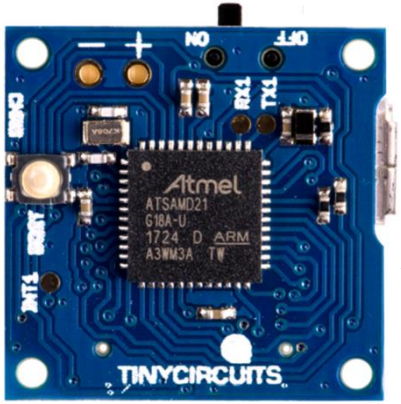
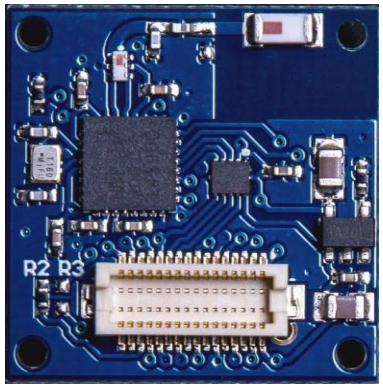
CSE190 Winter 2025



Lecture 13

Serial Busses (cont)



Wireless Embedded Systems

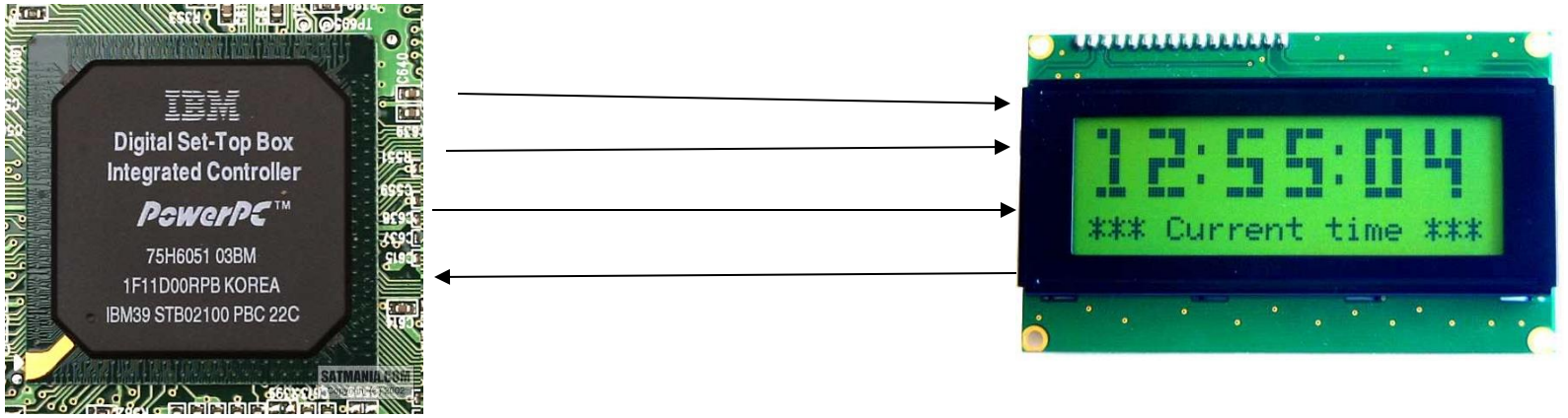
Aaron Schulman

Serial Peripheral Interconnect (SPI)

- Another kind of serial protocol in embedded systems (proposed by Motorola)
- Four-wire protocol
  - SCLK — Serial Clock
  - MOSI/SIMO — Master Output, Slave Input
  - MISO/SOMI — Master Input, Slave Output
  - SS — Slave Select
- Single primary device and with one or more secondary devices
- Higher throughput than I2C and can do “stream transfers”
- No arbitration required
- But
  - Requires more pins
  - Has no hardware flow control
  - No secondary acknowledgment (primary could be talking to thin air and not even know it)

What is SPI?

