Department of Electronics & Telecommunication Engineering

INDEX

Subject: VLSI DEVICES & TECHNOLOGY (404181)

ROLL NO.: 42247 DIV. 6 YEAR: 2021-2022 SEMESTER: 7

List of Laboratory Experiments

S. No.	Title of Experiment	Performance Date	Submission Date	Sign
	PART-A To write VHDL code, simulate with test bench, synthesis, implement on PLD			
A.1	4 bit ALU for add, subtract, AND, NAND, XOR, XNOR, OR, & ALU pass.	14/09/2021	21/09/2021	
A.2	Universal Shift register with mode selection input for SISO, SIPO, PISO, & PIPO modes	21/09/2021	28/09/2021	
A.3	FIFO memory (4 X 8)	05/10/2021	12/10/2021	
A.4	LCD interface	12/10/2021	19/10/2021	
	PART-B To prepare CMOS layout in selected technology, simulate with and without capacitive load, comment on rise, and fall times.			
B.1.a	INVERTER & Analyses	09/11/2021	16/11/2021	
B.1.b	NAND – AND , NOR – OR Logic Gates	09/11/2021	16/11/2021	
B.1.c	TRANSMISSION GATE (TG)	16/11/2021	23/11/2021	
B.2	2:1 MUX Using TG	16/11/2021	23/11/2021	
B.1.d	HALF-ADDER (Using MUX , MUX Using TG)	16/11/2021	23/11/2021	
B.3.a	1-bit SRAM Cell Using NMOS Switch	23/11/2021	30/11/2021	
B.3.b	1-bit SRAM Cell Using TG Switch	23/11/2021	30/11/2021	

This is to certify that Shri/Kum Makarand Milind Pundlik has carried out the above mentioned 11 experiments in VLSI Devices & Technology laboratory of the institute.

For PUNE INSTITUTE OF COMPUTER TECHNOLOGY,
Pune: 43

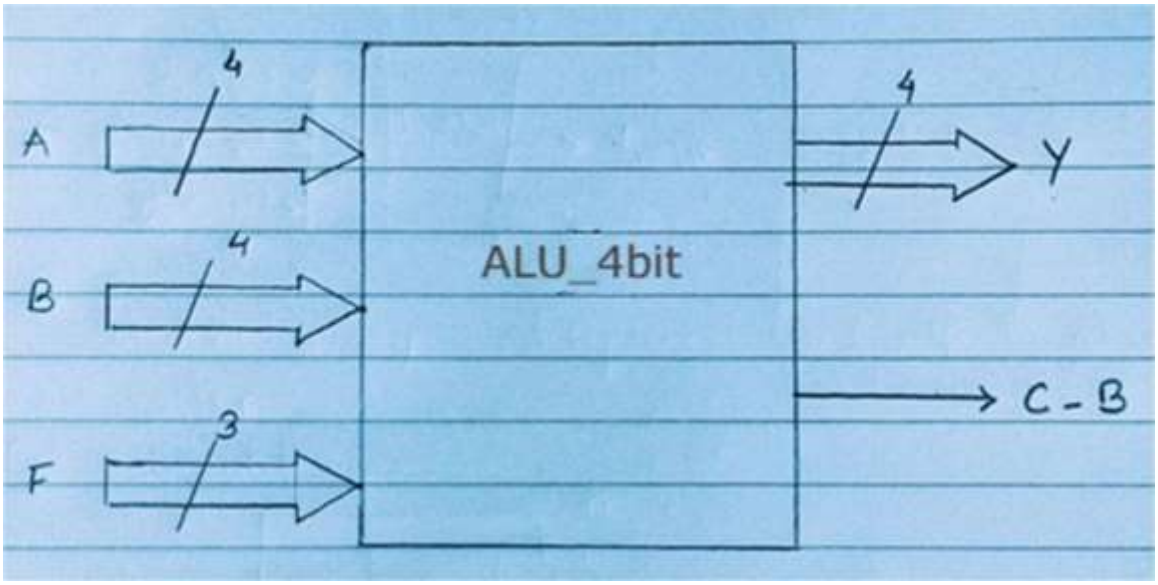
Date: 16/12/2021
PRIINCIPAL

} Staff

In charge

Class	:	BE-6
Roll. No	:	42247
Assignment No.	:	A.1
Assignment Name	:	4-Bit ALU
Date Of Performance	:	18/08/2021

BLOCK DIAGRAM


A
4
B
4
F
3
ALU_4bit
4
Y
C - B

FUNCTION TABLE

F			Y	C_B
F(2)	F(1)	F(0)		
0	0	0	A.B	x
0	0	1	$\overline{A.B}$	x
0	1	0	A+B	x
0	1	1	$A \oplus B$	x
1	0	0	$\overline{A \odot B}$	x
1	0	1	$\overline{A+B}$	x
1	1	0	$A + B$	CARRY
1	1	1	$A - B$	BORROW

MAIN VHDL MODEL (MVM)

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

entity ALU_4bit is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
           F : in  STD_LOGIC_VECTOR (2 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0);
           C_B : out  STD_LOGIC
             );
end ALU_4bit;

architecture ALU_4bit_arch of ALU_4bit is
```

```
signal result:STD_LOGIC_VECTOR(4  downto 0):="00000";
```

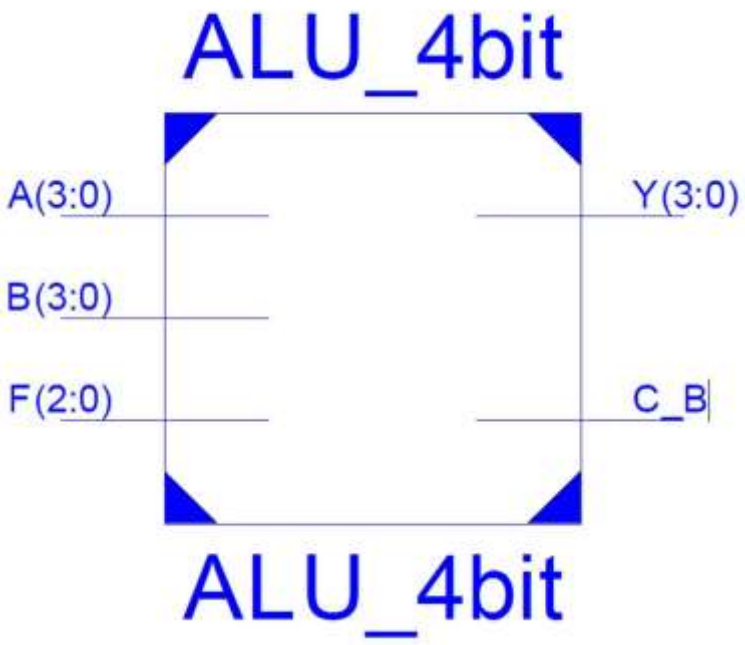
```
begin

process(A,B,F)
```

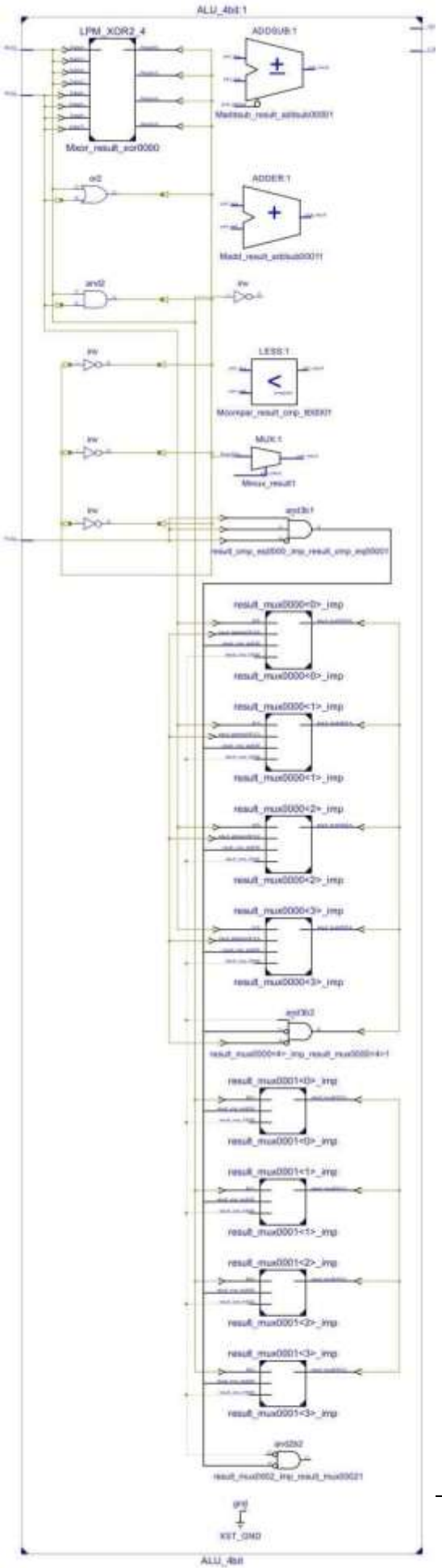
begin

```
    CASE F IS

        when "000" =>
            result <= '0' & (A AND B);

        when "001" =>           result <= '0' & (A NAND B);

        when "010" =>           result <= '0' & (A OR B);

        when "011" =>           result <= '0' & (A XOR B);

        when "100" =>           result <= '0' & (A XNOR B);

        when "101" =>           result <= '0' & (A NOR B);

        when "110" =>           result <= ('0' & A)+('0' & B);
       when others =>           if A < B then               result <= '0' & (NOT B);            result <= result+1;
result <= ('0' & A) + result;           result <= (NOT result) +1;
              result <= (NOT(('0' & A) + ('0' &(NOT B)) + 1))+1;          else
              result <=('0' & A)-('0' & B);
           end if ;

    end CASE;

end process;

Y <= result(3 downto 0);
C_B <= result(4);
```

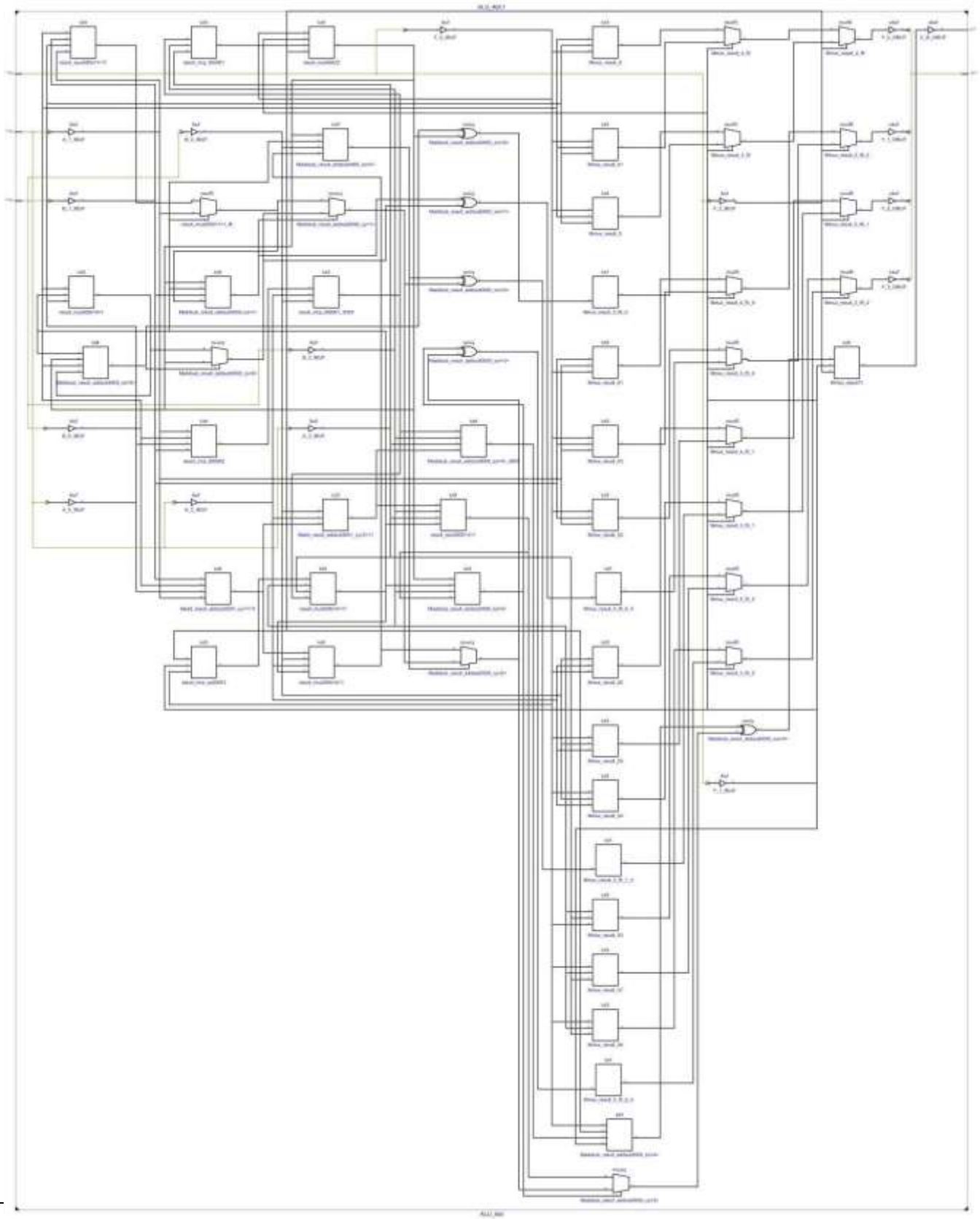
```
end ALU_4bit_arch;
```

RTL SCHEMATIC:


ALU_4bit
A(3:0)
B(3:0)
F(2:0)
Y(3:0)
C_B
ALU_4bit


ALU_4bit:1
LPM_XOR2_4
Mxor_result_xor0000
ADDSUB:1
Madd:sub_result_addsub0001
ADDER:1
Madd_result_addsub0001
LESS:1
Mcompar_result_cmp_lt0001
MUX:1
Mmux_result1
result_mux0000<0>_imp
result_mux0000<1>_imp
result_mux0000<2>_imp
result_mux0000<3>_imp
result_mux0001<0>_imp
result_mux0001<1>_imp
result_mux0001<2>_imp
result_mux0001<3>_imp
result_mux0002_imp_result_mux0002
KST_GND
ALU_4bit

TECHNOLOGY SCHEMATIC



SYNTHESIS REPORT

a) Device Utilization Summary:

==========

* Final Report *

Final Results

RTL Top Level Output File Name : ALU_4bit.ngr

Top Level Output File Name : ALU_4bit

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : No

Design Statistics

IOs : 16

Cell Usage :

BELS : 57

LUT1 : 4

LUT3 : 18

LUT4 : 13

MUXCY : 4

MUXF5 : 9

MUXF6 : 4

XORCY : 5

IO Buffers : 16 # IBUF : 11 # OBUF :

========== Device utilization summary:

Selected Device : 3s250epq208-5

Number of Slices:	19	out of 2448	0%
Number of 4 input LUTs:	35	out of 4896	0%
Number of IOs:	16		
Number of bonded IOBs:	16	out of 158	10%

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:

No clock signals found in this design Asynchronous Control Signals Information:

No asynchronous control signals found in this design Timing Summary:

Speed Grade: -5

Minimum period: No path found

Minimum input arrival time before clock: No path found

Maximum output required time after clock: No path found

Maximum combinational path delay: 13.714ns

Timing Detail:

All values displayed in nanoseconds (ns)

TESTBENCH VHDL MODEL (TVM)

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY ALU_4bit_tb IS
END ALU_4bit_tb;

ARCHITECTURE behavior OF ALU_4bit_tb IS

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

   COMPONENT ALU_4bit
   PORT(
A: IN  std_logic_vector(3 downto 0);
B: IN  std_logic_vector(3 downto 0);
       F : IN  std_logic_vector(2 downto 0);
       Y : OUT  std_logic_vector(3 downto 0);
                              C_B : OUT  std_logic
      );
   END COMPONENT;


  --Inputs
  signal A : std_logic_vector(3 downto 0) := "0010";    signal B : std_logic_vector(3 downto 0) :=
"1111";    signal F : std_logic_vector(2 downto 0) := (others => '1');

          --Outputs
  signal Y : std_logic_vector(3 downto 0);
  signal C_B : std_logic;
  -- No clocks detected in port list. Replace <clock> below with
  -- appropriate port name

BEGIN
```

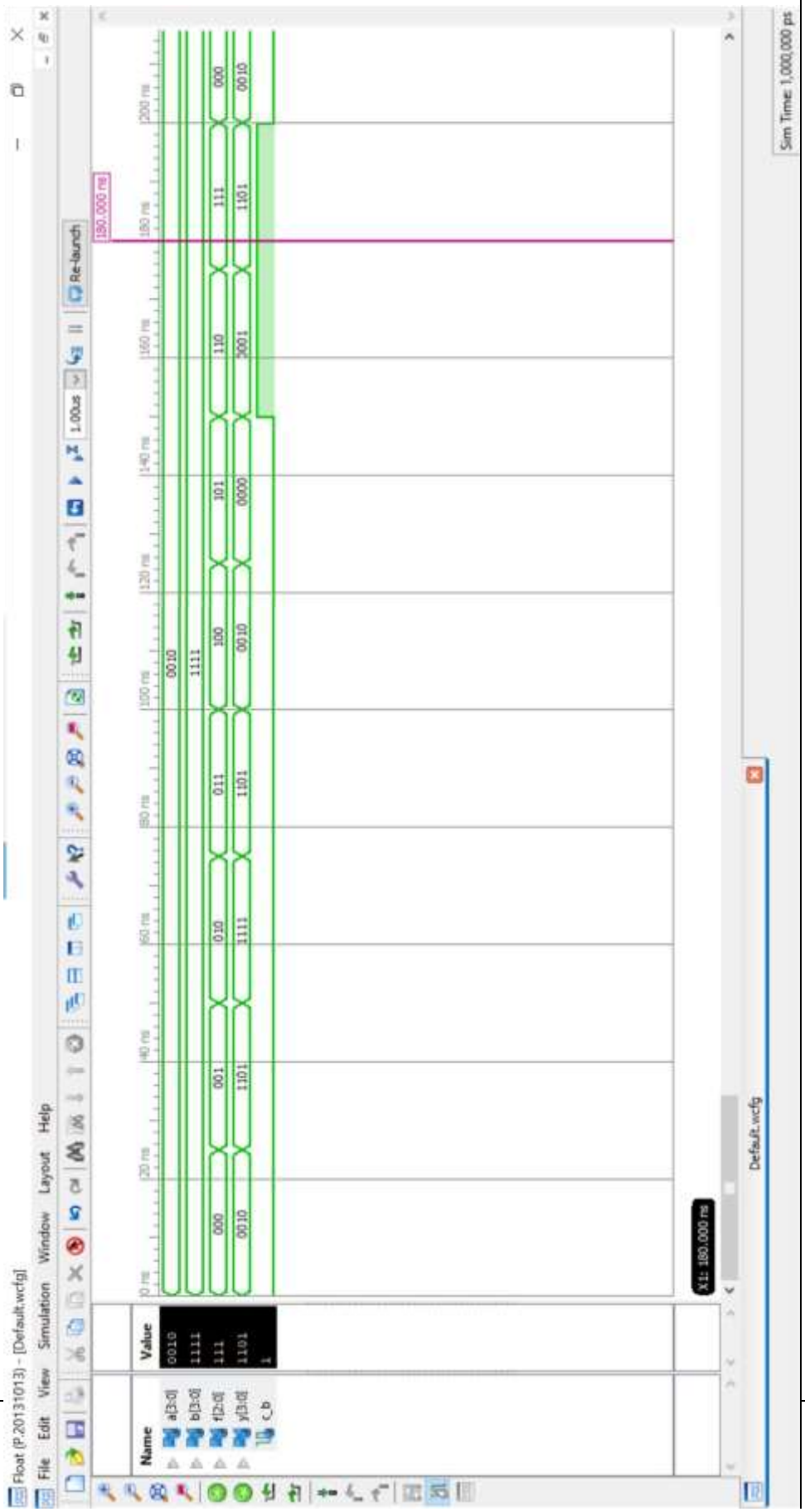
```
-- Instantiate the Unit Under Test (UUT)
```

```
  uut: ALU_4bit PORT MAP (
A => A,
B => B,
        F => F,
        Y => Y,
        C_B => C_B
       );

  -- Stimulus process    stim_proc_F: process
  begin
```

```
F <= F + 1;
```

```
    wait for 25 ns;

  end process;

END;
```

ISIM WAVEFORMS



PIN-LOCKING REPORT

PlanAhead Generated physical constraints

NET "A[3]" LOC = P205;

NET "A[2]" LOC = P206;

NET "A[1]" LOC = P203;

NET "A[0]" LOC = P200;

NET "B[3]" LOC = P192;

NET "B[2]" LOC = P193;

NET "B[1]" LOC = P189;

NET "B[0]" LOC = P190;

NET "F[2]" LOC = P179;

NET "F[1]" LOC = P180;

NET "F[0]" LOC = P177;

NET "Y[3]" LOC = P165;

NET "Y[2]" LOC = P167;

NET "Y[1]" LOC = P163;

NET "Y[0]" LOC = P164;

NET "C_B" LOC = P153;

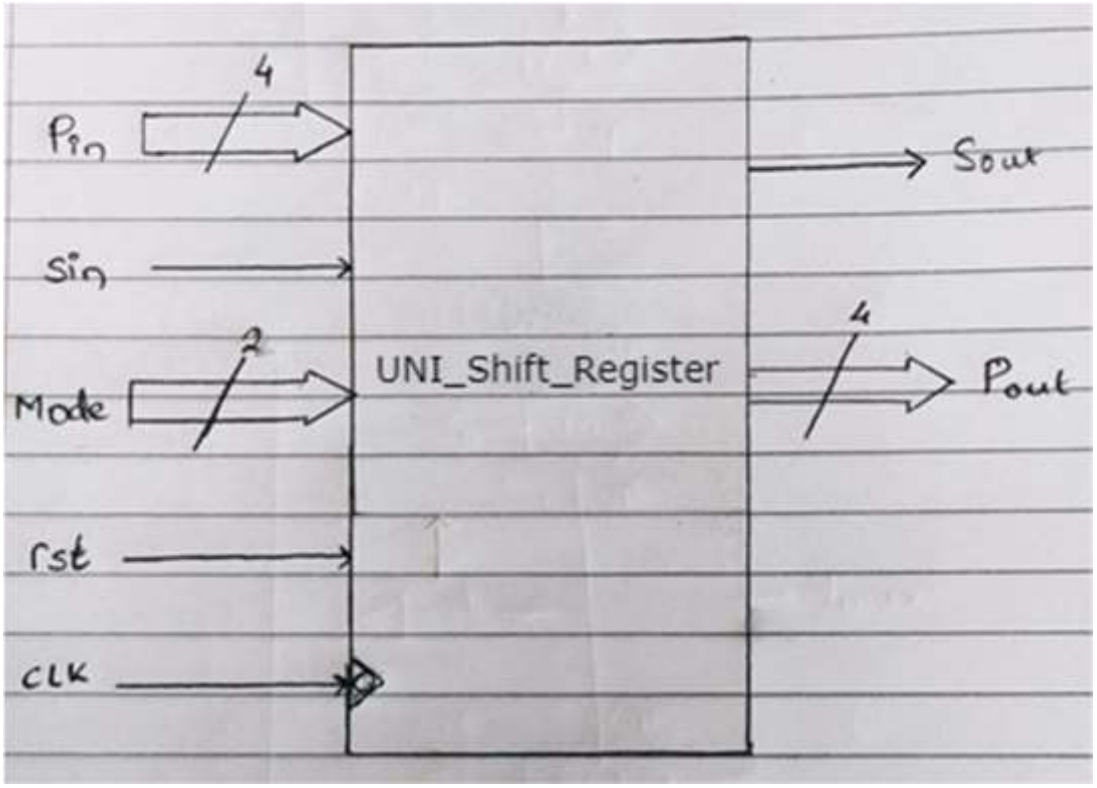
CONCLUSION

Thus, we have:

1) Modeled a 4-Bit ALU using Behavioral Modeling Style.
2) Observed following Schematics: RTL & Technology Schematics generated Post-Synthesis.
3) Interpreted Device Utilization 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, setup time, hold time.
5) Written a TESTBENCH to verify the functionality of 4-Bit ALU & verified the functionality as per the FUNCTION-TABLE, by observing ISIM Waveforms.
6) Used PlanAhead Editor for pin-locking.
7) Prototyped the FPGA XC3S250EPQ208-5 to realize 4-Bit ALU & verified its operation by giving suitable input combinations.

