need to modify this!

To write to a configuration register, the first byte sent over the SPI is always followed by the 2 bytes of data to be saved into the register. While these 2 data bytes are written, the data on SO is invalid.

	Sent First ← Byte Number → Sent Later	
	1	2 - 3
Sent on SI	address byte, described above	16 bits of data to be written to register
Received on SO	status byte	16'bX

Reading from a configuration register is similar to reading from the RX FIFO (below), except that the address byte is always followed by the reception on SO of the 2 bytes contained in the register.

2.1.2.2 Command Strobe Registers
The command strobe registers change the state of the CC2420. A full list of the purpose of the registers begins on p. 60 of the datasheet.

Register	Address	Purpose
SNOP	0x00	No operation. Used to check the status of the CC2420, since it always replies to an address with status bits. Note that this is *not* the only way to check the status (e.g. the CC2420 replies with the status byte for the first byte sent during all SPI sessions).
SXOSCON	0x01	Turns on the crystal oscillator and will be used as part of the initialization and channel changing process.

SRXON	0x03	Moves the CC2420 into the receive state and will be used as part of the initialization process.
STXON	0x04	Instructs the CC2420 to transmit the data contained in the TX FIFO.
SRFOFF	0x06	Turns off RX/TX and frequency synthesizer and will be used as part of channel changing.
SFLUSHRX	0x08	Flushes the RX FIFO.

To issue a command strobe, send the command strobe address as the only byte of a SPI session.

Byte Number	
	1
Sent on SI	address byte, described above
Received on SO	status byte

2.1.2.3 FIFO Access Registers
The TX and RX access registers allows use of the TX and RX RAM as FIFO buffers. We will not be accessing the RAM in any other way.

Register	Address	Purpose
TXFIFO	0x3E	All bytes contained in the TX FIFO will be transmitted upon issuance of the STXON command strobe. You *must not* write data to the FIFO while a transmission is in progress, as indicated by the SFD signal. For more information, refer to p. 37 of the datasheet.
RXFIFO	0x3F	The RX FIFO is automatically filled by the CC2420 when it detects valid packets on the transmission medium.

To write to the TX FIFO register, the first byte sent over the SPI is followed by the bytes that are to be saved into the register. The session only ends when CS_ is driven high; i.e. you can write multiple bytes in a single SPI session, as long as the FIFO does not overflow. While each byte is written, the status byte is returned on the SO line.

Sent First ← Byte Number → Sent Later	
1	2 to n
Sent on SI address byte, described above	data bytes to be transmitted
Received on SO	status byte

To read from the RX FIFO register, the first byte sent over the SPI is followed by don't cares on the SI line. While the don't cares are written, data contained in the FIFO is returned on the SO line. The session only ends when CS_ is driven high; i.e. you can read multiple bytes in a single SPI session, as long as the FIFO does not underflow.

Received First ←	Byte Number →	Received Later
	1	2 to n
Sent on SI	address byte, described above	8'bX
Received on SO	status byte	data from the RX FIFO

2.1.3 Single Bit Control Signals

Signal Name	Purpose
VREG_EN	Drive this signal high to enable the voltage regulator on the CC2420.
RF_RESET_	Pulse this signal (active low) to reset the CC2420.
FIFO	This signal goes high when there is at least one unread byte in the RX FIFO. It will go low while the last byte is being read out of the FIFO.
FIFOP	This signal goes high when the number of unread bytes in the RX FIFO exceeds IOCFG0.FIFOP_THR, which defaults to 64. It also goes high when the last byte of a new packet is received, even if the threshold is not exceeded. In this case, it goes low after one byte has been read from the FIFO.
SFD	On receive, SFD goes high after a valid start of frame delimiter (SFD) is detected and goes low once the entire packet is received. On transmit, SFD goes high after the SFD is transmitted and goes low once the entire packet is transmitted.
CCA	As long as the CC2420 has been in receive mode for 8 symbol periods, the CCA will be high if the channel is clear and low otherwise. The default CCA mode should be sufficient. Refer to p. 49.

FIFO low and FIFOP high indicate an RX underflow (attempt to read data from an empty FIFO) or RX overflow. When this happens, the CC2420 can only resume regular operation after the SFLUSHRX command strobe is issued twice. For more information, refer to p. 31 of the datasheet. There are many situations that will cause an underflow or overflow to occur. *Your design must be able to recover!*

2.2 TX AND RX
Before creating your design, make sure you have a thorough understanding of the CC2420 internal state machine on p. 43 of the datasheet.

