

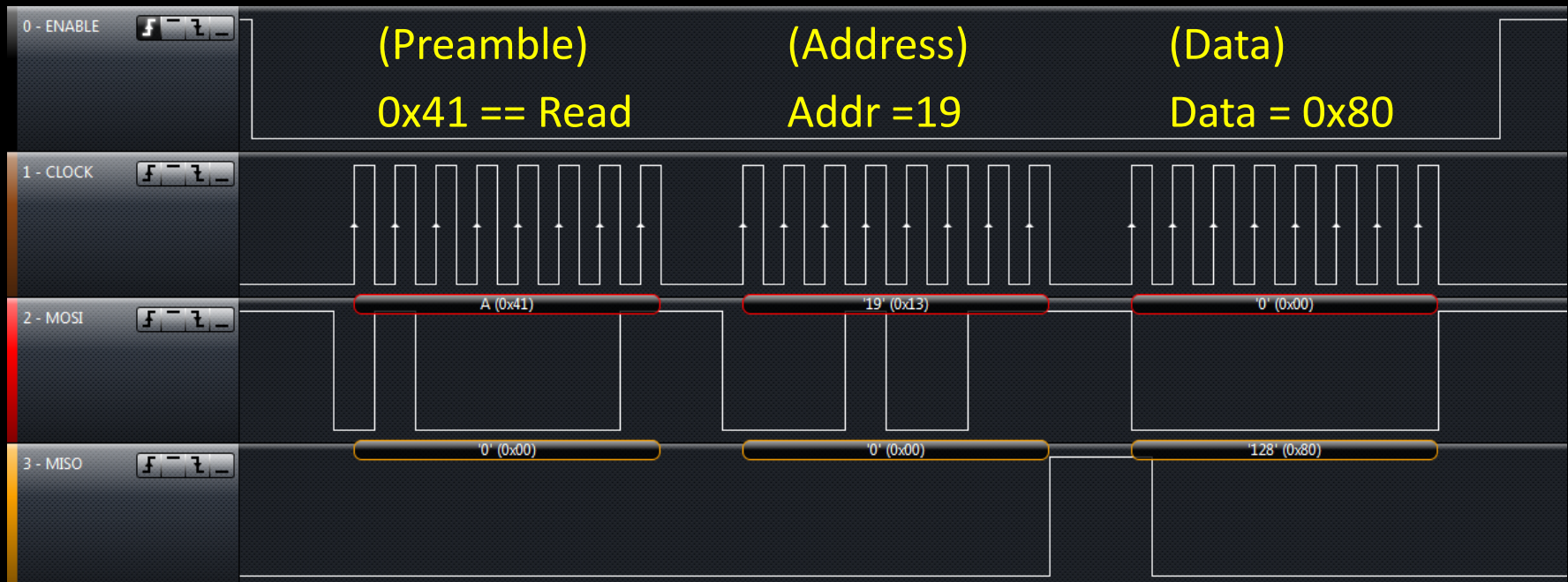
# SPI Interfacing

Lab4-aid

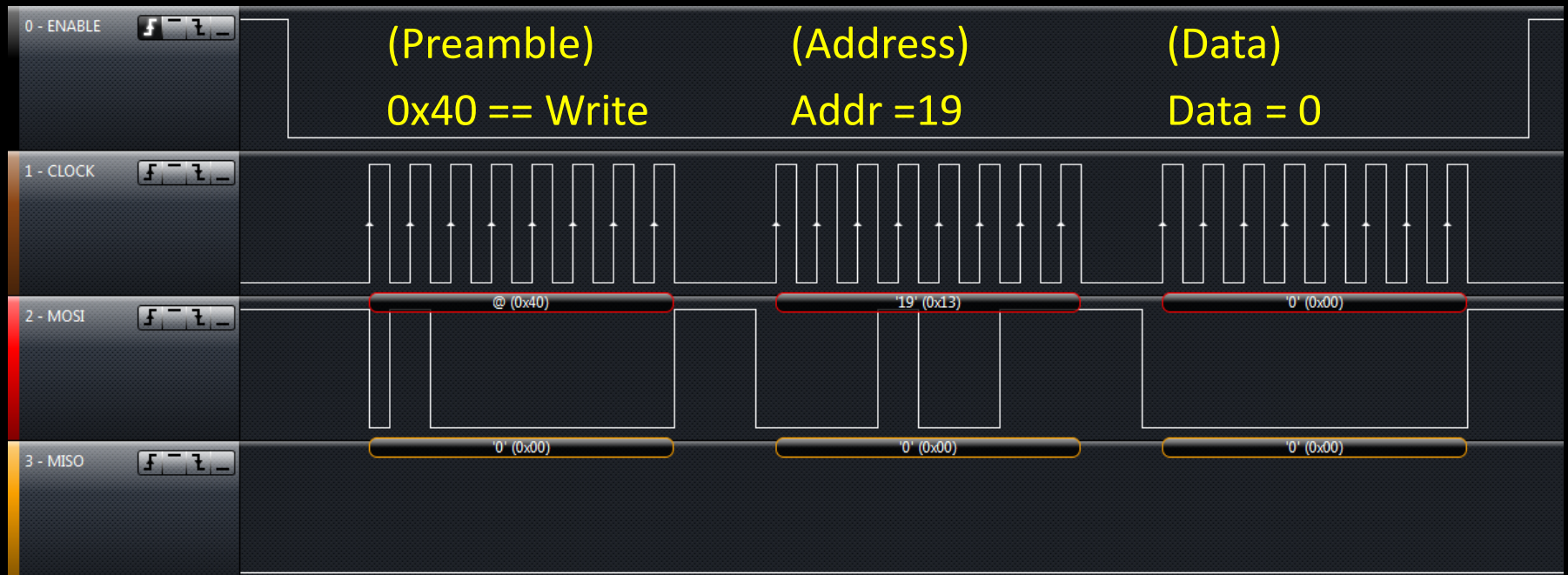
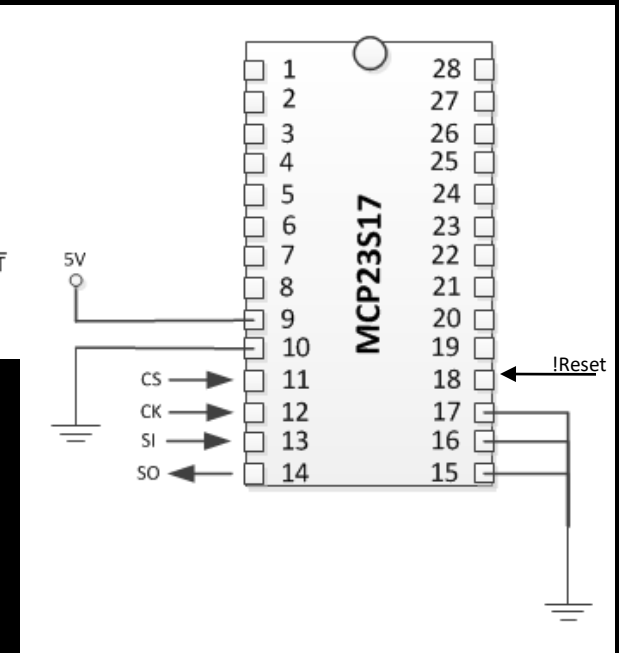
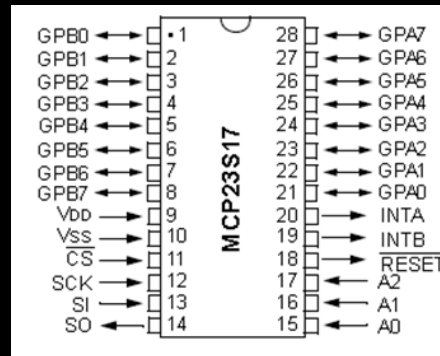
Connecting to the MCP23S17 SPI  
GPIO Expander

## SPI Basics

- 3 Output Pins
- 1 Input Pin
- 8-bit reads and writes
- Operates in simple mode
- Lots of High Speed Modes
- Enable (active low !CS)
- Clock
- MOSI (Master Out Slave In)
- MISO (Master In Slave Out)
- Input and output data is sampled at CLK rising edge.
- We will use Mode-0 where we start with CLK high and CS high
- Once CS is lowered data should be presented on the serial line then CLK raised from low to high



- The data output presents data when clock goes high
- The data inputs read the data soon after the clock goes high.
- A clever routine can be written to both send and receive
- Slave device generally sets data on falling edge of clock
- Use the Tiva Reset signal to reset the expander



- Wikipedia has a good SPI reference search for “SPI”
- Bit-Banging refers to using GPIO pins to emulate a protocol.
- Most complex ASIC systems have SPI blocks built in (Our Tiva has one and we may use it later)
- Look at the definitions of the setXX and getXX macros to map back to which pins

```
uint8_t transfer(uint8_t out)
{
    uint8_t count, in = 0;

    setSCK(LOW);
    for (count = 0; count < 8; count++)
    {
        in <=< 1;
        setMOSI(out & 0x80);    // set Output bit
        setSCK(HIGH);           // Clock Rising Edge
        in += getMISO();         // Read the data bit
        setSCK(LOW);            // Clock Rising Edge
        out <=< 1;               // shift read bit
    }
    setMOSI(0);

    return (in);
}
```

- 
- The Transfer() function manages one byte of data to and from the device at the same time.
  - The expanderReadByte() function manipulates CS and multi-byte sequences.

```
static uint8_t expanderReadByte(uint8_t address)
{
    uint8_t value, preRead = 0x41;

    setCS(LOW);
    transfer(preRead);
    transfer(address);
    value = transfer(0);
    setCS(HIGH);

    return value;
}
```

- The traces below show the 4, 3-byte transactions for `expanderInit()`

