M04: Serial Communication Part 2: USB and SSI

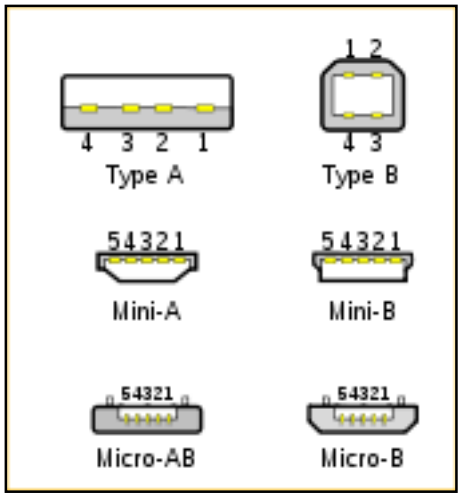
Prof. Rosa Zheng

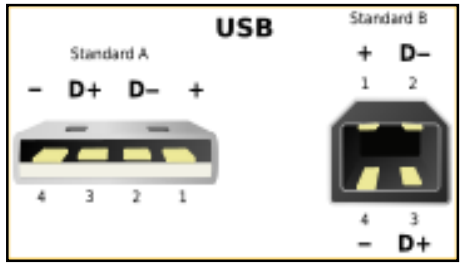
References:

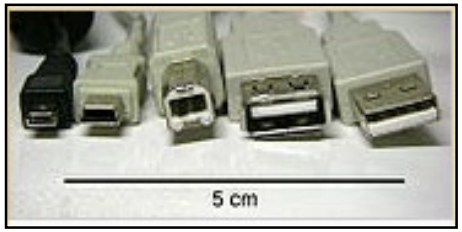
1. TIVA C Workshop student workbook.
2. TM4C123G datasheet: spms376e.pdf
3. Understanding the I2C bus: http://www.ti.com/lit/an/slva704/slva704.pdf
4. Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, Chapter 28. free online http://www.dspguide.com/pdfbook.htm

USB

- **4 pins – power, ground and 2 data lines**
  (5th pin ID for USB 2.0 connectors)
- **Standards:**
  - **USB 1.1**
    - Defines **Host** (master) and **Device** (slave)
    - Speeds to 12 Mbits/sec
    - Devices can consume 500 mA (100 mA for startup)
  - **USB 2.0**
    - Speeds to 480 Mbits/sec
    - OTG addendum
  - **USB 3.0**
    - Speeds to 4.8 Gbits/sec
    - New connector(s)
    - Separate transmit/receive data lines







USB Basics

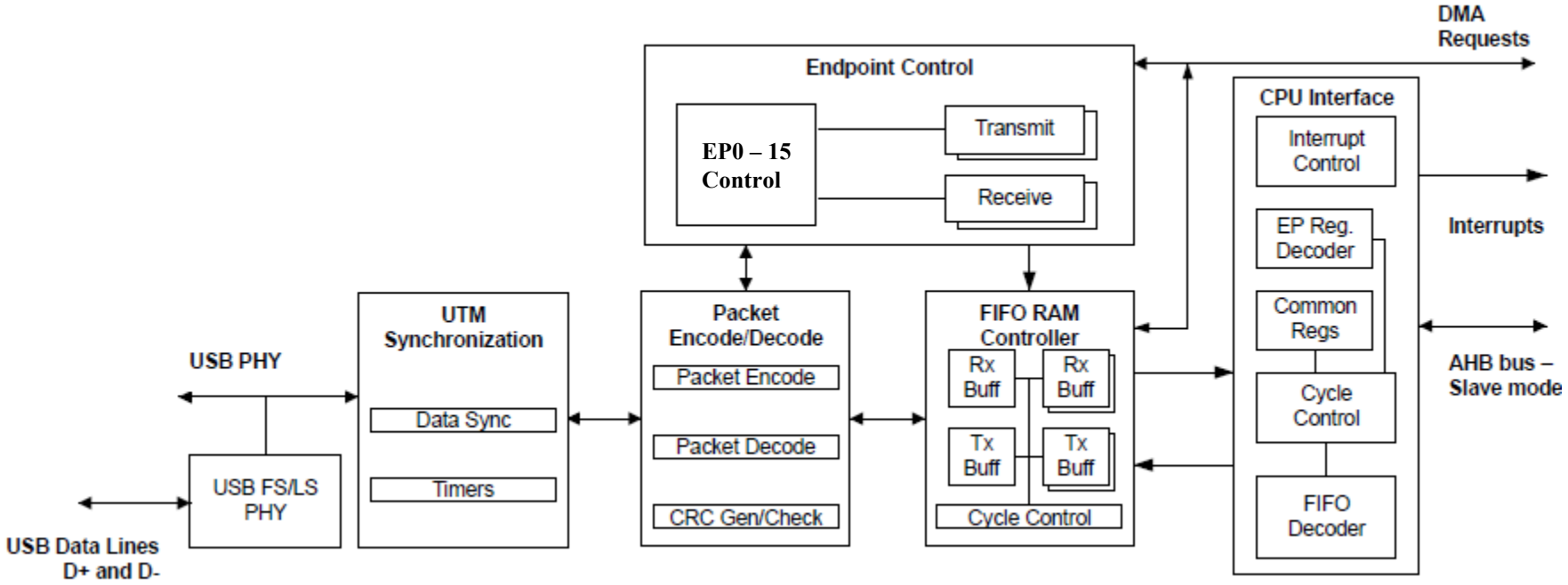
USB Device … most USB products are slaves
USB Host … usually a PC, but can be embedded
USB OTG … On-The-Go

- Dynamic switching between host and device roles
- Two connected OTG ports undergo host negotiation

Host polls each Device at power up. Information from Device includes:

- Device Descriptor (Manufacturer & Product ID so Host can find driver)
- Configuration Descriptor (Power consumption and Interface descriptors)
- Endpoint Descriptors (Transfer type, speed, etc)
- Process is called *Enumeration* … allows Plug-and-Play

USB Peripheral Block Diagram



USB 2.0 full speed (12 Mbps) and low speed (1.5 Mbps) operation

On-the-go (OTG), Host and Device functions

Integrated USB Controller and PHY with up to 16 Endpoints

- 1 dedicated control IN endpoint and 1 dedicated control OUT endpoint
- Up to 7 configurable IN endpoints and 7 configurable OUT endpoints
- 4 KB dedicated endpoint memory (not part of device SRAM)
- Separate DMA channels (up to three IN Endpoints and three OUT Endpoints)
- 1 endpoint may be defined for double-buffered 1023-bytes isochronous packet size

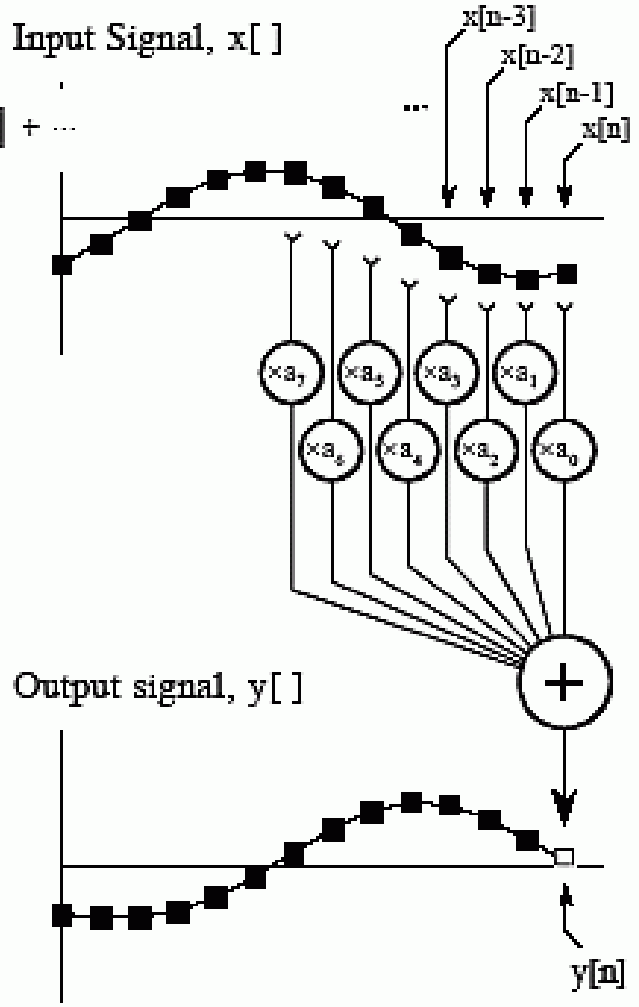
DSP and Filtering

- **An FIR filter:**

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + a_3x[n-3] + a_4x[n-4] + \cdots$$

- **Linear Buffer: addr is fixed**

Memory Address	Stored Value	Input signal
0x200040	-0.113940	X[n]
0x200041	-0.228918	X[n-1]
0x200042	-0.269847	X[n-2]
0x200043	-0.225767	X[n-3]
0x200044	0.462791	X[n-4]
0x200045	0.371370	X[n-5]
0x200046	0.222977	X[n-6]
0x200047	0.048679	X[n-7]



Circular Buffer

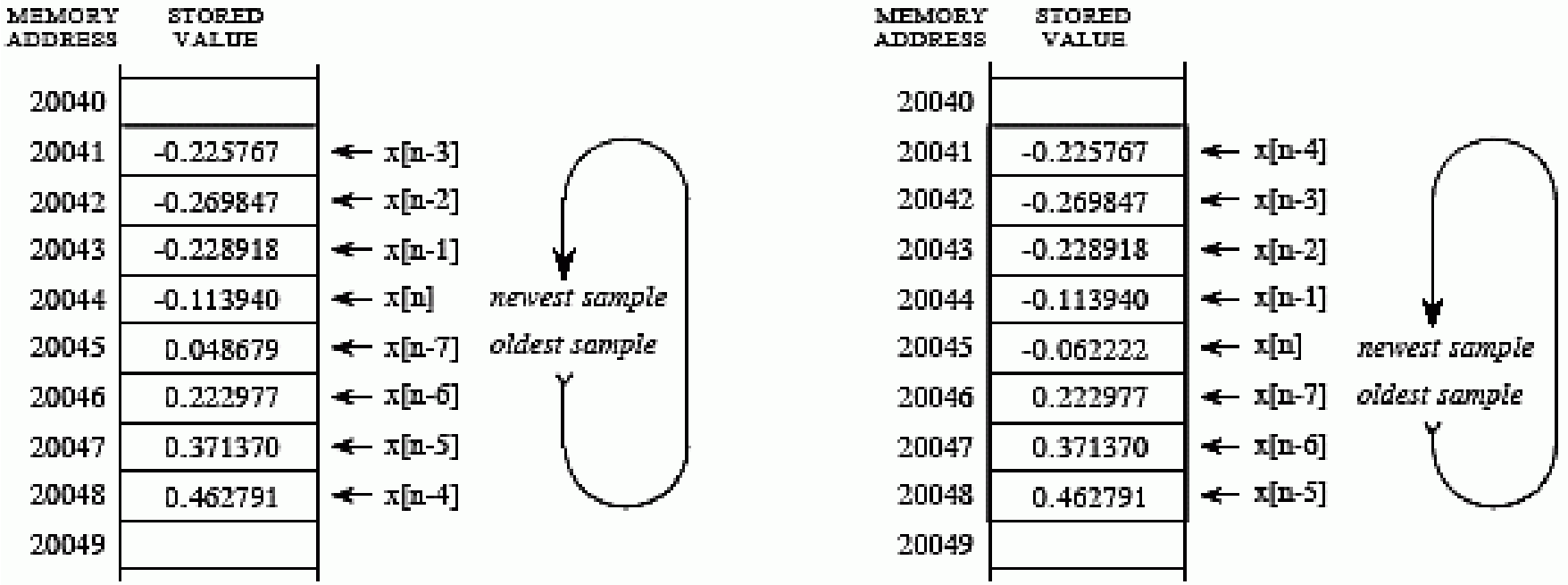


FIGURE 28-3
Circular buffer operation. Circular buffers are used to store the most recent values of a continually updated signal. This illustration shows how an eight sample circular buffer might appear at some instant in time (a), and how it would appear one sample later (b).

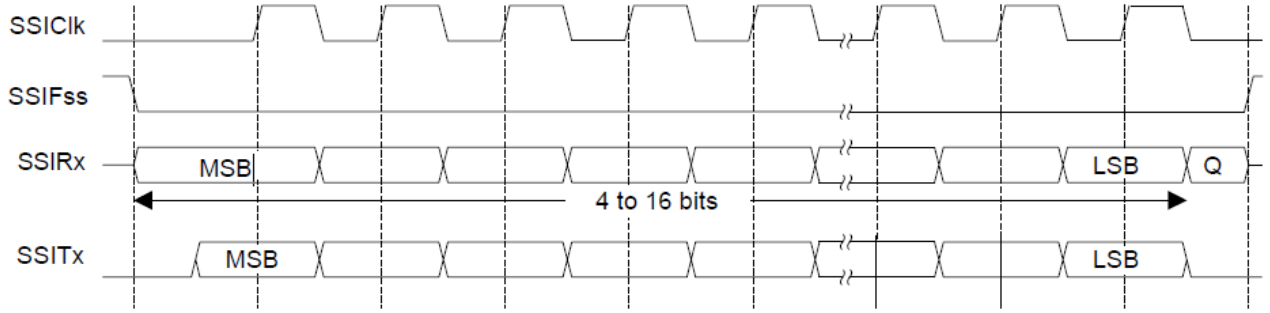
TM4C123GH6PM SSI Bus

- **Four SSI modules. Each with:**
  - Freescale SPI, MICROWIRE or TI Synchronous Serial interfaces
  - Master or Slave operation, Interrupts and µDMA support
  - Programmable bit clock rate and clock pre-scaler
  - Programmable data frame size from 4 to 16-bits
  - Separate Tx and Rx FIFOs (8 x16-bits)
- **SSI Interrupts**
  - Single interrupt per module, cleared automatically
  - Interrupt conditions:
    - Transmit/ Receive FIFO service (when transmit/receive FIFO is half full or less/more)
    - Receive FIFO time-out , overrun, End of transmission, or
    - Transmit /Receive DMA transfer complete

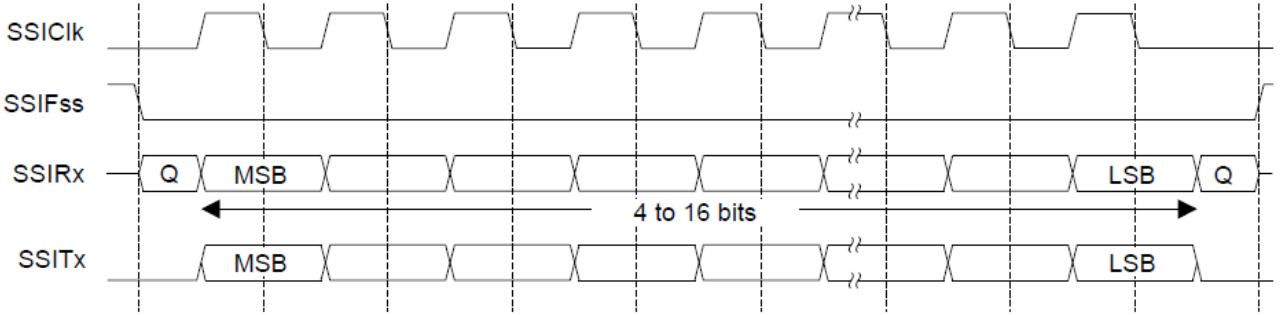
Freescale SPI Signal Formats

- Four wire interface. Full duplex. SSIFss acts as chip select
- Inactive state and clock phasing are programmable via the SPO and SPH bits (SSI_FRF_MOTO_MODE_0-3 parameter)
  - SPO = 0: SSIClk low when inactive. SPO = 1: high
  - SPH = 0: Data is captured on 1st SSIClk transition. SPH = 1: 2nd

