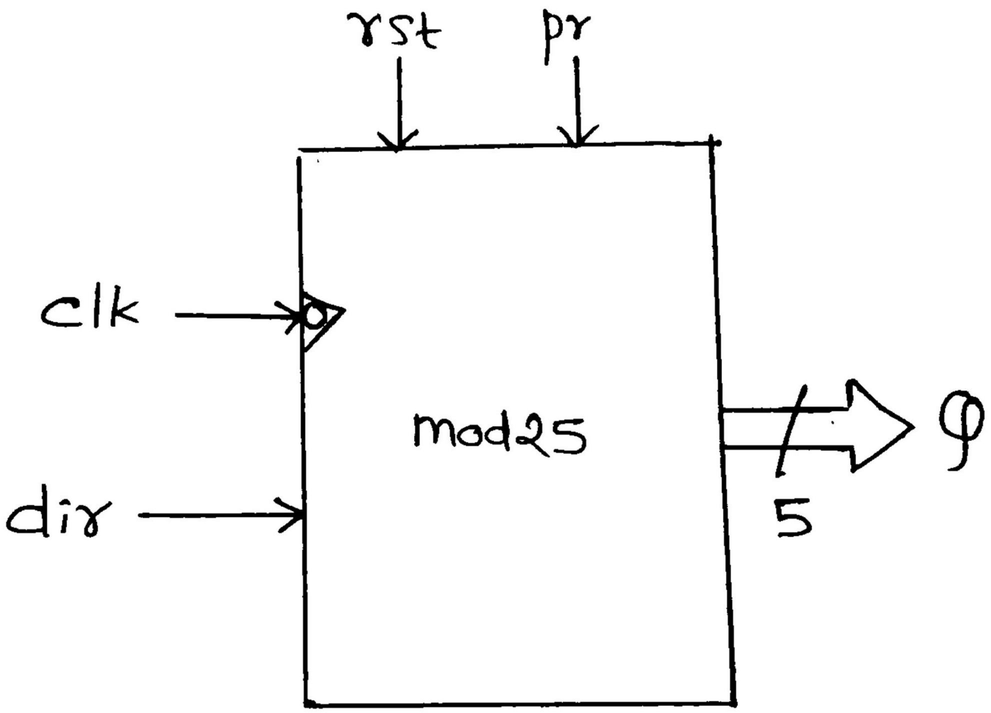
Expt . No:5 Date :	Mod-N Counter

Aim: To model a MOD-25 UP-DOWN Counter(25 States,up-down counter with Async-Reset, Preset)

BLOCK DIAGRAM



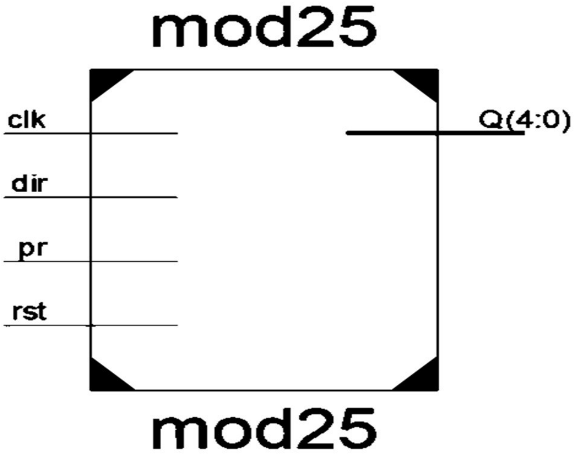
FUNCTION TABLE

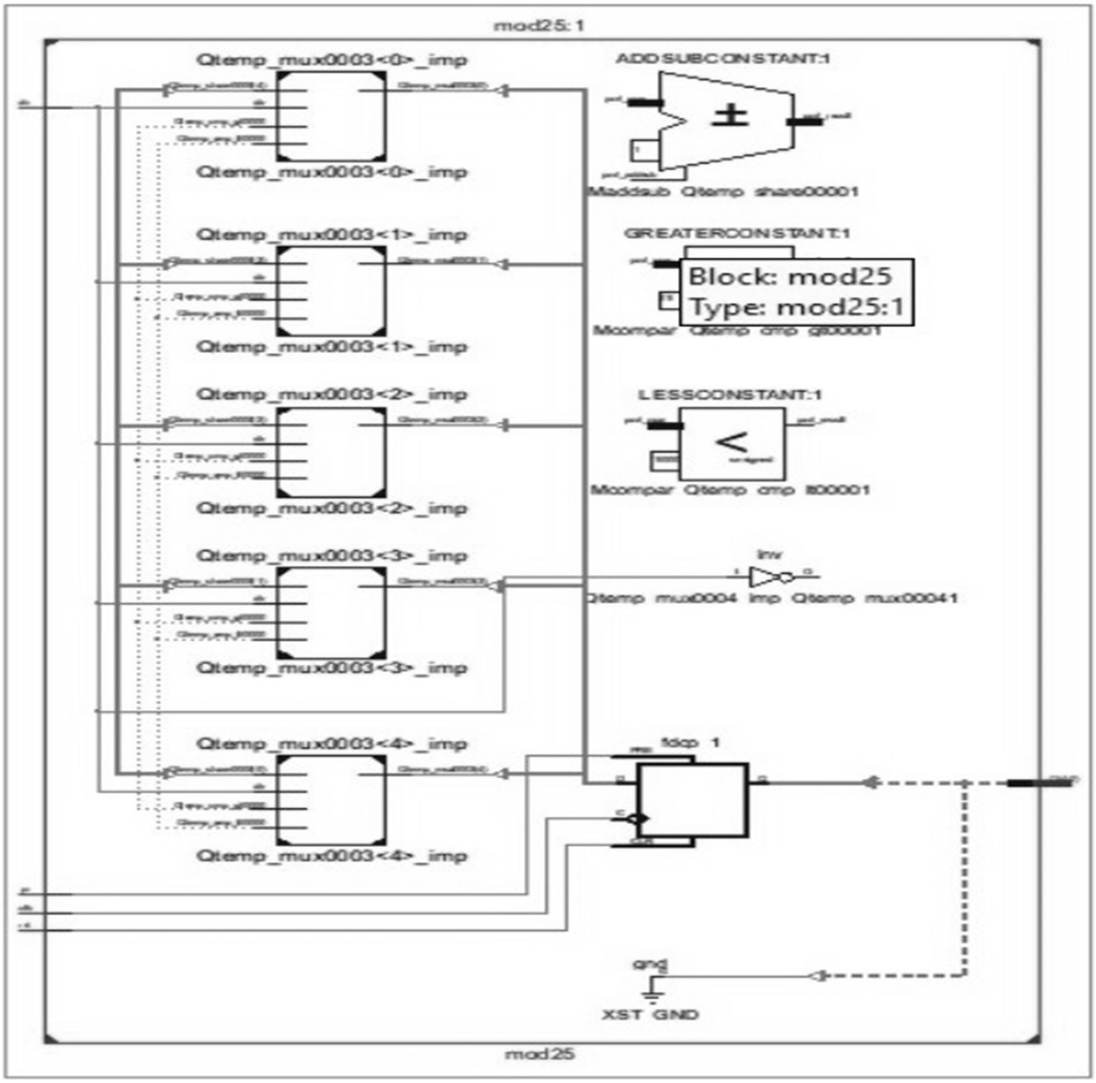
rst	**pr**	**clk**	**dir**	**Q_{n+1} (Next State)**	**Operation**
1	0	x	x	00000	Reset
0	1	x	x	11111	Set
0	0	↓	1	Q_n + 1	$(0)_{10}$ □ $(24)_{10}$
0	0	↓	0	Q_n - 1	$(24)_{10}$ □ $(0)_{10}$

