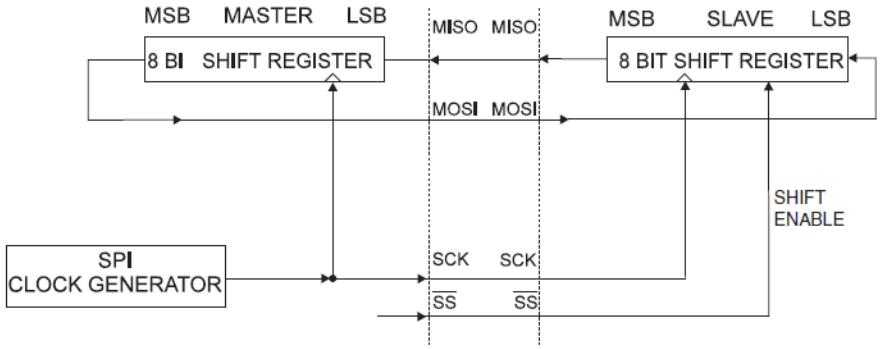
Lab Hand In and Grades

- Navigate to your lab's Teams group
- Select "Assignments" to view the lab problems and submit your lab report
- Select "Grades" to see all your work marked from Lab 5 onwards

Serial Peripheral Interface

Figure 18-2. SPI Master-slave interconnection.

