Class : BE 6

Roll. No : 42260

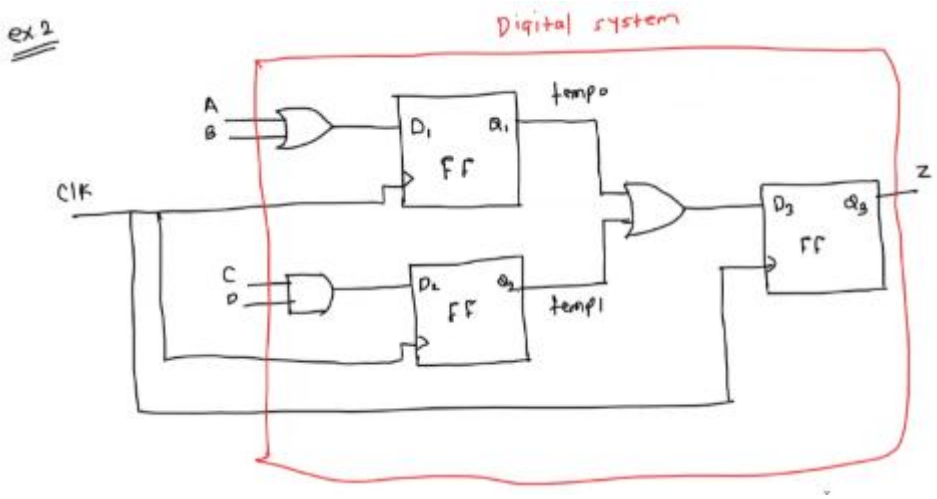
Assignment No. : 1

Assignment Name : Sequential Process Block

Date Of Performance :

Assignment: Write a VHDL Program to Model a Sequential Process Block. Synthesize the model for the target PLD. Simulate the model using a TestBench. Implement it on the target PLD

Block Diagram:

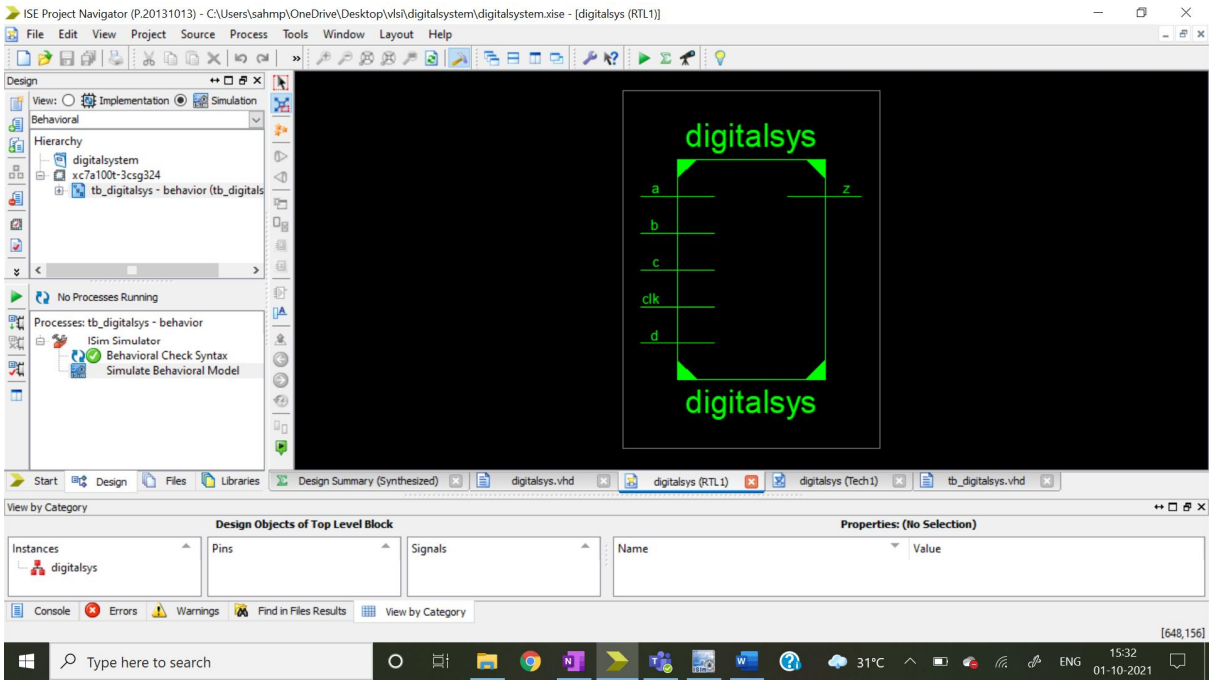


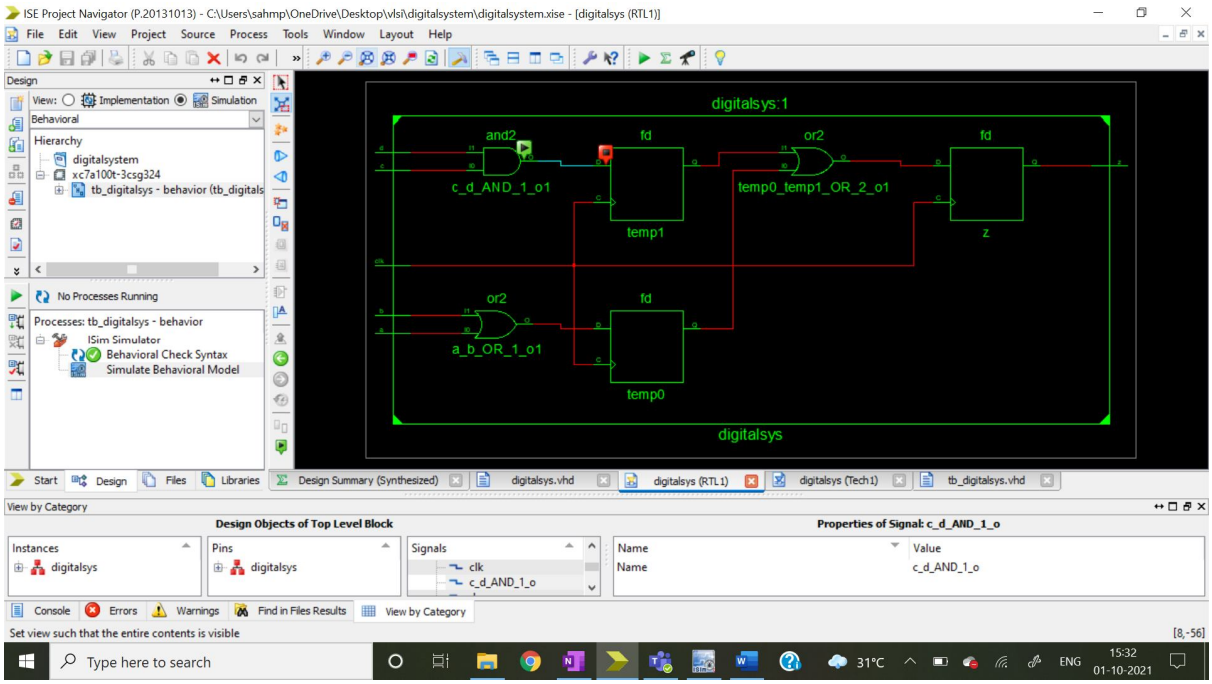
Main VHDL Program:

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


entity digitalsys is
   Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          c : in  STD_LOGIC;
          d : in  STD_LOGIC;
          clk : in  STD_LOGIC;
          z : out  STD_LOGIC);
end digitalsys;
```

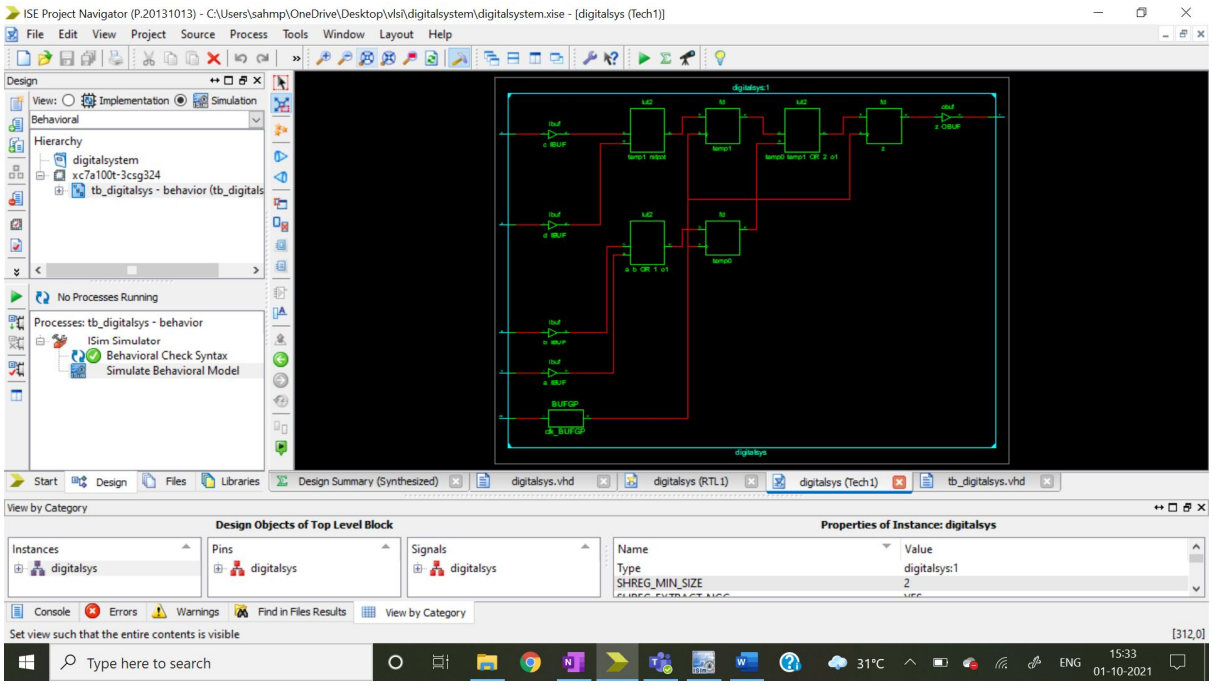
```
architecture Behavioral of digitalsys is
signal temp0,temp1: std_logic;
begin
process (clk)
        begin
           if(clk'event and clk='1') then --flipflop using <= for rising edge of clock
             temp0<= a or b;
              temp1<= c and d;
              z<= temp0 or temp1;
         end if;
       end process;
end Behavioral;
```

RTL Schematic:





Technology Schematic:



Synthesis Report

a) Device Utilisation Summary

Top Level Output File Name : digitalsys.ngc

Primitive and Black Box Usage:

BELS : 3

```
#      LUT2                 : 3
# FlipFlops/Latches         : 3
#      FD                   : 3
# Clock Buffers             : 1
#      BUFGP                : 1
# IO Buffers                : 5
#      IBUF                 : 4
#      OBUF                 : 1

Device utilization summary:
---------------------------

Selected Device : 7a100tcsg324-3


Slice Logic Utilization:
 Number of Slice Registers:            3  out of  126800     0%
 Number of Slice LUTs:                 3  out of  63400     0%
    Number used as Logic:              3  out of  63400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     6
   Number with an unused Flip Flop:      3  out of      6    50%
   Number with an unused LUT:            3  out of      6    50%
   Number of fully used LUT-FF pairs:    0  out of      6     0%
   Number of unique control sets:        1

IO Utilization:
 Number of IOs:                        6
 Number of bonded IOBs:                6  out of    210     2%
```

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                  | 3     |
-----------------------------------+------------------------+-------+
```

Asynchronous Control Signals Information:

```
----------------------------------------
```

No asynchronous control signals found in this design

Timing Summary:

```
---------------
```

Speed Grade: -3

Minimum period: 0.845ns (Maximum Frequency: 1183.012MHz)
Minimum input arrival time before clock: 0.485ns
Maximum output required time after clock: 0.640ns
Maximum combinational path delay: No path found

Timing Details:

```
---------------
```

All values displayed in nanoseconds (ns)

Testbench Program:

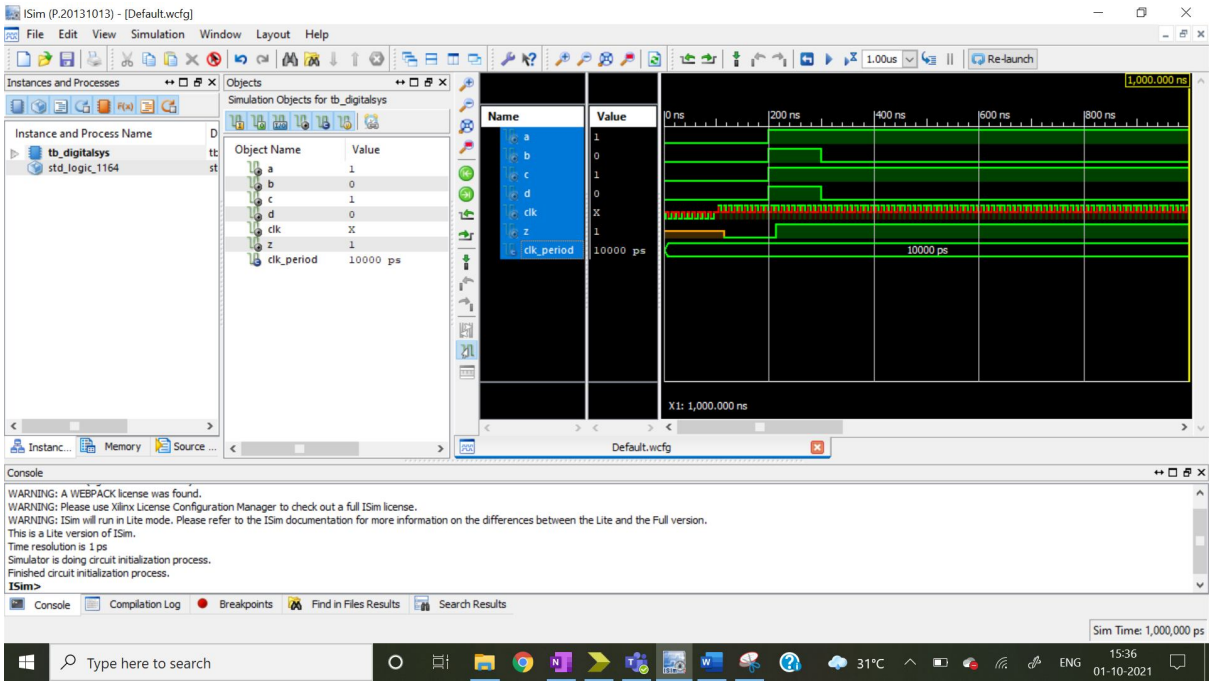
```
LIBRARY ieee;

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_digitalsys IS
END tb_digitalsys;

ARCHITECTURE behavior OF tb_digitalsys IS

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

   COMPONENT digitalsys
   PORT(
        a : IN  std_logic;
        b : IN  std_logic;
        c : IN  std_logic;
        d : IN  std_logic;
        clk : IN  std_logic;
        z : OUT  std_logic
       );
   END COMPONENT;


  --Inputs
  signal a : std_logic := '0';
  signal b : std_logic := '0';
  signal c : std_logic := '0';
  signal d : std_logic := '0';
  signal clk : std_logic := '0';

            --Outputs
  signal z : std_logic;

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

```
BEGIN

        -- Instantiate the Unit Under Test (UUT)
  uut: digitalsys PORT MAP (
       a => a,
       b => b,
       c => c,
       d => d,
       clk => clk,
       z => z
      );

  -- 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
                    a<='0';
                    b<='0';
                    c<='0';
                    d<='0';
                    clk<='1';
                    wait for 100ns;
                    a<='1';
                    b<='1';
                    c<='1';
                    d<='1';
                    clk<='1';
                    wait for 100ns;
                    a<='1';
                    b<='0';
                    c<='1';
                    d<='0';
                    clk<='1';
                    wait for 100ns;
     wait;
  end process;

END;
```

Isim Waveforms:



Conclusion:

Thus we have :

1) Modeled a Sequential Process Block 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 Sequential Process Block & verified the functionality as per the TRUTH-TABLE ,by observing ISIM Waveforms.

6) Prototyped the FPGA XC3S250EPQ208-5 to realize 4:1 Sequential Process Block & verified its operation by giving suitable input combinations