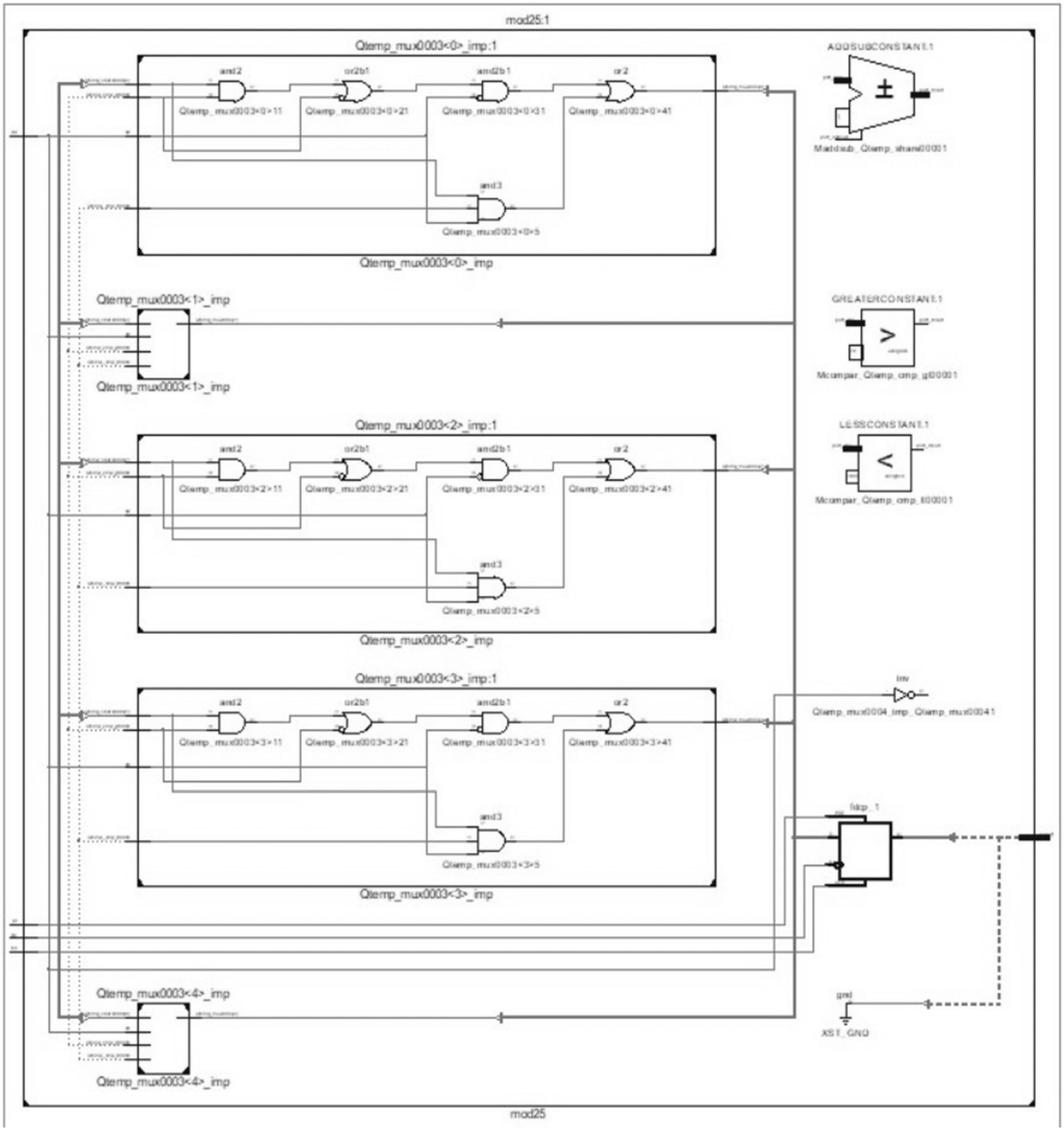
MAIN VHDL MODEL (MVM)

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

entity mod25 is
  Port ( rst : in STD_LOGIC;
                         pr : in STD_LOGIC;
      clk : in  STD_LOGIC;
      dir : in STD_LOGIC;
      Q : out STD_LOGIC_VECTOR (4 downto 0));
end mod25;

architecture mod25_arch of mod25 is
signal Qtemp : STD_LOGIC_VECTOR (4 downto 0) := "00000";

begin
        process(rst,pr,clk,dir)

        begin
                if rst ='1' then
                        Qtemp <= (OTHERS =>'0');
                elsif pr='1' then
                        Qtemp <= (OTHERS =>'1');
                elsif falling_edge(clk) then
                        if dir = '1' then
                                        if Qtemp < 24 then
                                                Qtemp <= Qtemp + 1;
                                        else
                                                Qtemp <= "00000";
                                        end if;
                        else
                                        if Qtemp > 7 then
                                                Qtemp <= Qtemp - 1;
                                        else
                                                Qtemp <= "11111";
                                        end if;
                        end if;
                end if;

        end process;
        Q<=Qtemp;


