Digital Design and Computer Organization Laboratory

UE23CS251A

3rd Semester, Academic Year 2024-25

Date: 30/10/2024

Name : ABHISHEK P	SRN : PES2UG23AM002	Section : A

Experiment Number: 8 Week # : 8

Title of the Program:

16 Bit Program Counter

Aim of the Program:

To Design and Implementation of a 16 bit Program Counter

Source code

```
module fa(input wire i0, i1, cin, output wire sum, cout);
  wire t0, t1, t2;
  xor3 xor3_0(i0, i1, cin, sum);
  and2 and2_0(i0, i1, t0);
  and2 and2_1(i0, cin, t1);
  and2 and2_2(i1, cin, t2);
  or3 or3_0(t0, t1, t2, cout);
endmodule

module addsub(input wire addsub, i0, i1, cin, output wire sumdiff, cout);
  wire t;
  fa fa_0(i0, t, cin, sumdiff, cout);
  xor2 xor2_0(i1, addsub, t);
```

```
endmodule

module pc_slice(input wire clk,reset,cin,load,inc,sub,offset,output wire cout,pc);
wire in,inc_;
invert invert_0(inc,inc_);
and2 and2_0(offset,inc_,t);
addsub addsub_0(sub,pc,t,cin,in,cout);
dfrl dfrl_0(clk,reset,load,in,pc);
endmodule

module pc_slice0(input wire clk,reset,cin,load,inc,sub,offset,output wire cout,pc);
wire in,t;
or2 or2_0(offset,inc,t);
addsub addsub_0(sub,pc,t,cin,in,cout);
dfrl dfrl_0(clk,reset,load,in,pc);
endmodule

module pc(input wire clk,reset,inc,add,sub,input wire [15:0]offset,output wire[15:0]pc);
input wire load;
input wire[15:0]c;
or3 or3_0(inc,sub,add,load);
pc_slice0 pc_slice_0(clk,reset,sub,load,inc,sub,offset[0],c[0],pc[0]);
pc_slice pc_slice_1(clk,reset,c[0],load,inc,sub,offset[1],c[1],pc[1]);
pc_slice pc_slice_2(clk,reset,c[1],load,inc,sub,offset[2],c[2],pc[2]);
pc_slice pc_slice_3(clk,reset,c[2],load,inc,sub,offset[3],c[3],pc[3]);
pc_slice pc_slice_4(clk,reset,c[3],load,inc,sub,offset[4],c[4],pc[4]);
pc_slice pc_slice_5(clk,reset,c[4],load,inc,sub,offset[5],c[5],pc[5]);
pc_slice pc_slice_6(clk,reset,c[5],load,inc,sub,offset[6],c[6],pc[6]);
pc_slice pc_slice_7(clk,reset,c[6],load,inc,sub,offset[7],c[7],pc[7]);
pc_slice pc_slice_8(clk,reset,c[7],load,inc,sub,offset[8],c[8],pc[8]);
pc_slice pc_slice_9(clk,reset,c[8],load,inc,sub,offset[9],c[9],pc[9]);
pc_slice pc_slice_10(clk,reset,c[9],load,inc,sub,offset[10],c[10],pc[10]);
pc_slice pc_slice_11(clk,reset,c[10],load,inc,sub,offset[11],c[11],pc[11]);
pc_slice pc_slice_12(clk,reset,c[11],load,inc,sub,offset[12],c[12],pc[12]);
```

```
pc_slice pc_slice_13(clk,reset,c[12],load,inc,sub,offset[13],c[13],pc[13]);
pc_slice pc_slice_14(clk,reset,c[13],load,inc,sub,offset[14],c[14],pc[14]);
pc_slice pc_slice_15(clk,reset,c[14],load,inc,sub,offset[15],c[15],pc[15]);
endmodule
```

Testbench code

```
`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

always@(reset or inc or add or sub )

$monitor("At time = %t, Reset= %b,inc=%b, add=%b,sub = %b,pc =%h ", $time,reset,inc,add,sub,pc);
endmodule
```

