

# 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

x

one.v

```
1 module invert (input wire i, output wire o);
2   assign o = !i;
3 endmodule
4
5 module and2 (input wire i0, i1, output wire o);
6   assign o = i0 & i1;
7 endmodule
8
9 module or2 (input wire i0, i1, output wire o);
10  assign o = i0 | i1;
11 endmodule
12
13 module xor2 (input wire i0, i1, output wire o);
14  assign o = i0 ^ i1;
15 endmodule
16
17 module nand2 (input wire i0, i1, output wire o);
18   wire t;
19   and2 and2_0 (i0, i1, t);
20   invert invert_0 (t, o);
21 endmodule
22
23 module nor2 (input wire i0, i1, output wire o);
24   wire t;
25   or2 or2_0 (i0, i1, t);
26   invert invert_0 (t, o);
27 endmodule
28
29 module xnor2 (input wire i0, i1, output wire o);
30   wire t;
31   xor2 xor2_0 (i0, i1, t);
32   invert invert_0 (t, o);
33 endmodule
34
35 module and3 (input wire i0, i1, i2, output wire o);
36   wire t;
37   and2 and2_0 (i0, i1, t);
38   and2 and2_1 (i2, t, o);
39 endmodule
40
41 module or3 (input wire i0, i1, i2, output wire o);
42   wire t;
43   or2 or2_0 (i0, i1, t);
44   or2 or2_1 (i2, t, o);
45 endmodule
46
47 module nor3 (input wire i0, i1, i2, output wire o);
48   wire t;
49   or2 or2_0 (i0, i1, t);
50   nor2 nor2_0 (i2, t, o);
51 endmodule
52
53 module nand3 (input wire i0, i1, i2, output wire o);
54   wire t;
55   and2 and2_0 (i0, i1, t);
56   nand2 nand2_1 (i2, t, o);
57 endmodule
58
59 module xor3 (input wire i0, i1, i2, output wire o);
60   wire t;
61   xor2 xor2_0 (i0, i1, t);
62   xor2 xor2_1 (i2, t, o);
63 endmodule
64
65 module xnor3 (input wire i0, i1, i2, output wire o);
66   wire t;
67   xor2 xor2_0 (i0, i1, t);
68   xnor2 xnor2_0 (i2, t, o);
69 endmodule
70
71 module mux2 (input wire i0, i1, j, output wire o);
72   assign o = (j == 0) ? i0 : i1;
```

```

78     mux2 mux2_1 (i[2], i[3], j0, t1);
79     mux2 mux2_2 (t0, t1, j1, o);
80 endmodule
81
82 module mux8 (input wire [0:7] i, input wire j2, j1, j0, output wire o);
83     wire t0, t1;
84     mux4 mux4_0 (i[0:3], j1, j0, t0);
85     mux4 mux4_1 (i[4:7], j1, j0, t1);
86     mux2 mux2_0 (t0, t1, j2, o);
87 endmodule
88
89 module fa (input wire i0, i1, cin, output wire sum, cout);
90     wire t0, t1, t2;
91     xor3 xor3_0 (i0, i1, cin, sum);
92     and2 and2_0 (i0, i1, t0);
93     and2 and2_1 (i1, cin, t1);
94     and2 and2_2 (cin, i0, t2);
95     or3 or3_0 (t0, t1, t2, cout);
96 endmodule
97
98 module addsub (input wire addsub, i0, i1, cin, output wire sumdiff, cout);
99     wire t;
100     xor2 xor2_0 (i1, addsub, t);
101     fa fa_0 (i0, t, cin, sumdiff, cout);
102 endmodule
103
104 module alu_slice (input wire [1:0] op, input wire i0, i1, cin, output wire o, cout);
105     wire t_sumdiff, t_and, t_or, t_andor;
106     addsub addsub_0 (op[0], i0, i1, cin, t_sumdiff, cout);
107     and2 and2_0 (i0, i1, t_and);
108     or2 or2_0 (i0, i1, t_or);
109     mux2 mux2_0 (t_and, t_or, op[0], t_andor);
110     mux2 mux2_1 (t_sumdiff, t_andor, op[1], o);
111 endmodule
112
113 module alu (input wire [1:0] op, input wire [15:0] i0, i1, output wire [15:0] o, output wire cout);
114     wire [14:0] c;
115     alu_slice alu_slice_0 (op, i0[0], i1[0], op[0], o[0], c[0]);
116     alu_slice alu_slice_1 (op, i0[1], i1[1], c[0], o[1], c[1]);
117     alu_slice alu_slice_2 (op, i0[2], i1[2], c[1], o[2], c[2]);
118     alu_slice alu_slice_3 (op, i0[3], i1[3], c[2], o[3], c[3]);
119     alu_slice alu_slice_4 (op, i0[4], i1[4], c[3], o[4], c[4]);
120     alu_slice alu_slice_5 (op, i0[5], i1[5], c[4], o[5], c[5]);
121     alu_slice alu_slice_6 (op, i0[6], i1[6], c[5], o[6], c[6]);
122     alu_slice alu_slice_7 (op, i0[7], i1[7], c[6], o[7], c[7]);
123     alu_slice alu_slice_8 (op, i0[8], i1[8], c[7], o[8], c[8]);
124     alu_slice alu_slice_9 (op, i0[9], i1[9], c[8], o[9], c[9]);
125     alu_slice alu_slice_10 (op, i0[10], i1[10], c[9], o[10], c[10]);
126     alu_slice alu_slice_11 (op, i0[11], i1[11], c[10], o[11], c[11]);
127     alu_slice alu_slice_12 (op, i0[12], i1[12], c[11], o[12], c[12]);
128     alu_slice alu_slice_13 (op, i0[13], i1[13], c[12], o[13], c[13]);
129     alu_slice alu_slice_14 (op, i0[14], i1[14], c[13], o[14], c[14]);
130     alu_slice alu_slice_15 (op, i0[15], i1[15], c[14], o[15], cout);
131 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 ddc
abhishekp@Abhisheks-MacBook-Air-3 ddc % iverilog -o dsn tb.v one.v
abhishekp@Abhisheks-MacBook-Air-3 ddc % 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)
```