end mod25_arch;
```

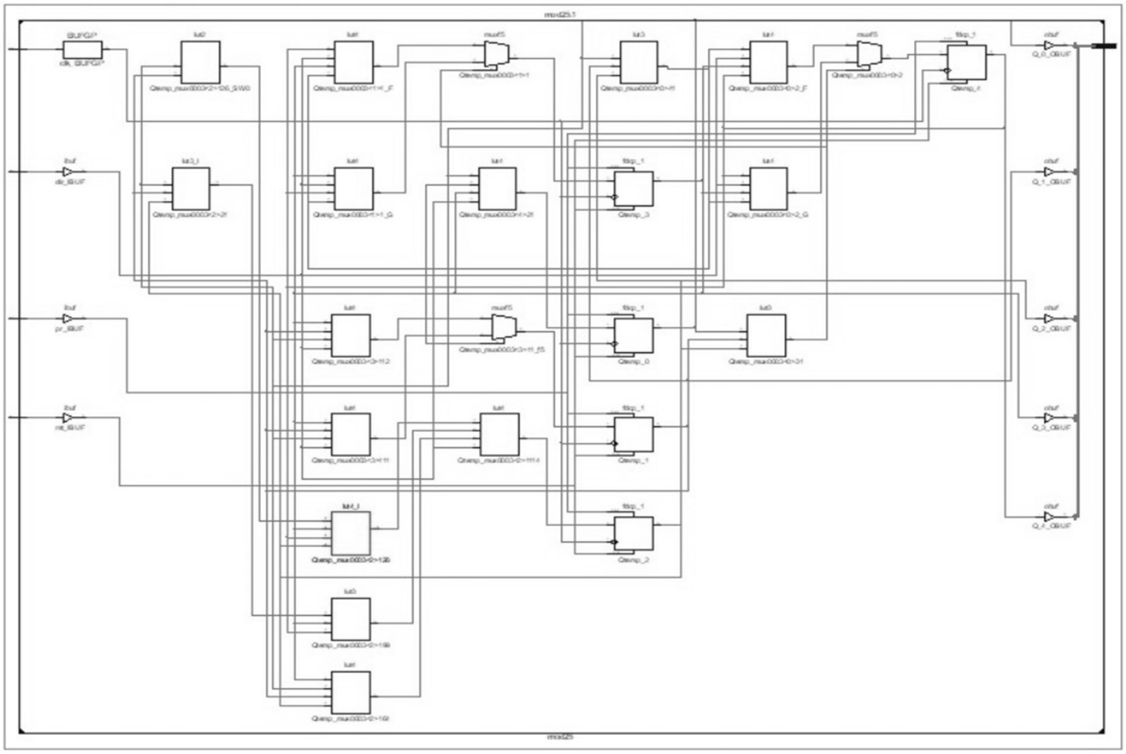
RTL SCHEMATIC:

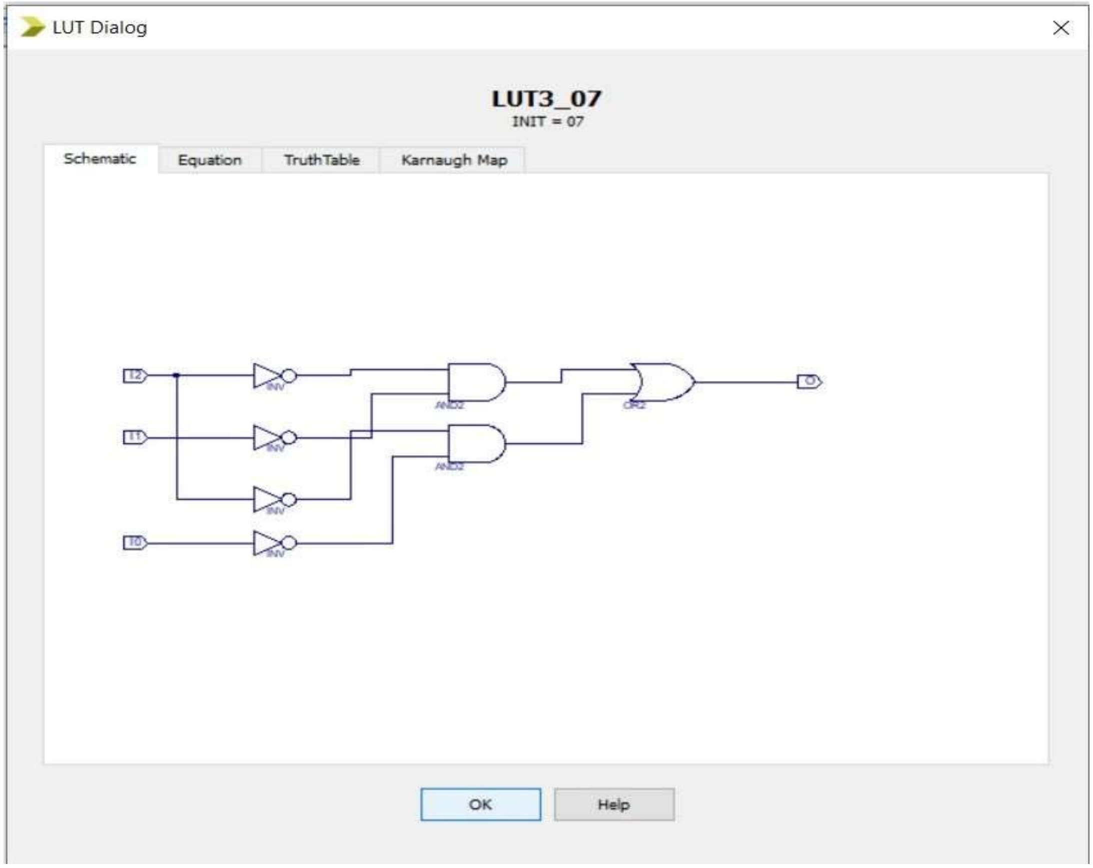



mod25:1
Qtemp_mux0003<0>_imp
Qtemp_mux0003<0>_imp
ADDSUBCONSTANT:1
±
Maddsub Qtemp share00001
Qtemp_mux0003<1>_imp
Qtemp_mux0003<1>_imp
GREATERCONSTANT:1
Block: mod25
Type: mod25:1
Mcompar Qtemp cmp gt00001
Qtemp_mux0003<2>_imp
Qtemp_mux0003<2>_imp
LESSCONSTANT:1
<
Mcompar Qtemp cmp lt00001
Qtemp_mux0003<3>_imp
inv
Qtemp mux0004 imp Qtemp mux00041
Qtemp_mux0003<3>_imp
Qtemp_mux0003<4>_imp
fdcp 1
Qtemp_mux0003<4>_imp
gnd
XST GND
mod25


mod25:1
Qtemp_mux0003<0>_imp:1
and2
or2b1
and2b1
or2
Qtemp_mux0003<0>11
Qtemp_mux0003<0>21
Qtemp_mux0003<0>31
Qtemp_mux0003<0>41
and3
Qtemp_mux0003<0>5
Qtemp_mux0003<0>_imp
ADDSUBCONSTANT.1
±
Maddsub_Qtemp_share0001
Qtemp_mux0003<1>_imp
Qtemp_mux0003<1>_imp
GREATERCONSTANT.1
>
Mcompar_Qtemp_cmp_gt0001
LESSCONSTANT.1
<
Mcompar_Qtemp_cmp_lt0001
Qtemp_mux0003<2>_imp:1
and2
or2b1
and2b1
or2
Qtemp_mux0003<2>11
Qtemp_mux0003<2>21
Qtemp_mux0003<2>31
Qtemp_mux0003<2>41
and3
Qtemp_mux0003<2>5
Qtemp_mux0003<2>_imp
Qtemp_mux0003<3>_imp:1
and2
or2b1
and2b1
or2
Qtemp_mux0003<3>11
Qtemp_mux0003<3>21
Qtemp_mux0003<3>31
Qtemp_mux0003<3>41
inv
Qtemp_mux0004_imp_Qtemp_mux00041
and3
Qtemp_mux0003<3>5
fdcp_1
Qtemp_mux0003<3>_imp
Qtemp_mux0003<4>_imp
gnd
XST_GND
Qtemp_mux0003<4>_imp
mod25

TECHNOLOGY SCHEMATIC





LUT Dialog

LUT3_07

INIT = 07

SYNTHESIS REPORT

a) Device Utilization Summary:

```
=================================================================
*                  Final Report                    *
=================================================================
Final Results
RTL Top Level Output File Name          :
mod25.ngr Top Level Output File Name    :
mod25 Output Format       : NGC
Optimization Goal              : Speed
Keep Hierarchy                 :
No Design Statistics
# IOs                   9
Cell Usage :
# BELS                   18
#     LUT2                1
#     LUT3                3
#     LUT3_L              1
#     LUT4                9
#     LUT4_L              1
#     MUXF5               3
# FlipFlops/Latches       5
#     FDCP_1              5
# Clock Buffers           1
#     BUFGP               1
# IO Buffers              8
#     IBUF                3
#     OBUF                5
=================================================================
Device utilization summary:
---------------------------