Class	:	BE-6
Roll. No	:	42247
Assignment No.	:	A.2
Assignment Name	:	Universal Shift Register
Date Of Performance	:	08/09/2021

BLOCK DIAGRAM


4
Pin
Sout
Sin
2
UNI_Shift_Register
4
Mode
Pout
rst
clk

FUNCTION TABLE

rst	clk	mode		Outputs
		mode(1)	mode(0)	
1	x	x	x	Sout = 0 , Pout = “0000”
0	↓	0	0	Serial In Serial Out (SISO)
0	↓	0	1	Serial In Parallel Out (SIPO)
0	↓	1	0	Parallel In Serial Out (PISO)
0	↓	1	1	Parallel In Parallel Out (PIPO)

MAIN VHDL PROGRAM (MVM)

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

entity UNI_Shift_Register is     Port ( rst : in  STD_LOGIC;          clk : in  STD_LOGIC;          Sin : in
STD_LOGIC;
        mode : in  STD_LOGIC_VECTOR (1 downto 0);
        Pin : in  STD_LOGIC_VECTOR (3 downto 0);
        Sout : out  STD_LOGIC;
        Pout : out  STD_LOGIC_VECTOR (3 downto 0)
                                );
end UNI_Shift_Register;
```

```
architecture UNI_Shift_Register_arch of UNI_Shift_Register is SIGNAL temp : STD_LOGIC_VECTOR (3
downto 0):="0000";
```

begin

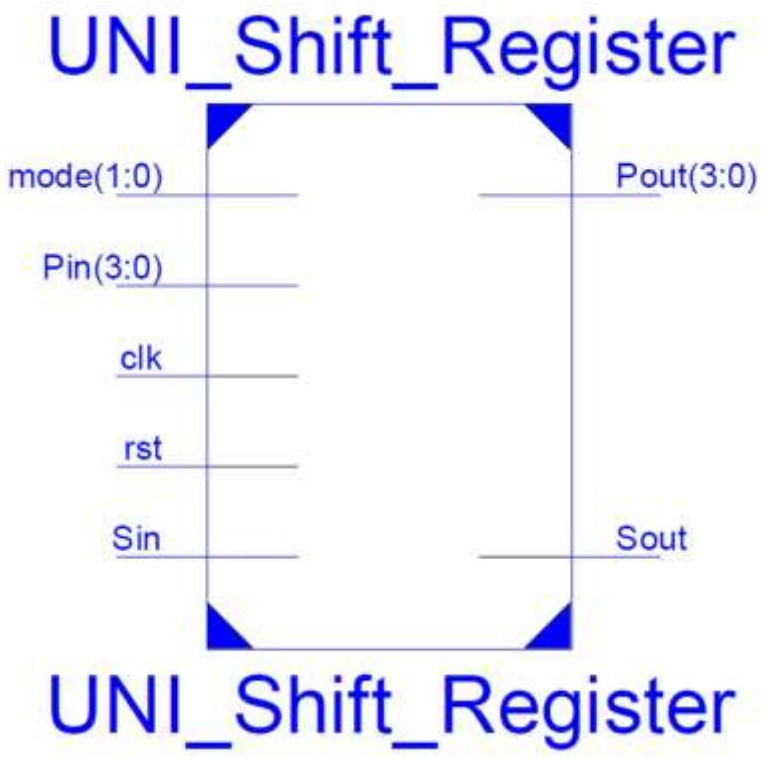
```
PROCESS(rst, clk, mode, Sin, Pin)
BEGIN
        IF rst = '1' THEN
                Pout <= "0000";
                Sout <= '0';

        ELSIF FALLING_EDGE(clk) THEN

                CASE mode IS

                        WHEN "00" =>
```

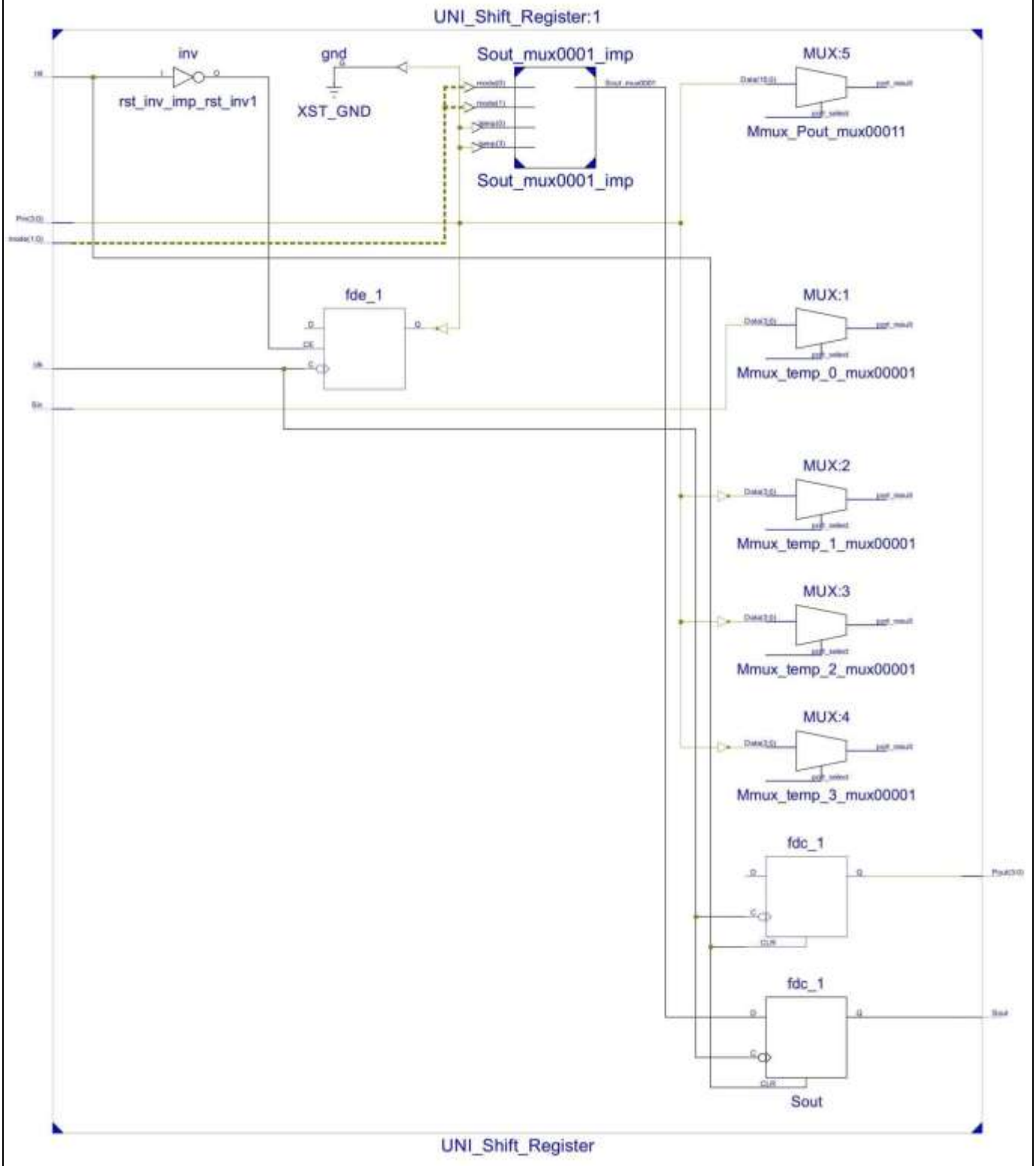
```
temp(3 downto 1) <= temp(2 downto 0);
```

```
                    temp(0) <= Sin;

                        Sout <= temp(3);
                        Pout <= "0000";

                    WHEN "01" =>
                        temp(3 downto 1) <= temp(2 downto 0);
                    temp(0) <= Sin;
    Pout <= temp;
                        Sout <= '0';

                    WHEN "10" =>
    temp <= Pin;                                    Sout <= temp(3);
temp(3 downto 1) <= temp(2 downto 0);
                        Pout <= "0000";

                    WHEN OTHERS =>
                        Pout <= Pin;
                        Sout <= '0';

                END CASE;
            END IF;
        END PROCESS;
```

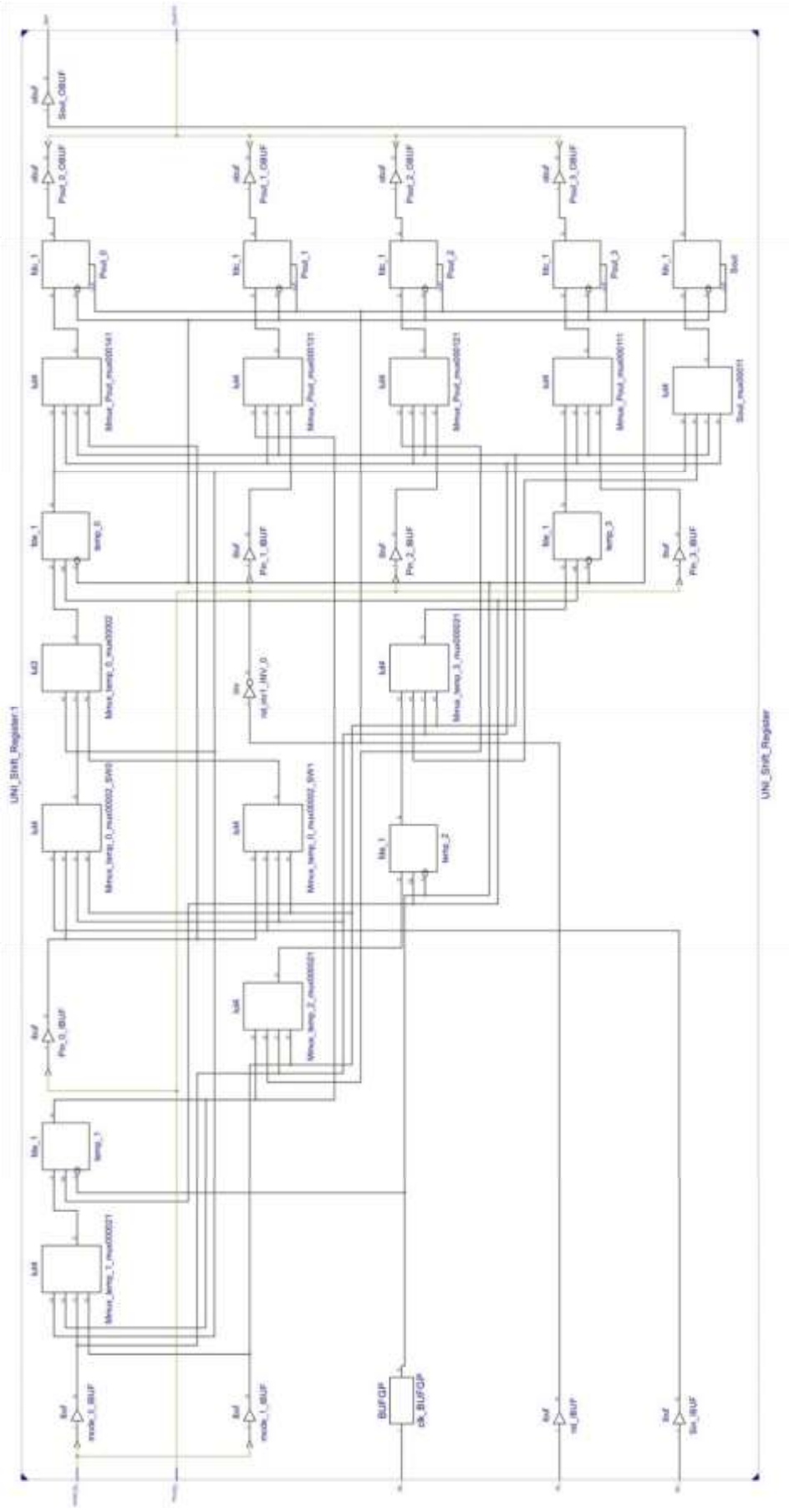
```
end UNI_Shift_Register_arch;
```

RTL SCHEMATIC:


UNI_Shift_Register
mode(1:0)
Pout(3:0)
Pin(3:0)
clk
rst
Sin
Sout
UNI_Shift_Register


UNI_Shift_Register:1
inv
rst_inv_imp_rst_inv1
gnd
XST_GND
Sout_mux0001_imp
Sout_mux0001_imp
MUX:5
Mmux_Pout_mux00011
fde_1
MUX:1
Mmux_temp_0_mux00001
MUX:2
Mmux_temp_1_mux00001
MUX:3
Mmux_temp_2_mux00001
MUX:4
Mmux_temp_3_mux00001
fdc_1
fdc_1
Sout
UNI_Shift_Register

TECHNOLOGY SCHEMATIC



SYNTHESIS REPORT

a) Device Utilization Summary:

==========

* Final Report *

==========

Final Results

RTL Top Level Output File Name : UNI_Shift_Register.ngr

Top Level Output File Name : UNI_Shift_Register

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : No

Design Statistics

IOs : 14

Cell Usage :

BELS : 12

INV : 1

LUT3 : 1

LUT4 : 10

FlipFlops/Latches : 9

FDC_1 : 5

FDE_1 : 4

Clock Buffers : 1

BUFGP : 1

IO Buffers : 13

```
#    IBUF                 : 8
```

OBUF : 5

========== Device

utilization summary:

Selected Device : 3s250epq208-5

Number of Slices: 6 out of 2448 0%

Number of Slice Flip Flops: 9 out of 4896 0%

Number of 4 input LUTs: 12 out of 4896 0%

Number of IOs: 14

Number of bonded IOBs: 14 out of 158 8% Number of GCLKs: 1 out of 24 4%

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:

Control Signal	Buffer(FF name)	Load
rst	IBUF	5

Timing Summary:

Speed Grade: -5

Minimum period: 1.923ns (Maximum Frequency: 520.034MHz)

Minimum input arrival time before clock: 4.009ns

Maximum output required time after clock: 4.040ns

Maximum combinational path delay: No path found

Timing Detail:

All values displayed in nanoseconds (ns)

TESTBENCH PROGRAM (TVM)

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL;

ENTITY UNI_Shift_Register_tb IS
END UNI_Shift_Register_tb;

ARCHITECTURE behavior OF UNI_Shift_Register_tb IS

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

   COMPONENT UNI_Shift_Register
   PORT(         rst : IN  std_logic;         clk : IN  std_logic;
         mode : IN  std_logic_vector(1 downto 0);
         Sin : IN  std_logic;
         Pin : IN  std_logic_vector(3 downto 0);
         Sout : OUT  std_logic;
         Pout : OUT  std_logic_vector(3 downto 0)
        );
   END COMPONENT;


  --Inputs    signal rst : std_logic := '0';    signal clk : std_logic := '1';
  signal mode : std_logic_vector(1 downto 0) := (others => '0');
  signal Sin : std_logic := '0';
  signal Pin : std_logic_vector(3 downto 0) := "1010";

            --Outputs
  signal Sout : std_logic;
  signal Pout : std_logic_vector(3 downto 0);

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

BEGIN
```

-- Instantiate the Unit Under Test (UUT) uut: UNI_Shift_Register PORT MAP (

```
rst => rst,          clk => clk,
mode => mode,
Sin => Sin,
Pin => Pin,
Sout => Sout,
Pout => Pout
```

```
);
```

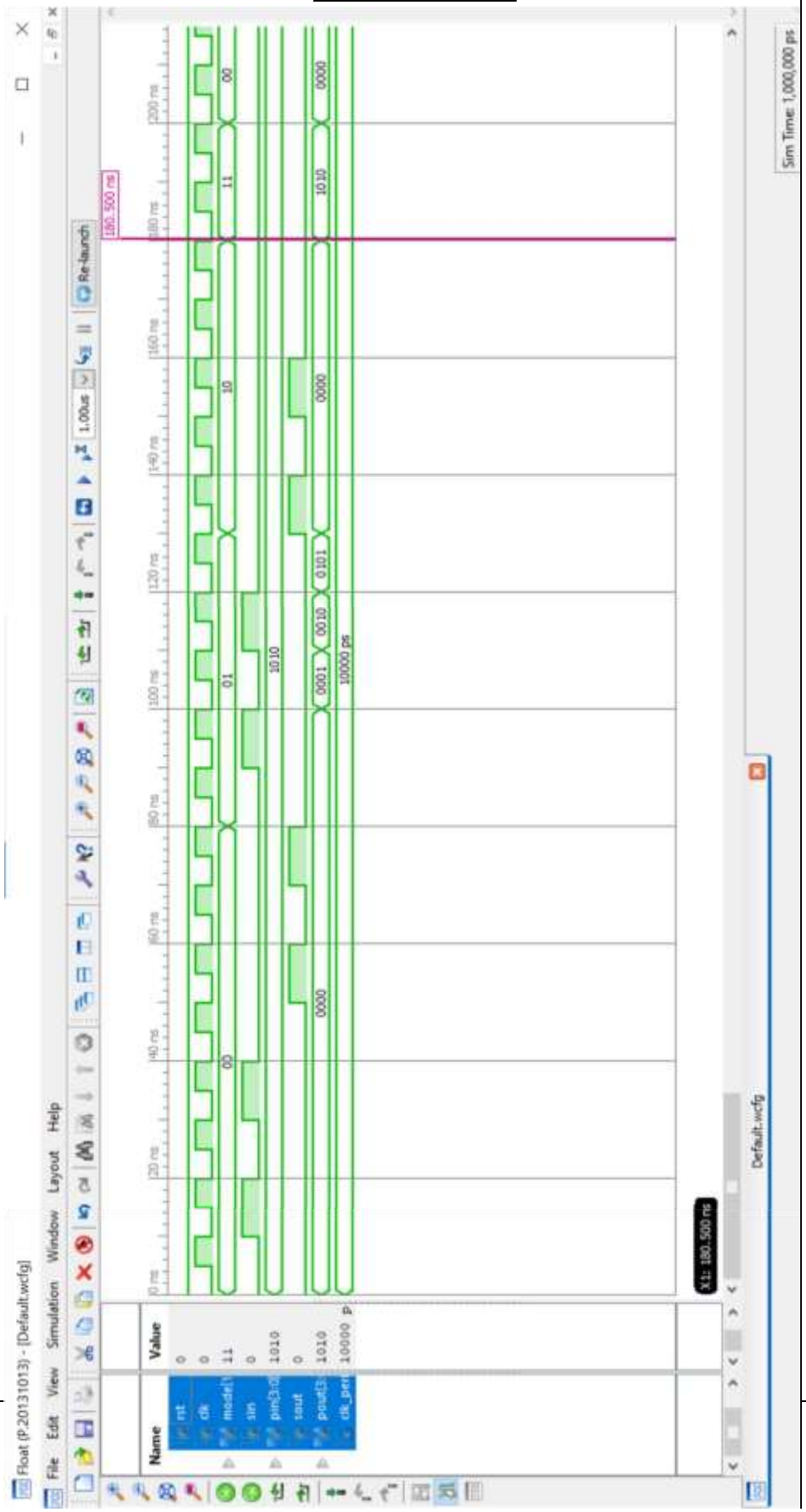
```
-- Clock process definitions    clk_process :process    begin

                clk<=NOT(clk);

                wait for clk_period/2 ;

end process;


-- Stimulus process    stim_proc_mode: process    begin                mode<="00";

        wait for 80 ns;

        mode<="01";

        wait for 50 ns;

        mode<="10";

        wait for 50 ns;

        mode<="11";

        wait for 20 ns;    end process;


stim_proc_Sin:process

        begin

        wait for 10 ns;

        Sin<='1';          wait for 10 ns;

        Sin<='0';          wait for 10 ns;

        Sin<='1';          wait for 10 ns;

        Sin<='0';          wait for 10 ns;
```

```
Sin<= '0';          wait for 40 ns;
```

```
Sin<='1';          wait for 10 ns;
```

```
        Sin<='0';          wait for 10 ns;

        Sin<='1';          wait for 10 ns;

        Sin<='0';          wait for 10 ns;

Sin<= '0';         wait ;

        end process;


        stim_proc_rst:process   begin
        rst<='0';
        wait for 300 ns;

        rst<='1';          wait for 10 ns;   end process;
END;
```

ISIM WAVEFORMS



PIN-LOCKING REPORT

PlanAhead Generated physical constraints

NET "clk" LOC = P132;

NET "rst" LOC = P204;

NET "mode[1]" LOC = P205;

NET "mode[0]" LOC = P206;

NET "Sin" LOC = P203;

NET "Pin[3]" LOC = P202; NET "Pin[2]" LOC = P197;

NET "Pin[1]" LOC = P199;

NET "Pin[0]" LOC = P196;

NET "Sout" LOC = P193;

NET "Pout[3]" LOC = P186;

NET "Pout[2]" LOC = P187;

NET "Pout[1]" LOC = P185;

NET "Pout[0]" LOC = P181;

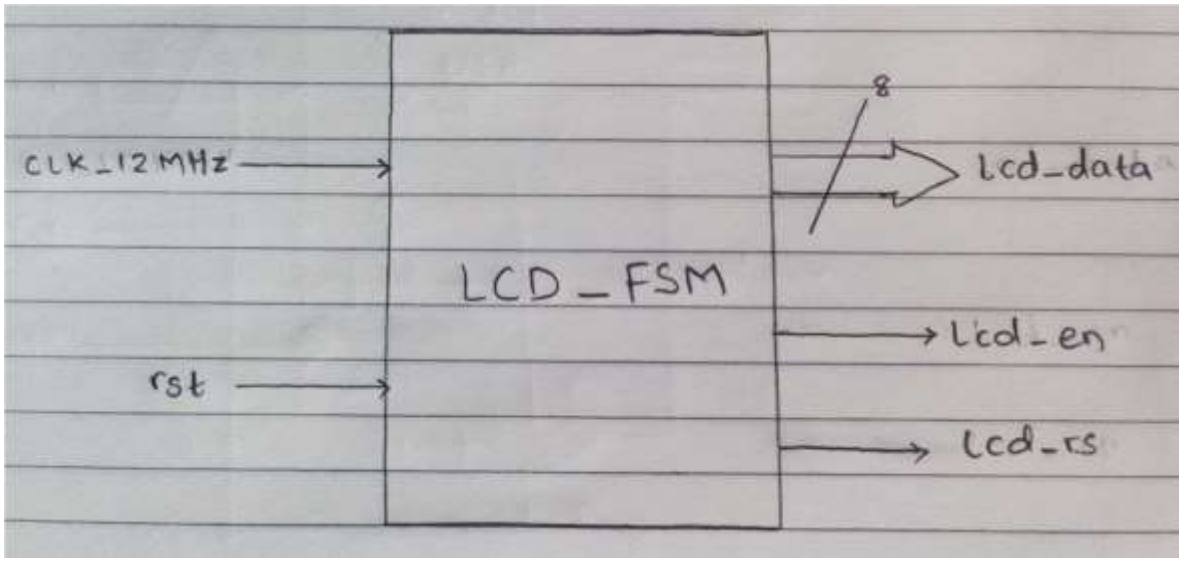
CONCLUSION

Thus, we have:

8) Modeled a Universal Shift Register using Behavioral Modeling Style.
9) Observed following Schematics: RTL & Technology Schematics generated Post-Synthesis.
10) Interpreted Device Utilization Summary in terms of LUTs, SLICES, IOBs, Multiplexers & D FFs used out of the available device resources.
11) Interpreted the TIMING Report in terms of Maximum combinational delay as indicative of the Maximum Operating Frequency.
12) Written a TESTBENCH to verify the functionality of Universal Shift Register & verified the functionality as per the FUNCTION-TABLE, by observing ISIM Waveforms.
13) Used PlanAhead Editor for pin-locking.
14) Prototyped the FPGA XC3S250EPQ208-5 to realize Universal Shift Register & verified its operation by giving suitable input combinations.

Class	:	BE-6
Roll. No	:	42247
Assignment No.	:	A.4
Assignment Name	:	FPGA-LCD Interfacing
Date Of Performance	:	22/09/2021

BLOCK DIAGRAM


CLK_12MHz
rst
LCD_FSM
8
lcd_data
lcd_en
lcd_rs

FUNCTION TABLE

rst	clk	lcd_data	lcd_rs	lcd_en
1	x	38h	0	x
0	↑	06h	0	↑
0	↑	0Ch	0	↑
0	↑	01h	0	↑
0	↑	50h (P)	1	↑
0	↑	49h (I)	1	↑
0	↑	43h (C)	1	↑
0	↑	54h (T)	1	↑
0	↑	20h ( )	1	↑

MAIN VHDL PROGRAM: (MVM)

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

entity LCD_FSM is
        Port (  rst : in std_logic;             -- reset
                clk_12Mhz : in std_logic;       --  high  freq.
                                                clock
                lcd_rs : out std_logic;         -- LCD RS control
                lcd_en : out std_logic;         -- LCD Enable
    lcd_data : out std_logic_vector(7 downto 0));    -- LCD Data port end LCD_FSM;

architecture Behavioral of LCD_FSM is

signal div : std_logic_vector(15 downto 0); --- delay timer 1 signal clk_fsm,lcd_rs_s: std_logic; -- LCD
controller FSM states
type state is (reset,func,mode,cur,clear,d0,d1,d2,d3,d4,hold); signal ps1,nx      : state;
```

```
signal dataout_s  : std_logic_vector(7 downto 0); --- internal data command multiplexer
```

begin

```
----- clk divider ---------- process(rst,clk_12Mhz) begin if(rst = '1')then
        div <= (others=>'0');
elsif( clk_12Mhz'event and clk_12Mhz ='1')then

        div <= div + 1;    end if;

end process;
---------- clk_fsm <= div(15);

