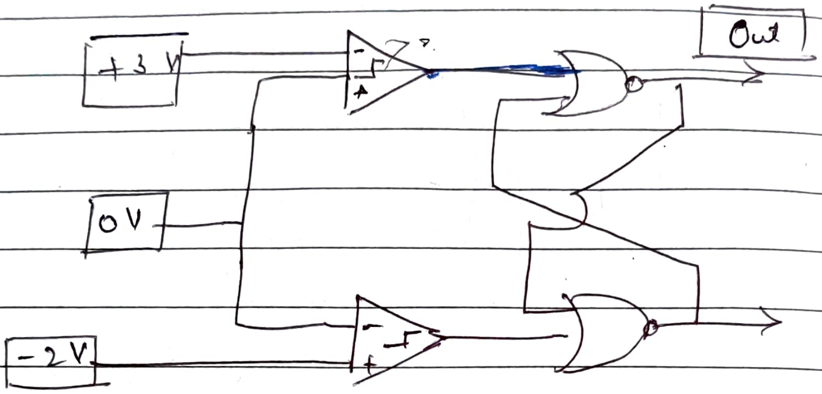
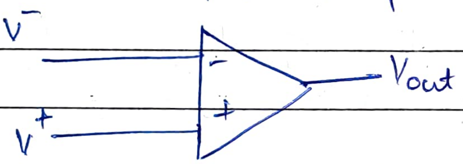
202051213
Archit Agrawal
EC201 Assignment 6-10

Assignment 6

1. Realize the circuit shown below contains two comparators



We know that, in comparator,



$V_{out} = 1$ when $V_+ > V_-$
and $V_{out} = 0$ when $V_+ < V_-$

Therefore, in the above given circuit, the comparator at top outputs 0 and the comparator at bottom also outputs 0.

∴ in the latch, we have $R = 0$ and $S = 0$, and this is the memory state in a ~~latch~~ SR latch. Hence, the circuit will retain the previous output i.e. it will retain its memory.

2. A new clocked A-B flip flop is defined with two inputs X AND Y in addition to the clock input. The flip flop function as follows:

→ If AB = 00 , the flip flop changes state with each clock pulse

→ If AB = 01 the flip flop state Z becomes 1 with the next clock pulse.

→ If AB = 10 , the flip flop state Z becomes 0 with next clock pulse.

→ If AB = 11 , the change of state occurs with the clock pulse.

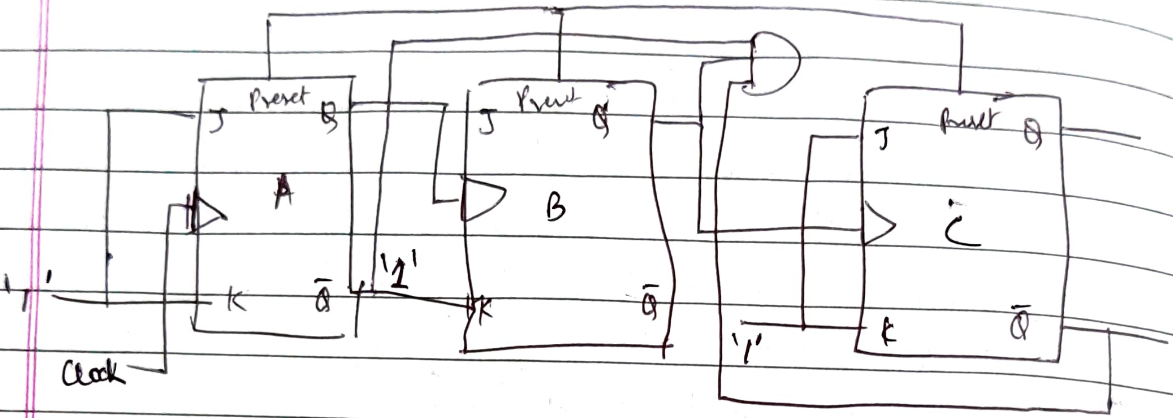
Write the truth-table for the A-B flip flop.