Selected Device : 3s250epq208-5
 Number of Slices:                  8  out of   2448     0%
 Number of Slice Flip Flops:        5  out of   4896     0%
 Number of 4 input LUTs:           15  out of   4896
                                   0% Number of IOs:
                                    9
 Number of bonded IOBs:             9  out of    158     5%
 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.
```

Speed Grade: -5

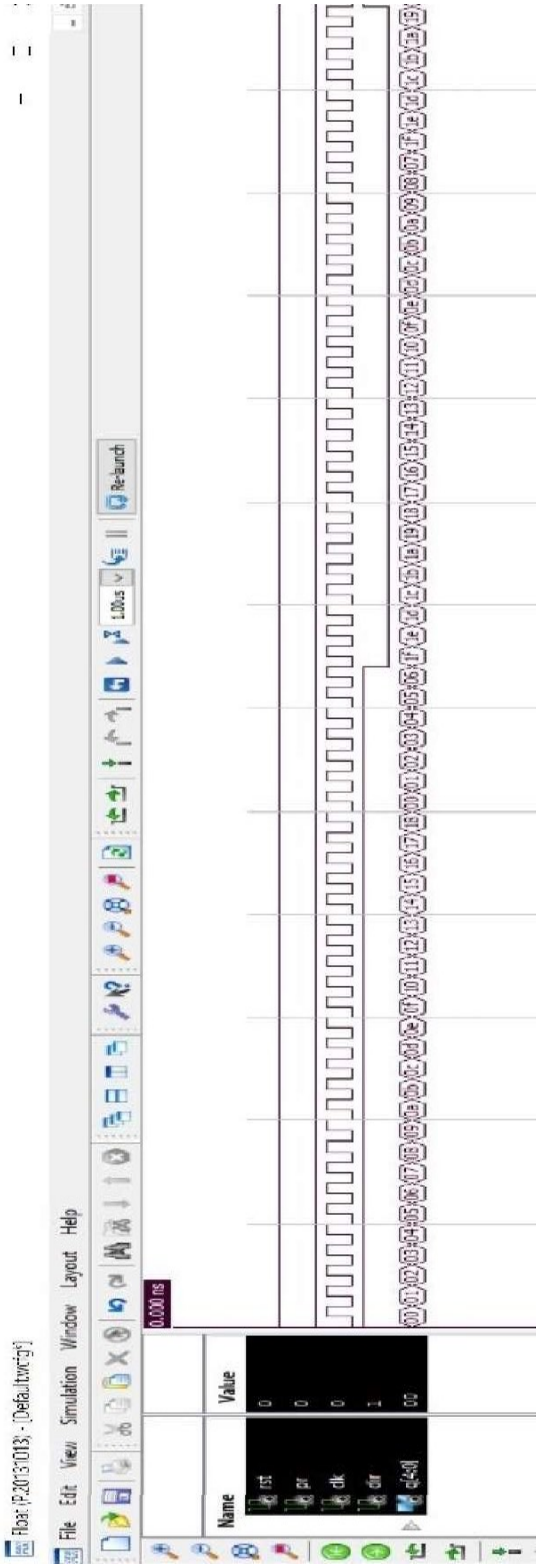
```
  Minimum period: 3.876ns (Maximum Frequency: 257.968MHz)
  Minimum input arrival time before clock: 3.059ns
  Maximum output required time after clock: 4.476ns
  Maximum combinational path delay: No path found
```

TESTBENCH VHDL MODEL (TVM)

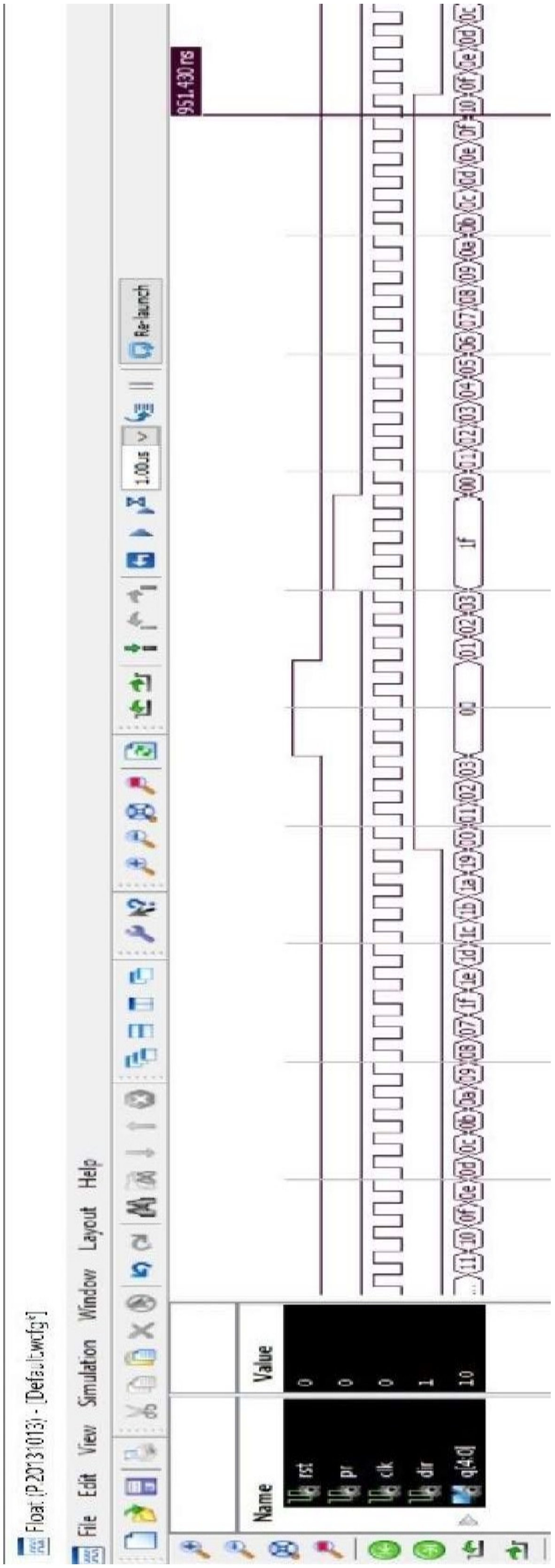
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY mod25_tb IS
END mod25_tb;

ARCHITECTURE behavior OF mod25_tb IS

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

   COMPONENT
   mod25 PORT(
       rst : IN  std_logic;
                                pr : IN std_logic;
       clk : IN std_logic;
       dir : IN  std_logic;
       Q : OUT std_logic_vector(4 downto 0)
      );
   END COMPONENT;


  --Inputs
  signal rst : std_logic := '0';
           signal pr : std_logic := '0';
  signal clk : std_logic := '0';
  signal dir : std_logic := '0';

           --Outputs
  signal Q : std_logic_vector(4 downto 0);

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

BEGIN

           -- Instantiate the Unit Under Test (UUT)
  uut: mod25 PORT MAP (
        rst => rst,
                                 pr => pr,
        clk      =>
        clk,     dir
        =>  dir, Q
        => Q
      );

  -- Clock process definitions
  clk_process :process
  begin
                    clk <= '0';
                    wait forclk_period/2;
                    clk <= '1';
```

```
              wait for clk_period/2;
  end process;

  -- Stimulus process

       stim_proc_dir: process
  begin

              dir <= not(dir);
              wait for 320 ns;

        end process;

  stim_proc_rst: process
  begin
    wait for 680 ns;
              rst <= '1';
              wait for 40 ns;
              rst <= '0'; wait;

        end process;

        stim_proc_pr: process
  begin
    wait for 750 ns;
              pr <= '1';
              wait for 40 ns;
              pr <= '0';
              wait;

        end process;

END;
```

ISIM WAVEFORMS


Float (P.20131013) - [Default.wcfg*]
File Edit View Simulation Window Layout Help
1.00us
Re-launch
0.000 ns

Name	Value
rst	0
pr	0
clk	0
dir	1
q[4:0]	00


Float (P.20131013) - [Default.wcfg*]
File Edit View Simulation Window Layout Help
1.00us
Re-launch
951.430 ns

Name	Value
rst	0
pr	0
clk	0
dir	1
q[4:0]	10

PIN-LOCKING REPORT

```
# PlanAhead Generated physical constraints

NET "Q[4]" LOC =
P164; NET "Q[3]" LOC
= P162; NET "Q[2]"
LOC = P161; NET
"Q[1]" LOC = P160;
NET "Q[0]" LOC =
P153; NET "clk" LOC =
P132; NET "dir" LOC =
P202; NET "pr" LOC =
P204; NET "rst" LOC =
P194;
```

CONCLUSION:-

Group B.

To prepare CMOS layout in selected technology, simulate with and without capacitive load, comment on rise, and fall times.

The MICROWIND software allows the designer to simulate and design an integrated circuit at physical description level. Born in Toulouse (France), Microwind is an innovative CMOS design tool for educational market. It is developed as comprehensive package on windows platform to enable students to learn smart design methods and techniques with more practice. With inbuilt layout editing tools, mix-signal simulator, MOS characteristic viewer and more, it allows students to learn complete design process with ease. It unifies schematic entry, pattern based simulator, SPICE extraction of schematic, Verilog extractor, layout compilation, on layout mix-signal circuit simulation, cross sectional & 3D viewer, netlist extraction, BSIM4 tutorial on MOS devices and sign-off correlation to deliver unmatched design performance and productivity.

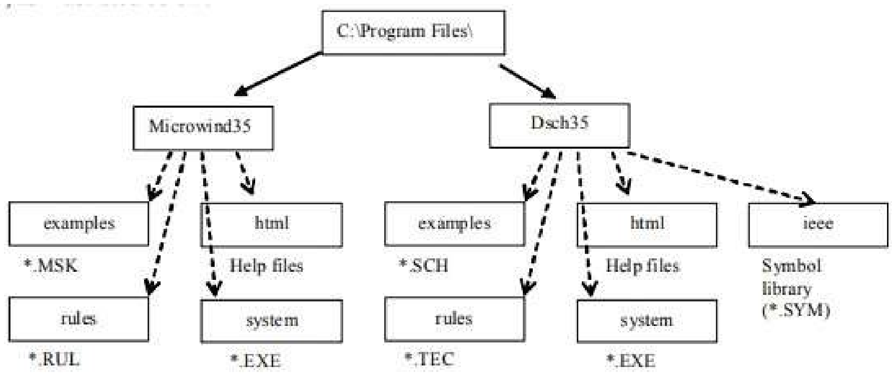
The MICROWIND program allows the student to design and simulate an integrated circuit at physical description level. The package contains a library of common logic and analog ICs to view and simulate. MICROWIND includes all the commands for a mask editor as well as original tools never gathered before in a single module (2D and 3D process view, Verilog compiler, tutorial on MOS devices). You can gain access to Circuit Simulation by pressing one single key. The electric extraction of your circuit is automatically performed and the analog simulator produces voltage and current curves immediately.