2.2.1 Data Frame Format
For simplicity, the official Zigbee frame format will *not* be used. The frames used for the project will be of the following format. When the CC2420 is in the receive mode, it begins reception of a new frame of data after detecting the preamble and SFD.

			MPDU			
Preamble	Start of Frame Delimiter (SFD)	Length	Source Address	Destination Address	Payload	Frame Check Sequence (CRC)
4 bytes	1 byte	1 byte	1 byte	1 byte	41 bytes	2 bytes
0x00	0x7A	length of MPDU	sender's address	recipient's address or 0xFF for broadcast	data	• On transmit, 0x00. • On receive, bit 7 of the 2nd byte is 1 when CRC ok, 0 otherwise.

On transmission of a data frame, the CC2420 automatically appends the preamble and SFD to the beginning of the packet and the CRC to the end of the packet.

2.2.2 TX
Perform the following sequence to transmit data:

1. Save the following data, in order, into the TX FIFO:
    a. Length of the MPDU part of the frame measured in BYTES.
    b. Your address.
    c. Recipient's address or 0xFF for broadcast.
    d. Forty-one bytes of data to be transmitted, starting from the MSB
    e. Two bytes of 0x00, which the CC2420 will automatically replace with CRC bits.
2. Check the CCA signal and do not proceed unless the channel is clear. Regardless of which method you choose in order to perform CCA, it is required that you retry after a waiting period of "**randomly**" varying length. Also CCA may not go high due to buffer overflow.
    a. Your random number should be a random 14-bit number that is unique to every individual running implementation.
3. Issue STXON command strobe.
4. Make sure the SFD line goes high then low *and wait for at least 60 clock cycles* before saving any new data to the TX FIFO.
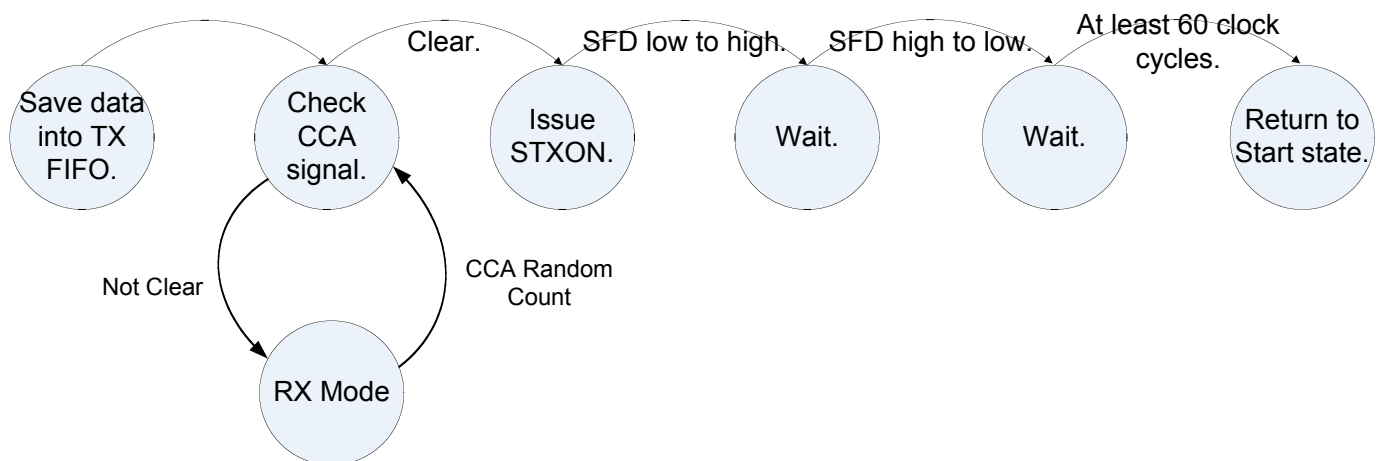
2.2.3 RX

On receipt of a data frame, the CC2420 automatically *calculates* the packet CRC. You will need to manually *check* the CRC bit in the last byte of the received frame. The first byte saved to the RX FIFO is the length byte.

The CC2420 will not receive any data until it has been in RX mode for 12 symbol periods. You *must* incorporate wait time between packet transmissions in order to allow the CC2420 to look for and receive data.