Let Q be the present state of the circuit and Q^* be the next state. The truth table ~~will be~~ for the flip flop will be.

X	Y	Q^*
0	0	$\bar{Q}$
0	1	1
1	0	0
1	1	$\bar{Q}$

Assignment 7

3. Find the mode of a counter and also find that it is up counter or down counter.

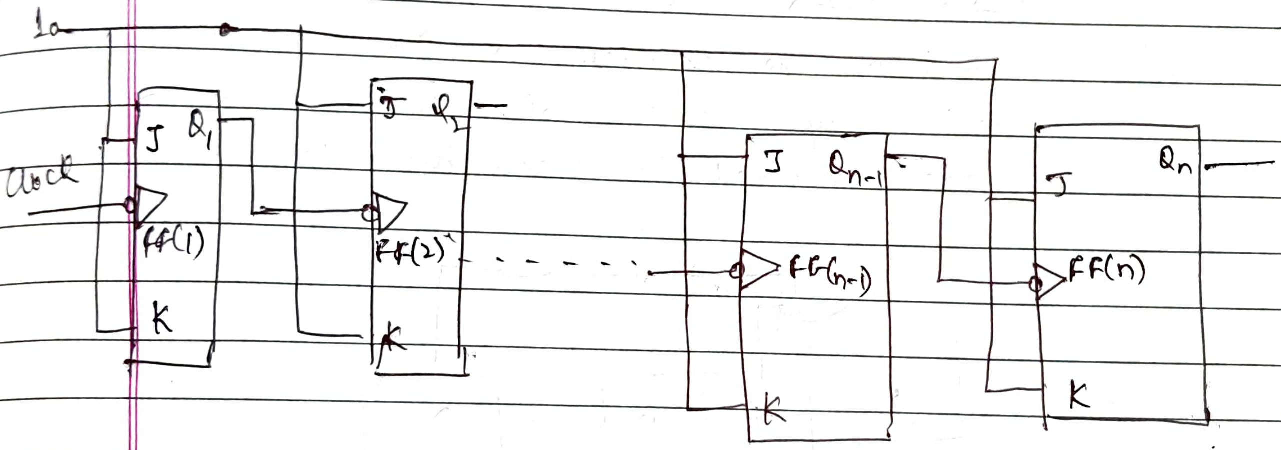


Clock is not neglected $\nRightarrow$ up counter

→ $\bar{Q}_0 = 1$, $Q_1 = 1$ < $\bar{Q}_2 = 1$; then the counter again reaches to 111 or initial value.

→ There fore, the last value of the counter will be 010 or 2.

4. It is desired to design a binary ripple counter of the type shown in fig. that is capable of counting the number of items passing on a conveyor belt. Each time an item passes ~~At~~ given point, a pulse is generated that can be used as clock signal. If the maximum number of items to be counted is 6000, determine the no. of flipflops required.



From the figure, it can be seen (also given in question) that it is a asynchronous counter. Asynchranous counters (binary) can store 2^n number if there are n - flip-flops present. (0 to $2^n - 1$)

We need to count ~~the~~ - a maximum number of 6000 items.

~~$\therefore$ $2^n \geq 6000$~~

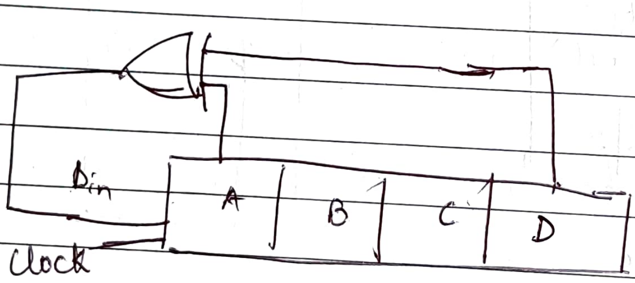
$\therefore$ $2^n - 1 \geq 6000$

$2^n \geq 6000$

the smallest integer that satisfies this condition is 13. Hence, a total of 13 flip-flops will be required to design ~~a~~ ~~counter~~ the given counter.

