CPE166 Advanced Logic Design

Lab 4 Report

Student Name: Mario Palacios

Date: 05/10/2020

California State University, Sacramento

TABLE OF CONTENTS

Section **Page**

1) Introduction

A microprocessor involves the functions of a central processing unit on a single integrated circuit (IC). It is multipurpose, clock driven, register based, digital integrated circuit that accepts binary data as its input, and process it according to instructions stored in the memory, in order to provide an output. Microprocessor have a design that incorporates both combinational and sequential digital logic. For this lab we were tasked to design a simplified 4-bit microprocessor. Where it would take in six inputs; M0, M1, M2, SW1, and CLK. The design would implement the following operation: R2 = M0 + (not M1) + Cin. In order to make this possible we need to create the Data Path as seen in Figure 1, and a Finite State Machine (FSM) that will go through the steps to reach the operation.

2) Microprocessor Data Path Design – Project 1

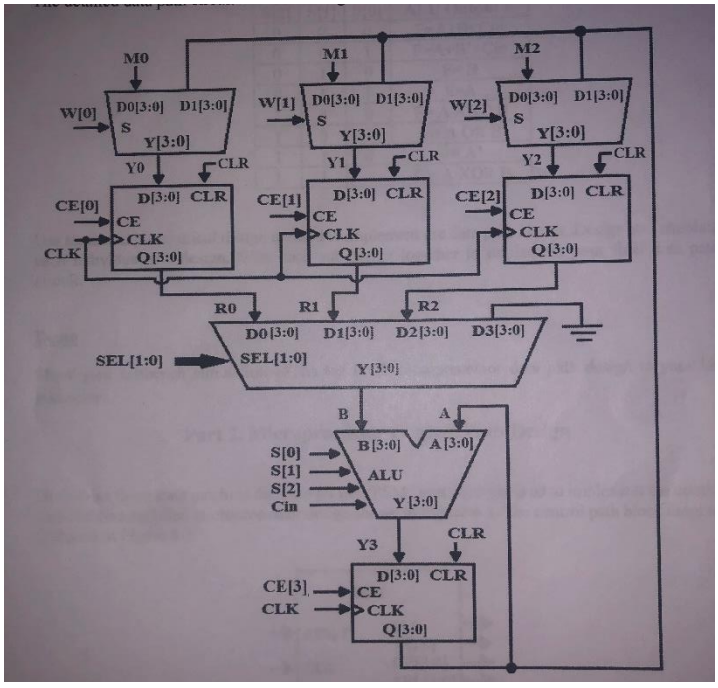

Figure 1. Simplified Microprocessor Data Path Circuit Schematic

2.2) Design Code

```
----------------------------------DATA PATH-----------------------------------

module dataPath(m0,m1,m2,clear,clock,select,cin,s,w,ce,r0,r1,r2);
   input       clock,clear,cin;
   input  [3:0] m0,m1,m2,ce;
   input  [2:0] s,w;
   input  [1:0] select;
   output [3:0] r0,r1,r2;
   wire   [3:0] y0,y1,y2,y3,a,b;

   mux2to1 g1(.d0(m0),.d1(a),.sel(w[0]),.y(y0));
```

```
    mux2to1 g2(.d0(m1),.d1(a),.sel(w[1]),.y(y1));
    mux2to1 g3(.d0(m2),.d1(a),.sel(w[2]),.y(y2));

    myDFF d1(.clk(clock),.clr(clear),.d(y0),.q(r0),.ce(ce[0]));
    myDFF d2(.clk(clock),.clr(clear),.d(y1),.q(r1),.ce(ce[1]));
    myDFF d3(.clk(clock),.clr(clear),.d(y2),.q(r2),.ce(ce[2]));
    myDFF d4(.clk(clock),.clr(clear),.d(y3),.q(a),.ce(ce[3]));

    mux4to1 g4(.d0(r0),.d1(r1),.d2(r2),.d3(4'b0000),.select(select),.y(b));

    myALU a1(.a(a),.b(b),.s(s),.cin(cin),.y(y3));
endmodule
```

