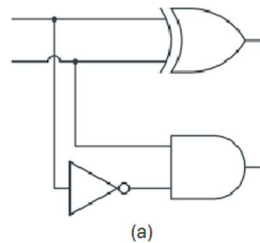
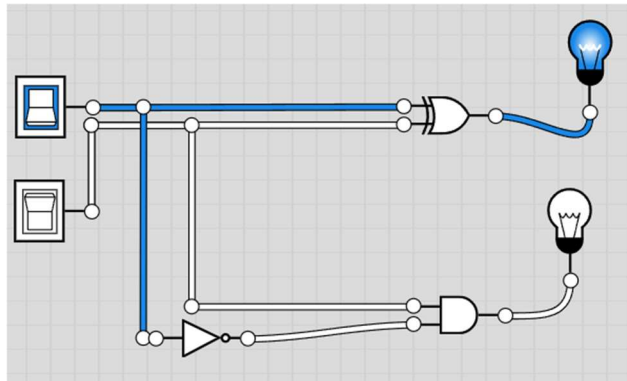


Class Activity 1

1. Design the below combinational circuits using VHDL and simulate the same in EDA playground with a suitable test case in testbench.



A.



Truth Table			
a	b	Sum	Carry
false	false	false	false
false	true	true	true
true	false	true	false
true	true	false	false

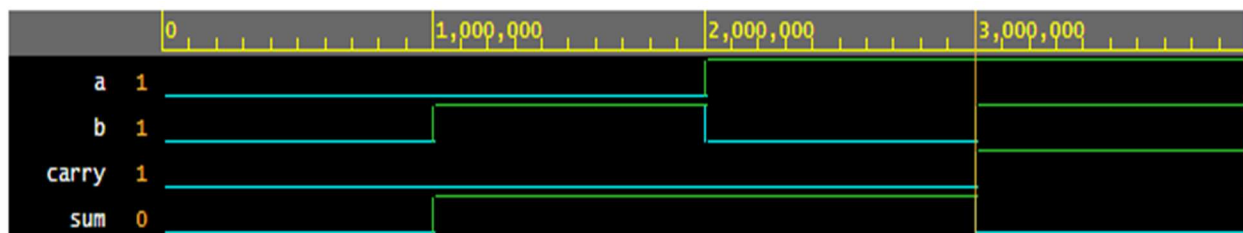
Design:

```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 -- ENTITY
5 entity HalfAdder is
6 port(
7     a: in std_logic;
8     b: in std_logic;
9     sum: out std_logic;
10    carry: out std_logic);
11 end HalfAdder;
12
13 -- ARCHITECTURE
14 architecture dataflow of HalfAdder is
15 begin
16     sum <= a xor b;
17     carry <= not a and b;
18 end dataflow;
```

Testbench:

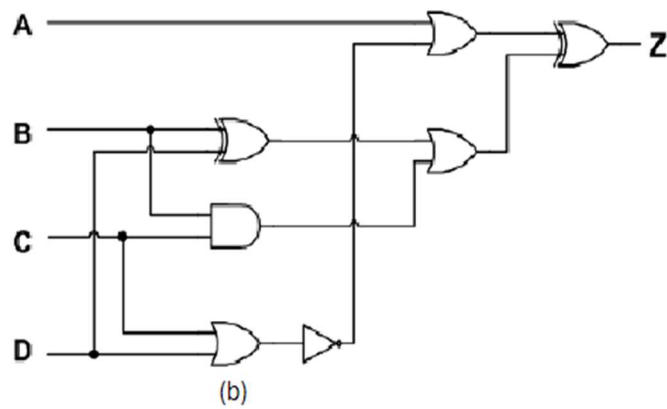
```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 -- TESTBENCH ENTITY
5 entity testbench is
6 --empty
7 end testbench;
8
9 architecture tb of testbench is
10 -- DUT COMPONENT
11 component HalfAdder is
12 port(
13     a: in std_logic;
14     b: in std_logic;
15     sum: out std_logic;
16     carry: out std_logic);
17 end component;
18
19 signal aIN, bIN, SUM, CARRY: std_logic;
20
21 begin
22
23 -- CONNECT DUT
24 DUT:HalfAdder port map(aIN, bIN, SUM, CARRY);
25
26     process
27     begin
28         aIN <='0';
29         bIN <='0';
30         wait for 1 ns;
31
32         aIN <='0';
33         bIN <='1';
34         wait for 1 ns;
35
36         aIN <='1';
37         bIN <='0';
38         wait for 1 ns;
39
40         aIN <='1';
41         bIN <='1';
42         wait for 1 ns;
43
44         -- CLEAR INPUTS
45         aIN <='0';
46         bIN <='0';
47         wait;
48     end process;
49 end tb;
```