Assignment 8

5. A 4-bit shift register configured for right-shift operation i.e. $D_{in} \rightarrow A$, $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, is shown. If the present state of the register is ABCD = 1101, how many number of clock cycles are required to reach the state = 1111?



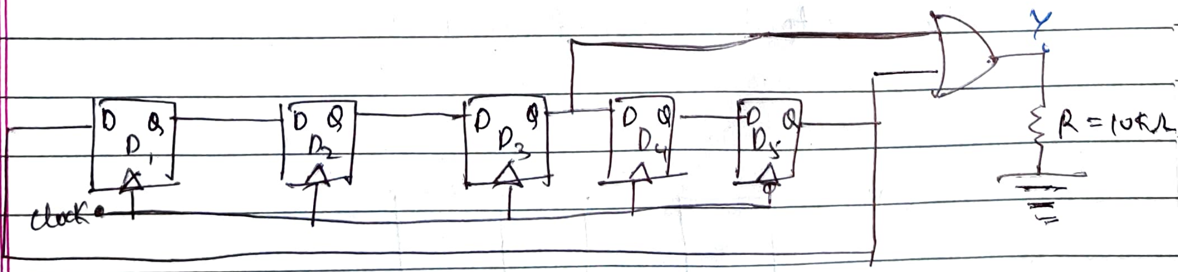
Since, A is XOR of previously stored A and D. New inputs to B, C, D are shown below. as a table for each pulse of clock.

Let clock pulse 0 denotes the present state i.e. 1101,

clk	A (A ⊕ D)	B	C	D	
0	1	1	0	1	← present state
1	0	1	1	0	
2	0	0	1	1	
3	1	0	0	1	
4	0	1	0	0	
5	0	0	1	0	
6	0	0	0	1	
7	1	0	0	0	
8	1	1	0	0	
9	1	1	1	0	
10	1	1	1	1	required state

Hence, 10 clock cycles will be required to reach the state 1111.

Q.6. Assume that all the digital gates in the circuit shown in figure are ideal, the resistor R = 10K Ω and the supply voltage is 5 V. The D flip flops D_1, D_2, D_3, D_4 and D_5 are initialized with logic values 0, 1, 0, 1 and 0, respectively. The clock has a 30% duty cycle.



Find out the average power dissipated (in mW) in the resistor R?

Let us first analyze the circuit and try to configure the output wave form.

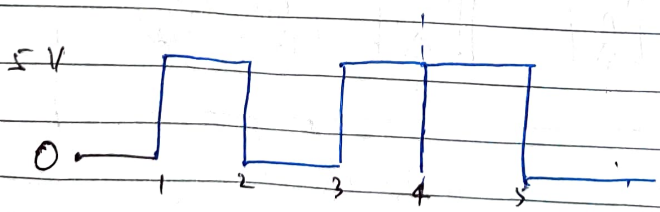
the output ~~is~~ $Y = D_3 + D_5$.

→ output ~~of~~ D_1 is input to D_2, output of D_2 is input to D_3 and so on. The output of D_5 is input to D_1 also. Let us try to find the time period of output waveform.

	D_1	D_2	D_3	D_4	D_5	$Y = D_3 + D_5$
Present state	0	1	0	1	0	0
1→	0	0	1	0	1	1
2→	1	0	0	1	0	0
3→	0	1	0	0	1	1
4→	1	0	1	0	0	1
5→	0	1	0	1	0	0

In the 5th clock pulse, the flip flops reached their initial state; and hence, the output will repeat for each 5 clock pulses.