The DSCH program is a logic editor and simulator. DSCH is used to validate the architecture of the logic circuit before the microelectronics design is started. DSCH provides a user-friendly environment for hierarchical logic design, and fast simulation with delay analysis, which allows the design and validation of complex logic structures.
DSCH also features the symbols, models and assembly support for 8051 and PIC16F84 controllers. Designers can create logic circuits for interfacing with these controllers and verify software programs using DSCH.

Highlights

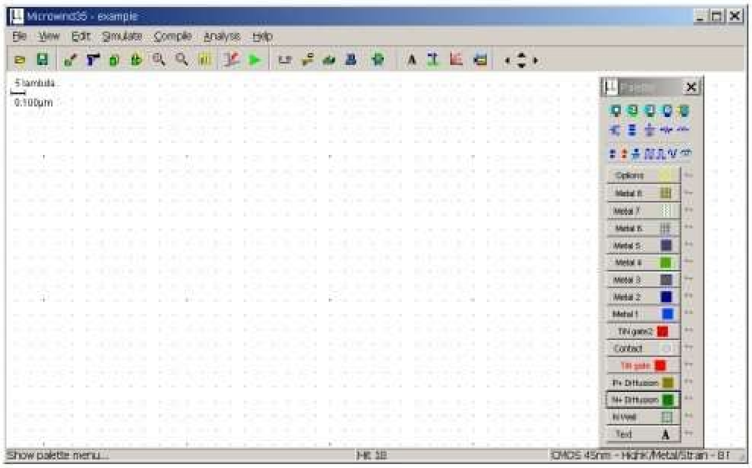
- User-friendly environment for rapid design of logic circuits.
- Supports hierarchical logic design.
- Added a tool on fault analysis at the gate level of digital. Faults: Stuck-at-1, stuck-at-0. The technique allows injection of single stuck-at fault at the nodes of the circuit.
- Improved interface between DSCH and Winspice.
- Handles both conventional pattern-based logic simulation and intuitive on screen mouse-driven simulation.
- Built-in extractor which generates a SPICE netlist from the schematic diagram (Compatible with PSPICETM and WinSpiceTM).
- Generates a VERILOG description of the schematic for layout conversion.
- Immediate access to symbol properties (Dela 7fanout).

- Model and assembly support for 8051 and PIC 16F84 microcontrollers.
- Sub-micron, deep-submicron, nanoscale technology support.
- Supported by huge symbol library



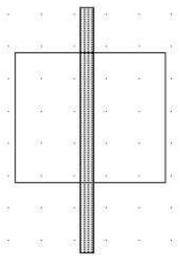
MOS layout:

Double-click on the MICROWIND icon. The MICROWIND display window includes four main windows: the main menu, the layout display window, the icon menu and the layer palette. The layout window features a grid, scaled in lambda (X) units. The lambda unit is fixed to half of the minimum available lithography of the technology. The default technology is a CMOS 8-metal layers 45 nm technology. In this technology, lambda is 0.02 pm (40 rim).

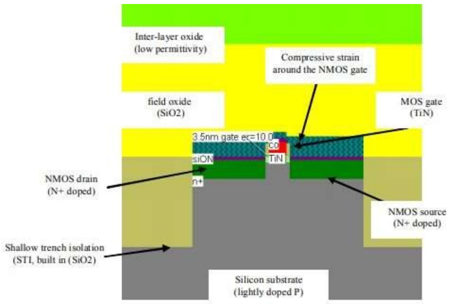


The palette is located in the lower right corner of the screen. A red color indicates the current layer. Initially the selected layer in the palette is polysilicon. By using the following procedure, you can create a manual design of the n-channel MOS. fi Fix the first corner of the box with the mouse.

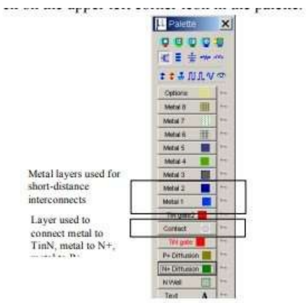
While keeping the mouse button pressed, move the mouse to the opposite corner of the box. Release the button. This creates a box in polysilicon layer as shown in Figure 2-3. The box width should not be inferior to 2 k, which is the minimum width of the polysilicon box. Change the current layer into N+ diffusion by a click on the palette of the Diffusion N+ button. Make sure that the red layer is now the N+ Diffusion. Draw a n-diffusion box at the bottom of the drawing as in Figure 2-3. N-diffusion boxes are represented in green. The intersection between diffusion and polysilicon creates the channel of the nMOS device.



Creating Nchannel MOS

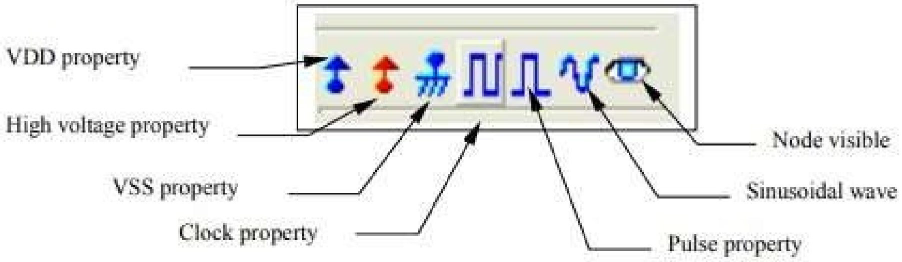

Inter-layer oxide (low permittivity)
Compressive strain around the NMOS gate
field oxide (SiO2)
MOS gate (TiN)
3.5nm gate ec=10.0
siON
TiN
n+
NMOS drain (N+ doped)
NMOS source (N+ doped)
Shallow trench isolation (STI, built in (SiO2)
Silicon substrate (lightly doped P)

Cross sectionof NMOS

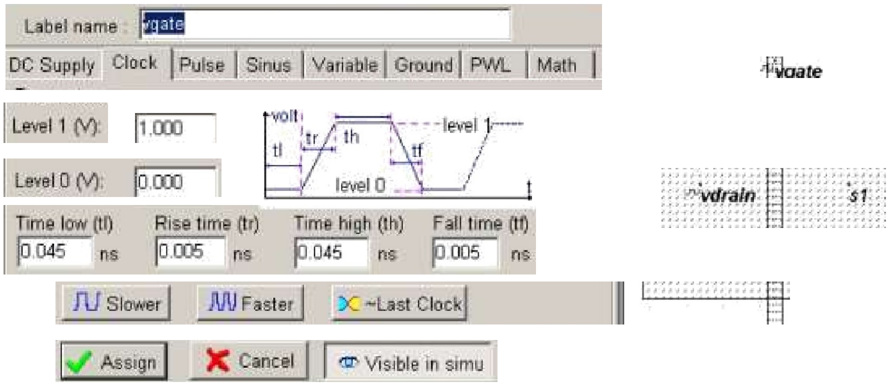


Metal layers

The physical properties of the source and of the drain are exactly the same. Theoretically, the source is the origin of channel impurities. In the case of this nMOS device, the channel impurities are the electrons. Therefore, the source is the diffusion area with the lowest voltage. The metal gate floats over the channel, and splits the diffusion into 2 zones, the source and the drain. The gate controls the current flow from the drain to the source, both ways. A high voltage on the gate attracts electrons below the gate, creates an electron channel and enables current to flow. A low voltage disables the channel.



