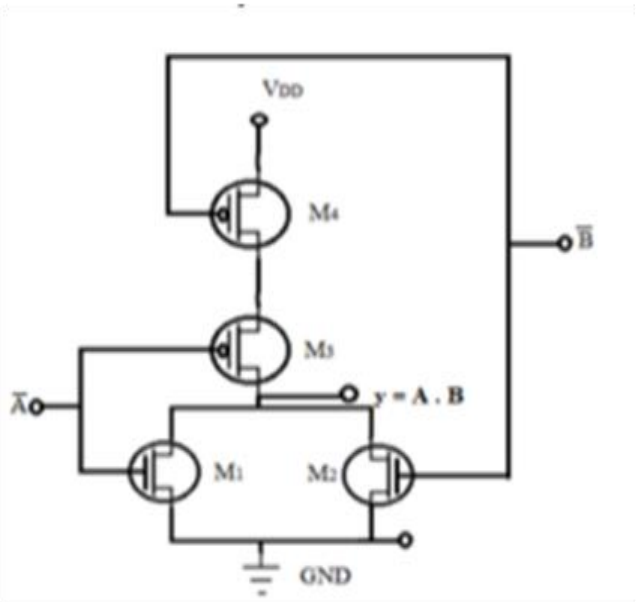
MODEL ANSWER

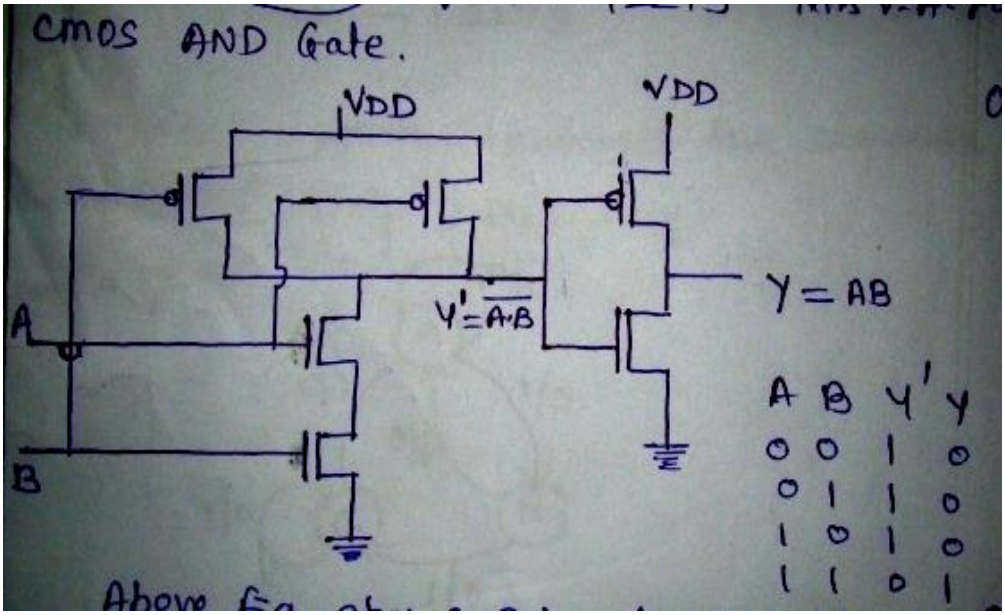
SUMMER– 18 EXAMINATION

Subject Title: VERY LARGE SCALE INTEGRATION **Subject Code:-** **17659**

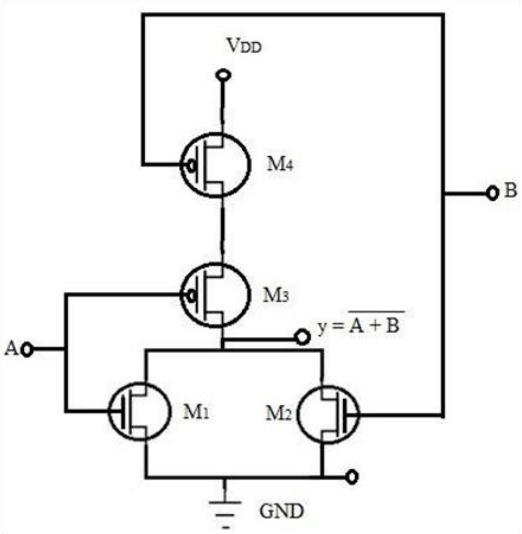
Important Instructions to examiners:

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.