SPO = 0
SPH = 0
Single
Transfer

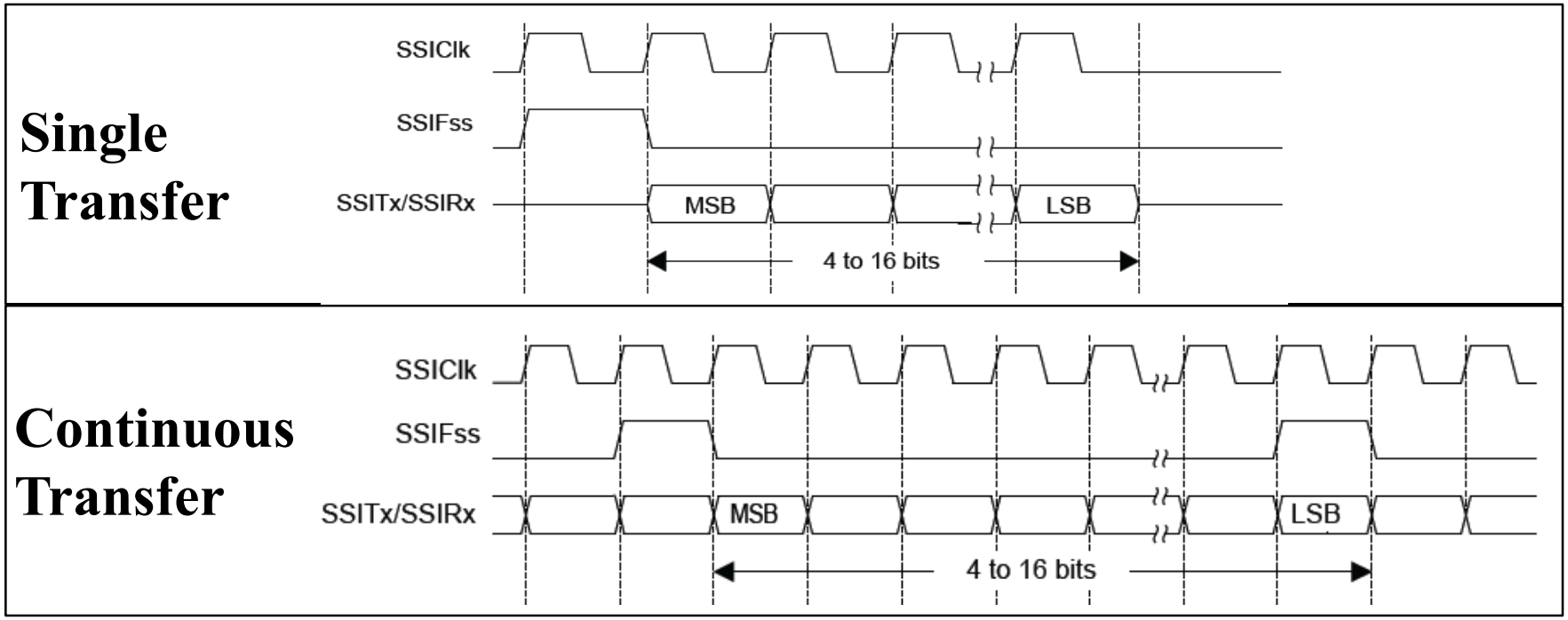


SPO = 0
SPH = 1
Single
Transfer



TI Synchronous Serial Signal Formats

- Three wire interface, instead of four wires of SPI
- Devices are always slaves
- **SSIClk** and **SSIFss** are forced low and **SSITx** is tri-stated when the SSI is idle



I2C (Inter-Integrated Circuit) bus

- A synchronous, multi-master, multi-slave, packet switched, single-ended, serial computer bus
- Physical I2C interface consists of the serial clock (SCL) and serial data (SDA) lines. Both SDA and SCL lines must be connected to VCC through a pull-up resistor.
- Data must be sent and received to or from the slave devices via registers