Course Name:	**Hardware Description Language Lab (2UXL401)**	**Semester:**	**IV**
Date of Performance:	**23 / 03 / 2021**	**Batch No:**	**B2**
Faculty Name:	**Prof. Bhargavi Kaslikar**	**Roll No:**	**1912060**
Faculty Sign & Date:		**Grade/Marks:**	

Experiment No: 4

Title: Use of sequential statements:

a) Counter using IC 74163
b) Asynchronous Counter using Flip flops(3 Bit)

Aim and Objective of the Experiment:

Write a VHDL code for implementing a JK flip flop.

Write a VHDL code for implementing a 3bit asynchronous counter with use of JK flip flop component. (Structural)

Write a testbench to verify your results.

Write a VHDL code for implementing a 4 bit up down synchronous counter using IC 74163

Write a testbench to verify your results.

Also, generate a programming file and download the code on CPLD kit and verify the results for synchronous counter

To study basic sequential statements of VHDL and to understand use of test bench for simulation.

COs to be achieved:

CO 1: Use basic Concurrent and Sequential statements in VHDL and write codes for simple applications

CO 2: Test a VHDL code and verify the circuit model.

CO 3: Synthesize and Implement the designed circuits on CPLD/ FPGA.

Work to be done

Upload codes for JK flip-flop, asynchronous counter and synchronous counter using 74163. Also upload test benches and simulation for the same.

Upload scanned copy of post lab questions

Code

IC74163:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity IC74163_Naik is
  port
      (
                        clk : in std_logic;
                        modeSelect : in std_logic_vector(3 downto 0);
                        load : in std_logic_vector(3 downto 0);
                        q : out std_logic_vector(3 downto 0)
                );
end IC74163_Naik;

architecture IC74163_Naik_arch of IC74163_Naik is
        signal qs :std_logic_vector(3 downto 0);
        begin
          process(clk,modeSelect,load)
                begin
                if (clk'event and clk='0') then
                  if(modeSelect(0) = '0') then
                        qs <= "0000";
                  elsif(modeSelect(1) = '0') then
                        qs <= load;
                  else
                        if(modeSelect = "1111") then
                                qs <= qs+1;
                        end if;
                  end if;
                end if;
          end process;
          q <= qs;
```

```
end IC74163_Naik_arch;
```

IC74163 Testbench:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity IC74163_Naik_tb is
end IC74163_Naik_tb;

architecture IC74163_Naik_tb_arch of IC74163_Naik_tb is

        component IC74163_Naik is
          port
      (
                        clk : in std_logic;
                        modeSelect : in std_logic_vector(3 downto 0);
                        load : in std_logic_vector(3 downto 0);
                        q : out std_logic_vector(3 downto 0)
                );
        end component;

        signal clk: std_logic;
        signal modeSelect, load, q: std_logic_vector(3 downto 0);

        begin
                IC74163: IC74163_Naik port map(clk, modeSelect, load, q);

                process begin
                        clk <= '0';
                        wait for 20ns;

                        clk <= '1';
                        wait for 20ns;
                end process;

                process begin
                        modeSelect <= "0000";
                        wait for 20ns;

                        load <= "0000";
                        modeSelect <= "1000";
                        wait for 20ns;
```

```
                    modeSelect <= "1111";
                    wait for 2000ns;
                end process;

end IC74163_Naik_tb_arch;
```