Q. No.	Sub Q.N.	Answer	Marking Scheme
Q.1	A)	Attempt any three:	12-Total Marks
	i)	Draw the AND gate and NOR gate using CMOS.	4 Marks
	Ans:	AND gate using CMOS:  OR	2M each



NOR gate using CMOS:



	ii)	**Write VHDL code for 3-bit up counter.**	**4 Marks**
	Ans:	**Program**: library IEEE; use ieee.std_logic_1164.all; use ieee.std_logic_unsigned.all; entity counter is port (Clock, CLR : in std_logic; Q : out std_logic_vector(2 downto 0)); end counter;	**Entity-2 Marks** **Architecture-2 Marks**

```
architecture behavior of counter is
signal tmp: std_logic_vector (2 downto 0);
begin
process (Clock, CLR)
begin
if (CLR='1') then
tmp <= "000";
elsif (Clock'event and Clock='1') then
tmp <= tmp + 1;
end if;
end process;
Q <= tmp;

end behaviour;
```

Note: Any logic using with-select or case statement or if statement can be used for program.

iii) What is instantiation in VHDL? Write one example. — **4 Marks**

Ans:

Instantiation :

The instantiation means the precompiled entity architecture component is declared in another VHDL entity program. A component instantiated in a structural description is declared using component declaration. A component declaration declares the name and the interface of a component.

2 Marks of instantiation

Example

Note: Any suitable example. NAND gate using AND and NOT

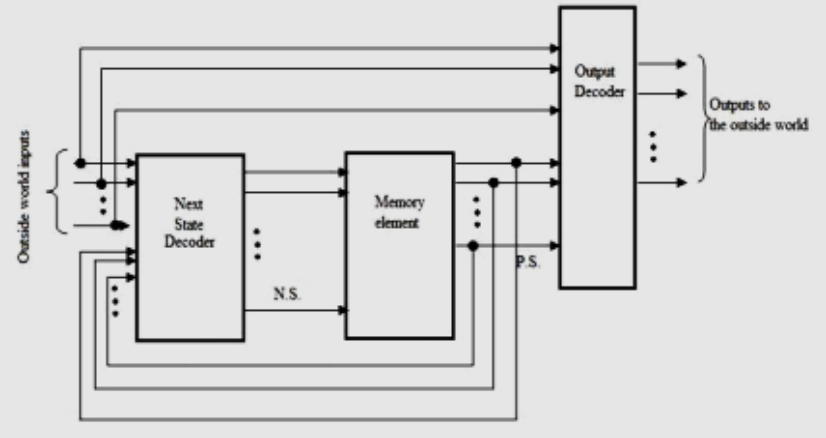
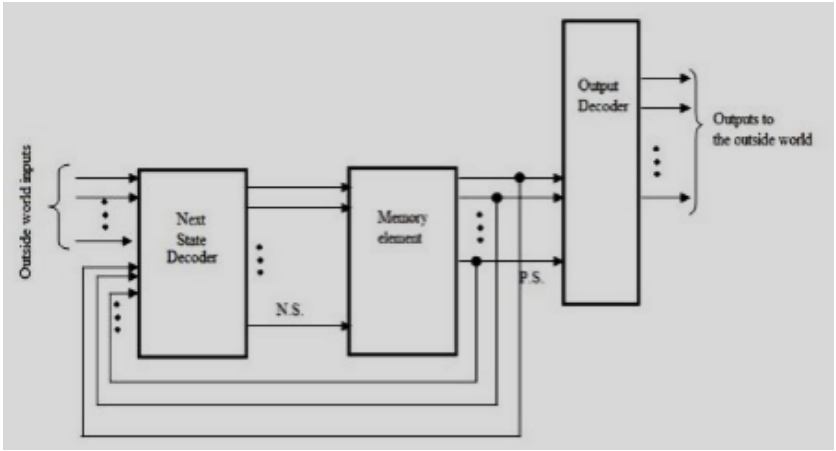
2Marks for example

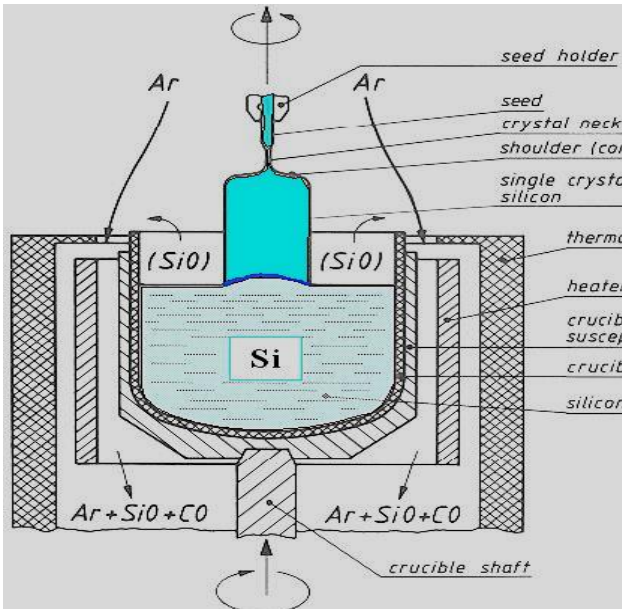
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity INVE is
   Port ( X : in  STD_LOGIC;
          Z : out  STD_LOGIC);
end INVE;

