**Name: SHANTANU DINESH DUKARE Batch:B11 RollNo.:E43089**

**Exp. No.01:** To write VHDL code for 4bit ALU for Add, Subtract, AND, NAND, OR, XOR & XNOR simulate with test bench, synthesis, implement on FPGA

# VHDL Code:

library IEEE;

Use IEEE.STD\_LOGIC\_1164.ALL;

UseIEEE.STD\_LOGIC\_unsigned.ALL; Use IEEE.NUMERIC\_STD.ALL;

library UNISIM;

Use UNISIM.VComponents.all;

entity vlsi is

Port ( a : in STD\_LOGIC\_VECTOR(3downto 0);

b : in STD\_LOGIC\_VECTOR (3 downto 0); sel : in STD\_LOGIC\_VECTOR (2 downto 0); y : out STD\_LOGIC\_VECTOR(3downto 0));

end vlsi;

architecture Behavioral of vlsi is begin

process(a,b ,sel) begin case sel is

null; end case;

end process; end Behavioral

when"000"=>y<=a+b; when"001"=>y<=a-b; when"010"=>y<=a and b; when"100"=>y<=a nand b; when"011"=> y<=a or b; when"101"=> y<=a nor b; when"110"=>y<=not a; when"111"=>y<=a; whenothers=>

# Test Bench:

LIBRARY ieee; USEieee.std\_logic\_1164.ALL; USE ieee.numeric\_std.ALL;

ENTITY ssss IS ENDssss;

ARCHITECTURE behavior OF ssss IS COMPONENT

vlsiPORT

(a : IN std\_logic\_vector(3 downto 0);

B : IN std\_logic\_vector(3 downto 0); sel : IN std\_logic\_vector(2 downto 0);y :

OUT

std\_logic\_vector(3 downto 0) );

END COMPONENT;

--Inputs

signal a : std\_logic\_vector(3 downto 0) := (others => '0'); signal b : std\_logic\_vector(3 downto 0) := (others => '0'); signal sel : std\_logic\_vector(2 downto 0) := (others => '0');

--Outputs

signal y : std\_logic\_vector(3 downto 0);

-- No clocks detected in port list. Replace below with

-- appropriate port name

-- constant \_period : time := 10 ns; BEGIN

-- Instantiate the Unit Under Test (UUT)uut:

vlsi PORT MAP

( a => a, b => b, sel => sel,y => y );

-- Clock process definitions

--<clock>\_process :process Begin

<clock>= '0';

wait for<clock> \_period/2;

<clock>= '1';

wait for<clock> \_period/2; end process;

-- Stimulus process stim\_proc:

processbegin

a<="01 01";

b<="01 00";

sel<="0 00";

-- hold

reset state for 100 ns.wait for 100 ns; a<="0101";

b<="0100"; sel<="001";

-- hold

reset state for 100ns.wait for 100 ns; a<="01 01";

b<="0100"; sel<="010";

-- hold

reset state for 100ns.wait for 100 ns; a<="0101";

b<="01 00";

sel<="0 11";

-- hold

reset state for 100ns.wait for 100 ns; a<="01 01";

b<="0100"; sel<="100";

-- hold reset state for 100ns.wait for 100 ns; a<="01 01";

b<="0100"; sel<="101";

-- hold

reset state for 100ns.wait for 100 ns; a<="0101";

b<="01 00";

sel<="1 7 10";

-- hold reset state for 100ns.wait for 100 ns; a<="01 01";

b<="0100"; sel<="111";

-- hold reset state for 100ns.wait for 100 ns;

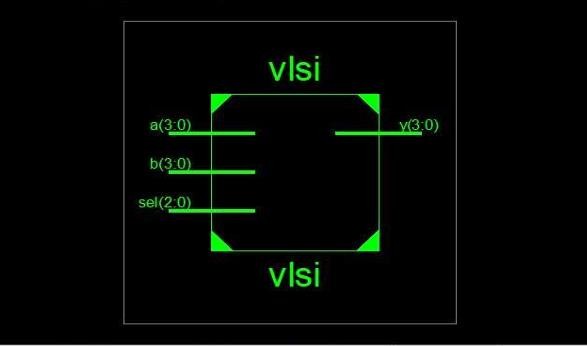
-- END;

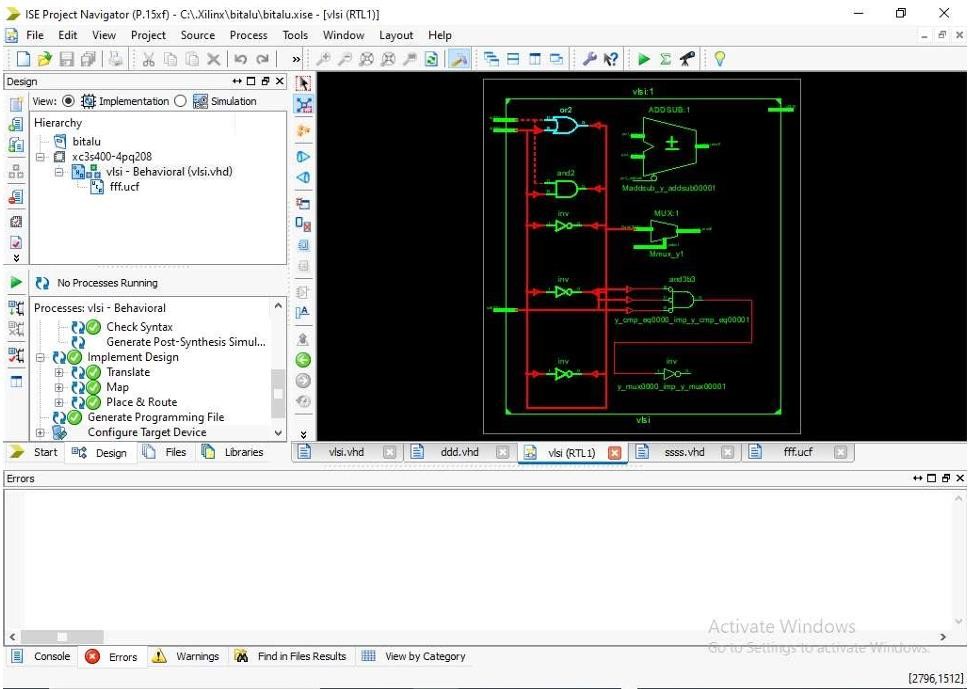
wait for \_period\*10;

-- insert stimulus here wait; end process

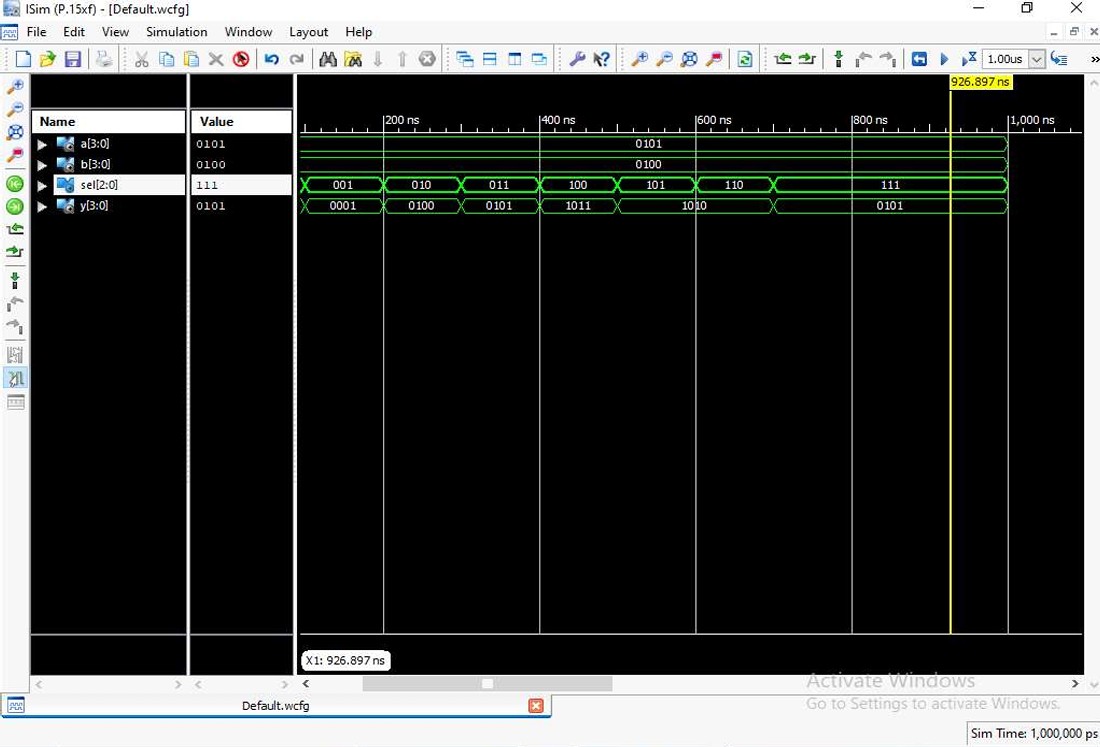
# Output:

Entity Diagram:





Waveform:



# Final Report:

Final Results: -

RTL TopLevel Output File Name: vlsi.ngr Top Level Output File Name: vlsi

|  |  |
| --- | --- |
| Output Format: |  |
| NGC  Optimization Goal: | Speed |
| Keep Hierarchy: | No |
| Design Statistics # IOs | 15 |
| Cell Usage # BELS: | 34 |
| #LUT2: | 5 |
| #LUT3: | 10 |
| #LUT4: | 6 |
| #MUXF5: | 9 |
| #MUXF6: | 4 |
| #IO Buffers: | 15 |
| #IBUF: | 11 |
| #OBUF: | 3 |

# UCF File:

net a(0) loc="p101"; net a(1) loc="p100"; net a(2) loc="p97"; net a(3) loc="p96"; net b(0) loc="p87"; net b(1) loc="p86"; net b(2) loc="p85"; net b(3) loc="p81"; net sel(0) loc="p78"; net sel(1) loc="p77"; net sel(2) loc="p74"; net y(0) loc="p168"; net y(1) loc="p171"; net y(2) loc="p172"; net y(3) loc="p175";

Release 14.1- xst P.15xf (nt64) Copyright (c) 1995-2012 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to xst/projnav.tmp Total REAL time to Xst completion:

0.00

secsTotal CPU time to Xst completion:

0.11 secs

--> Reading design: vlsi.prj

# Name: SHANTANU DINESH DUKARE Batch: B11 Roll No.: E43089

**Exp. No.02:** To write VHDL code for Universal shift register with mode selection input for SISO, SIPO, PISO, & PIPO simulate with test bench, synthesis, implement on FPGA.

