ECSE-426

I/O Interfacing – Serial Buses

Lab progress

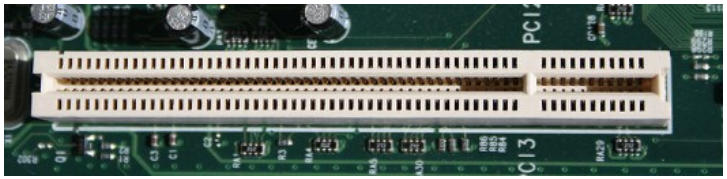
- Wish to keep the schedule
  - Meet requirements if possible...
- Next experiment will have more margin
  - More robust code for the final project
  - Less stress during midterms
- Keep in mind the Pareto principle
  - Applied to MicroP : 80 % of the results come from 20% of the effort
    - Unproven observation, but looks like it holds...
- Goals :
  - 1- Understand what you do
  - 2- Meet the requirements
  - 3- Add fluff, time permitting

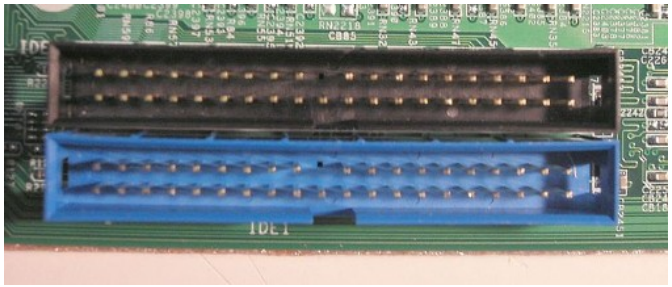
Today's lecture

- Serial Interfaces
  - Review of Serial Buses
  - Synchronous - I2C, SPI
- Understanding data sheets
  - Operating Conditions
  - Timing Diagrams
    - What to look for and why
- Course Specific
  - MSP430 USART (SPI Mode)
  - CC2500 Wireless module
  - Loopback Testing

