SUMMER – 13 EXAMINATION

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.

2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.

3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).

4) While assessing figures, examiner may give credit for principal components indicated in the
figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.

5) Credits may be given step wise for numerical problems. In some cases, the assumed constant
values may vary and there may be some difference in the candidate's answers and model answer.

6) In case of some questions credit may be given by judgement on part of examiner of relevant
answer based on candidate's understanding.

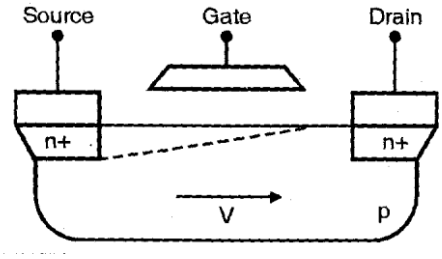
7) For programming language papers, credit may be given to any other program based on equivalent concept.

1. a) Attempt any THREE of the following :

12M

i) Explain the effect of channel length modulation on drain current ID.

Ans. (Diagram -1 mark,Suitable Explanation – 3 Marks)



- Channel length modulation is a shortening of the length of the inverted channel region with increase in drain bias for large drain biases. So increase in current with drain bias and a reduction of output resistance.
- The channel is formed by attraction of carriers to the gate and the current drawn through channel is nearly a constant independent of drain voltage in saturation mode. However, near the drain, the gate and drain jointly determine the electric field pattern.
- Instead of flowing in channel, beyond the pinch-off point the carriers flow in subsurface pattern mode possible because the drain and the gate both control the current.

ii) Explain the constant field scaling.

Ans. (introduction to Scaling -1 mark, Any 6 parameters – 3 marks)

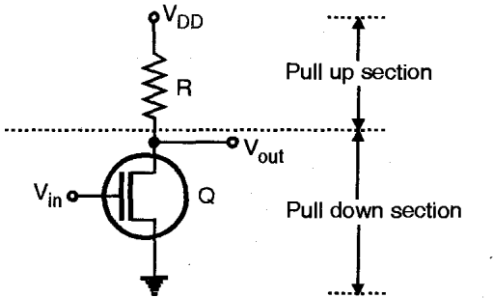
- Scaling refers to the decrease in dimensions of the device.
- Scaling generally leads to enhanced performance in terms of speed and parking density.
- Constant field scaling yields the largest reduction the power-delay product of a single transistor.
- It is not alt the magnitude of the internal fields in the MOSFET. dimensions are only scaled by a factor S.
- Also the poten must be scaled down proportionally.
- The effect of constant field scaling on device parameters is given below

Parameters	Scaling
Voltage	1/S
Gate area	$1/S^2$
Gate oxide capacitance	S
Gate capacitance	1/S

Parasitic capacitance	1/S
Carriers density	–
Saturation current	1/S
Current density	S
Power dissipation	$1/S^2$
Gates per unit area	S^2
Gate delay	1/S
Frequency	S
Speed power product	$1/S^3$

iii) Draw and explain working of resistive load inverter.

Ans.



- The basic inverter circuit requires an nMOS transistor with source connected to ground and load resistor 'R' are connected between drain and VDD.
- Here the output is taken from the drain and input is applied at the gateof MOS Inverter.
- When logic 0 is applied at the input of the inverter, as you all know by now, that transistor Q is in cut off state as channel is not formed between the source and drain making drain current ID = 0 and output high.
- When logic 1 is applied at the input, transistor conducts due to formation of channel between the source and drain allowing drain current ID to flow as a result of which the output voltage drops to zero.
- This means that ground gets connected to the output terminal and output voltage becomes zero.

iv) Explain the syntax of component initialization and write a component of NOT gate using VHDL.

- The component represents a precompiled entity architecture pair.

- We write a program for a component in VHDL and use it as an instance in other programs without writing its code again.
- A component instantiated in a structural description must first be declared using a component declaration.
- A component declaration declares the name and the interface of a component.
- The names of the ports in the declaration may be different from the names of the ports in the entity.

■ EXAMPLE

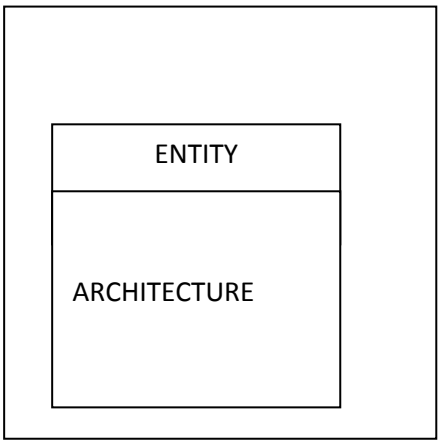
```
library IEEE;
use IEEE.std_logic_1164.all;
ENTITY not_gate IS
  PORT(A,B: IN std_logic;
Y,z: OUT std_logic);
END not_gate;
ARCHITECTURE structural OF not_gate  IS
   COMPONENT NOT1(m: IN std_logic;
                  n:OUT std_logic);
   END COMPONENT;
BEGIN
m0: NOT1  port map (a,y);
m1: NOT1  port map (b,z);
END structural;
```

b) Attempt any ONE of the following 6M

i) Explain entity and architecture with suitable example.

Ans.

(Entity 2marks and Architecture 2 marks)



VHDL program file structure

ENTITY

- All designs are expressed in terms of entities.
- An Entity is the most basic building block in a design.
- Without communication there is no system. In other words it must get some input data from environment & should output some data.
- Without an interface, the system would be useless.
- In VHDL, the systems external interface is described by its entity.(black box view)

PORT

- It is the list of interface pins (signals) of the entity along with their directions (mode) & type.
- Most frequently used types are:
  - Bit
  - Boolean
  - Integer.
  - Real.
  - Std_logic. (is same as BIT with few more advantages)
- Each interface port can have one of the following modes:

- IN: value can only be read within the entity model and can't be written.
- OUT: value can only be updated within the entity model; & can't be read.
- INOUT: value can be read and updated within the entity model.
- BUFFER: value can be read and updated within the entity model but it can't have more than one source.

- entity_name: It is an identifier and defined by the user to the entity name.The identifier for the entity must start with letter followed by arbitrary combination of letters, digits and underscore symbols. It is possible to write an entity without any generics, ports & passive statements (it is used for test benches).

Syntax of an ENTITY

entity<entity_name> is

generic (<generic_list>);

port (<port_list>);

end<entity_name>;

ARCHITECTURE

- An architecture body describes the internal view of an entity. It describes the behavior of the entity.
- An architecture body is used to describe the behavior, data flow, or structure of a design entity.
- Single entity can have several architectures, but architecture cannot be assigned to different entities.
- Architecture may not be used without an entity.Single entity can have ultiple architectures.
- All declarations defined in an entity are fully visible and accessible within each architecture assigned to this entity.
- Different types of statements (i.e. processes, blocks, concurrent signal assignments, component instantiations, etc.) can be used in the same architecture.

