# Digital Design and Computer Organization Laboratory

# UE19CS206

# $3^{rd}$ Semester, Academic Year 2020-21

Date: 18/11/2020

| | | |
|---|---|---|
| Name :<br>SUMUKH RAJU BHAT | SRN :<br>PES1UG19CS519 | Section :<br>H |

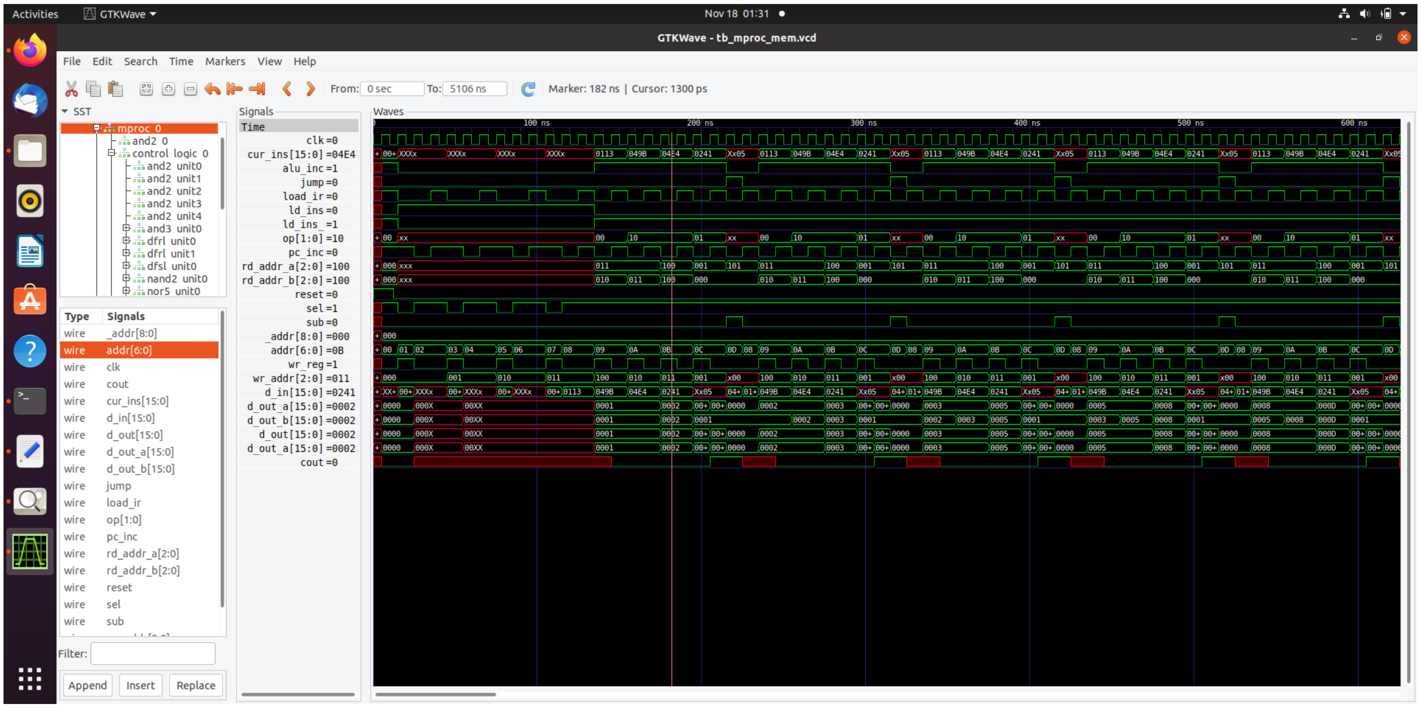
Experiment Number: 8 Week # : 9

## Title of the Program: MICROPROCESSOR-2 (Control logic with load and jump instructions)

## Code: (Entire mproc.v)

```
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


// REQUIRED LOGIC!!!!!!!!!!!
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 alu_inc,s,ld_ins,ld_ins_,w,u,ef,fi,f0,el,e0, wr_reg1,wr_reg2,l0,one;
   assign one = 1'b1;
   wire [0:4] ins;
   assign ins = cur_ins[15:11];
   assign rd_addr_a = cur_ins[2:0];
   assign rd_addr_b = cur_ins[5:3];
   assign wr_addr = cur_ins[8:6];
   assign op = cur_ins[10:9];
   nor5 nor5_unit0(ins,alu_inc);
   invert not_unit0(cur_ins[14],s);
   and2 and2_unit0(cur_ins[15],s,ld_ins);
   invert not_unit1(ld_ins,ld_ins_);
   invert not_unit2(cur_ins[10],w);
   invert not_unit3(cur_ins[15],u);
   and3 and3_unit0(cur_ins[14],u,ef,jump);
   dfsl dfsl_unit0(clk,reset,one,fi,f0);
   assign load_ir = f0;
   or2 or2_unit0(f0,el,pc_inc);
   dfrl dfrl_unit0(clk,reset,one,f0,e0);
   and2 and2_unit1(e0,ld_ins_,ef);
   and2 and2_unit2(ef,alu_inc,wr_reg1);
   and2 and2_unit3(e0,ld_ins,el);
   and2 and2_unit4(el,ld_ins,wr_reg2);
   or2 or2_unit1(wr_reg1,wr_reg2,wr_reg);
   nand2 nand2_unit0(el,ld_ins,sel);
   dfrl dfrl_unit1(clk,reset,one,el,l0);
   or2 or2_unit2(l0,ef,fi);

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



Activities GTKWave ▾ Nov 18 01:31 ●

GTKWave - tb_mproc_mem.vcd

File Edit Search Time Markers View Help

From: 0 sec To: 5106 ns Marker: 182 ns | Cursor: 1300 ps

SST

```
mproc 0
  and2 0
  control logic 0
    and2 unit0
    and2 unit1
    and2 unit2
    and2 unit3
    and2 unit4
    and3 unit0
    dfrl unit0
    dfrl unit1
    dfsl unit0
    nand2 unit0
    nor5 unit0
```

| Type | Signals |
|---|---|
| wire | _addr[8:0] |
| wire | addr[6:0] |
| wire | clk |
| wire | cout |
| wire | cur_ins[15:0] |
| wire | d_in[15:0] |
| wire | d_out[15:0] |
| wire | d_out_a[15:0] |
| wire | d_out_b[15:0] |
| wire | jump |
| wire | load_ir |
| wire | op[1:0] |
| wire | pc_inc |
| wire | rd_addr_a[2:0] |
| wire | rd_addr_b[2:0] |
| wire | reset |
| wire | sel |
| wire | sub |

Filter:

Append Insert Replace

Signals

```
Time
               clk =0
   cur_ins[15:0] =04E4
           alu_inc =1
              jump =0
           load_ir =0
            ld_ins =0
           ld_ins_ =1
          op[1:0] =10
            pc_inc =0
  rd_addr_a[2:0] =100
  rd_addr_b[2:0] =100
             reset =0
               sel =1
               sub =0
      _addr[8:0] =000
       addr[6:0] =0B
            wr_reg =1
   wr_addr[2:0] =011
      d_in[15:0] =0241
   d_out_a[15:0] =0002
   d_out_b[15:0] =0002
     d_out[15:0] =0002
   d_out_a[15:0] =0002
              cout =0
```

Waves