- Serial Peripheral Interface (SPI) protocol [1979]
- Fast (Mbps), easy to use (few wires), simple
- Nearly all microcontrollers support it



SPI Basics

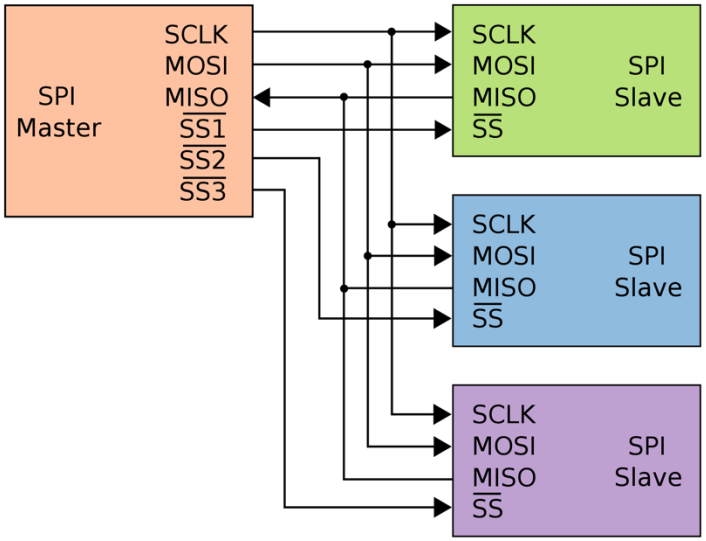
- Uses 4 wires (compared to UART's 2-wires)
  - Also known as a "4 wire" bus
- Used to communicate across short distances
- Multiple Secondaries, Single Primary
- Synchronized

SPI Capabilities

- Always Full Duplex
  - Communicating in two directions at the same time
  - Transmission need not be meaningful
- Multiple Mbps transmission speed
- Transfers data in 4 to 32 bit characters
- Multiple slaves
  - Daisy-chaining possible

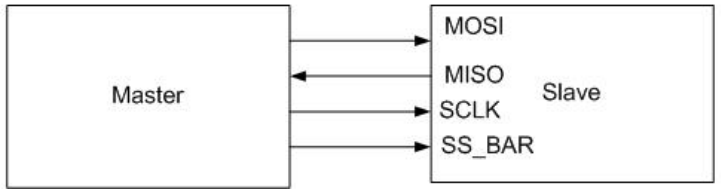
SPI Protocol

- Wires:
  - Master Out Slave In (MOSI)
  - Master In Slave Out (MISO)
  - System Clock (SCLK)
  - Slave Select 1…N
- Master Set Slave Select low
- Master Generates Clock
- Shift registers shift in and out data

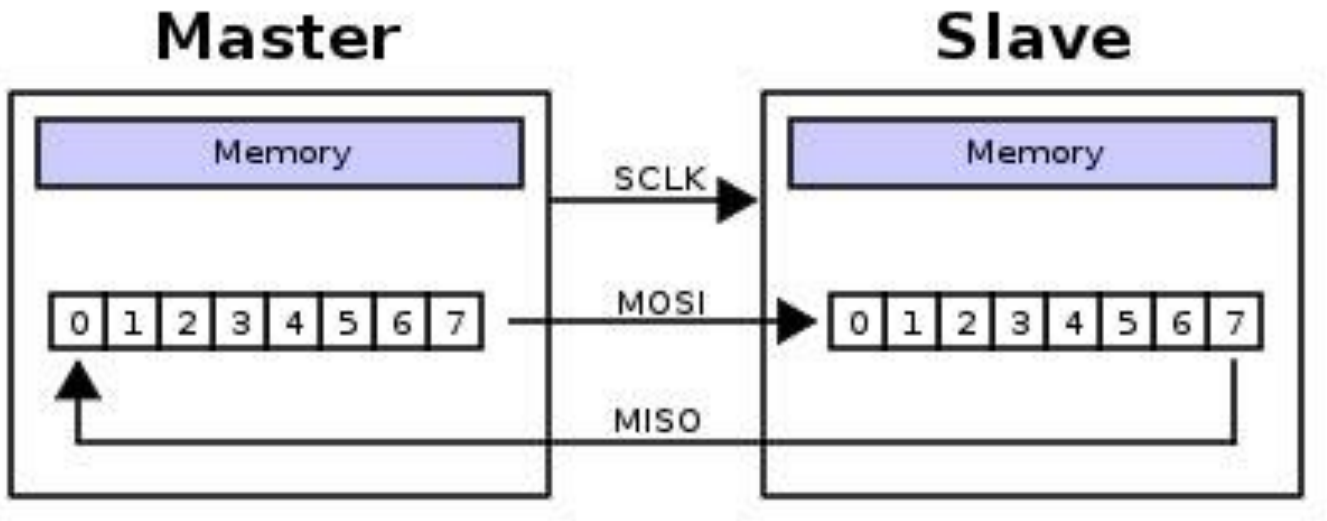


SPI Wires in Detail

- MOSI – Carries data out of Primary to Secondary
- MISO – Carries data from Secondary to Primary
  - Both signals happen for every transmission
- SS_BAR – Unique line to select a secondary
- SCLK – Primary-produced clock to synchronize data transfer



SPI is the quintessential “shift register” communication bus

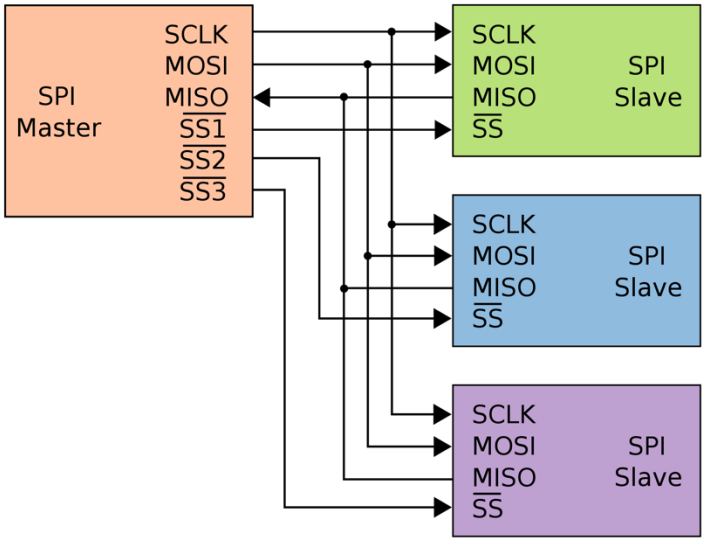


Primary shifts out data to Secondary, and shifts in data from Secondary

http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/SPI_8-bit_circular_transfer.svg/400px-SPI_8-bit_circular_transfer.svg.png

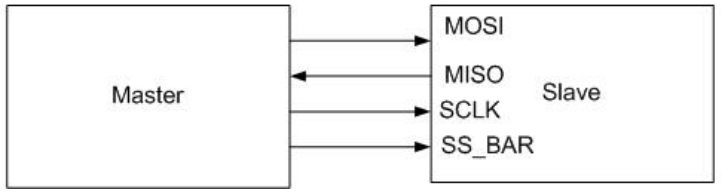
SPI Protocol

- Wires:
  - Master Out Slave In (MOSI)
  - Master In Slave Out (MISO)
  - System Clock (SCLK)
  - Slave Select 1…N
- Master Set Slave Select low
- Master Generates Clock
- Shift registers shift in and out data

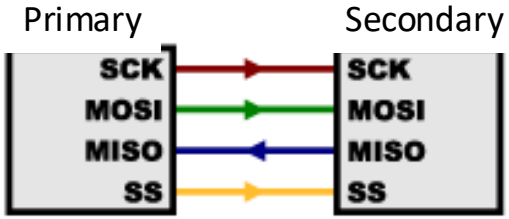


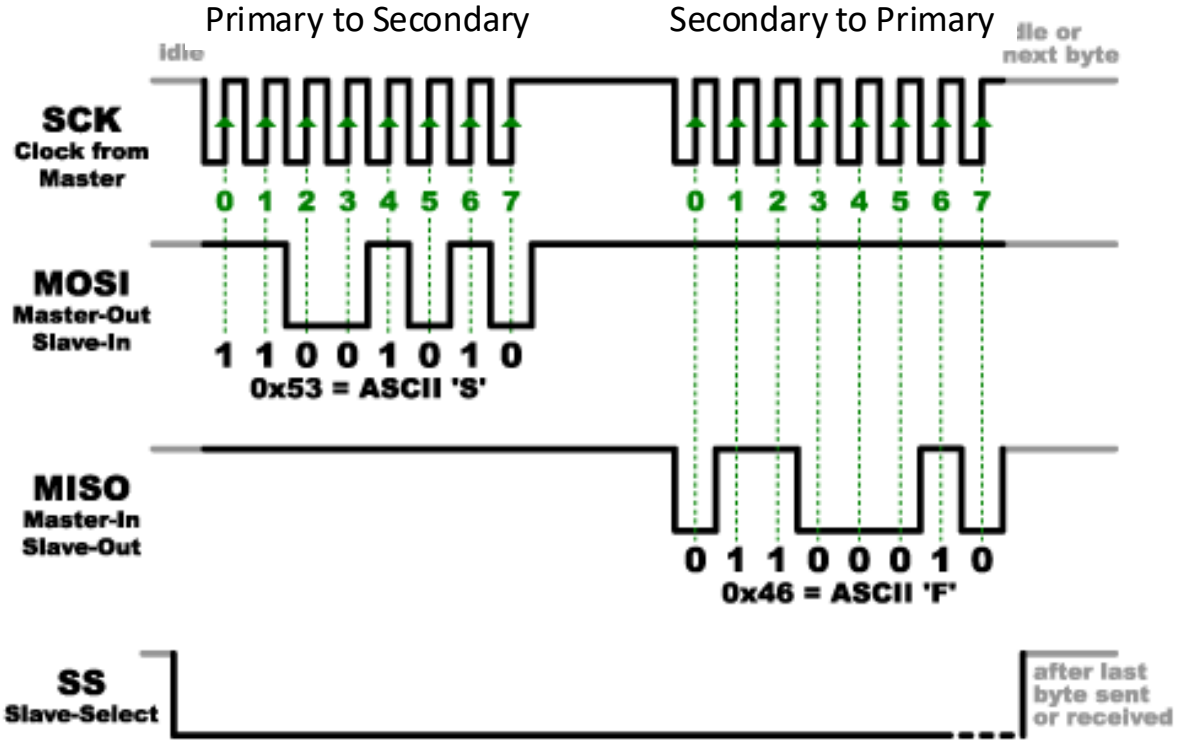
SPI Wires in Detail

- MOSI – Carries data out of Primary to Secondary
- MISO – Carries data from Secondary to Primary
  - Both signals happen for every transmission
- SS_BAR – Unique line to select a secondary
- SCLK – Primary-produced clock to synchronize data transfer

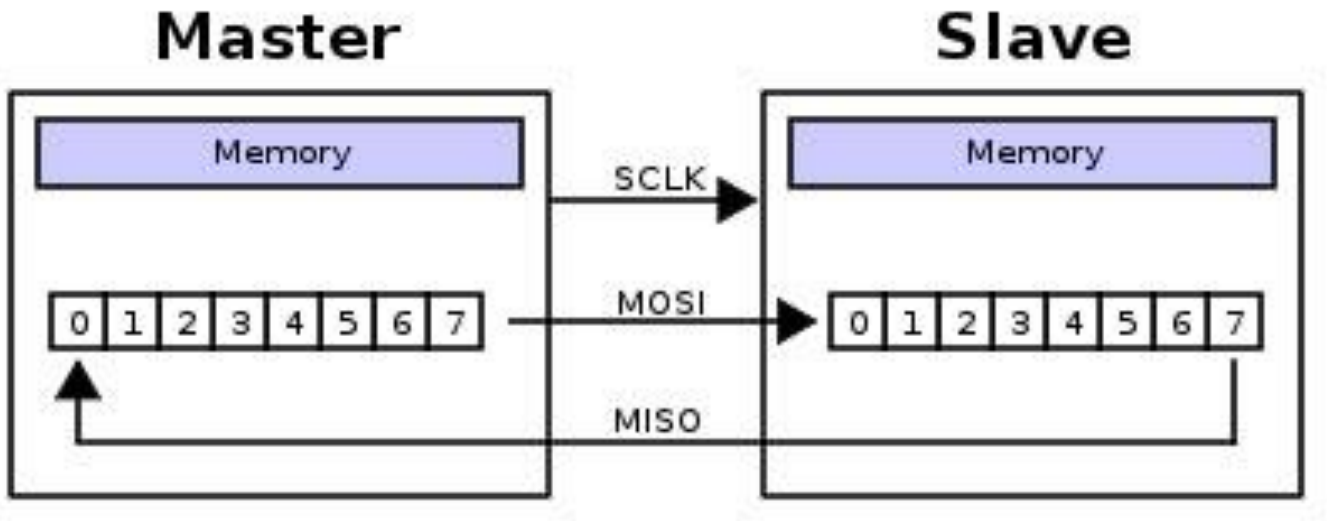


SPI Communication





SPI is the quintessential “shift register” communication bus



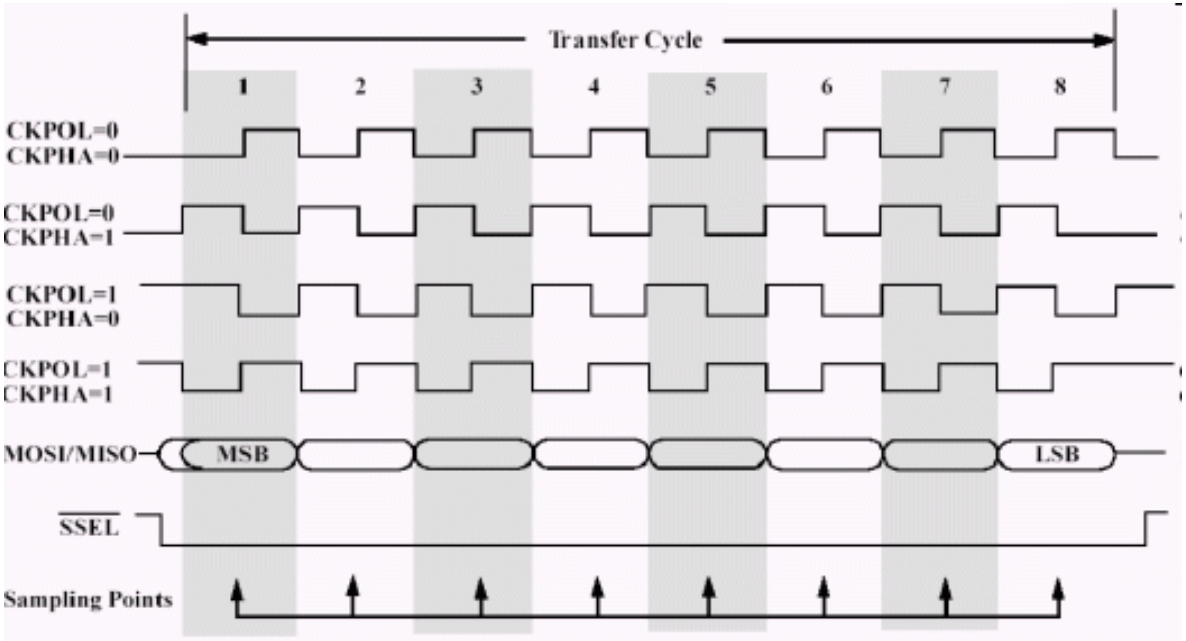
Primary shifts out data to Secondary, and shifts in data from Secondary

http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/SPI_8-bit_circular_transfer.svg/400px-SPI_8-bit_circular_transfer.svg.png

SPI clocking: **there is no "standard way"**

- Four clocking "modes"
  - Two phases
  - Two polarities
- Primary and *selected* secondary must be in the same mode
- During transfers with secondary's A and B, primary must
  - Configure clock to secondary A's clock mode
  - Select secondary A
  - Do transfer
  - Deselect secondary A
  - Configure clock to secondary B's clock mode
  - Select secondary B
  - Do transfer
  - Deselect secondary B
- primary reconfigures clock mode on-the-fly!

SPI timing diagram showing different clock modes



Timing Diagram – Showing Clock polarities and phases

http://www.maxim-ic.com.cn/images/appnotes/3078/3078Fig02.gif

SPI Pros and Cons

- Pros:
  - Fast and easy
    - Fast for point-to-point connections
    - Easily allows streaming/Constant data inflow
    - No addressing/Simple to implement
  - Everyone supports it

- Cons:
  - SS makes multiple secondaries very complicated
  - No acknowledgement ability
  - No inherent arbitration
  - No flow control