# VHDL Code:

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL; USE ieee.numeric\_std.ALL;

ENTITY shiftreg IS END shiftreg;

ARCHITECTURE behavior OF shiftreg IS COMPONENT unishiftreg

PORT(

--Inputs

si : IN std\_logic; clk : IN std\_logic; so : OUT std\_logic;

pin : IN std\_logic\_vector(3 downto 0); po : OUT std\_logic\_vector(3 downto 0); sel : IN std\_logic\_vector(1 downto 0)

);

END COMPONENT;

signal si : std\_logic := '0'; signal clk : std\_logic := '0';

signal pin : std\_logic\_vector(3 downto 0) := (others => '0'); signal sel : std\_logic\_vector(1 downto 0) := (others => '0');

--Outputs

signal so : std\_logic;

signal po : std\_logic\_vector(3 downto 0);

--Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

uut: unishiftreg PORT MAP ( si => si,

clk => clk, so => so, pin => pin, po => po, sel => sel

);

--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;

END;

-- insert stimulus here wait;

end process;

# Test Bench:

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL; USE ieee.numeric\_std.ALL;

ENTITY tb IS

END tb;

ARCHITECTURE behavior OF tb IS COMPONENT unishiftreg PORT(

si : IN std\_logic;

clk : IN std\_logic; so : OUT std\_logic; pin : IN std\_logic\_vector(3 downto 0); po : OUT std\_logic\_vector(3 downto 0); sel : IN std\_logic\_vector(1 downto 0) );

END COMPONENT;

--Inputs

signal si : std\_logic := '0'; signal clk : std\_logic := '0';

signal pin : std\_logic\_vector(3 downto 0) := (others => '0'); signal sel : std\_logic\_vector(1 downto 0) := (others => '0');

--Outputs

signal so : std\_logic;

signal po : std\_logic\_vector(3 downto 0);

--Clock period definitions constant clk\_period : time := 10 ns;

BEGIN

uut: unishiftreg PORT MAP ( si => si,

clk => clk, so => so, pin => pin, po => po, sel => sel

);

--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

END;

sel <= "00";

si <= '1';

wait for 100 ns; sel <= "01";

si <= '1';

wait for 100 ns; sel <= "10";

pin <= "1010";

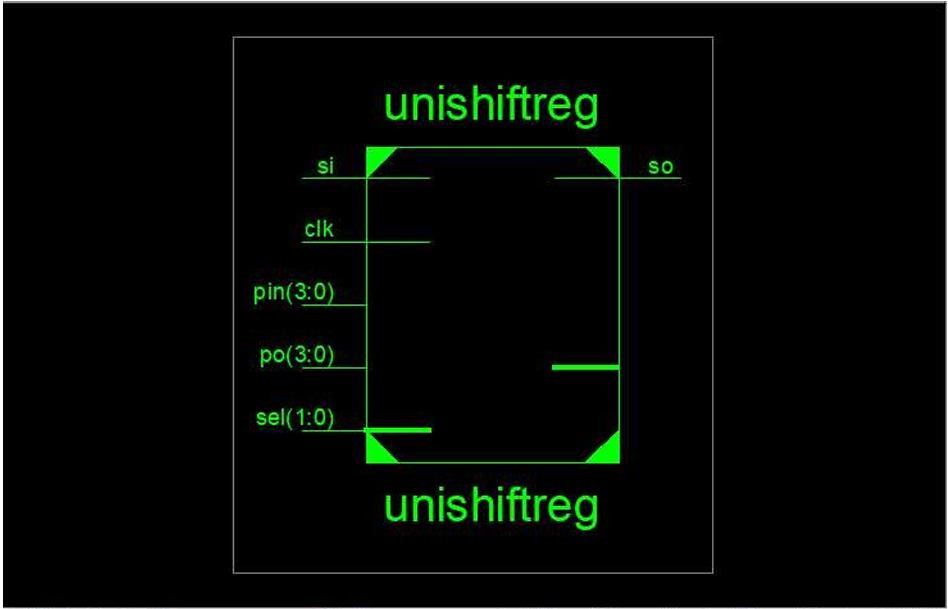
wait for 100 ns;

9 wait;

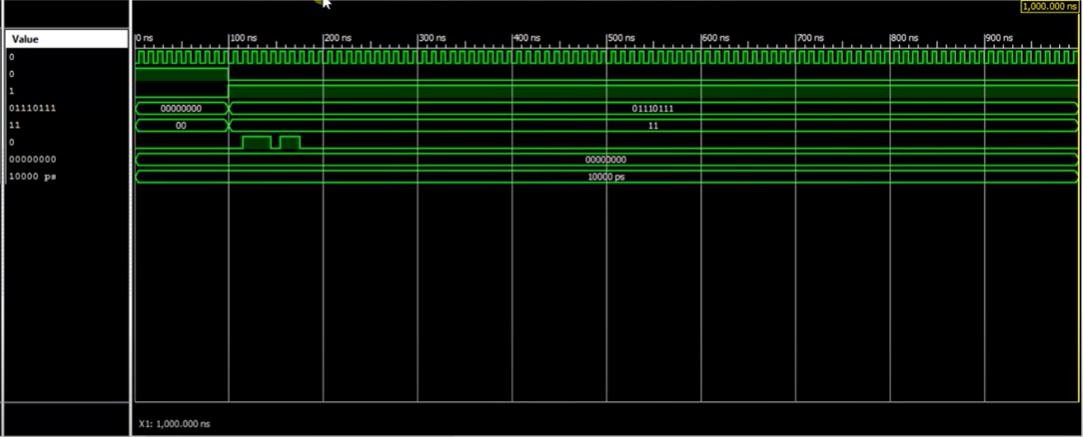
end process;

# Output:

Entity Diagram:



Waveform:



# Final Report:

Final Results: -

RTL Top Level Output File Name: unishiftreg.ngr Top Level Output File Name: unishiftreg Output Format: NGC

Optimization Goal: Speed

Keep Hierarchy: No

Design Statistics

# IOs: 13

Cell Usage

# BELS: 6

# LUT2: 2

# LUT3: 4

# FlipFlops/Latches: 9

# FD: 4

# FDE: 5

# Clock Buffers: 1

# BUFGP: 1

# IO Buffers: 8

# IBUF: 3

# OBUF: 5

Device utilization summary:

Selected Device: 3s400pq208-5

Number of Slices: 5 out of 3584 0% Number of Slice Flip Flops: 9 out of 7168 0% Number of 4 input LUTs: 6 out of 7168 0%

Number of IOs: 13

Number of bonded IOBs: 9 out of 141 6%

Number of GCLKs: 1 out of 8 12%

Partition Resource Summary:

No Partitions were found in this design.

# UCF File:

NET "Clk" LOC = "P182"; NET "Rst" LOC = "P102"; NET "Mode(0)" LOC = "P93";

NET "Mode(1)" LOC = "P90"; NET "si" LOC = "P101";

NET "so" LOC = "p181"; NET "pin(0)" LOC = "P87";

NET "pin(1)" LOC = "P86";

NET "pin(2)" LOC = "P85";

NET "pin(3)" LOC = "P81";

NET "pin(4)" LOC = "P80";

NET "pin(5)" LOC = "P78";

NET "pin(6)" LOC = "P77";

NET "pin(7)" LOC = "P74";

NET "po(0)" LOC = "P162";

NET "po(1)" LOC = "P165";

NET "po(2)" LOC = "P166";

NET "po(3)" LOC = "P167";

NET "po(4)" LOC = "P168";

NET "po(5)" LOC = "P171";

NET "po(6)" LOC = "P172";

NET "po(7)" LOC = "P175";

**Name: SHANTANU DINESH DUKARE Batch:B11 RollNo.:E43089**

**Exp. No.03:** To write VHDL code for Mod - N Counter simulate with test bench, synthesis, implement on FPGA.

# VHDL Code:

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 modncount is

Port ( clk : in STD\_LOGIC; clr : in STD\_LOGIC;

q : inout STD\_LOGIC\_VECTOR (2 downto 0)); end modncount;

architecture Behavioral of modncount is signal count: std\_logic\_vector(2 downto 0); begin

process(clk) begin

if (clr='1') then count <= "000";

elsif (rising\_edge (clk)) then if (count="100")

then count <= "000";

else count<=count+ 1; end if;

end if;

end process; q<=count;

end Behavioral;

# Test Bench:

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 hhh IS

END hhh;

ARCHITECTURE behavior OF hhh IS

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

COMPONENT modncount PORT(

clk : IN std\_logic; clr : IN std\_logic;

q : INOUT std\_logic\_vector(2 downto 0)

);

END COMPONENT;

--Inputs

signal clk : std\_logic := '0'; signal clr : std\_logic := '0';

--BiDirs

signal q : std\_logic\_vector(2 downto 0);

-- Clock period definitions

-- constant clk\_period : time := 10 ns;

BEGIN

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

clk => clk, clr => clr, q => q

);

-- Clock process definitions clk\_process :process

begin

clk <= '0'; wait for 10 ns; clk <= '1'; wait for 10 ns;

end process;

-- Stimulus process stim\_proc: process begin

clr<='1';

-- hold reset state for 100 ns. wait for 20 ns;

clr<='0';

-- hold reset state for 100 ns. wait for 20 ns;

-- hold reset state for 100 ns.

-- wait for 100 ns;

--

-- wait for clk\_period\*10;

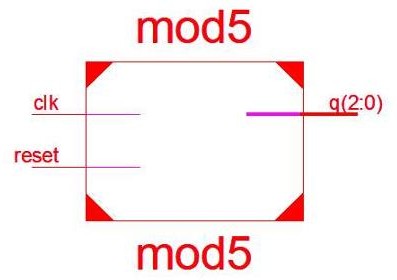
-- insert stimulus here

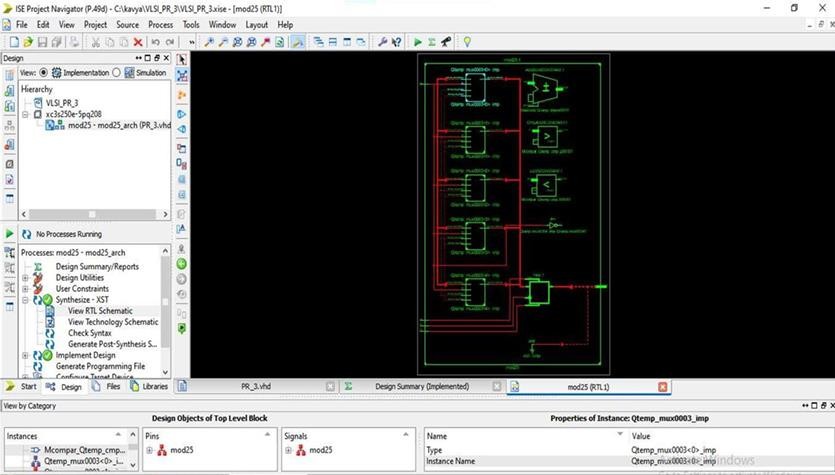
wait;

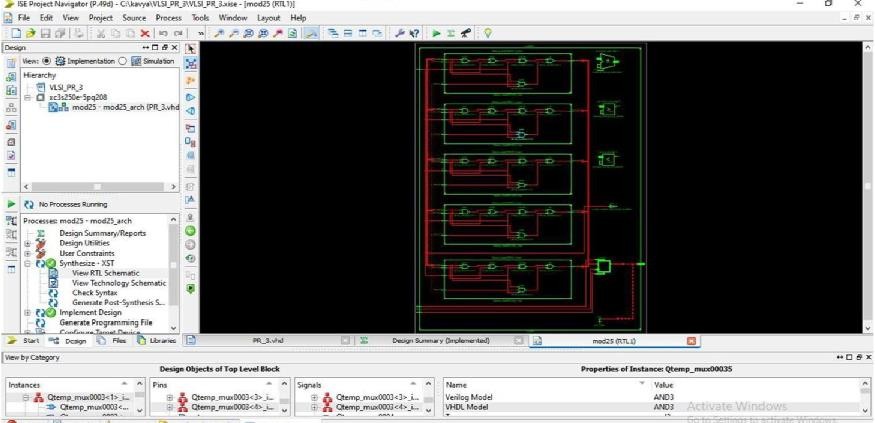
end process;

