MODEL ANSWER

WINTER– 17 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 anyequivalent 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.

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

Q. No.	Sub Q.N.	Answer	Marking Scheme
Q.1	**A)**	**Attempt any three :**	**12-Total Marks**
	a)	**Write VHDL program for 3 bit up counter.**	**4M**
	Ans:		**Entity-1M, Architecture-3M**

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

entity Counter_3bit is
   Port ( CLK : in  STD_LOGIC;
          Count : out  STD_LOGIC_VECTOR (2 downto 0));
end Counter_3bit;

architecture Behavioral of Counter_3bit is
   signal cin : std_logic_vector(2 downto 0) :="000";
begin
   process(CLK)
   begin
      if(rising_edge(CLK)) then
         if(cin = "111") then
cin<= "000";
         else
cin<= cin + 1;
         end if;
      end if;

      Count <= cin;
   end process;
end Behavioral;
```

		OR
		signal cin : std_logic_vector(2 downto 0) :="000";
		begin
		process(CLK)
		begin
		if (CLK= '1' and CLK'event) then
		cin<= cin + 1;
		end if;
		
		Count <= cin;
		end process;
		end Behavioral;	

	b)	**Define the terms:** **i) Noise margin** **ii) Skew.**	**4M**
	Ans:	**<u>Only definition is expected correct definition of each carries 2 Marks</u>** **<u>Noise Margin:</u>** Noise margin is the amount of noise that a CMOS circuit could withstand without compromising the operation of circuit. **<u>OR</u>** 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). **<u>Skew:</u>** 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. **<u>OR</u>** The difference in the clock arrival time is call clock skew. **<u>OR</u>** In circuit designs, **clock skew** (sometimes called **timing skew**) is a phenomenon in <u>synchronous circuits</u> in which the clock signal (sent from the <u>clock circuit</u>) arrives at different components at different times.	**2M** **2M**
	c)	**State the use and syntax of:** **i) Signal** **ii) Constant.**	**4M**
	Ans:	**<u>Signal:</u>** Signals are communication media between entities. Signals are nothing but wires which connect two or more components lying inside IC. Signals can be declared in package can be shared among entities and are called global signals. Signals hold a list of values which includes the current value of the signal and a set of possible future values that are to appear on the signal. Signal objects are used to connect entities together to form models. Signals declared in any entity declaration section are global to any architecture for that entity. Signals declared in architecture can only be referenced in that architecture only. **Syntax**: Signal signal_name :signal type := initial value. **<u>Constant:</u>** Constant objects are names assigned to specific values of a type. Constant is an	**2M each**

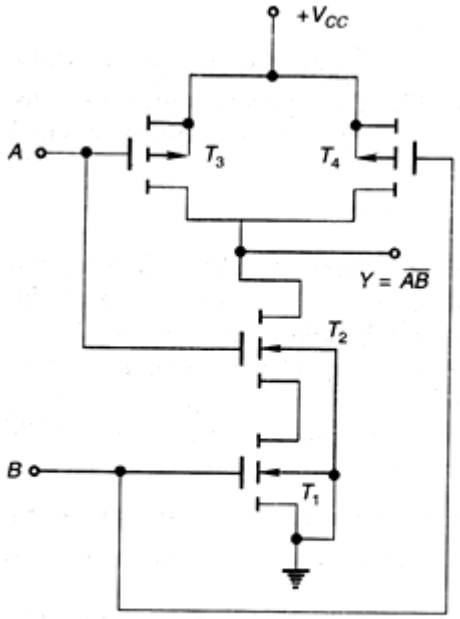
object in VHDL whose value cannot be changed once defined for the design. By use of constant model becomes readable and easy to update. A single value of a given type is assigned to the constant before simulation starts and this value cannot be changed during simulation.

Syntax: Constant constant_name: type_name := value.

Note : initial value if any

d) Draw two input NAND gate using CMOS technology. Write its truth table. **4M**

Ans: **Diagram-** **2M**



NAND Gate

Truth table-

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

Truth table 2M

e) Explain efficient coding style. **4M**

Ans: A coding style is set of rules that a programmer uses for choosing an expressive form to use in the given situation. **4M**

- There may be more than one method to model a particular design part but only a few would yield better performance.
- The essence of VHDL coding lies in understanding which style yields the ultimate performance under the given set of specifications.
- The key to higher performance is to avoid writing code that needlessly creates additional work for the HDL compiler and synthesizer, which, in turn, generates designs with greater number of gates.
- Basically, any coding style that gives the HDL simulator information about the design that

	cannot be passed onto the synthesis tool is a bad coding style. **Rules:** 1. Use optimised standard libraries: The performance is increased when standard libraries are used instead of unoptimized. 2. Reduce process sensitivity: this will prevent the function getting unnecessarily and repeatedly executed. 3. Reducing waits. 4. Reduce delay calculations. 5. Integers vs. Vectors: To increase the performance ranged integers are used in entity instead of std_logic_vectors. The simulator may be able to process the design faster and efficiently. 6. Optimize everything above 1%: The performance analyser will identify the lines of code that consumes the greatest CPU time and display these lines in order in the performance profile window.	
B)	**Solve any one :**	**12-Total Marks**
a)	**Explain different level of simulation in brief**	**6M**
Ans:	• **Behavioural Simulation:** Behavioural simulation employs a high level of abstraction to model the design. And also timing aspects are considered. **Ex:** f <=a and b after ns; • **Functional simulation:** It ignores the timing aspects of the circuit and verifies only the functionality of the design. • **Static timing analysis:** It has a built in tool that computes delay for each timing path. It does not require input stimuli. The reports are generated after simulation. • **Gate level simulation:** It is used to check the timing performance of a design. The delay parameters of logic cells are used to verify timings. • **Switch level simulation:** It is one level below the gate level simulation. It models transistors which are used in gates as switches. It provides more accurate timing predictions than the gate level simulation. **Transistor level or circuit level simulation:** It requires transistor models. The circuit is described in terms of resistances, capacitances, voltages and current sources. A set of mathematical equations relating current and voltage is set up and solved by numerical techniques. It requires data structures and large amount of computing resources and gives analog results. It is the most accurate method. The frequency response can also be computed.	**1M each**
b)	**Explain following statement with example:** **i) Process statement** **it) Wait statement.**	**6M**
Ans:	**i) Process statement:** Process statement is the primary mechanism used in sequential behavior It contains sequential statements, variable assignment (:=) statements or signal assignment (<=) statements etc. It may or may not contain sensitivity list. If there is an event occurs on	**3M Each**

any of the signals in the sensitivity list, the statements within the process is executed.
Inside the process the execution of statements will be sequential and if one entity is having two processes the execution of these processes will be concurrent. At the end it waits for another event to occur.

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

entity Counter_3bit is
   Port ( CLK : in  STD_LOGIC;
          Count : out  STD_LOGIC_VECTOR (2 downto 0));
end Counter_3bit;

architecture Behavioral of Counter_3bit is
   signal cin : std_logic_vector(2 downto 0) :="000";
begin

   process(CLK)
   begin
      if (CLK= '1' and CLK'event)  then
cin<= cin + 1;
         end if;


      Count <= cin;
   end process;
end Behavioral;
```

ii)Wait Statements:

wait: statement suspends the execution of a process or procedure until some conditions are met.
wait on [sensitivity list]; eg. wait on clk;

wait until [condition]; wait until clk='1''
wait for [time out expression]; wait for 20 ns;