Apply a clock to the gate. Click on the Clock icon and then, click on the polysilicon gate. The clock menu appears again. Change the name into Vgate and click on OK to apply a clock with 0.1 ns period (45 ps at “0”, 5 ps rise, 45 ps at “1”, 5 ps fal1)



Label name : vgate

DC Supply | Clock | Pulse | Sinus | Variable | Ground | PWL | Math

Level 1 (V): 1.000

Level 0 (V): 0.000

Time low (tl) 0.045 ns

Rise time (tr) 0.005 ns

Time high (th) 0.045 ns

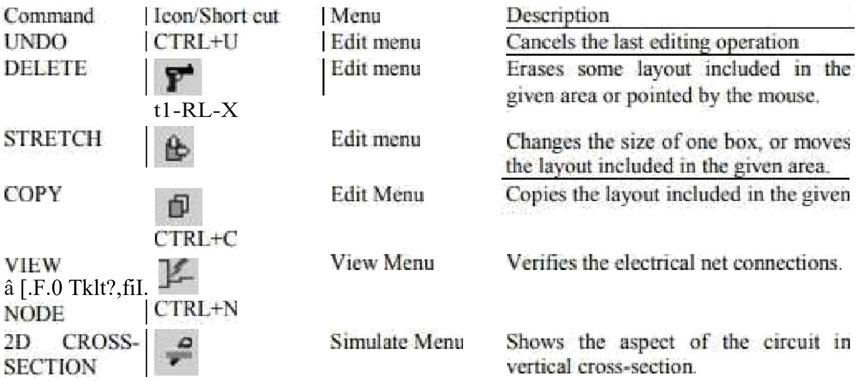
Fall time (tf) 0.005 ns

The clock menu and the clock property insertion directly on the MOS layout

Layout and simulation of the p-channel MOS (mypmos.MSK)

Useful Editing Tools

The inline ing c nmmands may iielp yt›u in the layout desieirand veriticalion processes.



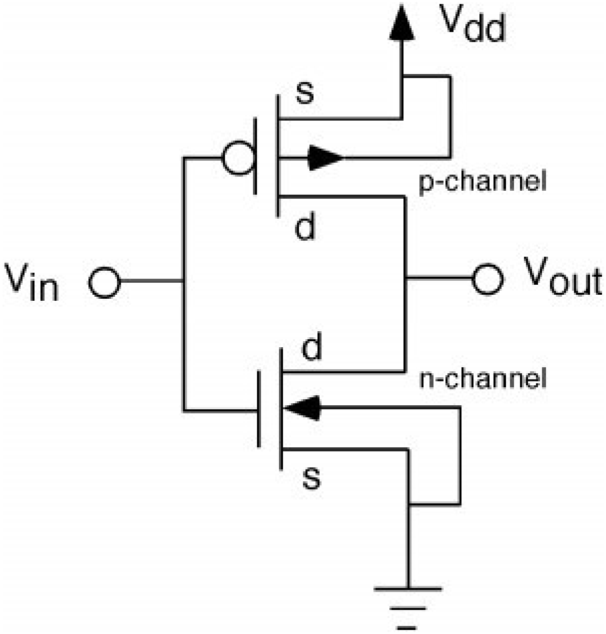
Command	Icon/Short cut	Menu	Description
UNDO	CTRL+U	Edit menu	Cancels the last editing operation
DELETE	t1-RL-X	Edit menu	Erases some layout included in the given area or pointed by the mouse.
STRETCH		Edit menu	Changes the size of one box, or moves the layout included in the given area.
COPY	CTRL+C	Edit Menu	Copies the layout included in the given
VIEW â [.F.0 Tklt?,fiI. NODE	CTRL+N	View Menu	Verifies the electrical net connections.
2D CROSS-SECTION		Simulate Menu	Shows the aspect of the circuit in vertical cross-section.

Expt . No: 6 Date :	Inverter, NAND, NOR gates

Aim: To prepare CMOS layout in selected technology, simulate with and without capacitive load, comment on rise, and fall times for Inverter, NAND, NOR gates.

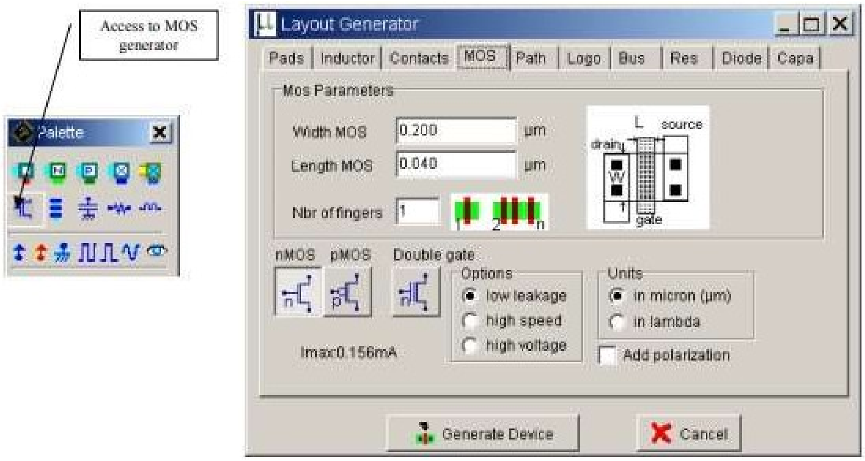
Description:

Here the p-channel MOS and the n-channel MOS transistors function as switches. When the input signal is logic 0 (Figure 3-4 left), the nMOS is switched off while PMOS passes VDD through the output. When the input signal is logic 1 (Figure 3-4 right), the pMOS is switched off while the nMOS passes VSS to the output.



A. INVERTER CMOS LAYOUT

Click the icon MOS generator on the palette. The following window appears. By default the proposed length is the minimum length available in the technology (2 lambda), and the width is 10 lambda. In 45-nm technology, where lambda is 20 rim (0.02 pm), the corresponding size is 0.02 pm for the length and 0.04 pm for the width. Simply click Generate Device, and click on the middle of the screen to fix the MOS device.



Width MOS 0.200 µm

Length MOS 0.040 µm

Nbr of fingers 1

● low leakage
○ high speed
○ high voltage

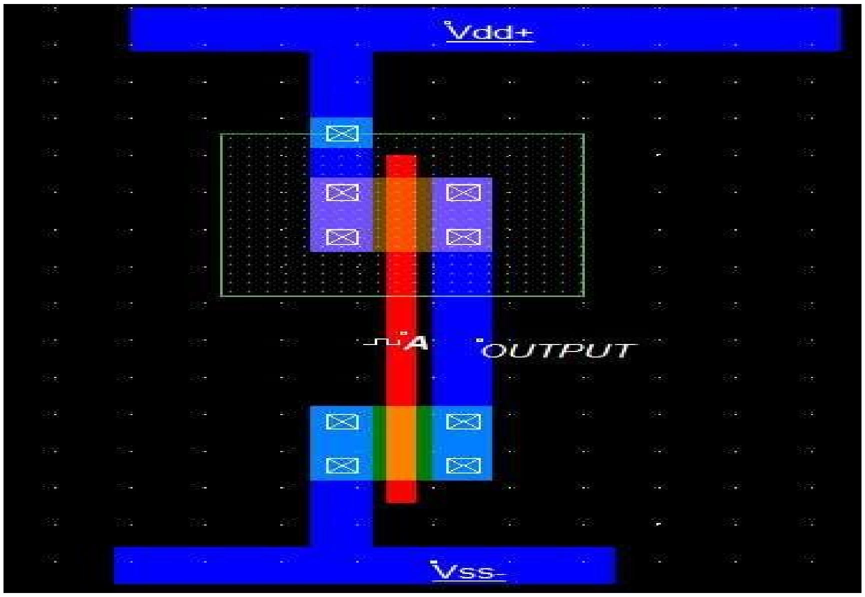
● in micron (µm)
○ in lambda

□ Add polarization

Imax:0.156mA

Generating a nMOS device

WITHOUT CAPACITIVE LOAD:



Output=l if A=0 & Output= 0 if A= 1

B. NAND CMOS LAYOUT WITHOUT CAPACITIVE LOAD

C. NOR CMOS LAYOUT WITHOUT CAPACITIVE LOAD

D. Half Adder (Carry is NAND+INV=AND)

The command Simulate —› Run Simulation gives access to the analog simulation. Select the simulation mode Voltage vs. Time. The analog simulation of the circuit is performed. The time domain waveform, proposed by default, details the evolution of the voltages in1 and outI versus time. This mode is also called transient simulation, as shown in figure 3-15. The truth-table is verified as follows. A logic “0” corresponds to 0 V a logic "1" to a 1. 0 V. When the input rises to "1", the output falls to "0", with a 7 pico-second delay (7.10-12 second). The reason why the delay is larger before time 1.0 ns is that the circuit is "warming up" as the voltage supply suddenly rises from 0 to VDD at time= 0.0ns. The steady-state is reached at time=1.0 ns.