Syntax of an ARCHITECTURE

architecture<architecture_name> of <entity_name> is
architecture_declarations (types, signals, constants, subprograms
(functions and procedures),components, and groups.)
begin
concurrent_statements (concurrent signal assignment, process statement,
component instantiation, concurrent procedure call, generate statement, concurrent assertion statement block statement.)
end [architecture] [architecture_name];

EXAMPLE OF A VHDL PROGRAM

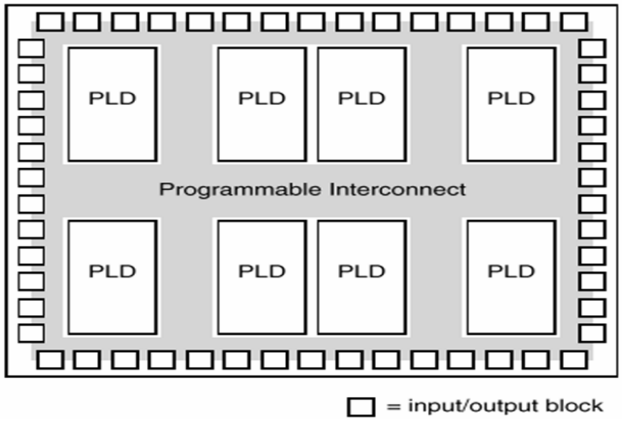
```
LIBRARY IEEE;
Use IEEE.std_logic_1164.all;
entity D_FF is
port (D,CLK : in BIT;
Q : out BIT := '0';
NQ : out BIT := '1' );
end entity D_FF;

architectureBehaviorial of D_FF is
```

```
begin
process (CLK)
begin
if CLK = '1' and CLK'Event then
    Q <=  D;
    NQ <= not D;
end if;
end process;
end  Behaviorial;
```

b) ii) Draw and explain block diagram of CPLD.

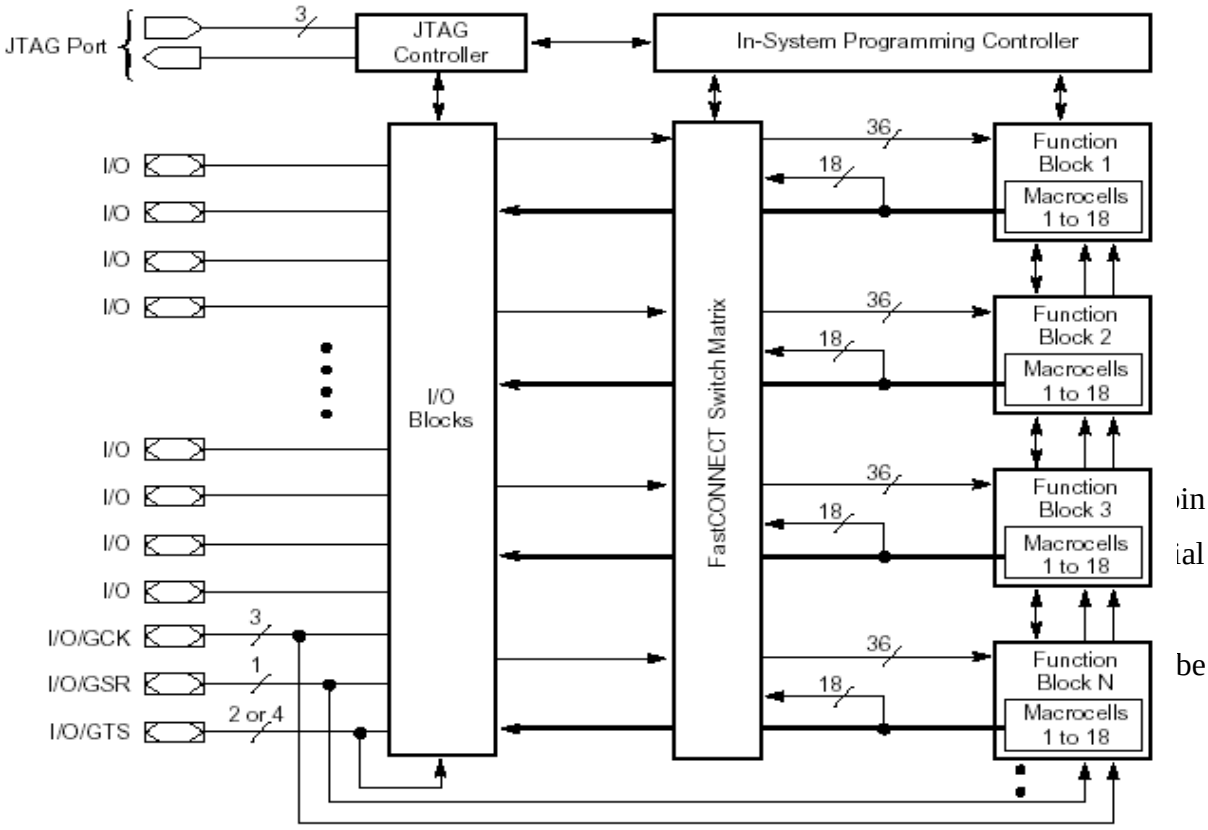
Ans.

Complex programmable logic device is a collection of individual PLD's on a single chip accompanied by a programmable interconnection structure that allows the PLD's to be hooked to each other on the same chip. The I/O blocks are attached to the I/O pins of the chip.



Generalized block diagram of CPLD

ARCHITECTURE OF **XILINX XC9500 CPLD FAMILY**



One pin can be used as a "Global Set/Reset "(GSR).Each macrocell can use this signal as

an asynchronous Preset or Clear.

Two or Four pins depending on the devices can be used as "Global Three State Controls"(GTS).One of the signals can be selected in each macrocell to output enable the corresponding output driver when the macrocell's output is hooked to an external I/O pin.

Only four Functional Blocks(FB) are shown but XC9500 scales to accommodate 16 FB's in the XC95288.Regardless of the specific family member each FB programmable receives 36 signals from the switch matrix. The inputs to the switch matrix are the 18 macrocell outputs from each of the functional blocks and the external inputs from the I/O pins.

Each Functional block also has 18 outputs that run under the switch matrix and connect to the I/O blocks. These are the output-enable signals for the I/O block output drives; they're used when FB macrocells's output is hooked up to an external I/O pin.

Each Functional Block has programmable logic capability with 36 inputs and 18

outputs. Fast Connect Switch Matrix connects all Functional Block outputs to the I/O blocks and the input signals from the I/O block to the Functional Block.

Q2. Attempt any FOUR of the following 16M

a) What is Testing ? Explain need of Testing.

Ans. (Introduction- 2marks,Need -2 Marks)

- Testing is an important technique to ensure that the model is operating as design
- It is used to check the correctness of design.
- Verifying a design via these techniques is certainly more cost effective than testing a fabricated part and then determining that it has a design error that must be fixed.
- A test bench is used to verify the functionality of design.
- The test bench allows the design to verify the functionality of the design at each step in the HDL synthesis based methodology.
- A test bench has three main purposes.
    - To generate stimulus for stimulation (waveforms).
    - To apply this stimulus to the entity under test and collect the output responses.
    - To compare output responses with expected values.
- VHDL testbench is an environment to simulate a VHDL program

b) Explain the concept of Package Declaration and Package Body.

PACKAGE

- The package is a unit that groups various declarations, which can be shared among several designs.
- A package provides convenient mechanism to store and share declarations that are common across many design units. A package is represented by.
    - A package declaration.
    - A package body. (optional).
- Packages are stored in libraries for greater convenience.
- Package declaration may contain a subprogram (function or procedure) declaration; subprogram body is not allowed here and must appear in the package body.