architecture Behavioral of INVE is

begin
 z <= not x;
end Behavioral;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity NANDG is
   Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          y : out  STD_LOGIC);
end NANDG;
architecture Behavioral of NANDG is
 component INVE
  port  ( X : in  STD_LOGIC;
          Z : out  STD_LOGIC);
end component;
 signal w : std_logic;
begin
  w <= a and b;
a0 :INVE
port map ( w, y);
end Behavioral;
```

	iv)	**Draw the diagram of Moore machine and Mealy machine. Write expression for its output.**	**4 Marks**
	Ans:	**Mealy Machine:**  Mealy machine is the sequential system where output depends on present input and state. f (o/p) = f (Input , Present State.) **Moore Machine**  Moore machine is the sequential system where output depends only on present state. f (o/p) = f (Present State.)	**Diagram-1 Marks each Output expression-1 Marks For Moore and Mealy each**
	B)	**Attempt any one:**	**8 Marks**
	a)	**Write any six features of spartan-3.**	**8 Marks**
	Ans:	1. Low-cost, high-performance logic solution for high-volume, consumer oriented applications 2. Densities up to 74,880 logic cells 3. Select IO interface signaling 4. Up to 633 I/O pins 5. 622+ Mb/s data transfer rate per I/O 6. 18 single-ended signal standards 7. 8 differential I/O standards including LVDS, RSDS 8. Termination by Digitally Controlled Impedance 9. Signal swing ranging from 1.14V to 3.465V	**Any six - 1.25Marks to each and final marks are rounded to next integer**

		10. Double Data Rate (DDR) support 11. DDR, DDR2 SDRAM support up to 333 Mb/s 12. Logic resources Abundant logic cells with shift register capability 13. Wide, fast multiplexers 14. Fast look-ahead carry logic 15. Dedicated 18 x 18 multipliers 16. JTAG logic compatible with IEEE 1149.1/1532 