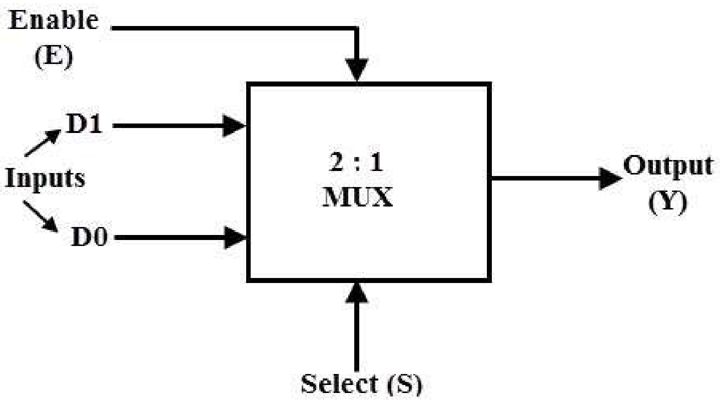
Observation:

Conclusion:

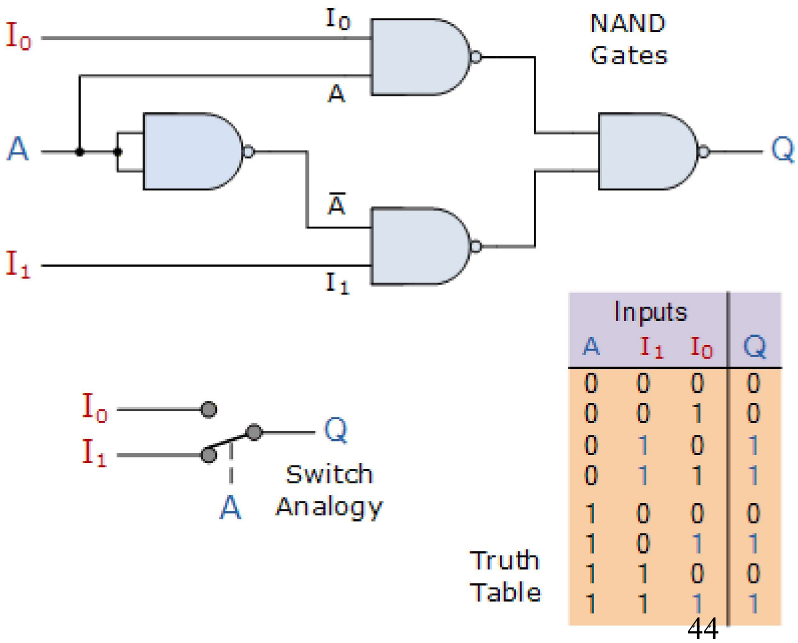
6.Expt . No: Date:	2:1 multiplexer using logic gates and transmission gates.

Aim: To prepare CMOS layout in selected technology, simulate with and without capacitive load, comment on rise, and fall times for 2:1 multiplexer using logic gates and transmission gates.

Description:

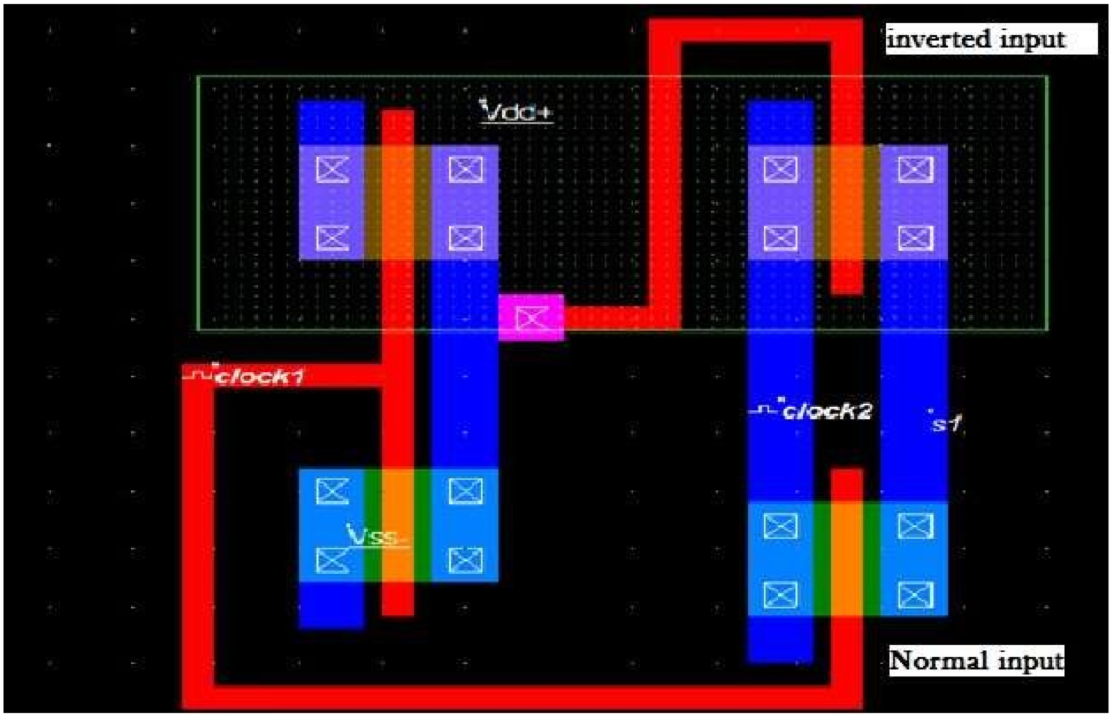


Transmission gate: Transmits input(clk2) tO the

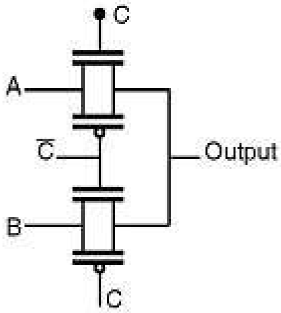


Inputs			
A	I1	I0	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

output(s1) if clk1=1



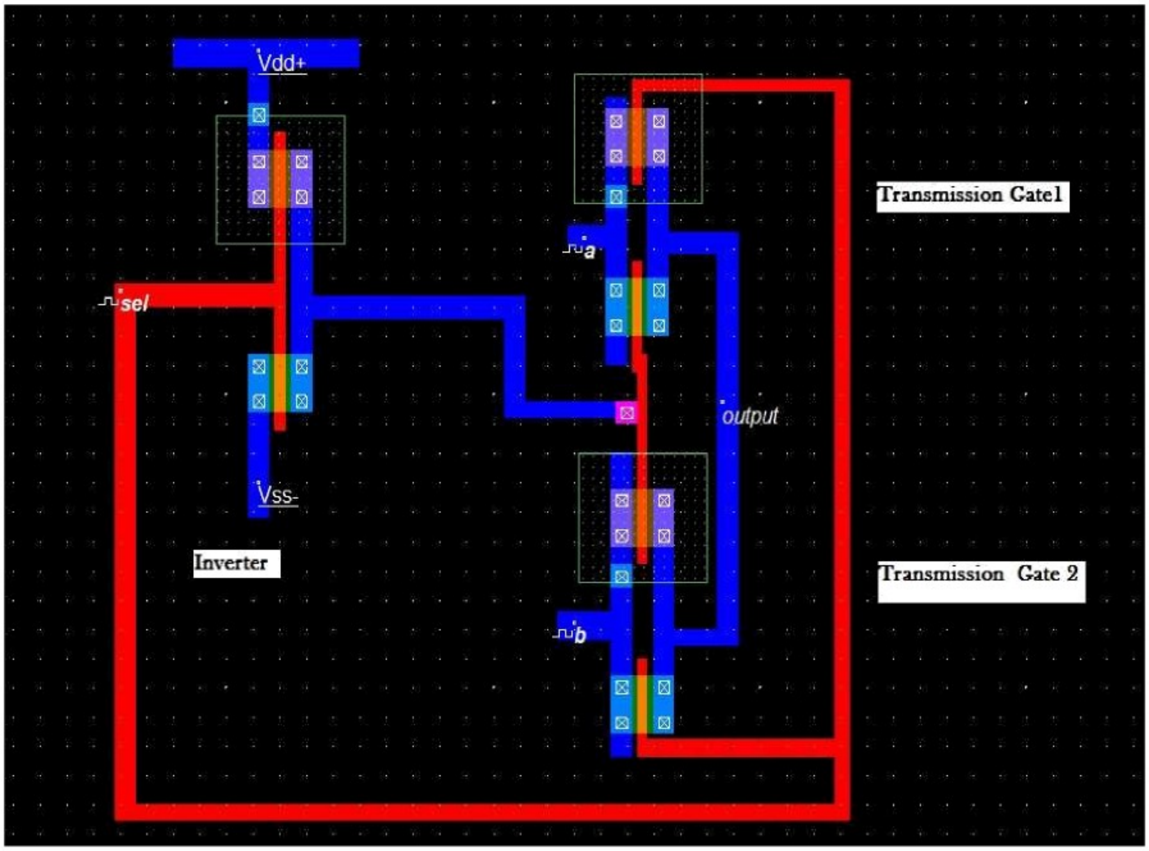
C1ockl=l then S1=C1ock2



2 : 1 MUX using transmission gate

2:1 MUX using Transmission gates

CMOS Transmission Gates: fi A Transmission Gate (TG) is a complementary CMOS switch. - Both transistors are ON or OFF simultaneously. - The NMOS switch passes a good zero but a poor 1. - The PMOS switch passes a good one but a poor 0. - Combining them we get a good 0 and a good 1 passed in both directions - - - Circuit Symbols for TGs: - TGs are efficient in implementing some functions such as multiplexers, XORs, XNORs, latches, and Flip-Flops. • 2 I./P XOR using TGs: F = A.B' + A'.B , we need
o:T GA) F B ass B' to F) if A=0 F = B (pass B to F) using TGs: 8 Ts (2 inverters for A and B and
V ll s 12 T



When Se1=0 =>output ='a' input, & Sel=1 =>output = 'b' input

Observation:

Conclusion:

7.Expt . No: **Date:**	Single bit SRAM cell.

Aim: To prepare CMOS layout in selected technology, simulate with and without capacitive load, **comment on** rise, **and fall times for Single bit SRAM** cell.

Description:

Static Random-Access Memory (static RAM or SRAM) is a type of semiconductor memory that uses bistable latching circuitry (flip-flop) to store each bit. It is volatile in the conventional sense that data is eventually lost when the memory is not powered.

The term static differentiates SRAM from DRAM (dynamic random-access memory) which must be periodically refreshed. SRAM is faster and more expensive than DRAM; it is typically used for CPU cache while DRAM is used for a computer's main memory.

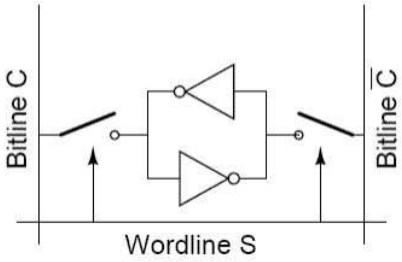
Advantages:

Low power consumption
Simplicity — a refresh circuit is not needed
Reliability

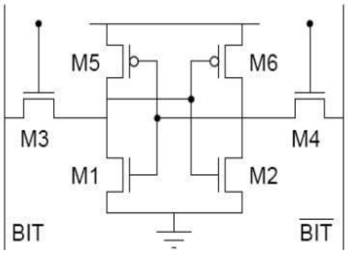
Disadvantages:

Price
Capacity
Varying power consumption

SRAM Basics The memory circuit is said to be static if the stored data can be retained indefinitely, as long as the power supply is on, without any need for periodic refresh operation. The data storage cell, i.e., the one-bit memory cell in the static RAM arrays, invariably consists of a simple latch circuit with two stable operating points. Depending on the preserved state of the two inverter latch circuit, the data being held in the memory cell will be interpreted either as logic '0' or as logic ’1'. To access the data contained in the memory cell via a bit line, we need at least one switch, which is controlled by the corresponding word line as shown in Figure



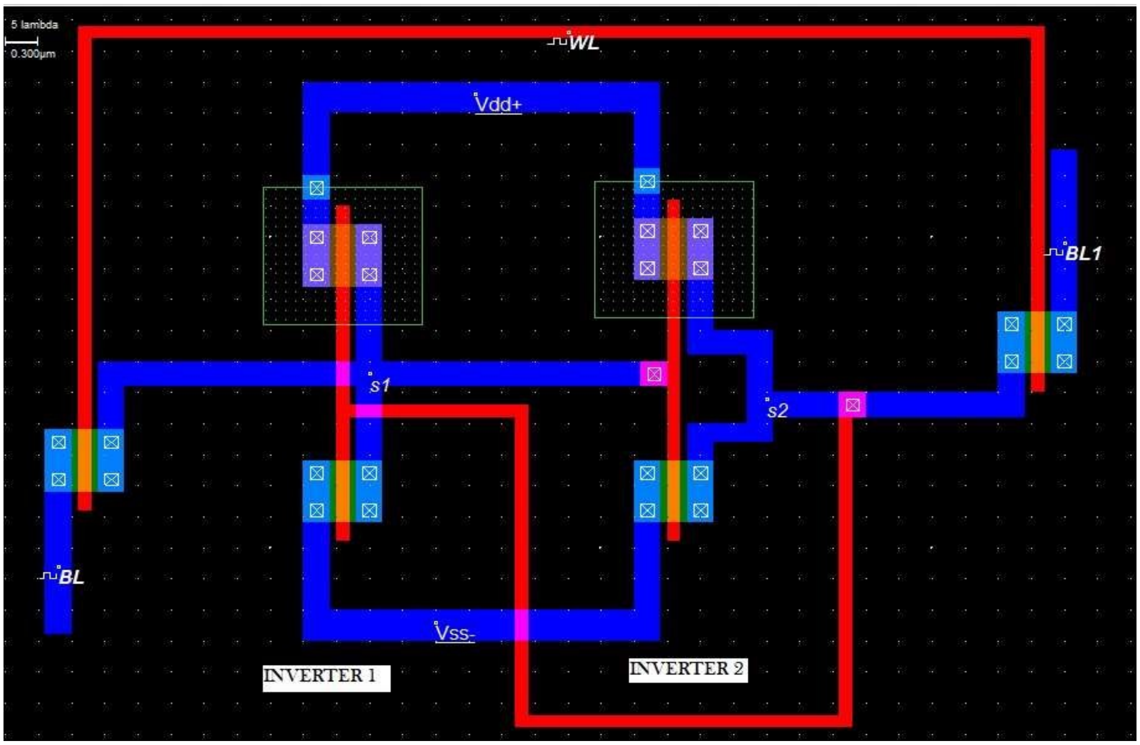
SRAM



CMOS SRAM

READ Operation Consider a data read operation, assuming that logic '0' is stored in the cell. The transistors M2 and M5 are turned off, while the transistors Ml and M6 operate in linear mode. Thus internal node voltages are V1 = 0 and V2 = VDD before the cell access transistors are turned on.

WRITE Operation Consider the write '0' operation assuming that logic '1' is stored in the SRAM cell initially. Figure 28.51 shows the voltage levels in the CMOS SRAM cell at the beginning of the data write operation. The transistors Ml and M6 are turned off, while M2 and M5 are operating in the linear mode. Thus the internal node voltage VI = VDD and V2 = 0 before the access transistors are turned on. The column voltage Vb is forced to '0' by the write circuitry. Once M3 and M4 are turned on, we expect the nodal voltage V2 to remain below the threshold voltage of Ml, since M2 and M4



Observation:

Conclusion: