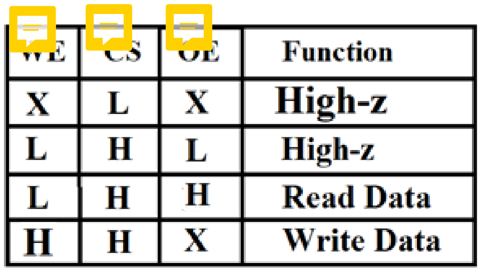
CPE166 Lab 4 Part 1
By: Prof. Pang

Lab 4

Part 1: SRAM Design

In your SRAM design, write 4’b1010 into the 32 address locations of the SRAM. After finishing writing, read data from RAM. Complete the design and run the test platform simulation in Verilog.

The SRAM is based on the following function table.



WE	CS	OE	Function
X	L	X	High-z
L	H	L	High-z
L	H	H	Read Data
H	H	X	Write Data

<u>Demo Requirement</u>

You need to demonstrate the simulation waveform of the final top level design work to your lab instructor.

CPE166 Lab 4 Part 1
By: Prof. Pang

Note: Before starting this experiment, all the necessary knowledge required to complete this work has been introduced in the CPE166 lecture session. The following examples are for you to refresh your learning.

Sample Verilog Codes:

```
 module ram ( address, data, cs, we, oe  );
 input      cs, we, oe   ;
 input [3:0] address     ;
 inout [7:0]  data       ;

 reg [7:0] data_out ;
 reg [7:0] mem [0:1023];      //10-bit address bus, 8-bit data bus.

assign data = (cs && oe &&  !we) ? data_out : 8'bzzzzzzzz;

 always @ (cs or we or data or address)
 if ( cs && we )
       mem[address] = data;


 always @ (cs or we or data or address or data)
 begin
   if (cs &&  !we && oe)
      data_out = mem[address];

  end

 endmodule
```

CPE166 Lab 4 Part 1
By: Prof. Pang

```
module mem_fsm ( clk, reset, address, data, cs, we, oe );

input            clk, reset;

output [3:0]   address;

inout  [7:0]     data;

output           cs, we, oe;

reg                cs, we, oe;

reg    [3:0]      address;

reg    [7:0]      data_reg;

reg    [2:0]     state;

parameter  idle = 3'b000, s1= 3'b001, .... //decide the numeber of states you need to have

assign data = data_reg;

 always@(posedge clk or posedge reset)

 begin

   if (reset)

   begin

        state      <= idle;

       address  <= 0;

   end

   else

     case (state)

       idle:   begin

              state      <= s1;

              address  <= 0;

              end


       s1:

             begin

             state <= s2;
```

```
            address   <= 0;
            end
      s2:   begin
            address <= address + 1;
            if(address == 15)
                state <= s3;
             else
               state <= s2;
            end
      ....     // describe other states
      default:
            begin
           state <= idle;
            address   <= 0;
            end
    endcase
  end

 always@(state)
 begin
      case (state)
      idle: begin
           cs = 0;
           we = 0;
           oe = 0;
           data_reg = 8'bZZZZZZZZ;
          end
      s1:   begin      //writing
```

```
            cs = 1;
            we = 1;
            oe = 0;
            data_reg = 8'b11000011;
           end
      s2:   begin        //writing
            cs = 1;
            we = 1;
            oe = 0;
            data_reg = 8'b11000011;
           end
      s3:   begin        //reading
            cs = 1;
            we = 0;
            oe = 1;
           end
      ....   // determine other states
           default:
           begin
            cs = 0;
            we = 0;
            oe = 0;
            data_reg = 8'bzzzzzzzz;
           end
      endcase
  end
endmodule
```

More Instructions

Next, you need to develop top.v and create one instance of ram and another instance of mem_fsm to be able to write and read data using ram.

At last, create testbench to run simulations.

Your Project Design

Use the same approach described above to complete your project based on the project description.

Lab4. Simplified Microprocessor Design

The purpose of parts 2, 3, and part 4 of Lab 4 is to design s simplified microprocessor. Figure 4-1 shows a simplified microprocessor diagram having M0, M1, M2 and Cin inputs, one input SW1 and also clock input. The SW1 serves as asynchronous reset function.

