**Digital Design and Computer Organization Laboratory**

**UE19CS206**

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

Date: 21/9/2020

|  |  |  |
| --- | --- | --- |
| Name :  SUMUKH RAJU BHAT | SRN :  PES1UG19CS519 | Section :  H |

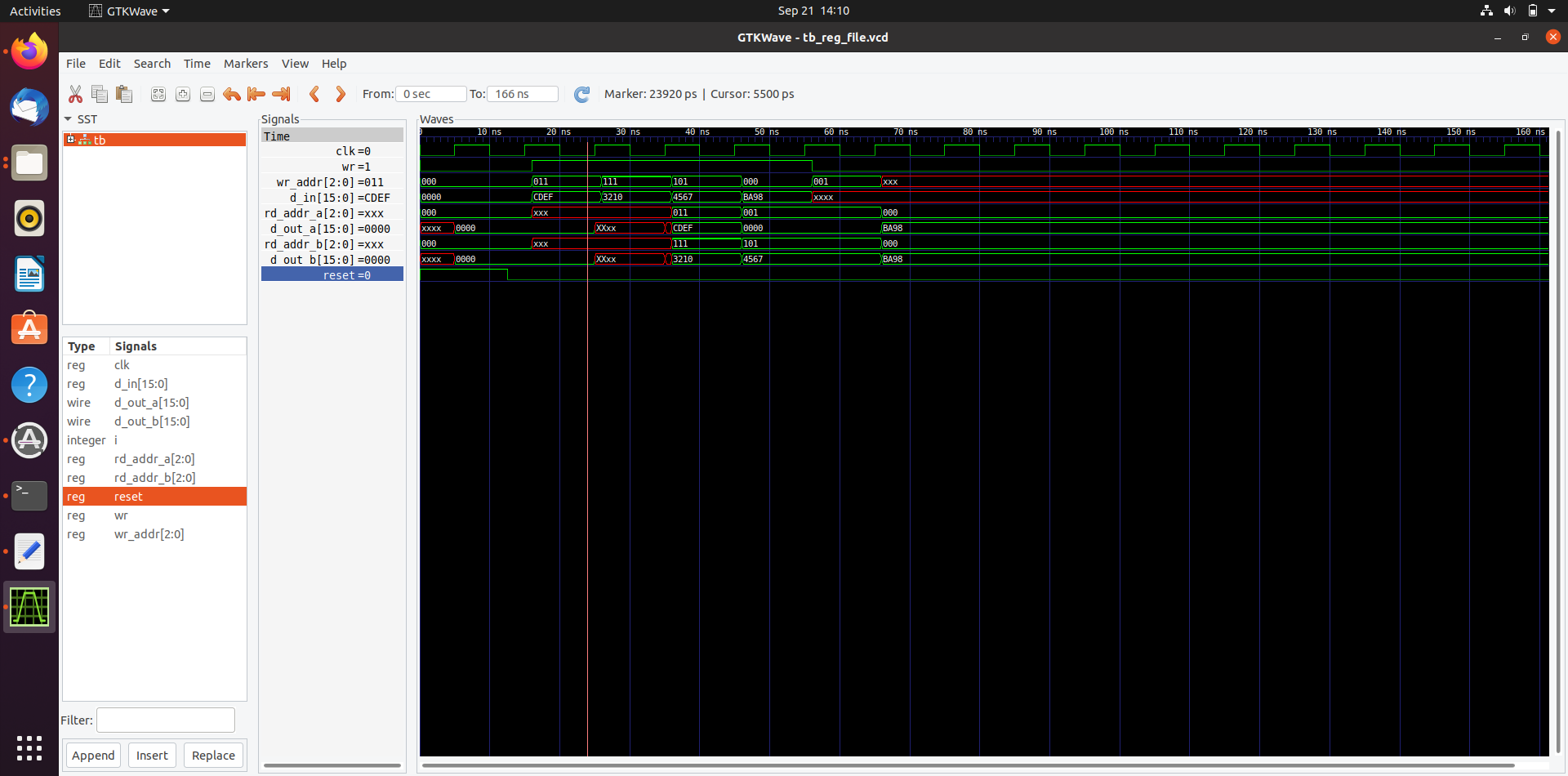
Experiment Number: 4 Week # : 5

**Title of the Program: REGISTER FILE**

**Code:**

// Write code for modules you need here  
module reg\_unit (input wire clk,reset,load, input wire [0:15] in,output wire [0:15] out);  
  
    dfrl dfr\_circuit\_a(clk,reset,load,in[0],out[0]);  
    dfrl dfr\_circuit\_b(clk,reset,load,in[1],out[1]);  
    dfrl dfr\_circuit\_c(clk,reset,load,in[2],out[2]);  
    dfrl dfr\_circuit\_d(clk,reset,load,in[3],out[3]);  
    dfrl dfr\_circuit\_e(clk,reset,load,in[4],out[4]);  
    dfrl dfr\_circuit\_f(clk,reset,load,in[5],out[5]);  
    dfrl dfr\_circuit\_g(clk,reset,load,in[6],out[6]);  
    dfrl dfr\_circuit\_h(clk,reset,load,in[7],out[7]);  
    dfrl dfr\_circuit\_i(clk,reset,load,in[8],out[8]);  
    dfrl dfr\_circuit\_j(clk,reset,load,in[9],out[9]);  
    dfrl dfr\_circuit\_k(clk,reset,load,in[10],out[10]);  
    dfrl dfr\_circuit\_l(clk,reset,load,in[11],out[11]);  
    dfrl dfr\_circuit\_m(clk,reset,load,in[12],out[12]);  
    dfrl dfr\_circuit\_n(clk,reset,load,in[13],out[13]);  
    dfrl dfr\_circuit\_o(clk,reset,load,in[14],out[14]);  
    dfrl dfr\_circuit\_p(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 [2:0] j,output wire [0:15] d\_o);  
  
