Name: Adithya M SRN:PES1UG20CS621 SEC: K

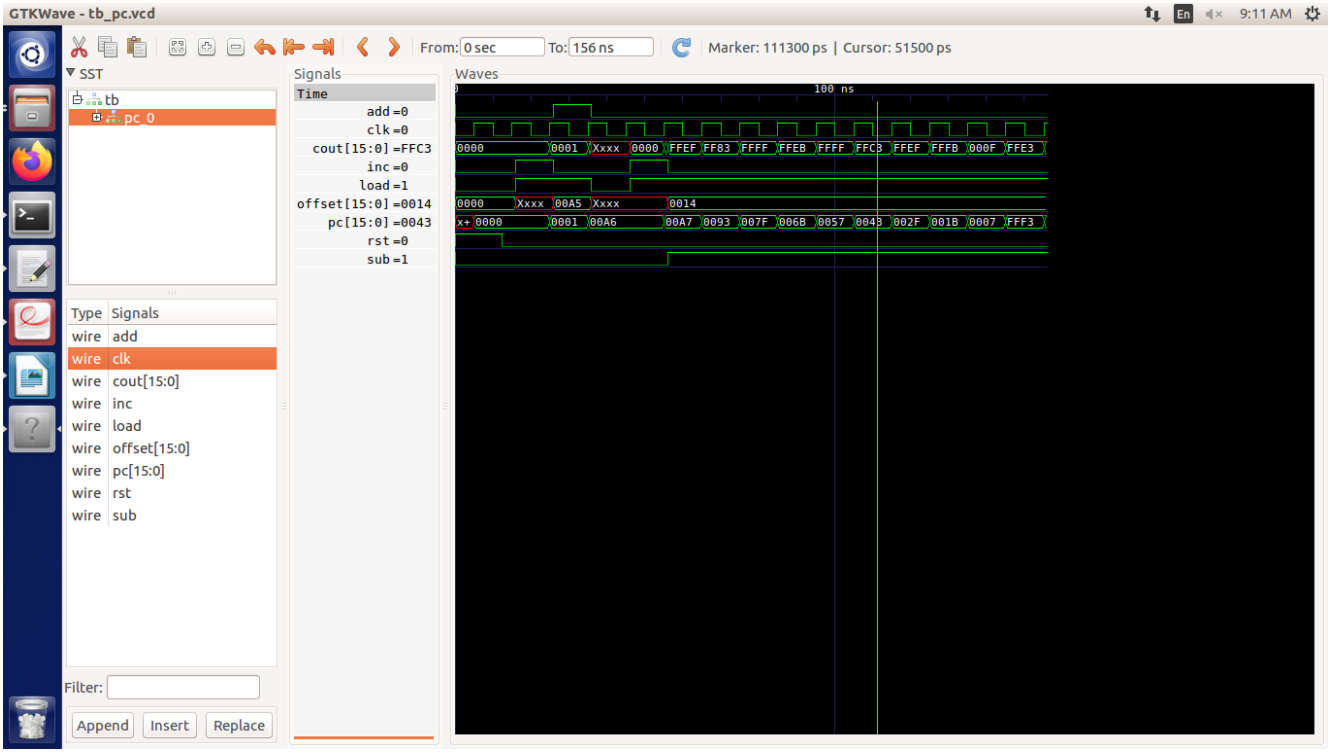
pc.v

```
// Write code for modules you need here
module fadd(input wire a, b, cin, output wire sum, cout);
wire t0,t1,t2,t3;
xor2 x0(a,b,t0);
xor2 x1(t0,cin,sum);
and2 x2(a,b,t1);
and2 x3(b,cin,t2);
and2 x4(cin,a,t3);
or3 x5(t1,t2,t3,cout);
endmodule

module addsub(input wire mode,i0,i1,cin,output wire sumdiff,cout);
wire t;
xor2 x0(i1,mode,t);
fadd f(i0,t,cin,sumdiff,cout);
endmodule

module pc_slice0 (input wire clk,rst,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,rst,load,t2,pc);
endmodule

module pc_slice1 (input wire clk,rst,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,rst,load,t3,pc);
endmodule

module pc (input wire clk, rst, 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,rst,offset[0],inc,sub,load,pc[0],cout[0]);
pc_slice1 s1(clk,rst,offset[1],inc,sub,load,cout[0],pc[1],cout[1]);
pc_slice1 s2(clk,rst,offset[2],inc,sub,load,cout[1],pc[2],cout[2]);
pc_slice1 s3(clk,rst,offset[3],inc,sub,load,cout[2],pc[3],cout[3]);
pc_slice1 s4(clk,rst,offset[4],inc,sub,load,cout[3],pc[4],cout[4]);
pc_slice1 s5(clk,rst,offset[5],inc,sub,load,cout[4],pc[5],cout[5]);
pc_slice1 s6(clk,rst,offset[6],inc,sub,load,cout[5],pc[6],cout[6]);
pc_slice1 s7(clk,rst,offset[7],inc,sub,load,cout[6],pc[7],cout[7]);
pc_slice1 s8(clk,rst,offset[8],inc,sub,load,cout[7],pc[8],cout[8]);
pc_slice1 s9(clk,rst,offset[9],inc,sub,load,cout[8],pc[9],cout[9]);
pc_slice1 s10(clk,rst,offset[10],inc,sub,load,cout[9],pc[10],cout[10]);
pc_slice1 s11(clk,rst,offset[11],inc,sub,load,cout[10],pc[11],cout[11]);
pc_slice1 s12(clk,rst,offset[12],inc,sub,load,cout[11],pc[12],cout[12]);
pc_slice1 s13(clk,rst,offset[13],inc,sub,load,cout[12],pc[13],cout[13]);
pc_slice1 s14(clk,rst,offset[14],inc,sub,load,cout[13],pc[14],cout[14]);
pc_slice1 s15(clk,rst,offset[15],inc,sub,load,cout[14],pc[15],cout[15]);

endmodule
// Declare wires here

// Instantiate modules here
```

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


GTKWave - tb_pc.vcd
9:11 AM
From: 0 sec
To: 156 ns
Marker: 111300 ps | Cursor: 51500 ps
SST
tb
pc_0
Type Signals
wire add
wire clk
wire cout[15:0]
wire inc
wire load
wire offset[15:0]
wire pc[15:0]
wire rst
wire sub
Filter:
Append
Insert
Replace
Signals
Time
add =0
clk =0
cout[15:0] =FFC3
inc =0
load =1
offset[15:0] =0014
pc[15:0] =0043
rst =0
sub =1
Waves
100 ns