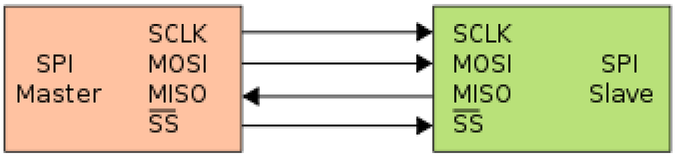
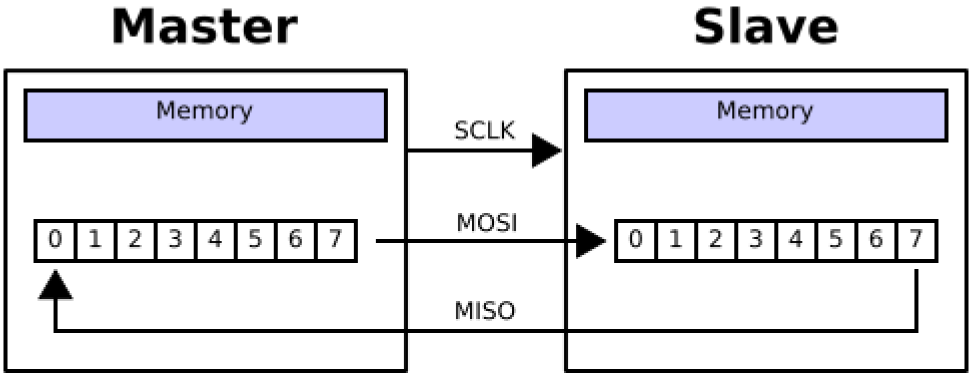
# 538 Lecture Notes Week 10

In addition to Chapter 10 in the text book, the wikipedia articlecontains an excellent overview of this protocol. (The notes below are very incomplete; I realized that the wikipedia article was sufficient...)

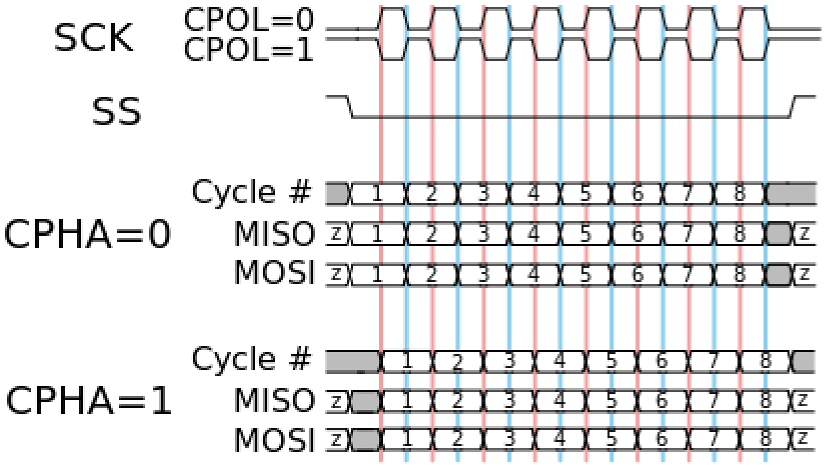
## *Serial Peripheral Interface (SPI)*

- *Synchronous* serial communication.
- Signals:
  - Clock (SCK)
  - Master Out/Slave In (MOSI) (serial data)
  - Master In/Slave Out (MISO) (serial data)
  - Slave Select (SS*)





The diagram below illustrates the timing options:



## Inter-Integrated Circuit (I²C) Interface

- Another *Synchronous* serial communication.
- Signals:
  - Clock (SCL)
  - SDA (serial data)
  - (Also Power and Ground)

See also Wikipedia article

## Hardware organization

In addition to Chapter 14 of the text, my notes on hardware organization as discussed in class are available here.

Last revised: November 7, 2017

The diagram below illustrates the timing options:

## *Review Questions*

**1.** Complete the empty slots in the following table.

```
Abs.    Loc    Obj. code   Source line
----    ------ ---------   -----------


                         org $3000
0400             product rmb 2

                         org   $4000

4000 CF40 00                 lds   #$4000
4003 8603                    ldaa #3
4005 36                      psha
4006 8604                    ldaa #4
4008 36                      psha
4009 1640 16                 jsr foo
400C C603                    ldab #3
400E 12                      mul
400F 1B82                    leas 2,sp
4011 7C04 00                 std product
4014 20FE                    bra *

                     foo:
4016 6383                    dec 3,sp
4018 2704                    beq done
401A 6284                    inc 4,sp
401C 20F8                    bra foo
401E A684        done        ldaa 4,sp
4020 3D                      rts
```

**2.** Translate the following C code into assembler. (Assume the return value is passed in Accumulator A.)

```
  char foo(char w, char z) {
    char x = 0;
    while(w > 0) {
       x += z;
       w--;
    }
 return x;
}
```

**3.** Athe Timer module is to generate an interrupt every millisecond.