END;

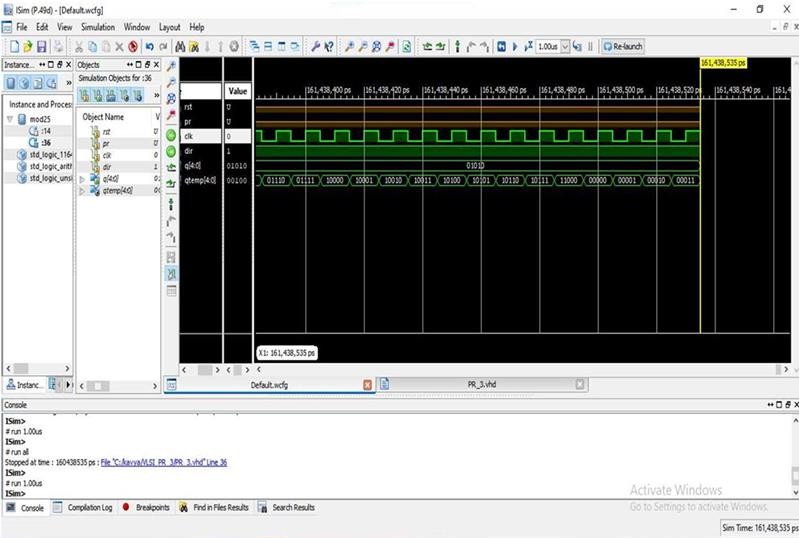
# Output:

Entity Diagram:





Waveform:



|  |  |
| --- | --- |
| **Final Report:** |  |
| Final Results: - |
| RTL Top Level Output File Name: | mod5.ngr |
| Top Level Output File Name: | mod5 |
| Output Format: | NGC |
| Optimization Goal: | Speed |
| Keep Hierarchy: | No |
| Design Statistics # Ios: | 5 |
| Cell Usage: # BELS: | 3 |
| # LUT2: | 1 |
| # LUT3:2  # FlipFlops/Latches: | 3 |
| # FDC: | 3 |
| # Clock Buffers: | 1 |
| # BUFGP: | 1 |
| # IO Buffers: | 4 |
| # IBUF: | 1 |
| # OBUF: | 3 |
| Device utilization summary: |  |
| Selected Device: | 3s400pq208-5 |
| Number of Slices: | 2 out of 3584 0% |
| Number of Slice Flip Flops: | 3 out of 7168 0% |
| Number of 4 input LUTs: | 3 out of 7168 0% |
| Number of IOs: | 5 |
| Number of bonded IOBs: | 5 out of 141 3% |
| Number of GCLKs: | 1 out of 8 12% |

# UCF File:

Net clk loc=”p182”; Net clr loc=”p102”; Net q(2) loc=”p165”; Net q(2) loc=”p166”; Net q(2) loc=”p167”;

# Name: SHANTANU DINESH DUKARE Batch:B11 Roll No.: E43089

**Exp. No.04:** To write VHDL code for FIFO memory simulate with test bench, synthesis, implement on FPGA.

# VHDL Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity FIFO is

Port ( DI : in std\_logic\_vector(3 downto 0); DO : out std\_logic\_vector(3 downto 0); RW : in std\_logic;

FULL : out std\_logic;

RS : in std\_logic;

CLK : in std\_logic);

end FIFO;

architecture Behavioral of FIFO is begin

process(RS,CLK)

type memory is Array( 0

to 3) of std\_logic\_vector(3 downto 0); variable mem: memory;

variable r\_p:integer range 0 to 3;

variable w\_p:integer range 0 to 3; variable overwrite :boolean; begin

if RS='1' then

DO<="0000";

elsif(CLK'event and CLK='1') then if RW='1'then

if (overwrite=False OR w\_p/=r\_p) then mem (w\_p):=DI;

if w\_p=3 then w\_p:=0 ;

end if;

else

end if;

w\_p:=w\_p+1; overwrite:=False;

elsif RW='0' then

if (overwrite=False OR w\_p/=r\_p) then DO<=mem(r\_p);

if (r\_p=3 ) then r\_p:=0 ; overwrite:=true;

end if;

end if;

else end if;

r\_p:=r\_p+1;

if w\_p=r\_p then if overwrite=true then FULL<='1';

end if;

else

end if; else

end if;

FULL<='0';

FULL<='0';

end process; end Behavioral;

# Test Bench:

ENTITY SSSA\_vhd IS

END SSSA\_vhd;

ARCHITECTURE behavior OF SSSA\_vhd IS COMPONENT fifo

PORT(

DI : IN std\_logic\_vector(3 downto 0); RW : IN std\_logic;

RS : IN std\_logic;

CLK : IN std\_logic;

DO : OUT std\_logic\_vector(3 downto 0); FULL : OUT std\_logic

);

END COMPONENT;

--Inputs

SIGNAL RW : std\_logic := '0'; SIGNAL RS : std\_logic := '0'; SIGNAL CLK : std\_logic := '0';

SIGNAL DI : std\_logic\_vector(3 downto 0) := (others=>'0');

BEGIN

--Outputs

SIGNAL DO : std\_logic\_vector(3 downto 0); SIGNAL FULL : std\_logic;

constant CLK\_period : time := 10 ns;

uut: fifo PORT MAP(

DI => DI, DO => DO, RW => RW,

FULL => FULL, RS => RS,

CLK => CLK

);

process begin

CLK<='0';

wait for CLK\_period/2; CLK<='1';

wait for CLK\_period/2; end process;

tb : PROCESS BEGIN

RS<='1';

wait for 100 ns;

RS<='0';

RW<='1';

DI<="0011";

-- Wait 100 ns for global reset to finish wait for 10 ns;

RW<='1';

DI<="0110";

-- Wait 100 ns for global reset to finish wait for 10 ns;

RW<='1';

DI<="1100";

-- Wait 100 ns for global reset to finish wait for 10 ns;

RW<='1';

DI<="1001";

-- Wait 100 ns for global reset to finish wait for 10 ns;

-- Place stimulus here RW<='0';

-- Wait 100 ns for global reset to finish wait for 10 ns;

RW<='0';

-- Wait 100 ns for global reset to finish wait for 10 ns;

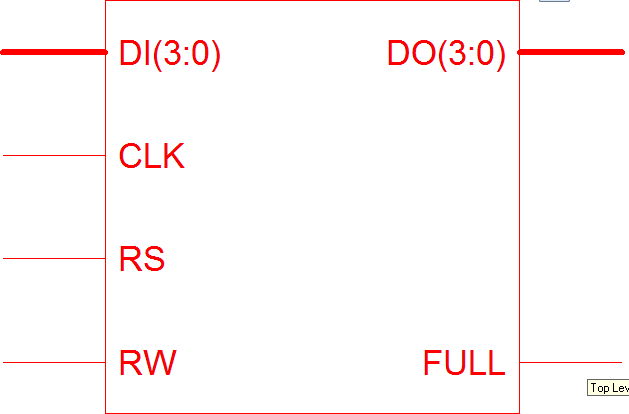
RW<='0';

-- Wait 100 ns for global reset to finish wait for 10 ns;

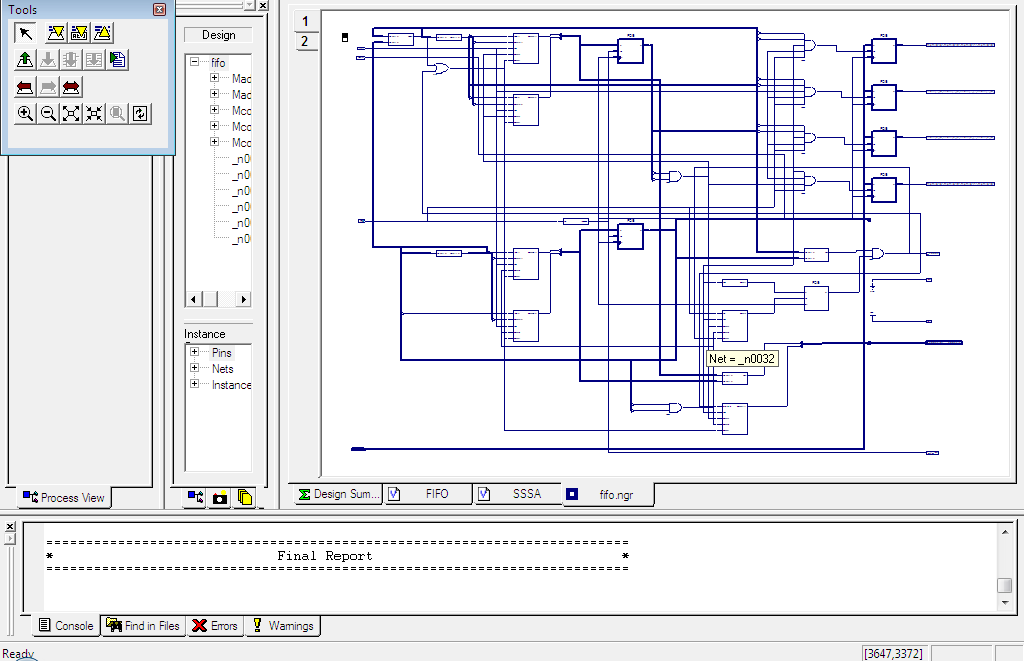
WAIT FOR CLK\_period\*10; wait; **--** will wait forever

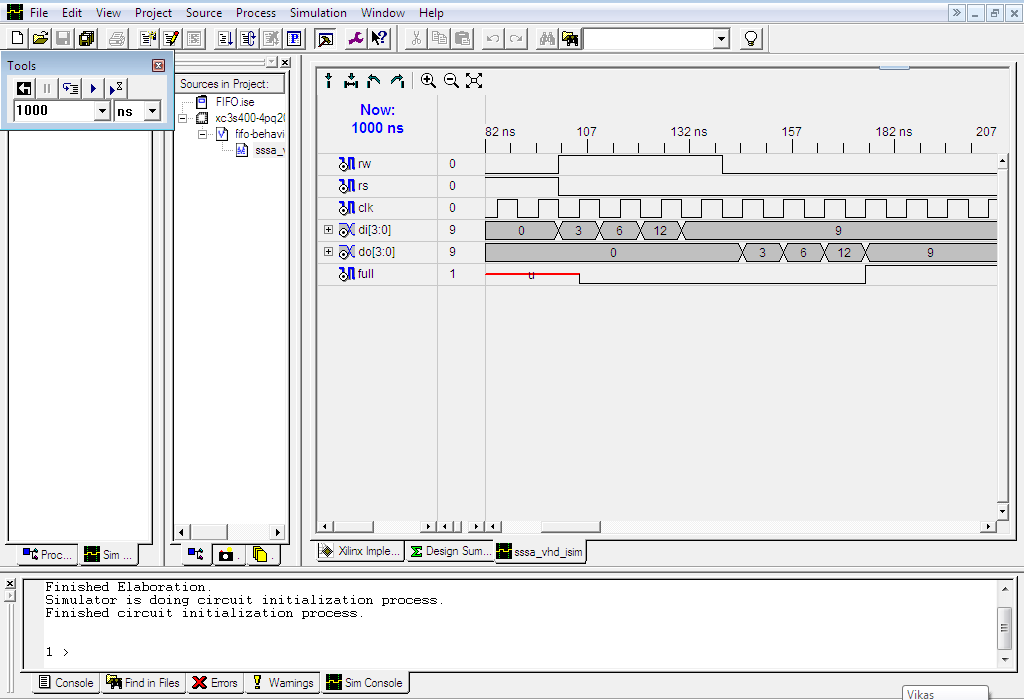
END PROCESS;

