



# Electrical and Computer Engineering EE3622 Embedded Systems Design



## SPI Data Transfer Laboratory Solution Dennis Silage, PhD

```
// SPI Lab 2008 Dennis Silage spilab08.v

// top module
module spilab08 (input CCLK, BTN0, BTN1, BTN2, BTN3, SW0, SW1, SW2, SW3,
                 input SW4, SW5, SW6, SW7, output LD7, LD6, LD5, LD4,
                 output LD3, LD2, LD1, LD0);

wire clk1, clk100, pbbtn0, pbbtn1, pbbtn2, pbbtn3;
wire swmread, swsread, sclk, mosi, ss, mrdslv;
wire [3:0] swmdata;
wire [3:0] mled;
wire [3:0] swsdata;
wire [3:0] sled;

assign swmdata[0]=SW4;
assign swmdata[1]=SW5;
assign swmdata[2]=SW6;
assign swmdata[3]=SW7;

assign swsdata[0]=SW0;
assign swsdata[1]=SW1;
assign swsdata[2]=SW2;
assign swsdata[3]=SW3;

assign LD4=mled[0];
assign LD5=mled[1];
assign LD6=mled[2];
assign LD7=mled[3];

assign LD0=sled[0];
assign LD1=sled[1];
assign LD2=sled[2];
assign LD3=sled[3];

clock M0 (CCLK, 2500000, clk1);           // 1 Hz
clock M1 (CCLK, 250000, clk100);         // 100 Hz
pbdebounce M2 (clk100, BTN0, pbbtn0);    // master reads data
```

```

pbdebounce M3 (clk100, BTN1, pbbtn1);      // slave reads data
pbdebounce M4 (clk100, BTN2, pbbtn2);      // clear master and slave data
pbdebounce M5 (clk100, BTN3, pbbtn3);      // master reads slave
spimcntrl M6 (clk1, pbbtn0, pbbtn2, pbbtn3, swmread, mrdslv);
spimdtpth M7 (clk1, swmread, pbbtn2, mrdslv, swmdata, mled, sclk, mosi, ss, miso);
spiscntrl M8 (clk1, pbbtn1, pbbtn2, swsread);
spisdpth M9 (clk1, swsread, pbbtn2, swsdata, sled, sclk, mosi, ss, miso);

```

```
endmodule
```

```

// SPI Lab 2008 Dennis Sllage spimaster.v
// SPI Master

```

```
// master controller
```

```
module spimcntrl (input spimcclk, mcrd, mcrst, mcrdslv, output reg rdmdat,
                 output reg rdslave);
```

```

always@(posedge spimcclk)
    begin
        if (mcrst==1) // external reset
            begin
                rdmdat=0;
                rdslave=0;
            end
        else if (mcrd==1) // read master data
            rdmdat=1;
        else if (mcrdslv==1) // transfer slave data
            rdslave=1;
    end
end

```

```
endmodule
```

```
// master datapath
```

```
module spimdtpth (input spimdclk, mrds, mdrst, rdslv, input [3:0] switch,
                 output reg [3:0] swled, output reg sclk,
                 output reg mosi, output reg ss, input miso);
```

```
reg [4:0] spimdstate=0;
```

```

always@(posedge spimdclk)
    begin
        if (mdrst==1) // external reset

```

```

        swled=0;
        mosi=0;
        spimdstate=0;
    end

case (spimdstate)
0:    begin                                // reset state
        sclk=0;
        ss=1;
        if (mrdsw==1)                      // master read switches
            spimdstate=1;
        if (rdslv==1)
            spimdstate=10;                 // master read slave
    end

1:    begin                                // master -> slave transfer
        swled=switch;
        ss=0;
        mosi=switch[3];                    // MSB
        spimdstate=2;
    end

2:    begin
        sclk=1;
        spimdstate=3;
    end

3:    begin
        sclk=0;
        mosi=switch[2];
        spimdstate=4;
    end

4:    begin
        sclk=1;
        spimdstate=5;
    end

5:    begin
        sclk=0;
        mosi=switch[1];
        spimdstate=6;
    end

6:    begin
        sclk=1;
        spimdstate=7;
    end

7:    begin
        sclk=0;
        mosi=switch[0];                    // LSB

```

```

        spimdstate=8;
    end
8:   begin
        sclk=1;
        spimdstate=9;
        ss=1;
    end
9:   begin
        sclk=0;
        spimdstate=0;
    end

10:  begin                                // slave -> master transfer
        sclk=0;
        ss=1;
        spimdstate=11;
    end
11:  begin
        sclk=1;
        spimdstate=12;
    end
12:  begin                                // MSB
        swled[3]=miso;
        spimdstate=13;
    end
13:  begin
        sclk=0;
        spimdstate=14;
    end
14:  begin
        sclk=1;
        spimdstate=15;
    end
15:  begin
        swled[2]=miso;
        spimdstate=16;
    end
16:  begin
        sclk=0;
        spimdstate=17;
    end
17:  begin
        sclk=1;
        spimdstate=18;
    end
18:  begin

```

```

        swled[1]=miso;
        spimdstate=19;
    end
    19: begin
        sclk=0;
        spimdstate=20;
    end
    20: begin
        sclk=1;
        spimdstate=21;
    end
    21: begin
        swled[0]=miso;    // LSB
        spimdstate=22;
    end
    22: begin
        sclk=0;
        spimdstate=0;
    end
endcase
end
endmodule

```

```

// SPI Lab 2008 Dennis Sllage spislave.v
// SPI Slave

// slave controller
module spiscntrl (input spiscclk, pbtn, scrst, output reg rdsdat);

always@(posedge spiscclk)
    begin
        if (scrst==1)    // external reset
            rdsdat=0;
        else if (pbtn==1) // read slave data
            rdsdat=1;
    end
endmodule

```

```

// slave datapath
module spisdtpth (input spisdclk, srds, sdrst, input [3:0] switch,
                 output reg [3:0] swled, input sclk, mosi, ss,
                 output reg miso);

reg [4:0] spisdstate=0;

always@(posedge spisdclk)
begin
    if (sdrst==1) // external reset
        begin
            swled=0;
            spisdstate=0;
        end

    case (spisdstate)
        0: begin // reset state
            if (srds==1) // slave read switches
                spisdstate=1;
            else if (ss==0)
                spisdstate=10;
        end

        1: begin // slave -> master transfer
            swled=switch;
            miso=swled[3]; // MSB
            if (ss==1)
                spisdstate=2;
        end

        2: begin
            if (sclk==1)
                spisdstate=3;
        end

        3: begin
            if (sclk==0)
                begin
                    miso=swled[2];
                    spisdstate=4;
                end
        end

        4: begin
            if (sclk==1)
                spisdstate=5;
        end

        5: begin
            if (sclk==0)

```

```

begin
    miso=swled[1];
    spisdstate=6;
end
end
6: begin
    if (sclk==1)
        spisdstate=7;
    end
7: begin
    if (sclk==0)
        begin
            miso=swled[0];    // LSB
            spisdstate=8;
        end
    end
8: begin
    if (sclk==1)
        spisdstate=9;
    end
9: begin
    if (sclk==0)
        spisdstate=0;
    end
10: begin
    // master -> slave transfer
    if (sclk==1)
        begin
            spisdstate=11;
            swled[3]=mosi;
        end
    end
11: begin
    if (sclk==0)
        spisdstate=12;
    end
12: begin
    if (sclk==1)
        begin
            spisdstate=13;
            swled[2]=mosi;    // MSB
        end
    end
13: begin
    if (sclk==0)
        spisdstate=14;

```

```
        end
14: begin
        if (sclk==1)
            begin
                spisdstate=15;
                swled[1]=mosi;
            end
        end
15: begin
        if (sclk==0)
            spisdstate=16;
        end
16: begin
        if (sclk==1)
            begin
                spisdstate=17;
                swled[0]=mosi; // LSB
            end
        end
17: begin
        if (sclk==0)
            spisdstate=0;
        end
    endcase
end
endmodule
```