Digital Design and Computer Organisation Laboratory

UE23CS252A

3rd Semester, Academic Year 2024-25

Date: 02-10-2024

Name: ABHISHEK P	SRN: PES2UG23AM002	Section: A

Week#____1_____ Program Number: ____3___

TITLE:

WRITE A VERILOG PROGRAM TO MODEL A 16 BIT ALU THAT CAN PERFORM FOUR OPERATIONS-ADDITION, SUBTRACTION, AND along with OR OPERATION. ALL THESE OPERATIONS GENERATE A SIXTEEN BIT RESULT.SHOW THE VVP OUTPUT.DISPLAY THE SIMULATION WAVEFORM USING GTKWAVE. VERIFY THE OUTPUT AND WAVEFORM WITH THE TRUTH TABLE

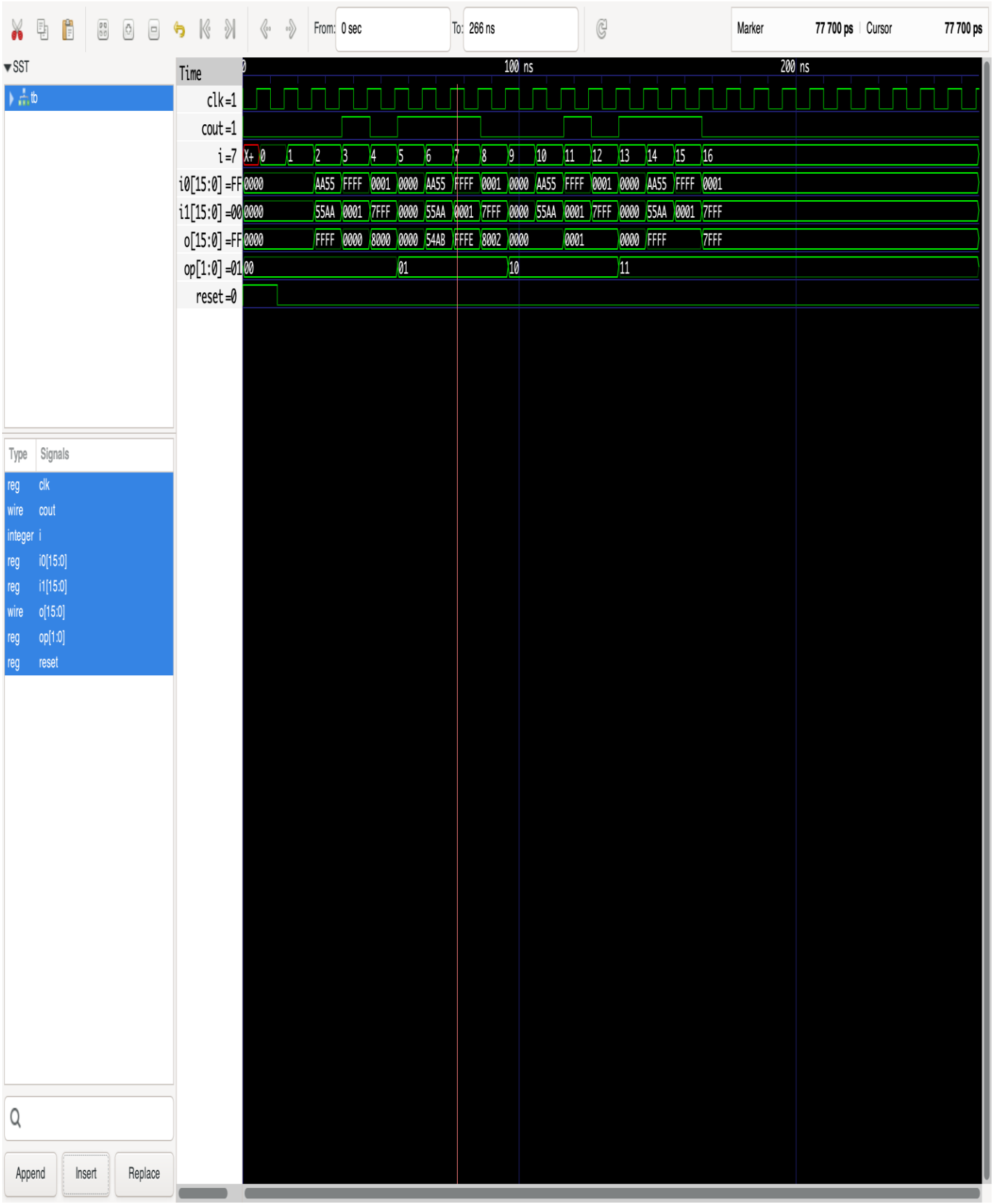
tb.v one.v

one.v

```v
module invert (input wire i, output wire o);
    assign o = !i;
endmodule

module and2 (input wire i0, i1, output wire o);
  assign o = i0 & i1;
endmodule

module or2 (input wire i0, i1, output wire o);
  assign o = i0 | i1;
endmodule

module xor2 (input wire i0, i1, output wire o);
  assign o = i0 ^ i1;
endmodule

module nand2 (input wire i0, i1, output wire o);
    wire t;
    and2 and2_0 (i0, i1, t);
    invert invert_0 (t, o);
endmodule

module nor2 (input wire i0, i1, output wire o);
    wire t;
    or2 or2_0 (i0, i1, t);
    invert invert_0 (t, o);
endmodule

module xnor2 (input wire i0, i1, output wire o);
    wire t;
    xor2 xor2_0 (i0, i1, t);
    invert invert_0 (t, o);
endmodule

module and3 (input wire i0, i1, i2, output wire o);
    wire t;
    and2 and2_0 (i0, i1, t);
    and2 and2_1 (i2, t, o);
endmodule

module or3 (input wire i0, i1, i2, output wire o);
    wire t;
    or2 or2_0 (i0, i1, t);
    or2 or2_1 (i2, t, o);
endmodule

module nor3 (input wire i0, i1, i2, output wire o);
    wire t;
    or2 or2_0 (i0, i1, t);
    nor2 nor2_0 (i2, t, o);
endmodule

module nand3 (input wire i0, i1, i2, output wire o);
    wire t;
    and2 and2_0 (i0, i1, t);
    nand2 nand2_1 (i2, t, o);
endmodule

module xor3 (input wire i0, i1, i2, output wire o);
    wire t;
    xor2 xor2_0 (i0, i1, t);
    xor2 xor2_1 (i2, t, o);
endmodule

module xnor3 (input wire i0, i1, i2, output wire o);
    wire t;
    xor2 xor2_0 (i0, i1, t);
    xnor2 xnor2_0 (i2, t, o);