# Output:

Entity Diagram:

Waveforms:





|  |  |
| --- | --- |
| **Final Report:** |  |
| Final Results  RTL Top Level Output File Name: | fifo.ngr |
| Keep Hierarchy: | NO |
| Design Statistics # IOs: | 12 |
| Cell Usage : # BELS: | 2 |
| # GND: | 1 |
| # VCC: | 1 |
| # FlipFlops/Latches: | 26 |
| # FDCE: | 4 |
| # FDE: | 22 |

CPU: 5.92 / 8.26 s | Elapsed: 6.00 / 8.00 s

-->

Total memory usage is 122500 kilobytes

Number of errors: 0 ( 0 filtered)

Number of warnings: 0 ( 0 filtered)

Number of infos: 0 ( 0 filtered

# UCF File:

NET "clk" LOC = "P3"; -- Change these LOC values according to your FPGA pin configuration

NET "rst" LOC = "P4"; NET "wr\_en" LOC = "P5"; NET "rd\_en" LOC = "P6";

NET "data\_in[0]" LOC = "P7"; NET "data\_in[1]" LOC = "P8"; NET "data\_in[2]" LOC = "P9"; NET "data\_in[3]" LOC = "P10"; NET "data\_in[4]" LOC = "P11"; NET "data\_in[5]" LOC = "P12"; NET "data\_in[6]" LOC = "P13"; NET "data\_in[7]" LOC = "P14";

NET "data\_out[0]" LOC = "P15"; NET "data\_out[1]" LOC = "P16"; NET "data\_out[2]" LOC = "P17"; NET "data\_out[3]" LOC = "P18"; NET "data\_out[4]" LOC = "P19"; NET "data\_out[5]" LOC = "P20"; NET "data\_out[6]" LOC = "P21"; NET "data\_out[7]" LOC = "P22";

NET "full" LOC = "P23"; NET "empty" LOC = "P24";

**Name: SHANTANU DINESH DUKARE Batch:B11 RollNo.:E43089**

**Exp. No.05:** To write VHDL code for LCD Interface simulate with test bench, synthesis, implement on FPGA.

# VHDL Code:

use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity lcd is

Port ( clk,reset : in std\_logic; RS,EN,RW : out std\_logic;

data : out std\_logic\_vector(7 downto 0)); end lcd;

architecture Behavioral of lcd is

type state\_type is (s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11, s12,s13,s14,s15,s16,s17,s18,s19,s20,s21,s22,s23);

signal state:state\_type;

SIGNAL count:std\_logic\_vector(22 downto 0); SIGNAL clk1:std\_logic;

begin process(Clk,Reset)

begin

end process; RW<='0';

if(Clk' event AND Clk='1')then count<=count+"0001";

end if; clk1<=count(20);

process(clk1, reset) begin

if reset = '1' then

state <= s0;

elsif rising\_edge(clk1) then if state = s0 then

state <= s1;

RS<='0'; -- Write commonds to LCD. EN <= '1';

data <= "00110000"; -- Function set for 8 bit interface, 1 line mode and 5x7 dot matrix. end if;

if state = s1 then

state <= s2;

EN <= '0';

end if;

if state = s2 then

state <= s3;

EN <= '1';

data <= "00001111"; -- Display cursor and blinking ON.

end if;

if state = s3 then

state <= s4;

EN <= '0';

end if;

if state = s4 then

state <= s5;

EN <= '1';

data <= "00000001"; -- Clear display.

end if;

if state = s5 then

state <= s6;

EN <= '0';

end if;

if state = s6 then

state <= s7;

EN <= '1';

data <= "10000100"; -- Display address.

end if;

if state = s7 then

state <= s8;

EN <= '0';

end if;

if state = s8 then

RS <= '1'; -- Write data to LCD. state <= s9;

EN <= '1';

data <= "00101010"; --(\*)

end if;

if state = s9 then

state <= s10;

EN <= '0';

end if;

if state = s10 then

state <= s11; EN <= '1';

data <= "01010011"; --S

end if;

if state = s11 then

state <= s12;

EN <= '0';

end if;

if state = s12 then

state <= s13;

EN <= '1';

data <= "01001011"; --K

end if;

if state = s13 then

state <= s14;

EN <= '0';

end if;

if state = s14 then

state <= s15;

EN <= '1';

data <= "01001110"; --N

end if;

if state = s15 then

state <= s16;

EN <= '0';

end if;

if state = s16 then

state <= s17;

EN <= '1';

data <= "01000011"; --C

end if;

if state = s17 then

state <= s18;

EN <= '0';

end if;

if state = s18 then

state <= s19;

EN <= '1';

data <= "01001111"; --O

end if;

if state = s19 then

state <= s20;

EN <= '0';

end if;

if state = s20 then

state <= s21;

EN <= '1';

data <= "01000101"; --E

end if;

if state = s21 then

state <= s22;

EN <= '0';

end if;

if state = s22 then

state <= s23;

EN <= '1';

data <= "00101010"; --(\*)

end if;

if state = s23 then

EN <= '0';

end if;

end if; end process;

end Behavioral;

# Test Bench:

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 lcdtest IS END lcdtest;

ARCHITECTURE behavior OF lcdtest IS

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

PORT(

clk1 : IN std\_logic; reset : IN std\_logic; RS : OUT std\_logic; EN : OUT std\_logic; RW : OUT std\_logic;

data : OUT std\_logic\_vector(7 downto 0)

);

END COMPONENT;

--Inputs

signal clk1 : std\_logic := '0'; signal reset : std\_logic := '0';

--Outputs signal RS : std\_logic; signal EN : std\_logic;

signal RW : std\_logic;

signal data : std\_logic\_vector(7 downto 0);

-- Clock period definitions

constant clk1\_period : time := 10 ns; BEGIN

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

clk1 => clk1,

reset => reset, RS => RS, EN => EN, RW => RW,

data => data

);

-- Clock process definitions clk1\_process :process begin

end process;

clk1 <= '0';

wait for clk1\_period/2; clk1 <= '1';

wait for clk1\_period/2;

-- Stimulus process stim\_proc: process begin

-- hold reset state for 100 ns.

reset<='0';

wait for clk1\_period\*10;

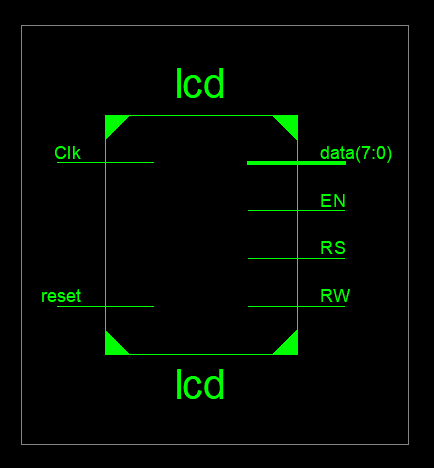
-- insert stimulus here wait;

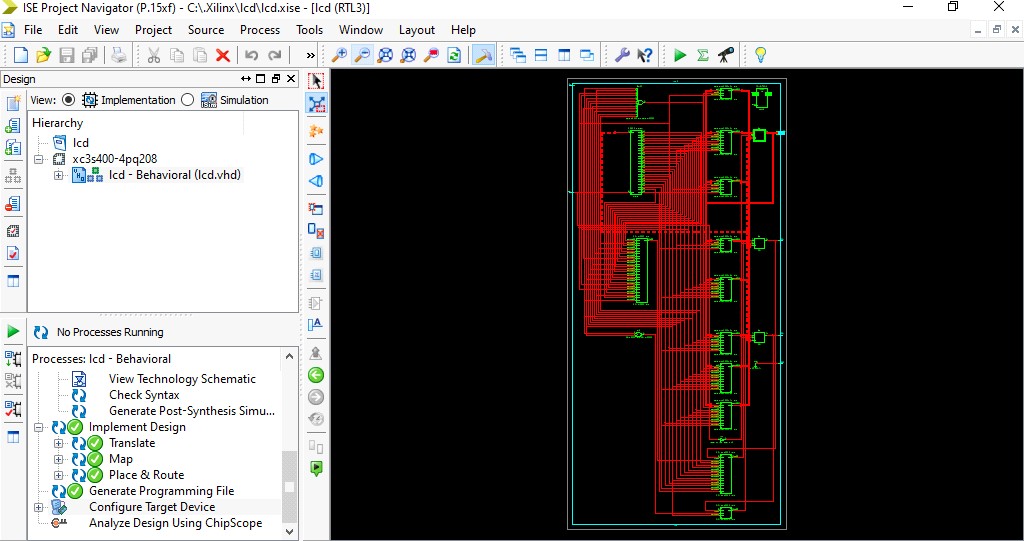
end process;

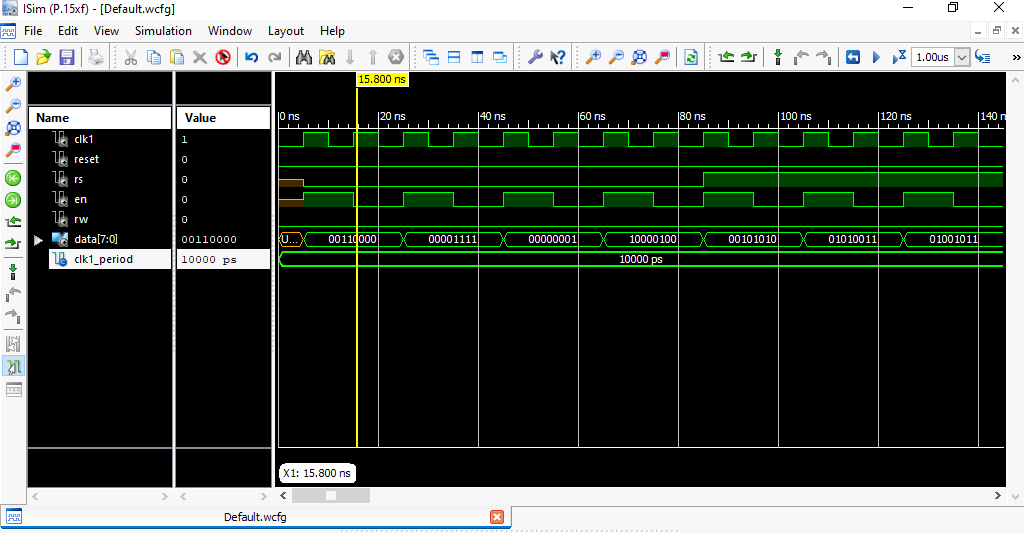
END;

# Output:

Entity Diagram:



Waveform:



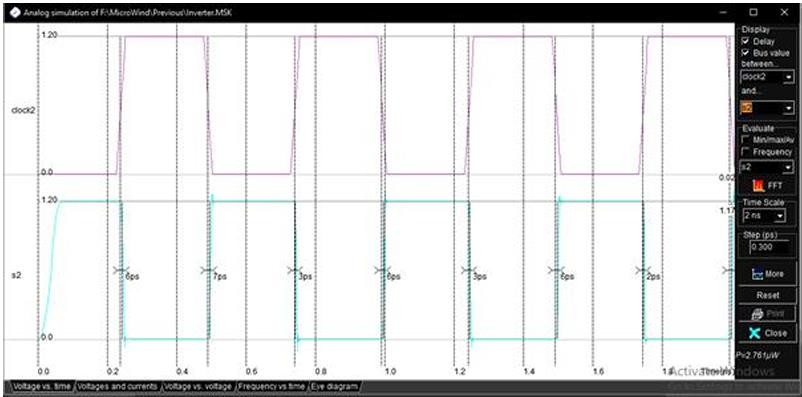
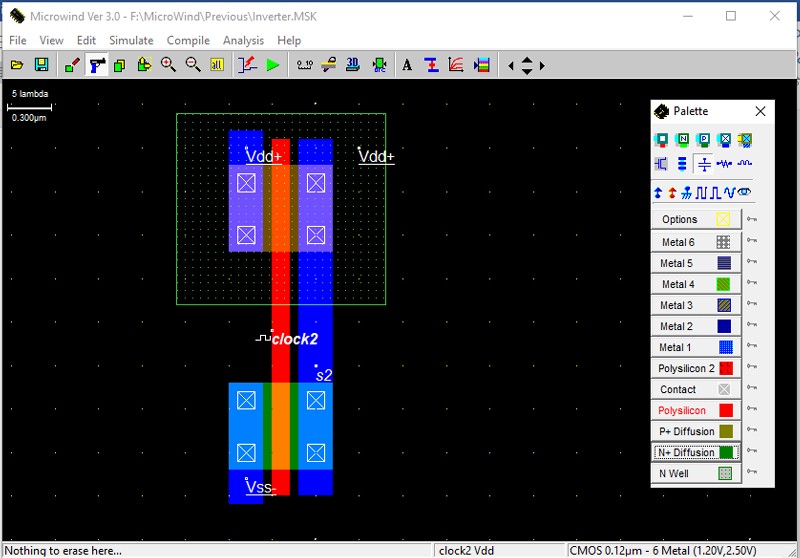
|  |  |
| --- | --- |
| **Final Report:** |  |
| RTL Top Level Output File Name: | lcd.ngr |
| Top Level Output File Name: | lcd |
| Output Format: | NGC |
| Optimization Goal: | Speed |
| Keep Hierarchy: | No |
| Design Statistics  # IOs: | 13 |
| Cell Usage:  # BELS: | 92 |
| # GND: | 1 |
| # INV: | 2 |
| # LUT1: | 20 |
| # LUT2: | 1 |
| # LUT2\_L: | 3 |
| # LUT3: | 3 |
| # LUT3\_D: | 1 |
| # LUT4: | 14 |
| # LUT4\_D: | 1 |
| # LUT4\_L: | 4 |
| # MUXCY: | 20 |
| # VCC: | 1 |
| # XORCY: | 21 |
| # FlipFlops/Latches: | 55 |
| # FD: | 21 |
| # FDC: | 22 |
| # FDCE: | 1 |
| # FDE: | 10 |
| # FDP: | 1 |
| # Clock Buffers: | 2 |
| # BUFG: | 1 |
| # BUFGP: | 1 |
| # IO Buffers: | 12 |
| # IBUF: | 1 |
| # OBUF: | 11 |
| **UCF File:** |  |
| NET data(0) LOC =P62;  NET data(1) LOC =P63; NET data(2) LOC =P64; NET data(3) LOC =P65; NET data(4) LOC =P67; NET data(5) LOC =P68; NET data(6) LOC =P71; NET data(7) LOC =P72; NET Clk LOC =P183; NET reset LOC =P102; NET RS LOC =P57;  NET EN LOC =P61; NET RW LOC =P58; |  |

# Name: SHANTANU DINESH DUKARE Batch:B11 Roll No.: E43089

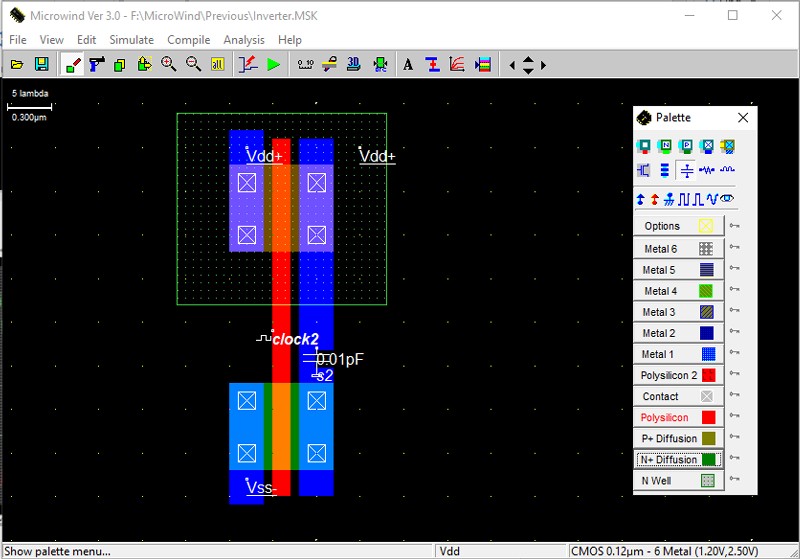
**Exp. No.06:** To prepare CMOS layout of Inverter, NAND, NOR gate in Microwind, simulate with & without capacitive load, comment on rise & fall times.

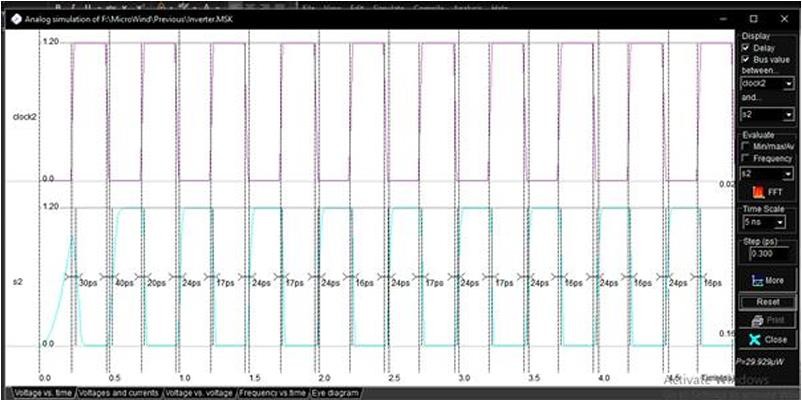
# CMOS Layout and Waveforms:

Inverter:

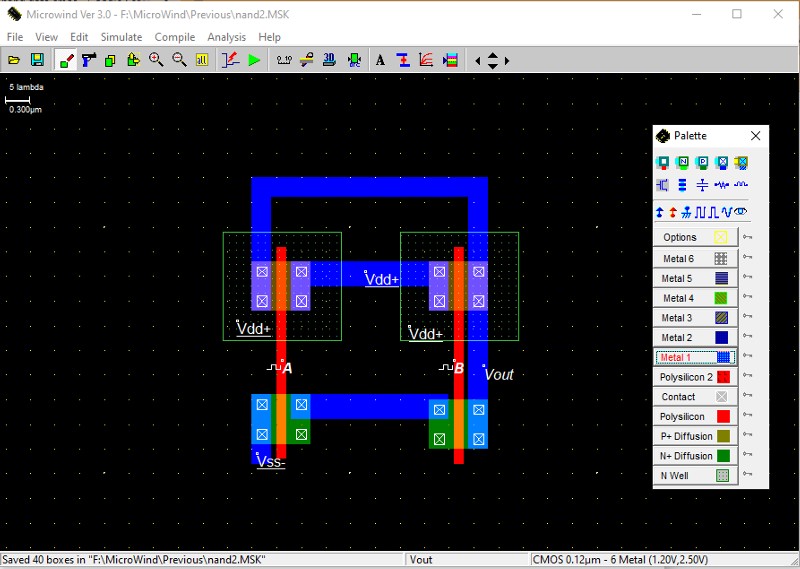


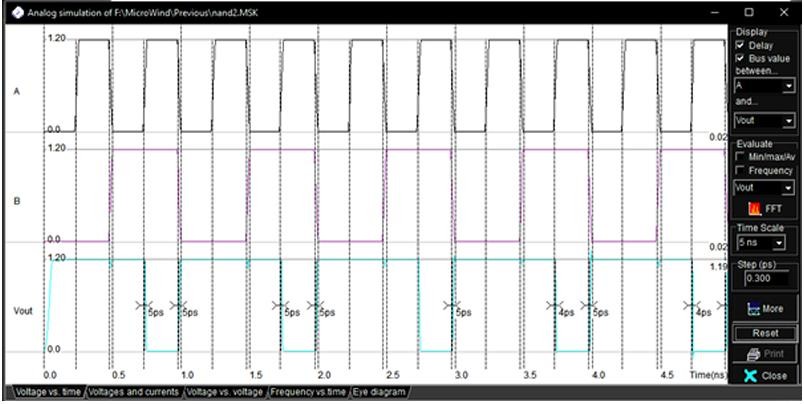
With Capacitor:



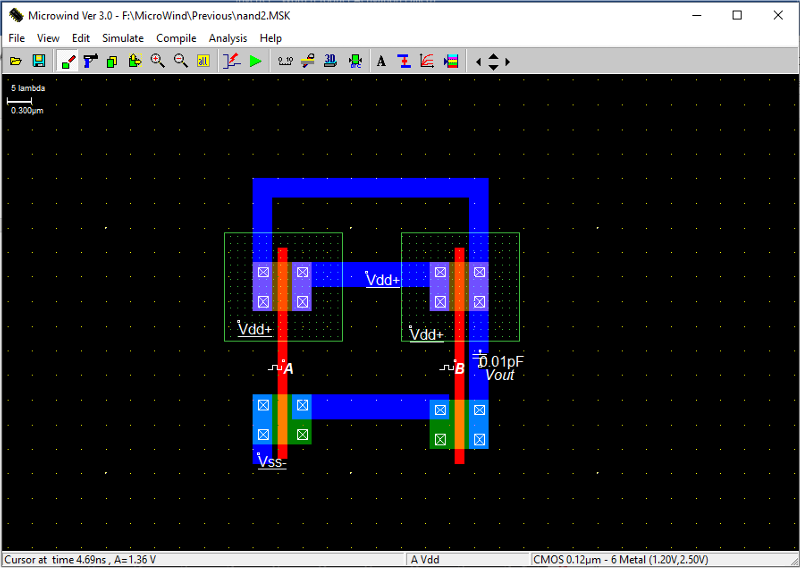


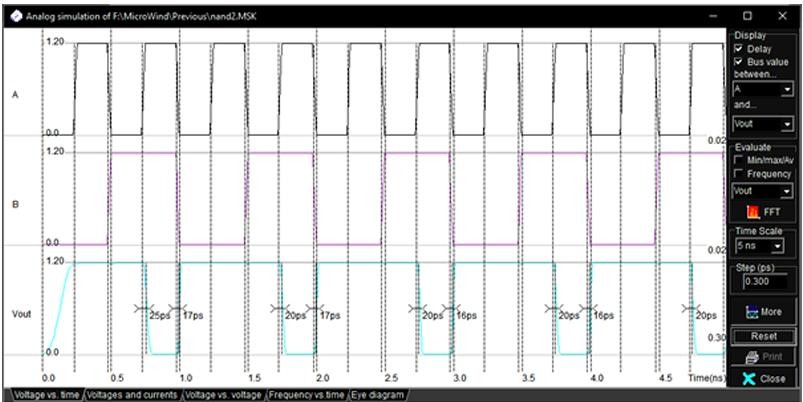
NAND:



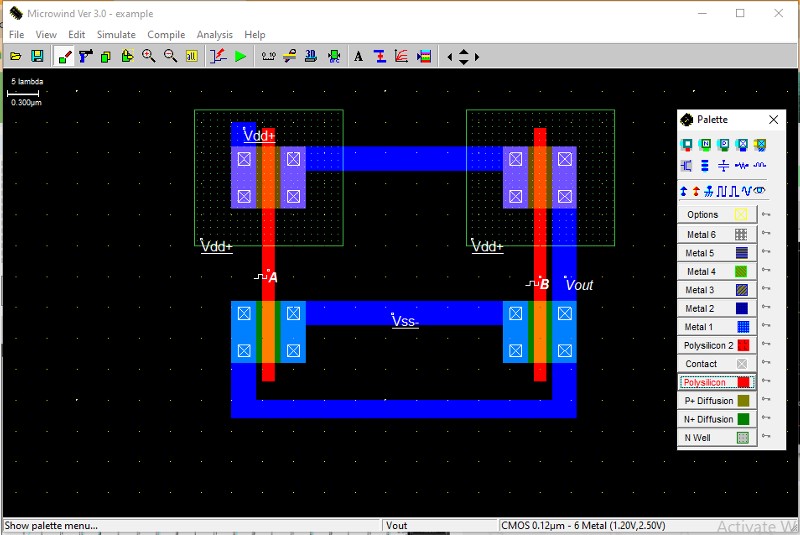


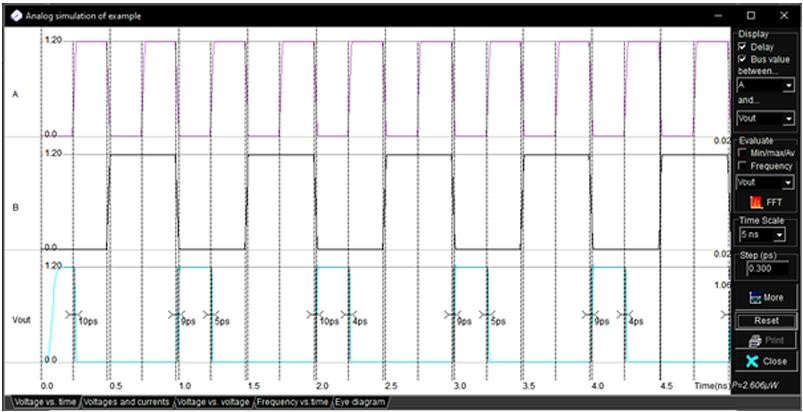
With Capacitor:



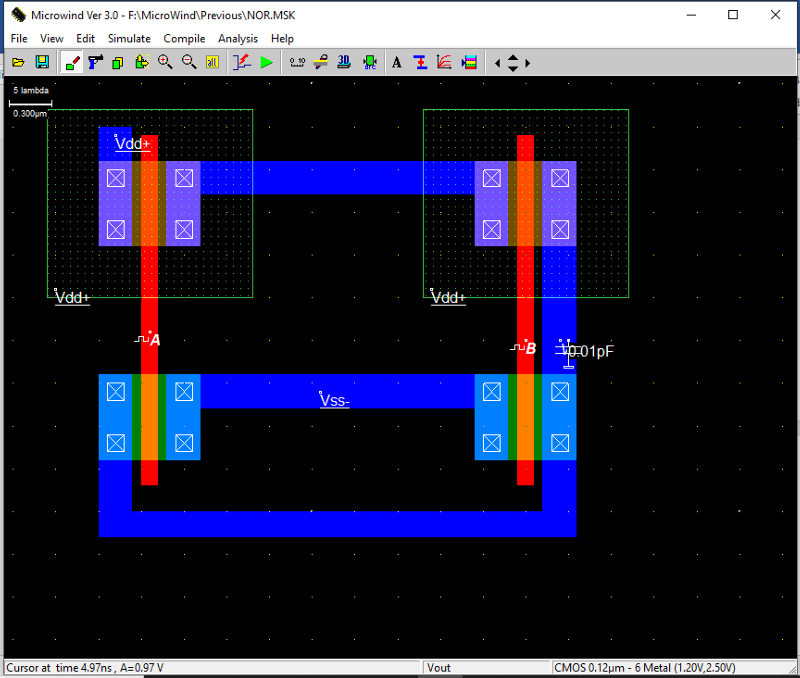


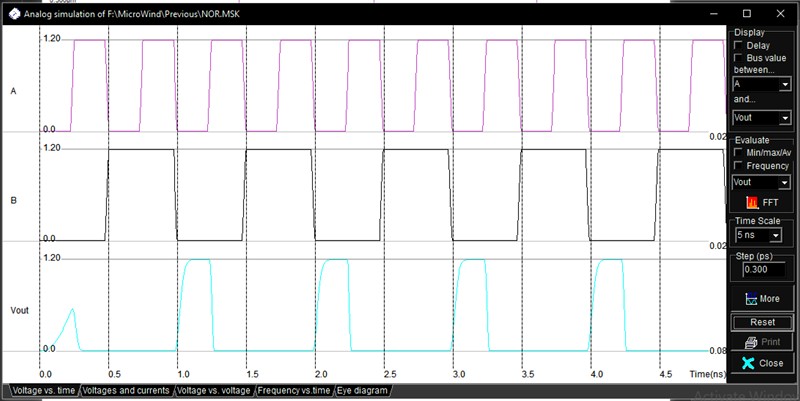
NOR:





With Capacitor:



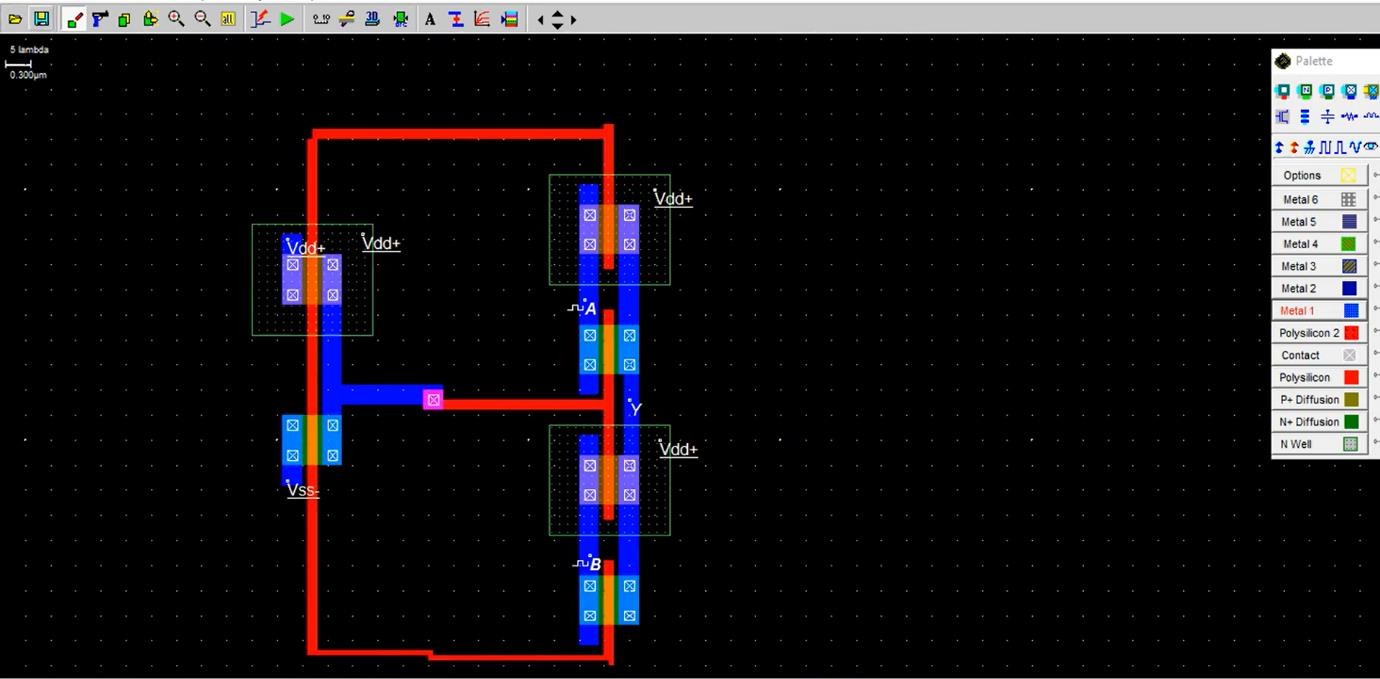


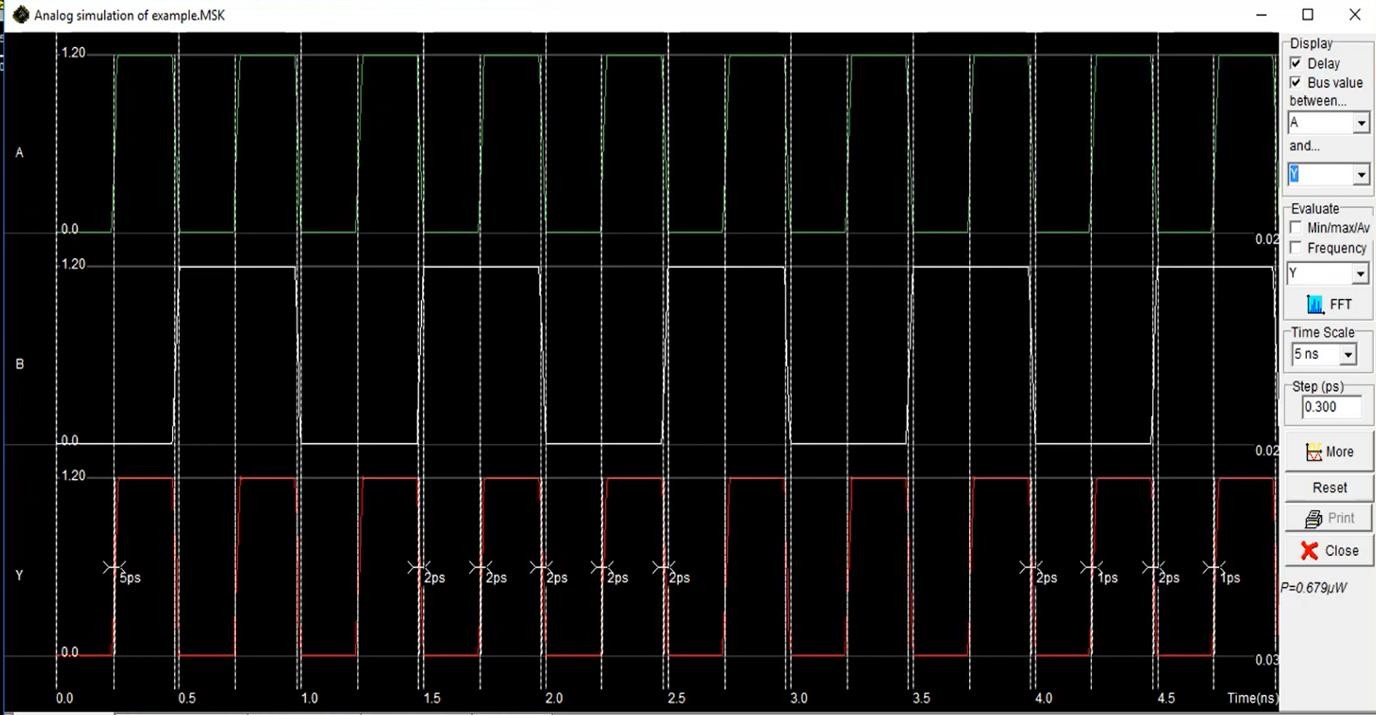
**Name: SHANTANU DINESH DUKARE Batch:B11 RollNo.:E43089**

**Exp. No.07:** To prepare CMOS layout of 2:1 Mux using logic gates & transmission gate in Microwind, simulate with & without capacitive load, comment on rise & fall times.