JK Flip Flop:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

 entity jk_naik is
   port (
       JK: IN STD_LOGIC_VECTOR(1 downto 0);
       clock: IN STD_LOGIC;
       reset: IN STD_LOGIC;
       q: out STD_LOGIC
    );
 end jk_naik;

 architecture jk_naik_arch of jk_naik is

 signal q_s  : std_logic := '0';

 begin
 process(reset,clock)
 begin
     if (reset = '1')then
         q_s <='0';
     elsif (clock'event and clock = '1')then
         case (JK) is
             when "00" => q_s <= q_s;
             when "01" => q_s <= '0';
             when "10" => q_s <= '1';
             when others => q_s <= not q_s;
         end case;
     end if;
      q <= q_s;
 end process;
 end jk_naik_arch;
```

3 bit Asynchronous Counter using JK Flip Flop:

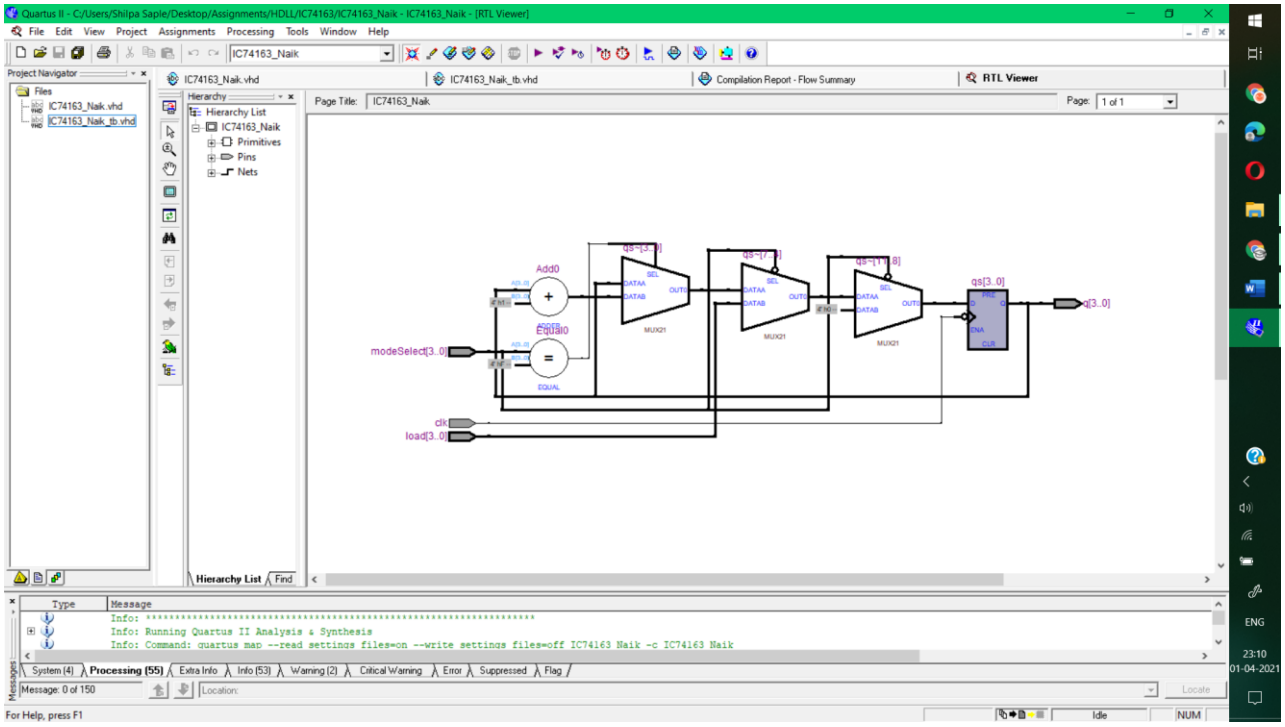
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity async_three_bit_naik is
port (
        clock : in std_logic;
        reset : in std_logic;
        count : out std_logic_vector(2 downto 0)
    );
 end async_three_bit_naik;

architecture async_three_bit_naik_arch of async_three_bit_naik is
component jk_naik is
   port (
        JK: IN STD_LOGIC_VECTOR(1 downto 0);
        clock: IN STD_LOGIC;
        reset: IN STD_LOGIC;
        q: out STD_LOGIC
    );
end component;

signal S:std_logic_vector(2 downto 0);
signal s1,s2:std_logic;
begin
            FF1:jk_naik port map("11",reset,clock,S(0));
            s1<=not S(0);
            FF2:jk_naik port map("11",reset,s1,S(1));
            s2<=not S(1);
            FF3:jk_naik port map("11",reset,s2,S(2));
            count<=S;
end async_three_bit_naik_arch;
```