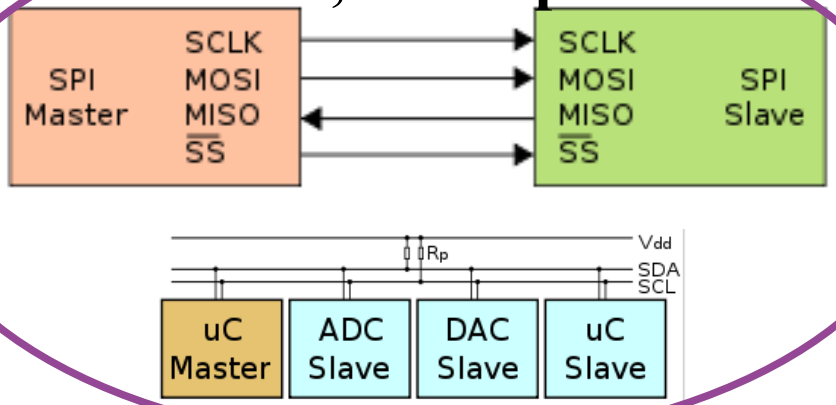
Computer Buses

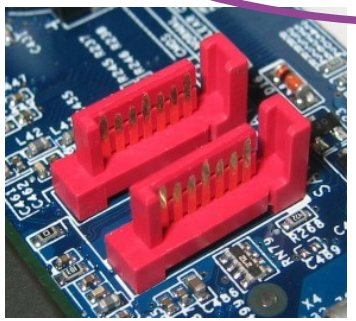
Parallel

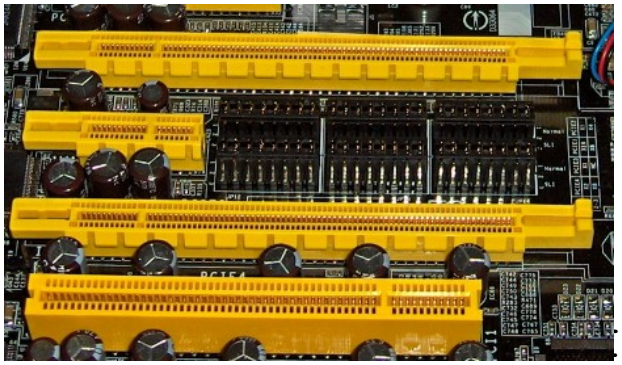




Serial, Low Speed







Source : Wikipedia

Review of Buses

- Recall from last lecture
- Parallel Buses
  - Limited clock rate due to skew, length
  - Scaling problem with bus width
    - 64 and 128 bits are pretty much the upper limit
  - Used for DRAM, flash and some peripherals in microcontroller systems
- High speed serial buses
  - Address the limits of parallel buses
    - Point to point links
    - Clock and data in the same differential pair
    - Much higher bandwidth **per pin**

Review of Buses (2)

- Expandable Serial Bus
  - Great example : PCI Express
  - Addresses scaling issues by using networking concepts
    - Switches, virtual channels, multiple layers of error correction
    - Link bandwidth is **negociated**
- Low-speed synchronous serial buses
  - Allows adding many peripherals to a system
  - Sensors, Memory, inter-chip communication
  - All this at a very low cost and low complexity

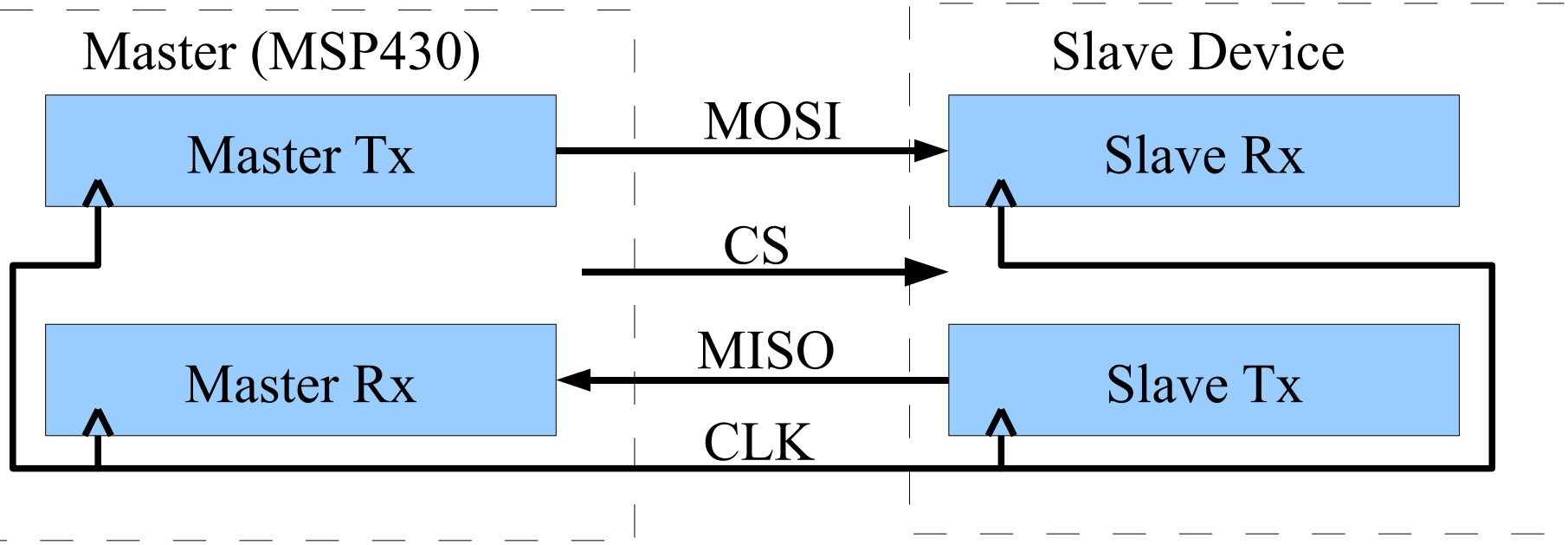
Synchronous Serial Buses

- Use a **clock line** to synchronize the data transfers
  - Contrast this with asynchronous UART mode
- Generally comprised of
  - One (sometimes multiple) master
  - One or many slaves
- Will usually break the problem in 2 abstraction levels.
  - Physical Level
    - Clocks (Polarity, Phase)
    - Control, Address, Data format
    - Chip Select(s)
  - Logical Layer (Driver layer)
    - Registers + FIFO memories
    - Status

