Class : BE 6

Roll. No : 42260

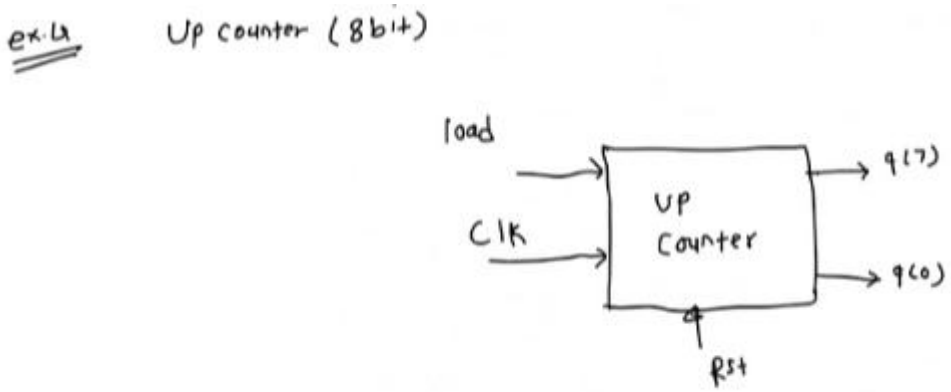
Assignment No. : 1

Assignment Name : 8 bit up counter

Date Of Performance : 13/9/21

Assignment: Write a VHDL Program to Model a 8 bit up counter. Synthesize the model for the target PLD. Simulate the model using a TestBench. Implement it on the target PLD

Block Diagram:



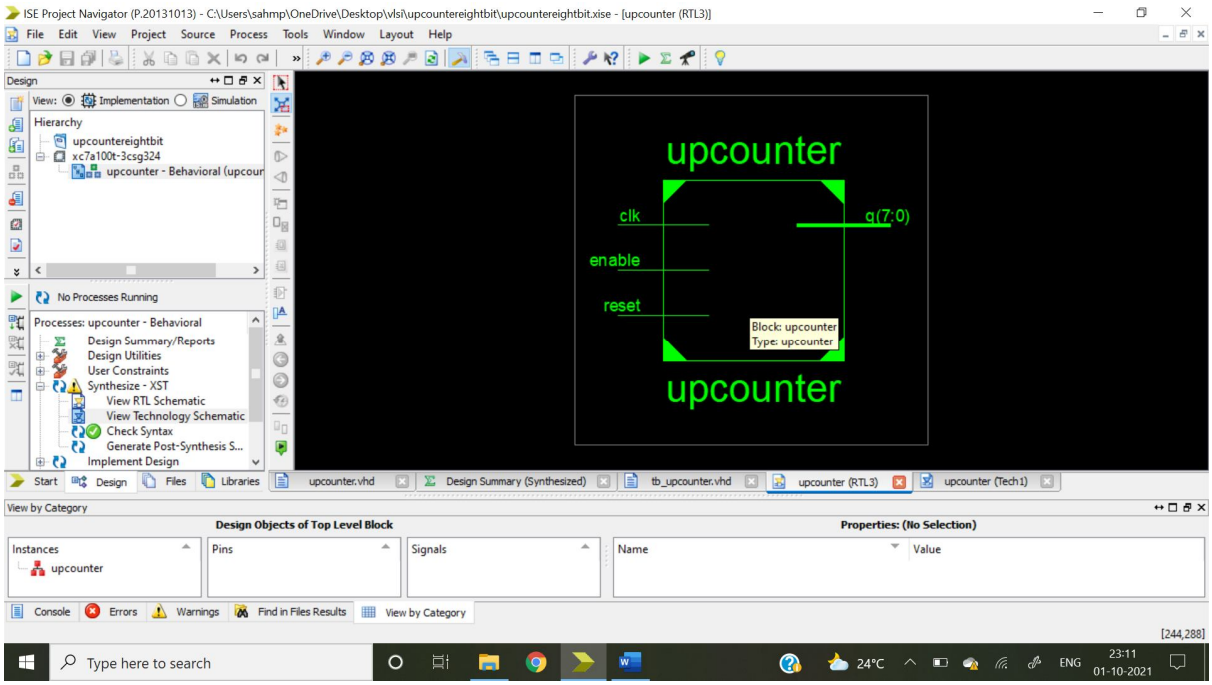

Main VHDL Program:

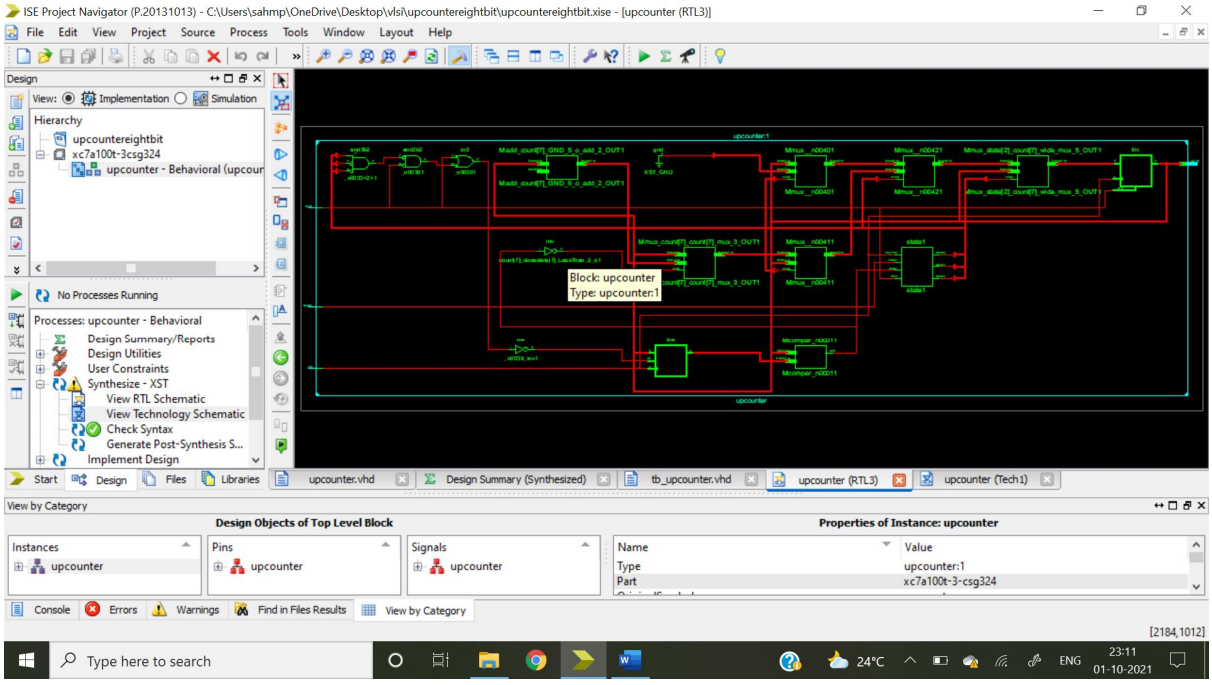
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
```

```
entity upcounter is
   Port ( enable : in  STD_LOGIC;
          clk : in  STD_LOGIC;
          reset : in  STD_LOGIC;
                                  q: out   STD_LOGIC_VECTOR (7 downto 0));
end upcounter;

architecture Behavioral of upcounter is
signal state : std_logic_vector(2 downto 0):="001";
signal count : std_logic_vector(7 downto 0):="00000000";
signal donestate : std_logic_vector(7 downto 0):="11111111";

constant initial : std_logic_vector(2 downto 0):="001"; --one hot encoding
constant compute : std_logic_vector(2 downto 0):="010";
constant done : std_logic_vector(2 downto 0):="100";