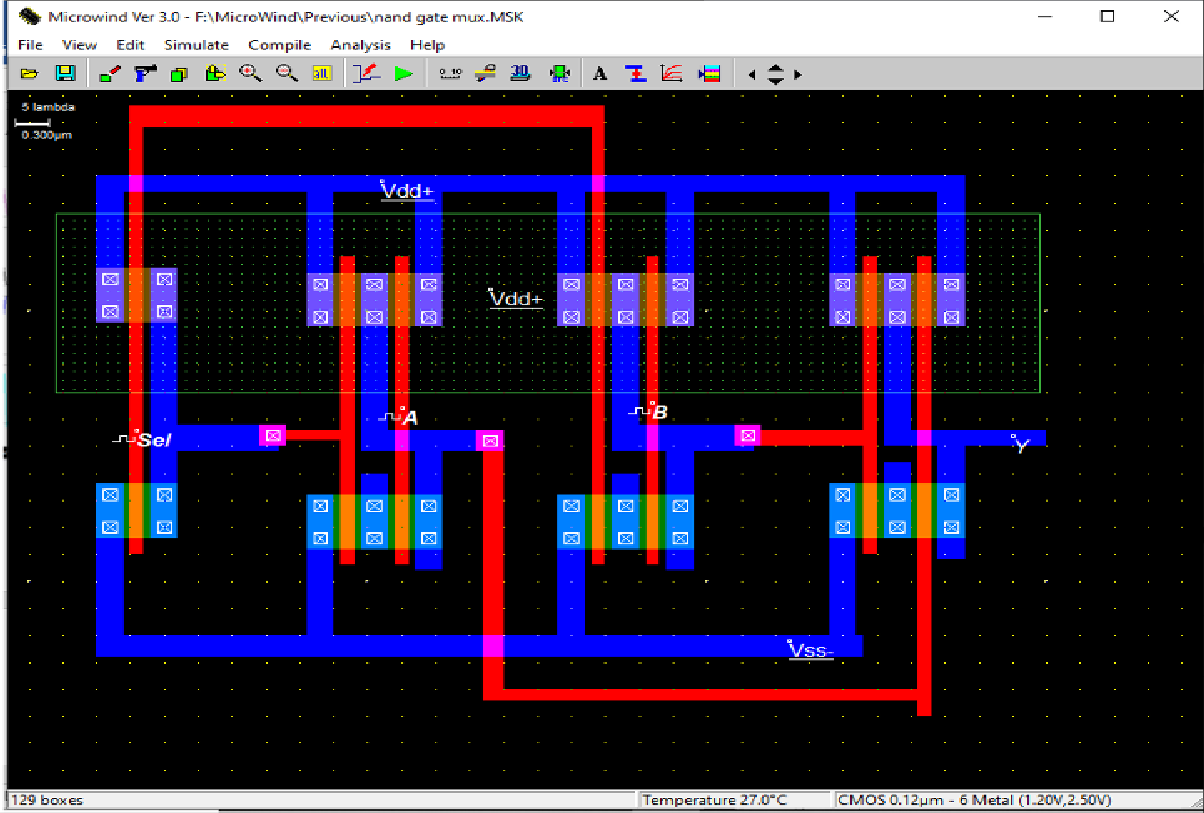
# CMOS Layout and Waveform of 2:1 MUX:

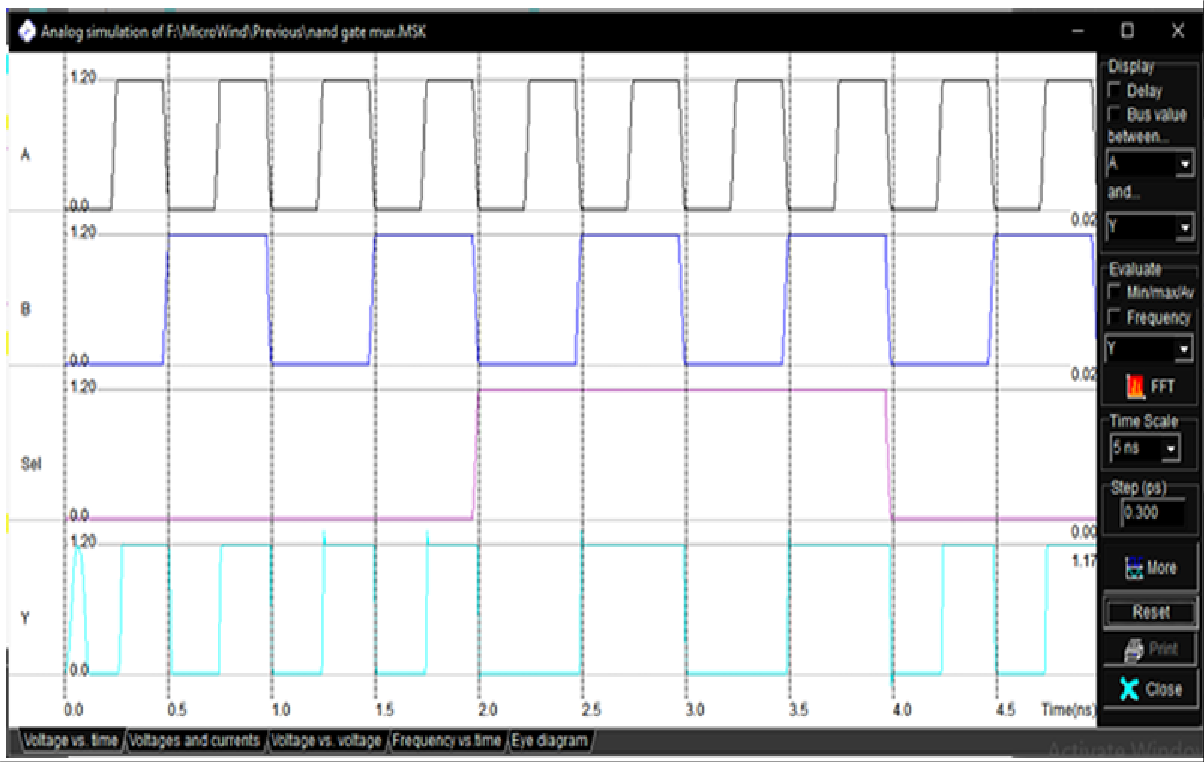
Using Logic (NAND) Gates:





Using Transmission Gate:



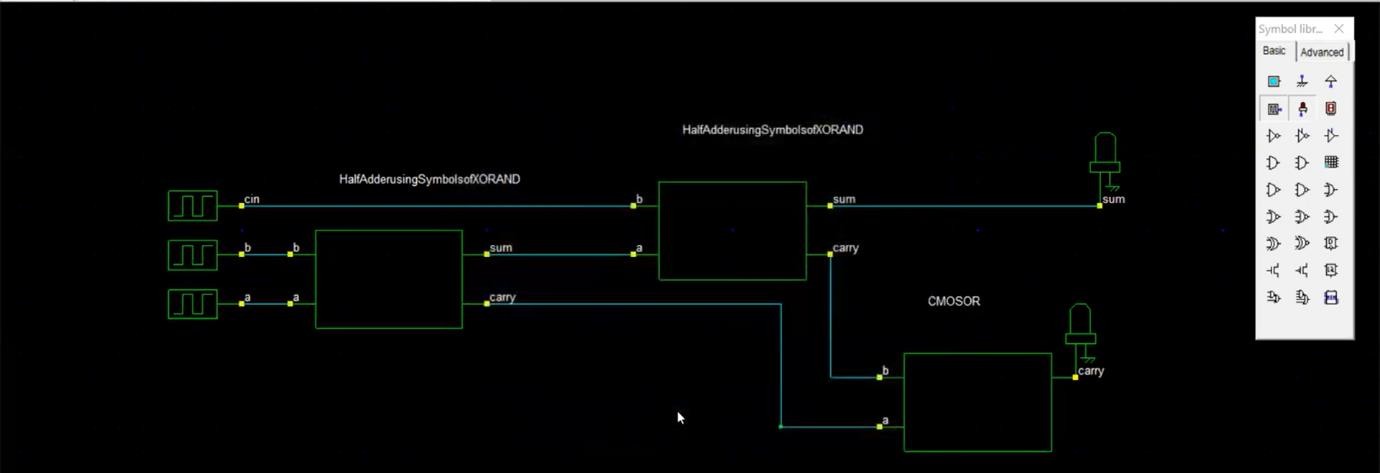


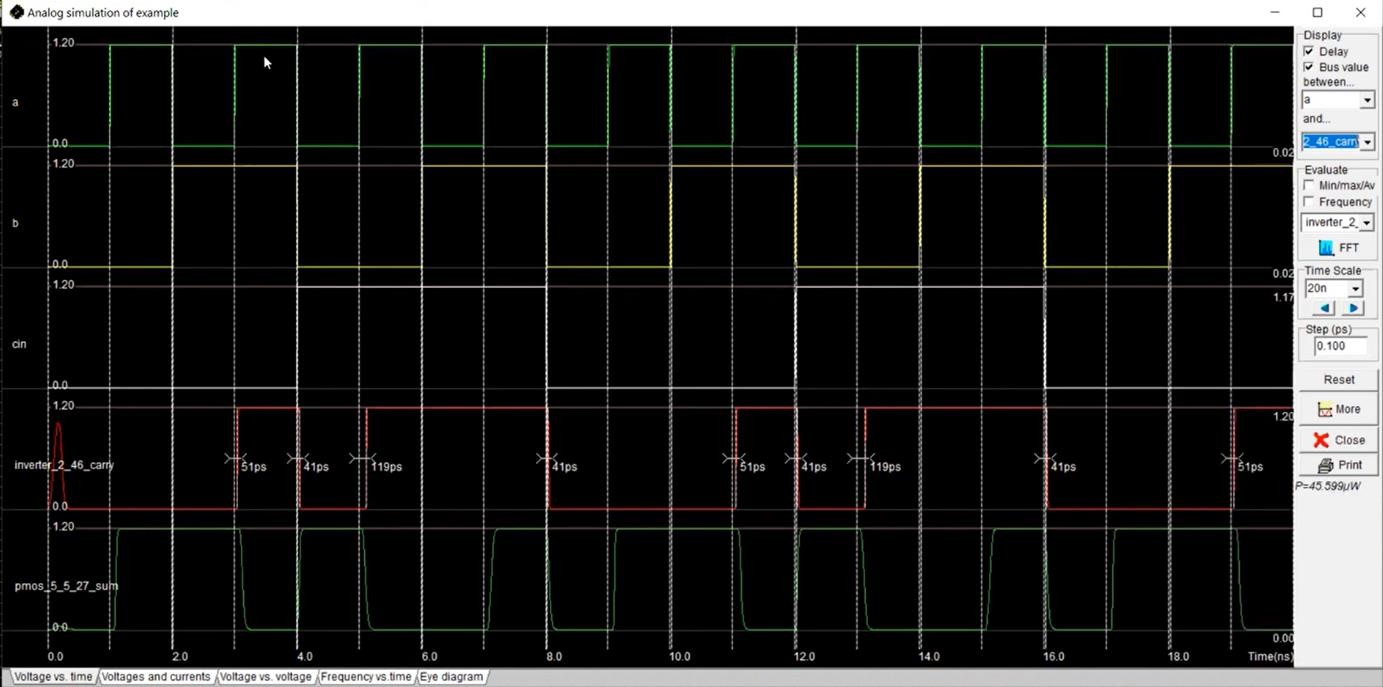
**Name: SHANTANU DINESH DUKARE Batch:B11 RollNo.:E43089**

**Exp. No.08:** To prepare CMOS layout of Half Adder & Full Adder in Microwind simulate with & without capacitive load, comment on rise & fall times.