Perform the following sequence to receive data:

1. Check the FIFO and FIFOP signals to see if new data has arrived.
2. If there is new data, begin reading from the RX FIFO.
   a. The first byte will be the length byte. Save it, as it will be the only indication of where the frame ends. Immediately flush the RX FIFO if the length is not correct.
   b. Receive the source address and check if it matches the expected sender. Optionally discard the *entire* frame if the address does not match. Look in section 4.4 for details on address checking.
   c. Receive the destination address and check if it matches your address or if it is broadcast, 0xFF. Optionally discard the *entire* frame if the address does not match.
   d. Receive and save the 41 bytes of payload.
   e. Receive the 2 bytes of CRC information. Discard the previously saved payload if CRC does not check.
3. If an RX FIFO underflow or overflow occurs at any time, immediately flush the FIFO.
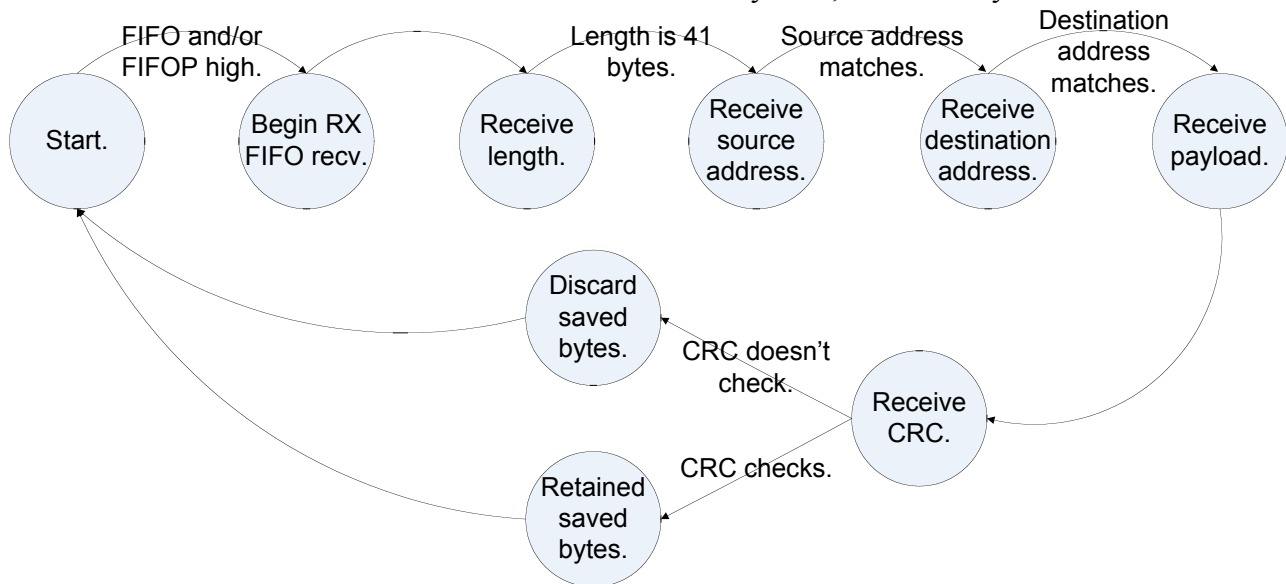
3.0 PRELAB

1. Examine the documents page of the website. You will need to get used to reading datasheets. They form the core of information available to hardware designers.
   a. http://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf
2. Examine the Verilog skeleton provided for this checkpoint.
   a. There isn't much, so it should be pretty clear.
3. Start your design ahead of time.
   a. Begin with schematics and bubble-and-arc diagrams.
   b. Come prepared for your design review. Make sure you understand the different commands that that need to be issued over SPI.
   c. The entire checkpoint will require significant debugging in simulation and ChipScope. Make sure to at least write a draft of it ahead of time.
      i. Build a thorough testbench. Errors take much longer to debug in hardware than in software.

4.0 LAB PROCEDURE

The best approach to designing a complex digital system is to abstract away details into many layers of operation. The interface to the CC2420 can be abstracted away into three distinct layers.

1. The SPI module (provided) deals with the timing of the SPI interface and simplifies the interface to parallel input and output ports. It does not, however, contain any intelligence about the format of the data to be sent to the CC2420. It is a generic SPI module that can be reused when interfacing with any other chip that uses the 3-wire interface.
2. An intermediate module drives the SPI module with data that is specific to the CC2420. This module should abstract away the details of what to send and receive on the SPI and when to send and receive it.
3. The Transceiver module is the highest level module. It presents the user with two 320-bit data busses for data transmission and receipt.

Use your creativity and ingenuity to design an efficient system for accomplishing the task. **You must adhere to the input/output specifications for the Transceiver module.** Since all other modules are instantiated within Transceiver, it is not necessary for your solution to have input/output specifications identical to what is listed below.

