**Digital Design and Computer Organization Laboratory**

**UE20CS206**

**3rd Semester, Academic Year 2021-22**

Date:

|  |  |  |
| --- | --- | --- |
| Name : SUNDEEP A | SRN : PES1UG20CS445 | Section : H |

Experiment Number: Week : 5

**Title of the Program: Reg-Alu**

**Code: in Reg\_Alu.v file**

module dfrl16(input wire clk,reset, load, input wire [15:0]in, output wire [15:0]out);

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

endmodule

module mux8\_16(input wire [15:0]i0,i1,i2,i3,i4,i5,i6,i7, input wire j0, j1, j2, output wire [15:0] o);

mux8 m8\_0({i0[0], i1[0], i2[0], i3[0], i4[0], i5[0], i6[0], i7[0]}, j0,j1,j2, o[0]);

mux8 m8\_1({i0[1], i1[1], i2[1], i3[1], i4[1], i5[1], i6[1], i7[1]}, j0,j1,j2, o[1]);

mux8 m8\_2({i0[2], i1[2], i2[2], i3[2], i4[2], i5[2], i6[2], i7[2]}, j0,j1,j2, o[2]);

mux8 m8\_3({i0[3], i1[3], i2[3], i3[3], i4[3], i5[3], i6[3], i7[3]}, j0,j1,j2, o[3]);

mux8 m8\_4({i0[4], i1[4], i2[4], i3[4], i4[4], i5[4], i6[4], i7[4]}, j0,j1,j2, o[4]);

mux8 m8\_5({i0[5], i1[5], i2[5], i3[5], i4[5], i5[5], i6[5], i7[5]}, j0,j1,j2, o[5]);

mux8 m8\_6({i0[6], i1[6], i2[6], i3[6], i4[6], i5[6], i6[6], i7[6]}, j0,j1,j2, o[6]);

mux8 m8\_7({i0[7], i1[7], i2[7], i3[7], i4[7], i5[7], i6[7], i7[7]}, j0,j1,j2, o[7]);

mux8 m8\_8({i0[8], i1[8], i2[8], i3[8], i4[8], i5[8], i6[8], i7[8]}, j0,j1,j2, o[8]);

mux8 m8\_9({i0[9], i1[9], i2[9], i3[9], i4[9], i5[9], i6[9], i7[9]}, j0,j1,j2, o[9]);

mux8 m8\_10({i0[10], i1[10], i2[10], i3[10], i4[10], i5[10], i6[10], i7[10]}, j0,j1,j2, o[10]);

mux8 m8\_11({i0[11], i1[11], i2[11], i3[11], i4[11], i5[11], i6[11], i7[11]}, j0,j1,j2, o[11]);

mux8 m8\_12({i0[12], i1[12], i2[12], i3[12], i4[12], i5[12], i6[12], i7[12]}, j0,j1,j2, o[12]);

mux8 m8\_13({i0[13], i1[13], i2[13], i3[13], i4[13], i5[13], i6[13], i7[13]}, j0,j1,j2, o[13]);

mux8 m8\_14({i0[14], i1[14], i2[14], i3[14], i4[14], i5[14], i6[14], i7[14]}, j0,j1,j2, o[14]);

mux8 m8\_15({i0[15], i1[15], i2[15], i3[15], i4[15], i5[15], i6[15], i7[15]}, j0,j1,j2, 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] r0, r1, r2, r3, r4, r5, r6, r7;

    dfrl16 reg0(clk,reset,load[0],d\_in,r0);

    dfrl16 reg1(clk,reset,load[1],d\_in,r1);

    dfrl16 reg2(clk,reset,load[2],d\_in,r2);

    dfrl16 reg3(clk,reset,load[3],d\_in,r3);

    dfrl16 reg4(clk,reset,load[4],d\_in,r4);

    dfrl16 reg5(clk,reset,load[5],d\_in,r5);

    dfrl16 reg6(clk,reset,load[6],d\_in,r6);

    dfrl16 reg7(clk,reset,load[7],d\_in,r7);

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

    mux8\_16 mm0(r0,r1,r2,r3,r4,r5,r6,r7,rd\_addr\_a[0], rd\_addr\_a[1], rd\_addr\_a[2],d\_out\_a);

    mux8\_16 mm1(r0,r1,r2,r3,r4,r5,r6,r7,rd\_addr\_b[0], rd\_addr\_b[1], rd\_addr\_b[2],d\_out\_b);

endmodule

module mux2for16(input wire [15:0] din\_regular, alu\_out, input wire selector, output wire [15:0]din\_final);

    mux2 m0(din\_regular[0], alu\_out[0], selector, din\_final[0]);

    mux2 m1(din\_regular[1], alu\_out[1], selector, din\_final[1]);

    mux2 m2(din\_regular[2], alu\_out[2], selector, din\_final[2]);

    mux2 m3(din\_regular[3], alu\_out[3], selector, din\_final[3]);

    mux2 m4(din\_regular[4], alu\_out[4], selector, din\_final[4]);

    mux2 m5(din\_regular[5], alu\_out[5], selector, din\_final[5]);

    mux2 m6(din\_regular[6], alu\_out[6], selector, din\_final[6]);

    mux2 m7(din\_regular[7], alu\_out[7], selector, din\_final[7]);

    mux2 m8(din\_regular[8], alu\_out[8], selector, din\_final[8]);

    mux2 m9(din\_regular[9], alu\_out[9], selector, din\_final[9]);

    mux2 m10(din\_regular[10], alu\_out[10], selector, din\_final[10]);

    mux2 m11(din\_regular[11], alu\_out[11], selector, din\_final[11]);

    mux2 m12(din\_regular[12], alu\_out[12], selector, din\_final[12]);

    mux2 m13(din\_regular[13], alu\_out[13], selector, din\_final[13]);

    mux2 m14(din\_regular[14], alu\_out[14], selector, din\_final[14]);

    mux2 m15(din\_regular[15], alu\_out[15], selector, din\_final[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] alu\_out;

    wire [15:0] newdin;

    mux2for16 select(d\_in, alu\_out, sel, newdin);

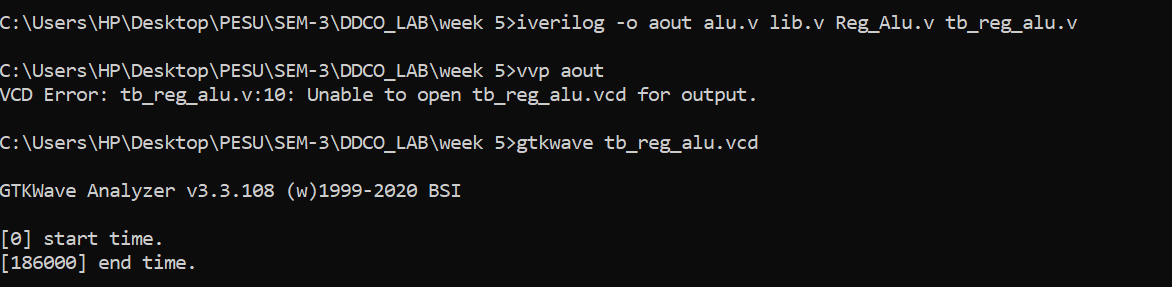
    reg\_file new\_reg(clk, reset, wr, rd\_addr\_a, rd\_addr\_b, wr\_addr, newdin, d\_out\_a, d\_out\_b);

    alu calc(op, d\_out\_a, d\_out\_b, alu\_out, cout);

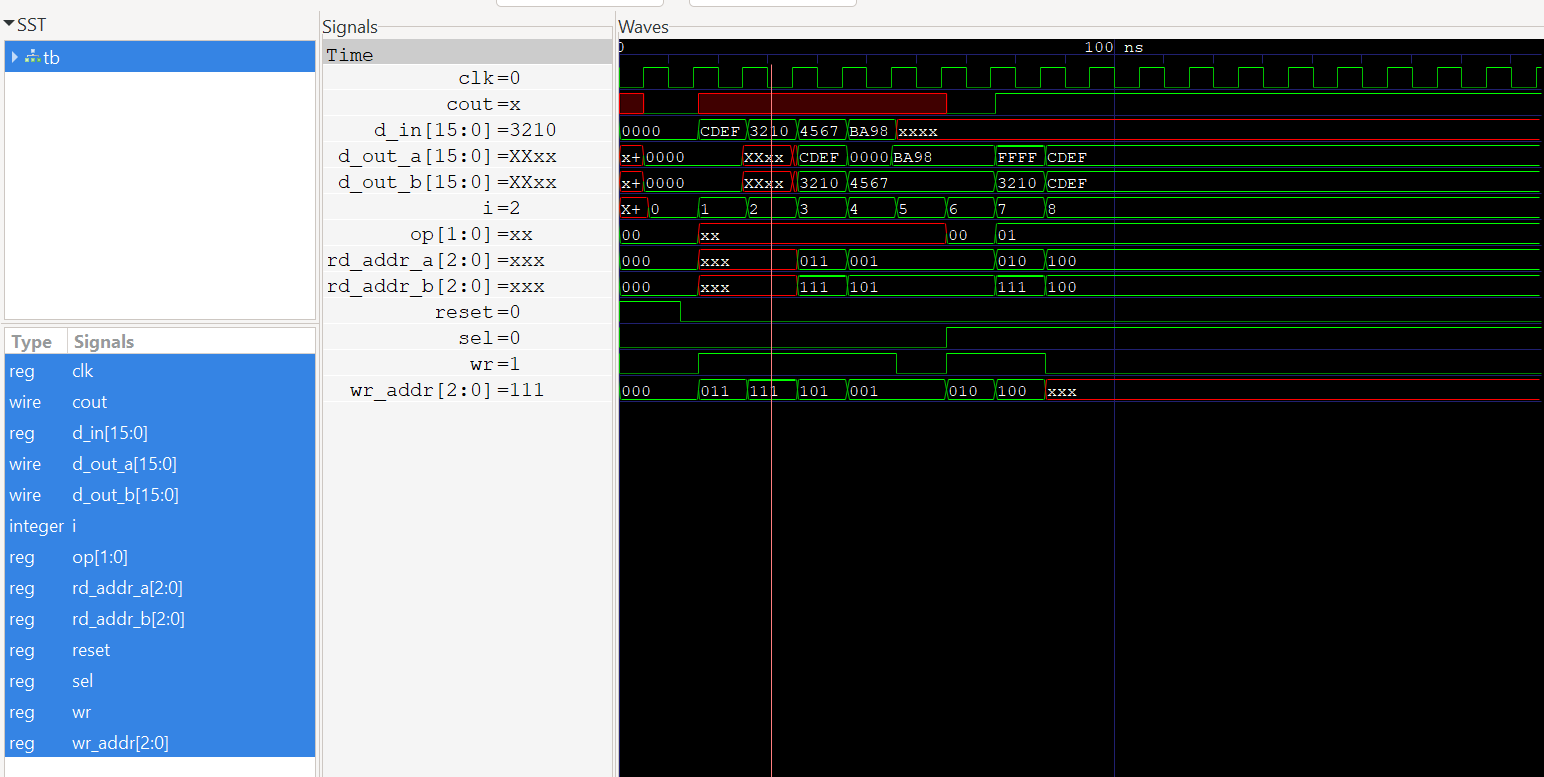
endmodule

**Output waveform :**

**In Terminal**

****

**In gtkwave**

****