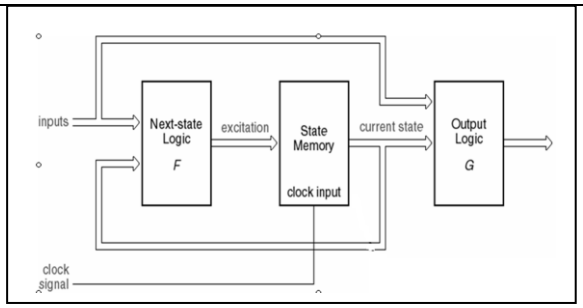
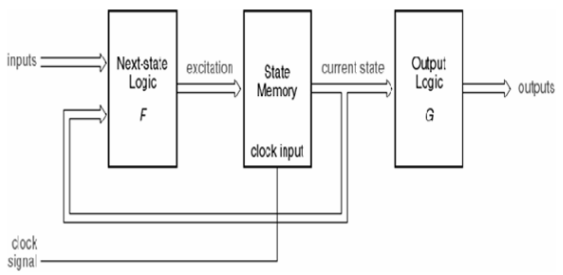
- Package body must accompany a package declaration if the declaration contains subprogram declarations or deferred constants.

```
packagepackage_name is
        package_declarations
end package package_name;
```

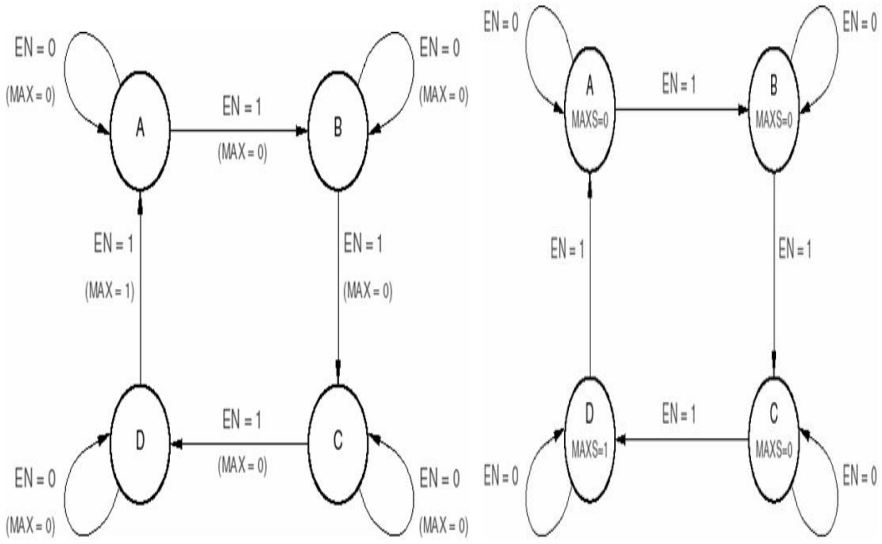
c) Explain Moore and Mealey machine with block diagram.

Ans.

- State Machine actually represents sequential circuit having a transition from previous state to present state.
- The state machine travels through a predefined set of values Eg:- Mode controlled ODD/ EVEN synchronous counter.
- The next state logic is predetermined by the present set of inputs and previous state.
- The output logic is decided by the present state and the present set of inputs.
- If the output depends on the present set of inputs it is called as a Mealy Machine else Moore Machine.

Mealy Machine	**Moore Machine**
	
In a Mealy machine the output depends on the present state and present set of inputs	In a Moore machine the output depends on the present state only.
Number of states required in the design of a Mealy is less in comparison to a Moore machine.	Number of states required in the design of a Moore is more in comparison to a Mealy machine.
Next state = F (Current state, input) Output = G (Current state, input)	Next state = F (Current state, input) Output = G (Current state)

In a state diagram the output is represented on the transition along with input.	In a state diagram the output is represented in the state itself.
Mealy machines are slow in comparison to a Moore Machine.	Moore machines are faster in comparison to a Mealy Machine.



State Diagram of Mealy and MooreState Machine

Qe) Explain Behavioural Modeling with one example.

- The architecture body describes only the behavior of the circuit, without any direct indication as to the hardware implementation.
- The behavior of the entity is expressed using sequentially executed, procedural code, which is very similar in syntax and semantics to that of high level programming language.
- A process statement is the primary mechanism used to model the behavior of an entity. The process is a collection of sequential statements.

process(sensitivity_list)
process_declarations (variables)
begin
sequential_statements
Variable assignment, signal assignment,
wait statement, if statement, case statement, loop statement, null statement, exit statement,
next statement, assertion, report, procedure call return.

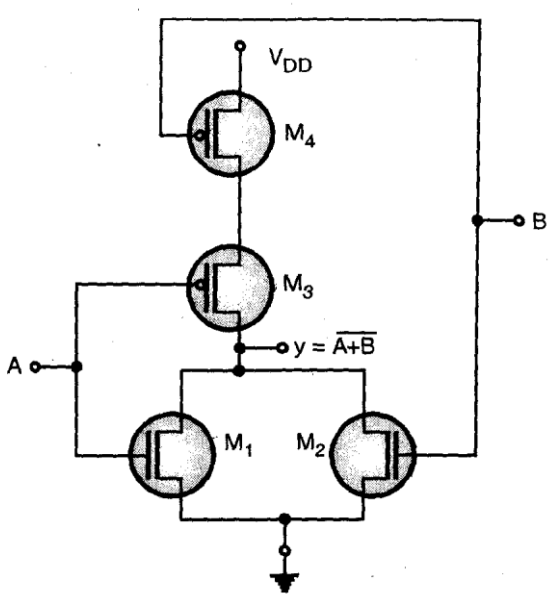
end process [process_label] ;

Example of Behaviorial Style of Modelling(D Flip Flop)

```
entity D_FF is
port (D,CLK : in BIT;
Q : out BIT := '0';
NQ : out BIT := '1' );
end entity D_FF;

architectureBehaviorial of D_FF is
begin
process (CLK)
begin
if CLK = '1' and CLK'Event then
    Q <=  D;
    NQ <= not D;
end if;
end process;
end  Behaviorial;
```

f) Draw and explain working of CMOS NOR gate.

Ans : (Diagram 2 marks+ Explanation 2 marks)



- Since pull down nMOS transistors are in parallel, pull up pMOS transistors are in series.
- Transistors M1 and M3 form one CMOS with input ‘A’ and transistors M2 and M4 form another CMOS with input ‘B’.