Wave:

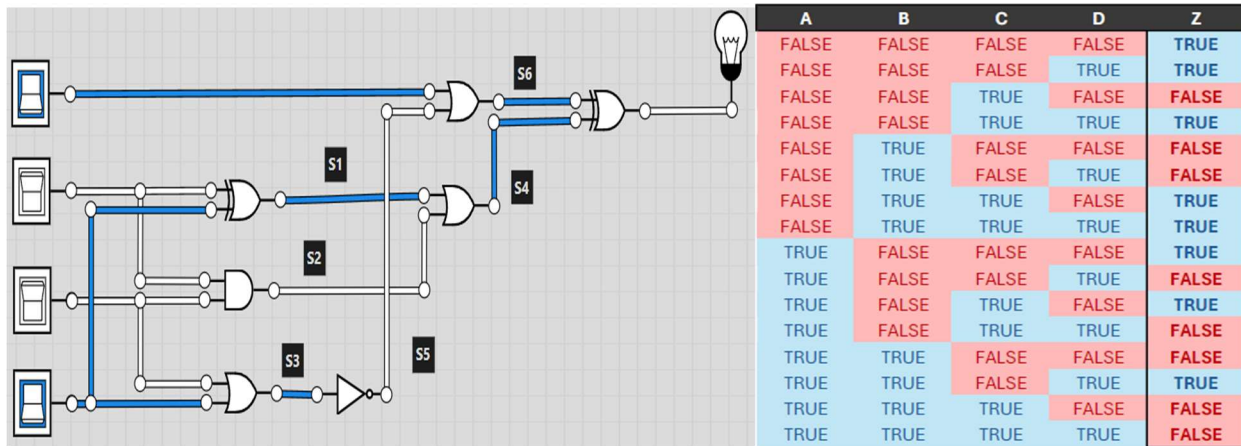


Observation:

All I did was modify my half adder design with a not function for the carry, but we can see from the wave and truth table that they are the same outputs.



B.



Design:

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  -- ENTITY
5  entity comboCircuit is
6  port(
7      a, b, c, d: in std_logic;
8      z: out std_logic
9  );
10 end comboCircuit;
11
12 -- ARCHITECTURE
13 architecture dataflow of comboCircuit is
14 signal S1, S2, S3, S4, S5, S6: std_logic;
15 begin
16     S1 <= b xor d;
17     S2 <= b and c;
18     S3 <= c or d;
19     S4 <= S1 or S2;
20     S5 <= not S3;
21     S6 <= a or S5;
22     z <= S6 xor S4;
23 end dataflow;
24

```

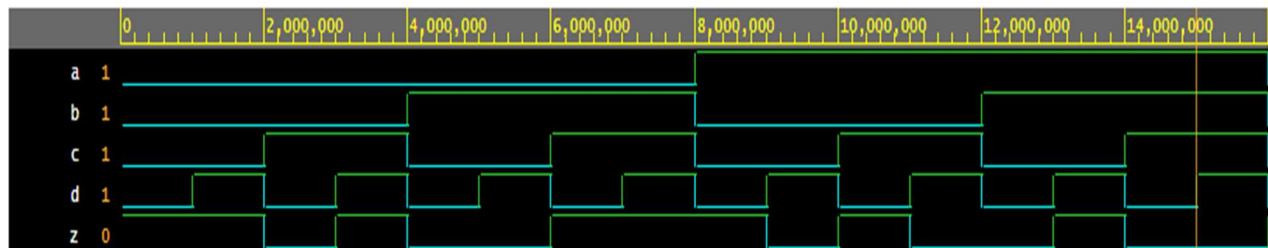
Testbench:

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  -- TESTBENCH ENTITY
5  entity testbench is
6  --empty
7  end testbench;
8
9  architecture tb of testbench is
10 -- DUT COMPONENT
11 component comboCircuit is
12 port(
13     a,b,c,d: in std_logic;
14     z: out std_logic);
15 end component;
16
17 signal aIN, bIN, cIN, dIN, zOUT : std_logic;
18
19 begin
20
21 -- CONNECT DUT
22 DUT: comboCircuit port map(aIN, bIN, cIN, dIN, zOUT);
23
24 process
25 begin
26     aIN <= '0';
27     bIN <= '0';
28     cIN <= '0';
29     dIN <= '0';
30     wait for 1 ns;
31
32     aIN <= '0';
33     bIN <= '0';
34     cIN <= '0';
35     dIN <= '1';
36     wait for 1 ns;
37
38     aIN <= '0';
39     bIN <= '0';
40     cIN <= '1';
41     dIN <= '0';
42     wait for 1 ns;
43
44     aIN <= '0';
45     bIN <= '0';
46     cIN <= '1';
47     dIN <= '1';
48     wait for 1 ns;
49
50     aIN <= '0';
51     bIN <= '1';
52     cIN <= '0';
53     dIN <= '0';
54     wait for 1 ns;
55
56     aIN <= '0';
57     bIN <= '1';
58     cIN <= '0';
59     dIN <= '1';
60     wait for 1 ns;
61
62     aIN <= '0';
63     bIN <= '1';
64     cIN <= '1';
65     dIN <= '0';
66     wait for 1 ns;
67
68     aIN <= '0';
69     bIN <= '1';
70     cIN <= '1';
71     dIN <= '1';
72     wait for 1 ns;
73
74     aIN <= '1';
75     bIN <= '0';
76     cIN <= '0';
77     dIN <= '0';
78     wait for 1 ns;
79
80     aIN <= '1';
81     bIN <= '0';
82     cIN <= '0';
83     dIN <= '1';
84     wait for 1 ns;
85
86     aIN <= '1';
87     bIN <= '0';
88     cIN <= '1';
89     dIN <= '0';
90     wait for 1 ns;
91
92     aIN <= '1';
93     bIN <= '0';
94     cIN <= '1';
95     dIN <= '1';
96     wait for 1 ns;
97
98     aIN <= '1';
99     bIN <= '1';
100    cIN <= '0';
101    dIN <= '0';
102    wait for 1 ns;
103
104    aIN <= '1';
105    bIN <= '1';
106    cIN <= '0';
107    dIN <= '1';
108    wait for 1 ns;
109
110    aIN <= '1';
111    bIN <= '1';
112    cIN <= '1';
113    dIN <= '0';
114    wait for 1 ns;
115
116    aIN <= '1';
117    bIN <= '1';
118    cIN <= '1';
119    dIN <= '1';
120    wait for 1 ns;
121
122    -- CLEAR INPUTS
123    aIN <= '0';
124    bIN <= '0';
125    cIN <= '0';
126    dIN <= '0';
127    wait;
128 end process;
129 end tb;

```

Wave:



Observation:

This one was a really good learning experience, I feel more comfortable in VHDL now after this exercise. As we can see, the wave and truth table match each other, implementation was once again successful.