November 11, 2019

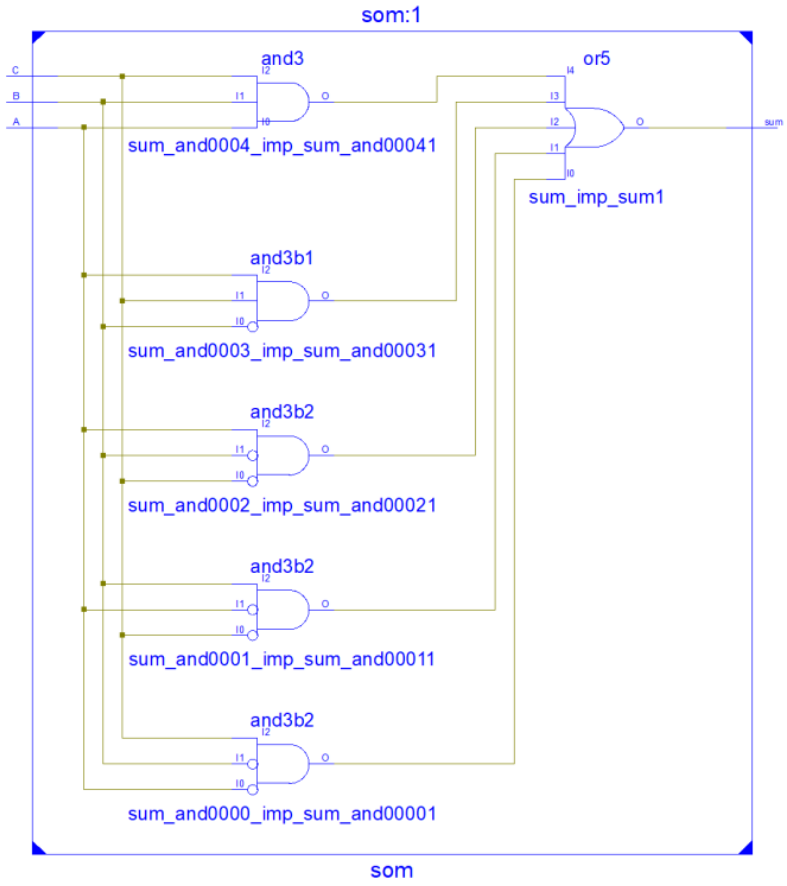
	HACETTEPE UNIVERSITY ELECTRICAL AND ELECTRONICS ENGINEERING ELE227 PRELIMINARY WORK #1
Name	Muhammed Baki Almacı
Student ID	21627983

1) Convert following decimal numbers to 8-bit binary numbers

2) Convert following 8-bit binary numbers to decimal and hexadecimal

3)

a.



RTL SCHEMATIC

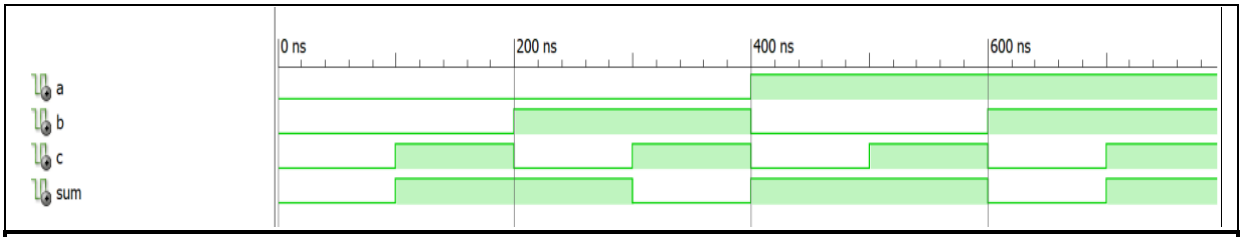
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity som is
   Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          C : in  STD_LOGIC;
          sum : out  STD_LOGIC);
end som;

architecture Behavioral of som is

begin
          sum <= ((NOT A) AND (NOT B) AND C) OR ((NOT A) AND (B) AND (NOT C)) OR ((A) AND
(NOT B) AND (NOT C)) OR ((A) AND (NOT B) AND C) OR((A) AND (B) AND (C));

end Behavioral;
```

VHDL CODE

b.