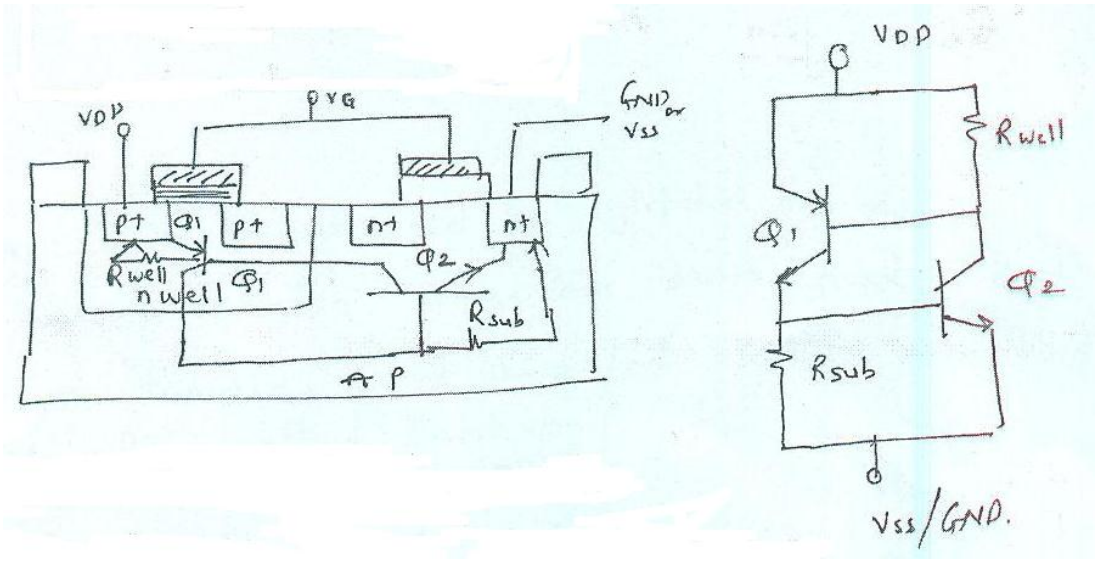
INPUT		MOSFET				OUTPUT
A	B	M1	M2	M3	M4	Y
0	0	OFF	OFF	ON	ON	1
0	1	OFF	ON	ON	OFF	0
1	0	ON	OFF	OFF	ON	0
1	1	ON	ON	OFF	OFF	0

Q5 Attempt any FOUR of the following (16)

a) Explain latch up in CMOS with diagram.

Ans :

Latch up in CMOS is the phenomenon that is inherent to CMOS fabrication process The parasitic transistors and SCRs are formed along with the design circuit. The firing of scr may take due to the imbalance at the input resulting in heavy current from Vdd to GND and damaging the IC



As shown in fig. there are 2 parasitic transistors formed Q1 vertical pnp and Q2 lateral npn transistor. The Q1 has its emitter formed by the p+ diffusion used in pmos transistor. The base is formed by n-well while the collector is formed by p- substrate. Similarly, Q2 has its emitter formed by n+ diffusion used in nmos, p-base is p type silicon substrate and the collector is n-well.

The R-well is the resistance offered by n-well and R substrate resistance offered by p-substrate.

Latch up may be induced by glitches on the supply rails or by incident radiation. Voltage drop across R-well or R- sub turns ON transistors and unwanted current flow between VDD and VSS. This is called Latch up.

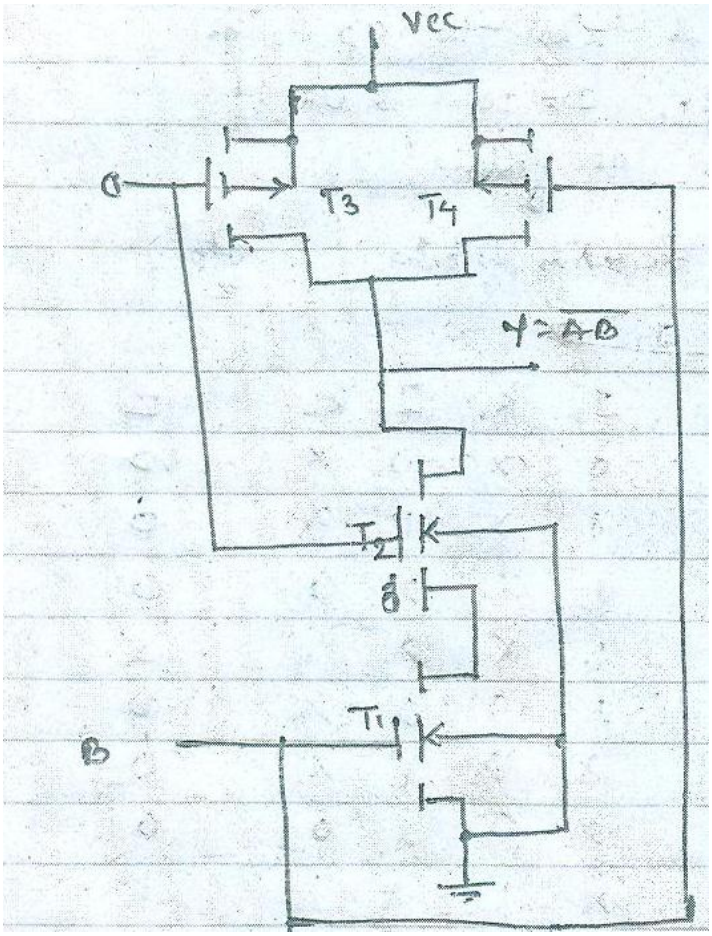
To avoid undesirable parasitic transistors:

1) Guard Rings / channel stops: Additional heavy diffusion region may be same type of material as of substrate or well.
2) Reduce gain: add thin epitaxy layer over substrate.

b) Draw and explain operation of CMOS NAND gate.

Ans :

Dia (2)



Desc (2)

Figure shows 2 Input CMOS NAND Gate where 2 PMOS Transistors T_B&T_4 are connected in parallel and 2 NMOS Transistor T_1& T_2 are connected in series. The transistor will remain ÓN or OFF state as per the following Table to work as a NAND Gate.

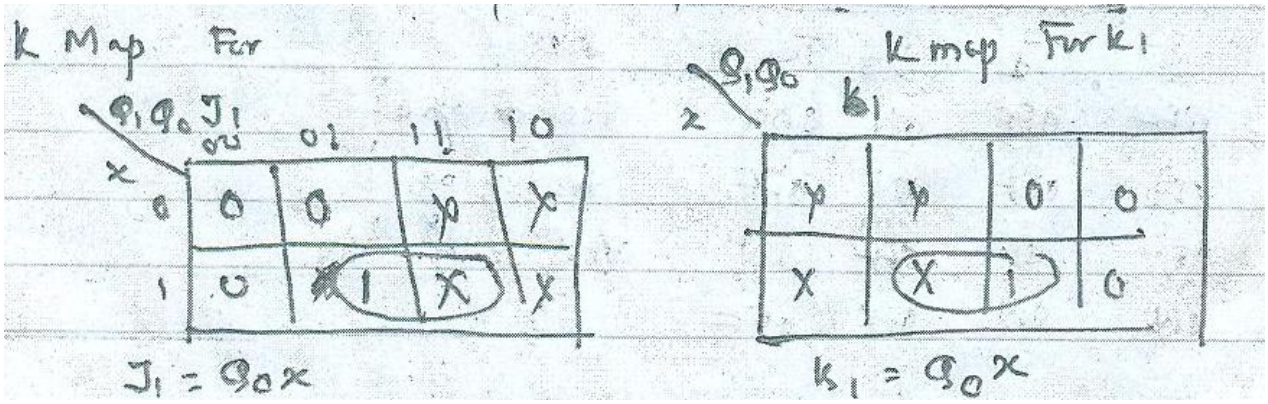
INPUT		T_1	T_2	T_3	T_4	OUTPUT
A	B					
0	0	OFF	OFF	ON	ON	Vcc or logic 1
0	1	ON	OFF	ON	OFF	Vcc or logic 1
1	0	OFF	ON	OFF	ON	Vcc or logic 1
1	1	ON	ON	OFF	OFF	“0”