begin
process(clk,reset,enable)
begin
if reset='1' then
 count <= (others=> '0');
elsif(rising_edge(clk)) then
case state is
when initial =>
 count <= "00000000";
 if enable='1' then
 state <= compute;
 else
 state <= initial;
 end if;
```

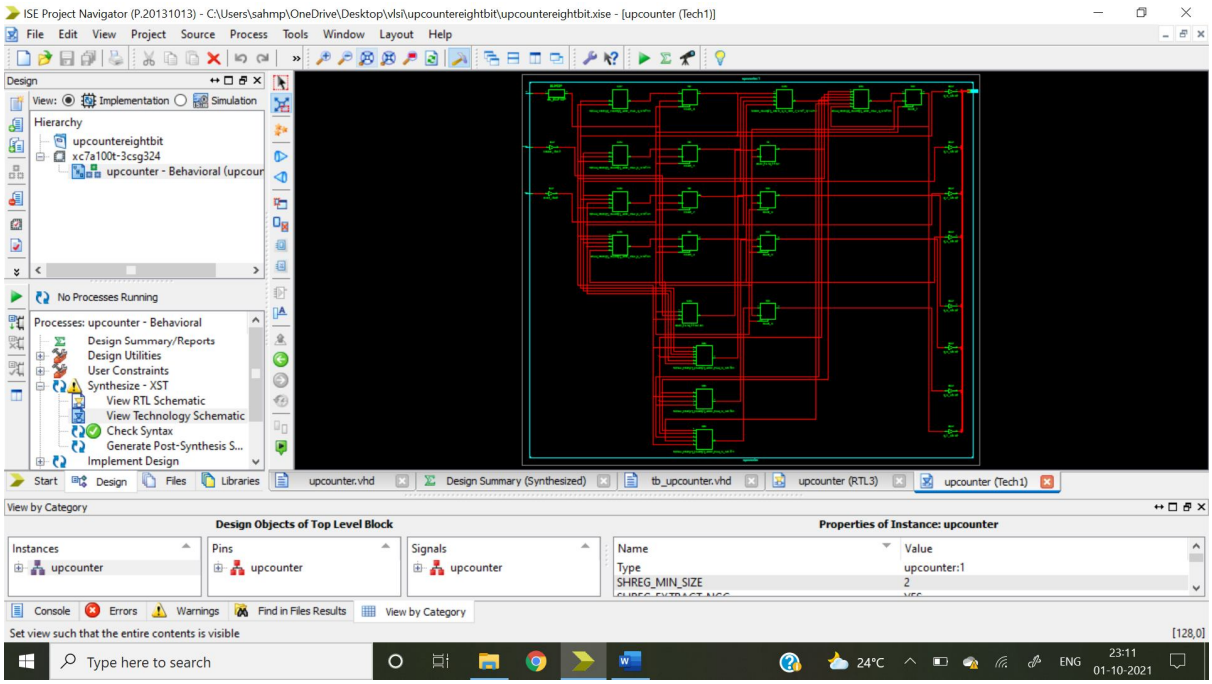
```
when compute =>
 if count <= donestate then
 count <= count + 1;
 state <= compute;
 else
 state <= done;
 end if;
when done =>
 donestate <= count;
 state <= initial;
when others =>
 state <= initial;
  end case;
 end if;
 end process;
q <= count;
end Behavioral;
```

## RTL Schematic:



Technology Schematic:




Synthesis Report

a) Device Utilisation Summary

Top Level Output File Name : upcounter.ngc

Primitive and Black Box Usage:

---

# BELS : 10
# LUT2 : 1
# LUT3 : 2
# LUT4 : 3
# LUT5 : 2
# LUT6 : 2
# FlipFlops/Latches : 9
# FD : 1
# FDC : 8
# Clock Buffers : 1
# BUFGP : 1
# IO Buffers : 10
# IBUF : 2
# OBUF : 8

Device utilization summary:

---

Selected Device : 7a100tcsg324-3

Slice Logic Utilization:
Number of Slice Registers: 9 out of 126800 0%
Number of Slice LUTs: 10 out of 63400 0%
Number used as Logic: 10 out of 63400 0%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 10
Number with an unused Flip Flop: 1 out of 10 10%
Number with an unused LUT: 0 out of 10 0%
Number of fully used LUT-FF pairs: 9 out of 10 90%
Number of unique control sets: 2

IO Utilization:
Number of IOs: 11
Number of bonded IOBs: 11 out of 210 5%

Specific Feature Utilization:
Number of BUFG/BUFGCTRLs: 1 out of 32 3%

b) Timing Report

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
---
| Clock Signal | Clock buffer(FF name) | Load |
|---|---|---|
| clk | BUFGP | 9 |

Asynchronous Control Signals Information:
---
No asynchronous control signals found in this design

Timing Summary:
---
Speed Grade: -3

Minimum period: 1.438ns (Maximum Frequency: 695.217MHz)
Minimum input arrival time before clock: 0.666ns
Maximum output required time after clock: 0.668ns
Maximum combinational path delay: No path found

Timing Details:

---

All values displayed in nanoseconds (ns)

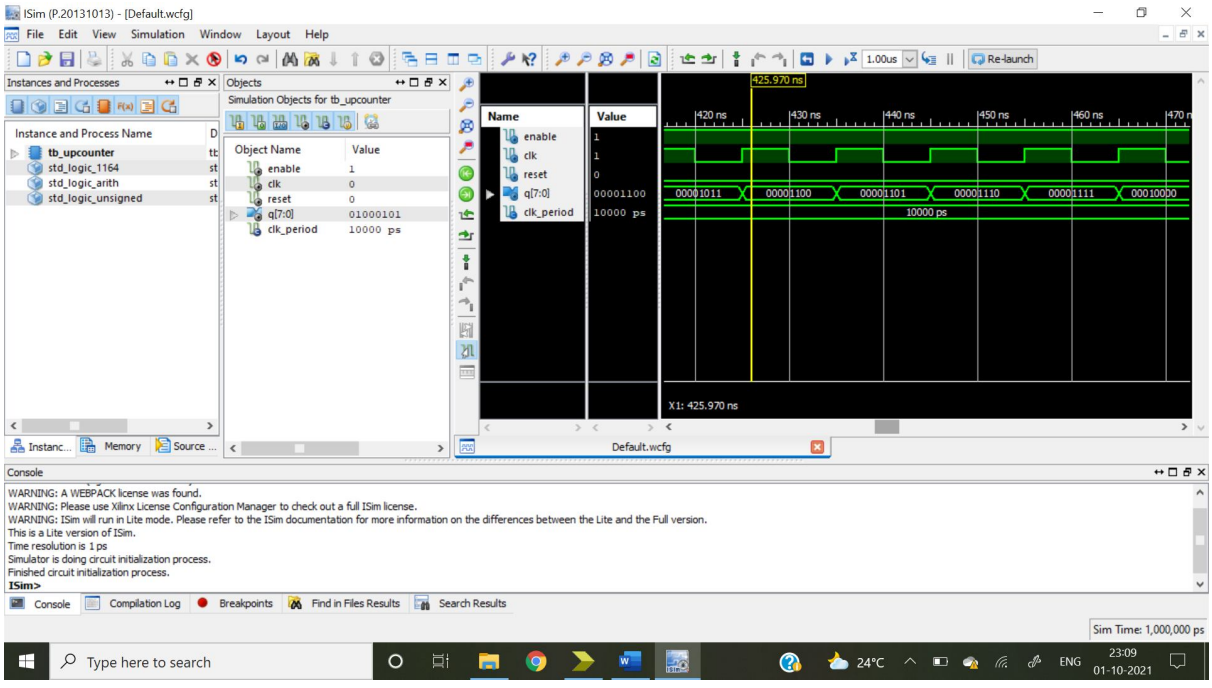
Testbench Program:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY tb_upcounter IS