vvp output

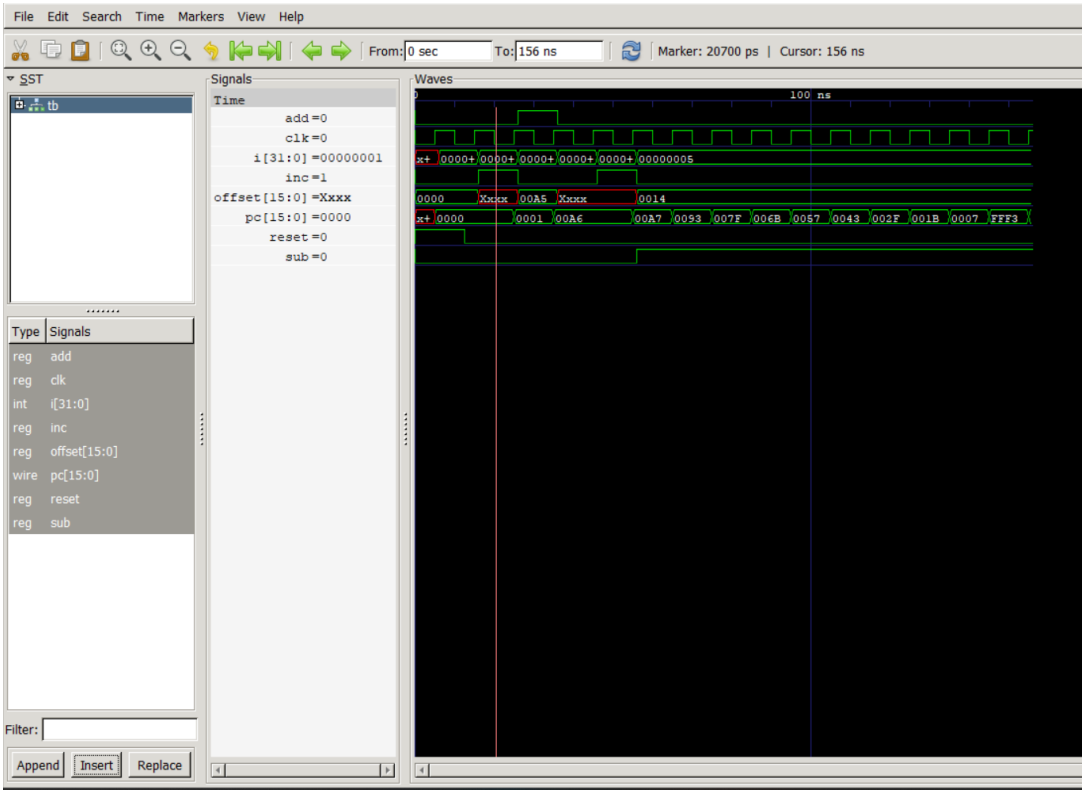
```
VCD info: dumpfile tb_pc.vcd opened for output.
At time =                    0, Reset= 1,inc=0, add=0,sub = 0,pc =xxxx
At time =                   50, Reset= 1,inc=0, add=0,sub = 0,pc =0000
At time =                  130, Reset= 0,inc=0, add=0,sub = 0,pc =0000
At time =                  160, Reset= 0,inc=1, add=0,sub = 0,pc =0000
At time =                  250, Reset= 0,inc=1, add=0,sub = 0,pc =0001
At time =                  260, Reset= 0,inc=0, add=1,sub = 0,pc =0001
At time =                  350, Reset= 0,inc=0, add=1,sub = 0,pc =00a6
At time =                  360, Reset= 0,inc=0, add=0,sub = 0,pc =00a6
At time =                  460, Reset= 0,inc=1, add=0,sub = 0,pc =00a6
At time =                  550, Reset= 0,inc=1, add=0,sub = 0,pc =00a7
At time =                  560, Reset= 0,inc=0, add=0,sub = 1,pc =00a7
At time =                  650, Reset= 0,inc=0, add=0,sub = 1,pc =0093
At time =                  750, Reset= 0,inc=0, add=0,sub = 1,pc =007f
At time =                  850, Reset= 0,inc=0, add=0,sub = 1,pc =006b
At time =                  950, Reset= 0,inc=0, add=0,sub = 1,pc =0057
At time =                 1050, Reset= 0,inc=0, add=0,sub = 1,pc =0043
At time =                 1150, Reset= 0,inc=0, add=0,sub = 1,pc =002f
At time =                 1250, Reset= 0,inc=0, add=0,sub = 1,pc =001b
At time =                 1350, Reset= 0,inc=0, add=0,sub = 1,pc =0007
At time =                 1450, Reset= 0,inc=0, add=0,sub = 1,pc =fff3
At time =                 1550, Reset= 0,inc=0, add=0,sub = 1,pc =ffdf
```