3 bit Asynchronous Counter using JK Flip Flop Testbench:

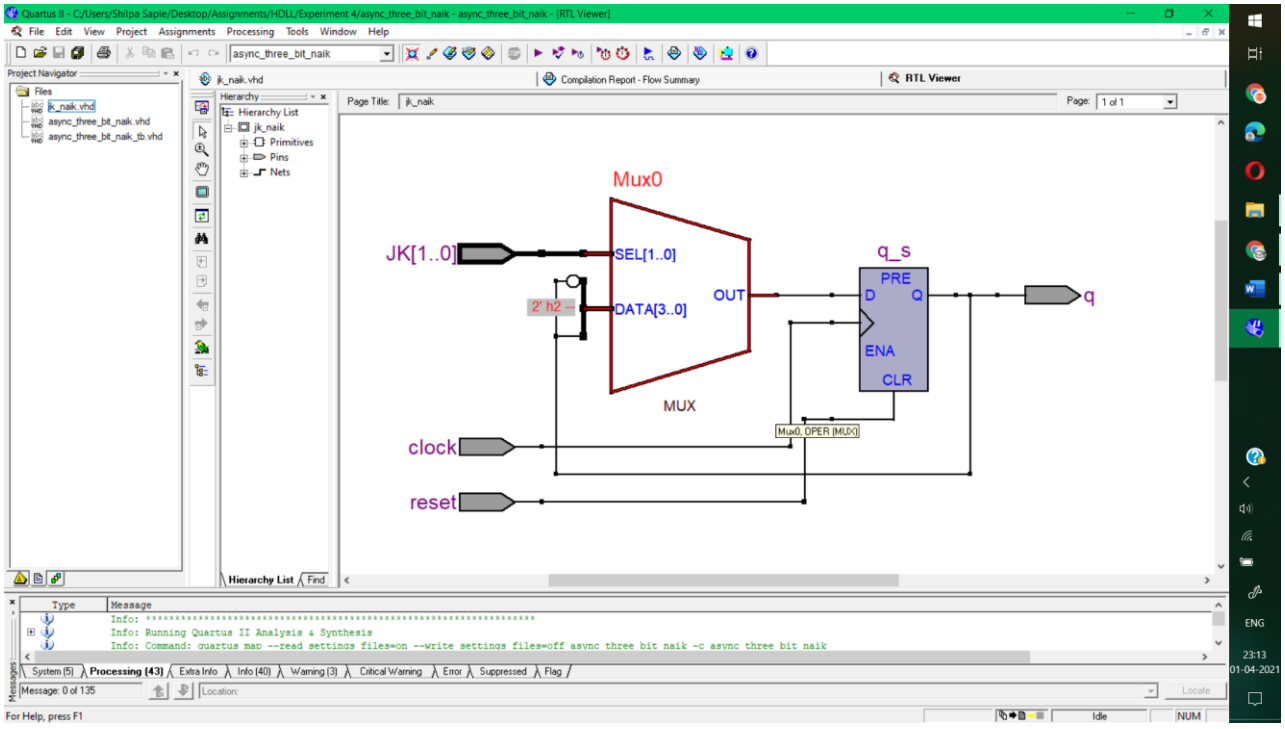
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity async_three_bit_naik_tb is
end async_three_bit_naik_tb;

architecture async_three_bit_naik_tb_arch of async_three_bit_naik_tb is
component async_three_bit_naik is
Port (
                clock : in std_logic;
      reset : in std_logic;
      count : out std_logic_vector(2 downto 0)
   );
end component;