----- Presetn state Register ---------- process(rst,clk_fsm) begin if(rst = '1')then
        ps1      <= reset;
elsif (rising_edge(clk_fsm)) then
        ps1      <= nx;

end if; end process; ----- state and output  decoding process  process(ps1) begin
case(ps1) is
```

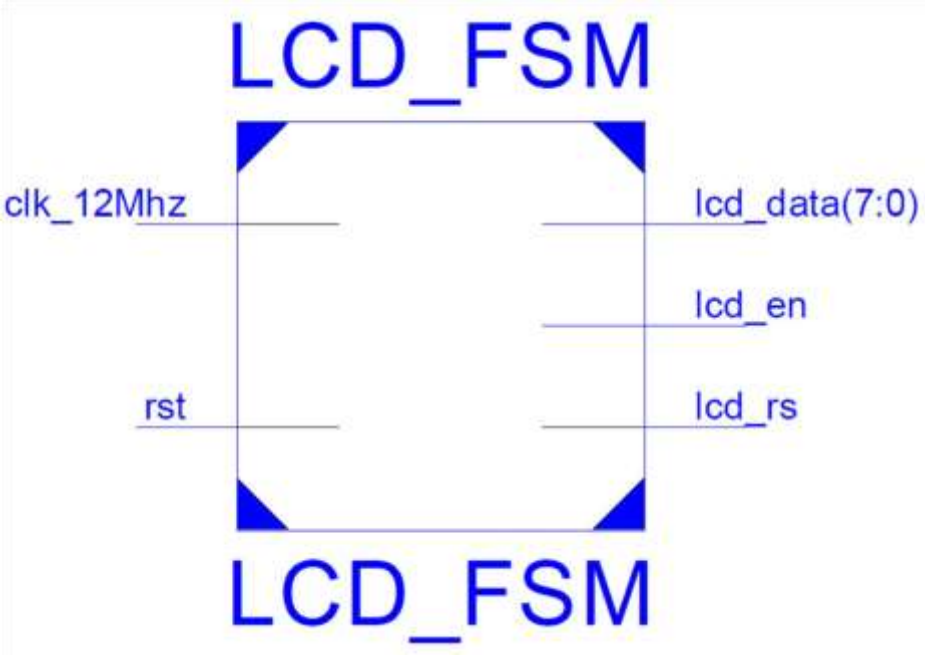
```
when reset =>
```

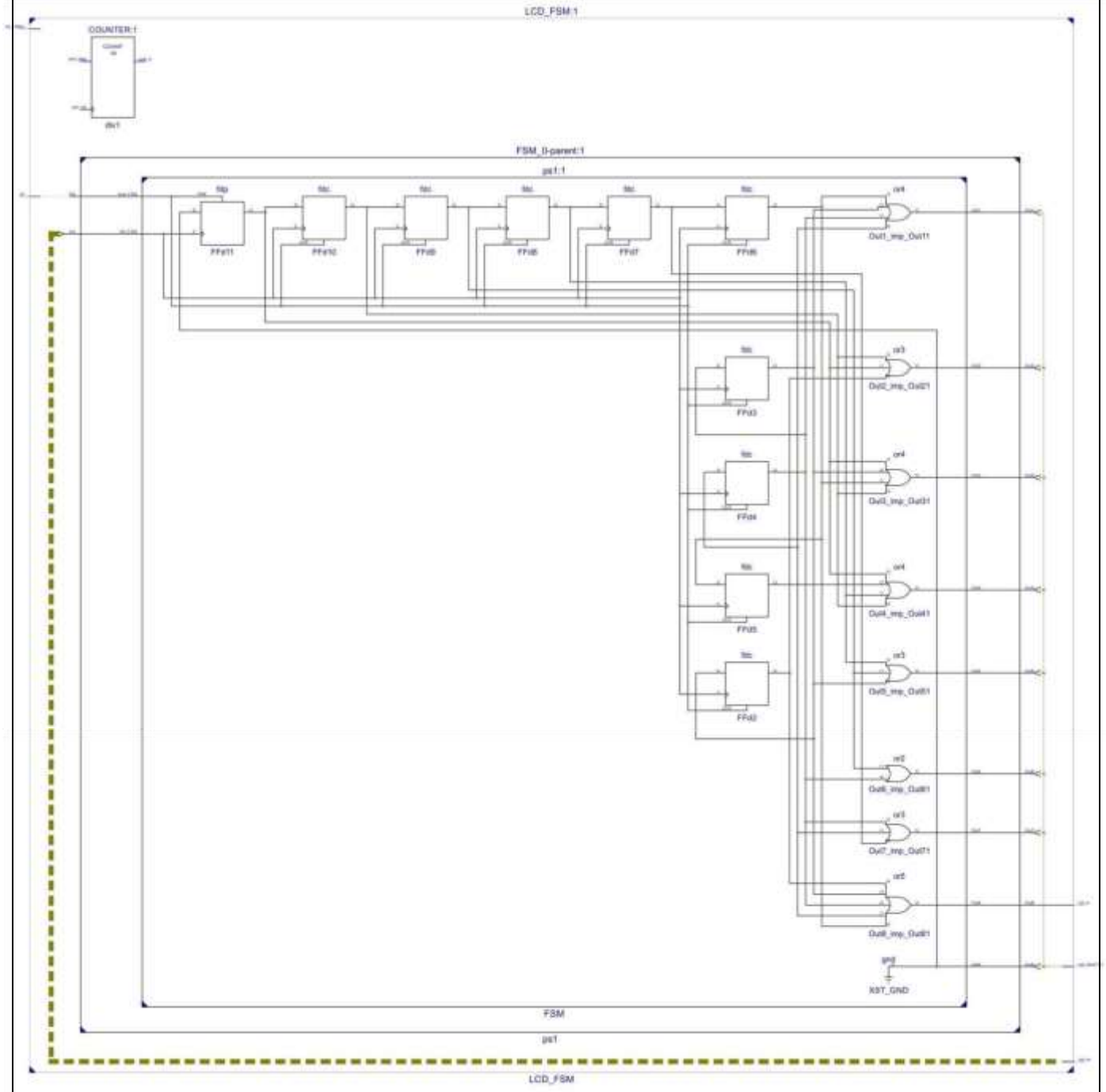
```
                    nx      <= func;
                    lcd_rs_s        <= '0';
                    dataout_s       <= "00111000";          -- 38h

            when func   =>
                    nx      <= mode;
                    lcd_rs_s        <= '0';
                    dataout_s       <= "00111000";          -- 38h

            when mode   =>
                    nx      <= cur;
                    lcd_rs_s        <= '0';
                    dataout_s       <= "00000110";          -- 06h

            when cur    =>
                    nx      <= clear;
                    lcd_rs_s        <= '0';
                    dataout_s <= "00001100";          -- 0Ch  curser at starting point of
line1

            when clear=>
                    nx      <= d0;
                    lcd_rs_s        <= '0';
                    dataout_s       <= "00000001";          -- 01h

            when d0     =>
                    lcd_rs_s        <= '1';
                    dataout_s       <= "01010000";          -- P ( Decimal = 80 , HEX =
                    50 )
                    nx      <= d1;

            when d1     =>
                    lcd_rs_s        <= '1';
                    dataout_s       <= "01001001";          -- I ( Decimal = 73 , HEX = 49
                    )
                    nx      <= d2;

            when d2     =>
                    lcd_rs_s        <= '1';
                    dataout_s       <= "01000011";          -- C ( Decimal = 67 , HEX =
                    43 )
                    nx      <= d3;
```

```
                when d3         =>
                                lcd_rs_s            <= '1';
                                dataout_s           <= "01010100";            -- T ( Decimal = 84 , HEX = 54
                                )
                                nx          <= d4;

                when d4         =>
                                lcd_rs_s            <= '1';
                                dataout_s <= "00100000";            -- space ( Decimal = 32 , HEX = 20 )
                                nx          <= hold;

               when hold        =>
                                lcd_rs_s            <= '0';
                                dataout_s <= "00000000";            -- hold  ( Decimal = 32 , HEX = 00 ) ,
        NULL
                                nx          <= hold;

        when others=>
                                nx          <= reset;
                                lcd_rs_s            <= '0';
                                dataout_s <= "00000001";  -- CLEAR ( Decimal = 1 , HEX = 01 )
end case; end process;


lcd_en <= clk_fsm;  lcd_rs <= lcd_rs_s;

lcd_data <= dataout_s;
```

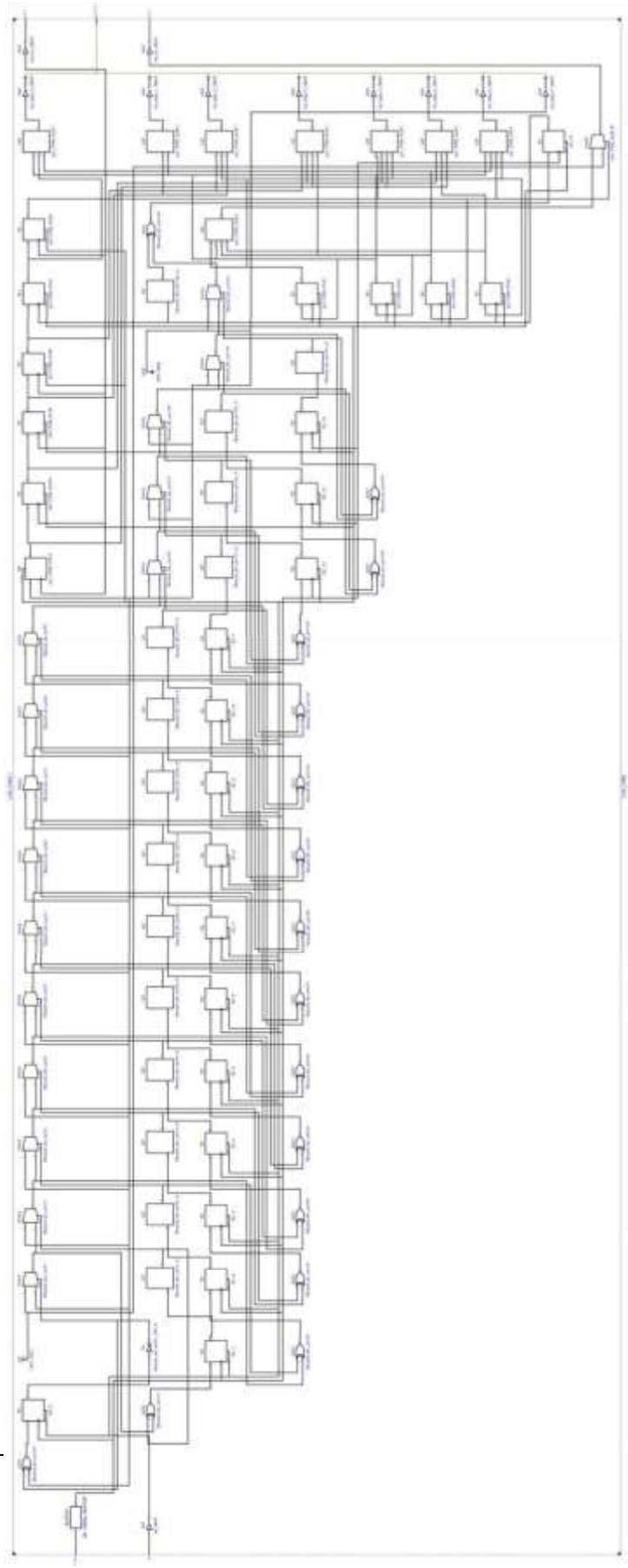
```
end Behavioral;
```

RTL SCHEMATIC:


LCD_FSM
clk_12Mhz
rst
lcd_data(7:0)
lcd_en
lcd_rs
LCD_FSM


LCD_FSM:1
COUNTER:1
FSM_0-parent:1
ps1:1
FFd11
FFd12
FFd0
FFd6
FFd7
FFd8
FFd3
FFd4
FFd5
FFd2
Out1_inp_Out11
Out2_inp_Out21
Out3_inp_Out31
Out4_inp_Out41
Out5_inp_Out51
Out6_inp_Out61
Out7_inp_Out71
Out8_inp_Out81
KST_GND
FSM
ps1
LCD_FSM

TECHNOLOGY SCHEMATIC



SYNTHESIS REPORT

a) Device Utilization Summary:

==========

* Final Report *

==========

Final Results

RTL Top Level Output File Name : LCD_FSM.ngr

Top Level Output File Name : LCD_FSM

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : No

Design Statistics

IOs : 12

Cell Usage :

BELS : 58

GND : 1

INV : 1

LUT1 : 15

LUT2 : 1

LUT3 : 3 # LUT4 : 4

MUXCY : 15

MUXF5 : 1

VCC : 1

```
#      XORCY                : 16
# FlipFlops/Latches         : 26
#      FDC                  : 25
#      FDP                  : 1
# Clock Buffers             : 1
#      BUFGP                : 1
# IO Buffers                : 11
#      IBUF                 : 1
#      OBUF                 : 10
```

==========

Device utilization summary:

Selected Device : 3s250epq208-5

Number of Slices: 15 out of 2448 0%

Number of Slice Flip Flops: 26 out of 4896 0%

Number of 4 input LUTs: 24 out of 4896 0%

Number of IOs: 12

Number of bonded IOBs: 12 out of 158 7% Number of GCLKs: 1 out of 24 4%

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_12Mhz | BUFGP | 16 | div_15 | NONE(ps1_FSM_FFd11) | 10 |

INFO:Xst:2169 - HDL ADVISOR - Some clock signals were not automatically buffered by XST with

BUFG/BUFR resources. Please use the buffer_type constraint in order to insert these buffers to the clock signals to help prevent skew problems.

Asynchronous Control Signals Information:

----------+----------+-------+

Control Signal | Buffer(FF name) | Load | ----------+----------
---+-------+ rst | IBUF | 26 |

----------+----------+-------+

Timing Summary:

Speed Grade: -5

Minimum period: 3.676ns (Maximum Frequency: 272.072MHz)

Minimum input arrival time before clock: No path found

Maximum output required time after clock: 5.537ns

Maximum combinational path delay: No path found

Timing Detail:

All values displayed in nanoseconds (ns)

TESTBENCH PROGRAM (TVM)

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

ENTITY LCD_Test IS
END LCD_Test;

ARCHITECTURE behavior OF LCD_Test IS
```

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

COMPONENT LCD_FSM
PORT(	rst : IN  std_logic;	clk_12Mhz : IN  std_logic;	lcd_rs : OUT  std_logic;
lcd_en : OUT  std_logic;
```

```
lcd_data : OUT  std_logic_vector(7 downto 0)
```

```
);
```

```
 END COMPONENT;


  --Inputs    signal rst : std_logic := '0';
  signal clk_12Mhz : std_logic := '0';

          --Outputs
  signal lcd_rs : std_logic;    signal lcd_en : std_logic;
  signal lcd_data : std_logic_vector(7 downto 0);

  -- Clock period definitions
  constant clk_12Mhz_period : time := 10 ns;

BEGIN

          -- Instantiate the Unit Under Test (UUT)
  uut: LCD_FSM PORT MAP (
       rst => rst,
       clk_12Mhz => clk_12Mhz,          lcd_rs => lcd_rs,          lcd_en => lcd_en,
       lcd_data => lcd_data
      );

  -- Clock process definitions    clk_12Mhz_process :process
  begin
                    clk_12Mhz <= '0';                    wait for clk_12Mhz_period/2;                clk_12Mhz
<= '1';
 wait for clk_12Mhz_period/2;
  end process;


  -- Stimulus process    stim_proc: process
  begin
    rst <= '1';
```

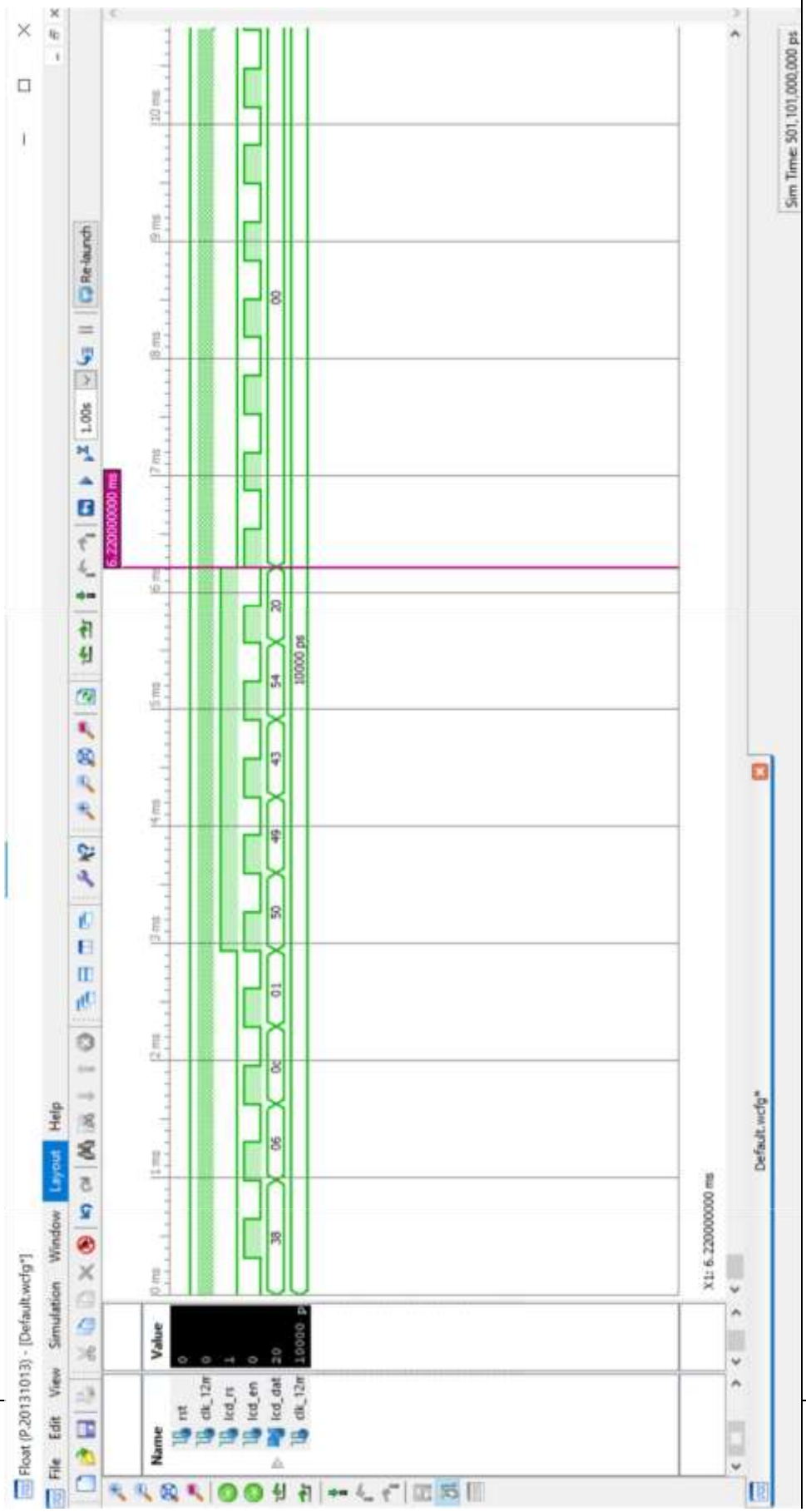
```
 wait for 20 ns;

rst <= '0';
```

-- insert stimulus here

```
    wait;
end process;
```

```
END;
```

ISIM WAVEFORMS



PIN-LOCKING REPORT

PlanAhead Generated physical constraints

NET "clk_12Mhz" LOC = P80;

NET "rst" LOC = P204;

NET "lcd_rs" LOC = P48;

NET "lcd_en" LOC = P49;

NET "lcd_data[0]" LOC = P47;

NET "lcd_data[1]" LOC = P41;

NET "lcd_data[2]" LOC = P39;

NET "lcd_data[3]" LOC = P35;

NET "lcd_data[4]" LOC = P33;

NET "lcd_data[5]" LOC = P31;

NET "lcd_data[6]" LOC = P29;

NET "lcd_data[7]" LOC = P24;

CONCLUSION

Thus, we have:

15) Modeled a FPGA-LCD Interfacing using Behavioral Modeling Style.

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

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

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

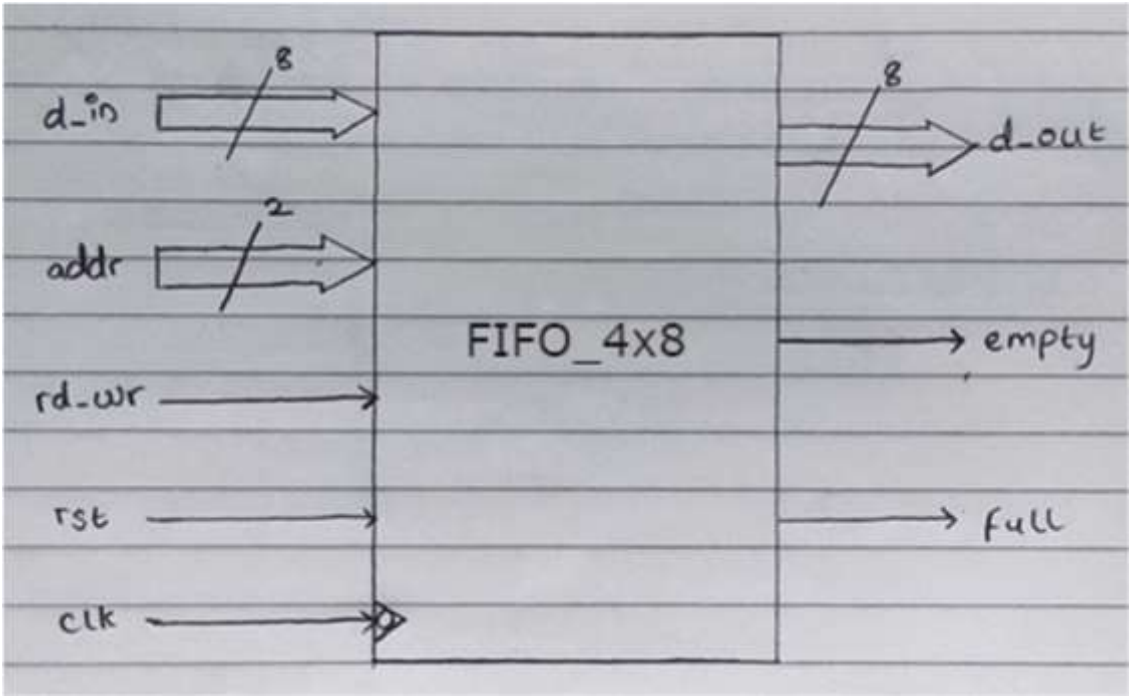
19) Written a TESTBENCH to verify the functionality of FPGA-LCD Interfacing & verified the functionality as per the FUNCTION-TABLE, by observing ISIM Waveforms.

20) Used PlanAhead Editor for pin-locking.

21) Prototyped the FPGA XC3S250EPQ208-5 to realize FPGA-LCD Interfacing & verified its operation by giving suitable input combinations.

Class	:	BE-6
Roll. No	:	42247
Assignment No.	:	A.3
Assignment Name	:	FIFO
Date Of Performance	:	15/09/2021

BLOCK DIAGRAM


d_in
8
addr
2
rd_wr
rst
clk
FIFO_4x8
8
d_out
empty
full

FUNCTION TABLE

rst	clk	addr		rd_wr	d_out	empty	full
		addr(1)	addr(0)				
1	x	x	x	x	$(00)_{16}$	1	0
0	↓	0	0	0	mem_0	0	1
0	↓	0	1	0	mem_1	0	1
0	↓	1	0	0	mem_2	0	1
0	↓	1	1	0	mem_3	0	1
0	↓	0	0	1	mem_0	0	0
0	↓	0	1	1	mem_1	0	0
0	↓	1	0	1	mem_2	0	0
0	↓	1	1	1	mem_3	0	1

MAIN VHDL PROGRAM (MVM)

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

entity FIFO_4x8 is     Port ( rst : in  STD_LOGIC;
        clk : in  STD_LOGIC;
        addr : in  STD_LOGIC_VECTOR (1 downto 0) := "00";          d_in : in  STD_LOGIC_VECTOR (7
downto 0);
        rd_wr : in  STD_LOGIC;          d_out : out  STD_LOGIC_VECTOR (7 downto 0) := "00000000";
        empty : out  STD_LOGIC := '1';          full : out  STD_LOGIC := '0'); end FIFO_4x8;

architecture FIFO_4x8_arch of FIFO_4x8 is
```

```
TYPE mem IS ARRAY(3 DOWNTO 0) OF STD_LOGIC_VECTOR (7 DOWNTO 0);
```

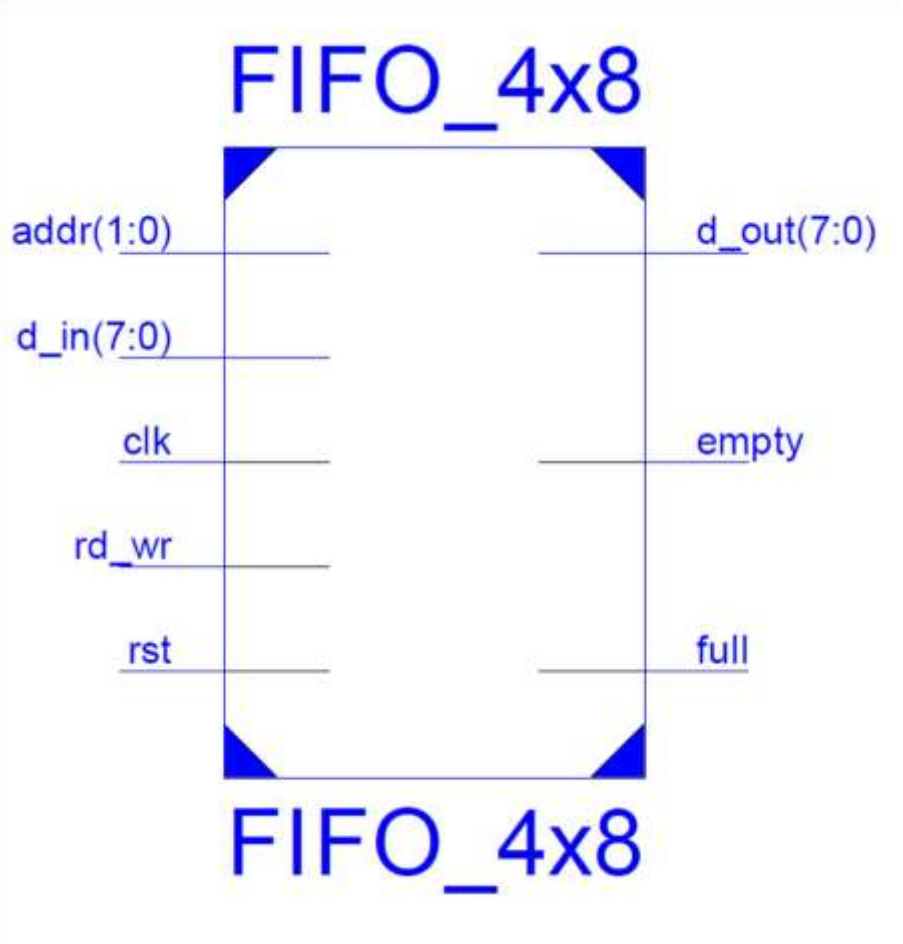
```
SIGNAL memory : mem := (others=>(others=>'0'));
```

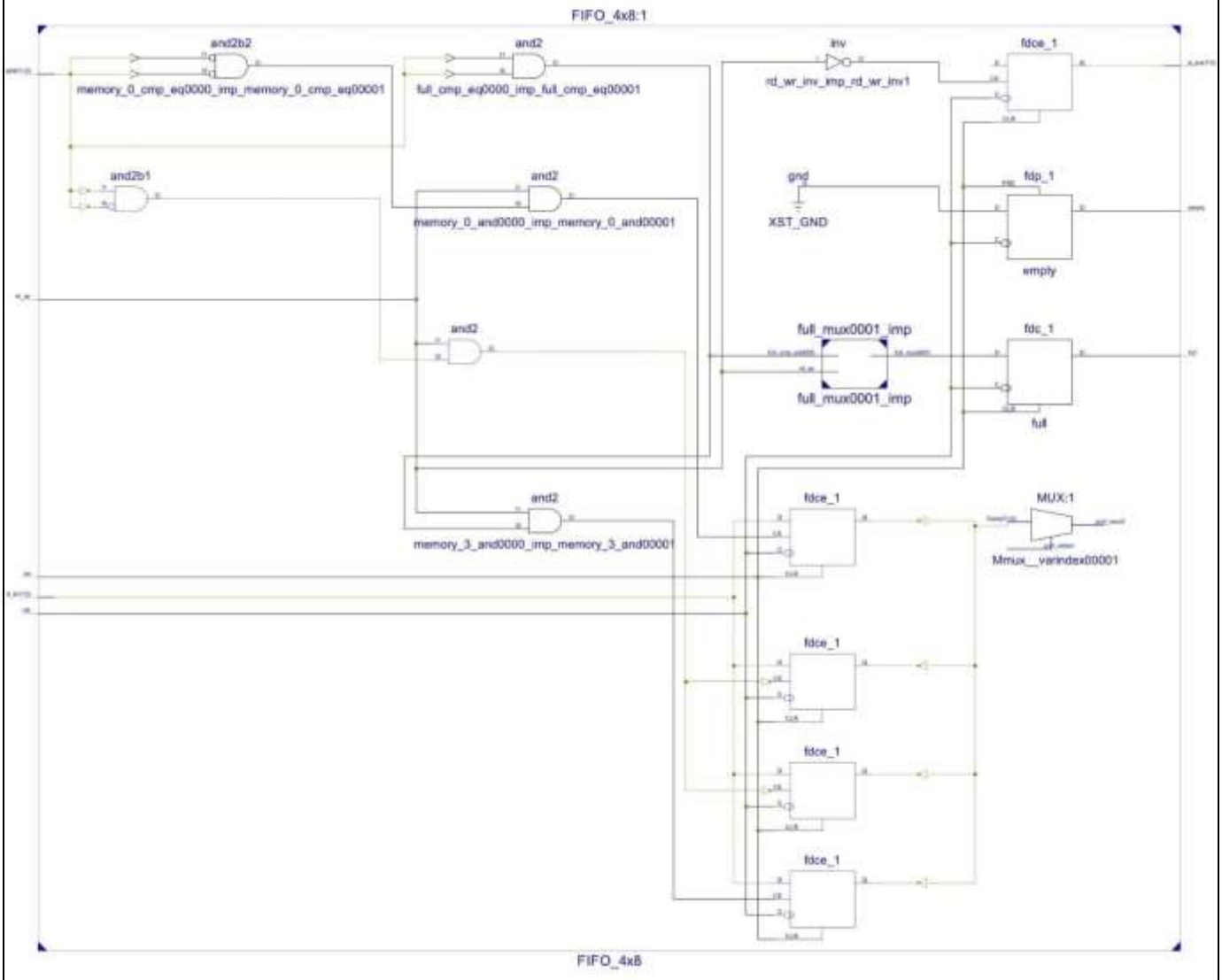
begin

```
PROCESS(rst, clk, addr, d_in, rd_wr)

        begin

          if rst = '1' then
                    d_out <= "00000000";
                                  empty <= '1';
                                  full <= '0';
                                  memory <= (others=>(others=>'0'));

   elsif falling_edge(clk) then

      case rd_wr is

                                    when '0' =>

                                          d_out <= memory(conv_integer(addr));
                                  empty <= '0';                                    full <= '1';

                                    when others =>

                                          memory(conv_integer(addr)) <= d_in;
                                                empty <= '0';
        if addr = "11" then
                                              full <= '1';
                                                    else
                                              full <= '0';
   end if;          end case;
   end if;
end process;
```

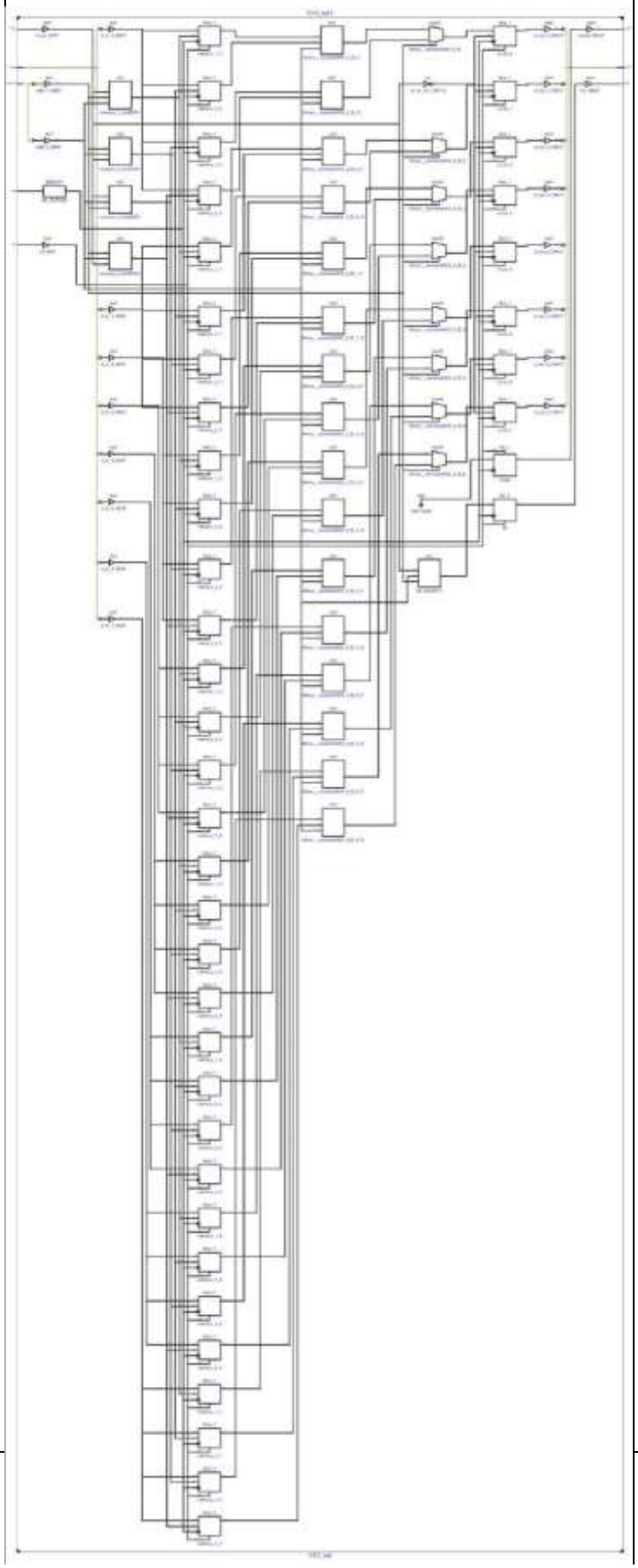
```
end FIFO_4x8_arch;
```

RTL SCHEMATIC:


FIFO_4x8
addr(1:0)
d_in(7:0)
clk
rd_wr
rst
d_out(7:0)
empty
full
FIFO_4x8


FIFO_4x8:1
and2b2
memory_0_cmp_eq0000_imp_memory_0_cmp_eq00001
and2
full_cmp_eq0000_imp_full_cmp_eq00001
inv
rd_wr_inv_imp_rd_wr_inv1
fdce_1
and2b1
and2
memory_0_and0000_imp_memory_0_and00001
gnd
XST_GND
fdp_1
empty
and2
full_mux0001_imp
full_mux0001_imp
fdc_1
full
and2
memory_3_and0000_imp_memory_3_and00001
fdce_1
MUX:1
Mmux__varindex00001
fdce_1
fdce_1
fdce_1
FIFO_4x8

TECHNOLOGY SCHEMATIC



SYNTHESIS REPORT

a) Device Utilization Summary:

```
=========================================================================
*                      Final Report                     *
=========================================================================
Final Results
RTL Top Level Output File Name     : FIFO_4x8.ngr
Top Level Output File Name         : FIFO_4x8
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 23

Cell Usage :
# BELS                             : 32
#      GND                         : 1
#      INV                         : 2
#      LUT3                        : 21
#      MUXF5                       : 8
# FlipFlops/Latches                : 42
#      FDC_1                       : 1
#      FDCE_1                      : 40
```

FDP_1 : 1

```
# Clock Buffers                 : 1
#      BUFGP                    : 1
# IO Buffers                    : 22
#      IBUF                     : 12
#      OBUF                     : 10
```

==========

Device utilization summary:

Selected Device : 3s250epq208-5

Number of Slices:	26	out of 2448	1%
Number of Slice Flip Flops:	40	out of 4896	0%
Number of 4 input LUTs:	23	out of 4896	0%
Number of IOs:	23		
Number of bonded IOBs:	23	out of 158	14%
IOB Flip Flops:	2		
Number of GCLKs:	1	out of 24	4%

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	42

Asynchronous Control Signals Information:

Control Signal | Buffer(FF name) | Load | ----------+----------+-------+ rst_inv(rst_inv1_INV_0:O) | NONE(d_out_0) | 42 |

----------+----------+-------+

Timing Summary:

Speed Grade: -5

Minimum period: 2.098ns (Maximum Frequency: 476.644MHz)

Minimum input arrival time before clock: 3.955ns

Maximum output required time after clock: 4.040ns

Maximum combinational path delay: No path found

Timing Detail:

All values displayed in nanoseconds (ns)

TESTBENCH PROGRAM (TVM)

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

ENTITY FIFO_4x8_tb IS

END FIFO_4x8_tb;

ARCHITECTURE behavior OF FIFO_4x8_tb IS

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

   COMPONENT FIFO_4x8

   PORT(         rst : IN  std_logic;          clk : IN  std_logic;
      addr : IN  std_logic_vector(1 downto 0);         d_in : IN  std_logic_vector(7 downto 0);
rd_wr : IN  std_logic;

      d_out : OUT  std_logic_vector(7 downto 0);

      empty : OUT  std_logic;
```

full : OUT std_logic

```
);
```

```
 END COMPONENT;


--Inputs    signal rst : std_logic := '0';    signal clk : std_logic := '1';
```

```
signal addr : std_logic_vector(1 downto 0) := (others => '0');    signal d_in : std_logic_vector(7
```

```
downto 0) := (others => '0');
  signal rd_wr : std_logic := '0';

          --Outputs
  signal d_out : std_logic_vector(7 downto 0);
  signal empty : std_logic;
  signal full : std_logic;

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

BEGIN

          -- Instantiate the Unit Under Test (UUT)
  uut: FIFO_4x8 PORT MAP (
       rst => rst,          clk => clk,          addr => addr,          d_in => d_in,          rd_wr => rd_wr,
d_out => d_out,
       empty => empty,
       full => full
      );

  -- Clock process definitions    clk_process :process
  begin
                    clk <= not(clk);
 wait for clk_period/2;    end process;


  -- Stimulus process    stim_proc: process
  begin

    rst <= '0';
                    wait for clk_period;
```

```
rst <= '1';   wait for clk_period;
```

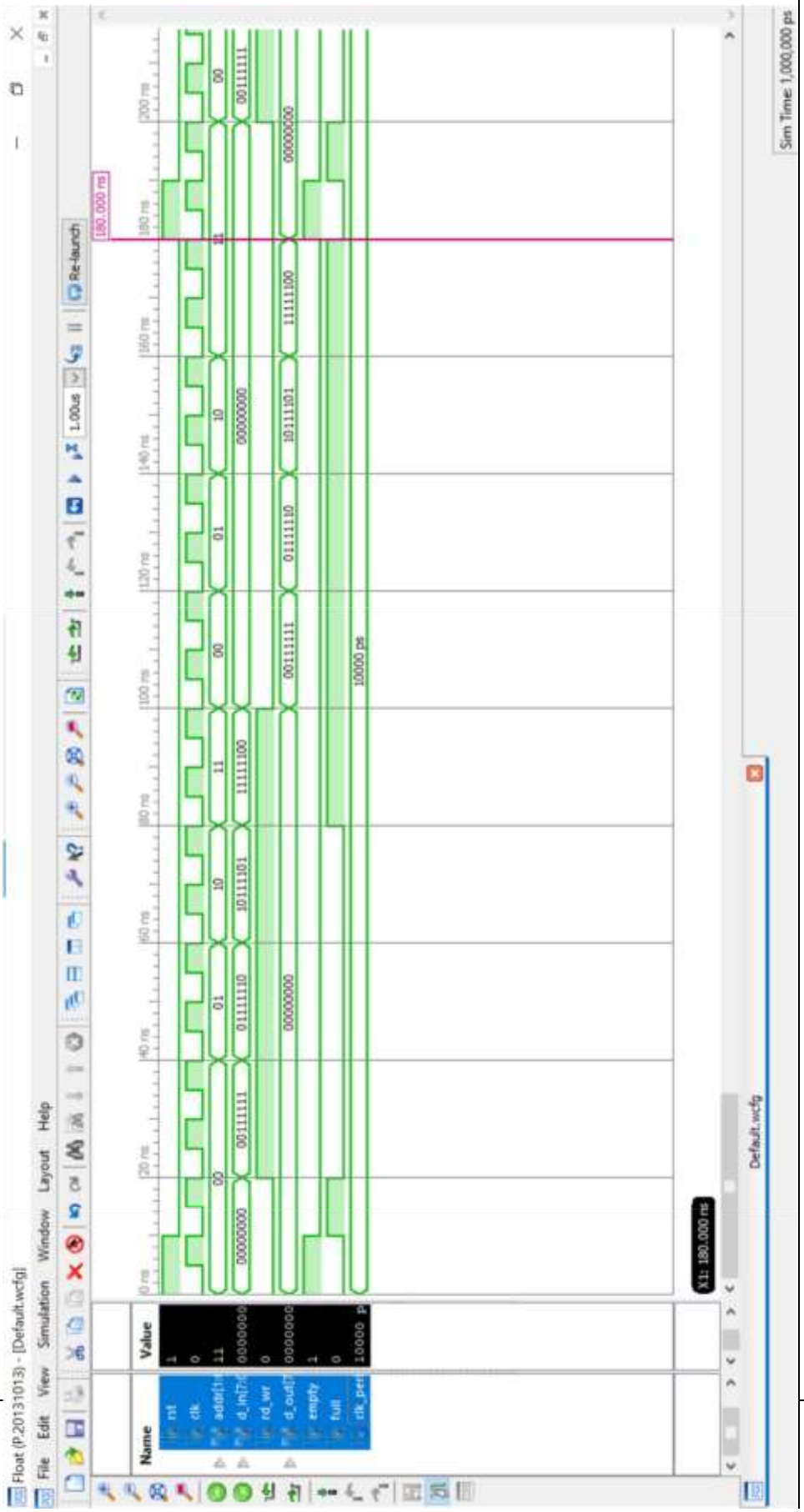
```
                rd_wr <= '1';
```

```
                for address in 0 to 3 loop                addr <=
std_logic_vector(to_unsigned(address, 2));
```

```
d_in <= std_logic_vector(to_unsigned(63*(address + 1), 8));
```

```
        wait for clk_period*2;
                end loop;

                d_in <= std_logic_vector(to_unsigned(0, 8));

                rd_wr <= '0';
                for address in 0 to 3 loop
                        addr <= std_logic_vector(to_unsigned(address, 2));
        wait for clk_period*2;              end loop;
                end process;

END;
```

ISIM WAVEFORMS



PIN-LOCKING REPORT

PlanAhead Generated physical constraints

```
NET "d_in[7]" LOC = P165;      #sw4-0
NET "d_in[6]" LOC = P167;      #sw4-1
NET "d_in[5]" LOC = P163;      #sw4-2
NET "d_in[4]" LOC = P164;
NET "d_in[3]" LOC = P161;
NET "d_in[2]" LOC = P162;
NET "d_in[1]" LOC = P160;
NET "d_in[0]" LOC = P153;      #sw4-7
NET "d_out[7]" LOC = P179;    #sw3-0
NET "d_out[6]" LOC = P180;    #sw3-1
NET "d_out[5]" LOC = P177;
NET "d_out[4]" LOC = P178;
NET "d_out[3]" LOC = P152;
NET "d_out[2]" LOC = P168;
NET "d_out[1]" LOC = P171;
NET "d_out[0]" LOC = P172;    #sw3-7
NET "clk" LOC = P132;
NET "rst" LOC = P204;             #k0
NET "rd_wr" LOC = P184;         #sw2-6
NET "empty" LOC = P199;#sw1-6
```

```
NET "full" LOC = P196; #sw1-7
```

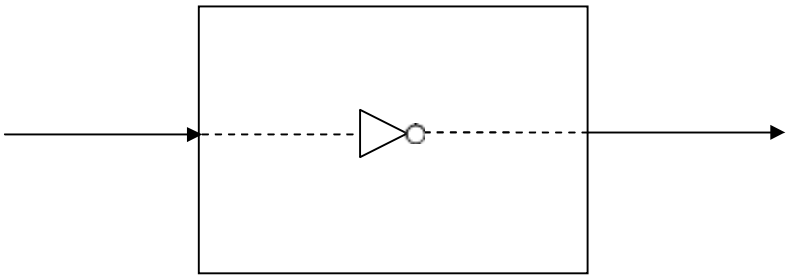
CONCLUSION

Thus, we have:

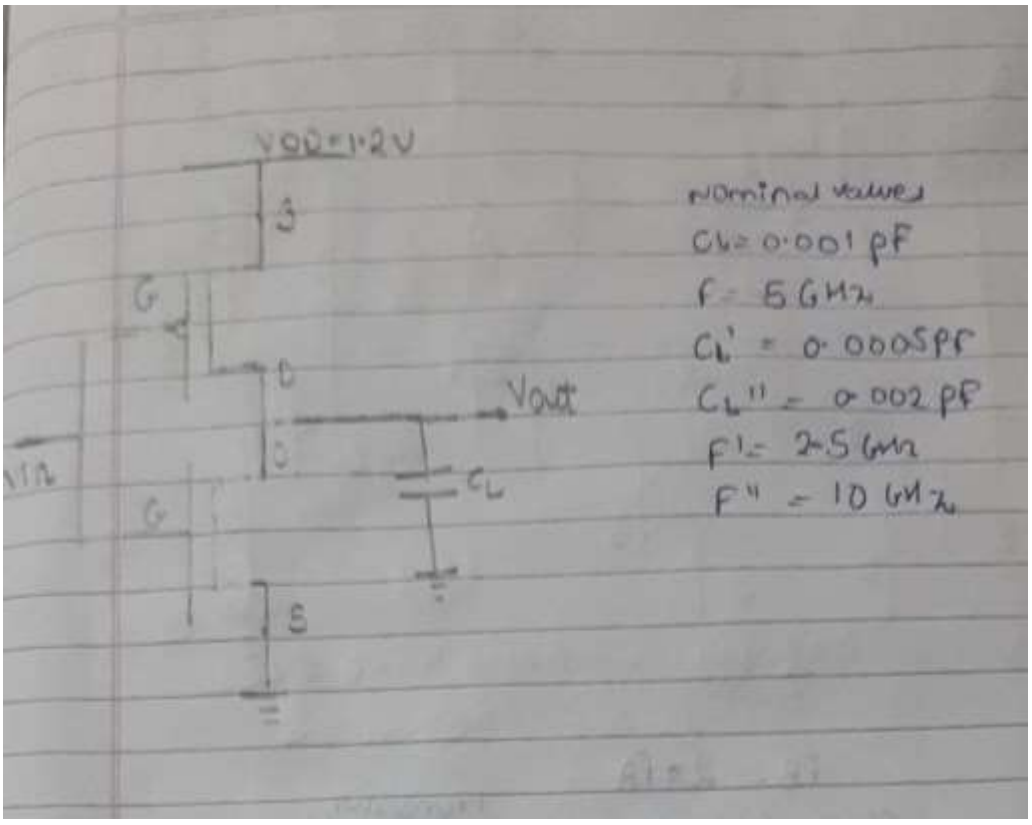
22) Modeled a 4x8 FIFO using Behavioral Modeling Style.
23) Observed following Schematics: RTL & Technology Schematics generated Post-Synthesis.
24) Interpreted Device Utilization Summary in terms of LUTs, SLICES, IOBs, Multiplexers & D FFs used out of the available device resources.
25) Interpreted the TIMING Report in terms of Maximum combinational delay as indicative of the Maximum Operating Frequency.
26) Written a TESTBENCH to verify the functionality of 4x8 FIFO & verified the functionality as per the FUNCTION-TABLE, by observing ISIM Waveforms.
27) Used PlanAhead Editor for pin-locking.
28) Prototyped the FPGA XC3S250EPQ208-5 to realize 4x8 FIFO & verified its operation by giving suitable input combinations.

Class	BE-6
Roll Number	42247
Assignment Number	B.1.a
Assignment Name	CMOS Inverter
Date of performance	29/09/2021

Block Diagram:-



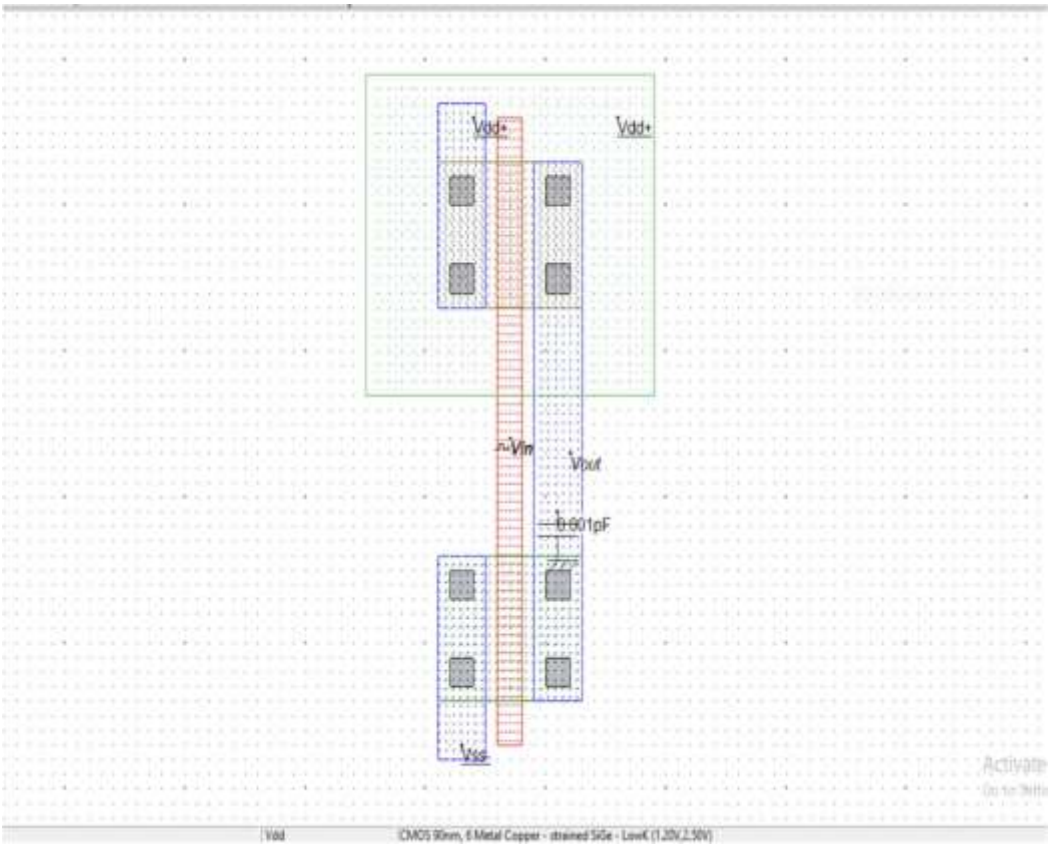
MOSFET-LEVEL SCHEMATIC :



Truth Table:-

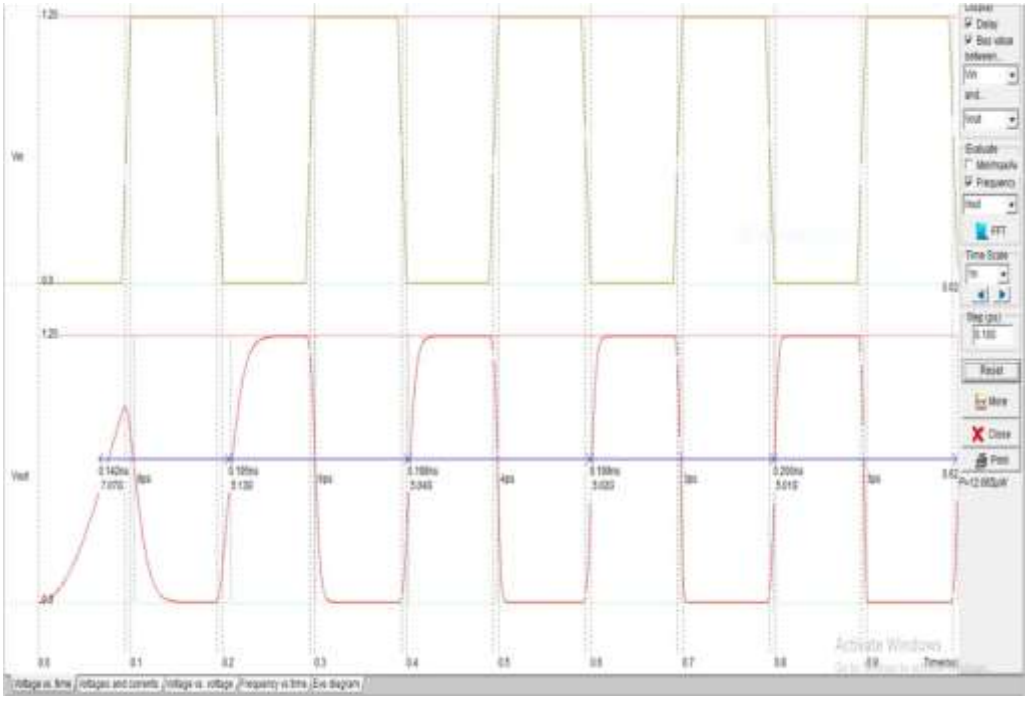
Vin	Vout
0	Strong-1
1	Strong-0

Layout:

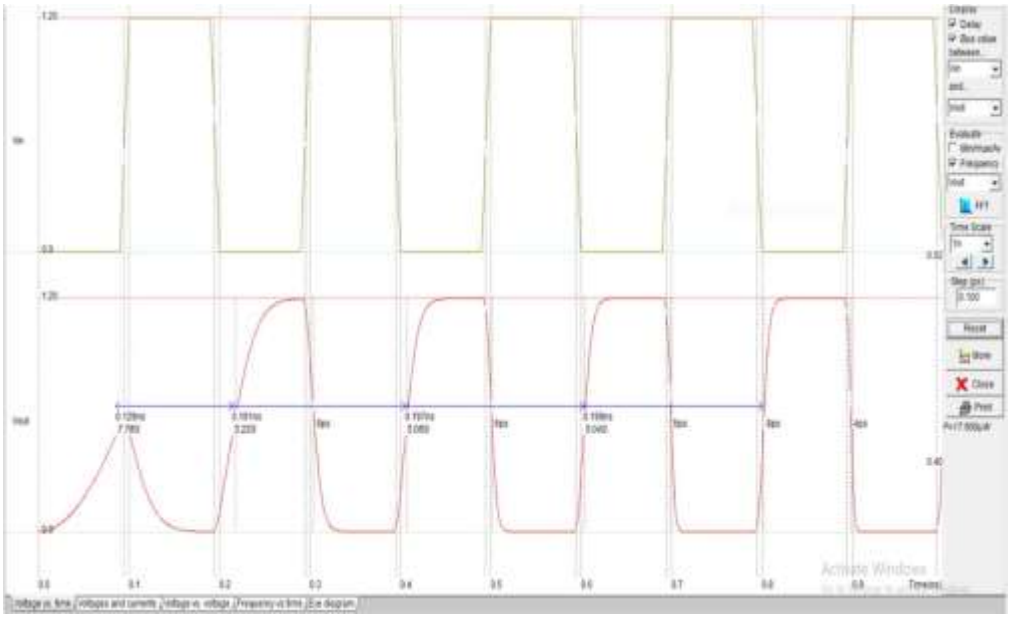


Waveforms:

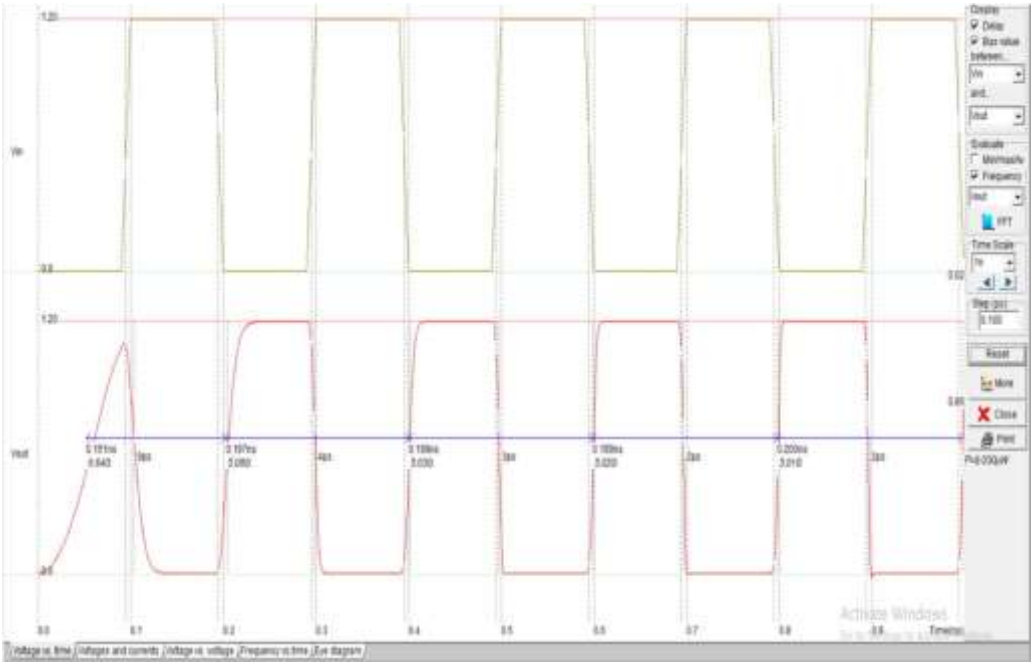
1:CL=0.001pF and F=5GHz



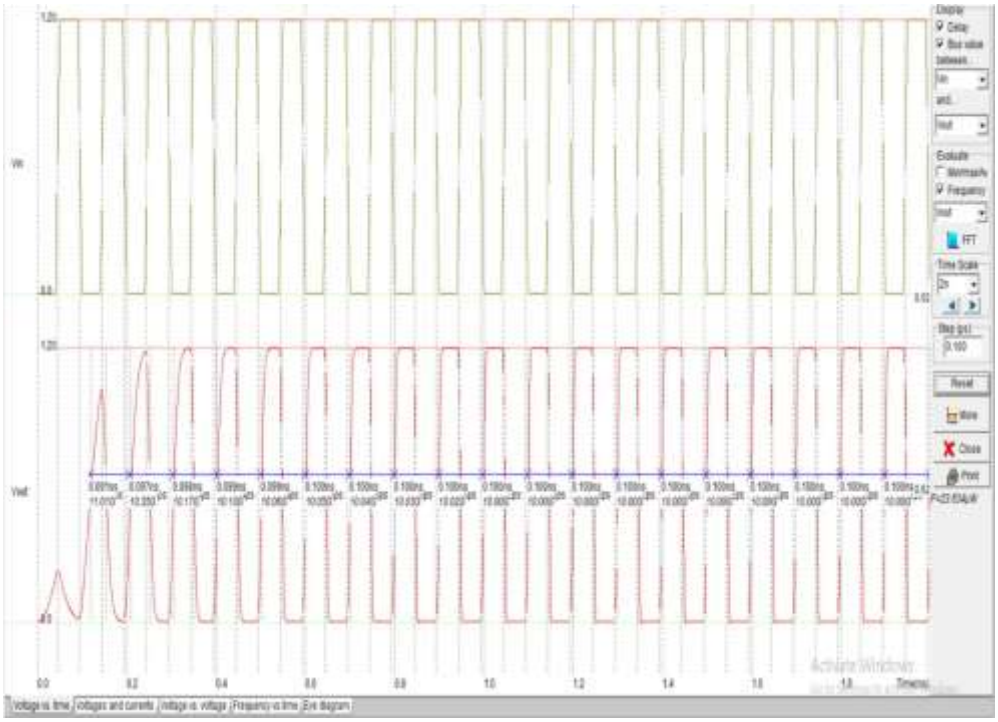
2:CL=0.002pF and F=5GHz



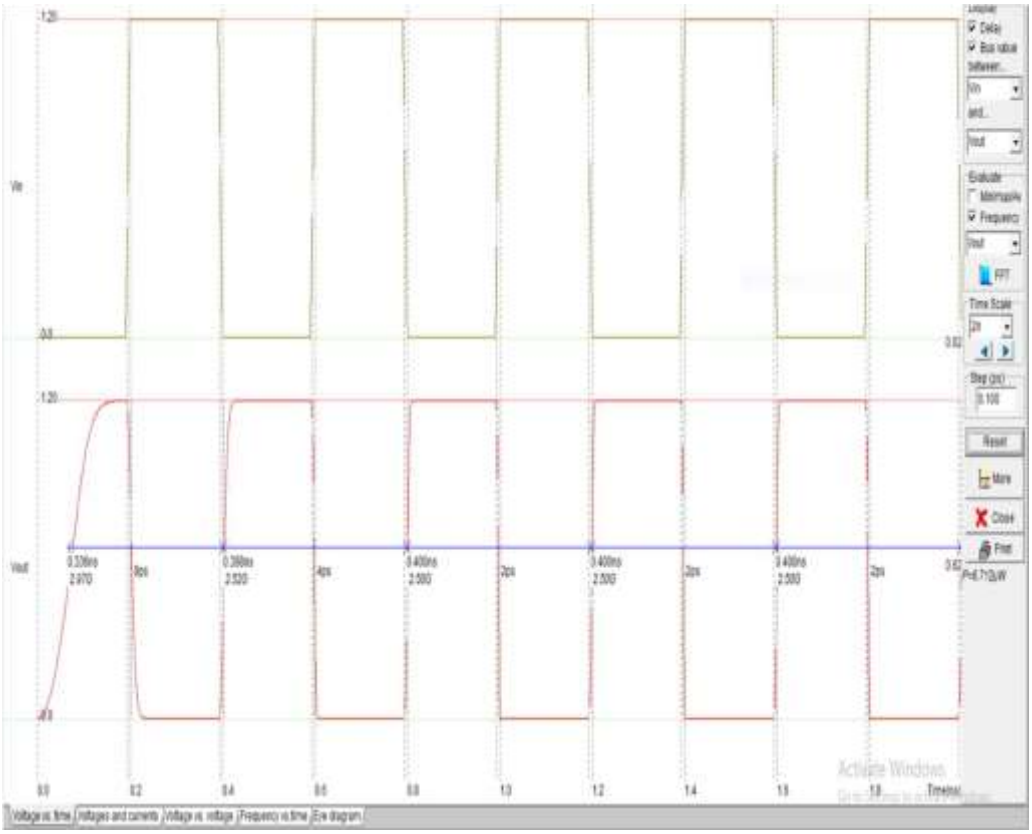
3:CL=0.0005pF and F=5Ghz



4:CL=0.001pF and F=10Ghz



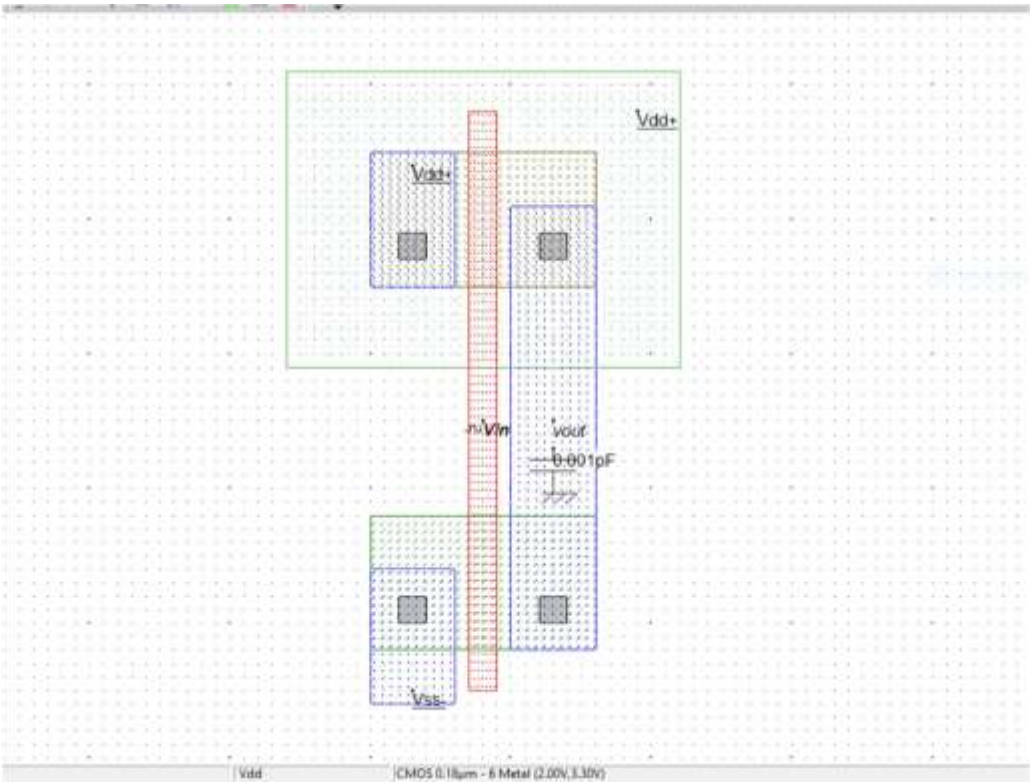
5:CL=0.001pF and F=2.5Ghz



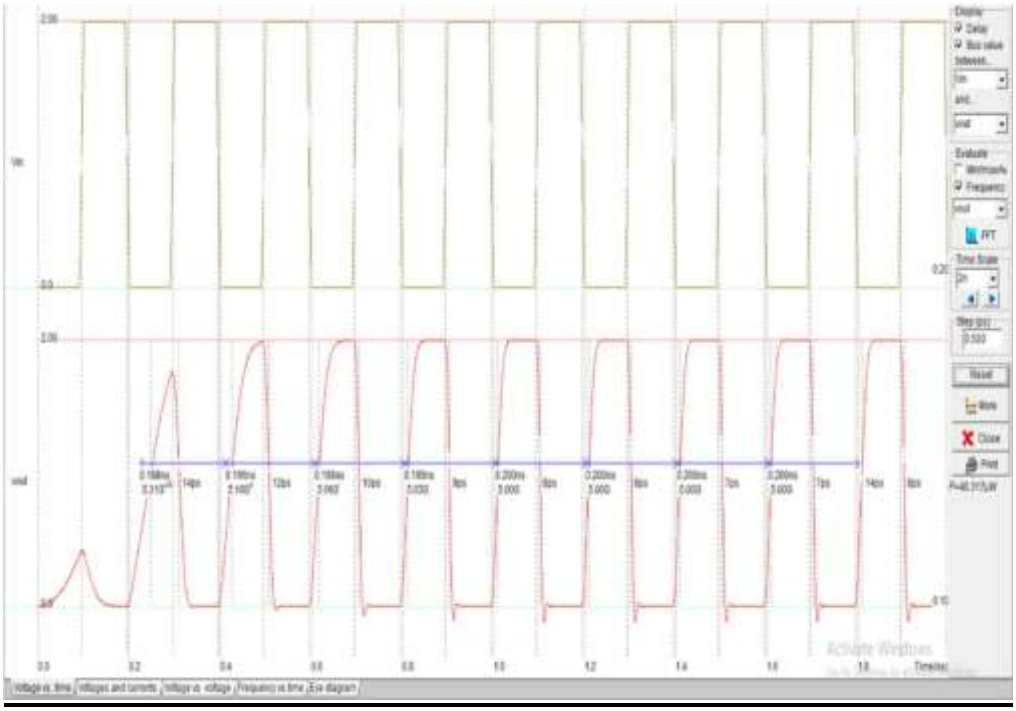
SR.NO	C_{load} (pF)	Value (uW)
1)	0.001	12.488
2)	0.002	18.627
3)	0.0005	9.485

SR.NO.	f_{clk} (GHz)	Value (uW)
1)	5	12.488
2)	2.5	6.712
3)	10	23.834

Layout for 180nm foundry:



Waveforms:-



Pdynamic Analysis: V_{dd} = 2 V

Sr.No	C_{load} (pF)	f_{clk} (GHz)	$P_{dynamic}$
1	0.001	5 GHz	40.32 uW

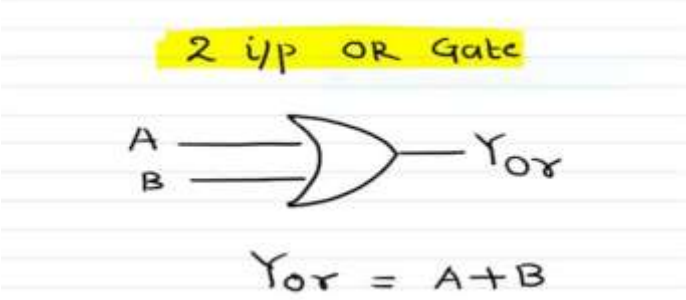
Conclusion:-

Thus we have :

1) Drawn the LAYOUT for CMOS Inverter using 90 nm & 180 nm Foundry.
2) Simulated the LAYOUT to observe waveforms & verified its functionality as per TRUTH-TABLE.
3) Noted the values of $P_{dynamic}$ for floating Load.
4) Appreciated the validity of the mathematical model :
   $P_{dynamic} = C_L * (Vdd)^2 * f_{clk}$ by Doubling & Halving the values of C& f_{clk}
5) Found a reduction in $P_{dynamic}$ by using a better Foundry i.e.,90 nm instead of 180 nm

Class	BE-6
Roll Number	42247
Assignment Number	B.1.b
Assignment Name	NAND-AND , NOR-OR Gates
Date of performance	06/10/2021

$YMBOLS:-



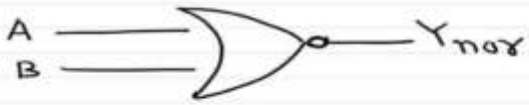
2 i/p NAND Gate

$$Y_{nand} = \overline{A \cdot B}$$

2 i/p AND Gate

$$Y_{and} = A \cdot B$$

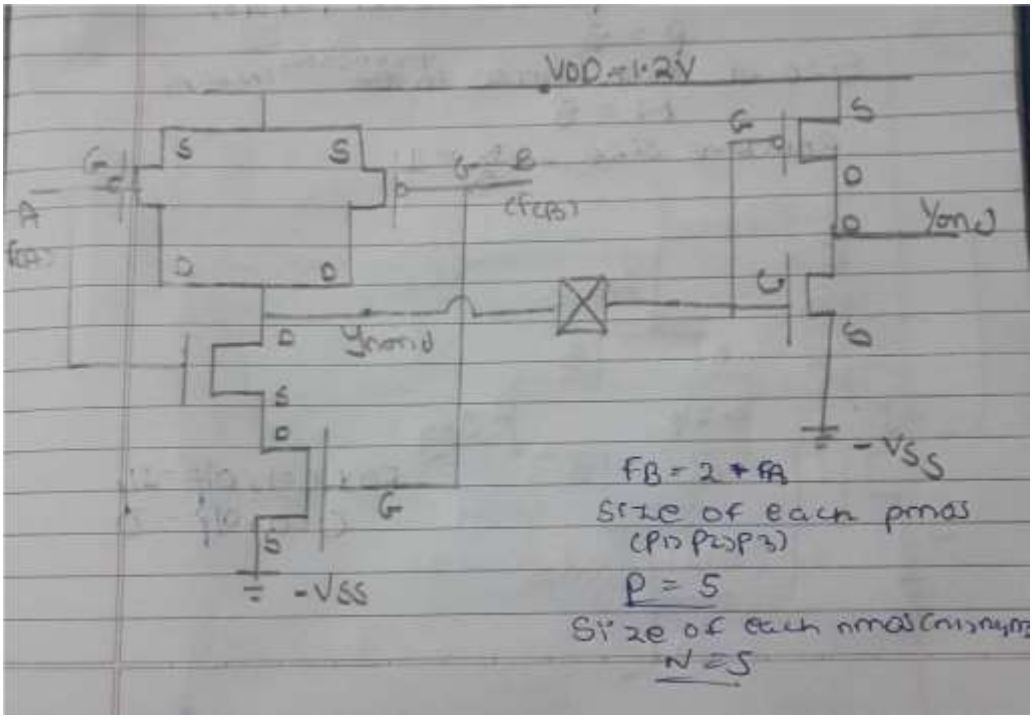
2 i/p NOR Gate



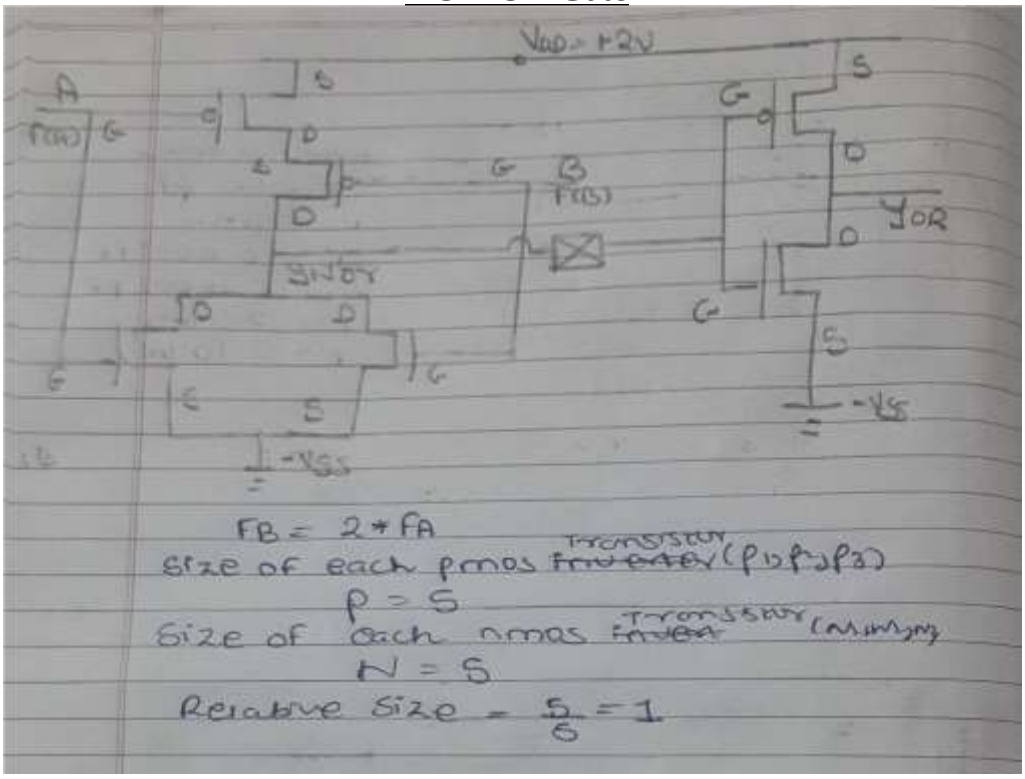
$$Y_{nand} = \overline{A + B}$$

MOSFET-LEVEL SCHEMATICS :

NAND- AND Gate



NOR-OR Gate



Truth Table:-

1. Nand Gate

A	B	Ynand
0	0	1
0	1	1
1	0	1
1	1	0

2.And Gate

A	B	Yand
0	0	0
0	1	0
1	0	0
1	1	1

3.Nor Gate

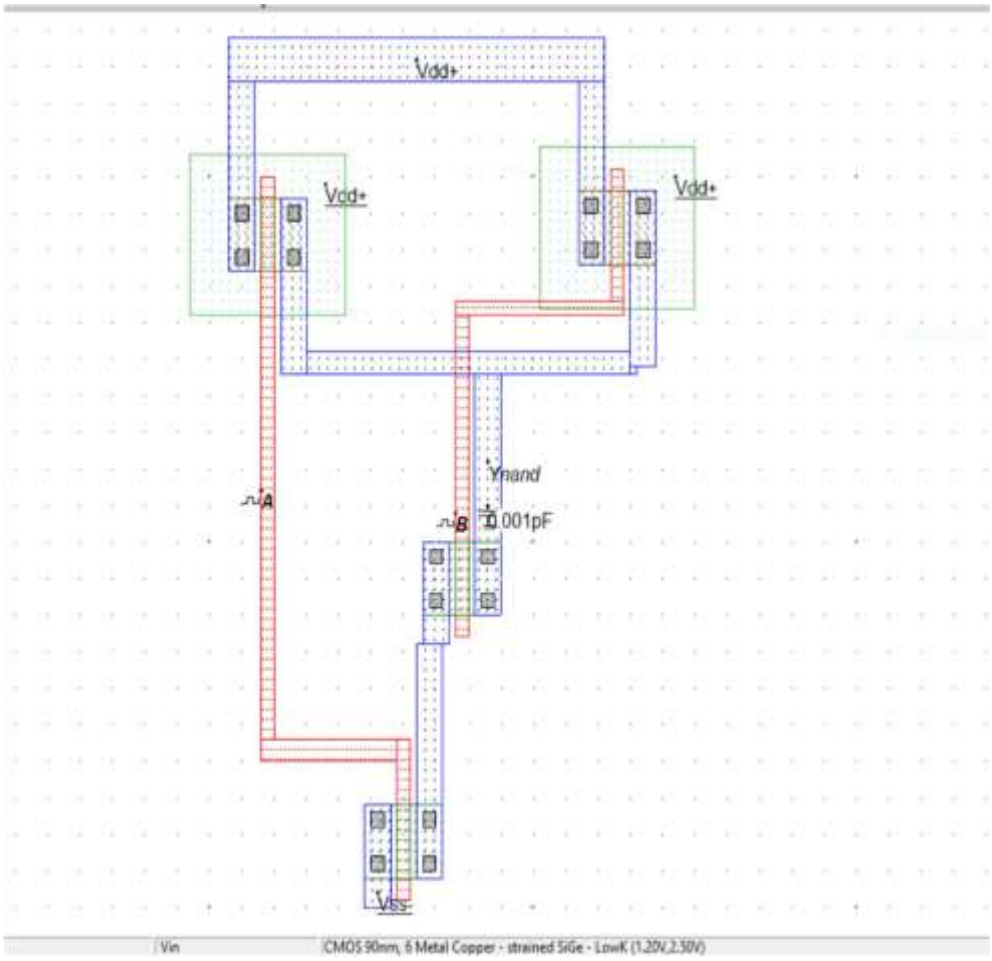
A	B	Ynor
0	0	1
0	1	0
1	0	0
1	1	0

4. Or Gate

A	B	Yor
0	0	0
0	1	1
1	0	1
1	1	1

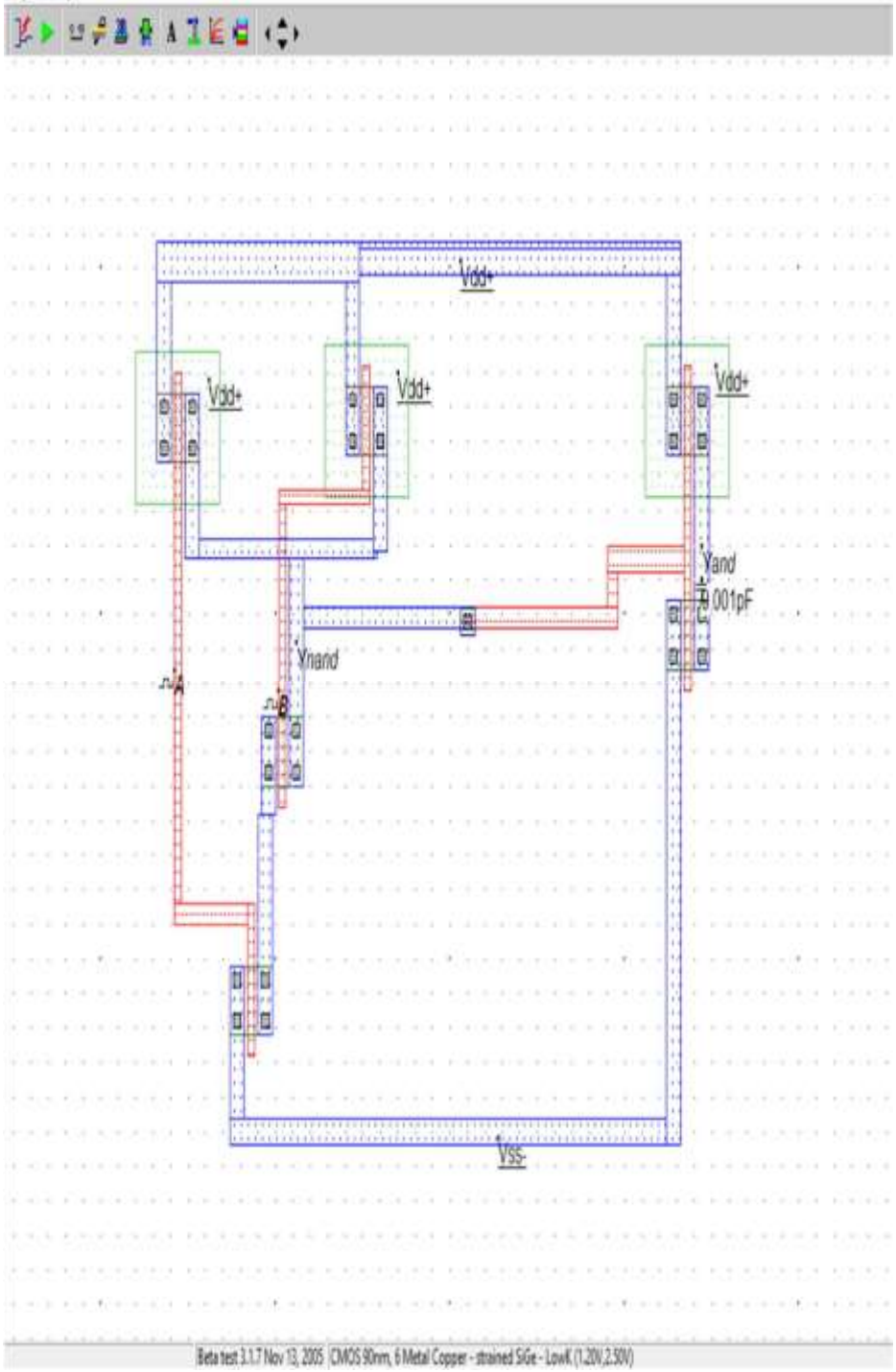
Layouts:

1.NAND Gate



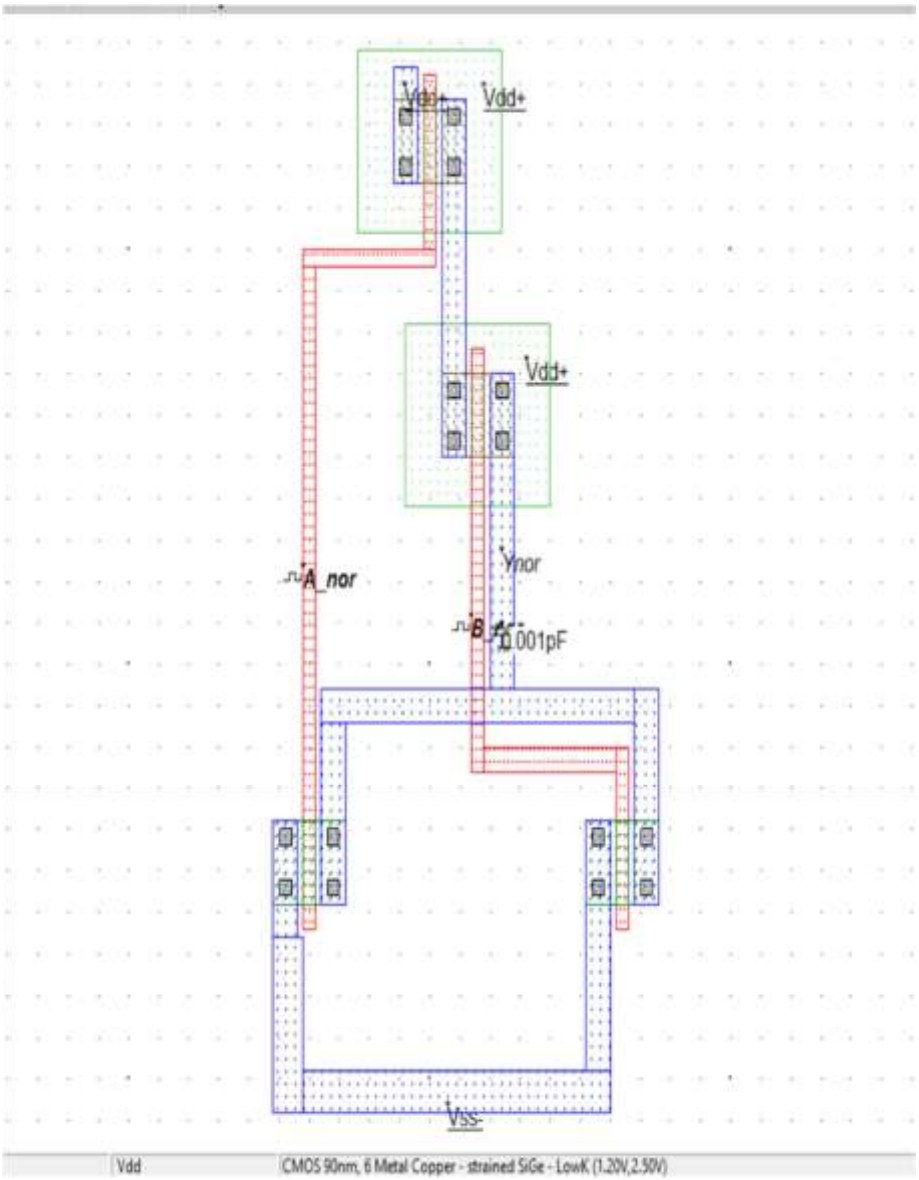
2.

AND Gate


Vdd+
Vdd+
Vdd+
Vdd+
Yand
0.001pF
Ynand
A
B
Vss-
Beta test 3.1.7 Nov 13, 2005 CMOS 90nm, 6 Metal Copper - strained SiGe - LowK (1.20V,2.50V)

3.

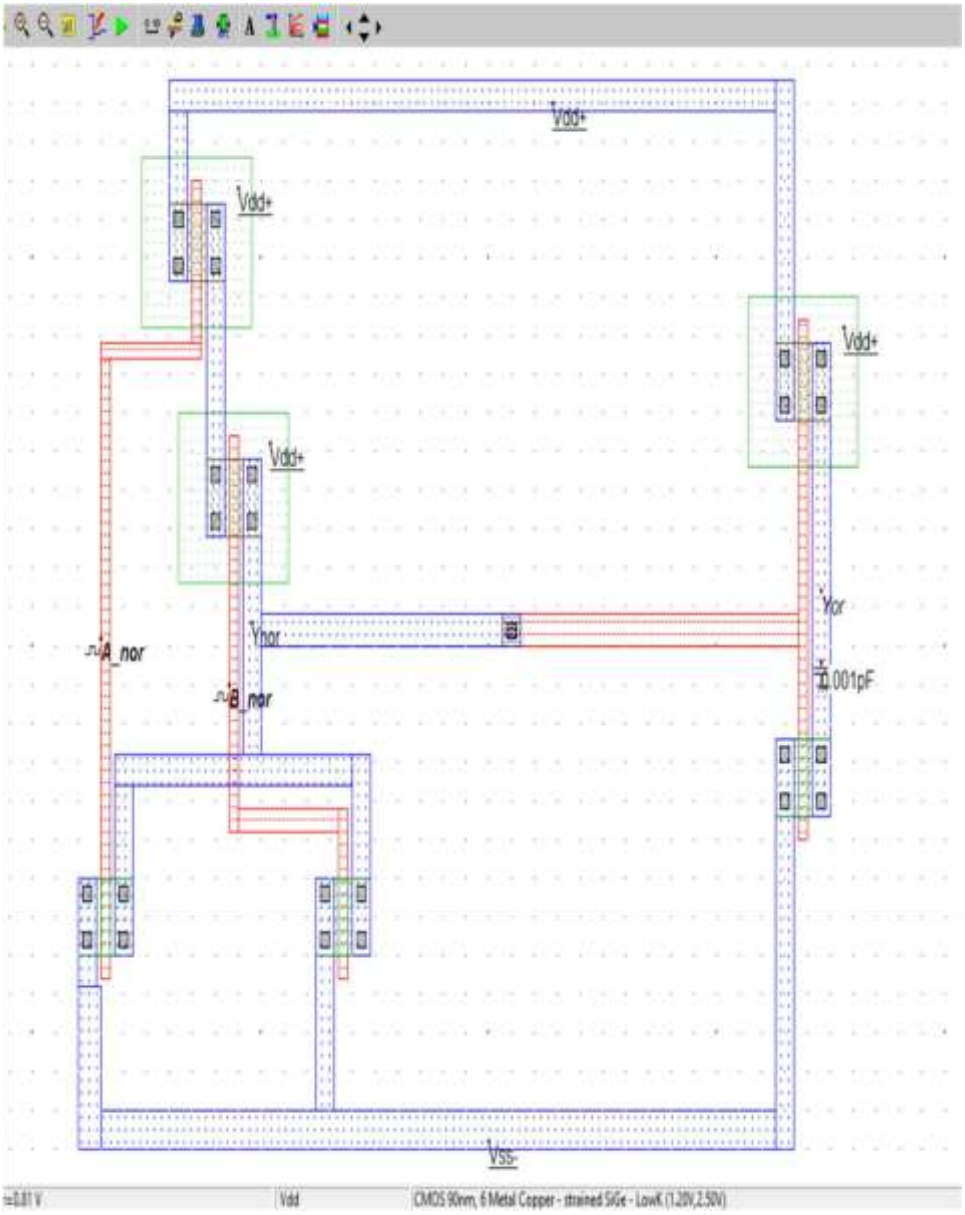
NOR Gate


Vdd+
Vdd+
Vdd+
Ynor
A_nor
B_nor
0.001pF
Vss-
Vdd
CMOS 90nm, 6 Metal Copper - strained SiGe - LowK (1.20V,2.50V)

4.

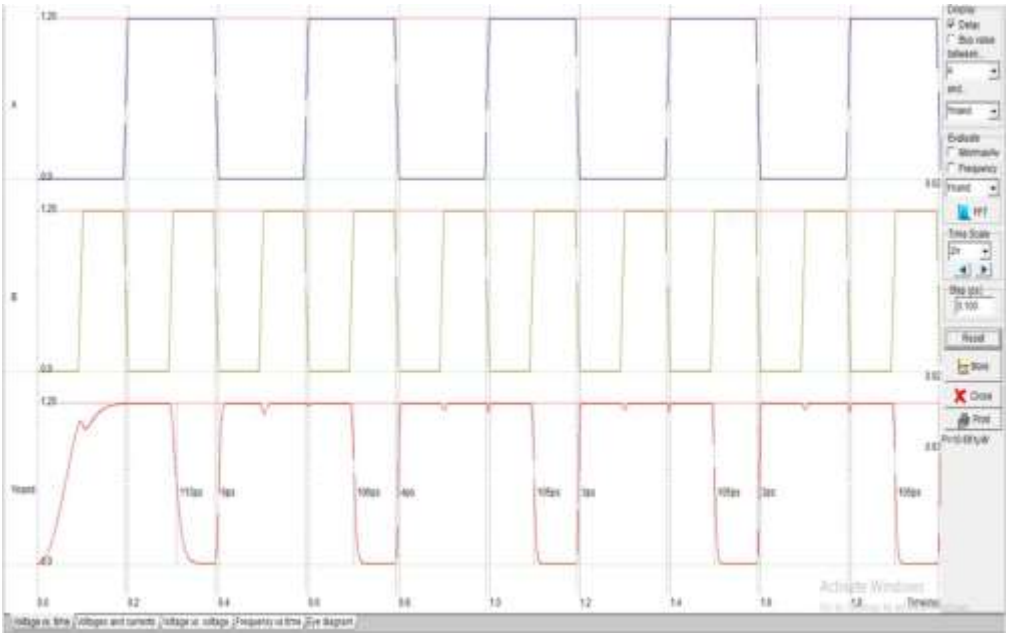
OR Gate

5.

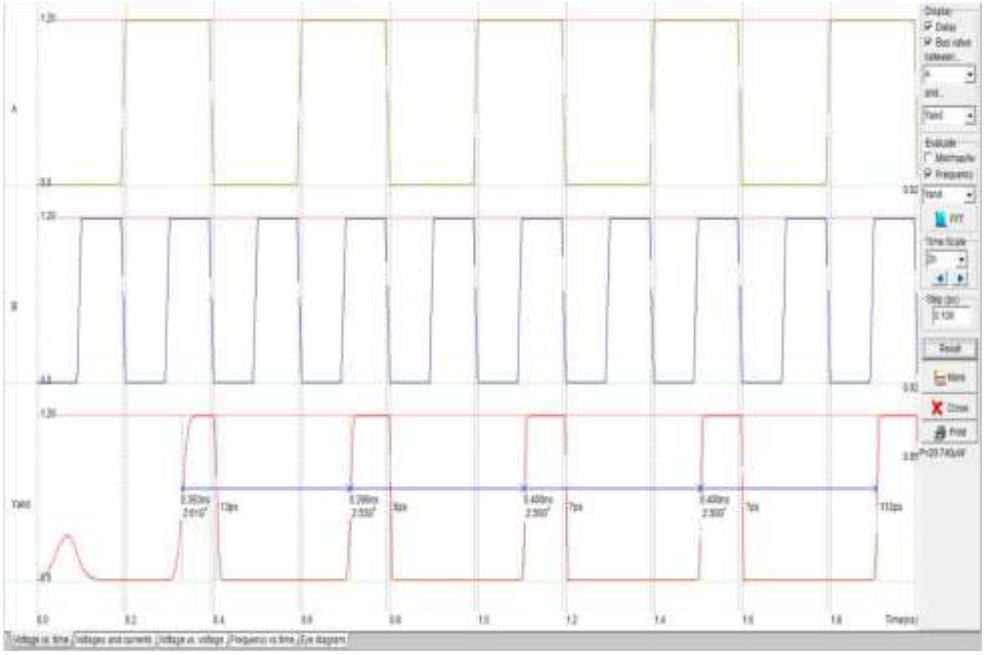

Vdd+
Vdd+
Vdd+
Vdd+
A_nor
B_nor
Yhor
Yor
0.001pF
Vss-
=0.81 V
Vdd
CMOS 90nm, 6 Metal Copper - strained SiGe - LowK (1.20V,2.50V)

Waveforms :

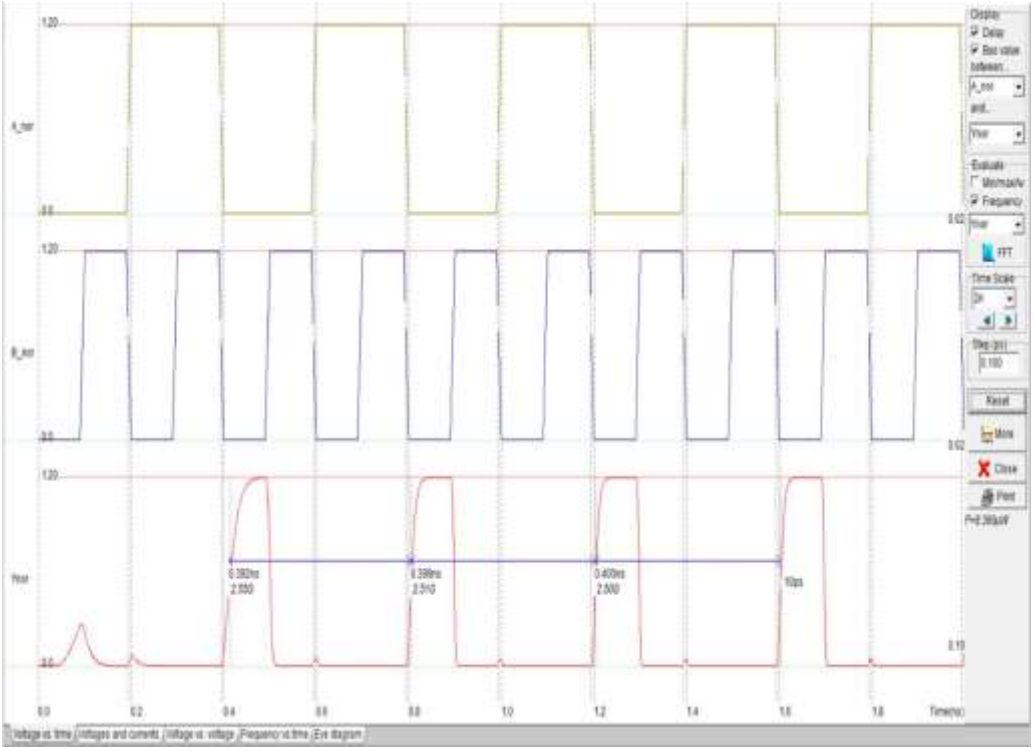
1. NAND Gate



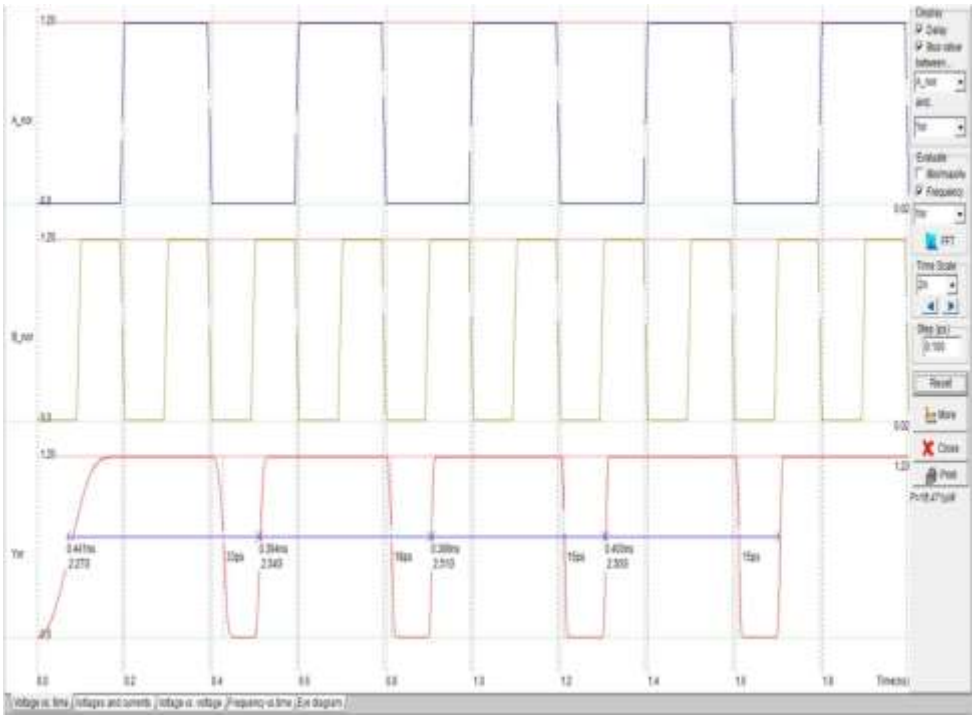
2. AND Gate



3. NOR Gate



4. OR Gate



Conclusion:-

Thus we have :

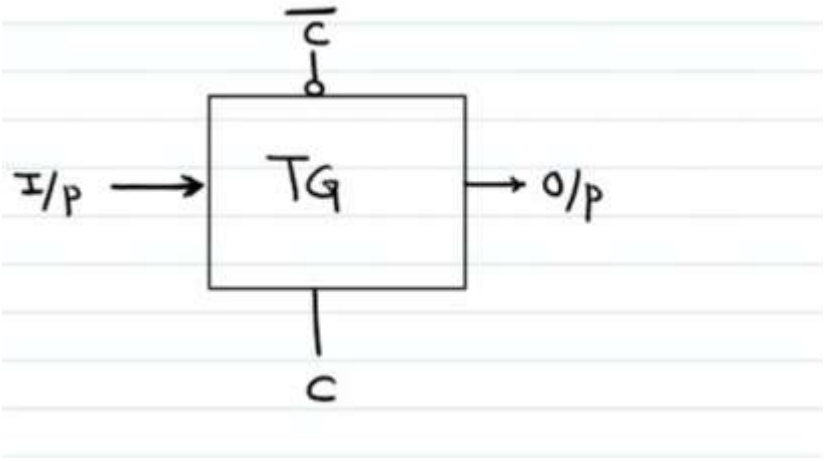
1) Drawn the LAYOUT for CMOS NAND-AND , NOR-OR Gates using 90 nm Foundry.

2) Simulated the LAYOUT to observe waveforms & verified its functionality as per TRUTH-

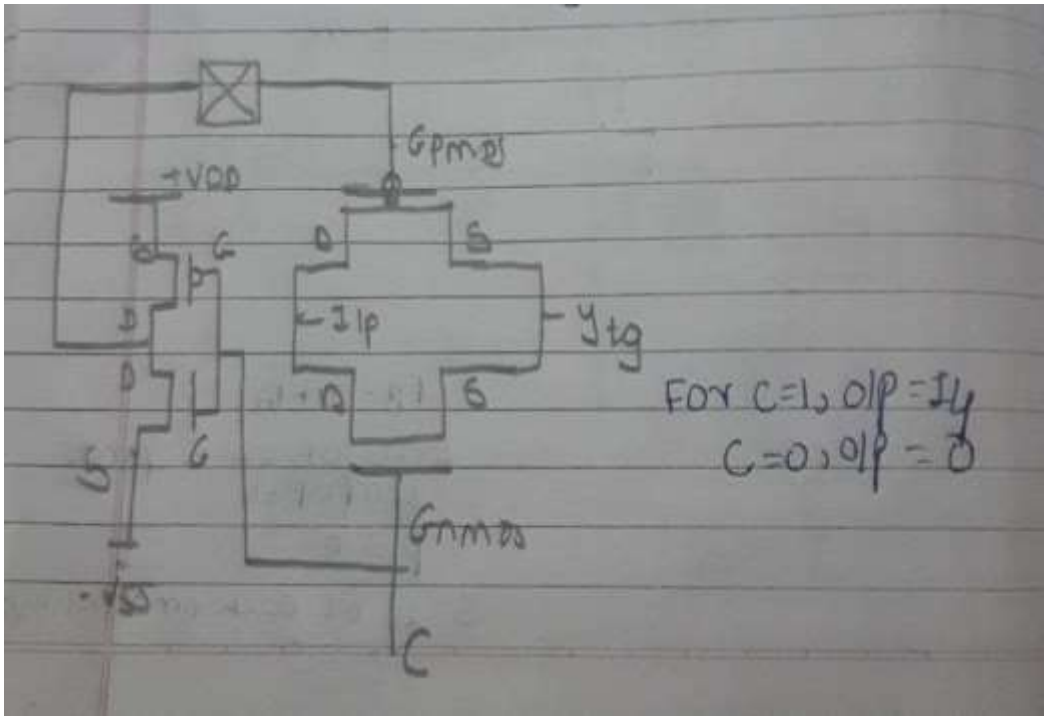
TABLE.

Class	BE-6
Roll Number	42247
Assignment Number	B.1.c
Assignment Name	Transmission Gate
Date of performance	27/10/2021

Block Diagram:-



MOSFET-LEVEL SCHEMATIC :

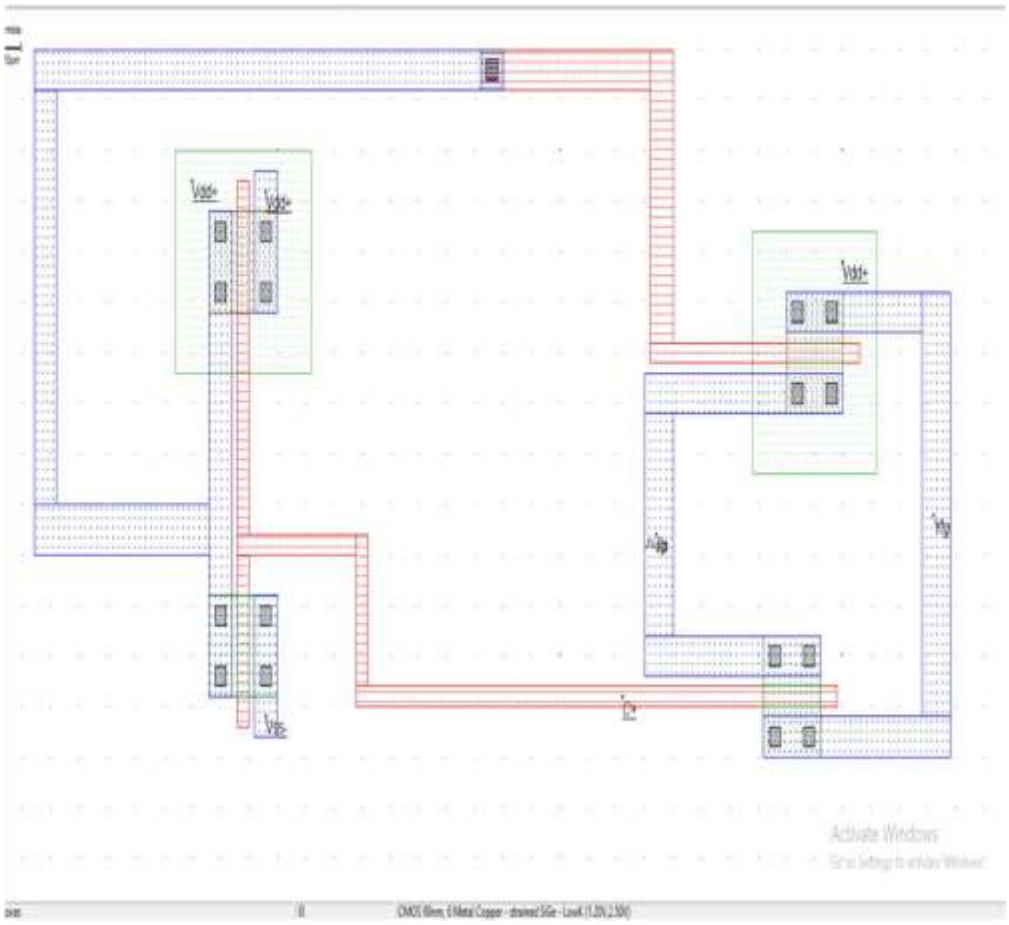


Truth Table:-

C	Y_{tg}
1	Input =1 / 0 , Strong -1 / Strong -0
0	I/P = Don't Care

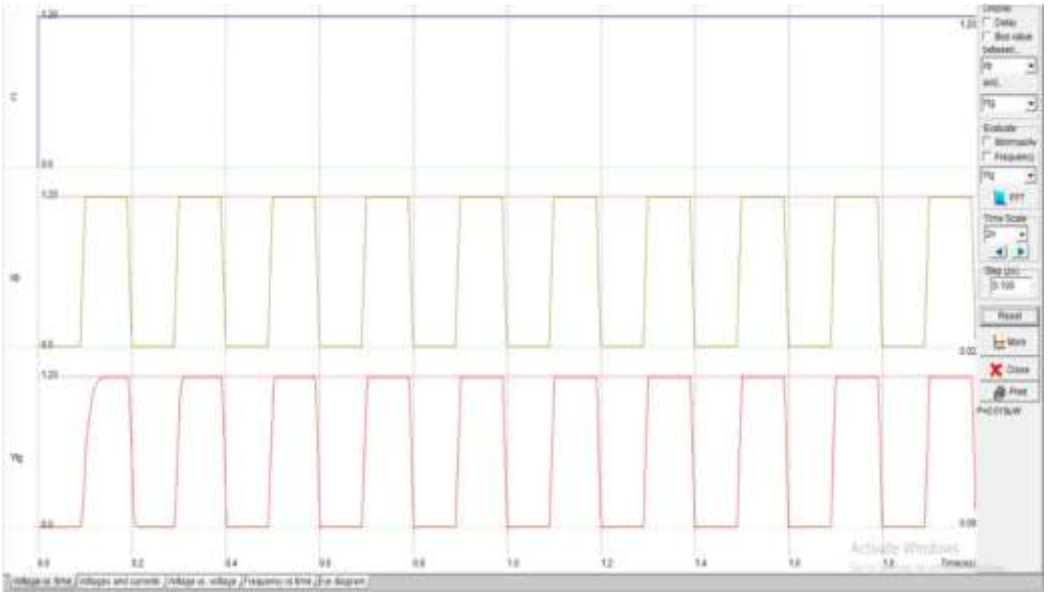
Layout:

1. TG layout:



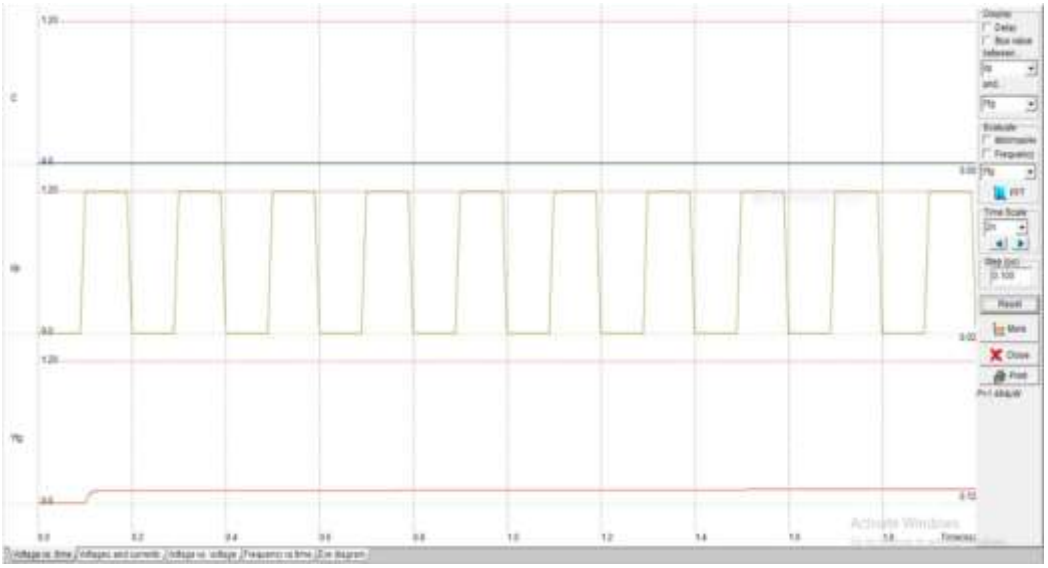
Waveform:

1: TG with C=1



Y_{tg} follows I/P , Y_{tg} = Strong-1 / Strong-0 for I/P = 1 / 0

2: TG with C=0



Y_{tg} does not follows I/P

Conclusion:-

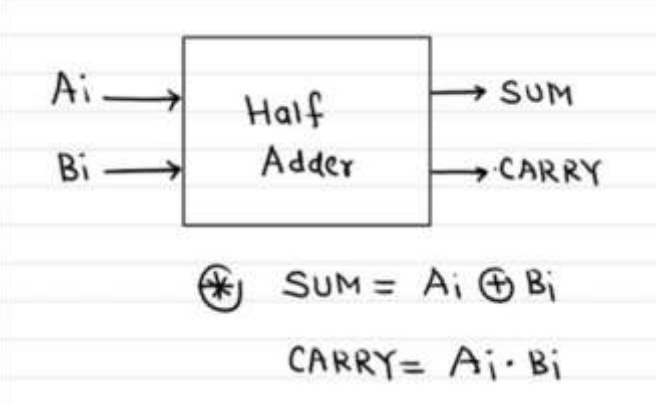
Thus we have :

1) Drawn the LAYOUT for TRANSMISSION GATE using 90 nm Foundry.
2) Simulated the LAYOUT to observe waveforms & verified its functionality as per TRUTH-TABLE.

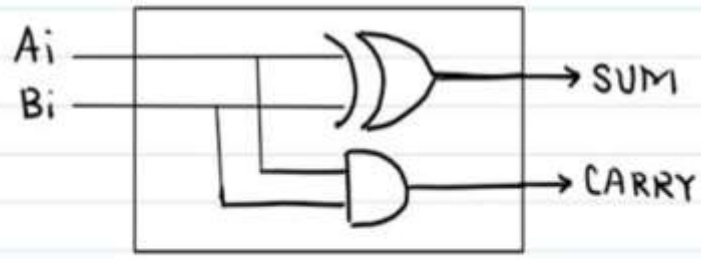
3) TG gives STRONG-1 ,STRONG-0 for I/P = 1 / 0 Respectively.

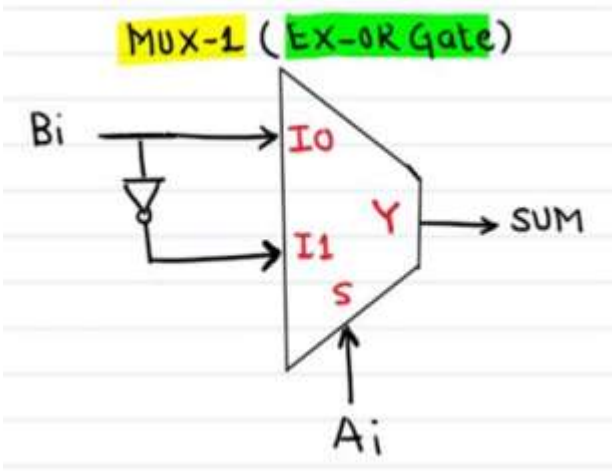
Class	BE-6
Roll Number	42247
Assignment Number	B.1.d
Assignment Name	Half-Adder Using MUX Using TG
Date of performance	27/10/2021

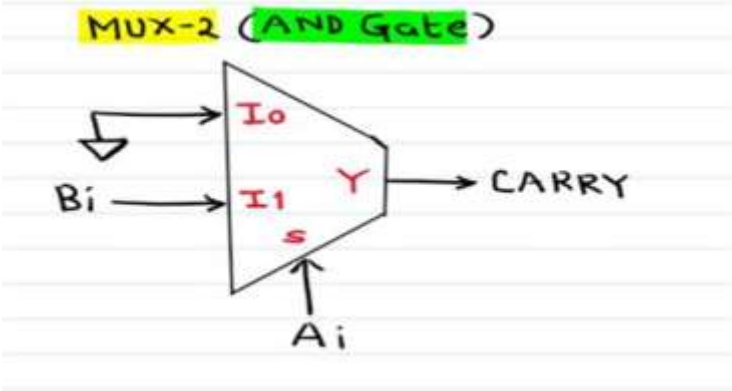
Block Diagrams:-



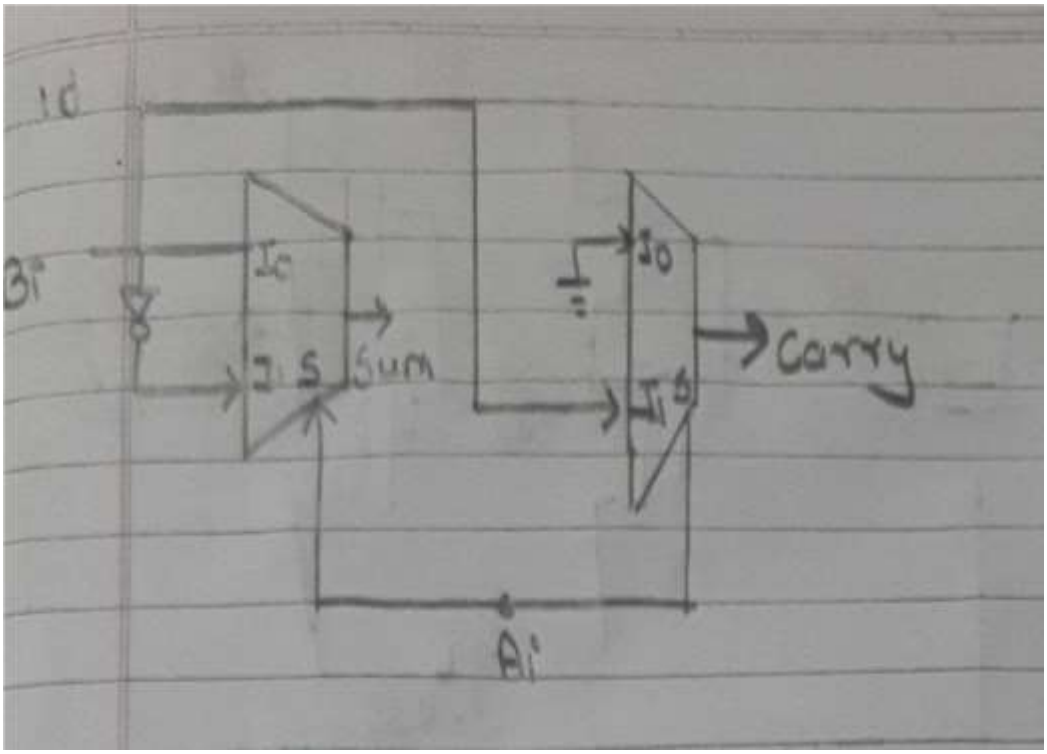
Gate-Level Schematic:-




MUX-1 (EX-OR Gate)
Bi
I0
I1
Y
SUM
S
Ai



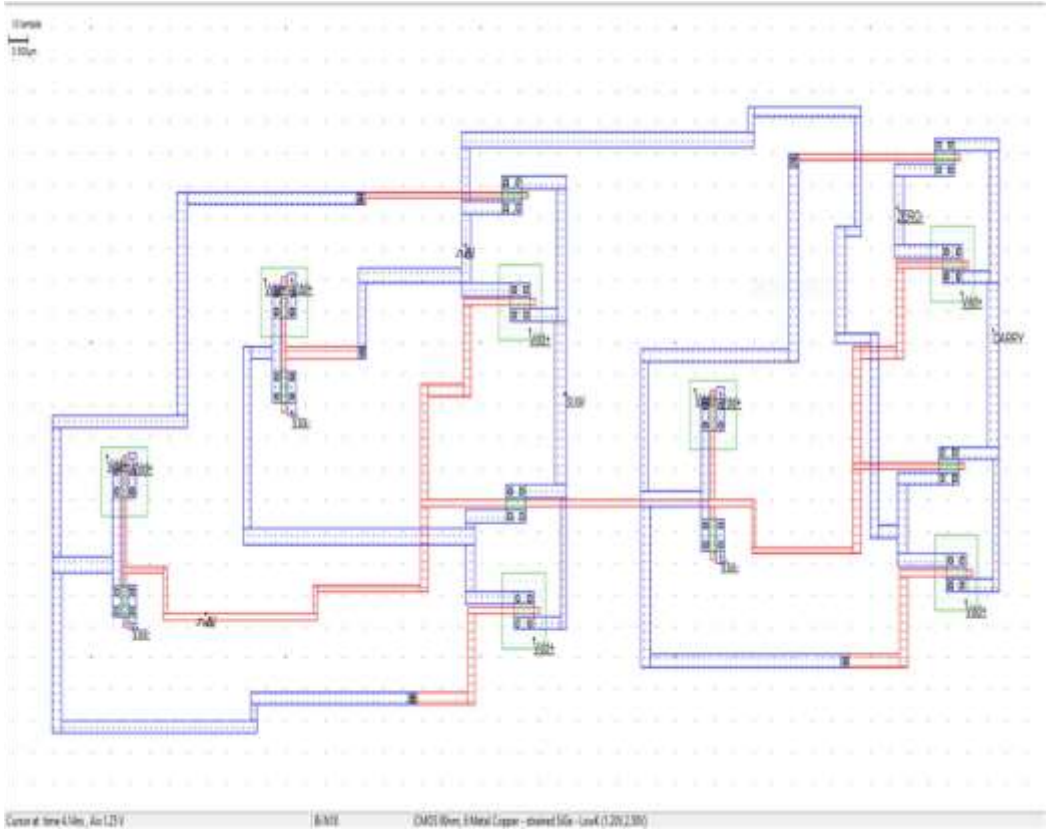
BLOCK-LEVEL SCHEMATIC :



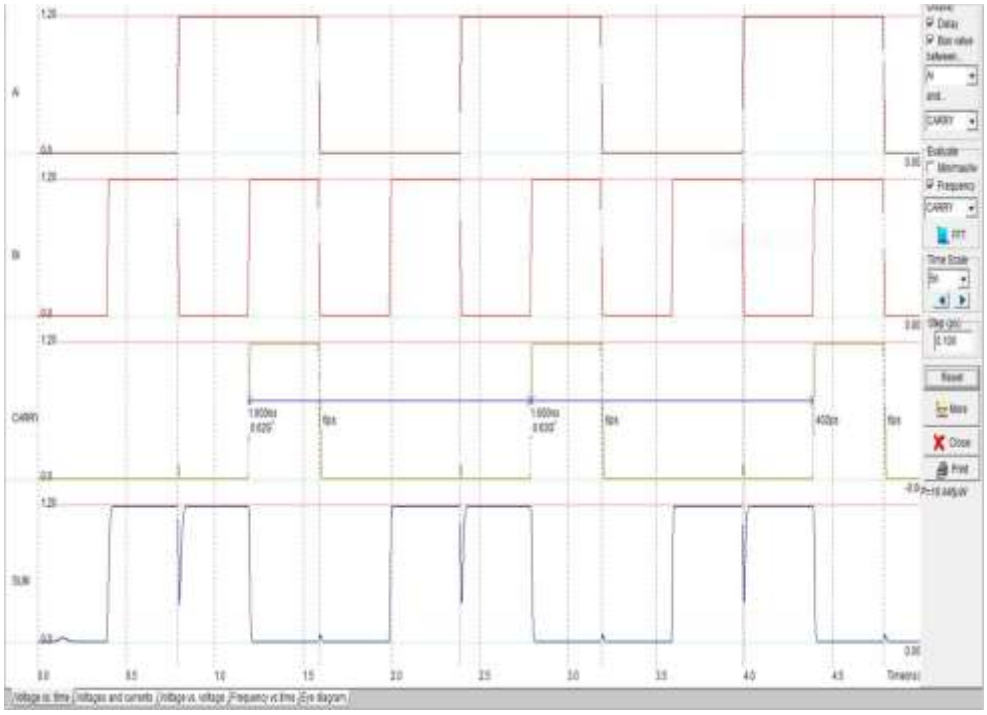
Truth Table:-

Ai	Bi	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Layouts:



Waveform:



Conclusion:-

Thus, we have :

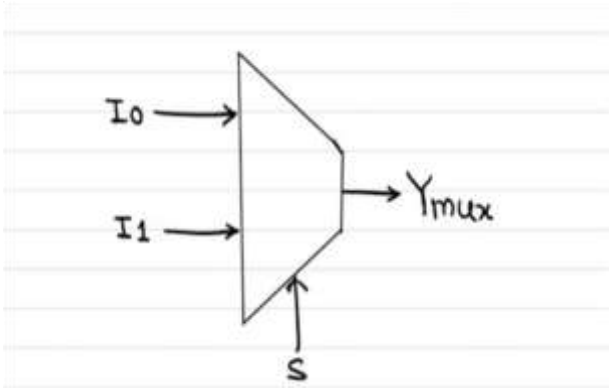
1) Drawn the LAYOUT for HALF-ADDER using 90 nm Foundry.
2) HALF-ADDER is implemented as a combination of EX-OR Gate & AND Gate.
3) EX_OR Gate & AND Gate are Implemented as a Modification of TWO 2:1 MUX's.
4) Each MUX is implemented as a combination of 2 TG's.

5) Simulated the LAYOUT to observe waveforms & verified its functionality as per TRUTH-

TABLE.

Class	BE-6
Roll Number	42247
Assignment Number	B.2
Assignment Name	2:1 Mux using Transmission Gate
Date of performance	13/10/2021

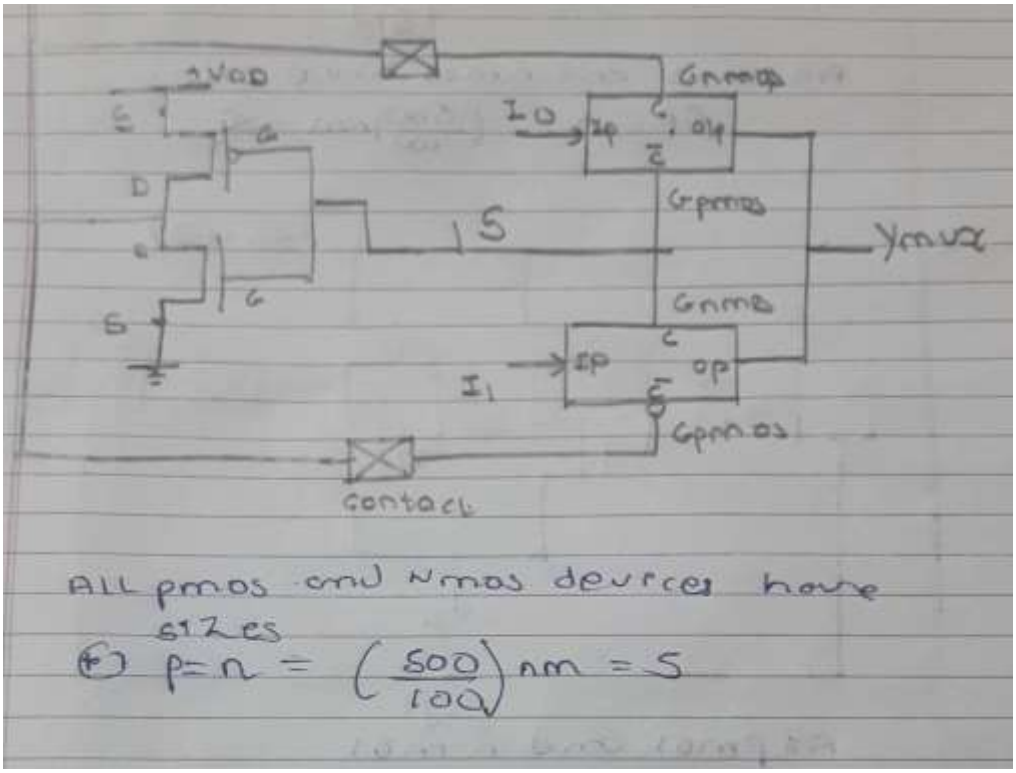
Block Diagram:-



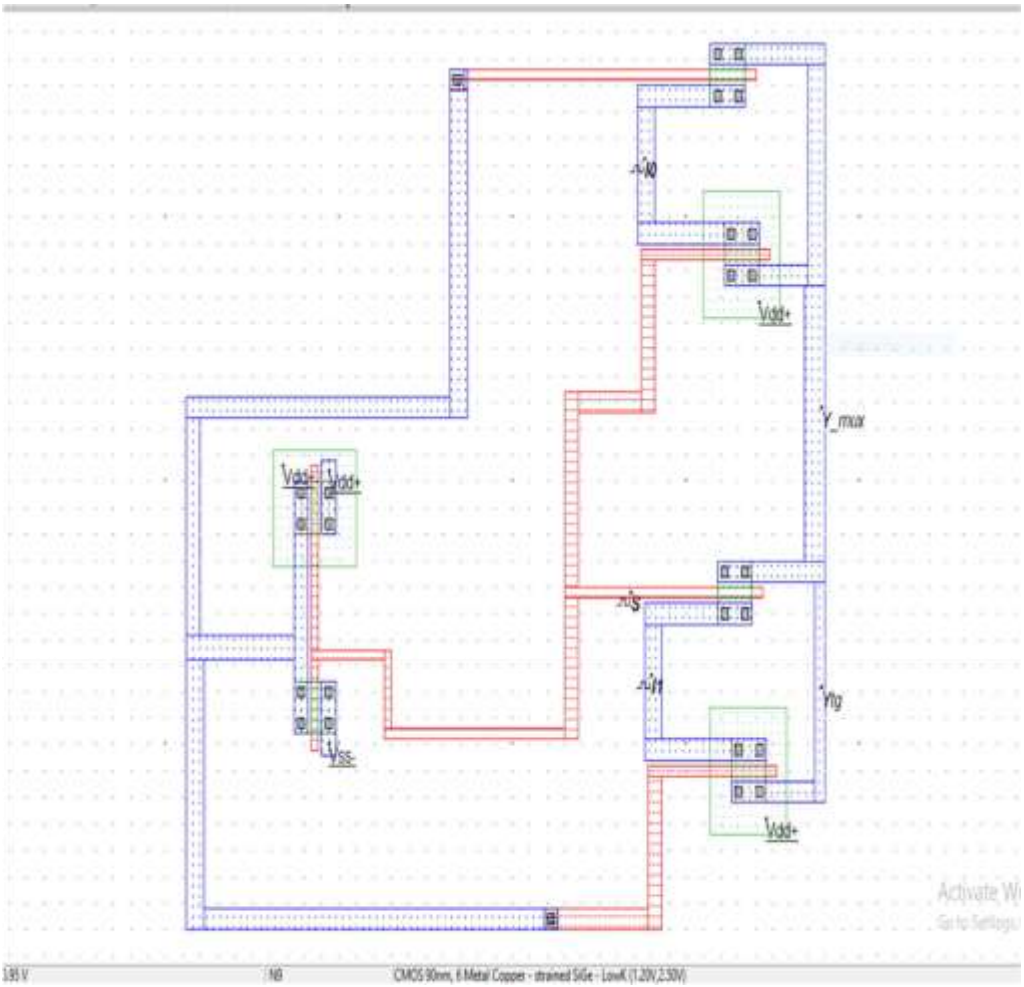
Truth Table:-

S	Ymux
0	I0
1	I1

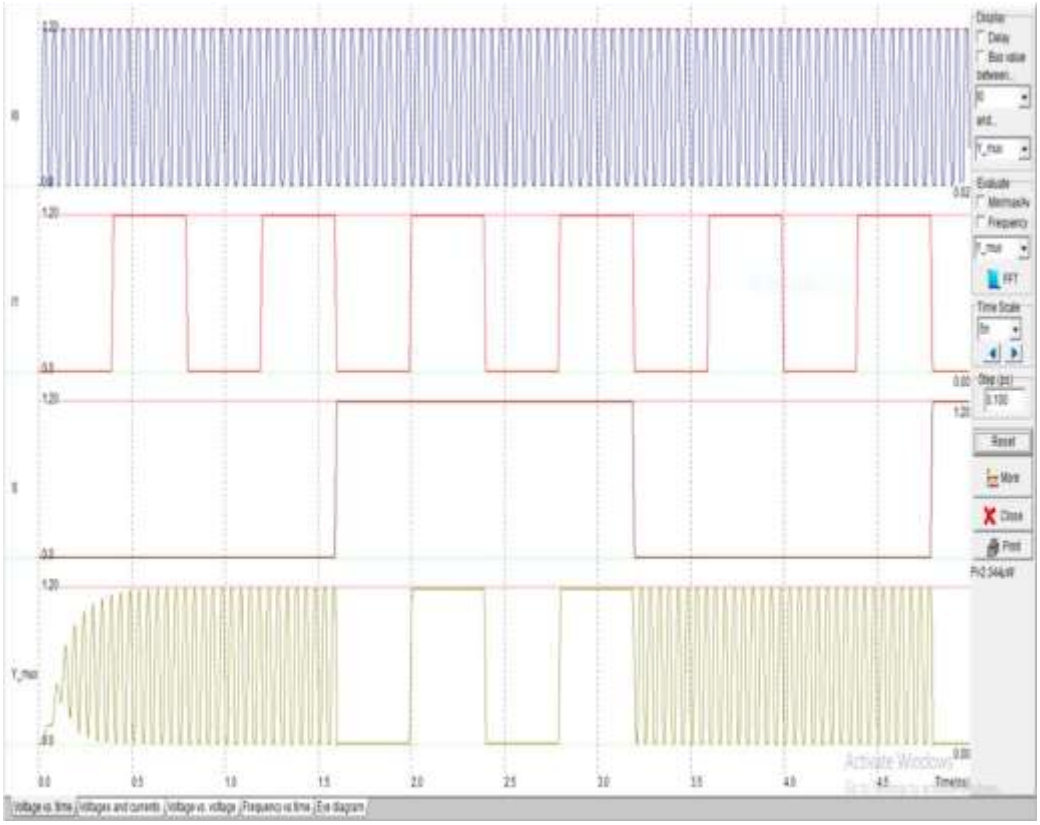
MOSFET-LEVEL SCHEMATIC :-



Layout:


Vdd+
Y_mux
Vdd+
Vss-
Yfg
Vdd+
Activate W
CMOS 90nm, 6 Metal Copper - strained SiGe - LowK (1.20V,2.50V)

Waveform:



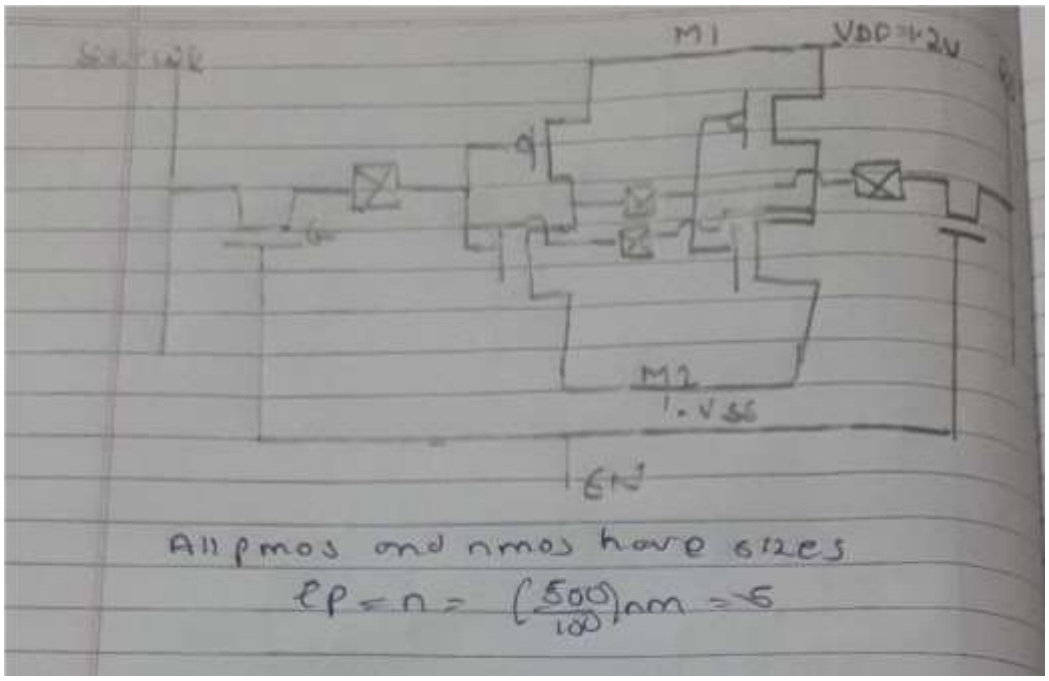
Conclusion:-

Thus we have :

1) Drawn the LAYOUT for 2:1 MUX using 90 nm Foundry.
2) MUX is implemented as a combination of 2 Transmission Gates.
3) Simulated the LAYOUT to observe waveforms & verified its functionality as per TRUTH-TABLE.

Class	BE-6
Roll Number	42247
Assignment Number	B.3.a
Assignment Name	1-bit SRAM cell using NMOS Switches
Date of performance	10/11/2021

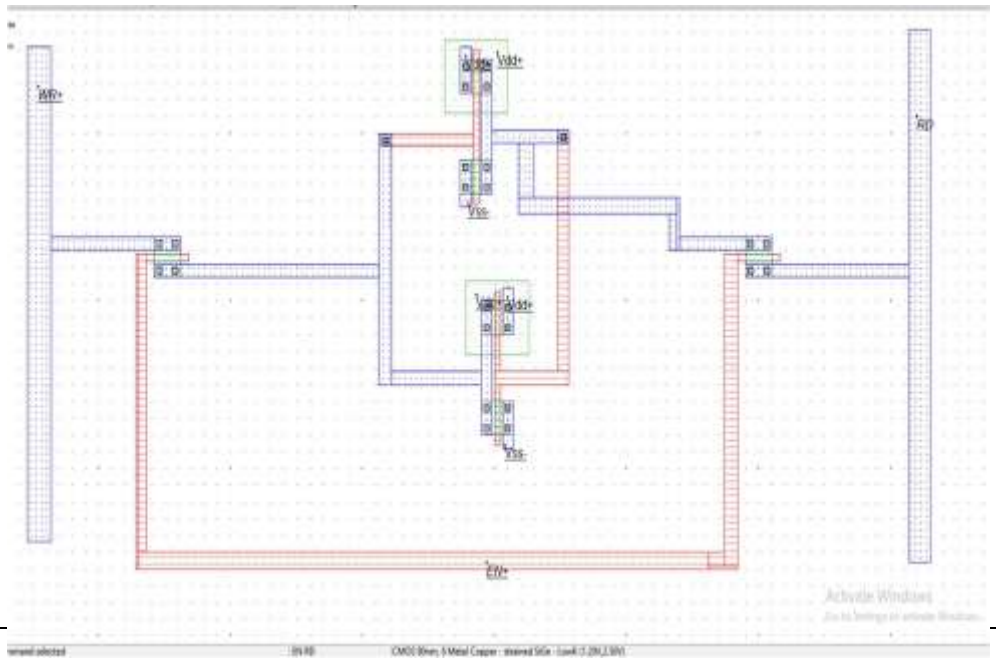
MOSFET-LEVEL SCHEMATIC :-



Truth Table:-

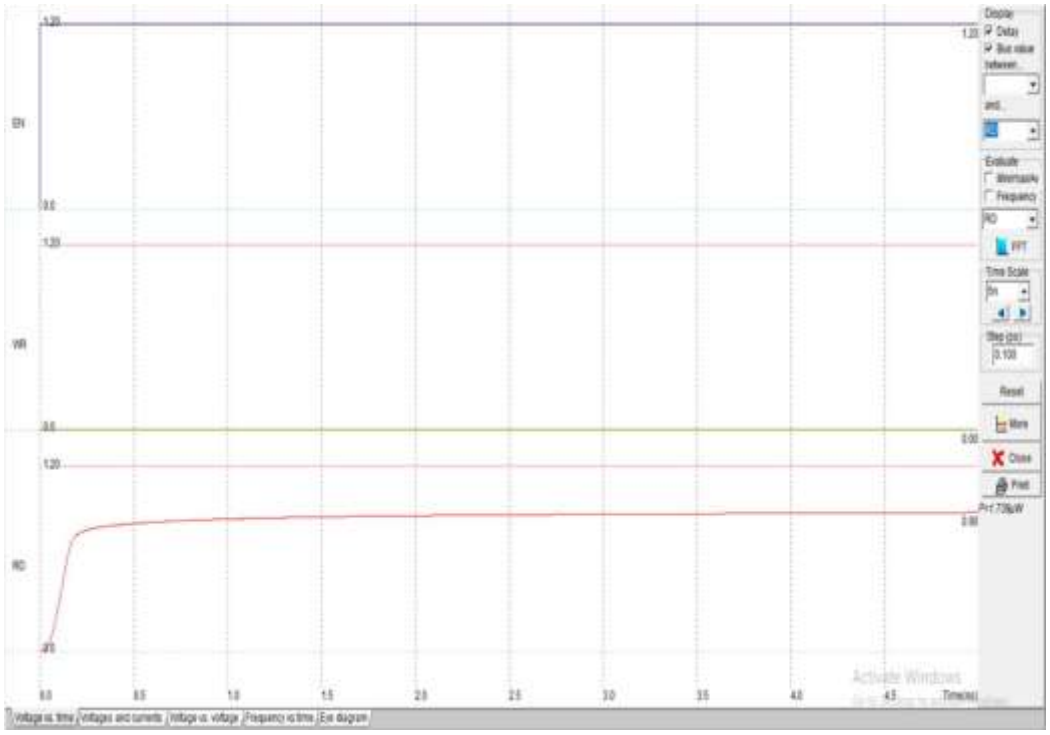
C	WR	RD
1	0	Weak-1
1	1	Strong-0
0	X	0/Hold

Layout:

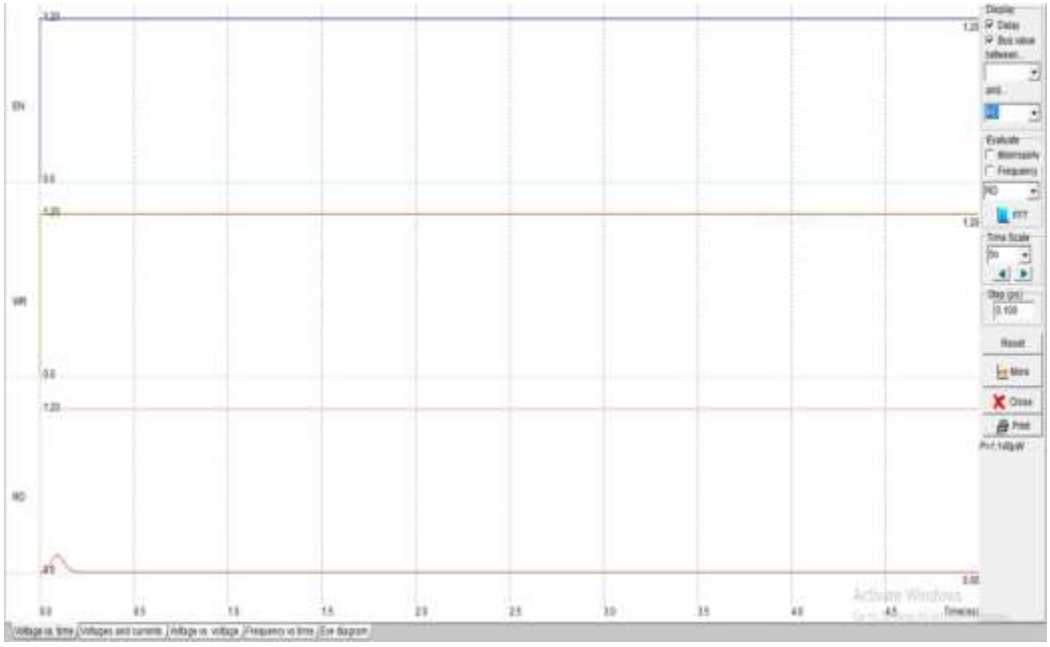


Waveform:

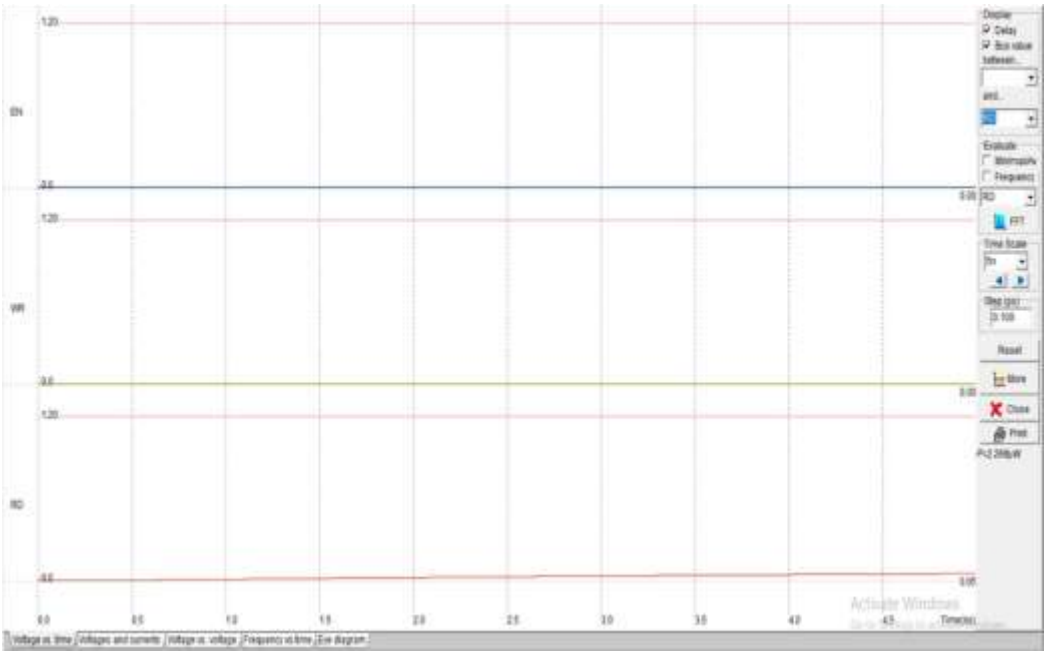
1: C=1 , WR=0 , RD =Weak-1:



2: C=1, WR=1, RD = Strong-0:-



3: C=0, WR = 0, RD= HOLD



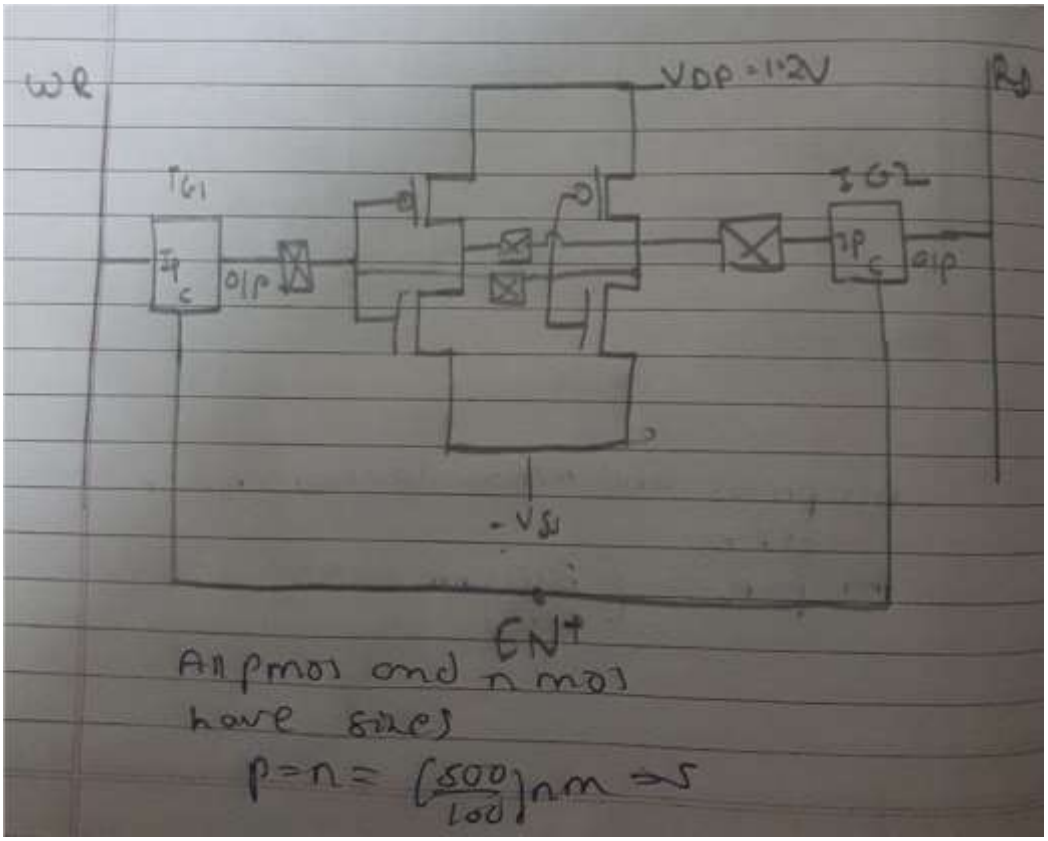
Conclusion:-

Thus we have :

1) Drawn the LAYOUT for I-bit SRAM Cell Using NMOS Switches using 90 nm Foundry.
2) Simulated the LAYOUT to observe waveforms & verified its functionality as per TRUTH-TABLE.
3) Logic-0 gets written as WEAK-1 (Unacceptable)
4) Logic-1 gets written as STRONG-0 (Acceptable)

Class	BE-6
Roll Number	42247
Assignment Number	B.3.b
Assignment Name	1-bit SRAM cell using Transmission Gate Switches
Date of performance	10/11/2021

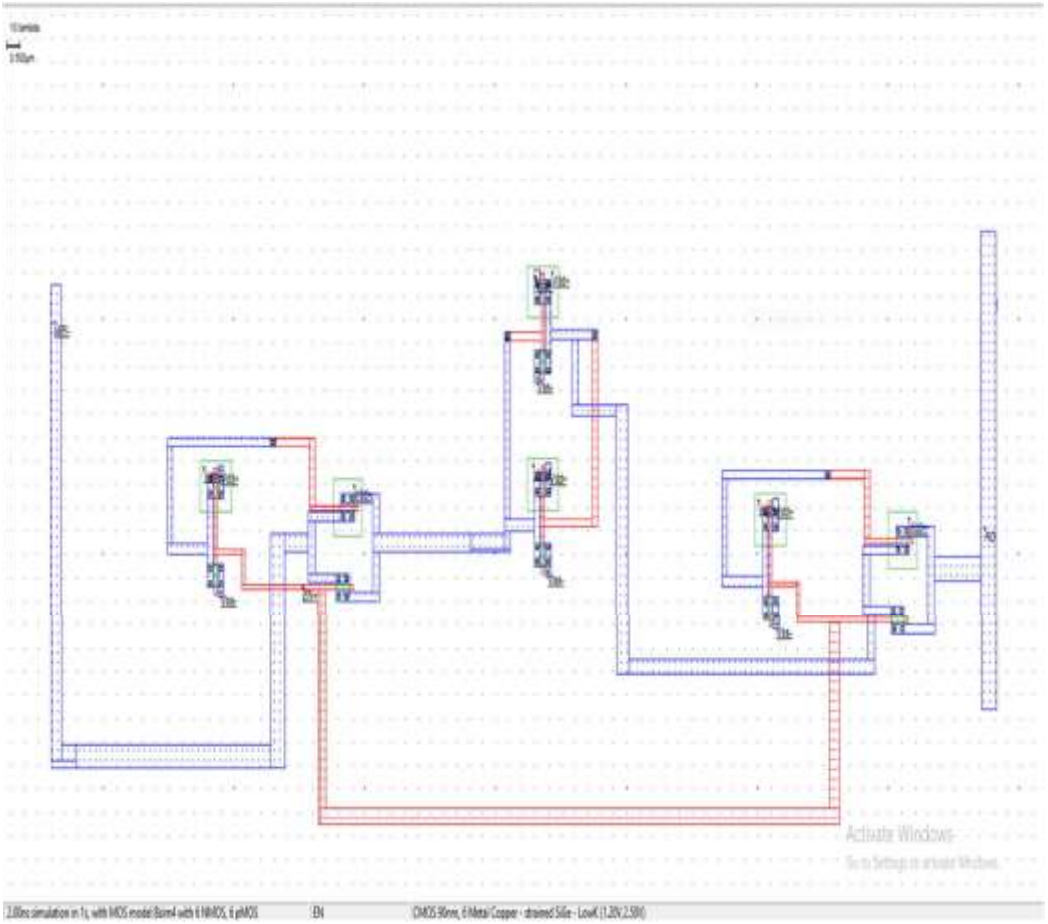
MOSFET-LEVEL SCHEMATIC :-



Truth Table:-

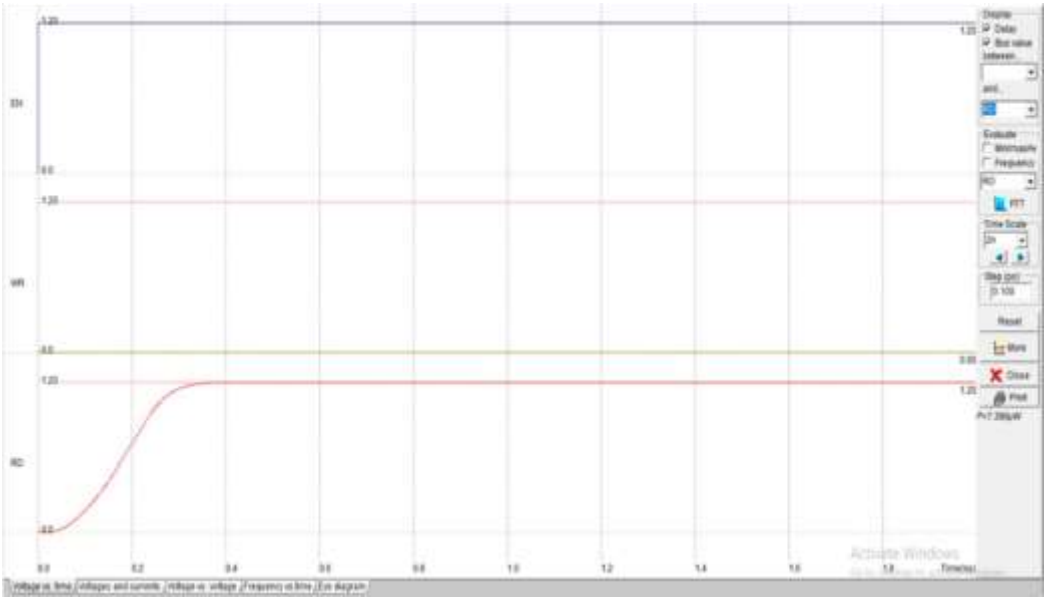
C	WR	RD
1	0	Strong-1
1	1	Strong-0
0	X	0/Hold

Layout:

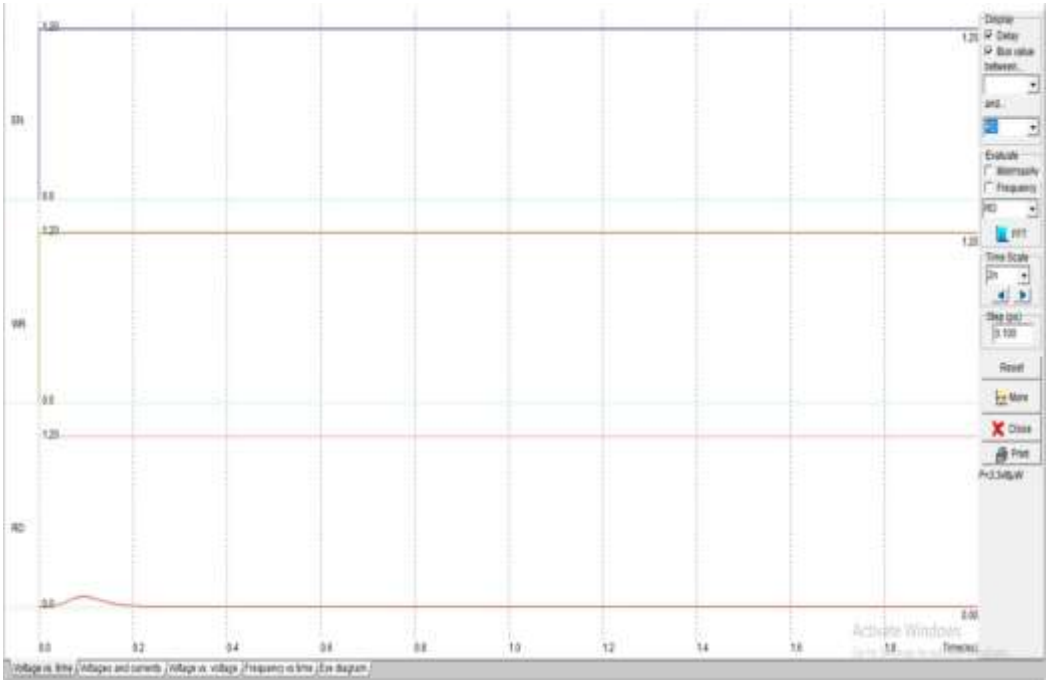


Waveforms:

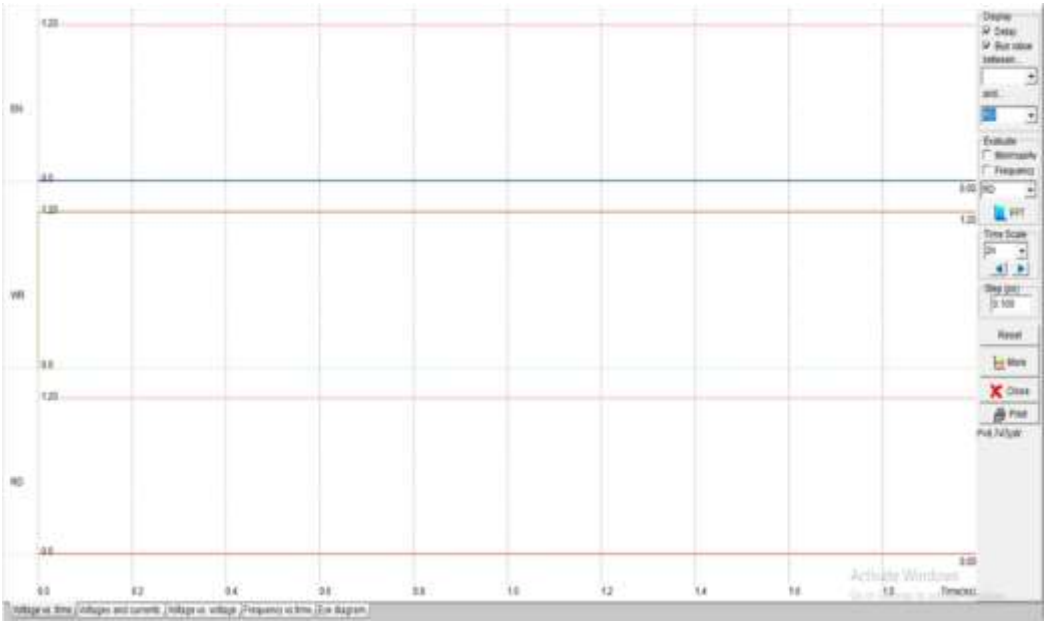
1: C=1,WR = 0 , RD = Strong-1:



2: C=1,WR=1,RD = Strong-0



3: C=0,WR=0, RD = 0



Conclusion:-

Thus we have :

1) Drawn the LAYOUT for I-bit SRAM Cell Using TG Switches using 90 nm Foundry.
2) Simulated the LAYOUT to observe waveforms & verified its functionality as per TRUTH-TABLE.
3) Logic-0 gets written as STRONG-1 (Acceptable)
4) Logic-1 gets written as STRONG-0 (Acceptable)