Digital Design and Computer Organization Laboratory

UE19CS206

3rd Semester, Academic Year 2020-21

Date: 24/10/22

Name : Sujal.S	SRN : PES2UG21CS548	Section : I

Experiment Number: 6 Week # : 6

Title of the Program:

16 Bit Program Counter

Aim of the Program:

To Design and Implementation of a 16 bit Program Counter

Code (pc.v)

```
module fa (input wire i0, i1, cin, output wire sum, cout);
  wire t0, t1, t2;
  xor3 _i0 (i0, i1, cin, sum);
  and2 _i1 (i0, i1, t0);
  and2 _i2 (i1, cin, t1);
  and2 _i3 (cin, i0, t2);
```

```
  or3 _i4 (t0, t1, t2, cout);
endmodule

module addsub (input wire addsub, i0, i1, cin, output wire sumdiff,
cout);
  wire t;
  fa _i0 (i0, t, cin, sumdiff, cout);
  xor2 _i1 (i1, addsub, t);
endmodule

module pc_slice (input wire clk, reset, cin, load, inc, sub, offset,
  output wire cout, pc);
  wire in, inc_, in_as;
  invert invert_0 (inc,inc_);
  and2 and2_0 (offset, inc_, in_as);
  addsub addsub_0 (sub, pc, in_as, 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, in_as;
  or2 or2_0 (offset, inc, in_as);
```

```
  addsub addsub_0 (sub, pc, in_as, sub, 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, add, sub, 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]);
```

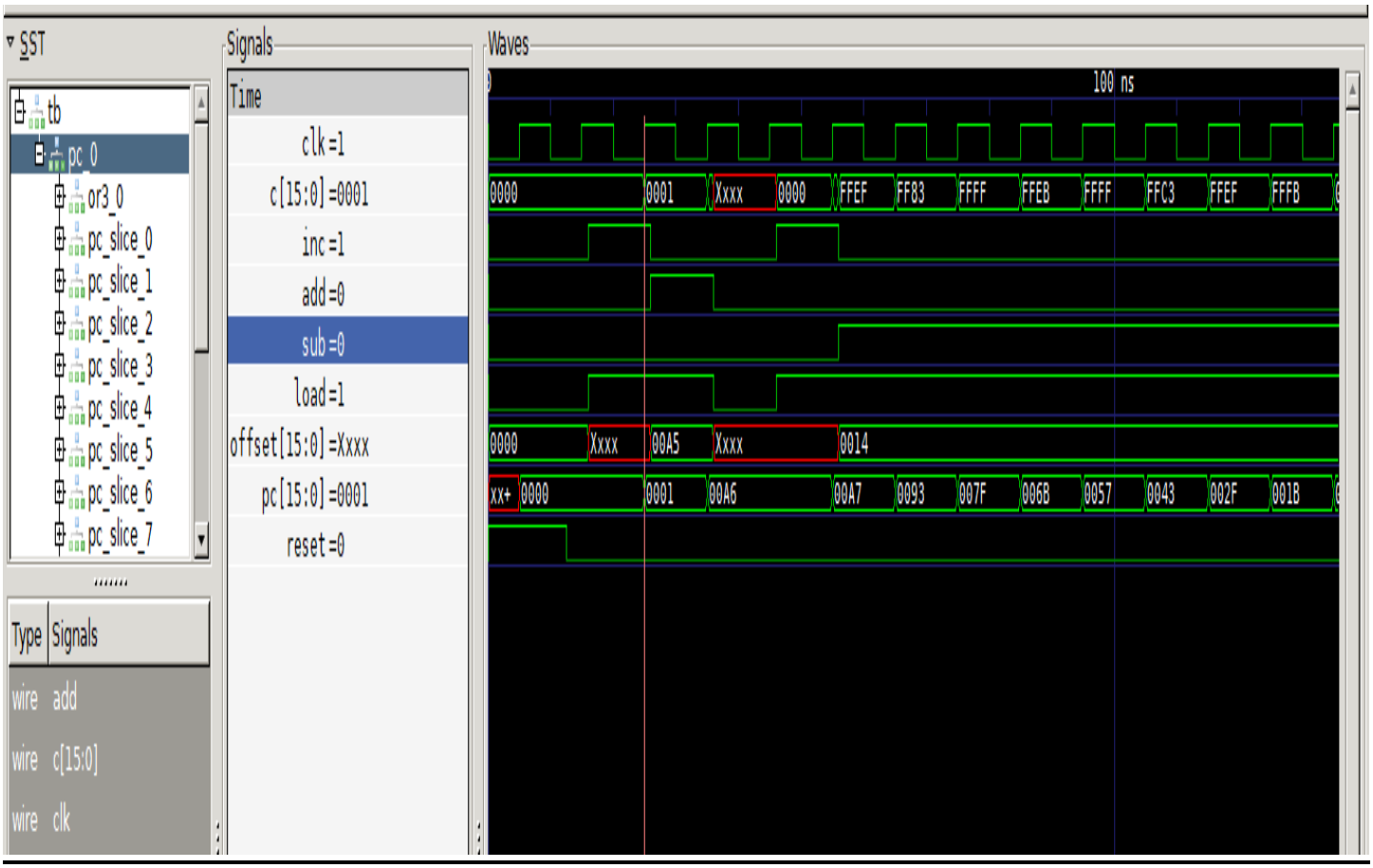
```
  pc_slice pc_slice_7 (clk, reset, c[6], load, inc, sub, offset[7], c[7],
pc[7]);

  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
```