Remember, there are many equally good ways of structuring this checkpoint.

4.1 Handshaking

Each layer of the interface communicates with the other layers of the checkpoint by using standard handshaking methods. Two methods are discussed below. The high-level transceiver module uses the ready/start method for both input and output. Either method can be used for other parts of the design.

4.1.1 InRequest/InValid Handshaking

The master device pulses the InRequest line whenever it is requesting data. If the slave device has new data, it responds to the InRequest by placing data on the In line and driving InValid high. InValid and In remain constant until the next InRequest. If the slave has new data, it keeps driving InValid high and changes In. If the slave does not have new data, it drives InValid low and the value on In no longer matters.
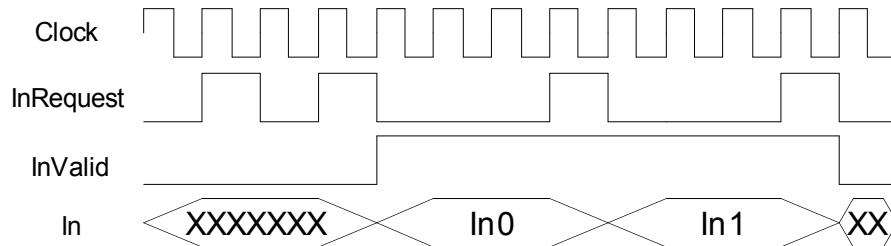
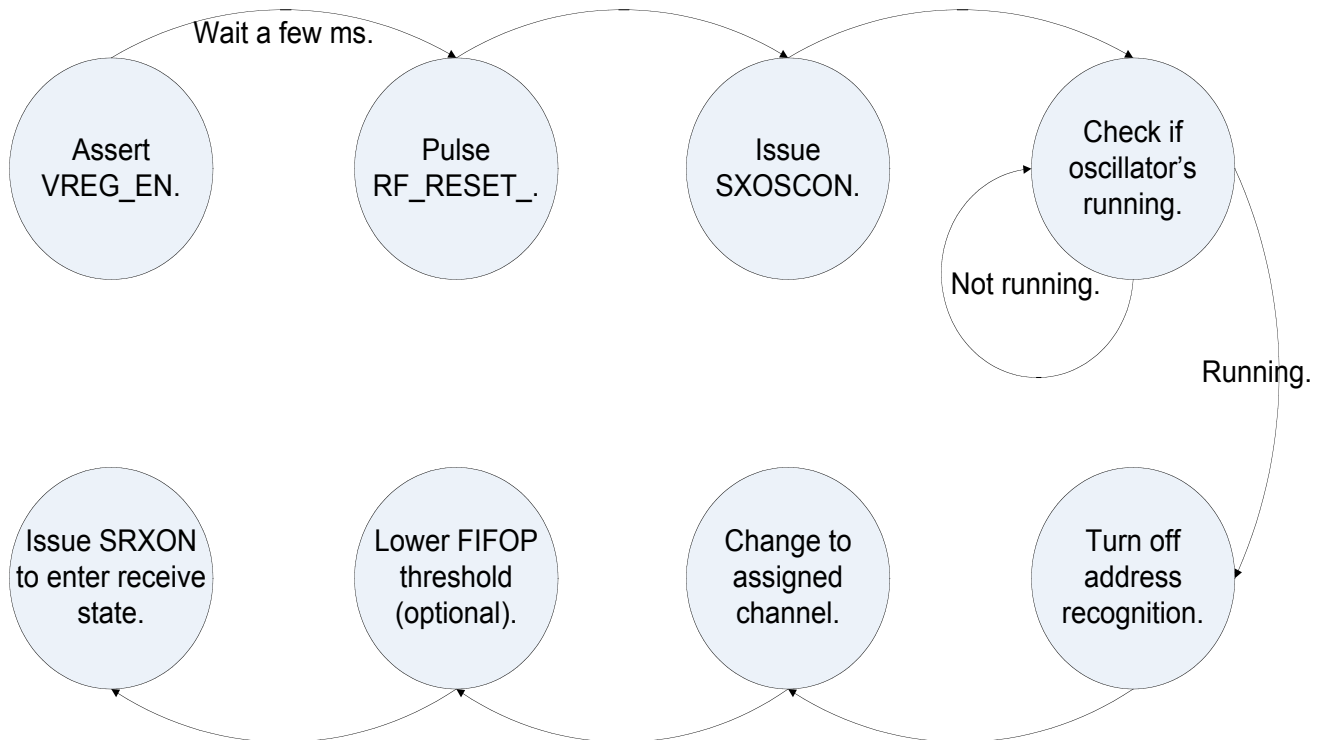### 4.1.2 Ready/Start Handshaking

