**Digital Design and Computer Organization Laboratory**

**UE19CS206**

**3rd Semester, Academic Year 2020-21**

Date: 18/11/2020

|  |  |  |
| --- | --- | --- |
| Name :  Pramatha Gajanan Bhat | SRN :  PES1UG19CS339 | Section :  F |

Experiment Number: Week # : 8

**Title of the Program:**

**Microprocessor Control Logic – 2 : Load and jump Instructions**

**Code:**

**module nor5 (input wire [0:4] i, output wire o);**

**wire t;**

**or3 or3\_0 (i[0], i[1], i[2], t);**

**nor3 nor3\_0 (t, i[3], i[4], o);**

**endmodule**

**module ir (input wire clk, reset, load, input wire [15:0] din, output wire [15:0] dout);**

**dfrl dfrl\_0 (clk, reset, load, din['h0], dout['h0]);**

**dfrl dfrl\_1 (clk, reset, load, din['h1], dout['h1]);**

**dfrl dfrl\_2 (clk, reset, load, din['h2], dout['h2]);**

**dfrl dfrl\_3 (clk, reset, load, din['h3], dout['h3]);**

**dfrl dfrl\_4 (clk, reset, load, din['h4], dout['h4]);**

**dfrl dfrl\_5 (clk, reset, load, din['h5], dout['h5]);**

**dfrl dfrl\_6 (clk, reset, load, din['h6], dout['h6]);**

**dfrl dfrl\_7 (clk, reset, load, din['h7], dout['h7]);**

**dfrl dfrl\_8 (clk, reset, load, din['h8], dout['h8]);**

**dfrl dfrl\_9 (clk, reset, load, din['h9], dout['h9]);**

**dfrl dfrl\_a (clk, reset, load, din['ha], dout['ha]);**

**dfrl dfrl\_b (clk, reset, load, din['hb], dout['hb]);**

**dfrl dfrl\_c (clk, reset, load, din['hc], dout['hc]);**

**dfrl dfrl\_d (clk, reset, load, din['hd], dout['hd]);**

**dfrl dfrl\_e (clk, reset, load, din['he], dout['he]);**

**dfrl dfrl\_f (clk, reset, load, din['hf], dout['hf]);**

**endmodule**

**module control\_logic (input wire clk, reset, input wire [15:0] cur\_ins, output wire [2:0] rd\_addr\_a, rd\_addr\_b, wr\_addr,**

**output wire [1:0] op, output wire sel, jump, pc\_inc, load\_ir, wr\_reg);**

**// Copy your assignment 3 logic here and modify.**

**wire x,w,y,wr\_reg1,wr\_reg2,alu\_ins,ld\_ins,ld\_ins\_,f1,f0,e1,e0,e2 ;**

**assign rd\_addr\_a[0] = cur\_ins[0];**

**assign rd\_addr\_a[1] = cur\_ins[1];**

**assign rd\_addr\_a[2] = cur\_ins[2];**

**assign rd\_addr\_b[0] = cur\_ins[3];**

**assign rd\_addr\_b[1] = cur\_ins[4];**

**assign rd\_addr\_b[2] = cur\_ins[5];**

**assign wr\_addr[0] = cur\_ins[6];**

**assign wr\_addr[1] = cur\_ins[7];**

**assign wr\_addr[2] = cur\_ins[8];**

**assign op[0] = cur\_ins[9];**

**assign op[1] = cur\_ins[10];**

**invert in1(cur\_ins[15],x);**

**invert in2(cur\_ins[10],w);**

**invert in3(cur\_ins[14],y);**

**invert in4(ld\_ins,ld\_ins\_);**

**and2 a1(cur\_ins[15],s,ld\_ins);**

**nor5 n5({cur\_ins[15],cur\_ins[14],cur\_ins[13],cur\_ins[12],cur\_ins[11]},alu\_ins);**

**and3 a2(cur\_ins[14],x,e2,jump);**

**dfrl d1(clk,reset,1'b1,f1,e0);**

**and2 a3(ld\_ins\_,e0,e2);**

**and2 a4(e2,alu\_ins,wr\_reg1);**

**or2 o3(wr\_reg1,wr\_reg2,wr\_reg);**

**and2 a5(e0,ld\_ins,e1);**

**and2 a6(ld\_ins,e1,wr\_reg2);**

**nand2 n1(e1,ld\_ins,sel);**

**dfrl d2(clk,reset,1'b1,e1,lo);**

**or2 o1(lo,e2,f1);**

**dfsl d3(clk,reset,1'b1,f1,f0);**

**assign load\_ir = f0;**

**or2 o2(load\_ir,e1,pc\_inc);**

**endmodule**

**module mproc (input wire clk, reset, input wire [15:0] d\_in, output wire [6:0] addr, output wire [15:0] d\_out);**

**wire pc\_inc, cout, cout\_, sub, sel, sel\_addr; wire [2:0] rd\_addr\_a, rd\_addr\_b, wr\_addr; wire [1:0] op; wire [8:0] \_addr;**

**wire [15:0] cur\_ins, d\_out\_a, d\_out\_b;**

**and2 and2\_0 (jump, cout, sub);**

**pc pc\_0 (clk, reset, pc\_inc, 1'b0, sub, {8'b0, cur\_ins[7:0]}, {\_addr, addr});**

**ir ir\_0 (clk, reset, load\_ir, d\_in, cur\_ins);**

**control\_logic control\_logic\_0 (clk, reset, cur\_ins, rd\_addr\_a, rd\_addr\_b, wr\_addr, op, sel, jump, pc\_inc, load\_ir, wr\_reg);**

**reg\_alu reg\_alu\_0 (clk, reset, sel, wr\_reg, op, rd\_addr\_a, rd\_addr\_b, wr\_addr, d\_in, d\_out\_a, d\_out\_b, cout);**

**assign d\_out = d\_out\_a;**

**endmodule**

**Output waveform**