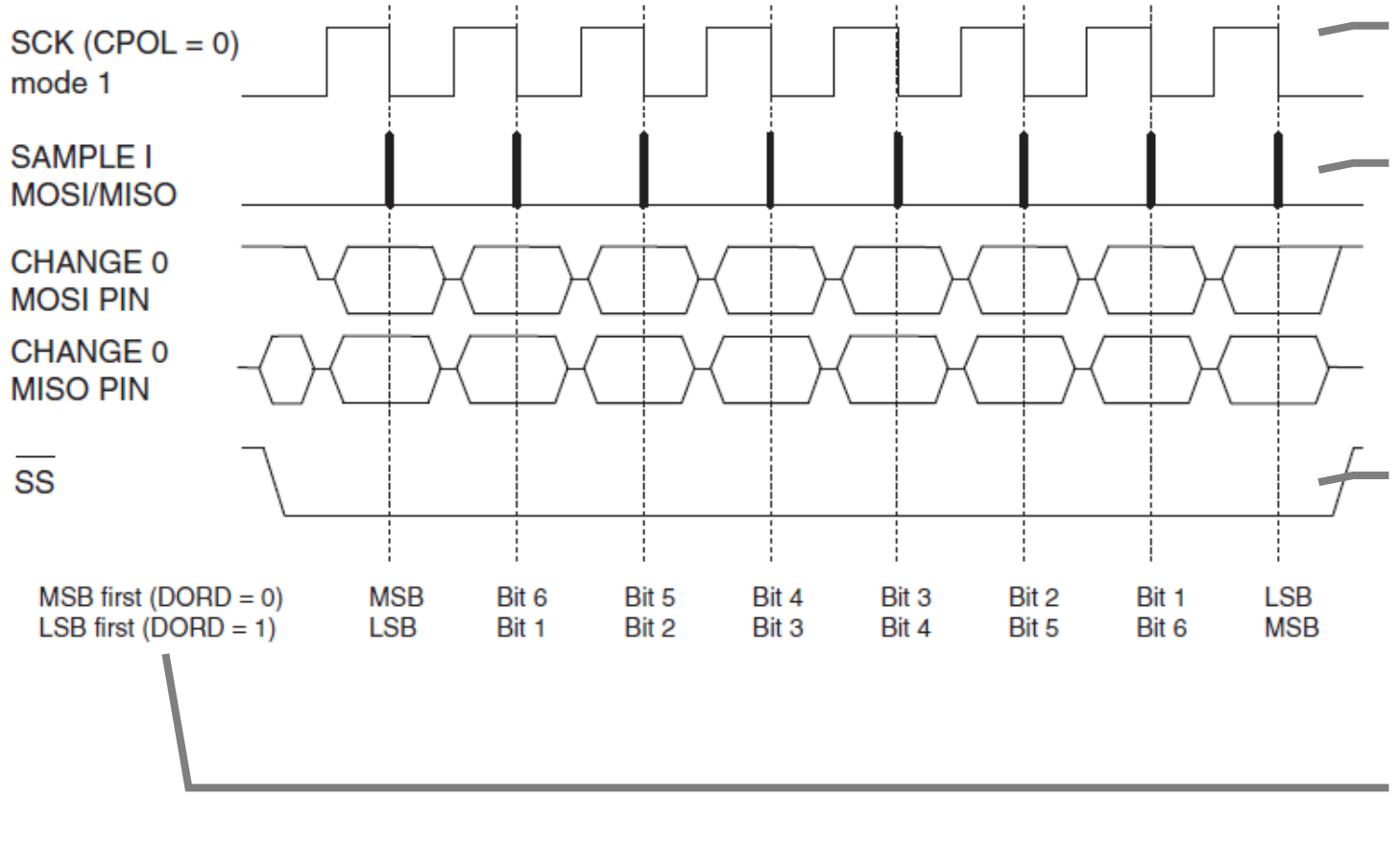


- Full Duplex – Transmit and receive data at the same time
- Synchronous – Shared clock between the Master and Slave
- The Clock signal needed to send or receive data is generated by the Microcontroller
  - It only generates this clock signal when the Microcontroller sends data
  - This is where you might see “garbage” being sent when we read data from an SPI device
- There are two registers that need to be set in the `setup()` to use the SPI
  - SPCR
  - SPSR

Microchip AT90USB1287 Datasheet

SPI – Data Packet

Figure 18-4. SPI transfer format with CPHA = 1.



Clock Signal generated by µC, Tx over *SCK* line

Sample Signal we want to send (0xFF)

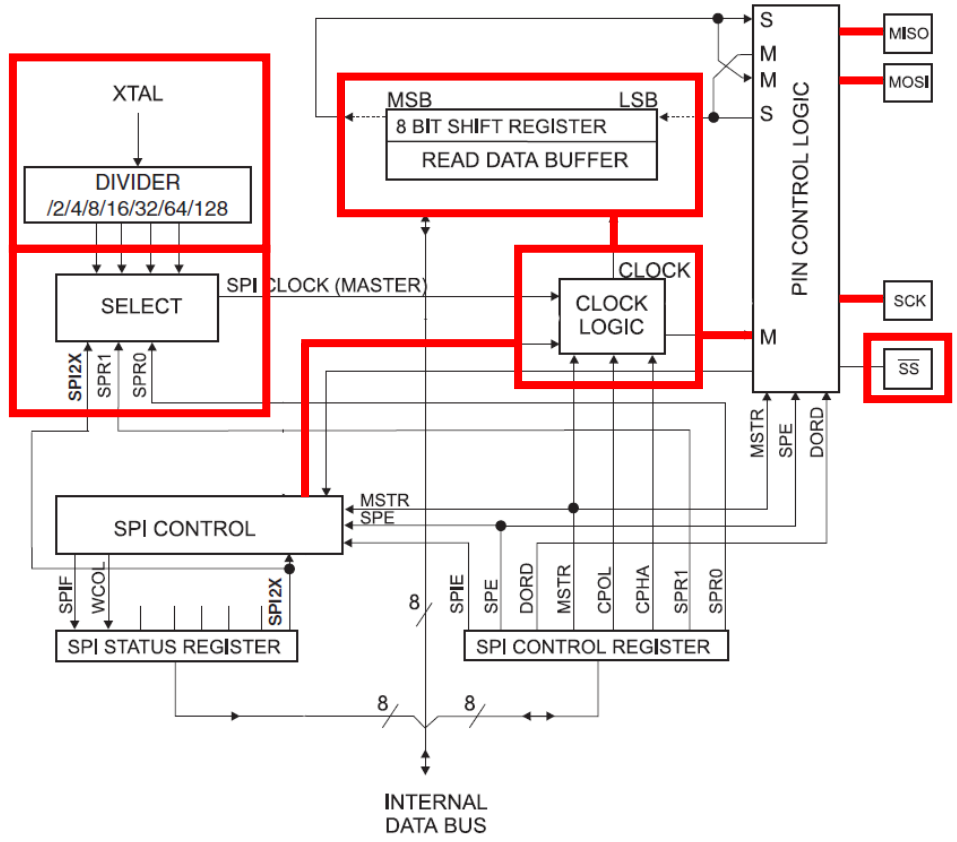
Slave Select (SS) line is pulled down

Notice: DORD changes the order of Bits sent

SPI – Block Diagram

- The Clock
  - Generated by the f_{osc}, and send through the divider
  - Selection of Clock Rate using SPI2X & SPR1-0 registers
  - Resulting Clock is sent to the Clock Logic
  - When the SPI Control activate the signal (i.e. Send Data), clock is sent to the Data Registers and SCK pin
- Slave Select Line
  - Selects the Slave to be used
  - Must be turned on before sending/receiving data
- The Data Registers
  - When data is written to SPDR, transmission starts
  - The signal from the Clock Logic is used to time both the Master and Slave
  - Data from the Read Data Buffer is placed into the Shift Register
  - Data is sent from the Shift Register to the Slave and received from the Slave into the Shift Register (1-in-1-out)

Figure 18-1. SPI block diagram [(1)].