2.3) Top Level Data Path Testbench

```
---------------------------------Testbench-----------------------------------

`timescale 1ns / 1ps
module dataPath_tb();
    reg       clock,clear,cin;
    reg  [3:0] m0,m1,m2,ce;
    reg  [2:0] s,w;
    reg  [1:0] select;
    wire [3:0] r0,r1,r2;

    dataPath uut(m0,m1,m2,clear,clock,select,cin,s,w,ce,r0,r1,r2);

    initial begin
        clear=1'b1; w=3'b0000; ce=4'b0000; select=2'b00; clock=1'b0;
        s = 3'b000; m0=4'b0101; m1=4'b0011; m2=4'b0000; cin=1;
    end
    always #1 clock = ~clock;

    initial begin
        #8 clear = 1'b0;
        #2 ce = 4'b0011; select = 2'b00; s = 3'b010;
        #2 ce = 4'b1011;
        #2 ce = 4'b0011; select = 2'b01; s = 3'b001;
        #2 ce = 4'b1000;
        #2 ce = 4'b0100; w = 3'b100;
        #2 w = 3'b000; ce = 4'b0000; select = 2'b00; s = 3'b000;
        #8 $stop;
    end
endmodule
```

2.3) Top Level Simulation Waveform

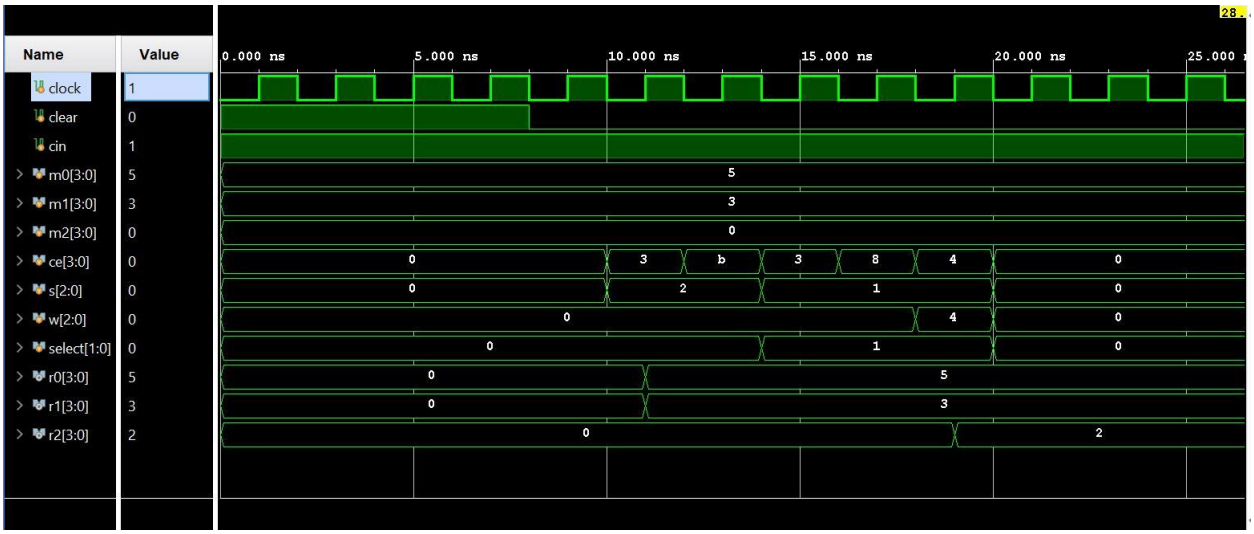

Figure 2. Data Path Simulation Waveform

2.4) Top Level Simulation Waveform Result Discussion

This part of the lab had the most work but it was not difficult because we had already done majority of the components in previous labs. The only new part was the Arithmetic Logic Unit (ALU). We can see in Figure 2; the MUX selects the proper D-Flip Flop and stores the data. While the ALU does all the proper calculations, in order to get the result of the operation that is stored in r2.

