**Digital Design and Computer Organization Laboratory**

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

Date: 25-10-2021

|  |  |  |
| --- | --- | --- |
| Name : M H SOHAN | SRN : PES1UG20CS235 | Section : 3 D |

Experiment Number: 6 Week # : 6

**Title of the Program:**

**Code:**

**pc.v:**

module fulladd(input wire a, b, cin, output wire sum, cout);

wire t0,t1,t2,t3;

xor2 xor\_0(a,b,t0);

xor2 xor\_1(t0,cin,sum);

and2 a1(a,b,t1);

and2 a2(b,cin,t2);

and2 a3(cin,a,t3);

or3 o5(t1,t2,t3,cout);

endmodule

module addsub(input wire as, i0, i1, cin, output wire sumdiff, cout);

wire t;

xor2 x0(i1,as,t);

fulladd f(i0,t,cin,sumdiff,cout);

endmodule

module pc\_slice0(input wire clk,reset,offset,inc,sub,load, output wire pc,cout);

wire t1,t2;

or2 o0(offset,inc,t1);

addsub o2(sub,pc,t1,sub,t2,cout);

dfrl o3(clk,reset,load,t2,pc);

endmodule

module pc\_slice1(input wire clk,reset,offset,inc,sub,load,cin, output wire pc,cout);

wire t1,t2,t3;

invert i(inc,t1);

and2 o0(offset,t1,t2);

addsub o2(sub,pc,t2,cin,t3,cout);

dfrl o3(clk,reset,load,t3,pc);

endmodule

module pc(input wire clk, reset, inc, add, sub, input wire [15:0] offset, output wire [15:0] pc);

wire[15:0]cout;

wire load;

or3 o(inc,add,sub,load);

pc\_slice0 s0(clk,reset,offset[0],inc,sub,load,pc[0],cout[0]);

pc\_slice1 s1(clk,reset,offset[1],inc,sub,load,cout[0],pc[1],cout[1]);

pc\_slice1 s2(clk,reset,offset[2],inc,sub,load,cout[1],pc[2],cout[2]);

pc\_slice1 s3(clk,reset,offset[3],inc,sub,load,cout[2],pc[3],cout[3]);

pc\_slice1 s4(clk,reset,offset[4],inc,sub,load,cout[3],pc[4],cout[4]);

pc\_slice1 s5(clk,reset,offset[5],inc,sub,load,cout[4],pc[5],cout[5]);

pc\_slice1 s6(clk,reset,offset[6],inc,sub,load,cout[5],pc[6],cout[6]);

pc\_slice1 s7(clk,reset,offset[7],inc,sub,load,cout[6],pc[7],cout[7]);

pc\_slice1 s8(clk,reset,offset[8],inc,sub,load,cout[7],pc[8],cout[8]);

pc\_slice1 s9(clk,reset,offset[9],inc,sub,load,cout[8],pc[9],cout[9]);

pc\_slice1 s10(clk,reset,offset[10],inc,sub,load,cout[9],pc[10],cout[10]);

pc\_slice1 s11(clk,reset,offset[11],inc,sub,load,cout[10],pc[11],cout[11]);

pc\_slice1 s12(clk,reset,offset[12],inc,sub,load,cout[11],pc[12],cout[12]);

pc\_slice1 s13(clk,reset,offset[13],inc,sub,load,cout[12],pc[13],cout[13]);

pc\_slice1 s14(clk,reset,offset[14],inc,sub,load,cout[13],pc[14],cout[14]);

pc\_slice1 s15(clk,reset,offset[15],inc,sub,load,cout[14],pc[15],cout[15]);

endmodule

**tb\_pc.v:**

`timescale 1 ns / 100 ps

`define TESTVECS 5

module tb;

reg clk, reset, inc, add, sub;

reg [15:0] offset;

wire [15:0] pc;

reg [18:0] test\_vecs [0:(`TESTVECS-1)];

integer i;

initial begin $dumpfile("tb\_pc.vcd"); $dumpvars(0,tb); end

initial begin reset = 1'b1; #12.5 reset = 1'b0; end

initial clk = 1'b0; always #5 clk =~ clk;

initial begin

test\_vecs[0][18] = 1'b1; test\_vecs[0][17] = 1'b0; test\_vecs[0][16] = 1'b0;

test\_vecs[0][15:0] = 15'hxx;

test\_vecs[1][18] = 1'b0; test\_vecs[1][17] = 1'b1; test\_vecs[1][16] = 1'b0;

test\_vecs[1][15:0] = 15'ha5;

test\_vecs[2][18] = 1'b0; test\_vecs[2][17] = 1'b0; test\_vecs[2][16] = 1'b0;

test\_vecs[2][15:0] = 15'hxx;

test\_vecs[3][18] = 1'b1; test\_vecs[3][17] = 1'b0; test\_vecs[3][16] = 1'b0;

test\_vecs[3][15:0] = 15'hxx;

test\_vecs[4][18] = 1'b0; test\_vecs[4][17] = 1'b0; test\_vecs[4][16] = 1'b1;

test\_vecs[4][15:0] = 15'h14;

end

initial {inc, add, sub, offset} = 0;

pc pc\_0 (clk, reset, inc, add, sub, offset, pc);

initial begin

#6 for(i=0;i<`TESTVECS;i=i+1)

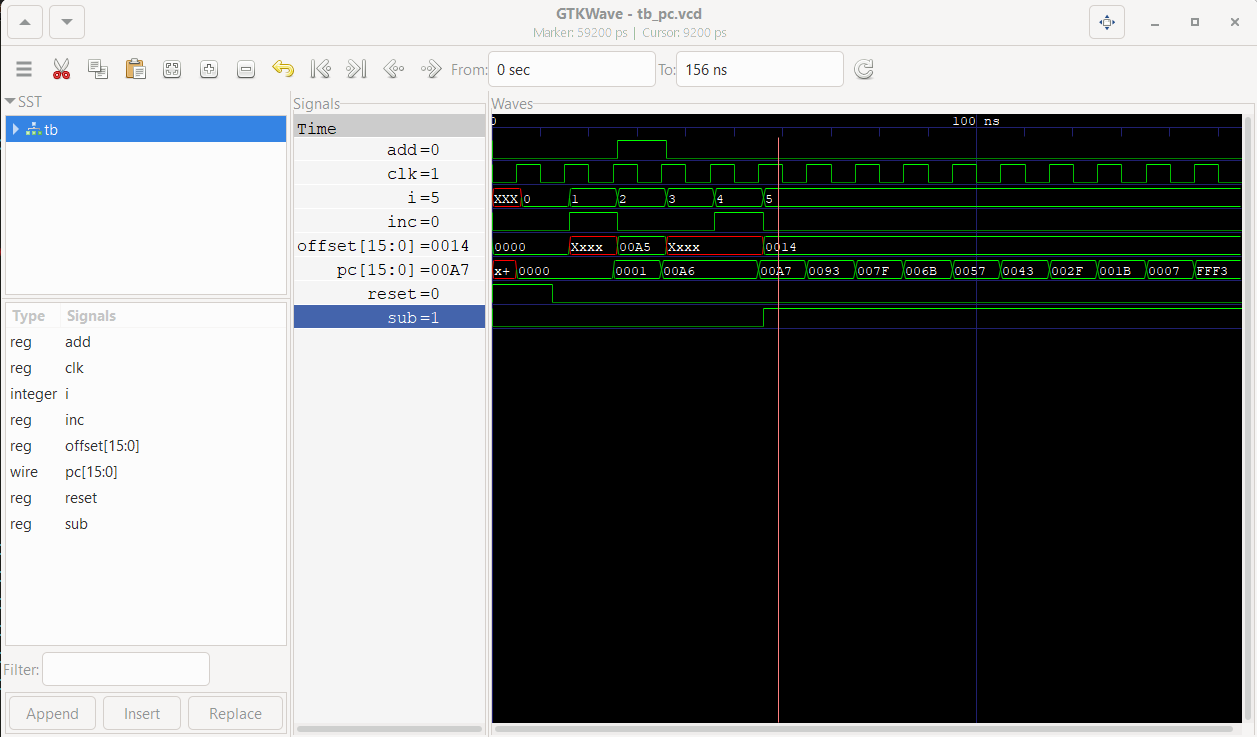
begin #10 {inc, add, sub, offset}=test\_vecs[i]; end

#100 $finish;

end

endmodule

**Output waveform**

****