Waveform simulation showing digital signals over time (0 to 266 ns). The signals are:

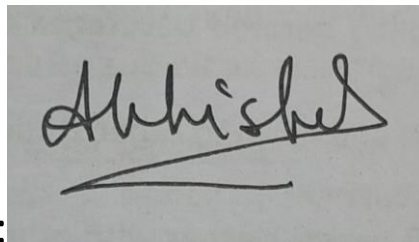
- clk=1 (Clock signal, periodic square wave)
- cout=1 (Carry-out signal, periodic square wave)
- i=7 (Index signal, constant at 7)
- i0[15:0] (Input 0, 16-bit bus, values: 0000, AA55, FFFF, 0001, 0000, AA55, FFFF, 0001, 0000, AA55, FFFF, 0001, 0000, AA55, FFFF, 0001)
- i1[15:0] (Input 1, 16-bit bus, values: 0000, 55AA, 0001, 7FFF, 0000, 55AA, 0001, 7FFF, 0000, 55AA, 0001, 7FFF, 0000, 55AA, 0001, 7FFF)
- o[15:0] (Output, 16-bit bus, values: 0000, FFFF, 0000, 8000, 0000, 54AB, FFFE, 8002, 0000, 0001, 0000, FFFF, 7FFF)
- op[1:0] (Operation code, 2-bit bus, values: 00, 01, 10, 11)
- reset=0 (Reset signal, constant at 0)

The waveform is displayed on a black background with green and blue traces. The time axis is marked at 0, 100 ns, and 200 ns. The signal names are listed on the left side of the waveform area.

	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.

A handwritten signature in black ink on a light gray background. The signature is cursive and appears to read 'Abhishek'. Below the main signature, there is a long, horizontal, slightly wavy line that extends to the left.

Signature:

Name: ABHISHEK P

SRN: PES2UG23AM002

Section: A

Date: 02-10-2024