3) Microprocessor Controller Design – Project 2

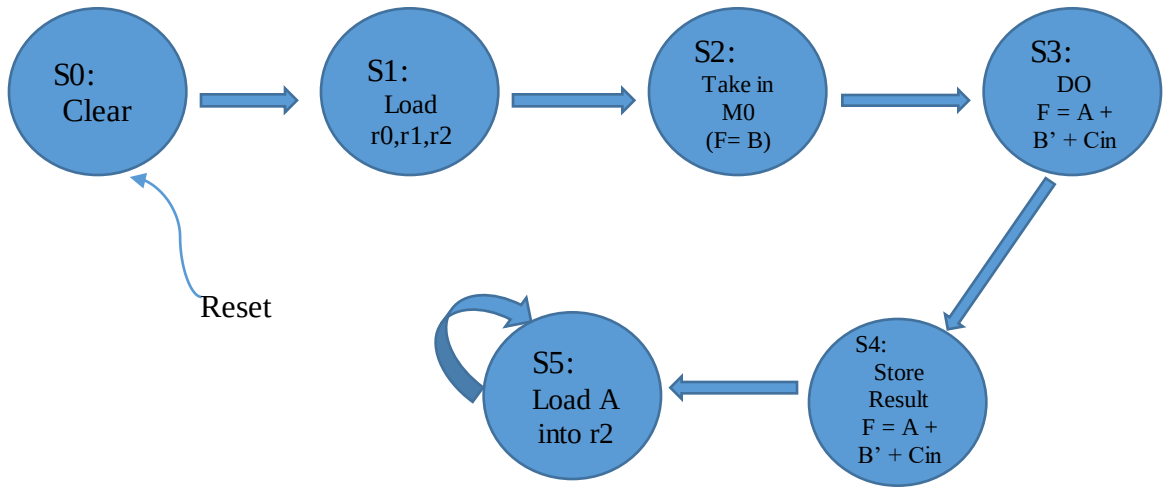

Figure 3. Finite State Machine Diagram

3.1) Controller Source Code

```
----------------------------------FSM-------------------------------------

module myFSM(clk,rst,clr,sel,s,ce,w);
   input clk,rst;
   output reg clr;
   output reg [3:0] ce;
   output reg [2:0] w,s;
   output reg [1:0] sel;

   reg [2:0] cs, ns;
   parameter s0=0,s1=1,s2=2,s3=3,s4=4,s5=5;

   always@(posedge clk or posedge rst) begin
      if(rst) begin
         cs <= s0;
      end else begin
         cs <= ns;
      end
   end

   always@(cs) begin
      case(cs)
         s0:
            begin
               clr = 1'b1;
               ce = 4'b0000;
               w = 3'b000;
               s = 3'b000;
               sel = 2'b11;
               ns <= s1;
            end

         s1:
            begin
               clr = 1'b0;
               ce = 4'b1111;
               w = 3'b000;
               s = 3'b010;
               sel = 2'b00;
               ns <= s2;
            end

         s2:
            begin
```

```
            clr = 1'b0;
            ce = 4'b1000;
            w = 3'b000;
            s = 3'b010;
            sel = 2'b00;
            ns <= s3;
         end

      s3:
         begin
            clr = 1'b0;
            ce = 4'b0000;
            w = 3'b000;
            s = 3'b001; //F = A + B' + CIN
            sel = 2'b01;
            ns <= s4;
         end

      s4:
         begin
            clr = 1'b0;
            ce = 4'b1100;
            w = 3'b100;
            s = 3'b001;
            sel = 2'b01;
            ns <= s5;
         end

      s5:
         begin
            clr = 1'b0;
            ce = 4'b1100;
            w = 3'b100;
            s = 3'b011;
            sel = 2'b11;
            ns <= s6;
         end
      default:
         ns = s0;
    endcase
  end
endmodule
```

3.2) Controller Testbench

```
---------------------------------Testbench-----------------------------------

`timescale 1ns / 1ps
module myFSM_tb();
   reg clk,rst;
   wire clr;
   wire [3:0] ce;
   wire [2:0] w,s;
   wire [1:0] sel;

   myFSM uut(clk,rst,clr,sel,s,ce,w);

   always #5 clk = ~clk;

   initial begin
      clk =0;
      #2 rst = 1;
      #10 rst = 0;
      #200$stop;
   end
endmodule
```

3.3) Controller Simulation Controller

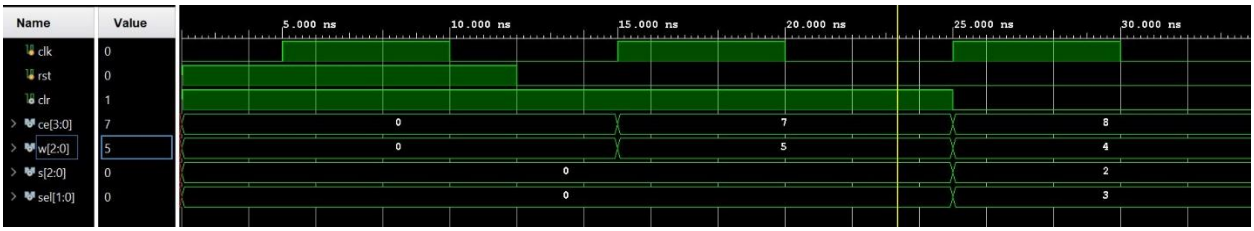

Figure 4. FSM Simulation Waveform

3.4) Controller Simulation Result Discussion

The FSM works as expected, and things work properly. It controls the inputs of the data path to load and to do the ALU calculations, and when it gets the calculation it stays on the last state. One of the hardest part of this was implementing the correct inputs and outputs.

4) Final Simplified Microprocessor Design– Project 3

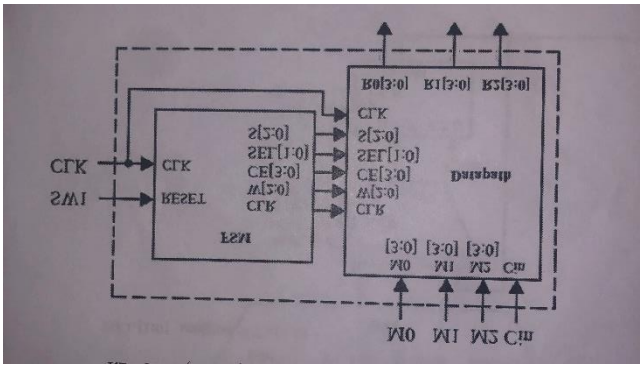

Figure 5. Final Simplified Microprocessor Block Diagram

4.1) Top Level Design Source Code

```
----------------------------------TOP LEVEL-------------------------------------

module topLVL(sw1,clk,m0,m1,m2,cin,r0,r1,r2);
   input          sw1,clk,cin;
   input      [3:0] m0,m1,m2;
   output wire [3:0] r0,r1,r2;

   wire clr;
   wire [3:0] ce;
   wire [2:0] w,s;
   wire [1:0] sel;

   dataPath
g1(.m0(m0),.m1(m1),.m2(m2),.clear(clr),.clock(clk),.select(sel),.cin(cin),.s(s),.w(w),.ce(ce),.r0
(r0),.r1(r1),.r2(r2));
   myFSM g2(.clk(clk),.rst(sw1),.clr(clr),.sel(sel),.s(s),.ce(ce),.w(w));
endmodule
```

4.2) Top Level Testbench

```
----------------------------------Testbench-------------------------------------

`timescale 1ns / 1ps
module top_tb();
   reg       sw1,clk,cin;
   reg  [3:0] m0,m1,m2;
   wire [3:0] r0,r1,r2;

   topLVL utt(sw1,clk,m0,m1,m2,cin,r0,r1,r2);
```

```
  initial clk = 0;
  always #5 clk = ~clk;
    initial begin
     sw1 = 1;
     m0  = 13;
     m1  = 3;
     m2  = 0;
     cin = 1;

     #10
     sw1 = 0;

     #70
     sw1 = 1;
     m0  = 2;
     m1  = 8;
     m2  = 5;
     cin = 0;

     #10
     sw1 = 0;

     #70 $stop;
  end
endmodule
```

4.3) Top Level Simulation Waveform

Figure 6. Top Level Simulation Waveform

4.4) Top Level Simulation Result Discussion

This part just was the easiest because it involved putting everything together. The results showed in r2 displays the correct calculation from the equation operation r2 = M0 + (not M1) + Cin. This can be seen in Figure 6, and the block diagram in Figure 5 helped me to name the variables and create a successful code.

5) Conclusion

In conclusion the main objective of this lab, was to design a simple microprocessor that implements the logic equation of r2 = M0 + (not M1) + Cin. The use of hierarchical design was used within different modules in order to link everything. This lab was a good way to practice everything that we learned throughout the semester. Overall the objectives for this lab were successfully accomplished and can be seen throughout my lab results.