Serial Peripheral Interface (SPI) Bus

- Very simple mechanism
  - Think of it as 2 shift registers in a chain
  - Chip Select => Enables the shift register
    - Often called nCS (Chip Select) of SS (Slave Select)
  - Clock => Clock Line
  - MOSI => Master Out Slave In
  - MISO => Master In Slave Out

SPI Interface – Physical Model

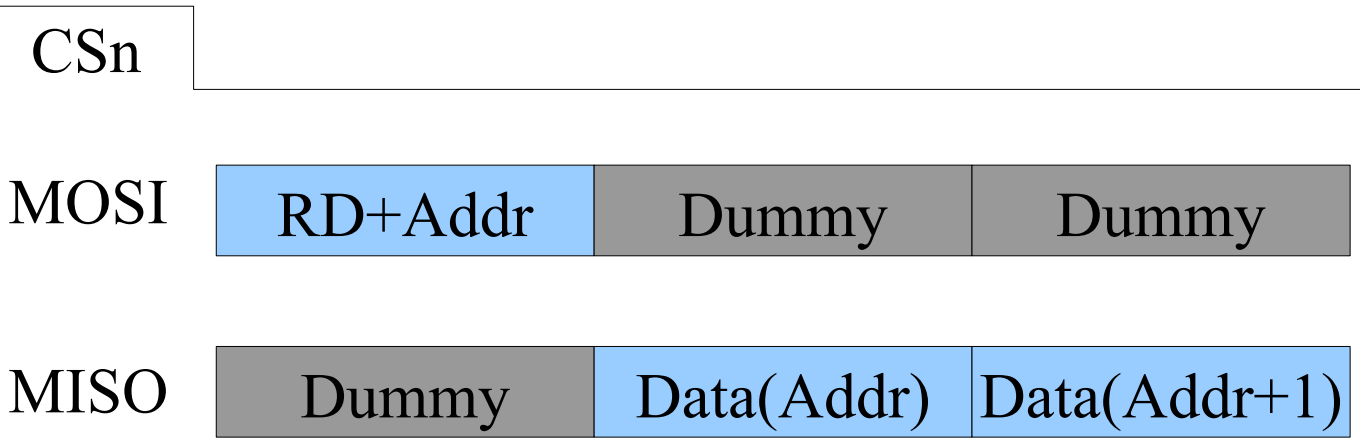


Can have multiple Slaves (one extra Chip Select per slave)
The transfer is done in 2 directions simultaneously