END tb_upcounter;

ARCHITECTURE behavior OF tb_upcounter IS

   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT upcounter
   PORT(
        enable : IN  std_logic;
        clk : IN  std_logic;
        reset : IN  std_logic;
        q : OUT  std_logic_vector(7 downto 0)
       );
   END COMPONENT;


  --Inputs
  signal enable : std_logic := '0';
  signal clk : std_logic := '0';
  signal reset : std_logic := '0';
```

```
  signal q : std_logic_vector(7 downto 0) := (others => '0');

  -- Clock period definitions
  constant clk_period : time := 10 ns;

BEGIN

        -- Instantiate the Unit Under Test (UUT)
  uut: upcounter PORT MAP (
       enable => enable,
       clk => clk,
       reset => reset,
       q => q
      );

  -- Clock process definitions
  clk_process :process
  begin
                clk <= '0';
                wait for clk_period/2;
                clk <= '1';
                wait for clk_period/2;
  end process;


  -- Stimulus process
  stim_proc: process
  begin
    -- hold reset state for 100 ns.
    wait for 100 ns;
```

```
    --wait for clk_period*10;

    -- insert stimulus here
                reset <= '1';

                wait for 100 ns;
                reset <= '0';
                enable <= '0';
                wait for 100 ns;
                reset <= '0';
                enable <= '1';

    wait;
  end process;

END;
```

Isim Waveforms:



Conclusion:

Thus we have :

1) Modeled a 8 bit up counter using Behavioral Modeling Style.

2) Observed following Schematics : RTL & Technology Schematics generated Post-Synthesis.

3) Interpreted Device Utilisation Summary in terms of LUTs , SLICES , IOBs , Multiplexers & D FFs used out of the available device resources.

4) Interpreted the TIMING Report in terms of Maximum combinational delay as indicative of the Maximum Operating Frequency .

5) Written a TESTBENCH to verify the functionality of 8 bit up counter & verified the functionality as per the TRUTH-TABLE ,by observing ISIM Waveforms.

6) Prototyped the FPGA XC3S250EPQ208-5 to realize 8 bit up counter & verified its operation by giving suitable input combinations