The master device raises the Ready flag when it is ready to receive data. The slave puts the input on the In line, then pulses Start for one clock cycle. On the next cycle, Ready is driven low unless the master is ready to receive the next chunk of data. The data on the In line should remain constant until Ready is reasserted.
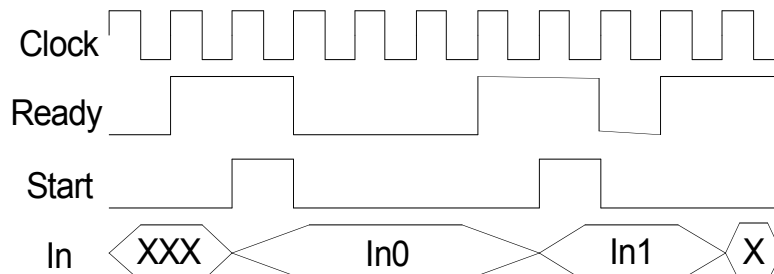
4.2 Initialization

Perform the following initialization steps at start-up:

1. Assert VREG_EN and keep it asserted forever. Wait a few ms before proceeding.
2. Pulse RF_RESET_. Remember that it is active low and that pulse width may need to be longer than 1 period of the 27 MHz clock.
3. Issue SXOSCON command strobe, and then repeatedly check for status bit 6 high (oscillator running) before proceeding.
4. Disable address recognition.
5. Optionally lower the threshold for FIFOP.
6. Change to default channel for your group. *Your group must perform this channel change properly!*
7. If you do not transmit immediately, issue SRXON to enter the receive state.

Wait a few ms.

(Assert VREG_EN.) → (Pulse RF_RESET_.) → (Issue SXOSCON.) → (Check if oscillator's running.)

Not running.

Running.

(Issue SRXON to enter receive state.) ← (Lower FIFOP threshold (optional).) ← (Change to assigned channel.) ← (Turn off address recognition.)

4.3 Channel Changing

Your code must be capable of changing the CC2420 operation channel without reset and re-initialization, for checkpoint 4. After changing the frequency programmed in the FSCTRL configuration register, described above, it is necessary to issue the SRFOFF command strobe. If this is not done, channel changing will not always work. Remember to subsequently issue SRXON if you do not immediately transmit a packet with STXON.

4.4 Addressing

Each group has been assigned an 8-bit address space that they will use to send and receive packets. The first two bytes of the MPDU are the source and destination addresses. On transmission, the input on SrcAddr is transmitted as the source address and the input on DestAddr is transmitted as the destination address. Destination address 0xFF has been reserved for broadcast. On reception, the transceiver module needs to discard packets with addresses that do not match the criteria listed below.

MPDU			
Source Address	Destination Address	Payload	Frame Check Sequence (CRC)
1 byte	1 byte	41 bytes	2 bytes
sender's address	recipient's address or 0xFF for broadcast	data	• On transmit, 0x00. • On receive, bit 7 of the 2nd byte is 1 when CRC ok, 0 otherwise.

- Accept all packets with broadcast destination address 0xFF.
- When *your* DestAddr input is set to broadcast (i.e. you are transmitting broadcast packets), 0xFF, accept all packets with a destination address that matches *your* SrcAddr.
- In all other cases, accept packets only if the packet source address matches *your* DestAddr and the packet destination address matches *your* SrcAddr.

4.5 IO Registers

The signals FIFO, FIFOP, SFD, and CCA pass through IO register modules before being input into the transceiver. These registers are instantiated in FPGA_TOP in the provided checkpoint 3 framework. They contain synthesis directives that direct the CAD tools to place the registers at the "edge" of the FPGA. Without IO registers, it is possible to encounter hardware errors that result in erratic logic behavior.

4.6 Transceiver

This is the highest level module, which presents the user of the transceiver with a simplified interface. Although there are a large number of port connections, many of the signals may not be handled directly by the transceiver, but can be passed through to the modules that are instantiated within it.