TABLE

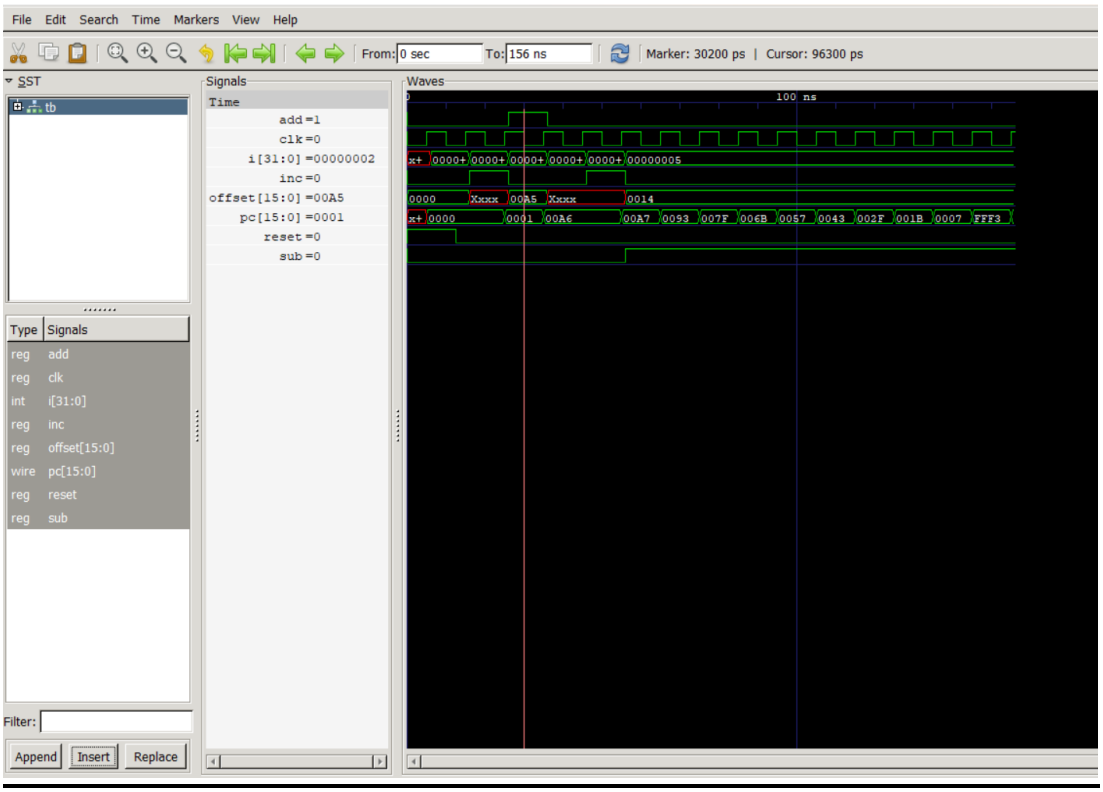
	inc	add	sub	offset [15:0]	output
	Bit 18	Bit 17	Bit 16	Bit 15 to Bit0	pc[15:0]
CASE 1	1	0	0	XXXX	0001
CASE 2	0	1	0	00A5	00A6
CASE 3	0	0	0	XXXX	00A6
CASE 4	1	0	0	XXXX	00A7
CASE 5	0	0	1	0014	0093

Output waveform

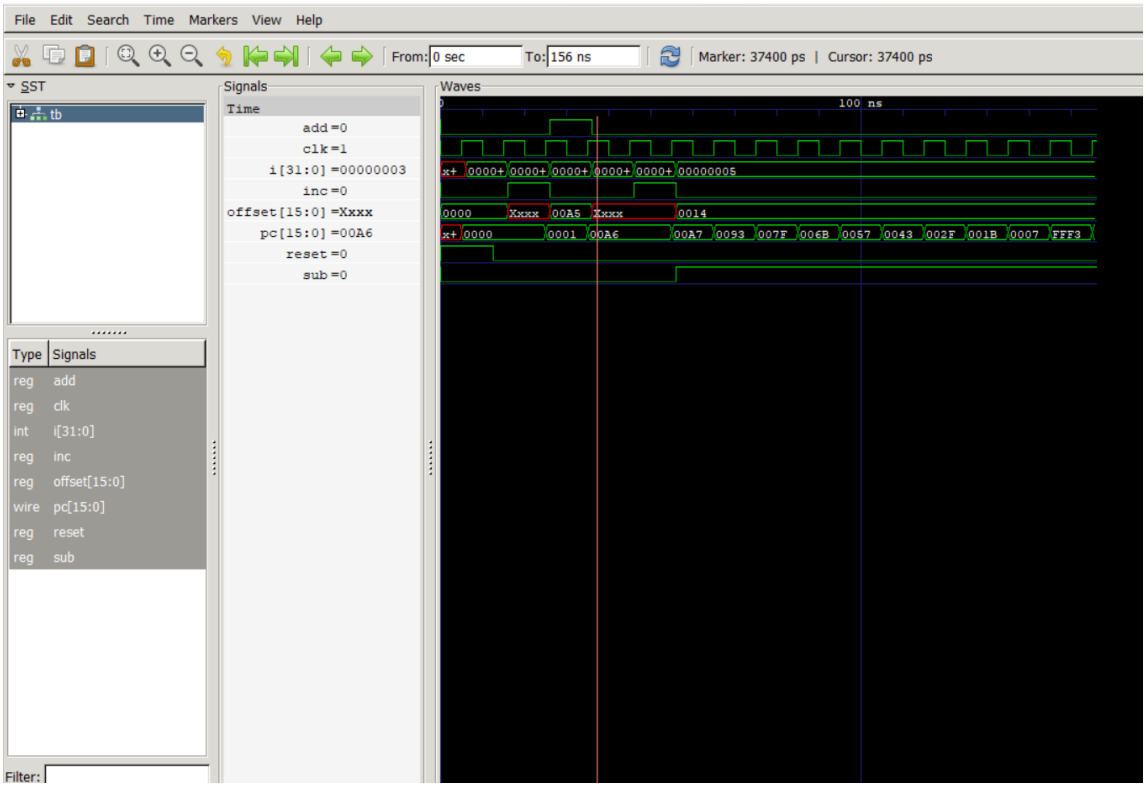
CASE1 :PC Increment Operation with no offset