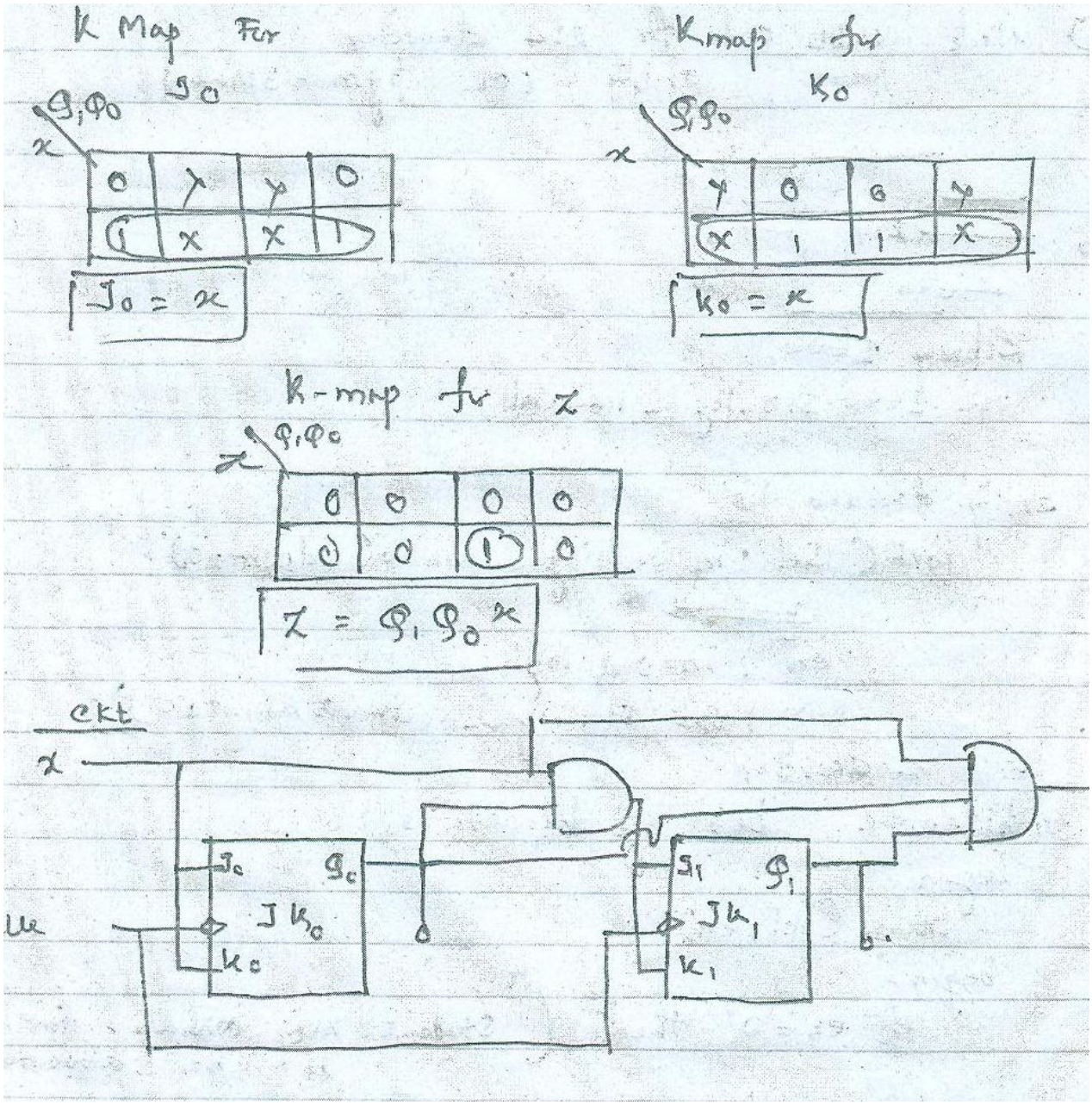
c) Design FSM using JK flipflop for Fig. No.1.

Ans :Design FSM using JK Flipflop(Correct ckt-1, State Table – 2, K-map – 1 M)

Consider ‘x’ as input & z as the output. The states A=00, B=01, C=10 & D=11 which gives State Table as follows

Present	**State**	Input	**Next State**	**Next State**	Excitation Inputs				**Output**
Q_1	**Q_0**	x	**$Q_{1\ n+1}$**	**$Q_{0\ n+1}$**	**J_1**	**K_1**	**J_O**	**K_O**	**z**
0	0	0	0	0	0	X	0	X	0
0	0	1	0	1	0	X	1	X	0
0	1	0	0	1	0	X	X	0	0
0	1	1	1	0	1	X	X	1	0
1	0	0	1	0	X	0	0	X	0
1	0	1	1	1	X	0	1	X	0
1	1	0	1	1	X	0	X	0	0
1	1	1	0	0	X	1	X	1	1s





d) Write a testbench in VHDL to verify functionality of NAND gate.

Ans :Testbench for Nand Gate (any program describing simulation inputs combination of DUT is also used

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY nand_tbIS
END nand_tb;
```