TABLE

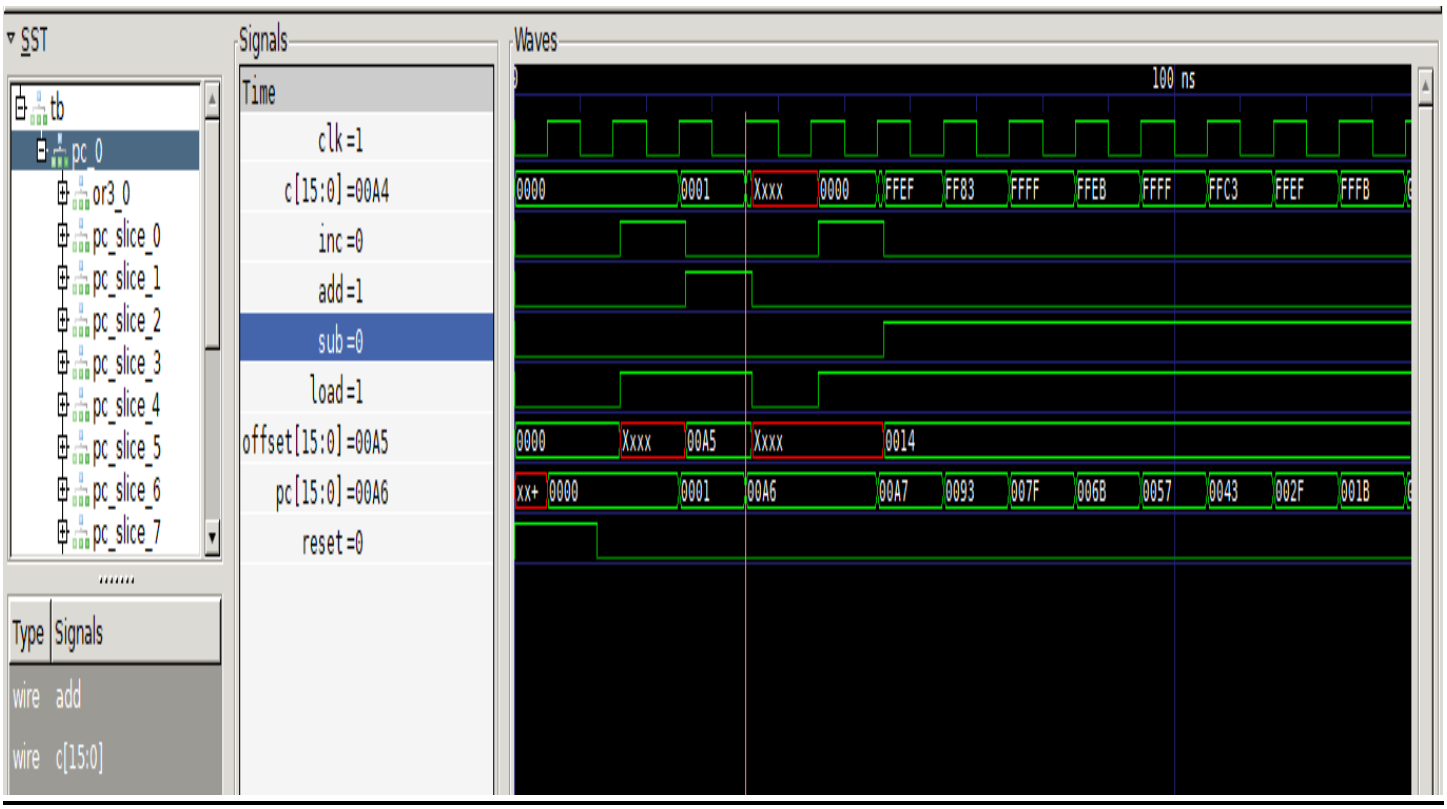
	inc	add	sub	offset [15:0]	output
	Bit 18	Bit 17	Bit 16	Bit 15 to Bit0	pc[15:0]
CASE 1	1	0	0	XXXX	0001
CASE 2	0	1	0	00A5	00A6
CASE 3	0	0	0	XXXX	00A6
CASE 4	1	0	0	XXXX	00A7
CASE 5	0	0	1	0014	pc-offset =00A7-0014 =0093