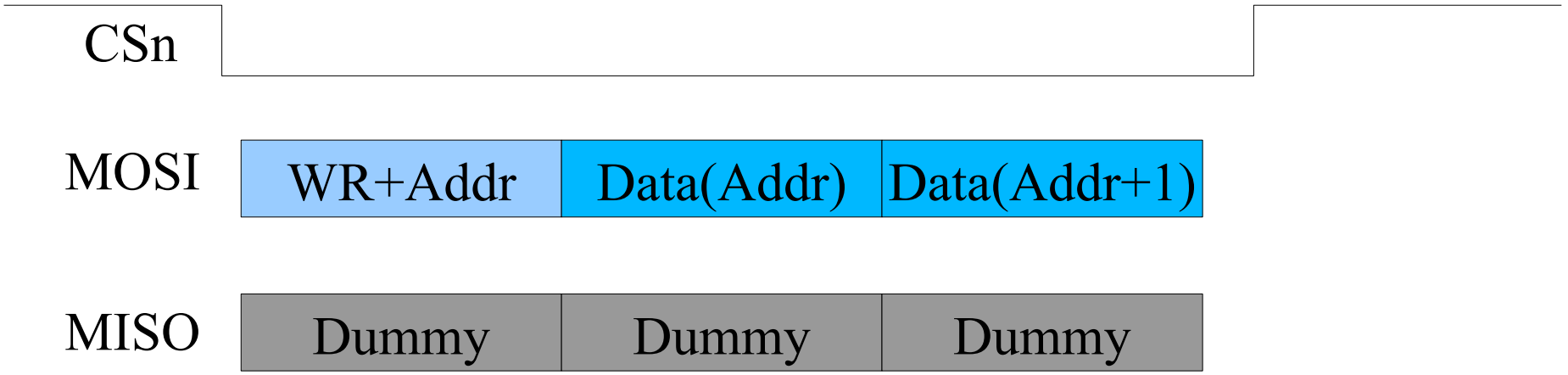
Transfer Details

- CSn : Active Low => Idles at '1'
  - Transition of CSn from '1' to '0' indicates start
- Typical Transfer are multiples of 8/16/32 bits
- Typically
  - First byte/word is command + Address
  - Then Data or “empty placeholder”
    - Recall that you have to send something to get something
  - Example: Read 2 bytes from Address 3
    - Send Read command + Address=3
    - Keeping CS low, send 2 bytes (anything...)
    - The initiator ignores the 1st word, then capture 2 bytes

Transfer Details - Read



Transfer Details - Write



Dummy Bytes ?

- Also called “don't care” bytes
  - You can send anything (all zero is common)
  - You can ignore the received value
    - However, to avoid overrun errors, you still need to read the receive register in the master device
- There's got to be a better use for those empty slots ?
  - Yes, they could carry “status” information
  - As in the case of your wireless transceiver
    - CC2500 from Texas Instruments
  - Every access you do to the slave gives you its status
    - If you just need the status, read any register...

Variations on SPI

- SPI is **not** a standard
  - More of an agreed upon method
  - Varies between devices
- You may encounter variations on SPI
  - Shared MOSI/MISO
    - Using tri-state drivers
    - Master drives, slave listens => Write
    - Master listens, slave drives => Read
    - Goal: Save one pin, reduce cost...
  - Other names
    - Synchronous Serial Interface
    - MicroWire (Motorola Trademark)

Other low speed serial interface - I2C

- Developed by Philips
- Uses only 2 wires
  - SDA : Serial Data, SCL : Serial Clock
  - Can support many slaves
    - Each slave has built-in, predefined address
  - Bidirectional communication
- Protocol is more elaborate
  - A few different states
  - Enumeration of slave devices