mux8 mux8\_a({i0[0],i1[0],i2[0],i3[0],i4[0],i5[0],i6[0],i7[0]},j[0],j[1],j[2],d\_o[0]);  
mux8 mux8\_b({i0[1],i1[1],i2[1],i3[1],i4[1],i5[1],i6[1],i7[1]},j[0],j[1],j[2],d\_o[1]);  
mux8 mux8\_c({i0[2],i1[2],i2[2],i3[2],i4[2],i5[2],i6[2],i7[2]},j[0],j[1],j[2],d\_o[2]);  
mux8 mux8\_d({i0[3],i1[3],i2[3],i3[3],i4[3],i5[3],i6[3],i7[3]},j[0],j[1],j[2],d\_o[3]);  
mux8 mux8\_e({i0[4],i1[4],i2[4],i3[4],i4[4],i5[4],i6[4],i7[4]},j[0],j[1],j[2],d\_o[4]);  
mux8 mux8\_f({i0[5],i1[5],i2[5],i3[5],i4[5],i5[5],i6[5],i7[5]},j[0],j[1],j[2],d\_o[5]);  
mux8 mux8\_g({i0[6],i1[6],i2[6],i3[6],i4[6],i5[6],i6[6],i7[6]},j[0],j[1],j[2],d\_o[6]);  
mux8 mux8\_h({i0[7],i1[7],i2[7],i3[7],i4[7],i5[7],i6[7],i7[7]},j[0],j[1],j[2],d\_o[7]);  
mux8 mux8\_i({i0[8],i1[8],i2[8],i3[8],i4[8],i5[8],i6[8],i7[8]},j[0],j[1],j[2],d\_o[8]);  
mux8 mux8\_j({i0[9],i1[9],i2[9],i3[9],i4[9],i5[9],i6[9],i7[9]},j[0],j[1],j[2],d\_o[9]);  
mux8 mux8\_k({i0[10],i1[10],i2[10],i3[10],i4[10],i5[10],i6[10],i7[10]},j[0],j[1],j[2],d\_o[10]);  
mux8 mux8\_l({i0[11],i1[11],i2[11],i3[11],i4[11],i5[11],i6[11],i7[11]},j[0],j[1],j[2],d\_o[11]);  
mux8 mux8\_m({i0[12],i1[12],i2[12],i3[12],i4[12],i5[12],i6[12],i7[12]},j[0],j[1],j[2],d\_o[12]);  
mux8 mux8\_n({i0[13],i1[13],i2[13],i3[13],i4[13],i5[13],i6[13],i7[13]},j[0],j[1],j[2],d\_o[13]);  
mux8 mux8\_o({i0[14],i1[14],i2[14],i3[14],i4[14],i5[14],i6[14],i7[14]},j[0],j[1],j[2],d\_o[14]);  
mux8 mux8\_p({i0[15],i1[15],i2[15],i3[15],i4[15],i5[15],i6[15],i7[15]},j[0],j[1],j[2],d\_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);  
  
// Declare wires here  
wire [0:7] loadWires;  
wire [0:15] d\_o0,d\_o1,d\_o2,d\_o3,d\_o4,d\_o5,d\_o6,d\_o7;  
  
// Instantiate modules here  
     
reg\_unit reg\_circuit\_a(clk,reset,loadWires[0],d\_in,d\_o0);  
reg\_unit reg\_circuit\_b(clk,reset,loadWires[1],d\_in,d\_o1);  
reg\_unit reg\_circuit\_c(clk,reset,loadWires[2],d\_in,d\_o2);  
reg\_unit reg\_circuit\_d(clk,reset,loadWires[3],d\_in,d\_o3);  
reg\_unit reg\_circuit\_e(clk,reset,loadWires[4],d\_in,d\_o4);  
reg\_unit reg\_circuit\_f(clk,reset,loadWires[5],d\_in,d\_o5);  
reg\_unit reg\_circuit\_g(clk,reset,loadWires[6],d\_in,d\_o6);  
reg\_unit reg\_circuit\_h(clk,reset,loadWires[7],d\_in,d\_o7);  
  
demux8 d(wr,wr\_addr[2],wr\_addr[1],wr\_addr[0],loadWires);  
mux8\_16 mux8\_16\_a(d\_o0,d\_o1,d\_o2,d\_o3,d\_o4,d\_o5,d\_o6,d\_o7,rd\_addr\_a,d\_out\_a);  
mux8\_16 mux8\_16\_b(d\_o0,d\_o1,d\_o2,d\_o3,d\_o4,d\_o5,d\_o6,d\_o7,rd\_addr\_b,d\_out\_b);  
  
endmodule

**Output waveform**

****