OR

Wait statements put the process execution on hold until the specified condition is fulfilled. If no condition is given, the process will never be reactivated again.
Wait statements must not be combined with a sensitivity list, independent from the application field. Wait statement stop the process execution. The Process is continued when the instruction is fulfilled

Different types of wait statement:

- Wait for a specific time: wait for SPECIFIC_TIME;
- Wait for a signal event: wait on SIGNAL_LIST;
- Wait for a true condition (requires an event): wait until CONDITION;

➢ Indefinite (process is never reactivated): wait ;

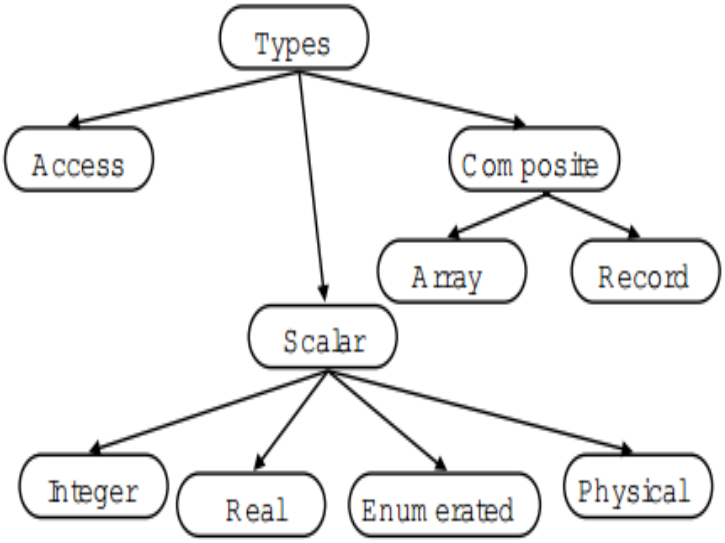
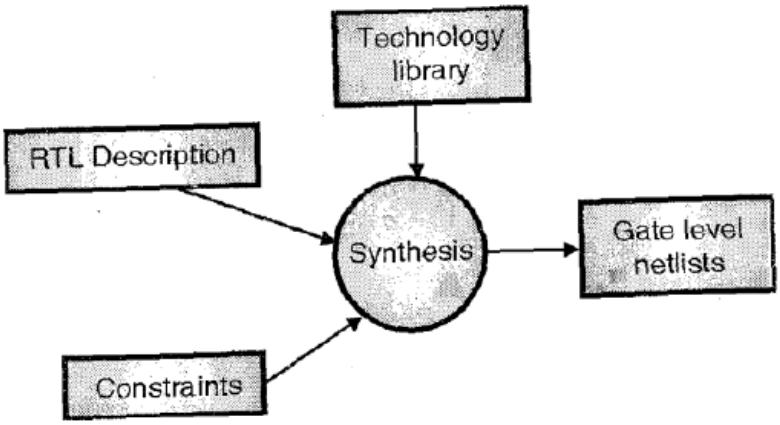
Example:

```
entity FF is
        port(D,      CLK      :      in
bit;
                Q              :     out
bit);
end FF;
architecture BEH1 of FF is
begin
  process
  begin
    wait on CLK;
    if (CLK = '1') then
      Q <= D;
    end if;
  end process;
end BEH1;
```

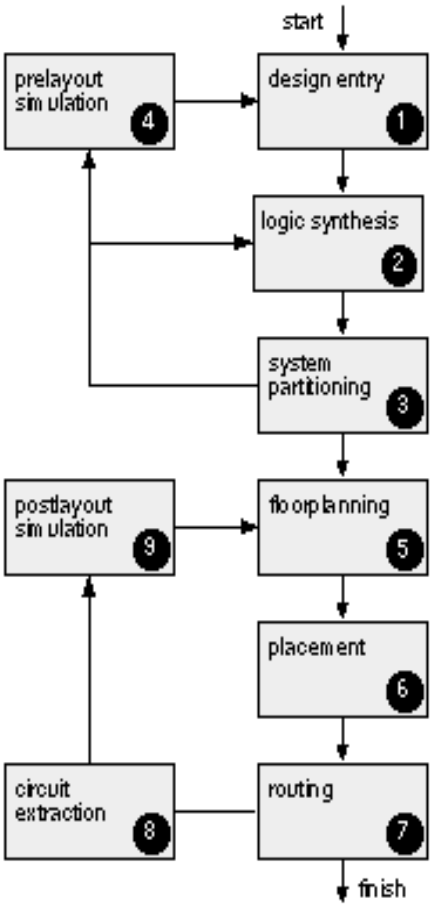
Q 2		**Attempt any two:**	**12-Total Marks**
	a)	**Differentiate between BJT and CMOS.**	**4 M**
	Ans:		**Any 4 points, 1M each**

Sr. No.	Bipolar Junction Transistor	Complementary Metal Oxide Semiconductor
1	BJT junctions are emitter base and collector	CMOS junctions are gate, source, drain and substrate
2	LOW power applications	High power applications
3	Bipolar device	Unipolar Device
4	Low input Impedance	High Input Impedance
5	Low current gain	High Current gain
6	More fan out	Less fan out
7	Low packing density	High Packing density
8	Connecting BJT's together gives rise to a family of logic gates known as TTL	Connecting NMOS and PMOS transistors together gives rise to the CMOS family of logic gates

	b)	**List the data types and explain anyone.**	**4M**

	Ans:	 There are mainly two types: 1. Scalar 2. Composite **<u>Scalar Types :</u>** Integer: Defines the value with Integer. (Integer Range). Real: Defines the value with number. Enumerated: Defines the set of user defined values consisting of identifiers and character literals Physical :used to represent physical quantities e.g. distance ,time **<u>Composite :</u>** 1. Array: Contain many elements of same type. 2. Record : Contain elements of different types	**1M for List, 1.5 for Scalar, 1.5 for Composite.**
	c)	**Draw and explain HDL design flow for synthesis.**	**4M**
	Ans:	**<u>Diagram-</u>** 	**2M**

		Explanation- **HDL Design flow for Synthesis:** • To convert RTL (register description to gates), there are three steps as under: 1. Translation 2. Boolean Optimization 3. Optimization 1. **Translation**: • RTL description is translated to un-optimized Boolean description usually with primitive gates like AND and OR gates, and flip-flops and latches. • RTL description to Boolean equivalent circuit is not controlled by user. **2. Boolean Optimization:** • Algorithms are executed on the Boolean optimized description. **3. Optimization:** • Optimization process takes an unoptimization Boolean description. • Optimization uses number of algorithms to convert unoptimizes Boolean descriptions to a very low level description (PLC format). • Thus, when we optimize 'PLA Format', we get description. We have to reduce the logic generated by sharing common terms. • Optimized Boolean equivalent description is mapped to actual logic gates by making use of technology library of target process. **OR** 1. Describe your design with HDL 2. Perform RTL simulation 3. Synthesizing your design 4. Create Xilinx Netlist Files (XNF/EDIF etc) 5. Perform Functional Simulation 6. Floor planning of design (optional) 7. Placing and routing 8. Perform a timing simulation (post layout)	**2M**
	d)	**Draw ASIC design flow and explain it.**	**4M**
	Ans:	**Diagram-**	**2M**

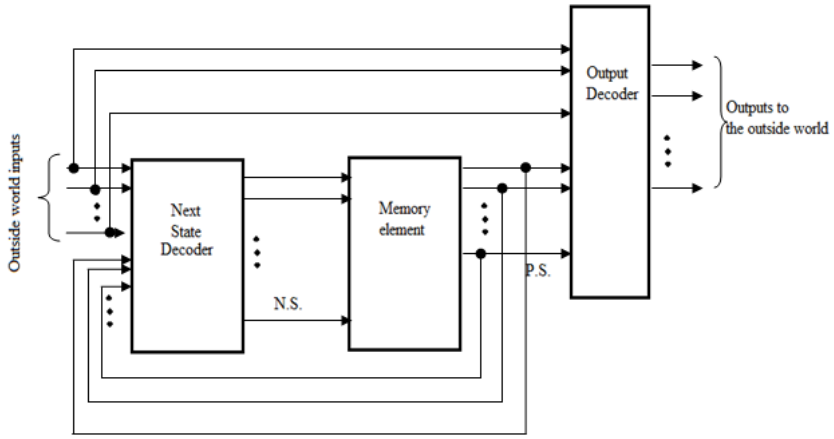


Explanation-

S-1 Design Entry: Schematic entry or HDL description
S-2: Logic Synthesis: Using Verilog HDL or VHDL and Synthesis tool, produce *a netlist*- logic cells and their interconnect detail
S-3 System Partitioning: Divide a large system into ASIC sized pieces
S-4 Pre-Layout Simulation: Check design functionality
S-5 Floorplanning: Arrange netlist blocks on the chip
S-6 Placement: Fix cell locations in a block
S-7 Routing: Make the cell and block interconnections
S-8 Extraction: Measure the interconnect R/C cost
S-9 Post-Layout Simulation

2M

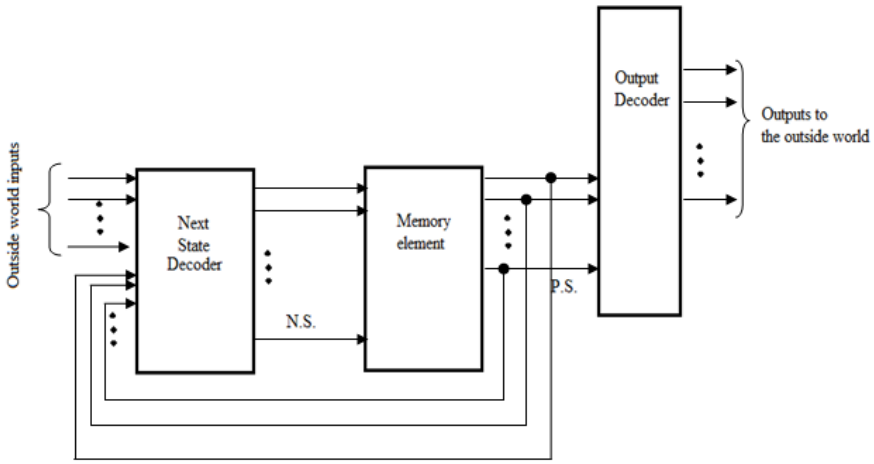
e)	**Draw block diagram of Me lay and Moore machine and explain it.**	**4M**
Ans:	**Melay Machine:**	



1M

Mealy machine is the sequential system where output depends on present input and state.
f (o/p) = f (i/p, P.S.) 1M

Moore Machine:



1M

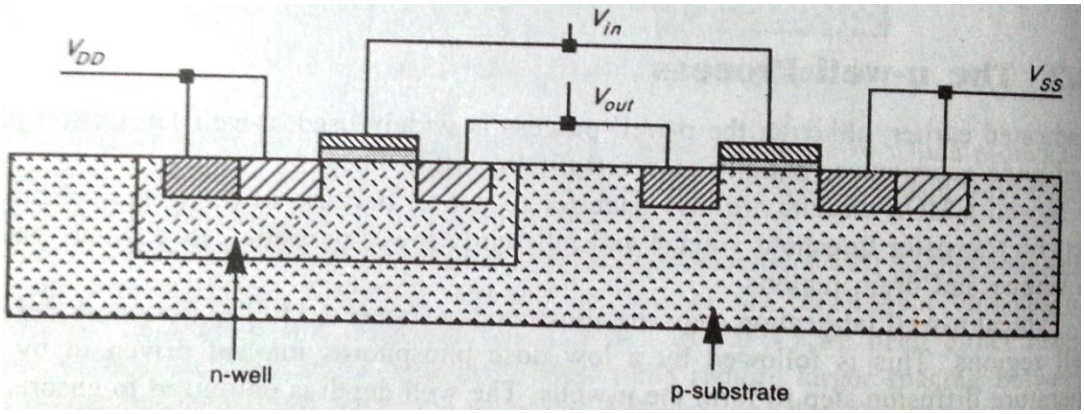
Moore machine is the sequential system where output depends only on present state.
f (o/p) = f (P.S.) 1M

Q. 3		Attempt any four:	16-Total Marks
	a)	**Explain fabrication ofN-well process.**	4M
	Ans:	**N-Well Process:** • The n-well CMOS circuits are also superior to p-well because of the lower substrate bias effect on transistor threshold voltage and inherently lower parasitic capacitance associated with source and drain regions. • The typical n-well fabrication steps are shown below. • The first mask defines the n-well regions. This is followed by a low dose phosphorous implant driven in by a high temperature diffusion step to form the n-well. • The well depth is optimized to ensure against p-substrate to p+ diffusion breakdown without compromising the n-well to n+ mask separation.	**Explanation 4M ,Diagram optional**

- The next steps are to define the devices and diffusion paths, grow field oxide, deposit and pattern the polysilicon, carry out the diffusion, make contact cuts and finally metallise.

OR

- Thick SiO2 layer is grown on p-type silicon wafer.
- After defining the area for N-well diffusion, using a mask, the SiO2 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 VDD and VSS 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.



	b)	**Define the terms with syntax:** **i) Entity** **ii) Architecture.**	**4M**
	Ans:	**<u>Entity:-</u>** • Entity is the description of inputs and outputs of the design. An entity is the most basic building block in the design. A design can have more than one entity block. • The entity statement declares the design name. Then it defines input output parameters and ports of the design entity. **<u>Syntax:-</u>** entity entity_name is generic (generic_list);	**1M explanation 1M for syntax (each)**

		port (port_list);] end entity entity_name; **<u>Architecture:-</u>** • All entities that are declared have an architecture associated with it. Architecture describes the behavior of the entity. An entity can have multiple architectures. • Architecture assigned to an entity describes internal relationship between input and output of the entity. First part of the architecture may contain declaration of types, signals, constants, subprograms etc. **<u>Syntax:-</u>** architecture architecture_name of entity_name is architecture_declarations begin concurrent_statements end [architecture] [architecture_name];	
	c)	**Write VHDLcode for D flip-flop.**	**4 M**

	Ans:	VHDL for D- flip flop
		library ieee;
		use ieee.std_logic_1164.all;
		entity flop is
		port(C, D : in std_logic;
		Q : out std_logic);
		end flop;
		architecture archi of flop is
		begin
		process (C,D)
		begin
		if (C'event and C= '1') then
		Q <= D;
		end if;
		end process;
		end archi;
			1M –entity, 3M- architecture.

	d)	**State and explain:** **i)Eventscheduling.** **ii) Simulation cycle.**	**4 M**
	Ans:	**<u>Event scheduling:</u>** Event is nothing but change on target signal which is to be updated. Ex. X<= a after 0.5ns when select=0 else X<= b after 0.5ns The assignment to signal x does not happen instantly. Each of the values assigned to x contain an **after** clause. The mechanism for delaying the new value is called scheduling an event. By assigning port x a new value, an event was scheduled 0.5ns in the future that contains the new value for signal x. when the event matures, signal receives a new value. **<u>Simulation cycle:</u>** Some designs are self-simulating and do not need any external stimulus, but in most of the cases VHDL designers use VHDL test bench to drive the design being tested. Test bench is used to verify the functionality or correctness of a HDL model. Tt is a	**2M each**

specification in HDL that plays the role of a complete simulation environment for the analyzed system.
A test bench is at the highest level in the hierarchy of the design. It instantiates the design under test (DUT) and provides the necessary input stimulus to DUT and examines the output from DUT.
The stimulus driver drives input to the DUT. DUT responds to the input signals and produces output. Finally, it compares the output results from DUT with the expected values and reports any discrepancies. A test bench has three main purposes.
(1) To generate stimulus for stimulation (waveforms).
(2) To apply this stimulus to the entity under test and collect the output responses.
(3) To compare output responses with expected values.

e) Differentiate between CPLD and FPGA. — **4 M**

Ans: — **Any 4 Points 1M each**

Sr. No.	CPLD	FPGA
1	It is complex programmable logic device.	It is field programmable gate arrays.
2	Capacity is defined in terms of number of macro-cells available.	Capacity is defined in terms of number of gates available.
3	CPLD consumes more power than FPGA devices.	FPGA consumes less power than CPLD
4	Number of input and output pins on CPLD are high.	Number of input and output pins on FPGA are less than CPLD.
5	CPLD are ideal for complex blocks with large number of inputs.	FPGA is suitable for designs with large number of simple blocks with few number of inputs.
6	CPLD based designs need less board space and less board layout complexity.	FPGA based designs require more board space and layout complexity is more.
7	It is easier to predict speed performance of design.	It is difficult to predict the speed performance of design.
8.	CPLDs contain fewer registers but has better performance.	FPGA are available in wide density range.

Q. 4 A) **Solve any three :** — **12-Total Marks**

a) **Define the term :** — **4M**
i) Fanout
il) Metastability
iii) Asynchronous sequential circuit.
iv) Synchronous sequential circuit.

Ans: **Fanout**: In digital electronics, the **fan-out** of a logic gate output is the number of gate inputs it can feed or connect to. The **maximum fan-out** of an output measures its load-driving capability: it is the greatest number of inputs of gates of the same type to which the output can be safely connected. — **1M each**

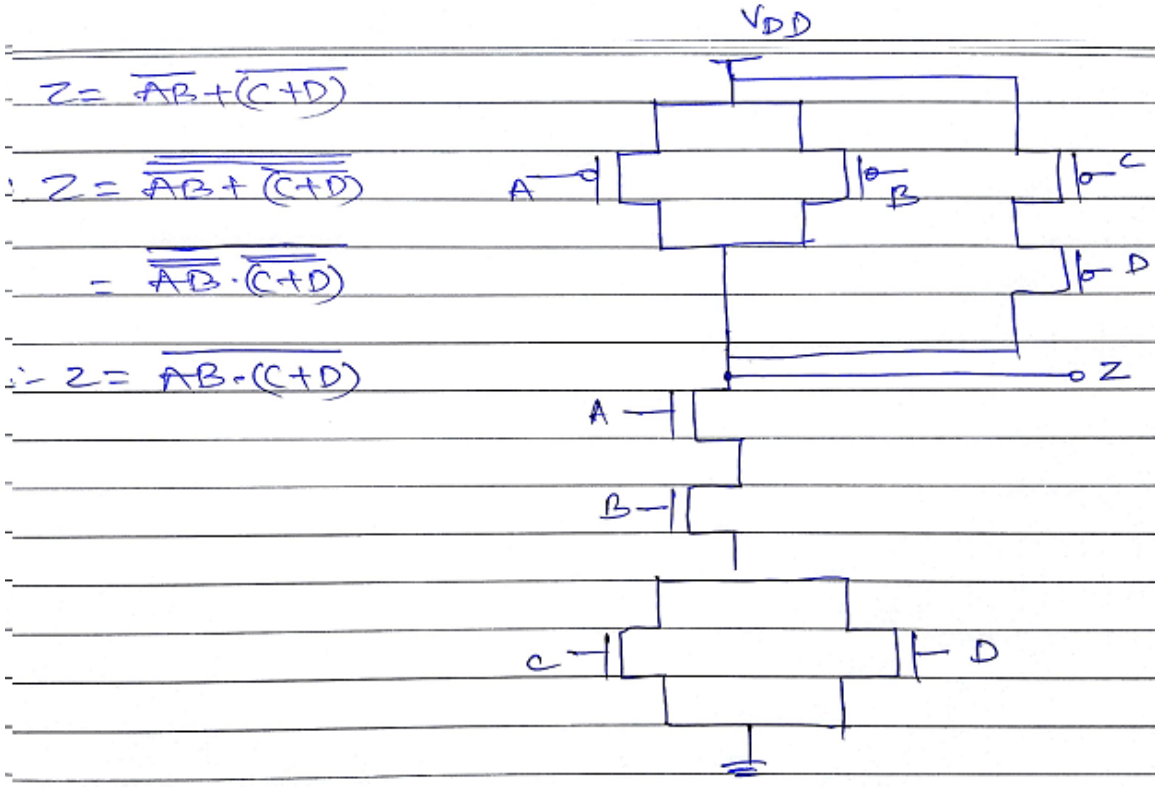
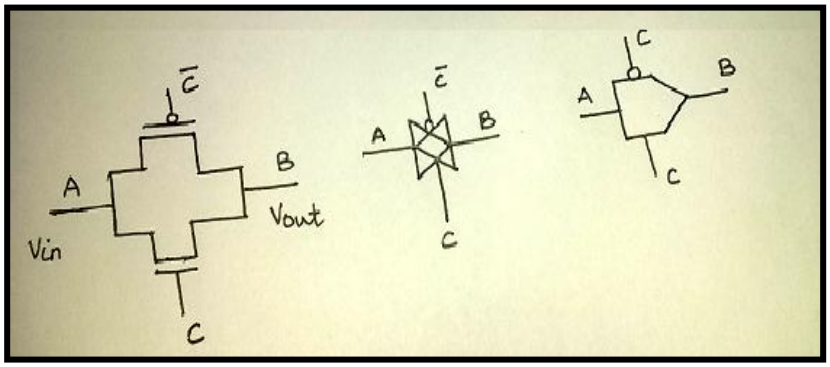
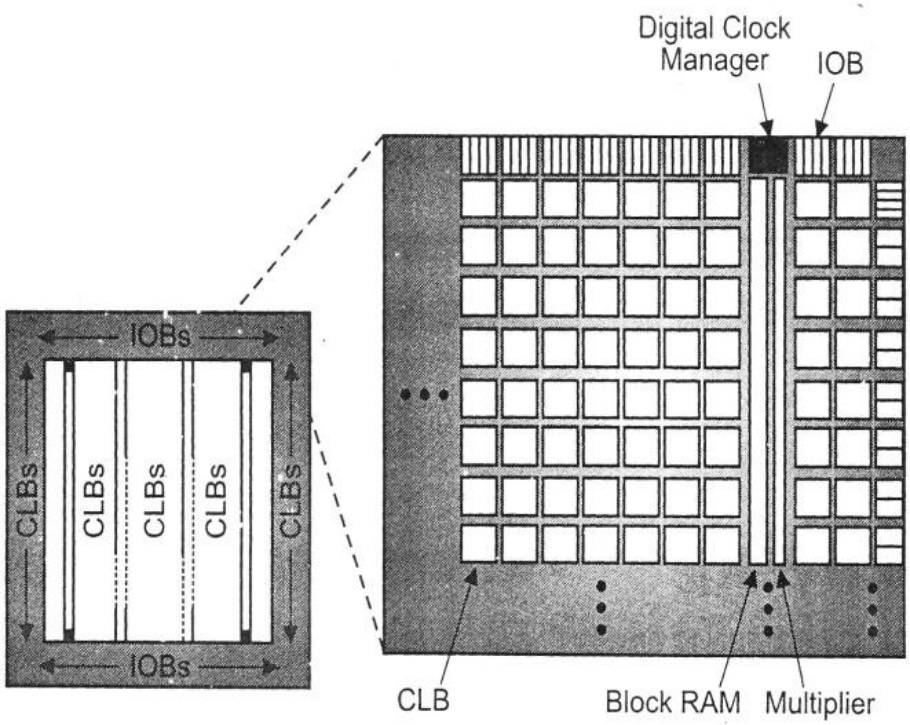
		Metastability: Metastability in electronics is the ability of a digital electronic system to persist for an unbounded time in an unstable equilibrium or metastable state. In metastable states, the circuit may be unable to settle into a stable '0' or '1' logic level within the time required for proper circuit operation. As a result, the circuit can act in unpredictable ways, and may lead to a system failure, sometimes referred to as a "glitch". **Asynchronous sequential circuit**: Output can be changed at any instant of time by changing the input. The status of memory element will change any time as soon as input is changed. These circuits are difficult to design and are faster. **Synchronous sequential circuit:**Output changes at discrete interval of time. The status of memory is affected only at the active edge of clock, if input is changed. These circuits are easy to design and are slower.	
	b)	**Design boolean equation using CMOS: $Z = \overline{AB} + \overline{(C + D)}$.**	**4M**
	Ans:		**4M**
	c)	**Draw transmission gate. Explain it.**	**4M**
	Ans:	**Explaination-** Transmission gate consists of one NMOS and one PMOS transistor in parallel. The gate voltages applied to these two transistors are also set to be complementary signals. The CMOS transmission gate operates as a bidirectional switch between the nodes A and B which is controlled by C. • If C is at high logic then both transistors are ON and provides a low resistance current path between the nodes A and B. • If C is low, then both the transistors are off and path between A and B is open circuit. This condition is called as high impedance state.	**2M**

Diagram- **2M**



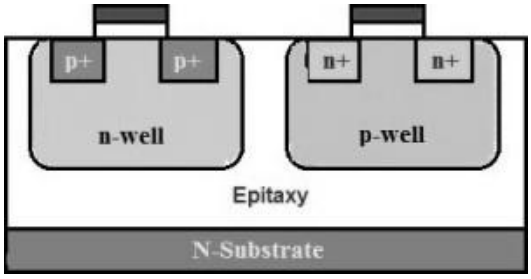
d)	**Draw neat diagram of architecture of SPARTAN- 3 FPGA family and explain it.**		**4M**
Ans:	**Diagram-**		**2M**



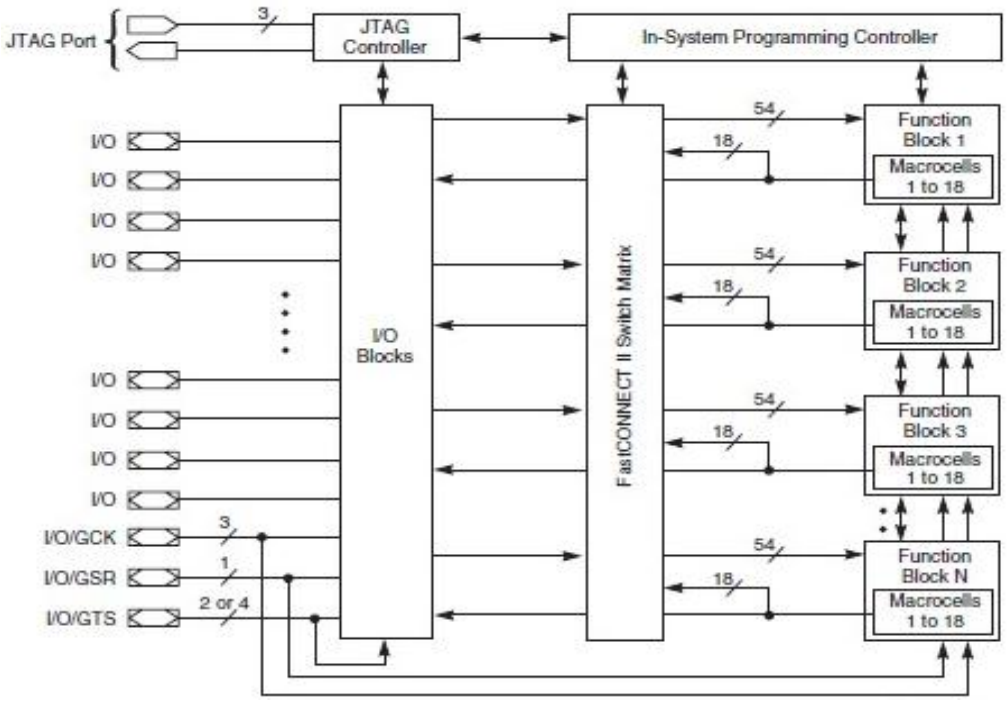
Explaination- **2M**

The Spartan-3 family architecture consists of five fundamental programmable functional elements:

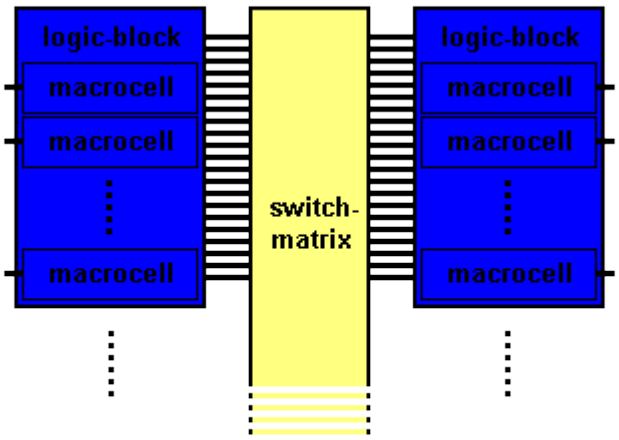
- Configurable Logic Blocks (CLBs): Configurable Logic Blocks (CLBs) contain RAM-based Look-Up Tables (LUTs) to implement logic and storage elements that can be used as flip-flops or latches. CLBs can be programmed to perform a wide variety of logical functions as well as to store data.
- Input/output Blocks (IOBs): 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. The Digitally Controlled Impedance (DCI) feature provides automatic on-chip terminations, simplifying board designs.

		• 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. These elements are organized as shown in Figure. A ring of IOBs surrounds a regular array of CLBs. The Spartan-3 family features a rich network of traces and switches that interconnect all five functional elements, transmitting signals among them. Each functional element has an associated switch matrix that permits multiple connections to the routing.	
	B)	**Solve any one :**	**6-Total Marks**
	a)	**Explain twin tube process with suitable diagram.**	**6M**
	Ans:	In this process the substrate can be of any type. Consider n type silicon substrate. The twin tub fabrication process is: **<u>Diagram -</u>**  **<u>Description-</u>** 1. The process is carried out on N type silicon substrate with lower doping or higher resistivity so that the lesser current flows through the substrate. On this, the n^+Si substrate is grown further i.e. epitaxial layer of required thickness is grown. 2. SiO_2 layer is grown all over the surface and the areas of P well and N well are defined. P well is diffused by masking N well area and N well is diffused by masking P well area. 3. A thin layer of SiO_2 thin ox is deposited all over the surface. Using masking and etching process unrequired thin ox is removed. The thin ox is required only on gate areas of both the transistors. 4. The polysilicon is deposited all over the surface and using a mask it is removed from areas other than the gate area. 5. Then the P well is covered with a photoresist mask and p^+ diffusion is carried out to form the source and drain of pMOS transistor. 6. Now the N well is covered with a photoresist mask and n^+ diffusion is carried out to form the source and drain of nMOS transistor. 7. The thick layer of SiO_2 is grown all over the surface for isolation. This SiO_2 layer is etched off to expose all the terminals. 8. The metal is deposited and patterned all over the wafer surface so that it makes contact with source, drain and gate terminals.	**Diagram - 2M** **Description - 4M**
	b)	**Draw architecture of CPLD. Explain in brief**	**6M**
	Ans:		

Architecture- **3M**



OR



CPLD-architecture

Explaination-

CPLD architecture consist of product terms generated in programmable macrocells, It presents typically one dedicated flip-flop per macrocell, and many macrocell per logic block. All logic block are identical. Logic blocks are routed through global switch matrix. Each Function Block, consist of 18 independent macrocells to implementing a **3M**

combinatorial or registered function. macrocell may be individually configured for a combinatorial or registered function.

The macrocell register can be configured as a D-type or T-type flip-flop, or it may be bypassed for combinatorial operation The product term allocator controls the direct product terms are assigned to each macrocell The I/O Block (IOB) interfaces between the internal logic and the device user I/O pins. Each IOB includes an input buffer, output driver, output enable selection multiplexer, and user programmable ground control

Q.5		**Attempt any FOUR-**	**16-Total Marks**
	a)	**State the pro's and con's ofVHDL.**	**4M**
	Ans:		**2M**

Pros:

- Strongly typed language:
  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.
- Ability to define custom types:
  A VHDL state machine can be coded naturally using the actual state names (e.g. wait, acknowledge, transmit, receive, etc.), not binary state numbers (e.g. 00, 01, 10, 11).
- Record types:
  Define multiple signals into one type.
- Natural coding style for asynchronous resets.
- Easily reverse bit order of a word.
- Logical statement (like case and if/then) endings are clearly marked.

Cons: **2M**

- Extremely verbose coding:
  VHDL modules must be defined by a prototype and declared before they're used, causing you to change code in at least 3 places if you want to make a change to the interface.
  The use of the keyword "downto" in every bit vector definition is tedious.
- Sensitivity lists: Missing a single signal in the sensitivity list can cause catastrophic differences between simulation and synthesis.
  Each process must have a sensitivity list that may sometimes be very long.
- Type conversions:
  Signal types that are clearly related (e.g. std_logic and std_logic_vector) cannot be simply used together and must be converted to another type.

OR

Pro's

*Using the same language it is possible to simulate as well as design a complex logic.

*Design reuse is possible

*Design can be described at various levels of abstractions.

*It provides modular design and testing.

*The use of VHDL has tremendously reduced the "Time to Market " for large and small design.

*VHDL designs are portable with synthesis and simulation tools, which adhere to the IEEE

1076 standard.

*Using VHDL makes the design device independent.

*The design description can be targeted to PLD, ASIC, FPGA very easily.

Con's

*Designer has very little control at gate level.

*The logic generated for the same description may vary from tool to tool. This may be due to algorithm used by the tools, which might be proprietary.

b) Compare concurrent and sequential statement. **4M**

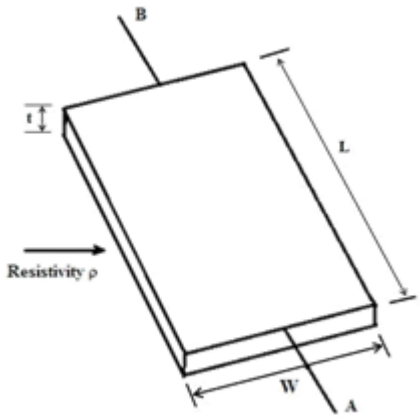
Ans:

Concurrent Statement	**Sequential Statement**
Many of these statements can be active at the same time.	A set of VHDL statements that executes in sequence is called sequential statements.
Inside Architecture	Inside process
Simple signal assignment statement	Sequential signal assignment statement
Conditional signal assignment statement	Variable assignment statement
E.g. Process, Component Instance, concurrent signal assignment.	If, for, switch-case, signal assignment.

1M Each Point

c) Explain estimation of channel resistance of CMOS. **4M**

Ans: Consider a uniform slab of conducting material of resistivity ρ. Let W be the width, t the thickness and *L* the length of the slab. **4M**



Hence the resistance between A and B terminal.

$R_{AB} = \frac{\rho L}{A}$ ohms.

Where A = cross-sectional area.

Thus $R_{AB} = \frac{\rho L}{t.W}$ ohms.

Consider the case in which L = W, that is a square of resistive material then

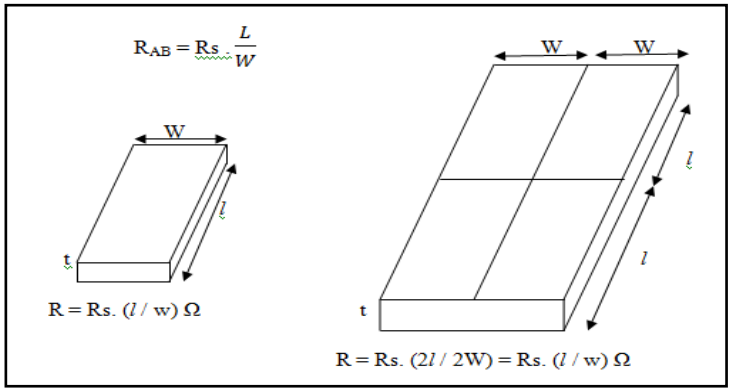
$$R_{AB} = \frac{\rho}{t} = Rs$$

Where

Rs = ohm per square or sheet resistance

Therefore, Rs = $\frac{\rho}{t}$ ohm per square

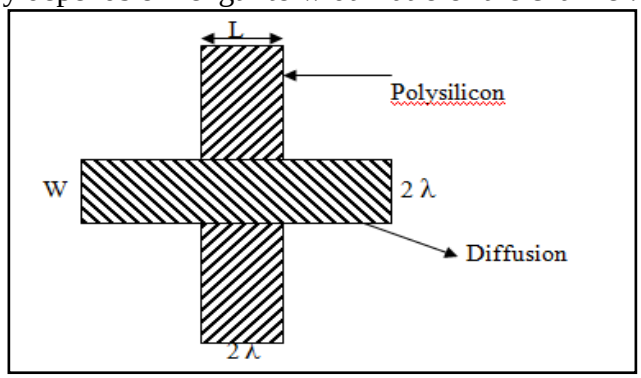
Hence Rs is completely independent of the area of the square.



Resistances of the two shapes shown in the above figure are same because the length to width ratio of both the slabs is same, even though the sizes are different. The voltage – current characteristics of a MOS transistor are generally nonlinear, it is used to approximate its behavior in terms of a change resistance to estimate the performance.

- The channel resistance Rc
- Rc = K ($\frac{L}{W}$)
- Where K = $\frac{1}{\mu \text{ Cox (vgs}-\text{Vt)}}$

  μ = surface mobility of majority carriers. (i.e. electrons in n-device and holes in p-device)
- Since mobility and threshold voltage are temperature dependent parameters, the channel resistance changes with temperature. But as given in equation of Rc, channel resistance mainly depends on length to width ratio of the channel.



- In the above diagram both poly and diffusion are of 2λ widths. The overlapping region is called a 'channel', with length and width 2 λ, as shown in figure. The thinnox is only in the channel region.
- In the above example channel length L = 2 λ and width W =2 λ.
- The channel is square in shape and channel resistance.

R = Rs ($\frac{L}{W}$)

Therefore, R = Rs $(\frac{2\lambda}{2\lambda})$

Therefore, R = Rs ohms.

d) Write various factor of selection ofFPGA. **4M**

Ans: **Factors for selection of FPGA** **Any 4 factors-4M**

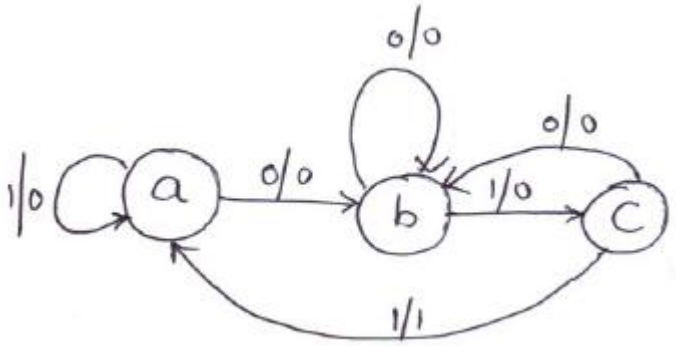
1. Technical Feasibility : Vendor
2. Cost
3. External Devices interface for download
4. No of I/O Pins
5. Frequency of operation
6. Number of flip flop, LUT, mux, adder, multipliers present

e) Design a sequence detector to detect 011 using JK flip-flop using Melay machine. **4M**

Ans: **State diagram-** **State diagram-1M**

State Table-1M

Circuit Diagram-2M



State table-

Previous State	Next State	Output	Next State	Output
	X=0		X=1	
a	b	0	a	0
b	b	0	c	0
C	b	0	a	1
d	X	X	X	X

Using straight binary assignment LET a= 00 b=01 c=10 and d=11

EXCITATION TABLE of JK Flip Flop

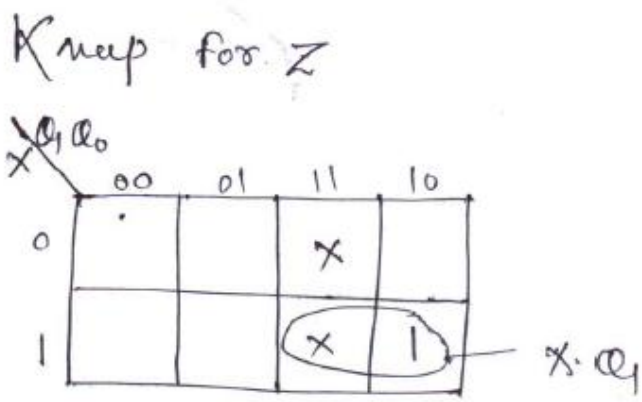
EXCITATION		INPUTS	
Q	Q*	J	K
0	0	0	X
0	1	1	X
1	0	X	1

1	1	X	0

STATE TABLE

Input	Previous State		Next State		Output	EXCITATION			
X	Q1	Q0	Q1*	Q0*	Z	J1	K1	J0	K0
0	0	0	1	0	0	1	X	0	X
0	0	1	1	0	0	1	X	X	1
0	1	0	1	0	0	X	0	0	X
0	1	1	X	X	X	X	X	X	X
1	0	0	0	0	0	0	X	0	X
1	0	1	1	0	0	1	X	X	1
1	1	0	0	0	1	X	1	0	X
1	1	1	X	X	X	X	X	X	X

K-MAPS and CIRCUIT DIAGRAM



$Z = X \cdot Q_1$

K map for J_1

X \ Q_1Q_0	00	01	11	10
0	1	1	X	X
1		1	X	X

$\overline{X}$

Q_0

$J_1 = \overline{X} + Q_0$

K map for K_1

X \ Q_1Q_0	00	01	11	10
0	X	X	X	
1	X	X	X	1

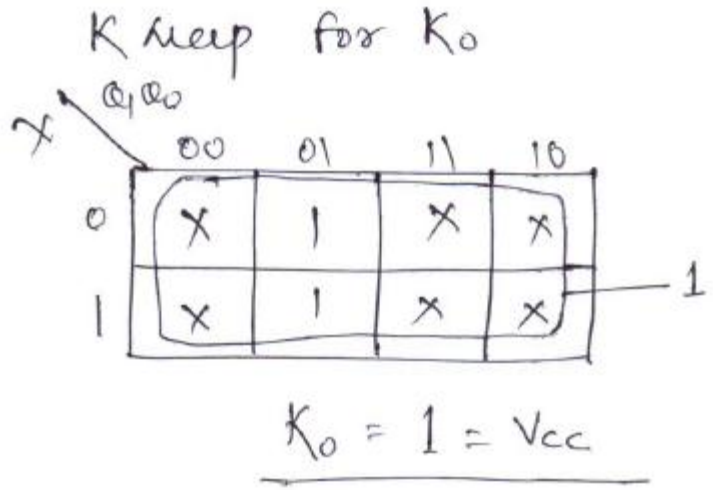
X

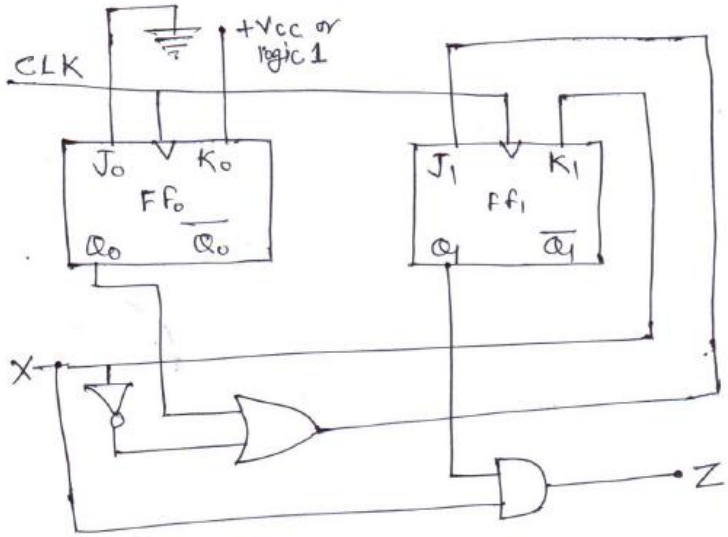
$K_1 = X$

Kmap for J_0

X \ Q_1Q_0	00	01	11	10
0		X	X	
1		X	X	

$J_0 = 0$





f)	**State and explain Delta delay.**	**4M**
Ans:	The real time that the simulator takes to execute one simulation cycle is known as delta delay for simulation delta with zero simulation time .A delta delay is very small and does not correspond to any real delay and actual simulation time does not advance. Delta delay is introduced to achieve concurrency and order independency . The simulator freezes simulation time until all scheduled assignments in current simulation time is finished and there are no more events in the sensitivity list.	**Definition-2M, Any suitable example-2M**

B<= NOT A; C<= B NAND CLK; D<= C AND B;	B<= NOT A; D<= C AND B; C<= B NAND CLK;
If A changes ‘1’ to ‘0’	
B <=1	B <=1
Evaluate AND (C=1)	Evaluate NAND
D<=1	C<=0
Evaluate NAND	Evaluate AND

C<=0	D<=0	
Evaluate AND		
D<=0		

Both the syntax gives the same values and achieved concurrency with independent order of syntax.

Q.6		Solve any four :	16-Total Marks
	a)	**Write VHDL program to implement 4: 1 mux using case statement.**	4M
	Ans:		Entity-1M Architecture-3M

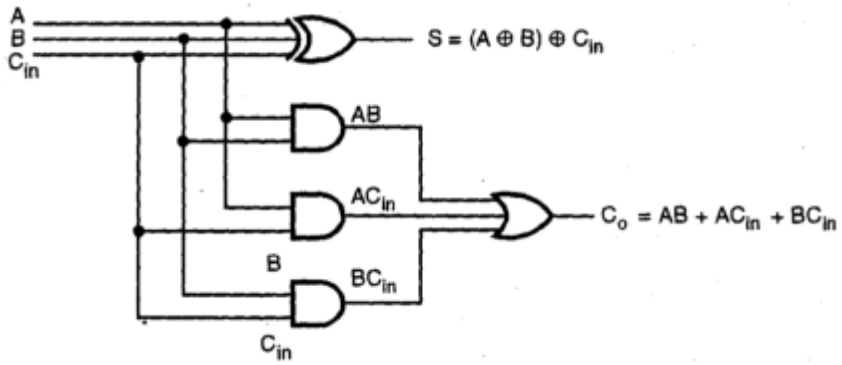
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity multiplexer4_1 is
port (
i0 : in std_logic;
i1 : in std_logic;
i2 : in std_logic;
i3 : in std_logic;
sel : in std_logic_vector(1 downto0);
y : out std_logic );
end multiplexer4_1;

architecture Behavioral of multiplexer4_1 is


begin
process(i0,i1,i2,i3,sel)
begin
case selis
when "00" => y <= i0;
when "01" => y <= i1;
when "10" => y <= i2;
when others => y <= i3;
end case;
end process;

end Behavioral;
```

	b)	**Write VHDL code for full adder. Draw the neat diagram.**	4M
	Ans:		Entity-1M Architecture-2M Diagram-1M

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

entity Full_Adder is
   Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          CIN : in  STD_LOGIC;
          S : out  STD_LOGIC;
          C : out  STD_LOGIC);
end Full_Adder;
```

```
architecture Behavioral of Full_Adder is

begin

        S <= A XOR B XOR CIN ;

        C <= (A AND B) OR (( A XOR B) AND CIN) ;

end Behavioral;
```

Diagram-



c) Differentiate software and hardware description language. **4M**

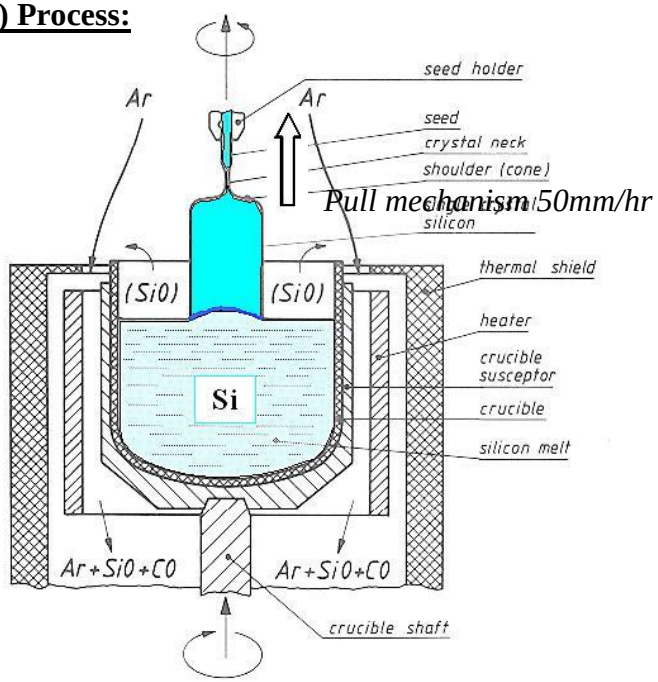
Ans: **Any 4 point- 1M each**

Sr. No.	Software Language	Hardware Descripting Language
1	It is High Level Language	It is used for implementing hardware circuit.
2	It handles sequential instruction.	VHDL allows both sequential and concurrent executions.
3	It can be written with oure logical or arithmetic thinking.	VHDL programmer needs knowledge of Hardware circuit.
4	These programs ran on computer with powerful processor with high speed so easy to implement image processing algorithm.	Difficult to implement image processing algorithm in VHDL
5	In a software language, all assignments are sequential. That means the order in which the statements appear is significant because they are executed in that way.	The events (change in value) in hardware are concurrent, and they must be represented in that way.
6	A software language cannot be used to describe hardware and so a hardware language is required.	A hardware language is used to describe the hardware.
7	In software language, the statements are evaluated sequentially.	In VHDL, concurrent statements are defined to take care of concurrency hardware.

8	We get different results when the order is changed.	The HDL is always concurrent.	

d) Explain CZ method for water processing with neat diagram. **4M**

Ans: **Czochralski (CZ) Process:** **2M**



Explaination-

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. 1425^0C), the atmosphere just above the polysilicon melt is typically helium or orgon for freezing. **2M**

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.

e) Compare between asynchronous and synchronous sequential circuit. **4M**

Ans: **Any 4 point- 1M each**

Parameter	Asynchronous	Synchronous
Definition	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	Synchronous is wherein all the flip-flops within the counter change state simultaneously. This is because all the flip-flops are clocked simultaneously.

	simultaneously.	
Clock required	It does not use a clock	It uses a clock pulse
o/p affected by	The state of circuit can change immediately when an input change occurs	A change of state occurs only in response to a synchronizing clock pulse.
Memory element	Either latches(unclocked FF) or logic gates	Clocked FF
	These circuits are difficult to design	These circuits are easy to design.
Speed	They are faster	They are slower

f) Explain zero modelling and sensitivity list. **4M**

Ans: **2M each**

<u>Zero Modeling:</u>

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.

In VHDL zero delay circuits and designs that depend on zero delay components can never be build. Simulation deltas are used to order some types of events during simulation. Specifically zero delay events must be ordered to produce consistent results. If they are not properly ordered results can vary between different simulation runs

<u>Sensitivity List:</u>

This list defines the signals that cause the statements inside the process statement to execute whenever one or more elements of the list change value.
Sensitivity list is the list of the signals that will cause the process to execute.
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. Process (CLK, RST)
The process is sensitive to RST and CLK signal i.e. an event on any of these signals will cause the process to resume.