Signal	Width	Dir.	Description
Wireless Data Transmission			
BigIn	320	I	Data to be transmitted over wireless.
Ready	1	O	Indicates that transceiver is ready to transmit data.
Start	1	I	Pulse high for 1 cycle to begin data transmission.
Wireless Data Receipt			
OutBig	320	O	Data received over wireless.
SenderAddr	8	O	Packet source address corresponding to the data on Out.
RecipientAddr	8	O	Packet destination address corresponding to the data on Out.
EndSession	1	O	This signal should indicate that the data on BigOut is now valid.
SPI Interface to CC2420			
CS_	1	O	Active low chip select.
SClk	1	O	SPI clock signal.
SI	1	O	SPI input.
SO	1	I	SPI output.
Single Bit Control Signals			
VReg_En	1	O	Drive high to turn on CC2420 voltage regulator.
Rf_Reset_	1	O	Active low reset to CC2420.
FIFO	1	I	Indicates a non-empty RX FIFO.
FIFOP	1	I	Indicates RX FIFO threshold exceeded or new packet received. Read above for more information.
SFD	1	I	Goes high when SFD sent/received and then low when packet sent/received. Read above for more information.
CCA	1	I	Goes high when the channel is clear. Low otherwise.
Other Communication Settings			
Channel	4	I	Transceiver channel.
SrcAddr	8	I	Packet sender's address.
DestAddr	8	I	Packet receiver's address.
Standard Synchronous Module Inputs			
Clock	1	I	27 MHz clock signal.
Reset	1	I	Resets entire module.

4.8 SPI

The SPI module is provided. Refer to the SPI section above for the interface timing that this module implements.
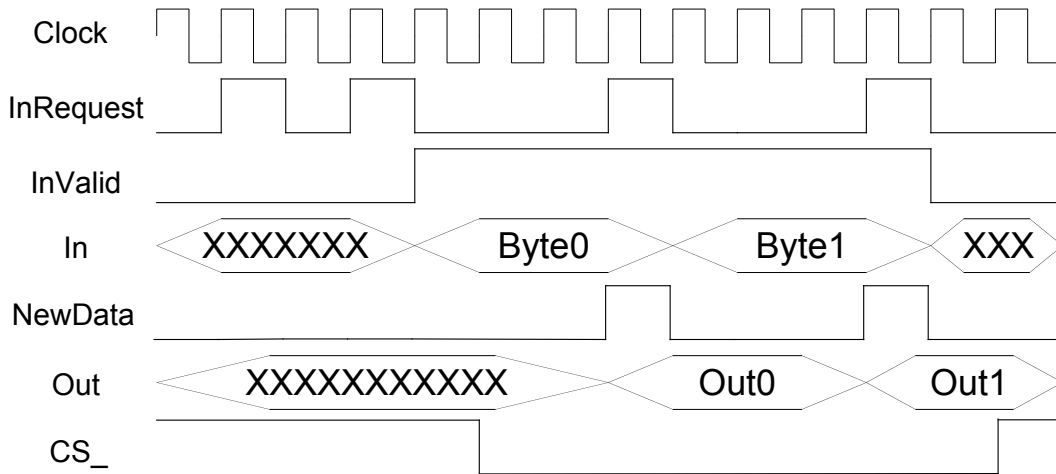
Note: More cycles pass between InRequest pulses than shown above!

Interfacing to this module is based on the InRequest/InValid handshaking described above. When SPI is not in use, it will pulse InRequest every other cycle. Once it is in use, it will pulse InRequest once it needs a new byte of data to transmit on SI. It transmits the data on In corresponding to InValid replies to consecutive InRequest pulses in a *single SPI session*.

Signal	Width	Dir.	Description
Data to CC2420			
In	8	I	Data to be transmitted to CC2420 over SPI.
InRequest	1	O	Indicates that SPI is ready for new data.
InValid	1	I	Indicates that there is new data on In.
Data from CC2420			
Out	8	O	Data received.
NewData	1	O	Indicator of new received data.
SPI Interface to CC2420			
CS_	1	O	Active low chip select.
SClk	1	O	SPI clock signal.
SI	1	O	SPI input.
SO	1	I	SPI output.
Standard Synchronous Module Inputs			
Clock	1	I	27 MHz clock signal.
Reset	1	I	Resets entire module.

5.0 Additional checkpoint Requirements

- You MUST be able to receive data while you're waiting for CCA.
- You must adhere to the CCA standard such that if you have 2 solutions communicating with each other they must have different random CCA wait times. (Hint: To achieve this you can take whatever your sudo-random number is and XOR it with your SRC address.)
- Do not modify the time between transmissions for this checkpoint. The channel will become congested if too many people try to transmit too much data on a single channel.