endmodule

module mux2 (input wire i0, i1, j, output wire o);
```

```
  mux2 mux2_1 (i[2], i[3], j0, t1);
  mux2 mux2_2 (t0, t1, j1, o);
endmodule

module mux8 (input wire [0:7] i, input wire j2, j1, j0, output wire o);
  wire t0, t1;
  mux4 mux4_0 (i[0:3], j1, j0, t0);
  mux4 mux4_1 (i[4:7], j1, j0, t1);
  mux2 mux2_0 (t0, t1, j2, o);
endmodule

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 (i1, cin, t1);
  and2 and2_2 (cin, i0, t2);
  or3 or3_0 (t0, t1, t2, cout);
endmodule

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

module alu_slice (input wire [1:0] op, input wire i0, i1, cin, output wire o, cout);
  wire t_sumdiff, t_and, t_or, t_andor;
  addsub addsub_0 (op[0], i0, i1, cin, t_sumdiff, cout);
  and2 and2_0 (i0, i1, t_and);
  or2 or2_0 (i0, i1, t_or);
  mux2 mux2_0 (t_and, t_or, op[0], t_andor);
  mux2 mux2_1 (t_sumdiff, t_andor, op[1], o);
endmodule

module alu (input wire [1:0] op, input wire [15:0] i0, i1, output wire [15:0] o, output wire cout);
  wire [14:0] c;
  alu_slice alu_slice_0 (op, i0[0], i1[0], op[0], o[0], c[0]);
  alu_slice alu_slice_1 (op, i0[1], i1[1], c[0], o[1], c[1]);
  alu_slice alu_slice_2 (op, i0[2], i1[2], c[1], o[2], c[2]);
  alu_slice alu_slice_3 (op, i0[3], i1[3], c[2], o[3], c[3]);
  alu_slice alu_slice_4 (op, i0[4], i1[4], c[3], o[4], c[4]);
  alu_slice alu_slice_5 (op, i0[5], i1[5], c[4], o[5], c[5]);
  alu_slice alu_slice_6 (op, i0[6], i1[6], c[5], o[6], c[6]);
  alu_slice alu_slice_7 (op, i0[7], i1[7], c[6], o[7], c[7]);
  alu_slice alu_slice_8 (op, i0[8], i1[8], c[7], o[8], c[8]);
  alu_slice alu_slice_9 (op, i0[9], i1[9], c[8], o[9], c[9]);
  alu_slice alu_slice_10 (op, i0[10], i1[10], c[9], o[10], c[10]);
  alu_slice alu_slice_11 (op, i0[11], i1[11], c[10], o[11], c[11]);
  alu_slice alu_slice_12 (op, i0[12], i1[12], c[11], o[12], c[12]);
  alu_slice alu_slice_13 (op, i0[13], i1[13], c[12], o[13], c[13]);
  alu_slice alu_slice_14 (op, i0[14], i1[14], c[13], o[14], c[14]);
  alu_slice alu_slice_15 (op, i0[15], i1[15], c[14], o[15], cout);
endmodule
```