Output waveform

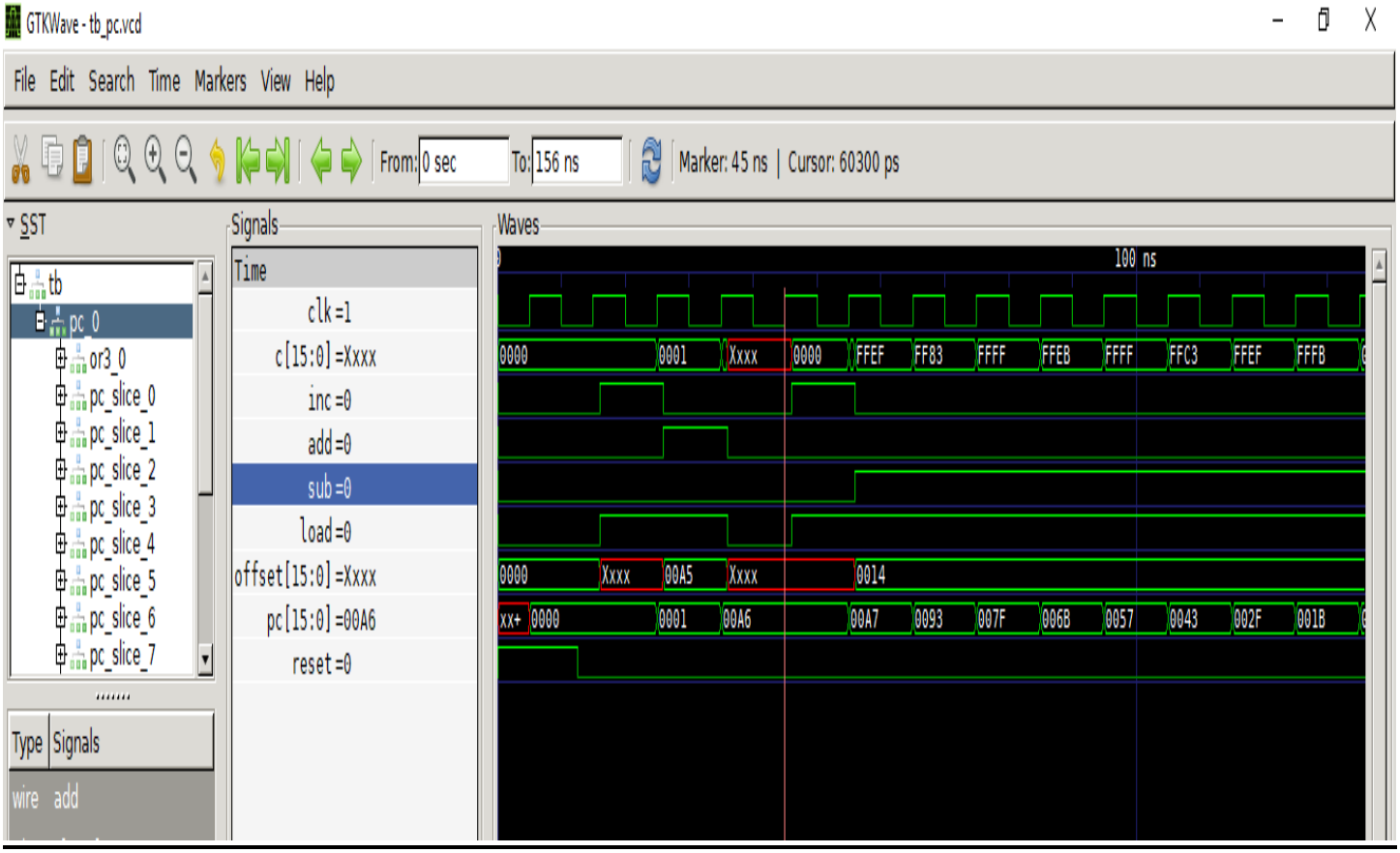
CASE1 :PC Increment Operation with no offset