Example System Using I2C

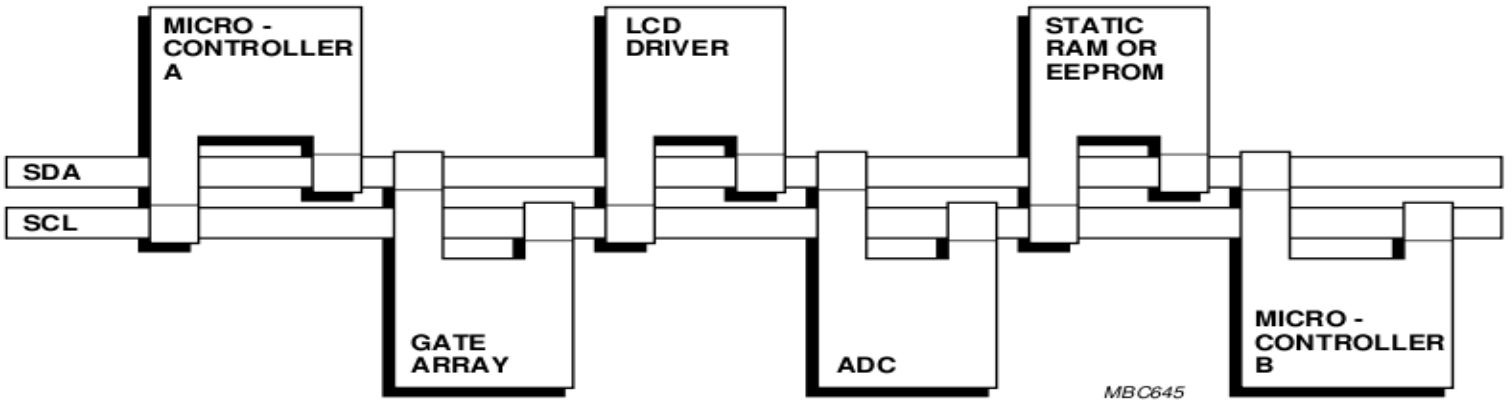


Fig.2 Example of an I²C-bus configuration using two microcontrollers.

- Multi-master is possible
- Easy to add peripherals
- Not as fast as SPI
  - 400 kBps and now 3.4 Mbps
  - Good enough for **a lot** of applications

The I2C-Bus Specifications Version 2.1, nxp.com, p.7

Where is I2C used ?

- Television Sets
  - Volume control, channel tuning
  - On-screen display
- Computers – Stricter specs => SMBus
  - Touch screen controllers
  - Power management (fans, voltage control)
- Embedded systems
  - Device Identification
  - ADC/DAC for process calibration
  - Configuration save/restore on EEPROMs

I2C Advantages

- Less complexity on the board
  - Cuts the cost of I/O connectors, chip, etc.
- Can be expanded
  - Simply connect the new slave on the two lines
- Software can enumerate devices
  - Each slave has unique identifier
  - Software can be made more generic
  - Configuration can be determined at run time
    - Contrast this with SPI : Designers need to plan ahead the chip select configuration

Comparison among serial buses

Pros and Cons of the different buses

UART	CAN	USB	SPI	I²C
• Well Known • Cost effective • Simple	• Secure • Fast	• Fast • Plug&Play HW • Simple • Low cost	• Fast • Universally accepted • Low cost • Large Portfolio	• Simple • Well known • Universally accepted • Plug&Play • Large portfolio • Cost effective
• Limited functionality • Point to Point	• Complex • Automotive oriented • Limited portfolio • Expensive firmware	• Powerful master required • No Plug&Play SW - Specific drivers required	• No Plug&Play HW • No “fixed” standard	• Limited speed

DesignCon 2003 TecForum I²C Bus Overview 27

nxp.com, AN10216-01 – I2C Manual, p12

Datasheets

How to deal with complexity

Datasheets - Structure

- Overview of the device features
  - Executive summary
  - Highlight of “best” characteristics
    - Always take this with a grain of salt
    - Valid for under certain conditions
    - Need to dig a bit for the exact details
- Device overview
  - Package Information, Pin Layout
- Pin information
  - Pin direction, I/O levels, Description, special considerations

Datasheets – Structure (2)

- Detailed Characteristics
  - Absolute Maximum Ratings
    - Respect those or the chip will be busted/damaged
  - Operating Conditions
    - This is how you should drive the chip if you want it to last...
  - General Characteristics
  - DC Conditions
    - Static information, logic levels
  - AC Conditions
    - Timing parameters, bus interface characteristics

Datasheets – Structure (3)

- Application Details
  - Register Information
  - Modes of operation
  - Feature Details and information
- Package Outline and information
  - Physical Dimensions
- Ordering Information
  - Device part number for a given footprint
  - Temperature rating, performance rating
- Packaging information
  - For mass production (tape, reels, etc.)

Datasheets - Considerations

- Not always a self-contained monolithic document
  - Replicated features in many devices
    - Result : Family guide + Device Specific datasheet
  - Physical packaging
    - Sometimes in a different document
- Datasheets are not the same as application notes
  - Datasheets = Concentrated essential information
  - Datasheets = Hard to understand
    - Application notes cover the “how to use” aspects
- No universal format for datasheets
  - They tend to follow a similar structure

Datasheet of the CC2500

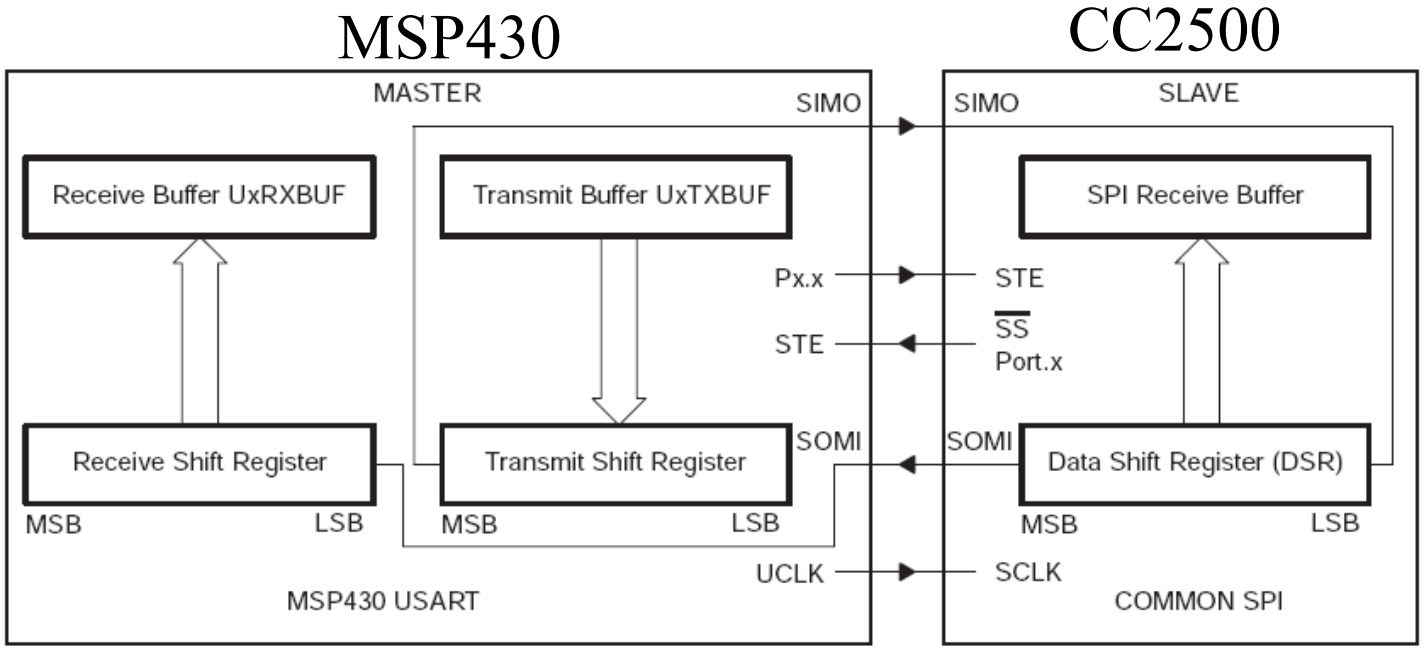
- Low power RF transceiver
  - Highly integrated
  - Many modulations possible
  - High sensitivity
    - Will pick up very faint signals
- Lets dig into this datasheet as an example

MSP430 USART – SPI Mode

Your communication path to the wireless world

MSP430 USART

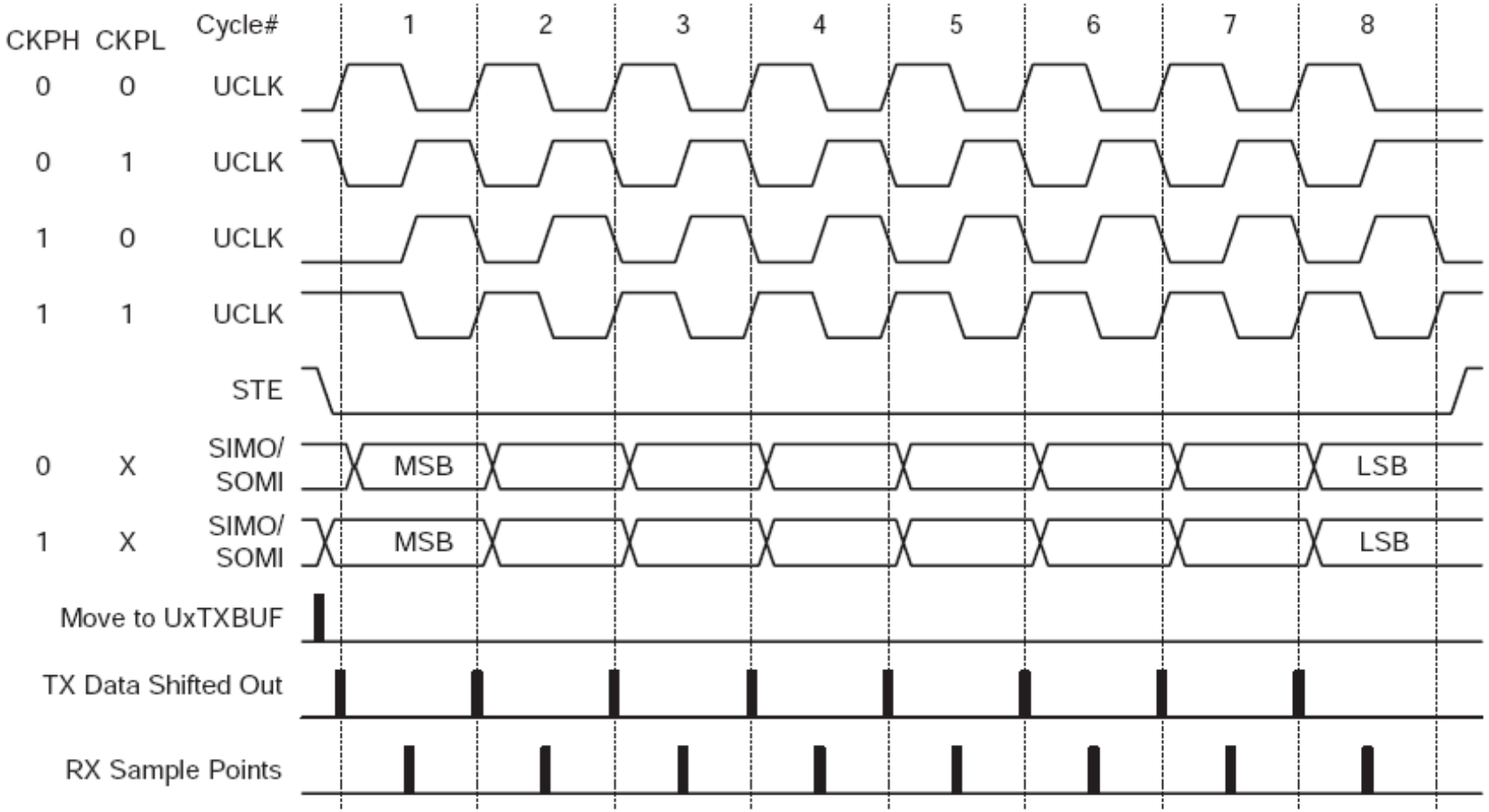
- SPI (Synchronous) mode
  - Chapter 14 of MSP430 Family User's Guide
  - MSP430 is **Master**



A few notes – STE and Clock

- STE
  - Allows another master on the bus
    - You have **only one master**... Make sure STE is driven properly.
- Serial Clock is provided by the USART baud rate engine
  - SPI clock can be quite fast
    - But should respect the receiver's timing
    - CC2500 is quite fast, but better check...

SPI Clock Phase and Polarity



SPI Modes

- You want to use the full SPI with bidirectional support
  - Called 4-wire SPI master mode
- What is 3-wire mode then ?
  - Only one data pin (bidirectional) between master and slave
  - Need to change direction dynamically
- Again, only one master on the bus, so check the STE pin...