signal reset,clock:std_logic;
signal count:std_logic_vector(2 downto 0);

begin
uut:async_three_bit_naik port map(reset,clock,count);
process
        begin
                clock<='1';
                wait for 10ns;
                clock<=not clock;
                wait for 10ns;
end process;

process
        begin
                reset<='1';wait for 20ns;
                reset<='0';wait for 200ns;
end process;
end async_three_bit_naik_tb_arch;
```

RTL View:

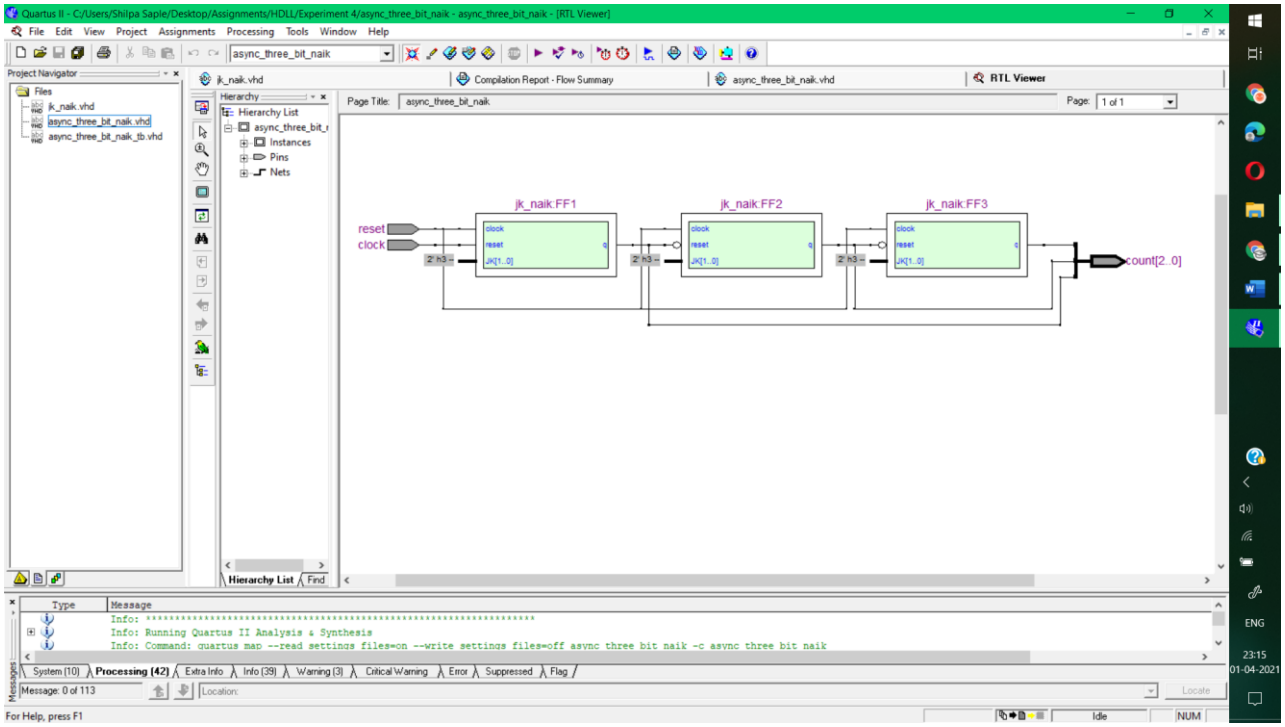
IC74163:



JK Flip Flop:

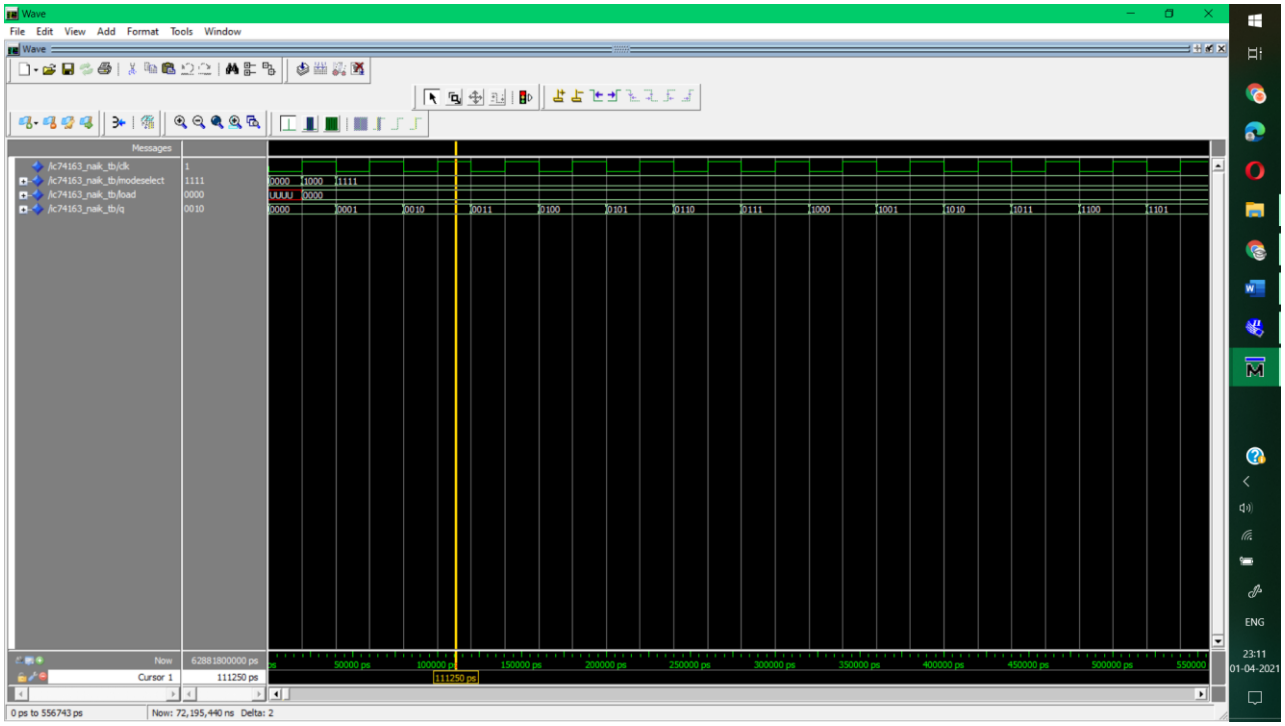


3 bit Asynchronous Counter using JK Flip Flop:

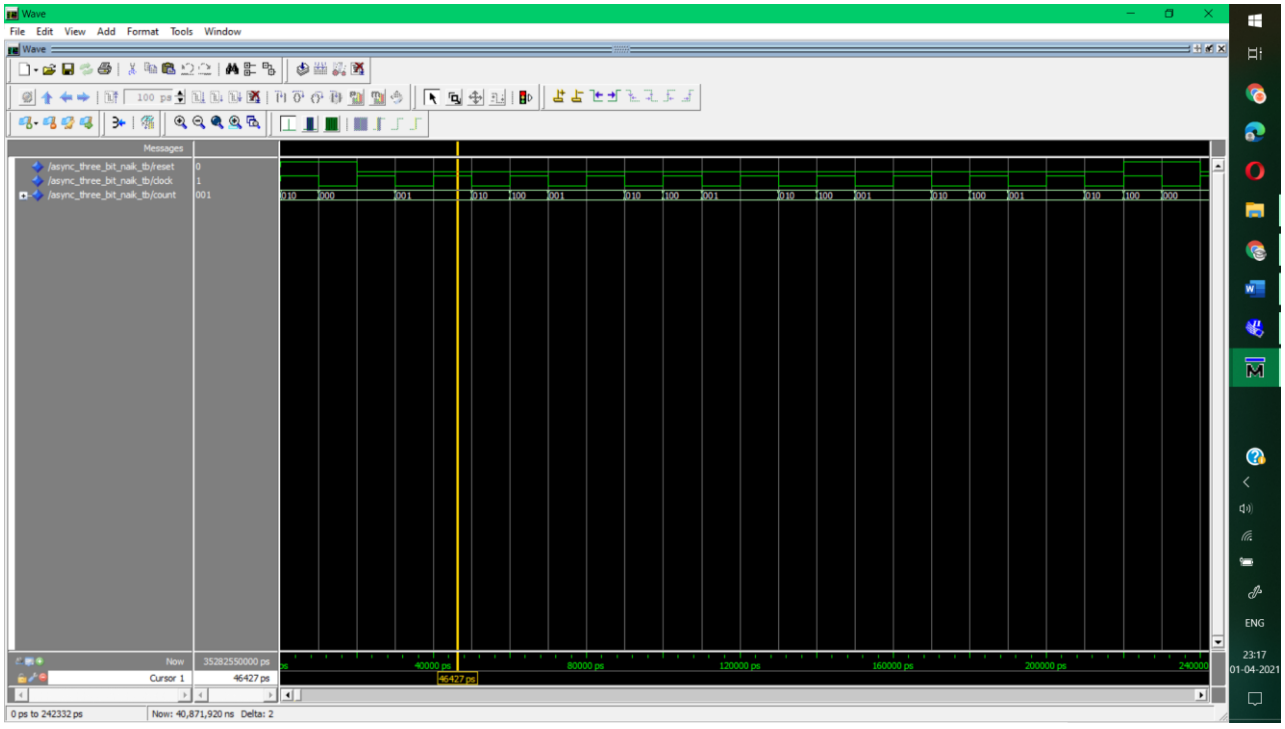


Output waveform:

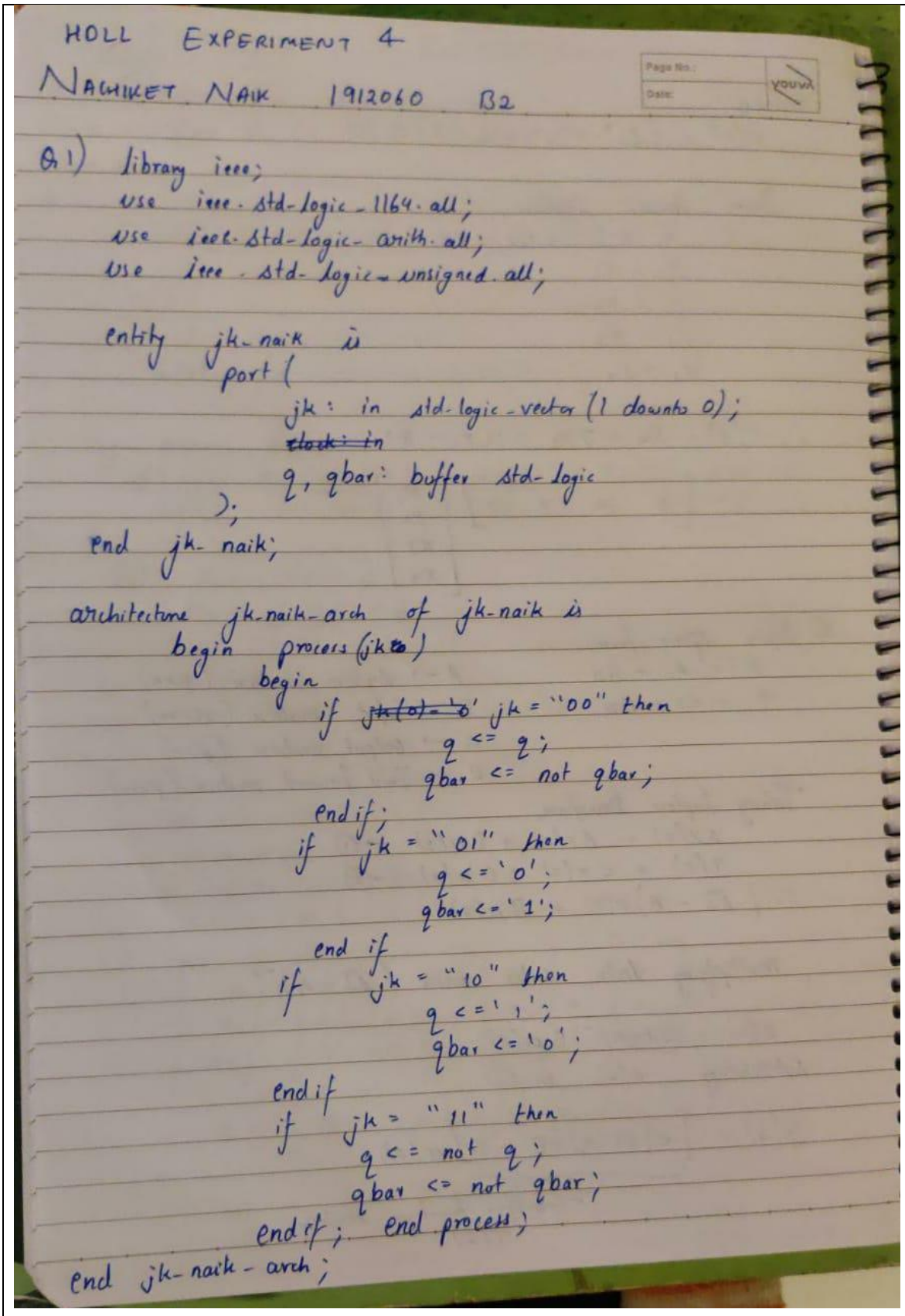
IC74163:



3 bit Asynchronous counter using JK Flipflop



Post Lab Subjective/Objective type Questions:
Upload Answer of following question before coming to next laboratory. **Q1. If synchronous preset and asynchronous clear inputs are to be added in the above flip-flops then how will you modify the code?**



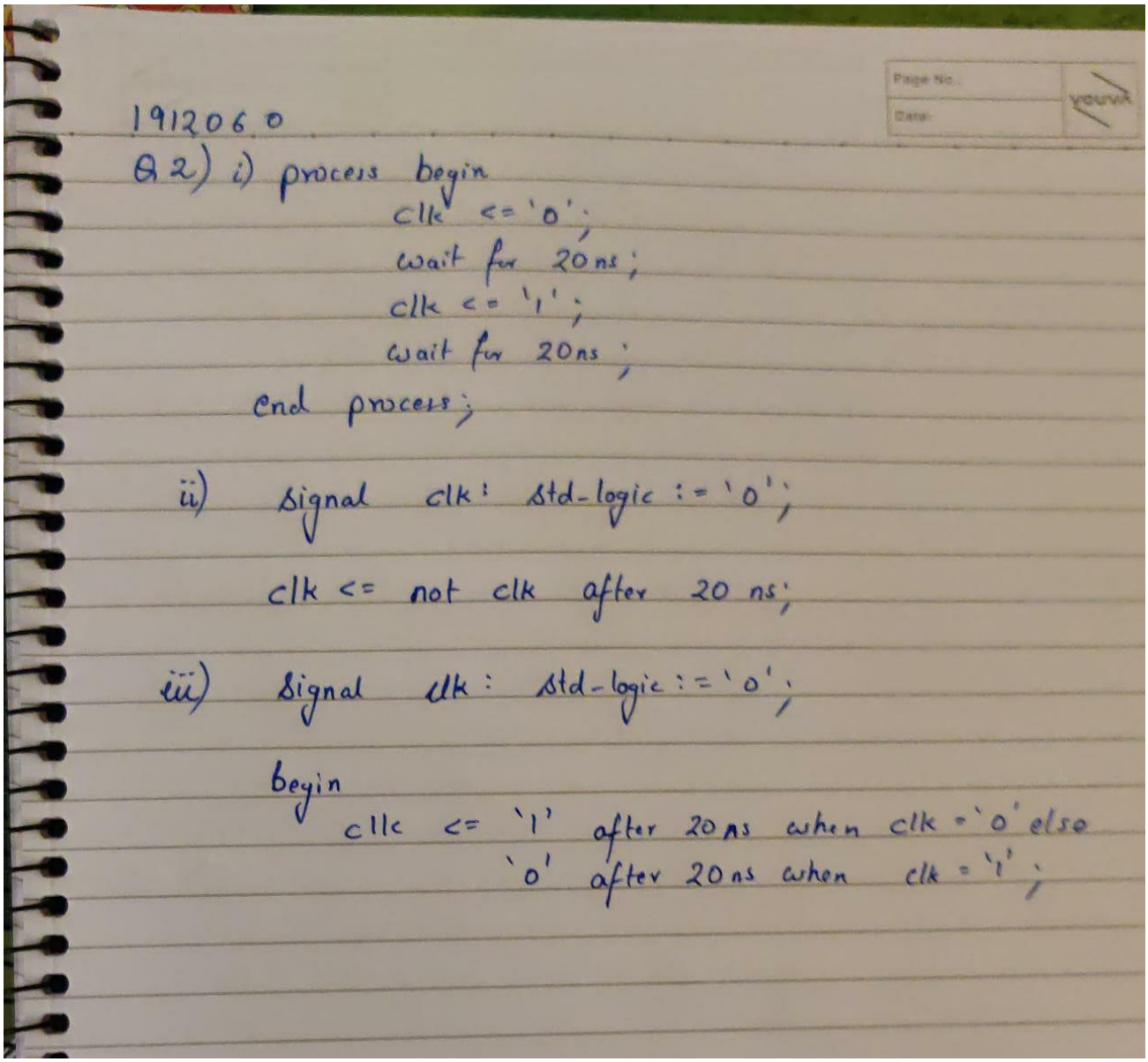
HDL EXPERIMENT 4

NACHIKET NAIK 1912060 B2

Q1)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity jk_naik is
port(
        jk: in std_logic_vector(1 downto 0);
        clock: in
        q, qbar: buffer std_logic
);
end jk_naik;

architecture jk_naik_arch of jk_naik is
begin process(jk
begin
        if jk(0)='0' jk = "00" then
                q <= q;
                qbar <= not qbar;
        end if;
        if jk = "01" then
                q <= '0';
                qbar <= '1';
        end if
        if jk = "10" then
                q <= '1';
                qbar <= '0';
        end if
        if jk = "11" then
                q <= not q;
                qbar <= not qbar;
        end if; end process;
end jk_naik_arch;
```

Q.2. What are the ways in which clock can be generated in test bench?



1912060

Q.2) i) process begin
clk <= '0';
wait for 20 ns;
clk <= '1';
wait for 20 ns;
end process;

ii) signal clk: std_logic := '0';

clk <= not clk after 20 ns;

iii) signal clk: std_logic := '0';

begin
clk <= '1' after 20 ns when clk = '0' else
'0' after 20 ns when clk = '1';

Conclusion:

In this experiment we have used behavioural architecture to make IC74163 and structural architecture to make a three bit asynchronous counter by using JK Flip Flop.

The IC74163 can have the following modes of operation:

Reset - Output is reset to "0000"

Load - Output is loaded with the given load input

Count - Output is incremented

No change

In the 3 bit counter the flipflops are being triggered with different clock signals, hence it is an asynchronous counter.

Signature of faculty in-charge with Date: