Digital Design and Computer Organization Laboratory

UE19CS206

3rd Semester, Academic Year 2020-21

Date: 26/09/2020

Name :	SRN :	Section :
SUMUKH RAJU BHAT	PES1UG19CS519	H

Experiment Number: 5 Week # : 6

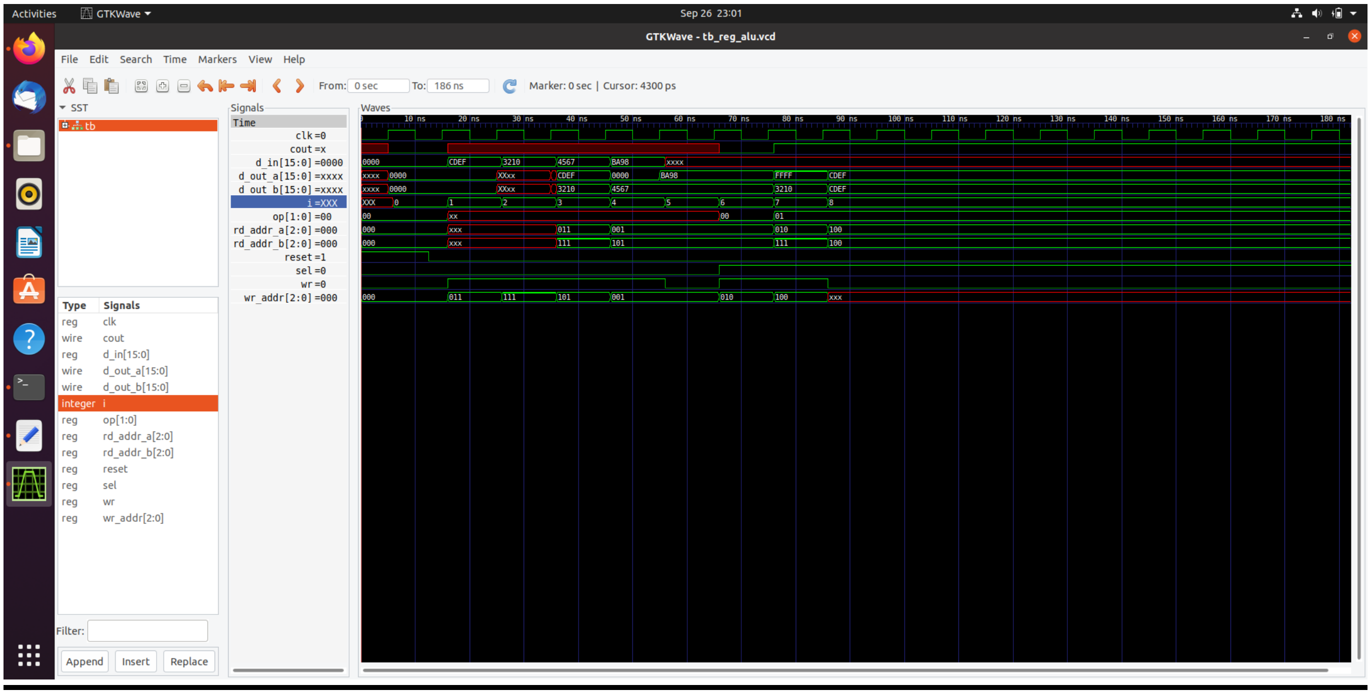
Title of the Program: Register ALU

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_unit_0(clk,reset,load,in[0],out[0]);
    dfrl dfr_unit_1(clk,reset,load,in[1],out[1]);
    dfrl dfr_unit_2(clk,reset,load,in[2],out[2]);
    dfrl dfr_unit_3(clk,reset,load,in[3],out[3]);
    dfrl dfr_unit_4(clk,reset,load,in[4],out[4]);
    dfrl dfr_unit_5(clk,reset,load,in[5],out[5]);
    dfrl dfr_unit_6(clk,reset,load,in[6],out[6]);
    dfrl dfr_unit_7(clk,reset,load,in[7],out[7]);
    dfrl dfr_unit_8(clk,reset,load,in[8],out[8]);
    dfrl dfr_unit_9(clk,reset,load,in[9],out[9]);
    dfrl dfr_unit_10(clk,reset,load,in[10],out[10]);
    dfrl dfr_unit_11(clk,reset,load,in[11],out[11]);
    dfrl dfr_unit_12(clk,reset,load,in[12],out[12]);
    dfrl dfr_unit_13(clk,reset,load,in[13],out[13]);
    dfrl dfr_unit_14(clk,reset,load,in[14],out[14]);
    dfrl dfr_unit_15(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 mux_8_unit_0({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 mux_8_unit_1({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 mux_8_unit_3({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 mux_8_unit_4({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 mux_8_unit_5({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 mux_8_unit_6({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 mux_8_unit_7({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 mux_8_unit_8({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 mux_8_unit_9({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 mux_8_unit_10({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 mux_8_unit_11({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 mux_8_unit_12({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 mux_8_unit_13({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 mux_8_unit_14({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 mux_8_unit_15({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 mux_8_unit_16({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_o_0,d_o_1,d_o_2,d_o_3,d_o_4,d_o_5,d_o_6,d_o_7;

