**Digital Design and Computer Organization Lab**

**WEEK 5**

**13 Oct 2021**

|  |  |  |  |
| --- | --- | --- | --- |
| Vanshika Goel | **PES1UG20CS484** | H Section | Roll No: 40 |

**Integration of ALU and register files to create microprocessor**

**Input:**

module dfrl\_16(input wire clk,reset,load,input wire [0:15] in,output wire [0:15] out);

dfrl dfrl\_f0(clk,reset,load,in[0],out[0]);

dfrl dfrl\_f1(clk,reset,load,in[1],out[1]);

dfrl dfrl\_f2(clk,reset,load,in[2],out[2]);

dfrl dfrl\_f3(clk,reset,load,in[3],out[3]);

dfrl dfrl\_f4(clk,reset,load,in[4],out[4]);

dfrl dfrl\_f5(clk,reset,load,in[5],out[5]);

dfrl dfrl\_f6(clk,reset,load,in[6],out[6]);

dfrl dfrl\_f7(clk,reset,load,in[7],out[7]);

dfrl dfrl\_f8(clk,reset,load,in[8],out[8]);

dfrl dfrl\_f9(clk,reset,load,in[9],out[9]);

dfrl dfrl\_f10(clk,reset,load,in[10],out[10]);

dfrl dfrl\_f11(clk,reset,load,in[11],out[11]);

dfrl dfrl\_f12(clk,reset,load,in[12],out[12]);

dfrl dfrl\_f13(clk,reset,load,in[13],out[13]);

dfrl dfrl\_f14(clk,reset,load,in[14],out[14]);

dfrl dfrl\_f15(clk,reset,load,in[15],out[15]);

endmodule

module mux8\_16 (input wire[0:15] i0,i1,i2,i3,i4,i5,i6,i7,input wire [0:2] j,output wire [0:15] o);

mux8 mux8\_0({i0[0],i1[0],i2[0],i3[0],i4[0],i5[0],i6[0],i7[0]},j[0],j[1],j[2],o[0]);

mux8 mux8\_1({i0[1],i1[1],i2[1],i3[1],i4[1],i5[1],i6[1],i7[1]},j[0],j[1],j[2],o[1]);

mux8 mux8\_2({i0[2],i1[2],i2[2],i3[2],i4[2],i5[2],i6[2],i7[2]},j[0],j[1],j[2],o[2]);

mux8 mux8\_3({i0[3],i1[3],i2[3],i3[3],i4[3],i5[3],i6[3],i7[3]},j[0],j[1],j[2],o[3]);

mux8 mux8\_4({i0[4],i1[4],i2[4],i3[4],i4[4],i5[4],i6[4],i7[4]},j[0],j[1],j[2],o[4]);

mux8 mux8\_5({i0[5],i1[5],i2[5],i3[5],i4[5],i5[5],i6[5],i7[5]},j[0],j[1],j[2],o[5]);

mux8 mux8\_6({i0[6],i1[6],i2[6],i3[6],i4[6],i5[6],i6[6],i7[6]},j[0],j[1],j[2],o[6]);

mux8 mux8\_7({i0[7],i1[7],i2[7],i3[7],i4[7],i5[7],i6[7],i7[7]},j[0],j[1],j[2],o[7]);

mux8 mux8\_8({i0[8],i1[8],i2[8],i3[8],i4[8],i5[8],i6[8],i7[8]},j[0],j[1],j[2],o[8]);

mux8 mux8\_9({i0[9],i1[9],i2[9],i3[9],i4[9],i5[9],i6[9],i7[9]},j[0],j[1],j[2],o[9]);

mux8 mux8\_10({i0[10], i1[10], i2[10], i3[10], i4[10], i5[10], i6[10], i7[10]},j[0], j[1], j[2], o[10]);

mux8 mux8\_11({i0[11], i1[11], i2[11], i3[11], i4[11], i5[11], i6[11], i7[11]},j[0], j[1], j[2], o[11]);

mux8 mux8\_12({i0[12], i1[12], i2[12], i3[12], i4[12], i5[12], i6[12], i7[12]},j[0], j[1], j[2], o[12]);

mux8 mux8\_13({i0[13], i1[13], i2[13], i3[13], i4[13], i5[13], i6[13], i7[13]},j[0], j[1], j[2], o[13]);