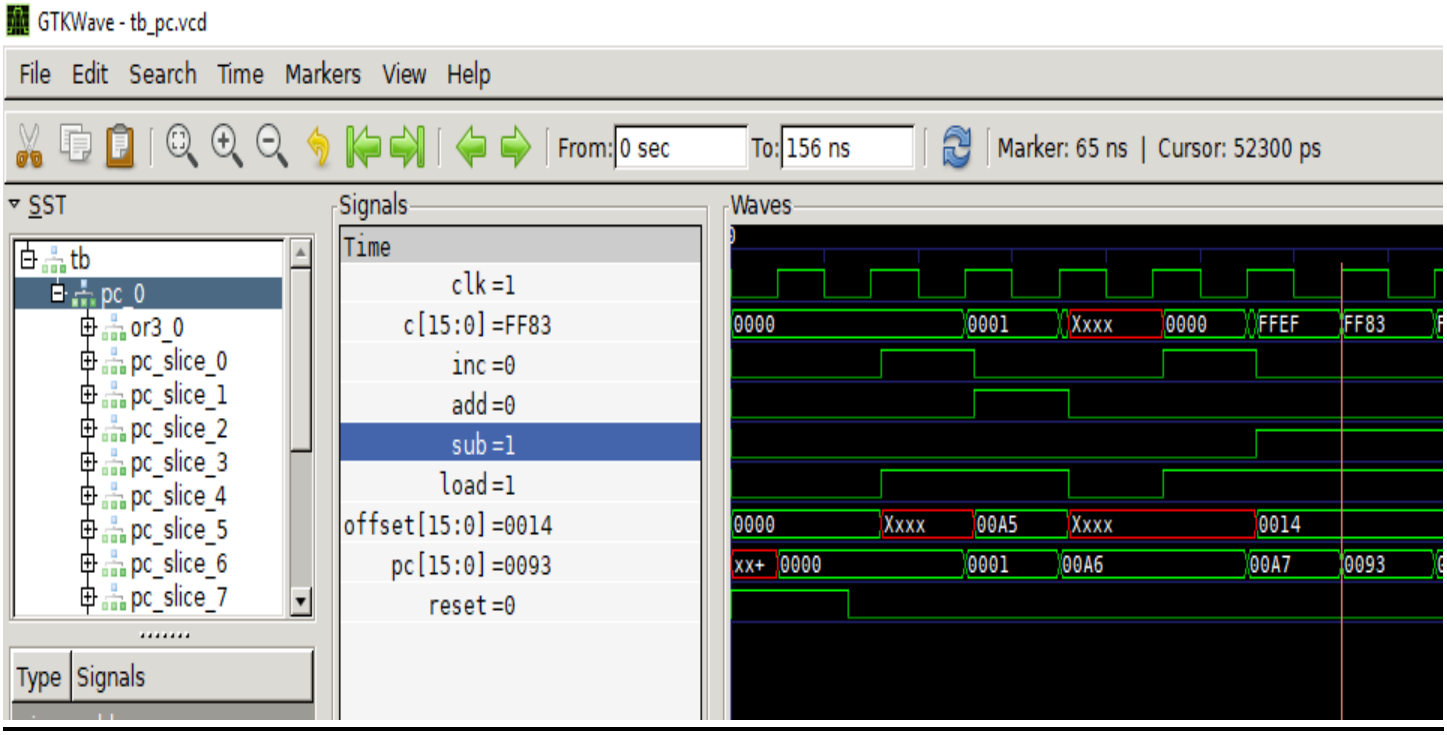
CASE 2 :Add Offset to PC



CASE 3 :No change in PC



CASE 4 :Auto increment current value of PC



CASE 5 :Subtract offset contents from PC