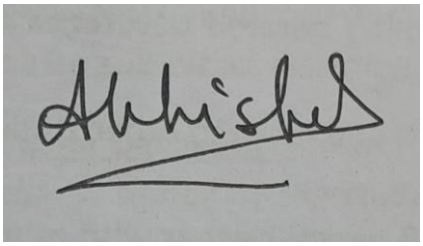
```
Last login: Tue Oct  1 08:30:58 on console
abhishekp@Abhisheks-MacBook-Air-3 ~ % cd documents
abhishekp@Abhisheks-MacBook-Air-3 documents % cd ddco
abhishekp@Abhisheks-MacBook-Air-3 ddco % iverilog -o dsn tb.v one.v
abhishekp@Abhisheks-MacBook-Air-3 ddco % vvp dsn
VCD info: dumpfile aluf.vcd opened for output.
Time    Reset   Op      I0      I1      O       Cout
0       1       00      0000    0000    0000    0
13      0       00      0000    0000    0000    0
26      0       00      aa55    55aa    ffff    0
36      0       00      ffff    0001    0000    1
46      0       00      0001    7fff    8000    0
56      0       01      0000    0000    0000    1
66      0       01      aa55    55aa    54ab    1
76      0       01      ffff    0001    fffe    1
86      0       01      0001    7fff    8002    0
96      0       10      0000    0000    0000    0
106     0       10      aa55    55aa    0000    0
116     0       10      ffff    0001    0001    1
126     0       10      0001    7fff    0001    0
136     0       11      0000    0000    0000    1
146     0       11      aa55    55aa    ffff    1
156     0       11      ffff    0001    ffff    1
166     0       11      0001    7fff    7fff    0
tb.v:92: $finish called at 2660 (100ps)
```


From: 0 sec
To: 266 ns
Marker 77 700 ps | Cursor 77 700 ps
SST
tb
Time
100 ns
200 ns
clk=1
cout=1
i=7
i0[15:0]=FF
i1[15:0]=00
o[15:0]=FF
op[1:0]=01
reset=0
Type Signals
reg clk
wire cout
integer i
reg i0[15:0]
reg i1[15:0]
wire o[15:0]
reg op[1:0]
reg reset
Append
Insert
Replace

	op[1:0]	i0[15:0]	i1[15:0]	Output
TESTVECTOR0	2’b00	16’h0000	16’h0000	16’h0000
TESTVECTOR1	2’b00	16’haa55	16’h55aa	16'h0000
TESTVECTOR2	2’b00	16'hffff	16'h0001	16'h0001
TESTVECTOR3	2’b00	16'h0001	16'h7fff	16'h0001
TESTVECTOR4	2'b01	16'h0000	16'h0000	16'h0000
TESTVECTOR5	2'b01	16’haa55	16’h55aa	16'hffff
TESTVECTOR6	2'b01	16'hffff	16'h0001	16'hffff
TESTVECTOR7	2'b01	16'h0001	16'h7fff	16'h7fff
TESTVECTOR8	2’b10	16’h0000	16’h0000	16'h0000
TESTVECTOR9	2’b10	16’haa55	16’h55aa	16'hffff
TESTVECTOR10	2’b10	16'hffff	16'h0001	16'hfffe
TESTVECTOR11	2’b10	16'h0001	16'h7fff	16'h7ffe
TESTVECTOR12	2’b11	16’h0000	16’h0000	16'h0000
TESTVECTOR13	2’b11	16’haa55	16’h55aa	16'h0000
TESTVECTOR14	2’b11	16'hffff	16'h0001	16'hfffe
TESTVECTOR15	2’b11	16'h0001	16'h7fff	16'h0000

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature: 

Name: ABHISHEK P

SRN: PES2UG23AM002

Section: A

Date: 02-10-2024