```
ARCHITECTURE behavioral OF nand_tbIS
-- Component declaration of the tested unit
COMPONENT nand
PORT(
A :IN STD_LOGIC;
B :IN STD_LOGIC;
Result :OUT STD_LOGIC );
END COMPONENT;
SIGNAL test_vector: STD_LOGIC_VECTOR(1 DOWNTO 0);
        SIGNAL test_result : STD_LOGIC;
BEGIN
Ut :nand
PORT MAP (
A =>test_vector(1),
B =>test_vector(0),
Result =>test_result);
);
Testing: PROCESS
BEGIN
test_vector<= "00";
WAIT FOR 10 ns;
test_vector<= "01";
WAIT FOR 10 ns;
test_vector<= "10";
WAIT FOR 10 ns;
test_vector<= "11";
WAIT FOR 10 ns;
test_vector<= "10";
WAIT FOR 10 ns;
test_vector<= "01";
WAIT FOR 10 ns;
test_vector<= "10";
WAIT FOR 10 ns;
test_vector<= "11";
```

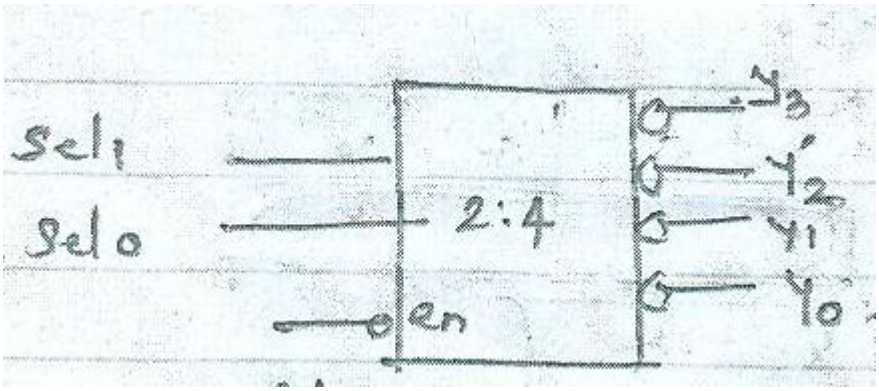
WAIT FOR 10 **ns**;

END PROCESS;

END behavioral;

e) Write VHDL code for 2:4 Decoder.

Ans :Entity (1 M, Pgm. 3 M)



Library IEEE;

Use IEEE . S +D _ logic _ 1164.all;

Entity decoder is

Part (Sel : in std-logic-vector (1 down to 0)

en : in std_logic;

y : out std-logic-vector (3 downto 0)

end decoder

architecturebeh of decoder is

begin

process (sel, en)

begin

if en = '0' then (student may consider active high i.e. en=1aslo correct).

Case sel is

When "00" => y α = "1110" (or "0001"active high)

When "01" => y α = "1101"

When "10" => y α = "1011"

When "11" => y α = "0111"

When others=> y α = "1111"

End case,

End if :

End process :

End beh.,

Student may use another logic & syntax

f) Draw and explain typical macrocell of CPLD.

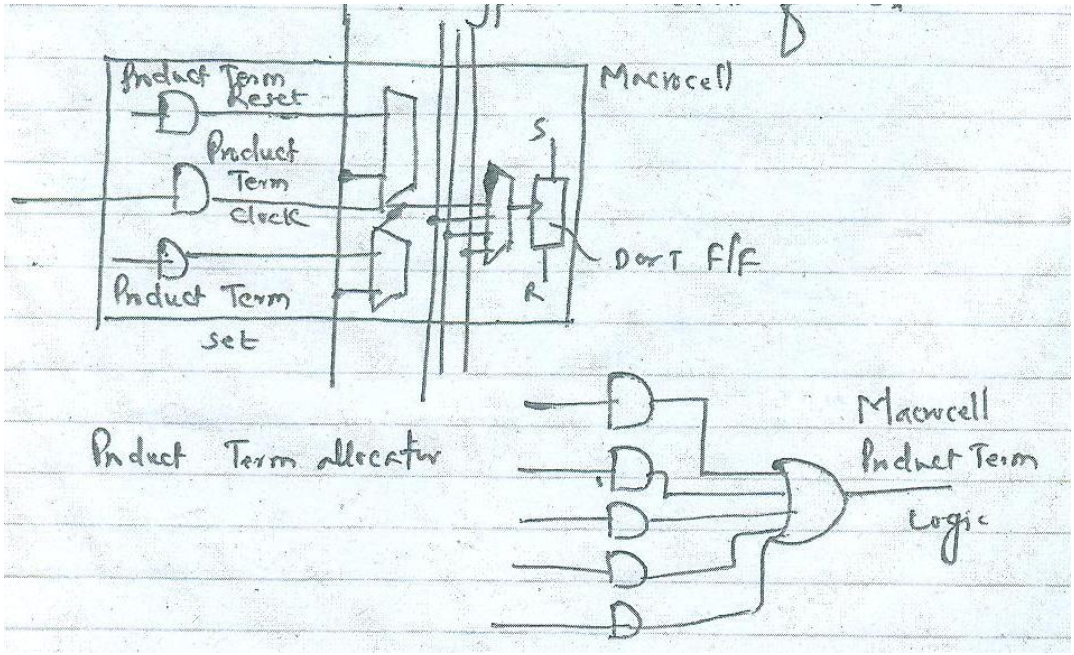
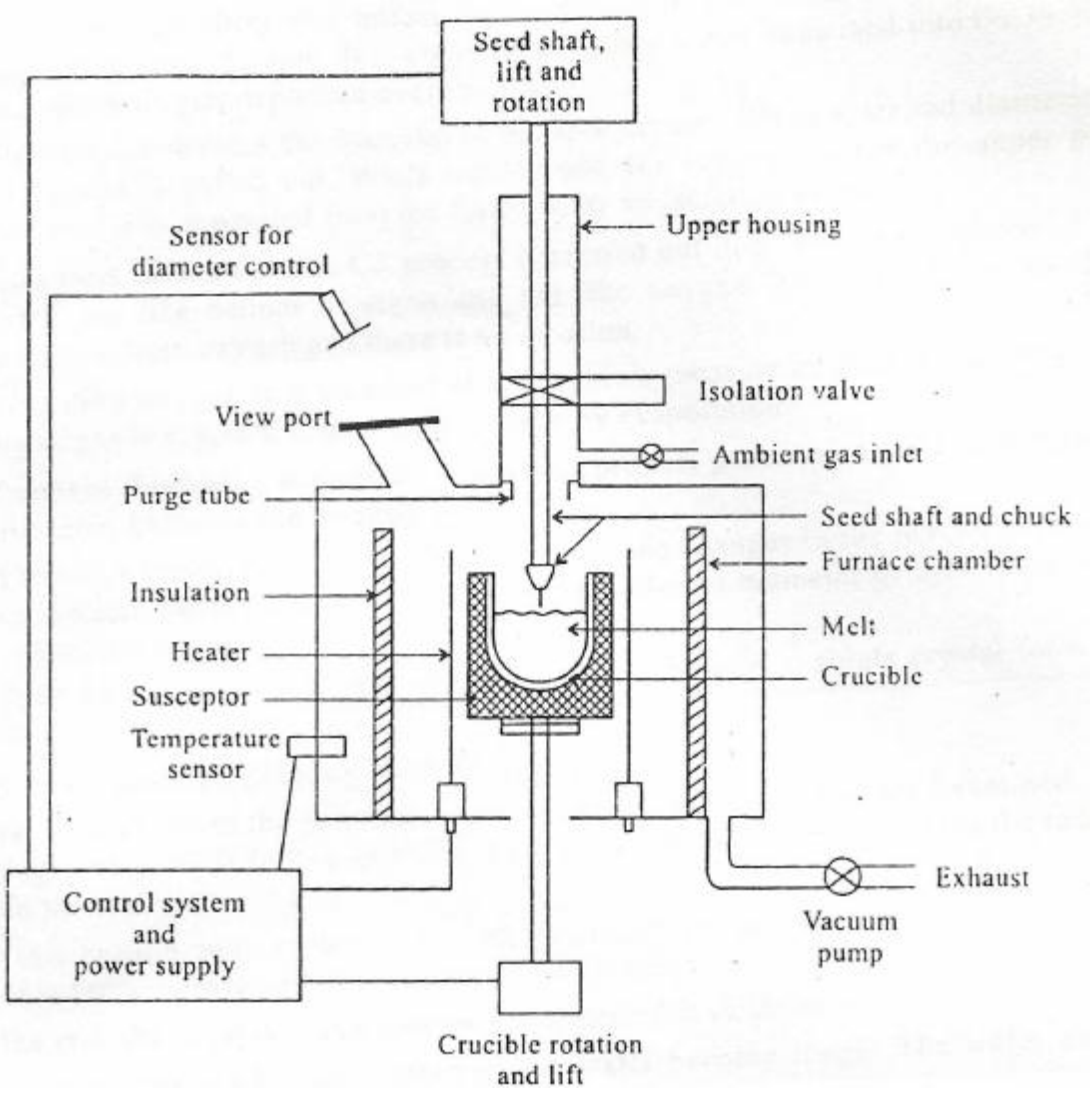


Figure shows macrocell which is individually configured for a combinatorial or registered function. The macrocell register can be configured as D-type or T-type flipflop or it may be bypassed for combinatorial operation. The register supports both asynchronms set & reset operations. Fire product terms are available for each microcell.

Q6. Attempt any FOUR of the following (16)

a) Explain wafer processing using C-Z method

Ans :



The basic raw material used in modern semiconductor plants is wafer or a disc of silicon. The size of wafer varies from 75mm to 230mm in diameter and is less than 1mm thick.Czocharlski or C-Z process is a process of solidification of atoms from liquid or gaseous phase or it is a process of change in polycrystalline silicon to single crystal silicon .

The above fig shows C_Z method for manufacturing silicon ingots. It contains:

1) Furnace
2) Crystal pulling system
3) Ambient control
4) control system

1) Furnace

   It holds the most important part called crucible. Crucible contains polycrystalline silicon melt inside it. Therefore the material of the crucible should be selected such that it does not chemically react with the silicon melt. Also the crucible should have very high melting point and thermal stability. Also crucible is held by a graphite susceptor for mechanical support.

   Graphite gives good thermal conditions. Also furnace has heating element with the insulators. Insulators are used to provide insulation between the heater and chamber wall.

2) Crystal pulling system

   In the crucible the silicon melt is heated. A seed is immersed into this melt. This is done with the great precision. There is a seed shaft with a rod. At the end of the rod, there is a holder which holds the seed. This rod with the holder is kept perpendicular to the melt. It is continuously rotated and immersed into the melt the silicon from the melt gets deposited over the seed. This rotation decides the diameter of the seed crystal. When the desired diameter is achieved the crystal is pulled out. While pulling out the crystal reaches upper portion of the chamber, it is separated from the furnace by an isolation valve.