The output wave form is:



∴ average power dissipated will be:

$$P_{avg} = \frac{V^2}{R} \times \text{duty cycle}$$

$$= \frac{V^2}{R} \times \frac{T_{High}}{T_{total}}$$

$$= \frac{V^2}{R} \times \frac{3}{5}$$

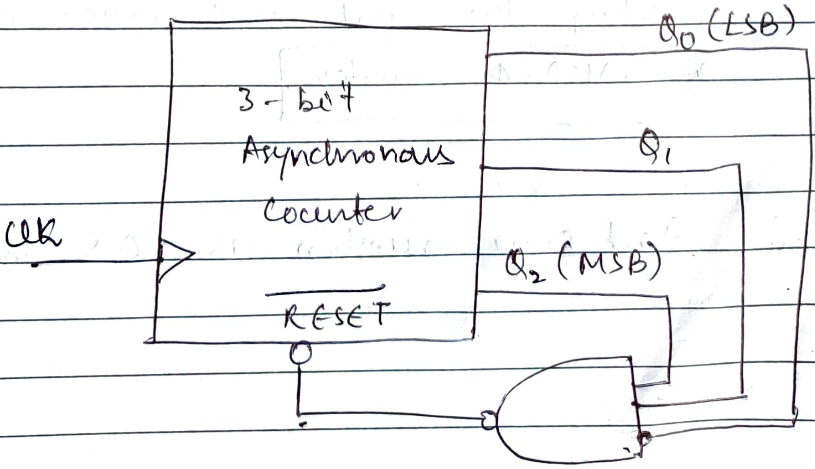
$$P_{avg} = \frac{5 \times \cancel{5}}{10K} \times \frac{3}{\cancel{5}}$$

$$\boxed{P_{avg} = 1.5\, mW}$$

Hence, the average power dissipated will be 1.5 mW.

Assignment-9

Q.7. For the circuit in the figure, the delay of bubbled NAND gate is 2ns. and that of the counter is assumed to be zero.



If the clock (clk) frequency is 1 GHz, then the counter behaves as a ____ ?

Let us try analyzing the counter circuit ~~to~~ table.

	current states			o/p of	next states		
clk	Q_2	Q_1	Q_0	NAND	Q_2^*	Q_1^*	Q_0^*
0	0	0	0	1	0	0	0
1	0	0	1	1	0	1	0
2	0	1	0	1	0	1	1
3	0	1	1	1	1	0	0
4	1	0	0	1	1	0	1
5	1	0	1	1	1	1	0
6	1	1	0	0	0	0	0

As in 6th cycle, preset input recieved 0 as input. Hence, it resets the counter.

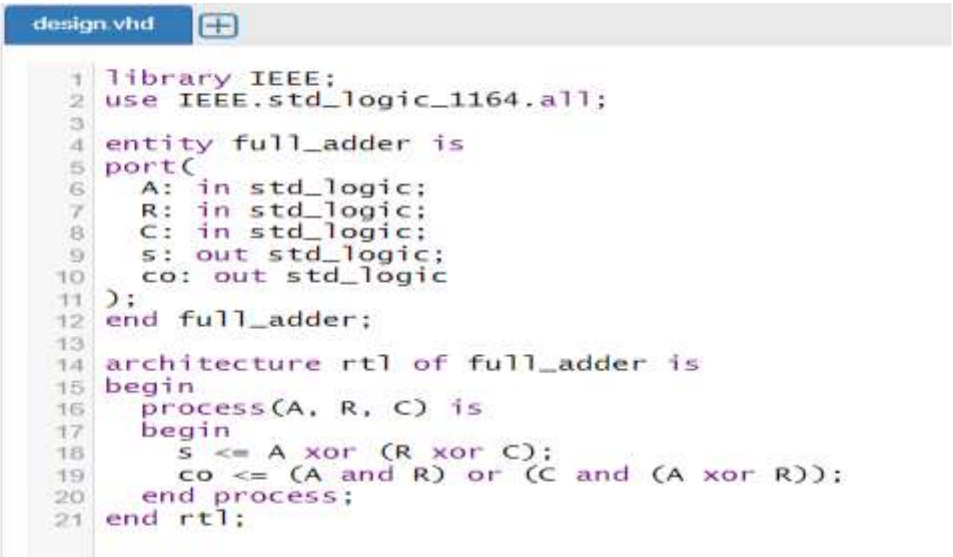
When, there was no propagation delay, then the counter was acting as MOD-6 counter. But since, there is a propagation delay of 2 ns in NAND gate and frequency of clock is 1 GHz, the NAND gate will allow 2 extra clock pulses before reseting the count and hence, it is a $\boxed{\text{MOD-8 Counter.}}$

Q.8. Exact same question as Q.5 in Assigment 8.