```
stim_proc: process
  begin
        A <= '0';
        B <= '0';
        C <= '0';
        wait for 100 ns;
        A <= '0';
        B <= '0';
        C <= '1';
        wait for 100 ns;
        A <= '0';
        B <= '1';
        C <= '0';
        wait for 100 ns;
        A <= '0';
        B <= '1';
        C <= '1';
        wait for 100 ns;
        A <= '1';
        B <= '0';
        C <= '0';
        wait for 100 ns;
        A <= '1';
        B <= '0';
        C <= '1';
        wait for 100 ns;
        A <= '1';
        B <= '1';
        C <= '0';
        wait for 100 ns;
        A <= '1';
        B <= '1';
        C <= '1';
   wait;
end process;
```

TEST BENCH CODE



TEST BENCH GRAPH

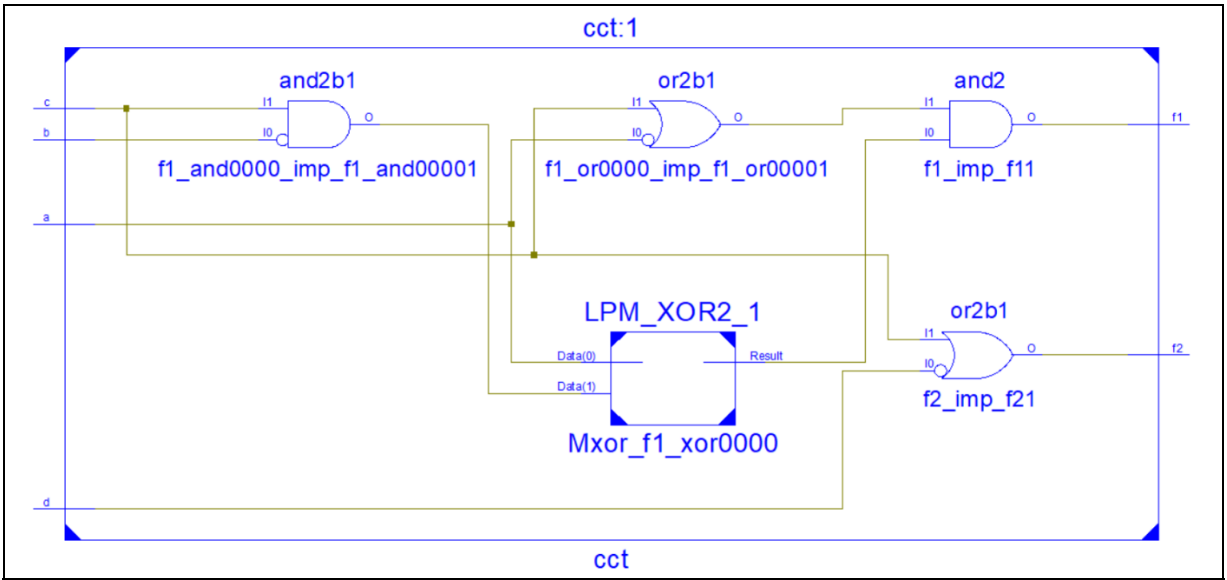
4)

a.

b.

c.

d.



RTL SCHEMATIC

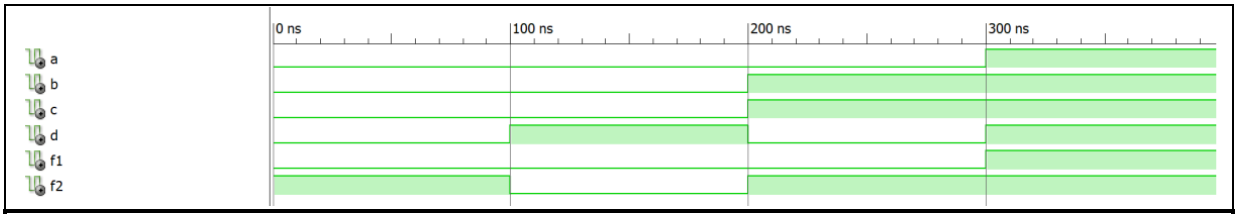
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity cct is
   Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          c : in  STD_LOGIC;
          d : in  STD_LOGIC;
          f1 : out  STD_LOGIC;
          f2 : out  STD_LOGIC);
end cct;

architecture Behavioral of cct is

begin
            f1 <= ((c and (not b)) xor a) and ((not a) or c);
            f2 <= (c and d) or (not d);
end Behavioral;
```

VHDL CODE

e.

```
stim_proc: process
begin
  -- hold reset state for 100 ns.
                a <= '0';
                b <= '0';
                c <= '0';
                d <= '0';
  wait for 100 ns;
                a <= '0';
                b <= '0';
                c <= '0';
                d <= '1';
  wait for 100 ns;
                a <= '0';
                b <= '1';
                c <= '1';
                d <= '0';
  wait for 100 ns;
                a <= '1';
                b <= '1';
                c <= '1';
                d <= '1';
  wait for 100 ns;
                wait;
end process;
```

TEST BENCH CODE



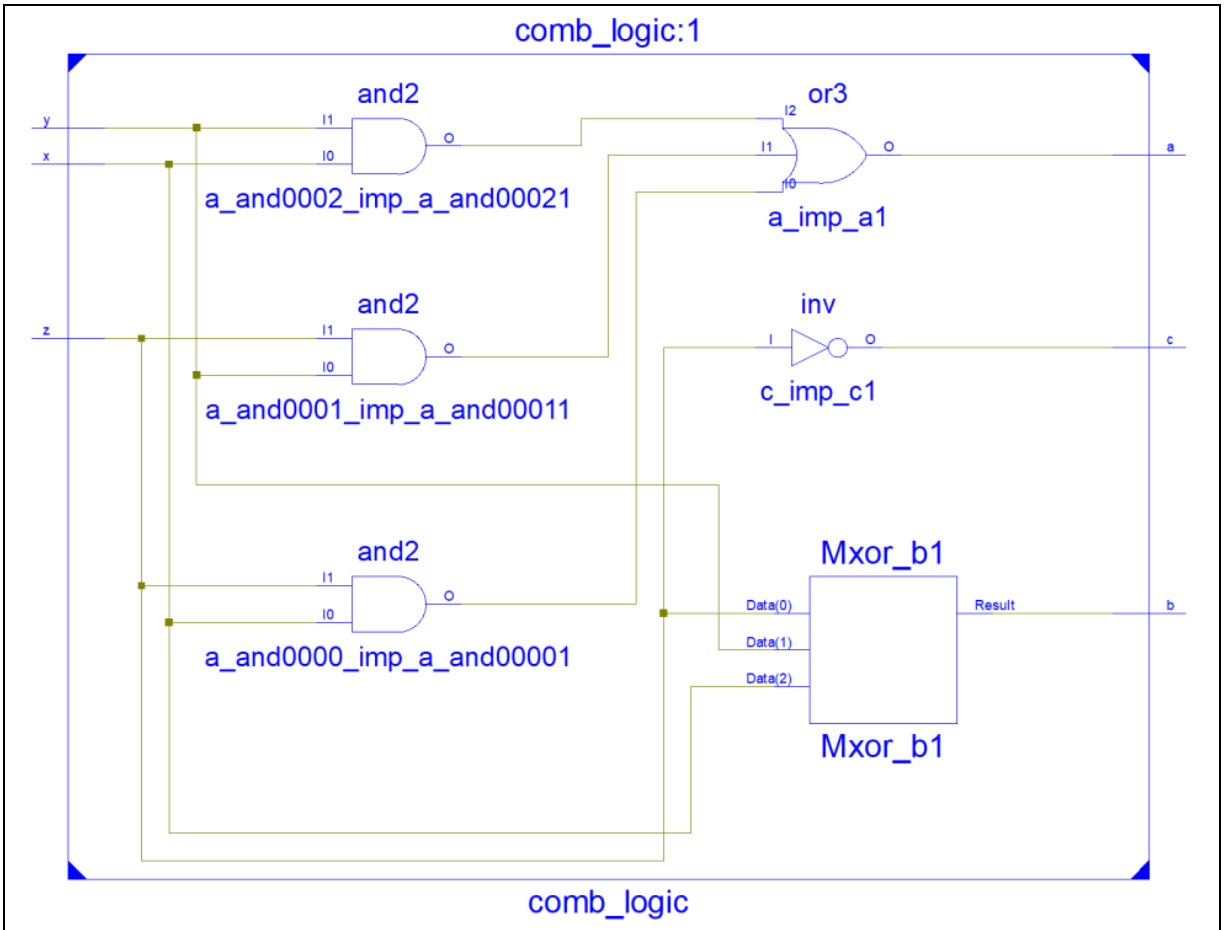
TEST BENCH GRAPH

5)

a.

b.

c.



RTL SCHEMATIC

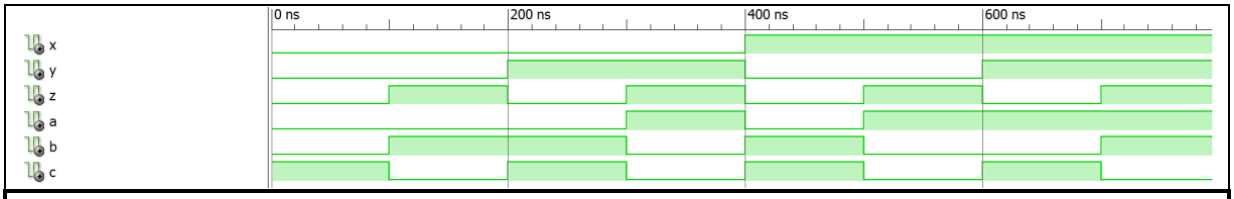
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity comb_logic is
   Port ( x : in  STD_LOGIC;
          y : in  STD_LOGIC;
          z : in  STD_LOGIC;
          a : out  STD_LOGIC;
          b : out  STD_LOGIC;
          c : out  STD_LOGIC);
end comb_logic;

architecture Behavioral of comb_logic is

begin
            a <= (x and z) or (y and z) or (x and y);
            b <= x xor y xor z;
            c <= not z;

end Behavioral;
```

VHDL CODE

d.

```
stim_proc: process
begin
        x <= '0';
        y <= '0';
        z <= '0';
  wait for 100 ns;
        x <= '0';
        y <= '0';
        z <= '1';
  wait for 100 ns;
        x <= '0';
        y <= '1';
        z <= '0';
  wait for 100 ns;
        x <= '0';
        y <= '1';
        z <= '1';
  wait for 100 ns;
        x <= '1';
        y <= '0';
        z <= '0';
  wait for 100 ns;
        x <= '1';
        y <= '0';
        z <= '1';
  wait for 100 ns;
        x <= '1';
        y <= '1';
        z <= '0';
  wait for 100 ns;
        x <= '1';
        y <= '1';
        z <= '1';
  wait;
end process;
```

TEST BENCH CODE



TEST BENCH GRAPH