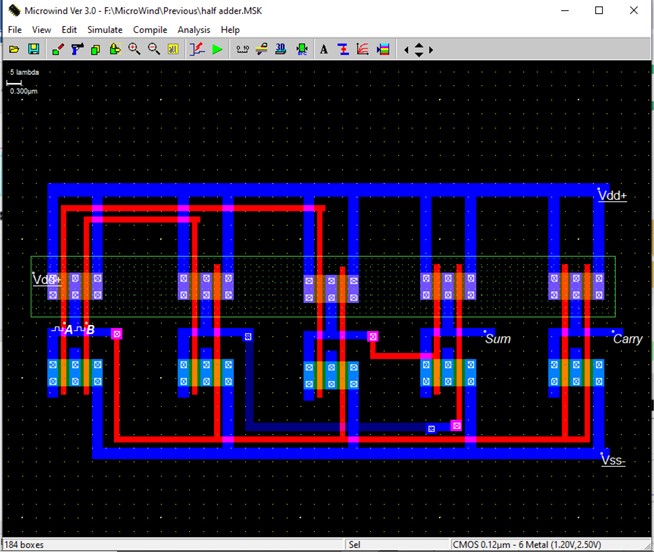
# CMOS Layout and Waveform:

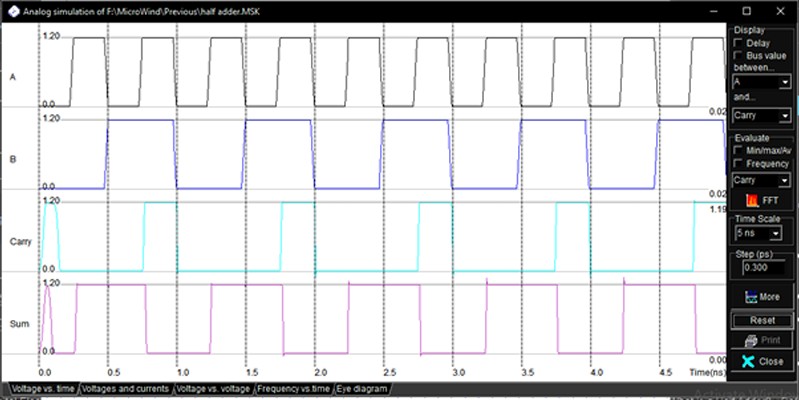
Full Adder:





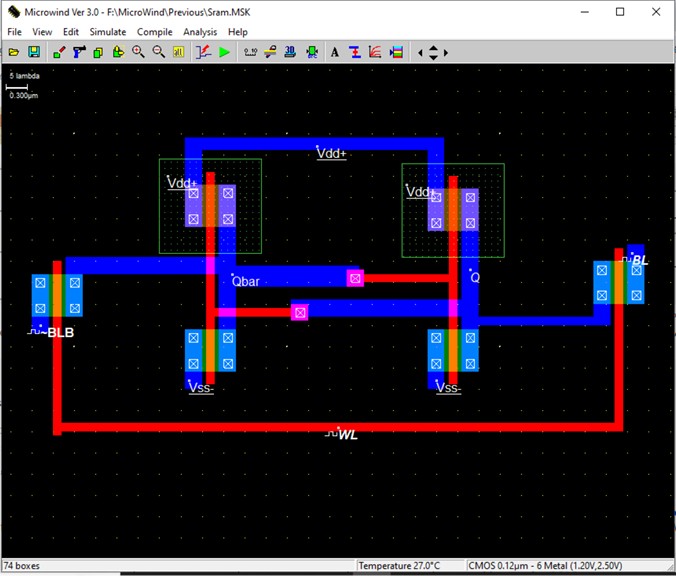
Half Adder:



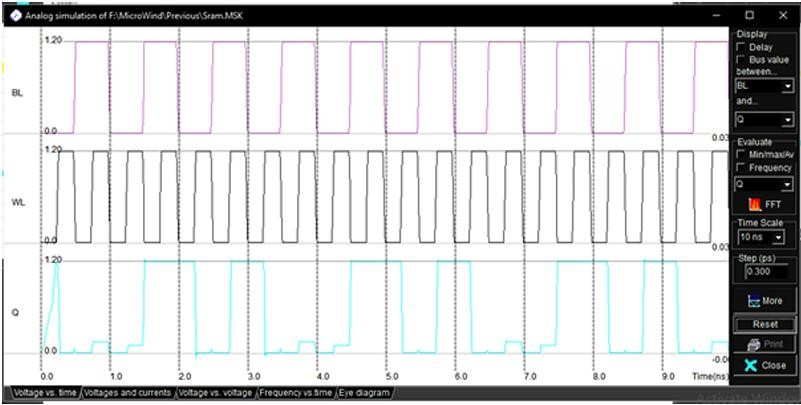


**Name: SHANTANU DINESH DUKARE Batch:B11 Roll No.: E43089Exp. No.09:** To prepare CMOS layout of one bit SRAM Cell in Microwind simulate with & without capacitive load, comment on rise & fall times.

# CMOS Layout:



**Waveform:**



**Name: SHANTANU DINESH DUKARE Batch:B11 RollNo.:E43089**

**Exp. No.10:** To Write VHDL code for Full adder by using half adder simulate with test bench synthesis, implement on FPGA.

# VLDL Code:

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;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.--library UNISIM;

--use UNISIM.VComponents.all;

entity HA is

Port (

);

end HA;

a : in STD\_LOGIC; b : in STD\_LOGIC;

sum : out STD\_LOGIC; carry : out STD\_LOGIC

architecture Behavioral of HA is begin

sum<= a xor b; carry<= a and b;

end Behavioral;

Full adder 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;

--Uncomment the following library declaration if instantiating

--any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity fa is

Port (

a : in STD\_LOGIC;

end fa;

b : in STD\_LOGIC;

cin : in STD\_LOGIC; s : out STD\_LOGIC;

cot : out STD\_LOGIC);

architecture Behavioral of fa is component HA

Port (

a : in STD\_LOGIC; b : in STD\_LOGIC;

sum : out STD\_LOGIC; carry : out STD\_LOGIC

);

begin

end component;

signal s1, c1, c2: std\_logic;

HA1: HA port map( a,b, s1, c1); HA2: HA port map(s1, cin, s, c2); cot<= c1 or c2;

end Behavioral;

# Test Bench:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Full\_Adder\_tb is end Full\_Adder\_tb;

architecture Behavioral of Full\_Adder\_tb is signal A, B, Cin, Sum, Cout: STD\_LOGIC;

component Full\_Adder

Port ( A : in STD\_LOGIC; B : in STD\_LOGIC;

Cin : in STD\_LOGIC; Sum : out STD\_LOGIC; Cout : out STD\_LOGIC);

end component;

begin

uut: Full\_Adder Port map ( A => A, B => B, Cin => Cin, Sum => Sum, Cout => Cout );

stim\_proc: process begin

-- Test cases

A <= '0'; B <= '0'; Cin <= '0'; wait for 10 ns;

A <= '0'; B <= '0'; Cin <= '1'; wait for 10 ns;

A <= '0'; B <= '1'; Cin <= '0'; wait for 10 ns;

A <= '0'; B <= '1'; Cin <= '1'; wait for 10 ns;

A <= '1'; B <= '0'; Cin <= '0'; wait for 10 ns;

A <= '1'; B <= '0'; Cin <= '1'; wait for 10 ns;

A <= '1'; B <= '1'; Cin <= '0'; wait for 10 ns;

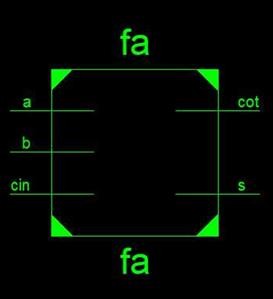
A <= '1'; B <= '1'; Cin <= '1'; wait for 10 ns;

wait;

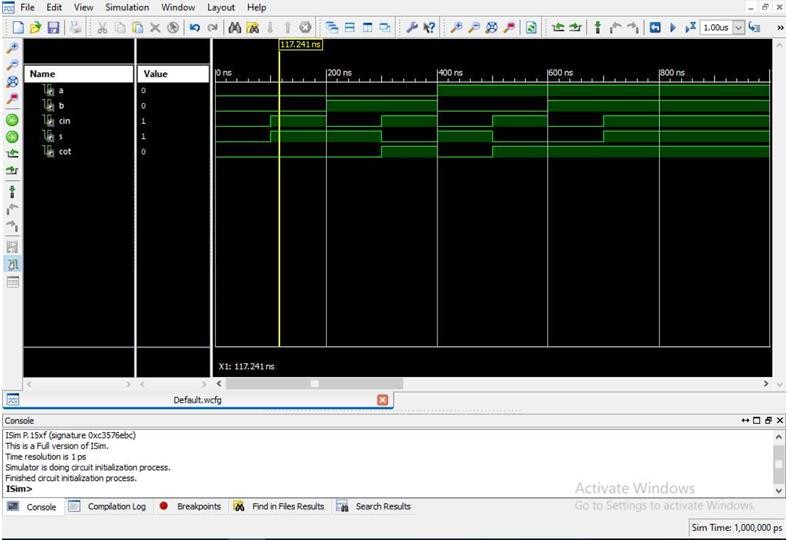
end process; end Behavioral;

# Output:

Entity Diagram:



Waveform:



# Final Report:

Final Results: -

RTL Top Level Output File Name: fa.ngr Top Level Output File: fa

Name Output Format: NGC

Optimization Goal: Speed

Keep Hierarchy: No

Design Statistics

# IOs: 5

Cell Usage

# BELS: 2

# LUT3: 2

# IO Buffers: 5

# IBUF: 3

# OBUF: 2

# UCF File:

net a loc = p87; net b loc= p86; net cin loc= p85; net s loc = p162;

net cot loc = p165;

Design summary

Release 14.1 - xst P.15xf (nt64)

Copyright (c) 1995-2012 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to xst/projnav.tmp Total REAL time to Xst completion: 0.00 secs Total CPU time to Xst completion: 0.09 secs 5

--> Parameter xsthdpdir set to xst

Total REAL time to Xst completion: 0.00 secs Total CPU time to Xst completion: 0.09 secs