3) Ambient control

   The C-Z process is carried out in an inert atmosphere i.e the chamber has the gas like helium or argon and not like oxygen. Therefore hot graphite is protected from oxygen and there is no errosion.

   Also the inert gas is maintained at atmospheric pressure or at reduced pressure.

   Helium or argon gas is supplied from a liquid sorce by evaporation.

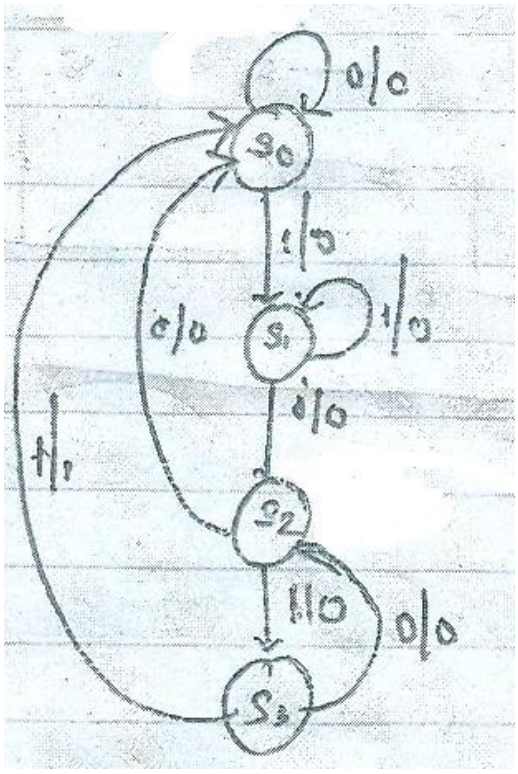
4) control system

   It provides control of process parameters such as temperature , crystal diameter , pull rate and rotation speed. There is a temperature sensor which detects the changes in temperature and its value is calibrated for the diameter. When a desired value of diameter of crystal is achieved the crystal is pulled out.

Thus the end product of C-Z process is silicon invert in single crystal form with desirable diameter.

b) Design 1011 sequence detector.

Ans :State Diagram

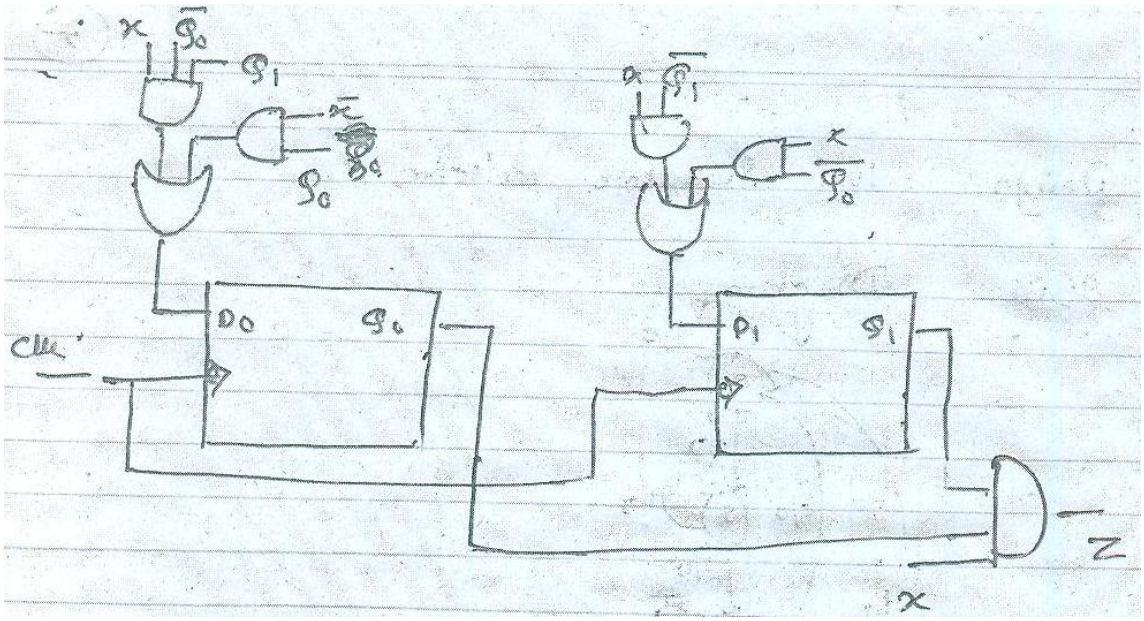


Consider x as input for which sequence is to be detected & Z is the output for detector.

The states are S_0=00, S_1=01, S_2=10, S_3=11

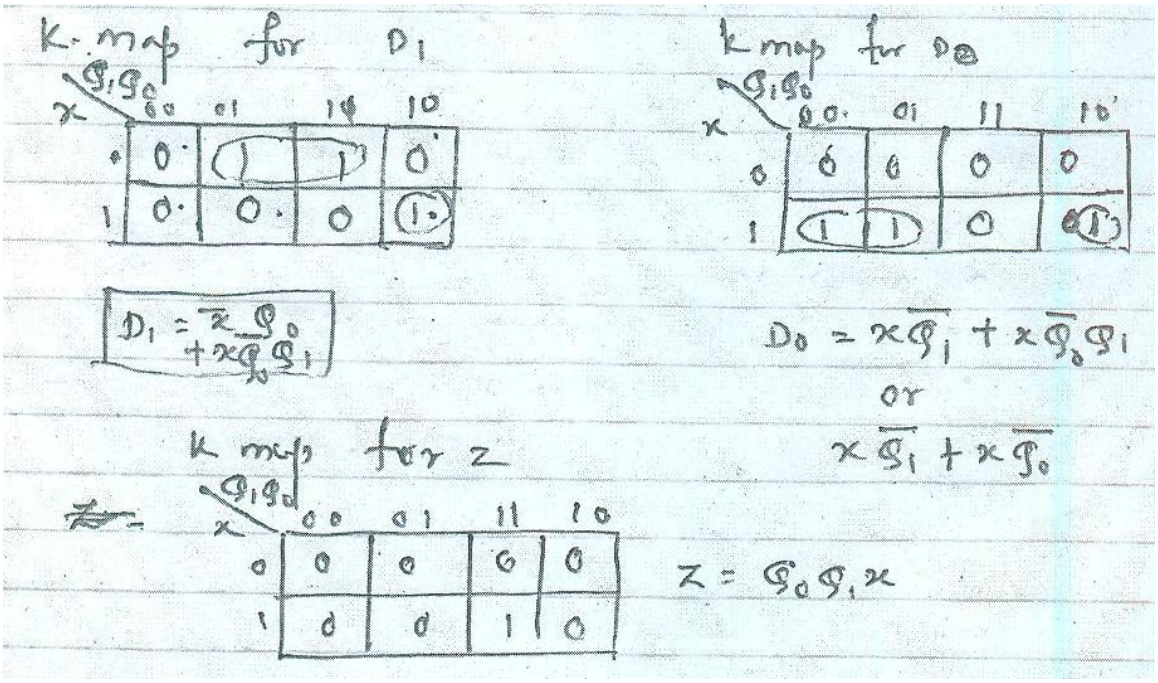
[To design the ckt JK or D or T-FF may be considered]

To obtain the solution D-FF is considered.



State Table

Q_{1n}	Q_{0n}	x	Q_{1n+1}	Q_{0n+1}	D_1	D_0	z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	0	0	0	0	0
1	0	1	1	1	1	1	0
1	1	0	1	0	1	0	0
1	1	1	0	0	0	0	1



c) Define threshold voltage. Give calculation of threshold voltage for enhancement type MOSFET.

Ans : Threshold Voltage – It is the voltage applied between the gate and source of MOS device, below which the drain to source current Ids drops to zero (channel is cut-off).

90% ICs are CMOS

Threshold Voltage Equation. It may be expressed as

$V_t = V_{tmos} + V_{fb}$

Where $V_{tmos}$is ideal threshold of an ideal MOS capacitor (No work for between gate &subgate).

V_{fb} = flat band voltage

The ideal threshold voltage may be expressed as

$$V_{tmos} = 2\phi_b + \frac{Q_b}{Cox}$$

Where ϕ_b = bulk potential

$$= \frac{KT}{q} \ln\left(\frac{N_A}{N_i}\right)$$

N_A = density of carriers in doped semiconductors substrate.

N_i = Carrier concentration, in intrinsic (undopped) silicon

K = Boltzman's constant = 1.380 x 10^{-23}J/° K

T = temp (°K)

Q = electronic charge 1.602 x 10^{-19}C

KT /q = 0.2586v at 300° K

Cox = Oxide capacitance $1/t_{ox}$

t_{ox} = gate oxide thickness.

Q_b = bulkcharge = $\sqrt{2\epsilon si\, qN_A 2\, \phi_b}$

Esi = permittivity of Silicon = 1.06 x 10^{-12}F/cm

V_{tmos} = +ve for n-transistor

= -ve for p-transistor

Therefore, V_{fb} = flat band voltage is given by

$$V_{fb} = \phi_{ms} - \frac{Q_{fc}}{Cox}$$

Q_{fc} = Fixed charge due to surface states that arises due to imperfections in silicon oxide interface &dopping.

ϕ_{ms} = work function difference between Gate material & silicon substrate

[ϕ gate $-\phi_{si}$]

For n^+ gate over p-substrate - n-transistor

Therefore, $\phi_{ms} = -\left(\frac{Eg}{2} + \phi_b\right) = -0.9V$ ($N_A = 1.x10^6$ cm^3)

Eg= band gap energy of silicon [1.16]

p –transistor

$\phi_{ms} = -\left(\frac{Eg}{2} - \phi_b\right) + 0.2$ V

Therefore, V_t may be varied by changing dopping concentration of substrate.

d) Write a testbench in VHDL for 4-bit up-down counter.

Ans :Testbench for 4 bit up down counter

(any program describing simulation inputs combination of DUT is also used

LIBRARY ieee;

USE ieee.std_logic_1164.all;