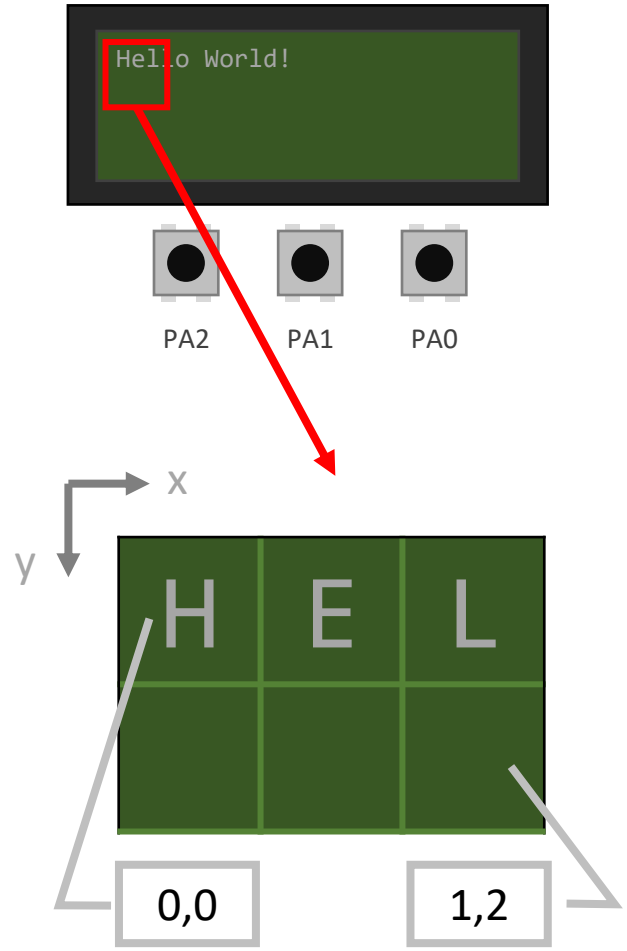


SPI – Transmission

- Enable the Slave Select Line
- Write data to the SPDR register
  - Note – Even if you are wanting to receive data, as the clock is controlled by the Data Register, you will need to send data to receive data!
- Wait for the SPI to complete (check the SPIF flag)
- Read data from the SPDR register
- Disable the Slave Select Line

LCD Display

- This week will require using the 20x4 Alphanumeric display
  - Review the contents of your lab manual to see how this is setup
  - The example on Page 24 is a good example to refer to
- Call these functions in your `setup()`
  - `SLCDInit()` – This will initiate the TWI serial protocol to allow your µC to talk to the LCD
  - `SLCDDisplayOn()` – This will turn on the display to show characters
  - `SLCDClearScreen()` – This will clear the screen of any existing content (such as the last students code)
- The LCD is split into a 20 x 4 grid
  - The top-left cell is position 0,0
  - Use the `SLCDSetCursorPosition(y,x)` function to set the cursor position
  - Use the `SLCDWriteString(charArrayToWrite)` function to place text on the screen
  - The text needs to be in the form of a `char` array (use the `sprintf()` function)



sprintf Function

```
// Example of using the sprintf function

char line0[20];
sprintf(line0, "temperature - %3d",tempSensorValue);
```

- ‘line0’ is the char array in which the string will be stored in
  - This should be defined prior (as per the example above), and we set this to a blank array of 20 char values (since each line of the display can only show 20 characters)
- ‘temperature - %3d’ is the string we want to convert to show on the display
  - %3d indicates that we intend to reserve three (‘3’) spaces for a decimal number (‘d’) to be displayed
- ‘tempSensorValue’ is the number we plan to substitute in this location
  - The space we reserved earlier will be replaced by a 3-digit decimal number
- Multiple variables can be used for the sprintf function
  - See your lab manual for more information