MAHARASHTRASTATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

17659

MODEL ANSWER

WINTER– 18 EXAMINATION

Subject Title: Very Large Scale Integration Subject Code:17659

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.

<table>
<tr><th>Q. No.</th><th>Sub Q.N.</th><th>Answer</th><th>Marking Scheme</th></tr>
<tr><td><b>Q.1</b></td><td></td><td><b>Attempt any THREE :</b></td><td><b>12-Total Marks</b></td></tr>
<tr><td></td><td><b>1)</b></td><td><b>Compare Asynchronous sequential and synchronous sequential circuits.</b></td><td><b>4M</b></td></tr>
<tr><td></td><td><b>Ans:</b></td><td>
<table>
<tr><th>SR. NO.</th><th>ASYNCHRONOUS SEQUENTIAL CIRCUIT</th><th>SYNCHRONOUS SEQUENTIAL CIRCUITS.</th></tr>
<tr><td>1</td><td>Output can be changed at any instant of time by changing the input</td><td>Output changes at discrete interval of time</td></tr>
<tr><td>2</td><td>The status of memory element will change any time as soon as input is changed. It does not use a clock</td><td>The status of memory is affected only at the active edge of clock, if input is changed. It uses a clock pulse.</td></tr>
<tr><td>3</td><td>These circuits are easy to design</td><td>These circuits are difficult to design.</td></tr>
<tr><td>4</td><td>They are slower as compare to synchronous.</td><td>They are faster as compare to asynchronous.</td></tr>
<tr><td>5</td><td>Asynchronous is wherein all the flip-flops within the counter do not change state simultaneously. This is because all the flip-flops are not clocked simultaneously.</td><td>Synchronous is wherein all the flip-flops within the counter change state simultaneously. This is because all the flip-flops are clocked simultaneously.</td></tr>
</table>
</td><td><b>Each point 1 mark</b></td></tr>
<tr><td></td><td><b>2)</b></td><td><b>Explain estimation of channel capacitance of CMOS.</b></td><td><b>4M</b></td></tr>
<tr><td></td><td><b>Ans:</b></td><td><b><u>Capacitance estimation:</u></b> The dynamic response i.e. switching speed of MOS system depends on capacitance associated with the MOS devices which are formed by different layers in MOS transistors and interconnection capacitances that are formed by metal, poly</td><td><b>Any 2 capacita</b></td></tr>
</table>

and diffusion wires.

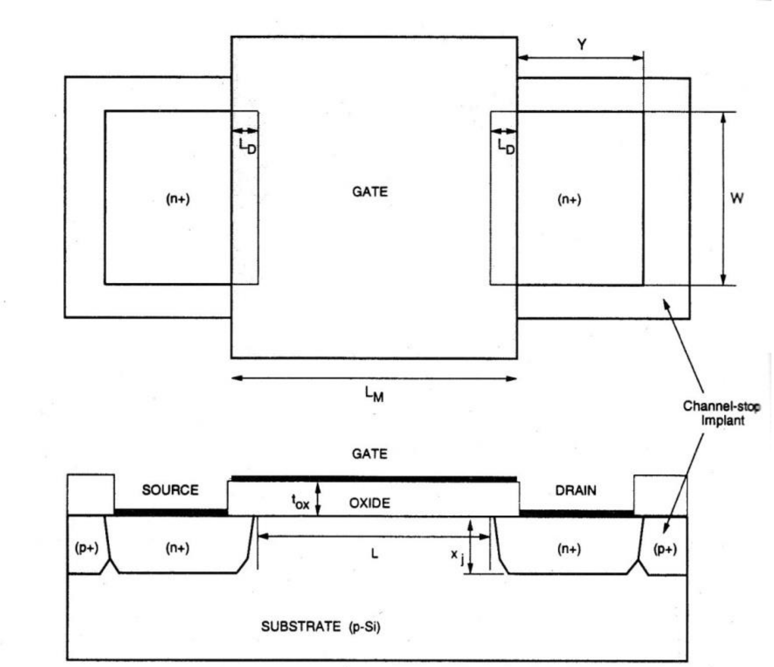
nces

Each 2 marks

The total load capacitance on the output of a CMOS gate is the sum of:

- Gate capacitance of other inputs connected to the output of the gate.
- Diffusion capacitance of the drain connected to the output.
- Routing capacitance i.e. capacitance of interconnects between the output and other inputs.

<u>MOSFET Capacitance:</u> The cross section view and top view of n channel MOSFET is as shown:

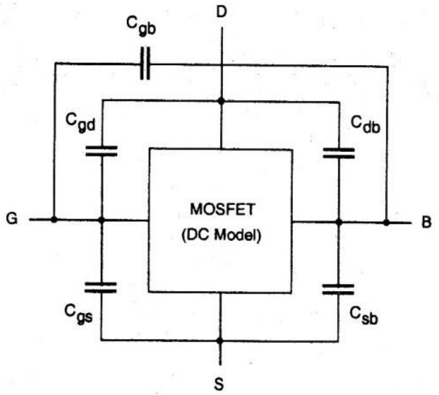


The mask length of the gate is indicated by L_M and the actual channel length is L. The extent of both the gate-source and the gate-drain overlaps are L_D.

Hence Channel length $L = L_M – 2L_D$.

The source and drain overlap region lengths are usually equal to each other because of the symmetry of the MOSFET structure. Generally L_D is of order of 0.1μm. Both the source and drain diffusion regions have a width of W. The diffusion region length is denoted by y. These regions surrounded by p+ channel stops implants as it avoids formation of unwanted channels between the two neighboring n+ diffusions. Most of the parasitic device capacitances are due to three dimensional, distributed charge voltage relations within the device structure.

The parasitic device capacitances are of two types:



1. Oxide related capacitances: The two capacitances that occur as a result of overlapping regions of source and drain are called $C_{GD\ (overlap)}$ and $C_{GS\ (overlap)}$ respectively:

   $C_{GD\ (overlap)} = C_{OX}.\ W.\ L_D$

   $C_{GS\ (overlap)} = C_{OX}.\ W.\ L_D$

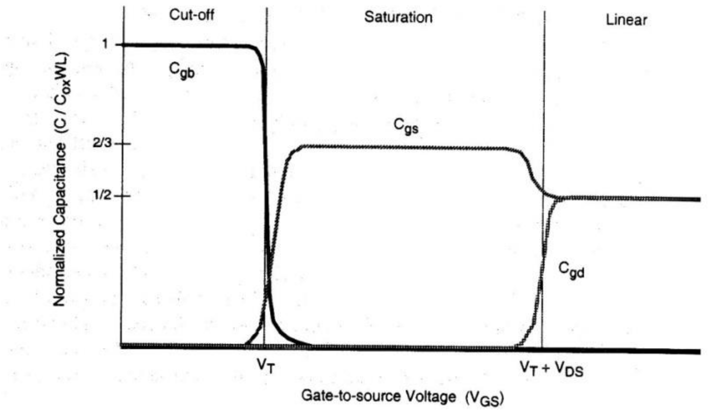
   With $C_{OX} = \frac{\epsilon_{ox}}{t_{ox}}$

   Both of these capacitances do not depend upon the bias condition that is they are voltage independent. Consider the capacitance that result from the interaction between the gate voltage and the channel charge. As the channel region is connected to the source, drain and substrate there are three capacitances between the gate and these regions: C_{GS}, C_{GD}, and C_{GB} respectively.

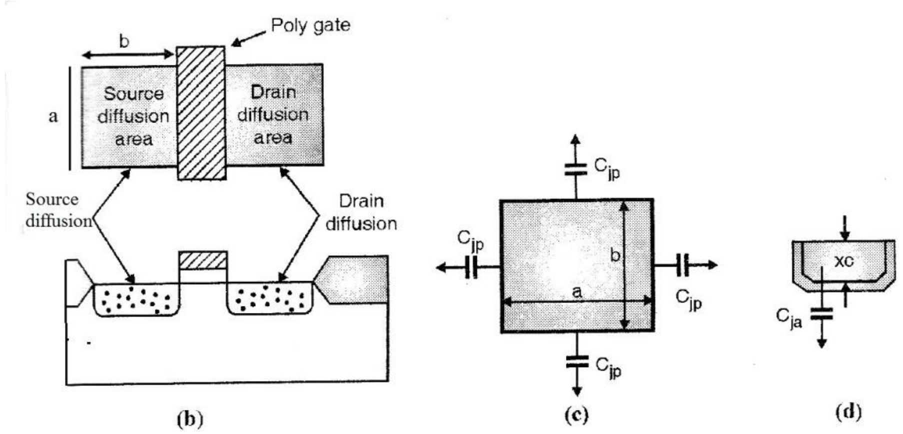
   The gate of channel capacitance is distributed and voltage dependent. The gate to source capacitance is actually the gate to channel capacitance between the gate and the source terminals, the gate to drain capacitance is the gate to channel capacitance between the gate and drain terminal. As these capacitances are voltage dependent these capacitances are determined in different biasing conditions during cut-off, linear and saturation modes:

   - In cut off mode surface is not inverted. There is no conducting channel that links the surface to the source and drain. Hence the gate to source and gate to drain capacitances: $C_{gs} = C_{gd} = 0$. Hence the gate to substrate capacitance can be approximated as: $C_{gb} = C_{ox}.W.L$
   - In linear mode of operation, the inverted channel extends across MOSFET, between the source and drain as shown. This conducting inversion layer on the surface effectively shields the substrate from the gate electric field, hence $C_{gb} = 0$. In this case the distributed gate to channel capacitance is shared between source and drain and is given as: $C_{gs} = C_{gd} = \frac{1}{2}.C_{OX}.W.L$
   - In saturation mode, the inversion layer on the surface does not extend to the drain, but is pinched off. The gate to drain capacitance is hence equal to zero. As the source is still linked to the conducting channel, its shielding effect also forces the gate to substrate capacitance to zero. Hence the distributed gate to channel capacitance as seen between gate and source is given as: $C_{gs} = \frac{2}{3}.C_{OX}.W.L$

To calculate total capacitance all the gate oxide capacitances are summed up.

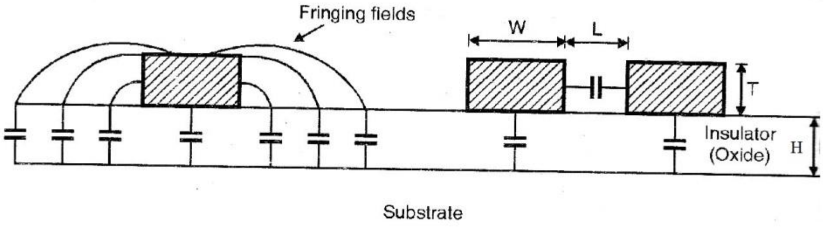


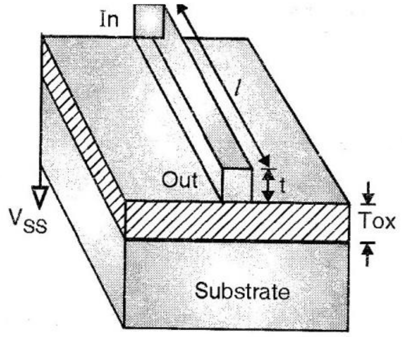
2. Junction Capacitance: Voltage dependent source substrate and drain substrate junction capacitances C_{sb} and C_{db} are formed due to the depletion charge surrounding the respective source to drain diffusion regions embedded in the substrate.



Both these junctions are reverse biased under normal operating conditions. These capacitances depend upon the voltage between the diffusion regions and substrate as well as on the effective area. The diffusion capacitance Cd is proportional to the total diffusion to substrate junctions area. There is a formation of base area and also of the area of the side wall periphery. The side wall capacitances can be characterized by a periphery capacitance per unit length.

3. Routing Capacitance: As a metal layer is deposited on the insulating layer in the final stage of transistor fabrication capacitances are formed between the metal or poly layers and substrate. These capacitances are approximated using a parallel plate model: $C = \frac{\varepsilon A}{t}$. The parallel plate approximation however ignores fringing fields that occurs at the edges of conductor due to its finite thickness. If two conductors are running in parallel a conductor can exhibit capacitance to an adjacent conductor on the same layer as shown:





To determine the routing capacitance the model is considered which is a simple isolated interconnect line:

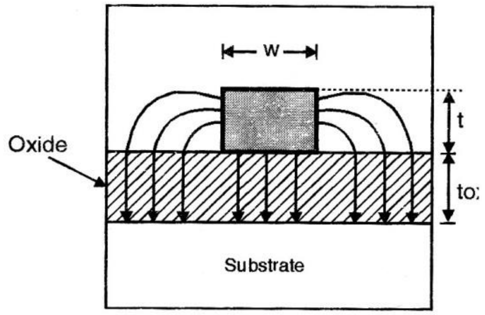
The interconnect line has a length l and width W and thickness t. The line resistance can estimated as: $R_{line} = R_s(l/w)$ ohms.

There is increase in parasitic resistance with length l.

The total line capacitance can be estimated by using formula: $C_{line} = \dfrac{\varepsilon_{ox} lw}{T_{ox}}$

T_{ox} is the thickness of the insulating oxide between the line and substrate.

Cline is also called the self capacitance of line. But this equation ignores the fringing electric field from the edges and sides when the line is at a positive voltage.



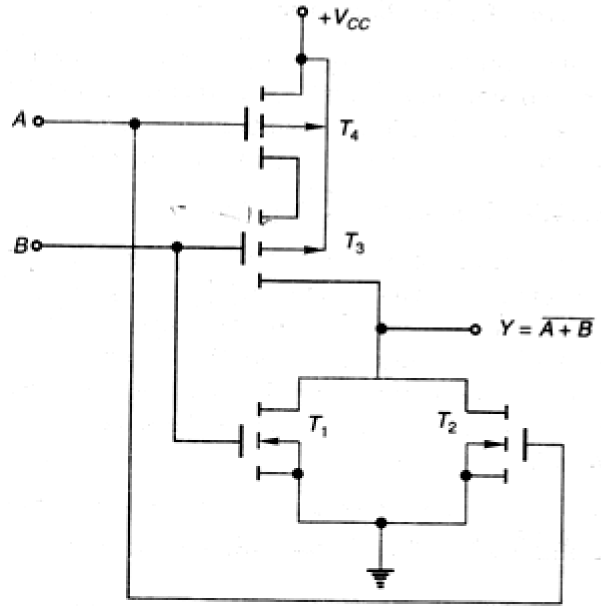
Hence capacitance per unit length C that accounts for this effect is:

$$C = \varepsilon_{ox}\left[1.15\left(\frac{W}{T_{ox}}\right) + 2.8\left(\frac{t}{T_{ox}}\right)^{0.222}\right] \text{ F/cm.}$$

The first term accounts for fringing from the bottom side of the line, while the second terms depends on the thickness t and is due to the side wall effects.

3) **Draw CMOS two input NOR gate and write it's truth table.** **4M**

Ans:



NOR Gate

Truth table:

A	**B**	**T1**	**T2**	**T3**	**T4**	**Y**
0	0	OFF	OFF	ON	ON	Vdd
0	Vdd	ON	OFF	OFF	ON	0
Vdd	0	OFF	ON	ON	OFF	0
Vdd	Vdd	ON	ON	OFF	OFF	0

Diagram = 3 marks

Truth table= 1 mark

4) **State any 4 features of VHDL.** **4M**

Ans:

1. It is a concurrent language that is it can execute statements at same time in parallel as in hardware.
2. It is a sequential language that is it can execute sequential statements one at a time in sequence.
3. It supports synchronous and asynchronous timing models.
4. Facilitates device independent of design portability.
5. It supports design libraries.
6. It has well defined interface.
7. Behaviour specification for simulation purpose.
8. Test Benches can also be generated.
9. Digital modelling techniques supported.
10. It is not technology specific.
11. VHDL has powerful construct language, constructs such as else if, with select, case, when etc.
12. VHDL supports flexible design methodologies top-down, bottom-up or mixed.
13. Strongly typed language:
14. Dealing with signed and unsigned numbers is natural, and there's less chance of making a precision mistake or assigning a 16-bit signal to a 4-bit signal.

15. Ability to define custom types:

Any 4 features 1 mark each

16. Record types: Define multiple signals into one type.
17. Natural coding style for asynchronous resets.
18. Logical statement (like case and if/then) endings are clearly marked.

B)	**Attempt any ONE :**	**6**
1)	**Compare FPGA and CPLD (any six pts).**	**6M**

Ans:

Sr no:	**FPGA**	**CPLD**
1.	It is field programmable gate array.	It is complex programmable logic device.
2.	Capacity is defined in terms of number of gates available.	Capacity is defined in terms of number of macro cells available.
3.	FPGA consumes less power than CPLD.	CPLD consumes more power than FPGA
4.	Number of input and output pins on FPGA are less than CPLD.	Number of input and output pins on CPLD are more than FPGA.
5.	FPGA is suitable for designs with large number of blocks with few number of inputs.	CPLD is ideal for complex blocks with large number of inputs.
6.	FPGA based designs require more board space and layout is more complex.	CPLD board designs need less board space and layout is less complex.
7.	It is difficult to predict the speed performance of design.	Speed performance can be easily predicted.
8.	FPGA are available in wide density range.	CPLD contain fewer registers but have better performance.

Any 6 points = 1 mark each

2)	**State any one process for wafer fabrication with diagram**	**6M**

Ans:

Wafer Processing:

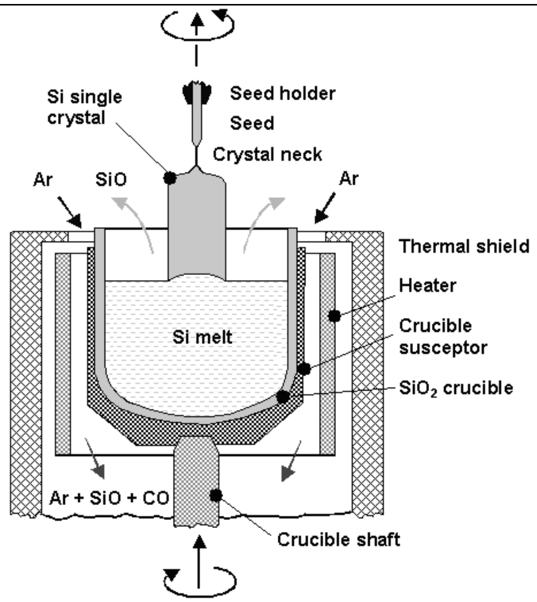
The basis raw material used is a disk of silicon, which is between 75 mm to 150 mm in diameter and is less than 1mm thick used in semi-conductor plants.

Wafers are cut from ignots of silicon crystal that have been pulled from a crucible melt of pure molten polysilicon silicon. This method is known as Czochralski [C-Z] method.

{An ingot is a piece of material, usually metal, that is cast into a shape suitable for further processing}

Diagram = 2 marks

Process = 4marks



For getting crystals with required electrical properties controlled amounts of impurities are added to the melt. To initiate single-crystal growth, the crystal orientation is determined by seed crystal is dipped into the melt and slowly pulled out. As the seed crystal is pulled out of the melt, it brings with it a solidified mass of silicon with the same crystalline structure as that of the seed.

During the crystal pulling process, the seed crystal and the crucible are rotated in opposite direction in order to produce ingots of circular cross- section. The diameter of the ingot is controlled by pulling rate and melt temperature. Ingot diameter is about 10 to 15 cm and length of order 100 cm. The ignot is also ground flat slightly to get reference plane. The ignot is then sliced using a stainless steel blade with industrial diamonds embedded into inner diameter cutting edge. This produces circular wafers of slices.

The silicon wafers obtained have very rough surface due to slicing operation. These wafers undergo a number of polishing steps to produce a flat surface.

Then one side of wafer is given a fine mirror smooth highly polished finish, where as the other side is simply lapped on an abrasive lapping machine to obtain acceptable degree of flatness. Finally the wafers are thoroughly rinsed and dried. A raw of thickness about 0.6 to 1mm produces wafer of about 0.15 to 0.8 mm thickness after all polishing steps.

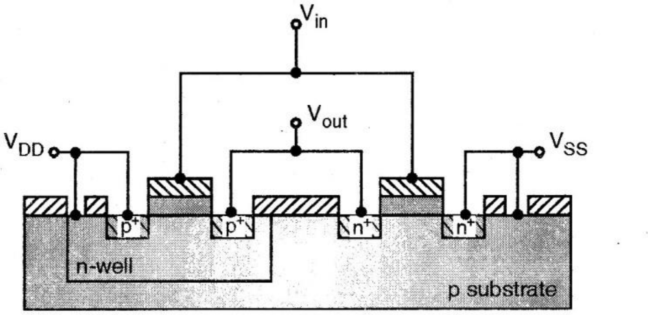
OR

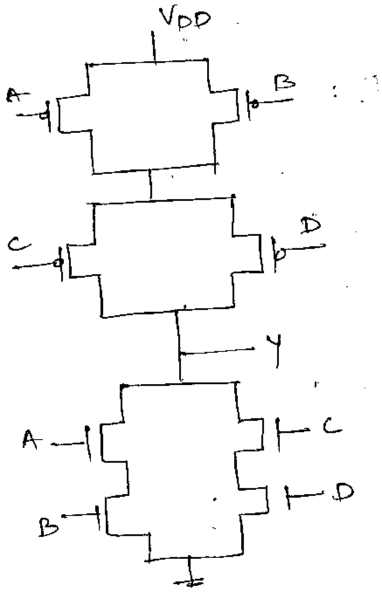
It consists of Quartz crucible, which is surrounded by a graphite radiator. The graphite is heated by radio frequency induction heating and temperature maintained a few degrees above the melting point of silicon (approx. 1425^{0}C), the atmosphere just above the polysilicon melt is typically helium or orgon for freezing.

A polycrystalline Si is melted in the crucible and controlled amount of impurities (p type or n type) are added to the melt to provide the crystal with required electrical properties.

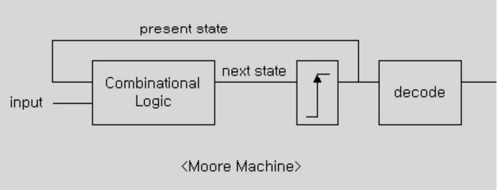
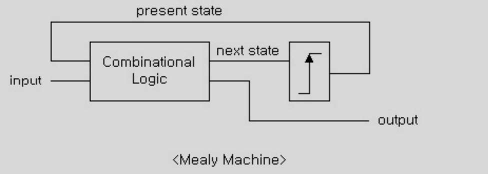
After the seed (single crystal silicon piece) is dipped into the melt, the seed is gradually withdrawn vertically from the melt while simultaneously being rotated. The molten polycrystalline silicon melts the tip of the seed and it is withdrawn, refreezing occurs. As the melt freezes, it assumes the single crystal form of the seed. This process is continued until the melt is consumed. The diameter of the ingot (rod of silicon) is determined by the seed withdrawn rate and seed rotation rate.

The produced crystalline silicon rod is then slicing into wafers using cutting tools like

		diamond blades. Following slicing at least one face of the wafer is polished to flat scratch free mirror finish surface.	
Q 2		**Attempt any FOUR :**	**16-Total Marks**
	1)	**Explain fabrication using N-well process.**	**4M**
	Ans:	**N-Well process:** The N-well CMOS circuits are getting more popular because of the lower substrate bias effect on transistor threshold voltage and lower parasitic capacitances associated with source and drain regions.  V_{in} V_{out} V_{DD} V_{SS} p^+ p^+ n^+ n^+ n-well p substrate The fabrication steps are as follows: • Thick SiO_2 layer is grown on p-type silicon wafer. • After defining the area for N-well diffusion, using a mask, the SiO_2 layer is etched off and n-well diffusion process is carried out. • Oxide in the n transistor region is removed and thin oxide layer is grown all over the surface to insulate gate and substrate. • The polysilicon is deposited and patterned on thin oxide regions using a mask to form gate of both the transistors. The thin oxide on source and drain regions of both the transistors is removed by proper masking steps. • Using n^+ mask and complementary n^+ mask, source and drain of both nMOS and pMOS transistors are formed one after another using respective diffusion processes. These same masks also include the V_{DD} and V_{SS} contacts. • The contacts are made using proper masking procedure and metal is deposited and patterned on the entire chip surface. • An overall passivation layer is formed and the openings for accessing bonding pads are defined.	**Diagram = 2 marks** **Process = 2 marks**
	2)	**Design Y=AB.CD using CMOS logic.**	**4M**

	Ans:	$Y = \overline{AB} \cdot \overline{CD}$ $\therefore \overline{Y} = \overline{\overline{AB} \cdot \overline{CD}}$ $= \overline{\overline{AB}} + \overline{\overline{CD}}$ $\overline{Y} = AB + CD$ $\therefore Y = \overline{\overline{Y}} = \overline{AB + CD}$  	
	3)	**What do you mean by simulation? Why it is necessary?**	**4M**
	Ans:	Simulation is functional emulation of a circuit design through software programs that use models to replicate how a device will perform in terms of timing and results. It should be performed at all stages of circuit design. Need of simulation: 1. VHDL simulation serves as a basis for testing complex designs and validating the design prior to fabrication. 2. It allows the observation of the circuits behavior at the inputs and outputs and all internal rules. 3. The simulation program processes as representation of input stimuli and determines the behavior of signal with respect to time stops. 4. The simulation is done for the verification of the behavior of the circuit for both time behavior and fundamental behavior. 5. Simulation operation can be used to verify independent delays in the circuits. 6. Simulation is used for design verification: Validate assumptions, Verify logic, Verify performance (timing)	**Definition = 1 mark** **Need = 3 marks.**
	4)	**Compare Mealy M/C with Moore M/C.**	**4M**

Ans:

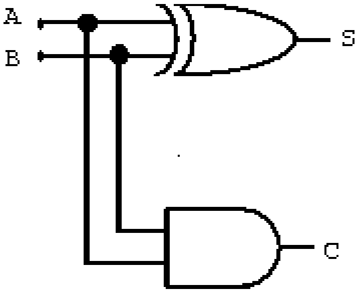
	MOORE MACHINE	MELAY MACHINE
1	Output is function of state of machine.	Output is function of state of machine and present input condition.
2	Requires more number of states.	Requires less number of states.
3	Faster.	Slower.
4	Simple design.	Complex design.
5	Output in state.	Output is at the time of state transition.
6	**Block diagram:** 	**Block diagram:** 

Any 4 points = 1mark each

5) Write VHDL code for half adder. **4M**

Ans:

Half adder which is having one XOR gate and a AND gate.



```
Library IEEE;
use
IEEE.STD_LOGIC_1164.all;

entity ha_en is
port (A,B:inbit;S,C:outbit);
end ha_en;

architecture ha_ar of ha_en is
begin
        S<=A xor B;
        C<=A and B;

end ha_ar;
```

Code=4 marks

Note – Any relevant Code Marks to be given.

Q.3 **Attempt any FOUR :** **16-Total Marks**

a) Write the syntax of entity and architecture used in VHDL and explain it. **4M**

	Ans:	**Entity declaration:** It defines the names. Input output signals and modes of a hardware module. Also it provides the external interface of an entity. It is a black box view. **Syntax**: entity entity _ name is Port declaration. end entity_name **Architecture**: It describes the internal description of it tells what is thereinside design. Each entity has at least one architecture and an entity can have many architecture. Architecture can be described using structural, dataflow, behavioral or mixed style. Architecture can be used to described a design at different levels of abstraction like gate level (RTL) or behavior level. **Syntax**: architecture architecture _name of entity_ name Architecture_ declaration_ name; begin Statement; end architecture_ name;	**1M Explanation 1M Syntax Each**
	b)	**Draw 2:4 decoder and write VHDL code for it.**	**4M**

	Ans:	***(Note – Any relevant code with different statements marks to be given. Decoder with active high is also consider)***
		)
		
		library IEEE;
		use IEEE.STD_LOGIC_1164.all;
		entity decoder is
		port(
		a : in STD_LOGIC_VECTOR(1 downto 0)
		b : out STD_LOGIC_VECTOR(3 downto 0));
		end decoder;
		architecture bhv of decoder is
		begin
		process(a)
		begin
		case a is
		when "00" => b <= "0001"; *"1110" – If written Active low bits marks to be given*
		when "01" => b <= "0010"; *"1101"*
		when "10" => b <= "0100"; *"1011"*
		when "11" => b <= "1000"; *"0111"*
		end case;
		end process;
		end bhv;
			**2M
			Entity
			2M
			Architeture**

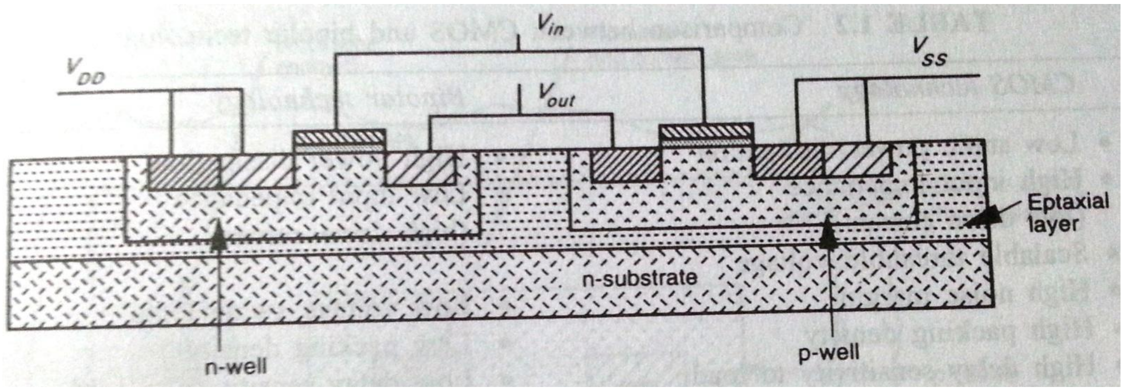
c) **Describe Twin-tube process with diagram.** **4M**

Ans:

Twin Tub Process:

- A logical extension of the p-well and n-well approaches is the twin-tub fabrication process.
- In this process, a substrate of high resistivity of n-type material is used and then in this n-type material both n-well and p-well regions are created.
- By using this process, it is possible to preserve the performance of n-transistors without compromising the p-transistors.
- The doping control is more rapidly achieved and some relaxation is manufacturing tolerances results.
- This is particularly important as far as latch up is concerned.

2M Explanation



2M Diagram

Note : step wise process is also suitably considered

d) **What do you meant by sensitivity list and zero modeling?** **4M**

Ans:

Sensitivity list:

Every concurrent statement has a sensitivity list. Statements are executed only when there is an event or signal in the sensitivity list, otherwise they are suspended.

Ex. F<=a and b;

A and b are in the sensitivity list of f. the statement will execute only if one of these will change.

Ex. Proc
ess(clk, RST)

The process is sensitive to RST and clk signal i.e. an event on any of these signals will cause the process toresume

Zero Modeling:

All digital circuit elements have a delay (propagation delay) which is very small in terms of nano sec. This nano sec delta delay will have little impact while writing the VHDL code. But for circuit realization this delay must be incorporated. The physical circuit always has finite delay.The ordering of zero delay events is handled with a fictitious unit called delta time. Delta time represents the executionof a simulation cycle without advancing Simulation time. The simulator models zero-delay events using delta time.Events scheduled at the same time are simulated in specific order during a delta time step.Related logic is then re-simulated to propagate the effects for another delta timestep. Delta time steps continue until there is no activity for
the same instant of simulated time

2M Each

e) Compare signals and variables in VHDL **4M**

Ans:

Sr. No	**Signals**	**Variables**
1	Signal objects are used to connect entities together to form models	Variables are used for local storage in process statements and subprograms.
2	Signals have their values scheduled in the future	Variables have all assignments to variables occur immediately
3	The keyword signal is followed by one or more signal names	The keyword variable is followed by one or more variable names
4	Signals can be declared in entity declaration sections architecture declarations and package declarations	Variables can be declared in the process declaration and subprogram declaration sections only.
5	Signals need more information so more memory	Variables take less memory

Any 4 points 1M each

Q.4 A) Attempt any THREE : **12-Total Marks**

1) Define the following terms :

1) **Noise margin**
2) **Power fanout**
3) **Skew**
4) **Meta stability**

4M

Ans:

1) **Noise Margin:**

It is a measure of noise immunity of a gate or circuit (noise immunity is the ability of a gate or circuit to tolerate any noise present in a signal without performing a wrong operation)

2) **Power fanout :**

It is the maximum number of load gates that can be connected at output without loading with same IC family and by maintaining its output within the specified limit.

OR

The ability to drive the similar number of gates.

1M each

3) **Skew:**

The clock signal, which is said to be applied simultaneously to all the flip-flops, may cause a minute delay changes due to some variation in the wiring between the components. Due to this, it may happen that the clock signal may arrive at the clock inputs of different flip-flops at different times. This delay is termed as skew.

OR

The difference in the clock arrival time is call clock skew.

4) **Meta stability:**

Metastability in electronics is the ability of a non-equilibrium
electronic state to persist for a long (and theoretically unboundable) period of time.

OR

A metastable state is half way between logic "0" and logic "1" .It is undefined state.

2) **Explainevent scheduling with suitable example.** **4M**

Ans:

Event scheduling:

The assignment to signal x does not happen instantly. Each of the values assigned to x contain an afterclause.
The mechanism for delaying the new value is called scheduling an event. By assigning port x a new value, anevent was scheduled 0.5ns in the future that contains the new value forsignal x. when the event matures, signal receives a new value.
Event is nothing but change on target signal which is to be updated.

Example: X<= a after 0.5ns when select=0 else
X<= b after 0.5ns

2M Explanation

2M example

3) **What is test bench? Write its applications** **4M**

Ans:

Test Bench:

A test bench is used to verify the functionality of the design.
We need to stimulate our designs in order to test their functionality. Stimulus in a real system is from an external source, not from our design. We need a method to test our designs that is not part of the design itself. This is called a "Test Bench". Test Benches are VHDL entity/architectures. We initiate the design to be tested using components. We call these instantiations "Unit Under Test" (UUT) or "Device Under Test". The entity has no ports .We create a stimulus generator within the architecture. We can use reporting features to monitor the expected outputs.

Applications:
1. A test bench is used to verify the functionality or correctness of the design.
2. It is useful to generate stimulus for stimulation.

2M Explanation

Applications 2M

3. It is used to analyze the design to compare the result of two simulations.
4. To compare the results of two simulations.
5. To apply this stimulus to the entity under test and to collect output responses.
6. To compare output responses with expected values.

4) List different concurrent statements and give the example of any two. **4M**

Ans:

Concurrent Statements in VHDL

- With –select statement
- When –else statement
- Generate statement
- Block statement

Note – Any Relevant Code using concurrent statements marks to be given, Only for the syntax with respect to example also marks to be given only architectural part is expected in example.

EXAMPLE

1M List

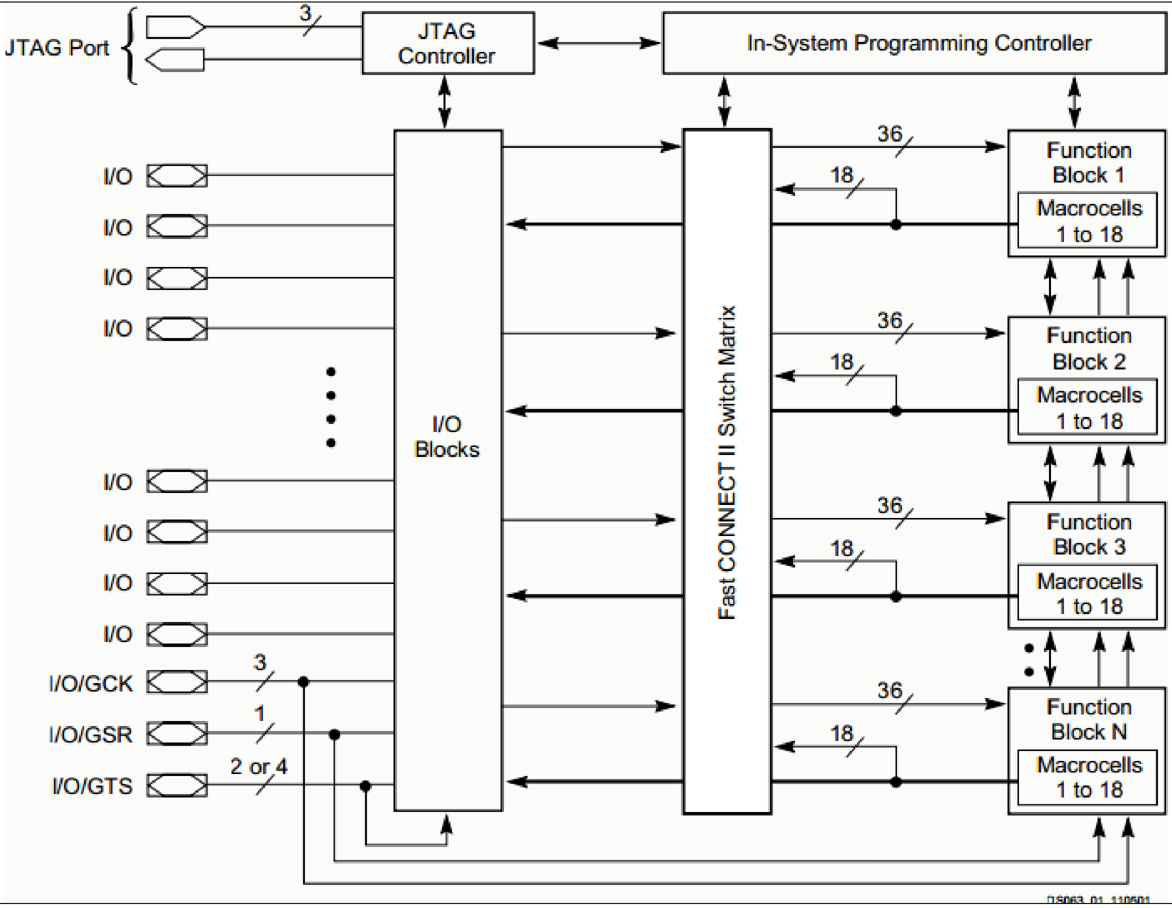
4:1 multiplexer using with-select statement.

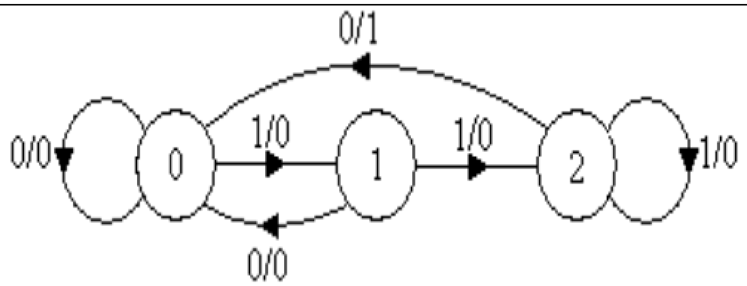
```
Library IEEE;
Use IEEE. Std_logic_1164.all;
Entity MUX4_1 is
          Port(I : in bit_vector (3 downto 0);
                S: in bit_vector (1 downto 0);
                Y: out bit);
          end MUX4_1;
architechture MUX of MUX4_1 is
begin
          with S Select
          Y <=   I(0) when “00”
                  I(1) when “01”
                  I(2) when “10”
                  I(3) when “11”;
                  ‘0’ when others; -- optional
end MUX;
```

1 1/2 M for each Example

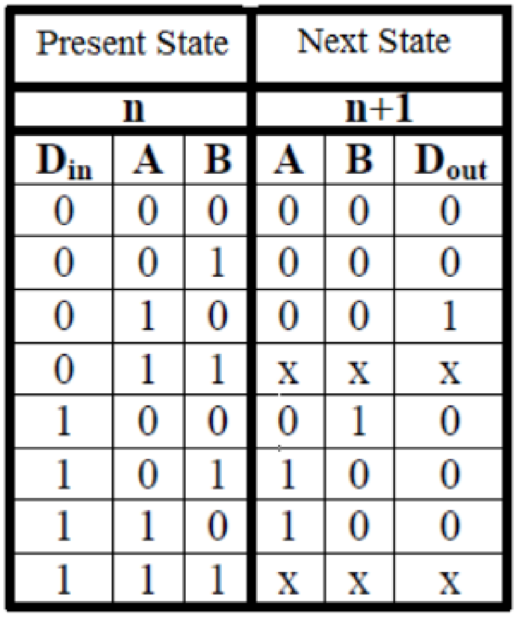
Design 4:1 MUX using when-else statements.

```
Library IEEE;
Use IEEE. Std_logic_1164.all;
Entity MUX4_1 is
          Port (I0, I1, I2, I3, S0, S1: in bit;
                  Y: out bit);
end MUX4_1
architechture MUX of MUX4_1 is
begin
```

		Y <= I0 when S0 = ‘0’ and S1 = ‘0’ else I1 when S0 = ‘0’ and S1 = ‘1’ else I2 when S0 = ‘1’ and S1 = ‘0’ else I3 when S0 = ‘1’ and S1 = ‘1’ else ‘0’ end MUX;	
	B)	**Attempt any ONE :**	**6**
	1)	**Draw architecture of XC9500 CPLD.**	**6M**
	Ans:	 JTAG Port 3 JTAG Controller In-System Programming Controller I/O I/O I/O I/O I/O Blocks I/O I/O I/O I/O I/O/GCK 3 I/O/GSR 1 I/O/GTS 2 or 4 Fast CONNECT II Switch Matrix 36 18 Function Block 1 Macrocells 1 to 18 36 18 Function Block 2 Macrocells 1 to 18 36 18 Function Block 3 Macrocells 1 to 18 36 18 Function Block N Macrocells 1 to 18 DS063_01_110501	**Diagram – 6M**
	2)	**Design a sequence detector to detect the sequence 110. Use D F/F to design the circuit.**	**6M**
	Ans:	Solution: The state transition diagram for the sequence 110 is as follows	**2M State Diagram** **1M State Table** **1M**



The state transition table for the above transition diagram is



Present State			Next State		
n			n+1		
D_{in}	A	B	A	B	D_{out}
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	x	x	x
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	x	x	x

The K – maps for the above table to determine the equations are as follows

Kmaps

2M Circuit Realization

AB

	A	00	01	11	10
D_{in}	0	0	0	x	0
	1	0	1	x	1

$$D_A = D_{in}(A + B)$$

AB

	B	00	01	11	10
D_{in}	0	0	0	x	0
	1	1	0	x	0

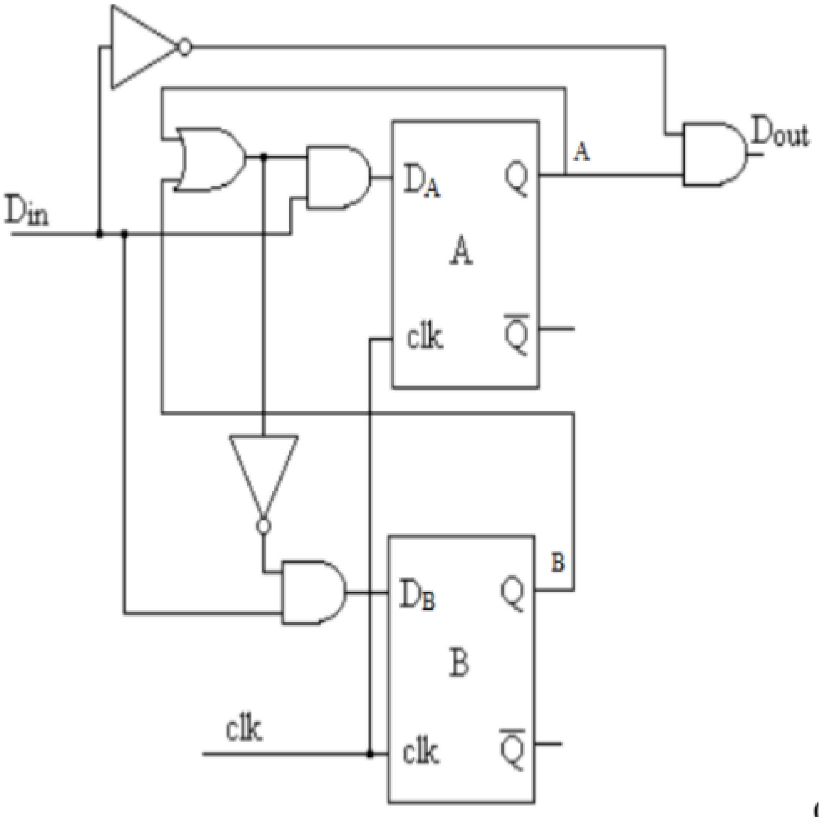
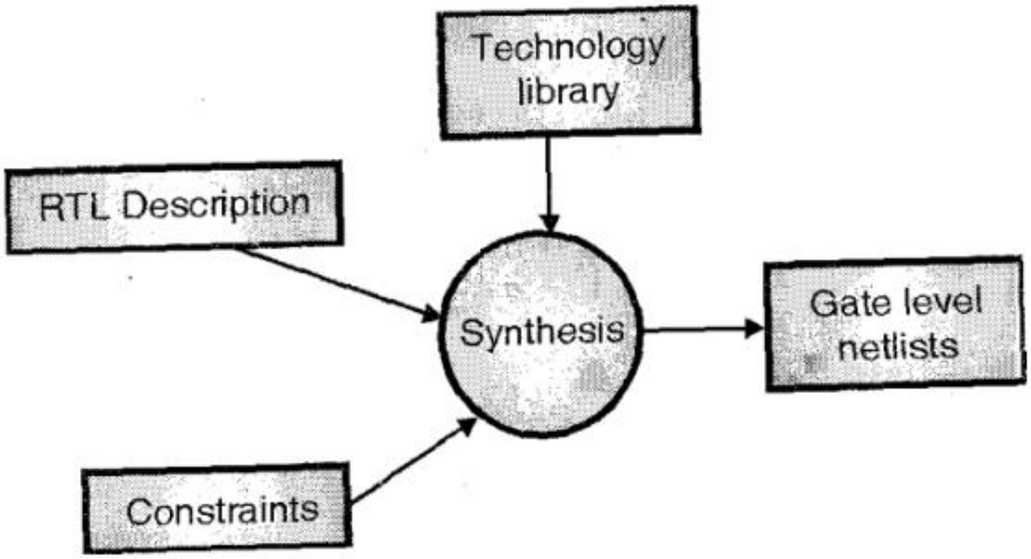
$$D_B = D_{in} \cdot \overline{A} \cdot \overline{B}$$

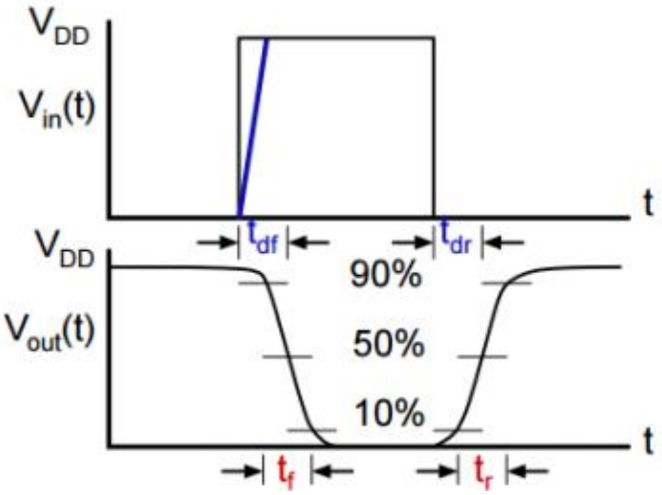
AB

	D_{out}	00	01	11	10
D_{in}	0	0	0	x	1
	1	0	0	x	0

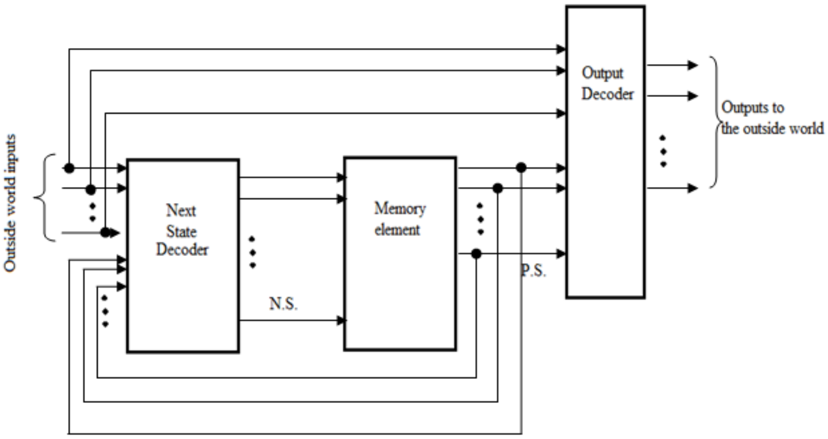
$$D_{out} = \overline{D_{in}}A$$

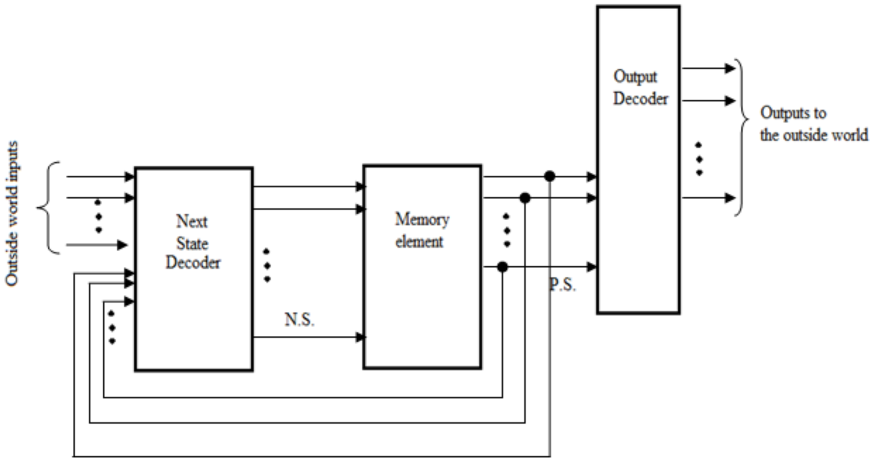
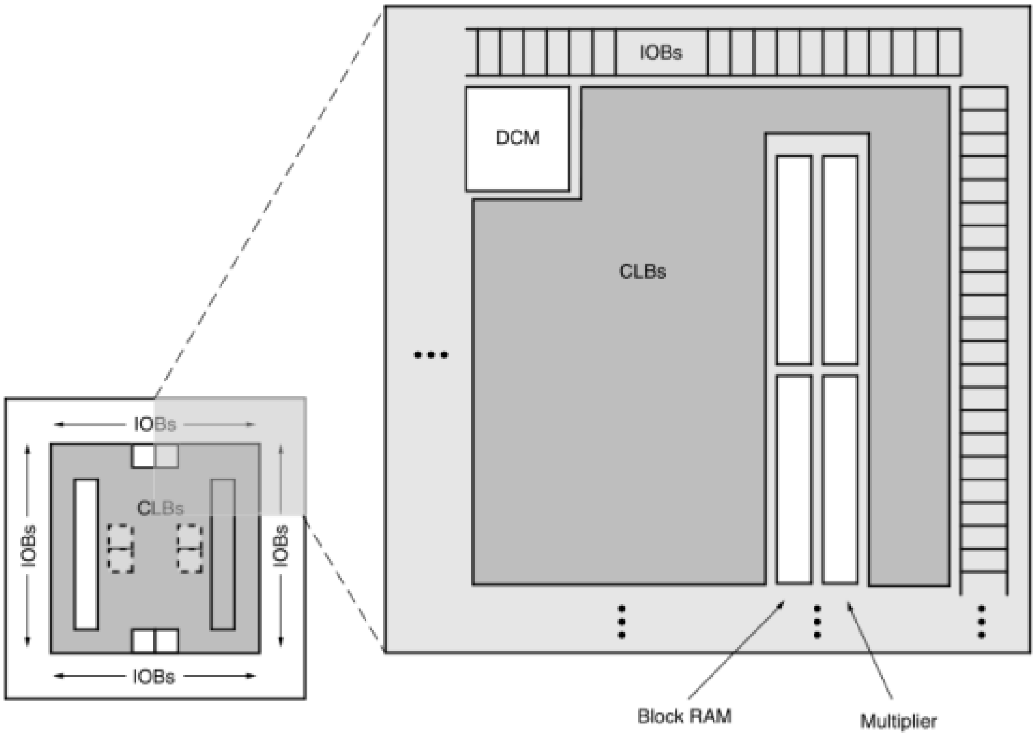
Final implementation of Sequence detector 110 is as follows

		 Din Dout A DA Q A clk Q̄ B DB Q B clk clk Q̄ *Note : the same eg can be done using Moore machine marks to be given*	
Q.5		**Attempt any FOUR :**	**16-Total Marks**
	1)	**Draw the HDL design flow for synthesis.**	**4M**
	Ans:	 Technology library RTL Description Synthesis Gate level netlists Constraints	**Diagram – 4M**
	2)	**State the function of each step elements of VHDL.**	**4M**

	Ans:	1. Library: Many design elements such as packages, definitions and entireentity architecture pairs can be placed in a library. 2. Entity: describes input and output 3. Architecture: operational flow of entity. 4. Component : iinstances of the other design elements 5. Package : Related declarations and design elements like subprograms and procedures can be placed in a “package” for re-use.	
	3)	**Draw CMOS inverter characteristic and explain it.**	**4M**
	Ans:	 V_{DD} V_{in}(t) t t_{df} t_{dr} V_{DD} V_{out}(t) 90% 50% 10% t t_f t_r Characteristics: The characteristics of CMOS inverter depend on the charging and discharging of the load capacitance CL through the PMOS and NMOS transistors respectively. The finite time taken for this charging and discharging is the reason for the delayin circuits. Rise time (tr) The time for a waveform to rise from 10% to 90% of its steady-state value Fall time (tf) The time for a waveform to fall from 90% to 10% steady-state value Delay time (td) The time difference between input transition (50%) and the 50% output level. (This is the time taken for a logic transition to pass from input to output High-to-low delay (tdf) Low-to-high delay (tdr)	**2M Diagram** **2M Explanation**
	4)	**Write VHDL code for 4 : 1 MUX.**	**4M**

	Ans:	**4:1 multiplexer using with-select statement.**
		
		Library IEEE;
		Use IEEE. Std_logic_1164.all;
		Entity MUX4_1 is
			2M entity

		Port(I : in bit_vector (3 downto 0); S: in bit_vector (1 downto 0); Y: out bit); end MUX4_1; architechture MUX of MUX4_1 is begin with S Select Y <= I(0) when “00” I(1) when “01” I(2) when “10” I(3) when “11”; ‘Z’ when others; -- optional end MUX; ***Note – Any relevant code using different statements marks to be given***	**2M architec ture**
	5)	**List the types of FSM. Draw labeled diagram of each.**	**4M**
	Ans:	**Types of FSM:** • Mealy Machine • Moore Machine **Mealy Machine:**  Output Decoder Outputs to the outside world Outside world inputs Next State Decoder Memory element N.S. P.S. **Moore machine:**	**1M List 1 ½ M for each Diagra m**

		 Output Decoder Outputs to the outside world Outside world inputs Next State Decoder Memory element N.S. P.S.	
Q.6		**Attempt any FOUR:**	**16-Total Marks**
	1)	**Explain basic architecture of Sparton-3 FPGA series.**	**4M**
	Ans:	 IOBs DCM CLBs Block RAM Multiplier IOBs CLBs IOBs IOBs IOBs The Spartan-3E family architecture consists of five fundamental programmable functional elements: **Configurable Logic Blocks (CLBs):** Contain flexible Look-Up Tables (LUTs) that implement logic plus storage elements used as flip-flops or latches. CLBs perform a wide variety of logical functions as well as store data. **Input/ Output Blocks (IOBs):** Control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Double Data-Rate (DDR) registers are included.	**2M Diagram** **2M Explanation**

Block RAM : Provides data storage in the form of 18-Kbit dual-port blocks.
Multiplier Blocks : Accept two 18-bit binary numbers as inputs and calculate the product.
Digital Clock Manager (DCM): Blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase-shifting clock signals.

2) **Write VHDL code for 3-bit down counter.** **4M**

Ans:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_Unsigned.all;
entity downcount is
     port( CLK: in std_logic;
preset:instd_logic;
count:instd_logic;
              Q:out std_logic_vector(3 downto 0));
     end downcount;
architecture behavorial of downcount is
signal value:std_logic_vector(3 downto 0);
begin
       process(CLK, preset)
            begin
              if preset=’1’ then
                  value<=(others=>’1’);
elsif (CLK’event and CLK=’1’) then
                  if count=’1’ then
                      value<=value-1;
endif;
endif;
        end process;
Q<=value;
end behavorial;
```

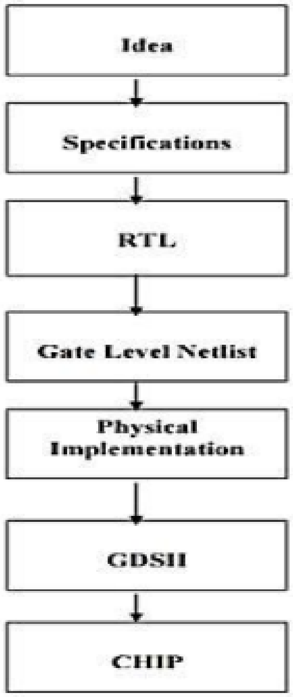
2M- entity
2M- architecture

NOTE : without Preset and count program is also considered

3) **Draw design flow of ASIC and explain it.** **4M**

Ans: **Note: Any other relevant diagram and explanation can be consider**

2M Diagram



2M

Specifications: In this step all the functionality and features are defined, such as power consumption, voltage reference, timing restrictions and performing criterion. Chip planning is also performed in this step.

The next step is to decide the architecture for the design from the specification.

RTL Coding: This is beginning of the ASIC design flow. The micro architecture is transformed into RTL code, RTL is expressed usually in Verilog or VHDL, by using a HDL one can describe any hardware (digital) at any level.

Simulation: Functional/Logical Verification is performed at this stage to ensure the RTL designed matches the idea.

Synthesis: Once Functional Verification is completed, the RTL is converted into an optimized Gate Level Net list. This step is called Logic/RTL synthesis. This is done by Synthesis Tools such as Design Compiler (Synopsys), Blast Create (Magma), RTL Compiler (Cadence) etc... A synthesis tool takes an RTL hardware description and a standard cell library as input and produces a gate-level net list as output. The resulting gate-level net list is a completely structural description with only standard cells at the leaves of the design.

At this stage, it is also verified whether the Gate Level Conversion has been correctly performed by doing simulation.

Physical Implementation: The next step in the ASIC flow is the PhysicalImplementation of the Gate Level Netlist. The Gate level Netlist is converted intogeometric representation. The geometric representation is nothing but the layout of thedesign. The layout is designed according to guidelines based on the limitations of thefabrication process.

The Physical Implementation step consists of three sub steps; Floor planning,

Placement, Routing. The file produced at the output of the Physical Implementation is the GDSII file. It is

the file used by the foundry to fabricate the ASIC. Physical Verification is performed

to verify whether the layout is designed according the rules.

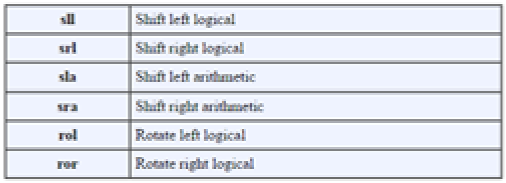
For any design to work at a specific speed, timing analysis has to be performed. We need to check whether the design is meeting the speed requirement mentioned in the specification. This is done by Static Timing Analysis Tool; it validates the timing performance of a design by checking the design for all possible timing violations for example; set up, hold timing.

After Layout, Verification, Timing Analysis, the layout is ready for Fabrication. The layout data is converted into photo lithographic masks. After fabrication, the wafer is diced into individual chips. Each Chip is packaged and tested.

4) **Explain the shift and logical operations.** **4M**

Ans:

Shift operators:
These are:



sll	Shift left logical
srl	Shift right logical
sla	Shift left arithmetic
sra	Shift right arithmetic
rol	Rotate left logical
ror	Rotate right logical

Each of the operators takes an array of BIT or BOOLEAN as the left operand and an integer value as theright operand and performs the specified operation.

The sll operator (shift left logical) and srl operator (shift right logical) fill the vacated bits with left operandtype LEFT.

The sla operator (shift left arithmetic) fills the vacated bits with rightmost bit of the left operand, whilethe sra operator (shift right arithmetic) fills the vacated bits with the left most of the left operand. Theoperator causes the vacated bits to be filled with the displaced bits in a circular fashion.

These are “1001010” sll 2 is “0101000” -- filled with BIT LEFT, which is ‘0’.

“1001010” srl 3 is “0001001” – filled with ‘0’.

2M each operato r

<table>
<tr><td></td><td></td><td>
“1001010” sla 2 is “0101000” – filled with rightmost bit which is ‘0’.
“1001010” sra 3 is “1111001” –filled with ‘1’which is the leftmost bit.
“1001010” rol 2 is “0101010” –rotate left
“1001010” ror3 is “0101001” –rotate right

<b>Logical Operators:</b>
The logical operators and, or, nand, nor, xor, xnor and not are defined for BIT and BOOLEAN types, as well as for one-dimensional arrays containing the elements of BIT and BOOLEAN. All these operators have the lowest priority, except for the operator not, which has the highest priority. The BIT type is represented by the values 0 and 1 while the Boolean type by Trueand False.
</td><td></td></tr>
<tr><td></td><td><b>5)</b></td><td><b>State different modeling styles used in VHDL and write VHDL code for 1 : 4 DEMUX using any one style.</b></td><td><b>4M</b></td></tr>
<tr><td></td><td><b>Ans:</b></td><td>
<b>State Different Modeling styles:</b>
1. Dataflow modeling
2. Structural modeling
3. Behavioural modeling
4. Mixed modelling

<b><u>1:4 DEMUX using Behavioural.</u></b>
<pre>
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity demultiplexer1_8 is
    port(
        din : in STD_LOGIC;
        sel : in STD_LOGIC_VECTOR(1downto 0);
        dout : out STD_LOGIC_VECTOR(3downto 0)
        );
end demultiplexer1_4;
architecture behavorial of demultiplexer1_4 is
begin
       dout<= (din & "000") when (sel="00") else
           ('0' & din & "00") when (sel="01") else
           ("00" & din & "0") when (sel="10") else
           ("000" & din) ;
   end behavorial;
</pre>
<b><i>Note – Any other Modeling style marks to be given</i></b>
</td><td><b>2M for State</b>

<b>2M Code</b></td></tr>
</table>