```
ENTITY counter_tbIS
END counter_tb;
ARCHITECTURE behavioral OF counter_tbIS
-- Component declaration of the tested unit
COMPONENT counter
PORT(
CLK ,rst: IN STD_LOGIC;
Mode :IN STD_LOGIC;
Result :OUT STD_LOGIC_vector(3 downto 0) );
END COMPONENT;
SIGNAL test_clk,test_rst,test_mode: STD_LOGIC;

        SIGNAL test_result :STD_LOGIC_vector(3 downto 0);
BEGIN
Ut : counter
PORT MAP (
CLK =>test_clk,
rst =>test_rst,
Mode =>test_mode,
Result =>test_result);
);
test_clk<= not test_clk after 100 ns
Testing: PROCESS
BEGIN

test_rst<= '0';
WAIT FOR 100 ns;
test_rst<= '1';
WAIT FOR 100 ns;
test_mode<= '1';
WAIT FOR 3200ns;
test_mode<= '0';
WAIT FOR 3200ns;
END PROCESS;
```

END behavioral;

e) Write VHDL code for JK Flipflop.

Ans :

```
Library IEEE;
Use  IEEE. STD _ Logic _ 1164.all :
Entity JKFF is
Port (J, K, CLK, rst : in Std-logic;
q  :  Out Std-logic);
end JKFF;
architectureBeh of JKFF is
begin
process (CLK, rst, J, K)
Variable state :Std_logic;
Begin
If rst = ‘0’then
State : = ‘0’
esif (CLK’ EVENT AND clk =  1  then
if ( J = ‘0’, K = ‘0’)  then
q < = state;
end if ;
if ( J = ‘0’and K = ‘0’)  then
q < = ‘0’ ;
end if :
if  (J = ‘1’and k = ‘0’) then
q < = ‘1’ ;
end if :
if( J = ‘1’, K = ‘1’)  then
qx = not state;
end if ;
endrfi;
end process,
endbeh.
```

OR

```
Library IEEE;
Use  IEEE. STD _ Logic _ 1164.all :
Entity JKFF is
Port (J, K, CLK  : in Std-logic;
Q  :inoutStd-logic;
Qn :inoutstd-logic)
end JKFF;
architectureBeh of JKFF is
begin
process (CLK, J, K)
begin


if (clk='1' and clk event   ) then
if ( J = '0'and K = '0')  then
Qα = Q;
Qnα = Qn;
elsif  (J = '0', K = '1')  then
Qα = '0'
Qnα = '1';
elsif  (J = '1', K = '0')  then
Qα = '1'
Qnα = '0';
elsif  (J = '1', K = '1')  then
Qα =  NOT Q :
Qnα = NOT Qni;
end if ;
end process;
endbeh.
```