Your design should finally implement the following operation:

R2 = M0 + (not M1) + Cin

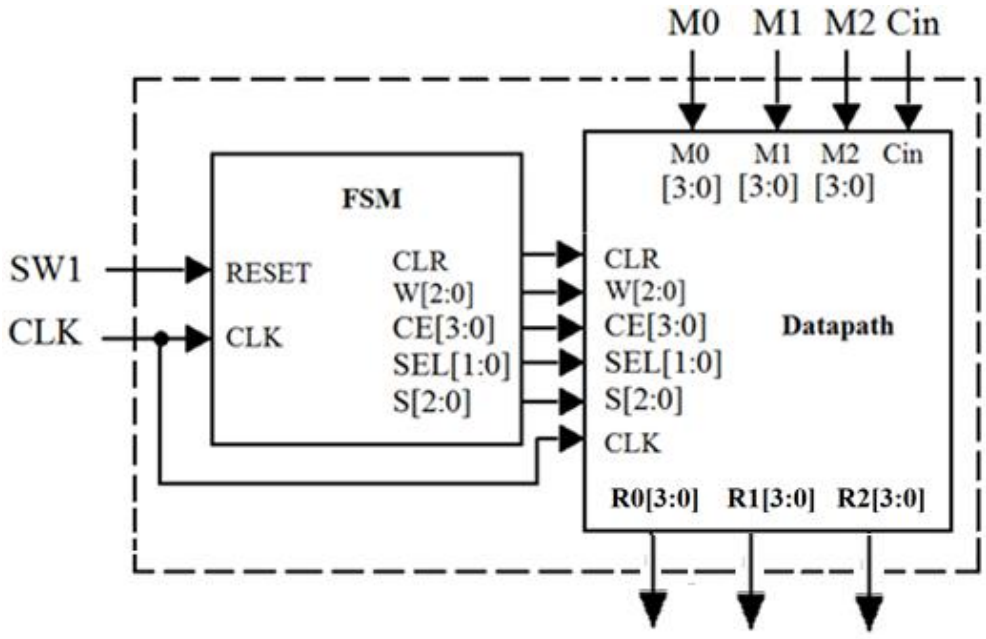


Figure 4-1. Simplified microprocessor block diagram

Part 2. Microprocessor Data Path Design

The detailed data path circuit is shown in Figure 4-2.

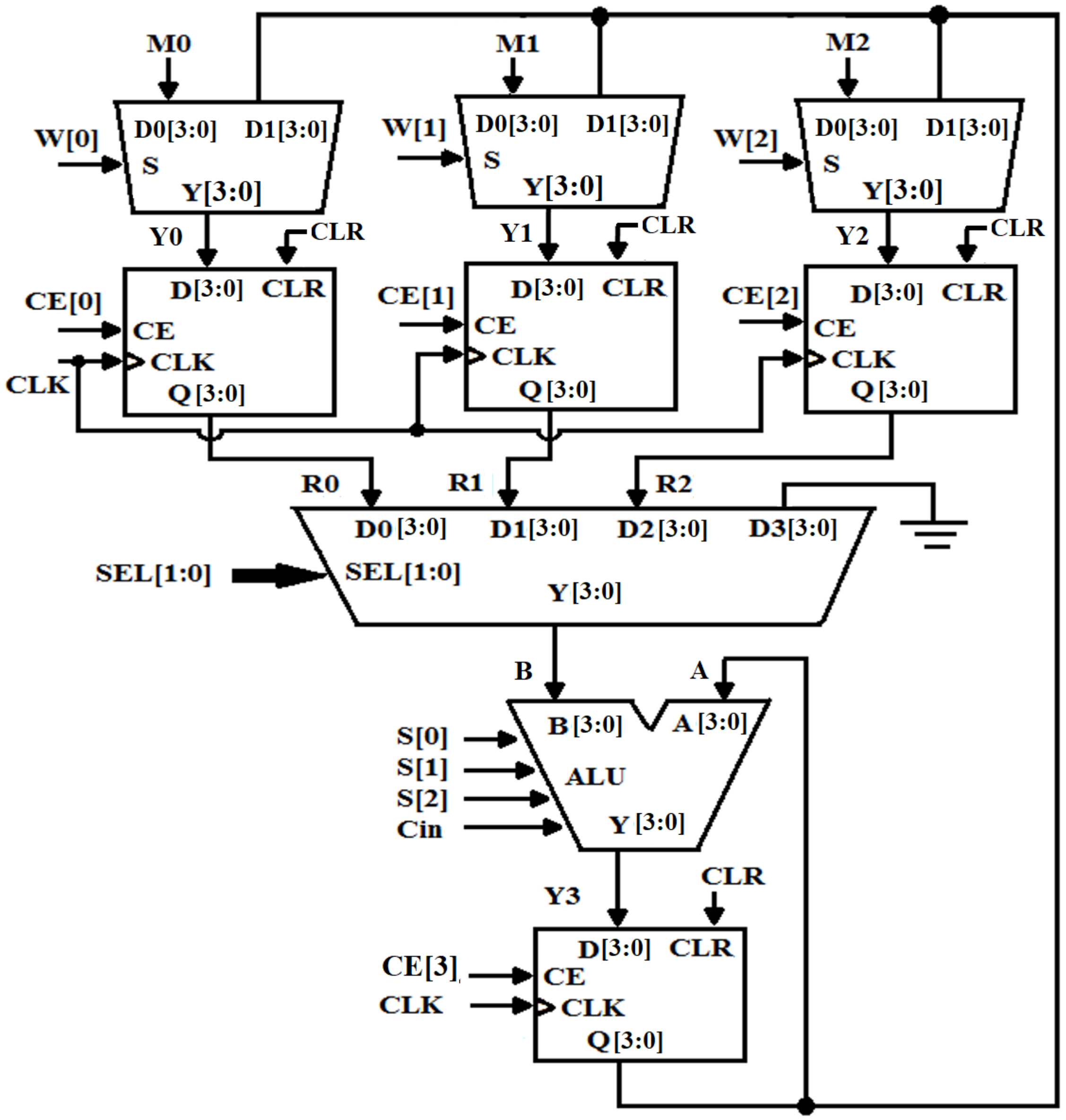


Figure 4-2. Simplified microprocessor data path circuit

The truth table of the ALU unit inside the data path circuit is shown in Table 4-1.

Table 4-1. ALU truth table

S[2]	**S[1]**	**S[0]**	**ALU Output F**
0	0	0	F=A+B+Cin
0	0	1	F=A+B'+Cin
0	1	0	F= B
0	1	1	F=A
1	0	0	F= A AND B
1	0	1	F= A OR B
1	1	0	F= A'
1	1	1	F = A XOR B

Use structural hierarchical design method to implement the data path circuit. Design and simulate each individual sub-design. Wire each sub-design together to implement your final data path circuit.

For this simplified design, all inputs and outputs are only required to be 4 bits, carry output signal won't be used. For example, A=$(0110)_2$, B=$(1100)_2$, Cin =1, then A+B+ Cin =$(0011)_2$.

<u>**Demo Requirement**</u>:

Show your testbench simulation of the top-level microprocessor data path design to your lab instructor.

Part 3. Microprocessor Control Path Design

Draw your finite state machine diagram for the “FSM” unit, which is used to implement the control path for the simplified microprocessor design shown in Figure 4-1. The control path block diagram is shown in Figure 4-3.

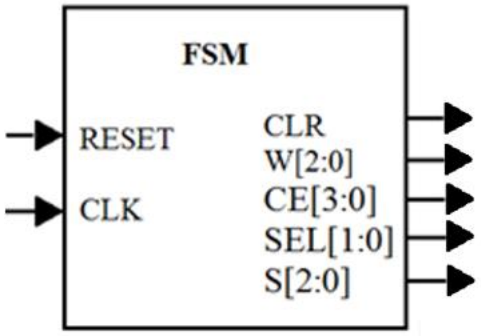


Figure 4-3. Simplified microprocessor control path block diagram

R0 and R1 are used to store operands M0 and M1 values.
R2 is used to store the final microprocessor operation result value, which is equal to:

M0 + (not M1) + Cin

<u>Demo Requirement</u>:

Show your testbench simulation of the microprocessor control path design to your lab instructor.

Part 4. Final 4-bit Microprocessor Design

This part is to complete the design shown in Figure 4-1 for the simplified 4-bit microprocessor.

Demo Requirement:

Demo your final simulation waveform of the top-level microprocessor design to your lab instructor.

Note, if you can demo part 4 successfully, you do not need to demo part 2, and part 3. After completing demo of part 4, you will get full credits of parts 2 ~ 4.

If you just demo part 2 or part 3 without completing part 4, you will be able to get credit for part 2 or part 3.