// Instantiate modules here

reg_unit reg_unit_0(clk,reset,loadWires[0],d_in,d_o_0);
reg_unit reg_unit_1(clk,reset,loadWires[1],d_in,d_o_1);
reg_unit reg_unit_2(clk,reset,loadWires[2],d_in,d_o_2);
reg_unit reg_unit_3(clk,reset,loadWires[3],d_in,d_o_3);
reg_unit reg_unit_4(clk,reset,loadWires[4],d_in,d_o_4);
reg_unit reg_unit_5(clk,reset,loadWires[5],d_in,d_o_5);
reg_unit reg_unit_6(clk,reset,loadWires[6],d_in,d_o_6);
reg_unit reg_unit_7(clk,reset,loadWires[7],d_in,d_o_7);
demux8 d(wr,wr_addr[2],wr_addr[1],wr_addr[0],loadWires);
```

```
mux8_16
mux8_16_0(d_o_0,d_o_1,d_o_2,d_o_3,d_o_4,d_o_5,d_o_6,d_o_7,rd_addr_a,d_out_a);
mux8_16
mux8_16_1(d_o_0,d_o_1,d_o_2,d_o_3,d_o_4,d_o_5,d_o_6,d_o_7,rd_addr_b,d_out_b);

endmodule

module mux2_16(input wire [0:15] i0,i1,input wire sel,output wire [0:15] o_16);

mux2 mux_unit_00(i0[0],i1[0],sel,o_16[0]);
mux2 mux_unit_01(i0[1],i1[1],sel,o_16[1]);
mux2 mux_unit_02(i0[2],i1[2],sel,o_16[2]);
mux2 mux_unit_03(i0[3],i1[3],sel,o_16[3]);
mux2 mux_unit_04(i0[4],i1[4],sel,o_16[4]);
mux2 mux_unit_05(i0[5],i1[5],sel,o_16[5]);
mux2 mux_unit_06(i0[6],i1[6],sel,o_16[6]);
mux2 mux_unit_07(i0[7],i1[7],sel,o_16[7]);
mux2 mux_unit_08(i0[8],i1[8],sel,o_16[8]);
mux2 mux_unit_09(i0[9],i1[9],sel,o_16[9]);
mux2 mux_unit_10(i0[10],i1[10],sel,o_16[10]);
mux2 mux_unit_11(i0[11],i1[11],sel,o_16[11]);
mux2 mux_unit_12(i0[12],i1[12],sel,o_16[12]);
mux2 mux_unit_13(i0[13],i1[13],sel,o_16[13]);
mux2 mux_unit_14(i0[14],i1[14],sel,o_16[14]);
mux2 mux_unit_15(i0[15],i1[15],sel,o_16[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);

// Declare wires here
wire [15:0] outputWire,mux_out;

// Instantiate modules here

mux2_16 mux_16_unit(d_in,outputWire,sel,mux_out);
reg_file reg_file_0(clk,reset,wr,rd_addr_a,rd_addr_b,wr_addr,mux_out,d_out_a,d_out_b);
alu alu_0(op,d_out_a,d_out_b,outputWire,cout);


endmodule
```

Output waveform



From: 0 sec To: 186 ns

Marker: 0 sec | Cursor: 4300 ps

```
Time
clk =0
cout =x
d_in[15:0] =0000
d_out_a[15:0] =xxxx
d_out_b[15:0] =xxxx
i =XXX
op[1:0] =00
rd_addr_a[2:0] =000
rd_addr_b[2:0] =000
reset =1
sel =0
wr =0
wr_addr[2:0] =000
```

Type	Signals
reg	clk
wire	cout
reg	d_in[15:0]
wire	d_out_a[15:0]
wire	d_out_b[15:0]
integer	i
reg	op[1:0]
reg	rd_addr_a[2:0]
reg	rd_addr_b[2:0]
reg	reset
reg	sel
reg	wr
reg	wr_addr[2:0]