Digital Design and Computer Organization Laboratory

UE19CS206

3rd Semester, Academic Year 2020-21

Date:

Name : TUSHAR Y S	SRN : PES1UG19CS545	Section : I

Experiment Number: 5 Week # : 6

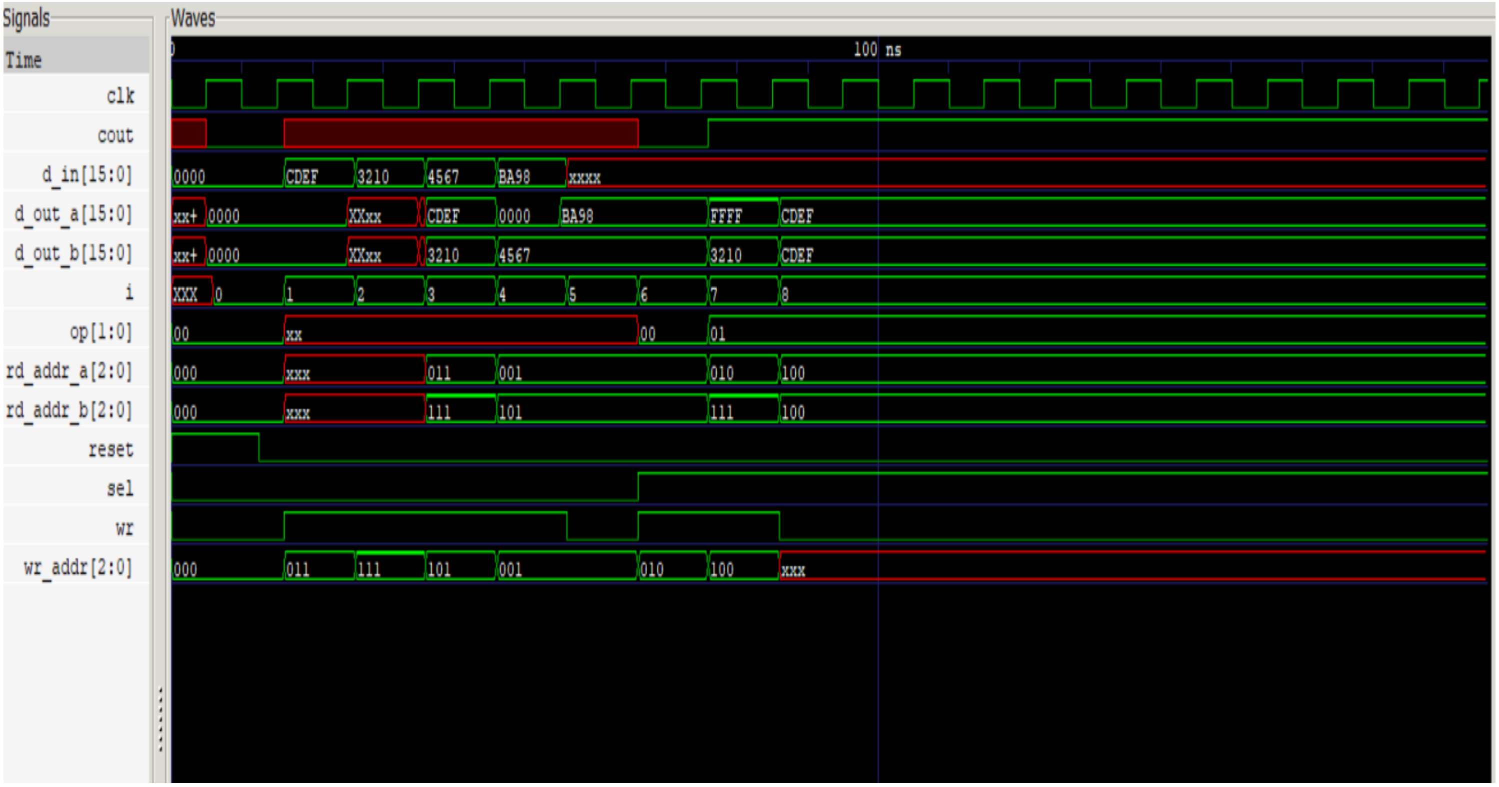
Title of the Program: Reg-ALU

Code:

```
module reg_16(input wire clk,reset,load,input wire [15:0]din,output wire [15:0]r);
        dfrl d0(clk,reset,load,din[0],r[0]);
        dfrl d1(clk,reset,load,din[1],r[1]);
        dfrl d2(clk,reset,load,din[2],r[2]);
        dfrl d3(clk,reset,load,din[3],r[3]);
        dfrl d4(clk,reset,load,din[4],r[4]);
        dfrl d5(clk,reset,load,din[5],r[5]);
        dfrl d6(clk,reset,load,din[6],r[6]);
        dfrl d7(clk,reset,load,din[7],r[7]);
        dfrl d8(clk,reset,load,din[8],r[8]);
        dfrl d9(clk,reset,load,din[9],r[9]);
        dfrl d10(clk,reset,load,din[10],r[10]);
        dfrl d11(clk,reset,load,din[11],r[11]);
        dfrl d12(clk,reset,load,din[12],r[12]);
        dfrl d13(clk,reset,load,din[13],r[13]);
        dfrl d14(clk,reset,load,din[14],r[14]);
        dfrl d15(clk,reset,load,din[15],r[15]);
endmodule

module mux8_16(input wire [15:0]i0,i1,i2,i3,i4,i5,i6,i7,input wire [2:0]j ,output wire [15:0]o);

        mux8 mux0({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 mux1({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 mux2({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 mux3({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 mux4({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 mux5({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 mux6({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 mux7({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({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 mux9({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 mux10({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 mux11({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 mux12({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 mux13({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 mux14({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 mux15({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]r0,r1,r2,r3,r4,r5,r6,r7;

        demux8 dmx(wr,wr_addr[2],wr_addr[1],wr_addr[0],load);
        reg_16 reg0(clk,reset,load[0],d_in,r0);
        reg_16 reg1(clk,reset,load[1],d_in,r1);
        reg_16 reg2(clk,reset,load[2],d_in,r2);
        reg_16 reg3(clk,reset,load[3],d_in,r3);
        reg_16 reg4(clk,reset,load[4],d_in,r4);
        reg_16 reg5(clk,reset,load[5],d_in,r5);
        reg_16 reg6(clk,reset,load[6],d_in,r6);
        reg_16 reg7(clk,reset,load[7],d_in,r7);
        mux8_16 m0(r0,r1,r2,r3,r4,r5,r6,r7,rd_addr_a, d_out_a);
        mux8_16 m1(r0,r1,r2,r3,r4,r5,r6,r7,rd_addr_b, 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


Signals
Waves
Time
0
100 ns
clk
cout
d_in[15:0]
0000
CDEF
3210
4567
BA98
xxxx
d_out_a[15:0]
xx+
0000
XXxx
CDEF
0000
BA98
FFFF
CDEF
d_out_b[15:0]
xx+
0000
XXxx
3210
4567
3210
CDEF
i
XXX
0
1
2
3
4
5
6
7
8
op[1:0]
00
xx
00
01
rd_addr_a[2:0]
000
xxx
011
001
010
100
rd_addr_b[2:0]
000
xxx
111
101
111
100
reset
sel
wr
wr_addr[2:0]
000
011
111
101
001
010
100
xxx