Assignment 10

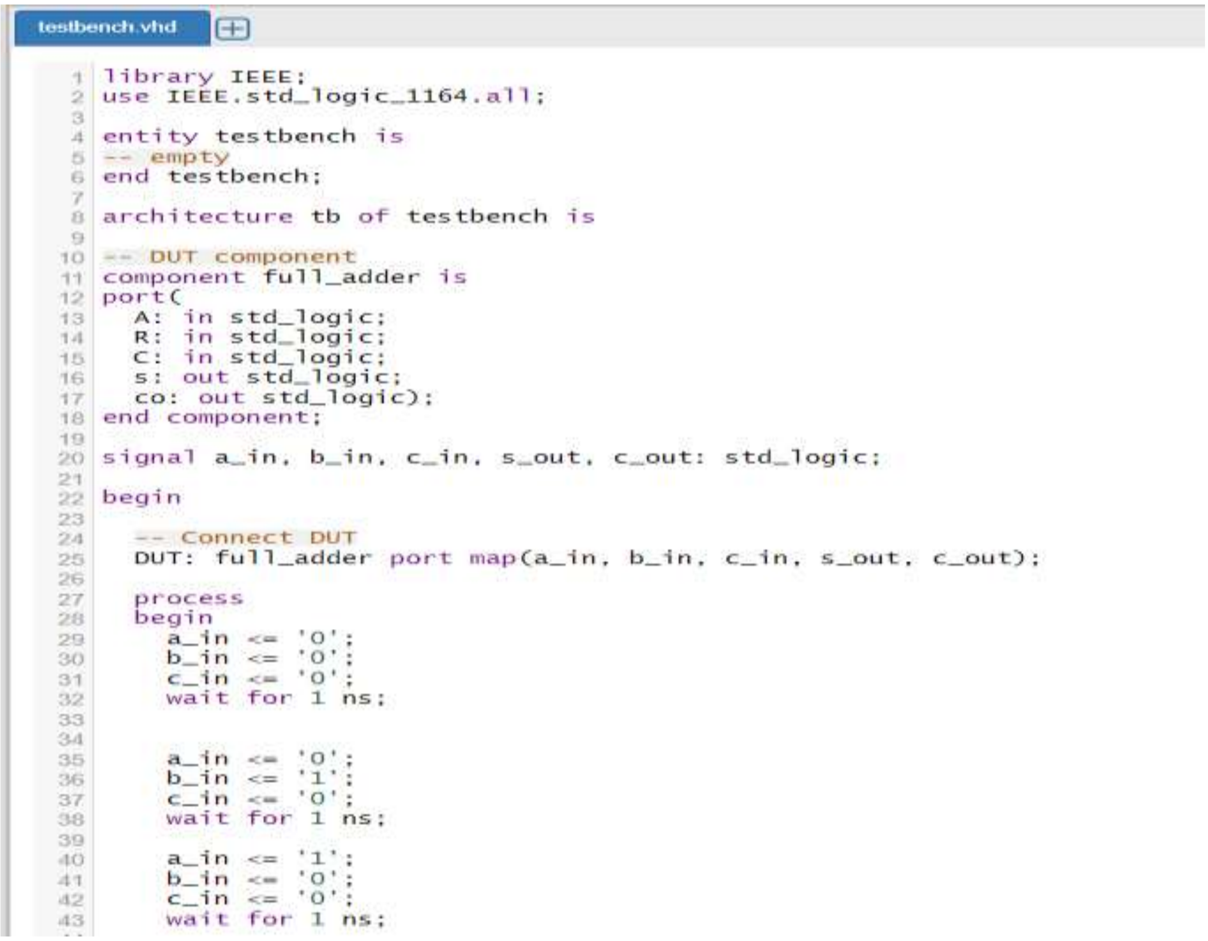
Q.9

Code of Design



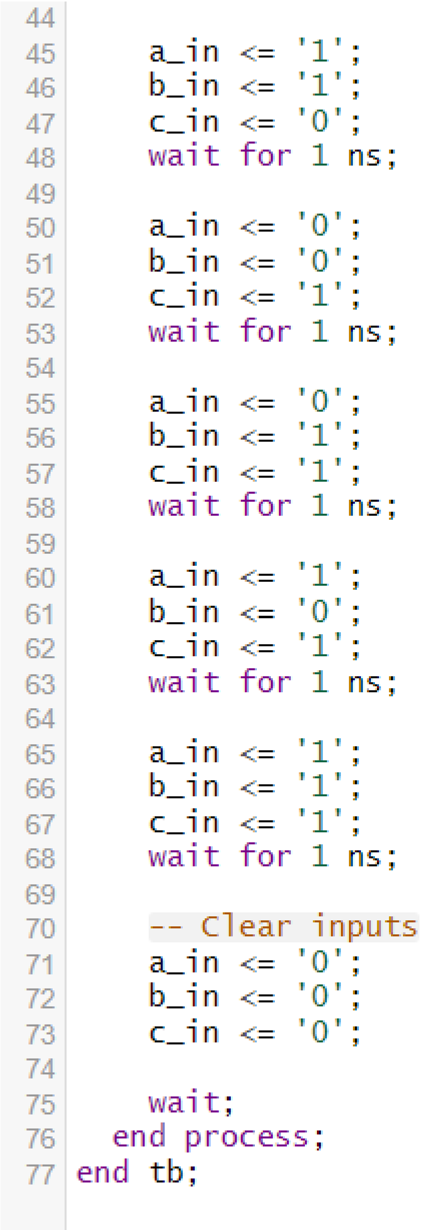
```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity full_adder is
port(
  A: in std_logic;
  R: in std_logic;
  C: in std_logic;
  s: out std_logic;
  co: out std_logic
);
end full_adder;

architecture rtl of full_adder is
begin
  process(A, R, C) is
  begin
    s <= A xor (R xor C);
    co <= (A and R) or (C and (A xor R));
  end process;
end rtl;
```

Code of Testbench



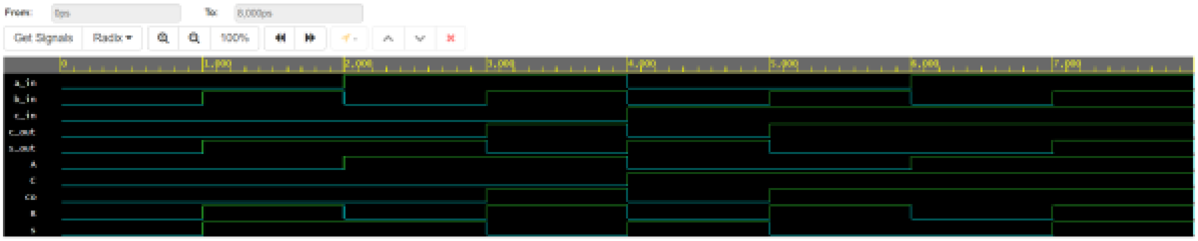
```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity testbench is
-- empty
end testbench;

architecture tb of testbench is

-- DUT component
component full_adder is
port(
  A: in std_logic;
  R: in std_logic;
  C: in std_logic;
  s: out std_logic;
  co: out std_logic);
end component;

signal a_in, b_in, c_in, s_out, c_out: std_logic;

begin

  -- Connect DUT
  DUT: full_adder port map(a_in, b_in, c_in, s_out, c_out);

  process
  begin
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;


    a_in <= '0';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;

    a_in <= '1';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;
```

Assignment 10



```

    a_in <= '1';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;

    a_in <= '0';
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;

    a_in <= '0';
    b_in <= '1';
    c_in <= '1';
    wait for 1 ns;

    a_in <= '1';
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;

    a_in <= '1';
    b_in <= '1';
    c_in <= '1';
    wait for 1 ns;

    -- Clear inputs
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';

    wait;
  end process;
end tb;
```

Output

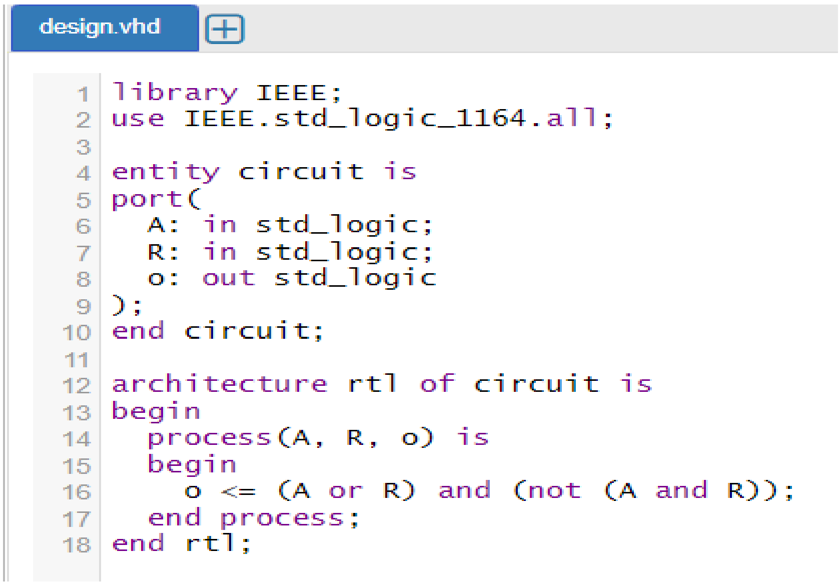


Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Assignment 10

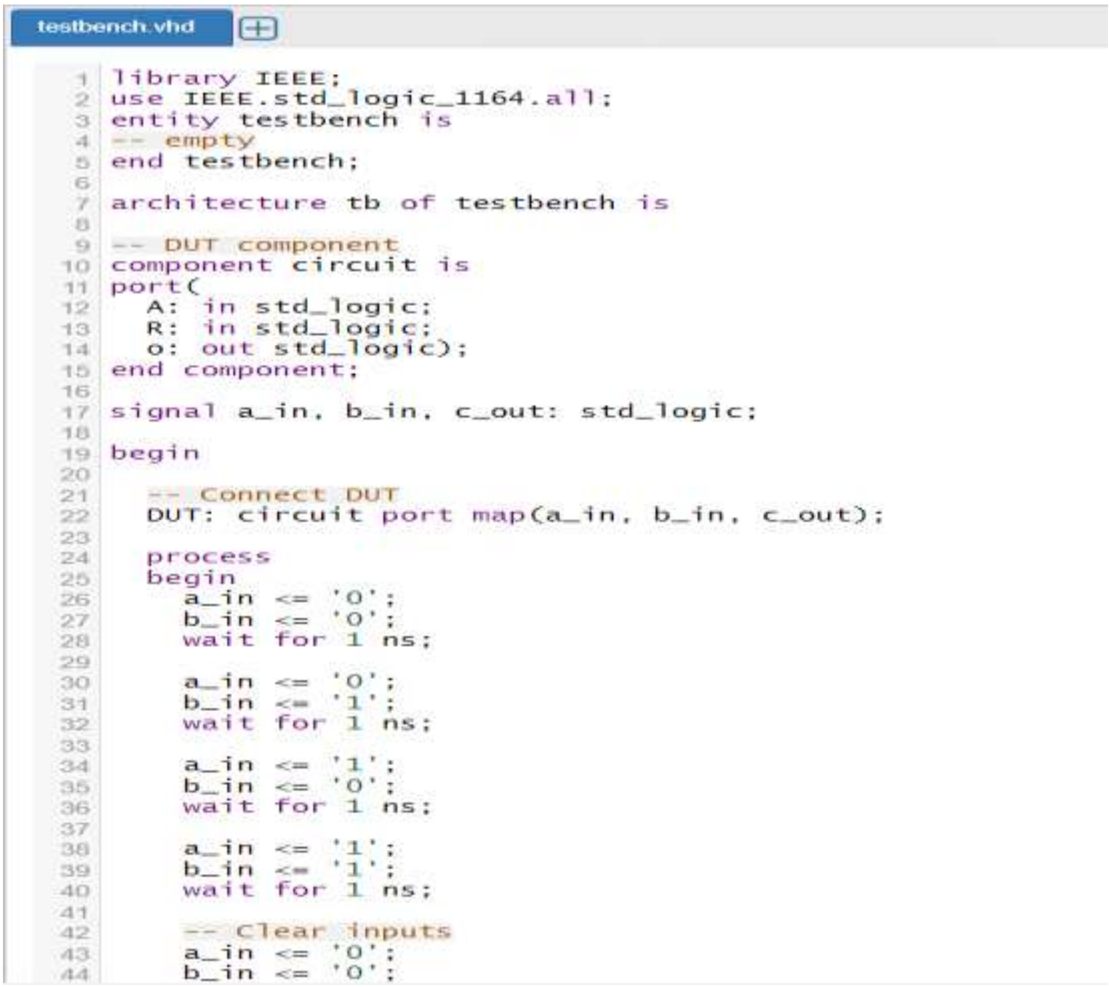
Q.10

Code of Design



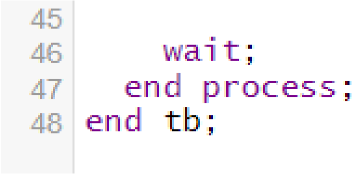
```
library IEEE;
use IEEE.std_logic_1164.all;

entity circuit is
port(
  A: in std_logic;
  R: in std_logic;
  o: out std_logic
);
end circuit;

architecture rtl of circuit is
begin
  process(A, R, o) is
  begin
    o <= (A or R) and (not (A and R));
  end process;
end rtl;
```

Code of testbench



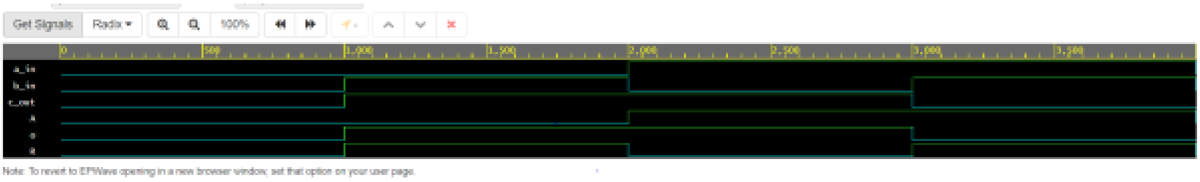
```
library IEEE;
use IEEE.std_logic_1164.all;
entity testbench is
-- empty
end testbench;

architecture tb of testbench is

-- DUT component
component circuit is
port(
  A: in std_logic;
  R: in std_logic;
  o: out std_logic);
end component;

signal a_in, b_in, c_out: std_logic;

begin

  -- Connect DUT
  DUT: circuit port map(a_in, b_in, c_out);

  process
  begin
    a_in <= '0';
    b_in <= '0';
    wait for 1 ns;

    a_in <= '0';
    b_in <= '1';
    wait for 1 ns;

    a_in <= '1';
    b_in <= '0';
    wait for 1 ns;

    a_in <= '1';
    b_in <= '1';
    wait for 1 ns;

    -- Clear inputs
    a_in <= '0';
    b_in <= '0';
```

Assignment 10



```
    wait;
  end process;
end tb;
```

Output



Note: To revert to EPWave opening in a new browser window, set that option on your user page.