Name and Student ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Lab Section:\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Submission Instructions:**

**Prelab:**

1. **Complete the prelab**
2. **Submit this report with the prelab completed to Canvas before your lab starts**

**Lab:**

1. **Complete the lab according to the instructions**
2. **Take screenshots of your ModelSim waveform (note: to receive points your NetID has to be present in the screenshot) and insert them into this document.**
3. **Include screenshots of any related block design files or Verilog files in the report**
4. **Complete this report and reupload it to Canvas**

**PRELAB:**

**Q1. Circuit diagram for a register file with four 2-bit registers.**

Please complete the following circuit diagram to implement a register file with four 2-bit registers, one write port, one read port, and one write enable line. Label all inputs and outputs of your circuit. Please use different colors for the different types of wires.



**The rest of this lab is about a register file with eight 4-bit registers.**

**Q2.** Write the Verilog code for a 4-bit 8-to-1 multiplexer below.

module Mux8\_4b(S, W0, W1, W2, W3, W4, W5, W6, W7, F);

 input [2:0] S;

 input [3:0] W0, W1, W2, W3, W4, W5, W6, W7;

 output [3:0] F;

 <<< insert code here >>>

endmodule

**Q3.** Write the Verilog code for a 3-to-8 decoder in the space below.

module Decoder3to8(EN,W,Y);

input EN;

input [2:0]W;

output [7:0]Y;

<<< insert code here >>>

endmodule

**Q4.** Using copies of the decoder, multiplexer, and 4-bit register from the previous steps, write the Verilog code that will provide the functionality of a register file with eight 4-bit registers. Your code should contain the decoder, the 4-bit registers, the multiplexers, and additional connections to make the whole circuit operational.

module regfile(DATAP,DATAQ,RP,RQ,WA,LD\_DATA,WR,CLRN,CLK);

 // address and control port

 input [2:0] RP, RQ, WA;

 input WR, CLRN, CLK;

 // input data port

 input [3:0] LD\_DATA;

 // output data ports

 output [3:0] DATAP,DATAQ;

 wire [3:0] VALUE0, VALUE1, VALUE2, VALUE3, VALUE4, VALUE5, VALUE6, VALUE7;

 wire [7:0] Y; //decoder ouput

 <<< insert code here >>>

endmodule

**LAB:**

**3.1 1-bit register**

Fill in the characteristic table for the one-bit parallel access register. CLRN is 1 at all time.

|  |  |  |
| --- | --- | --- |
| In | Load | Out |
| 0 | 0 |  |
| 1 | 0 |  |
| 0 | 1 |  |
| 1 | 1 |  |

**No waveform needed for the 1-bit register**

**3.2 4-bit register**

<<Insert a screenshot of the 4-bit register bdf file here>>

<<<Insert waveform of your multiplexer>>>

****

**4.0 Register File**

Fill in the table for the register file (section 4) with steps that will load the registers as follows: Reg[0]=F, Reg[1]=A. Reg[2]=C. Reg[3]=E, Reg[4]=2, Reg[5]=7, Reg[6]=6, and Reg[7]=1.

|  |  |
| --- | --- |
| LD\_DATA | WA |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**4.1 Multiplexer**

<<<Insert a screenshot of your multiplexer code here>>>

<<<Insert waveform of your multiplexer>>>

****

**4.2 Decoder**

<<<Insert a screenshot of your decoder code here>>>

<<<Insert waveform of your decoder >>>

****

**4.4 Bringing Everything Together (Register File)**

<<<Insert a screenshot of your register file code here>>>

<<<Insert waveform of your register file >>>

****

**5.0 Interaction of the Register File with Other Components (lab12\_final)**

Fill in the table below with the result produced by the register file (with CLRN=1).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LD\_DATA | Sel | WA | RP | RQ | CTRL | WR | Effect |
| 0110 | 0 | 111 | 111 | 111 | 0 | 1 | Reg[7]  6. |
| 0011 | 0 | 110 | 110 | 111 | 0 | 1 |  |
| 0010 | 0 | 101 | 101 | 110 | 1 | 1 |  |
| 0100 | 0 | 100 | 100 | 101 | 1 | 1 |  |
| 0101 | 0 | 011 | 011 | 100 | 0 | 1 |  |
| 0001 | 0 | 010 | 010 | 011 | 0 | 1 |  |
| 0111 | 0 | 001 | 001 | 010 | 1 | 1 |  |
| 1000 | 0 | 000 | 000 | 001 | 1 | 1 | Register File Contents: |
| 0000 | 1 | 001 | 000 | 001 | 0 | 1 |  |
| 0001 | 1 | 000 | 010 | 011 | 0 | 1 |  |
| 1111 | 1 | 010 | 100 | 101 | 1 | 1 |  |
| 1001 | 1 | 101 | 110 | 111 | 1 | 1 |  |
| 0100 | 1 | 010 | 010 | 101 | 1 | 0 | Register File Contents: |

<<<insert a screenshot of your lab12\_final bdf file>>>

<<< Insert a screenshot of your waveform of the implemented table above using DO file on the website (lab12\_final)>>>

****