mux8 mux8\_14({i0[14], i1[14], i2[14], i3[14], i4[14], i5[14], i6[14], i7[14]},j[0], j[1], j[2], o[14]);

mux8 mux8\_15({i0[15], i1[15], i2[15], i3[15], i4[15], i5[15], i6[15], i7[15]},j[0], j[1], j[2], o[15]);

endmodule

module reg\_file (input wire clk, reset, wr, input wire [2:0] rd\_addr\_a, rd\_addr\_b, wr\_addr, input wire [15:0] d\_in, output wire [15:0] d\_out\_a, d\_out\_b);

wire[0:7] load;

wire [0:15] dout\_0,dout\_1,dout\_2,dout\_3,dout\_4,dout\_5,dout\_6,dout\_7;

dfrl\_16 dfrl\_16\_0(clk, reset, load[0], d\_in,dout\_0);

dfrl\_16 dfrl\_16\_1(clk, reset, load[1], d\_in,dout\_0);

dfrl\_16 dfrl\_16\_2(clk, reset, load[2], d\_in,dout\_0);

dfrl\_16 dfrl\_16\_3(clk, reset, load[3], d\_in,dout\_0);

dfrl\_16 dfrl\_16\_4(clk, reset, load[4], d\_in,dout\_0);

dfrl\_16 dfrl\_16\_5(clk, reset, load[5], d\_in,dout\_0);

dfrl\_16 dfrl\_16\_6(clk, reset, load[6], d\_in,dout\_0);

dfrl\_16 dfrl\_16\_7(clk, reset, load[7], d\_in,dout\_0);

demux8 demux8\_0(wr, wr\_addr[2], wr\_addr[1], wr\_addr[0], load);

mux8\_16 mux8\_16\_9(dout\_0, dout\_1, dout\_2, dout\_3, dout\_4, dout\_5, dout\_6, dout\_7, rd\_addr\_a, d\_out\_a);

mux8\_16 mux8\_16\_10(dout\_0, dout\_1, dout\_2, dout\_3, dout\_4, dout\_5, dout\_6, dout\_7, rd\_addr\_b, d\_out\_b);

endmodule

module mux2\_16(input wire [15:0] i0, i1, input wire j,output wire [15:0] o);

mux2 mux2\_0(i0[0],i1[0],j,o[0]);

mux2 mux2\_1(i0[1],i1[1],j,o[1]);

mux2 mux2\_2(i0[2],i1[2],j,o[2]);

mux2 mux2\_3(i0[3],i1[3],j,o[3]);

mux2 mux2\_4(i0[4],i1[4],j,o[4]);

mux2 mux2\_5(i0[5],i1[5],j,o[5]);

mux2 mux2\_6(i0[6],i1[6],j,o[6]);

mux2 mux2\_7(i0[7],i1[7],j,o[7]);

mux2 mux2\_8(i0[8],i1[8],j,o[8]);

mux2 mux2\_9(i0[9],i1[9],j,o[9]);

mux2 mux2\_10(i0[10],i1[10],j,o[10]);

mux2 mux2\_11(i0[11],i1[11],j,o[11]);

mux2 mux2\_12(i0[12],i1[12],j,o[12]);

mux2 mux2\_13(i0[13],i1[13],j,o[13]);

mux2 mux2\_14(i0[14],i1[14],j,o[14]);

mux2 mux2\_15(i0[15],i1[15],j,o[15]);

endmodule

module reg\_alu (input wire clk, reset, sel, wr, input wire [1:0] op, input wire [2:0] rd\_addr\_a,

rd\_addr\_b, wr\_addr, input wire [15:0] d\_in, output wire [15:0] d\_out\_a, d\_out\_b, output wire cout);

wire [15:0] d\_in\_alu, d\_in\_reg;

wire cout\_0;

alu alu\_0(op,d\_out\_a,d\_out\_b,d\_in\_aolu,cout\_0);

reg\_file reg\_file\_0(clk,reset,wr,rd\_addr\_a,rd\_addr\_b,wr\_addr,d\_in\_reg,d\_out\_a,d\_out\_b);

mux2\_16 mux2\_16\_0(clk,reset,cout\_0,cout);

endmodule

**Output**

**Text

Description automatically generated**

**Graphical user interface

Description automatically generated**