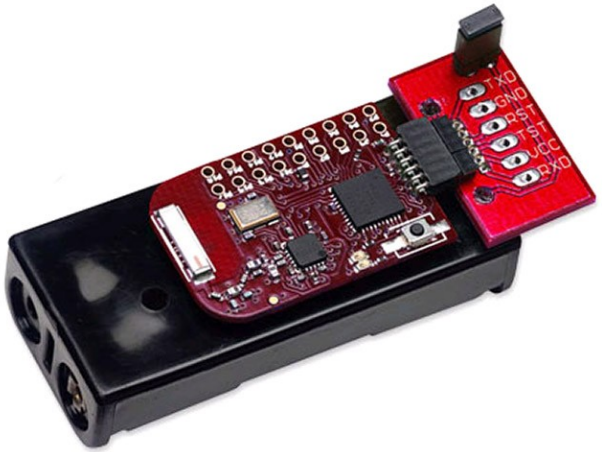
Lab Experiment #3

Overview

Wireless Transceiver Interfacing

CC2500 Transceiver

- Will be given as part of a kit
  - Kit include the MSP430F2xxx controller
    - We can ignore it – Program all its pins as inputs
  - We will concentrate on the CC2500
- You will connect the CC2500 to your McGumps kit via pin header



Putting it together - Tips

- Plan how you will connect your CC2500 transceiver
  - Understand the schematics of the module
  - Have your module's CPU erased by the TA
    - To remove any potential for I/O conflicts
- Write SPI driver and test alone on McGumps
  - Test using a **loopback**
    - **More on loopback testing in a few minutes**
  - Validate all the timing with the logic analyzer
    - Before worrying about the CC2500
    - TAs will ask you to demonstrate that your timing is OK and that you understand what you are doing.

Loopback testing

- Idea : send something and have it returned
  - At the same level (physical, link, network, etc.)
- Many levels possible
  - Physical → SPI loopback, UART Loopback
    - Example: short pin 2 and 3 on the PC RS-232 together (pin 2 = TxD, pin 3=RxD)
  - Link Level → Wireless Loopback
    - Similar to a network PING
  - Network/Application Level
    - Not really applicable in our course
    - E.g. Echo back HTTP get requests

Loopback Testing – Single GPIO

- You have the ability to “see” what you drive on a GPIO pin – Example:
  - Set a pin as output (PxDIR.n = '1')
  - Drive a logical '1' to PxOUT.n
  - Read PxIN.n
- You expect to read what you wrote, right ?
  - What if you drive '1' and read '0' in PxIN.n ?
    - Or drove '0' and read '1' ?
  - You can detect that the pin is misbehaving
- Possible causes
  - Pin is shorted to GND/Vcc
  - Another chip is driving the same pin
  - You forgot to set the PxDIR...

Testing for shorts across pins

- Take a pair of pins
  - Alternatively drive '0' or '1' on the pin
  - Check neighboring pins (configured as inputs) to see if they follow
    - Might indicate a short between two pins
- You may think that you can “beep the lines” with a conductivity tester
  - True...
  - But try to perform this test reliably on a circuit with 0.5mm between pins and underneath the latest tiny connectors
  - Sometimes, the “electronic” approach is the only easy way to test.

Loopback Testing – SPI Level

- Take MOSI and connect it to MISO
  - If you are going through the CPLD, its even easier to do... Just program the loopback in the CPLD
- Write to SPI transmit buffer
  - Expect the data to come back
- Recall Clock phase diagram
  - Notice launch (TX shifted out) and sample points (Rx sample points)
    - Delta is the timing budget
    - Rather large delay for slow SPI

Loopback testing - Generally

- Very useful, easy to do. Almost universal
- Telecom example
  - Inside a chip
    - Module loopback, with or without bypass
  - On a circuit board - buffer loopback
  - At the laser level - Fiber loopback
  - At the switch level
    - Cross connect loopback
  - At the remote end
    - This gets tougher
    - Some commands can be sent to remotely initiate loopback – Or a phone call to the other end...

Your work this week

- Finish Experiment #2
  - Focus on core requirements
- Book your demonstration time slot with the Tas
- Review the slides for Quiz #2