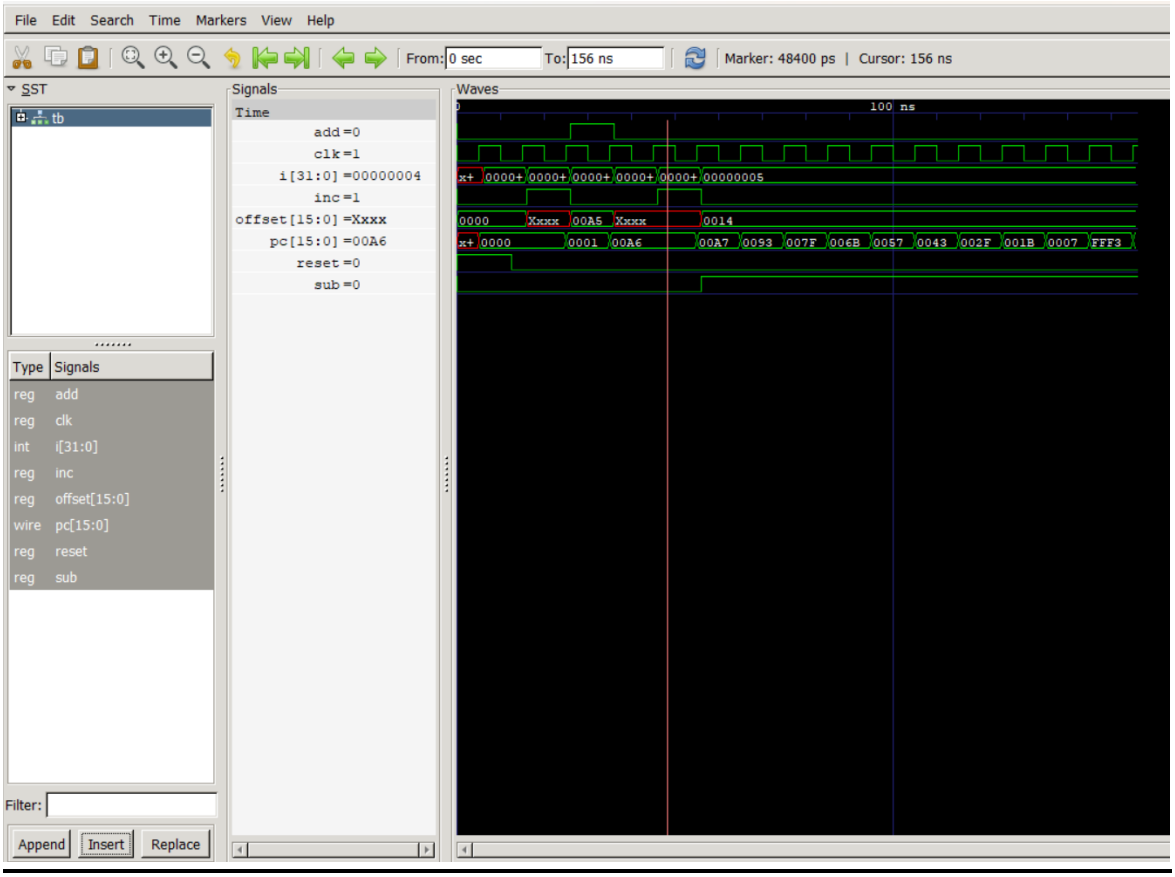
CASE 2 :Add Offset to PC



CASE 3 :No change in PC



CASE 4 :Auto increment current value of PC



CASE 5 :Subtract offset contents from PC