17. Select RAM hierarchical memory 18. Up to 1,872 Kbits of total block RAM 19. Up to 520 Kbits of total distributed RAM 20. Digital Clock Manager (up to four DCMs) 21. Clock skew elimination 22. Frequency synthesis 23. High resolution phase shifting 24. Eight global clock lines and abundant routing.	
	b)	**Draw the diagram of Cz process for water fabrication. List the steps involved in water fabrication.**	**8 Marks**
	Ans:	**<u>Czochralski (CZ) Process:</u>** *Pull* ⇧ *mechanism 50mm/hr*  The fabrication of wafer consists basically of the following process steps: ▪ Wafer Processing ▪ Oxidation	**Diagram-4M** **Steps-4M**

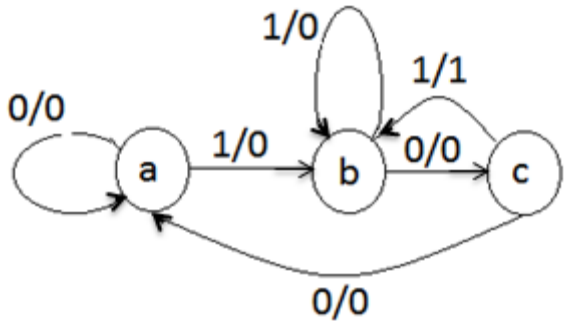
- Epitaxy
- Diffusion
- Ion Implantation
- Lithography
- Etching
- Deposition
- Metallization

Q 2 **A)** **Attempt any four of the following:** **16Marks**

a) **Design a mealy sequence detector circuit for detecting sequence 101 using J -K Flipflop.** **4 Marks**

Ans:

Step 1: State Diagram-



State Diagram-1 Marks

Step 2 State

Excitation table

OR

Table-

Present state	Next state		Output (Z)	
	X=0	X=1	X=0	X=1
(00)a	a	b	0	0
(01)b	c	b	0	0
(10)c	a	b	0	1

state table-1 Marks

Circuit diagram-2 Marks

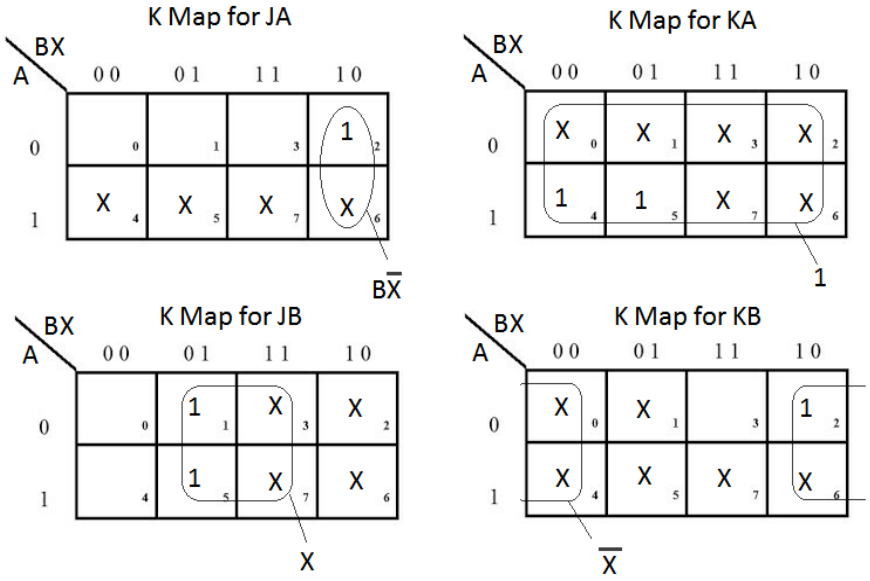
Step 3 Modified Sate Table-

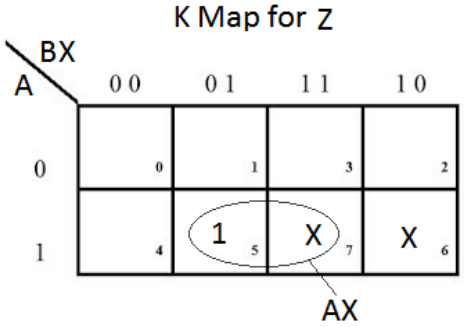
Present state		Input	Next State		Output
A	B	X	A+	B+	Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	0	1	1

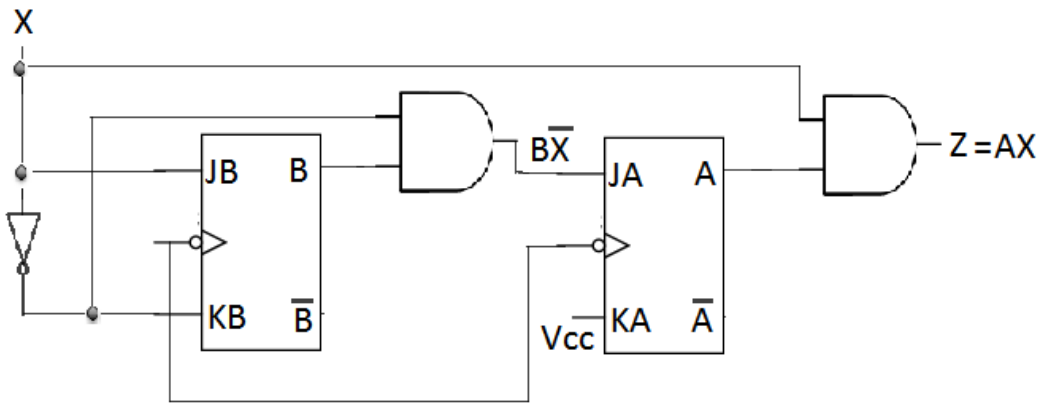
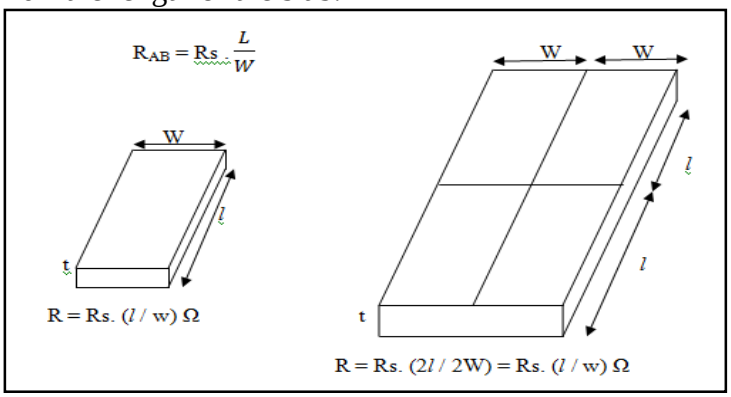
Step 4 Excitation Table-

Present State		Input	Next State		Flip Flop Input				Output
A	B	X	A+	B+	JA	KA	JB	KB	Z
0	0	0	0	0	0	X	0	X	0
0	0	1	0	1	0	X	1	X	0
0	1	0	1	0	1	X	X	1	0
0	1	1	0	1	0	X	X	0	0
1	0	0	0	0	X	1	0	X	0
1	0	1	0	1	X	1	1	X	1
1	1	0	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X

Step 5 K Map-





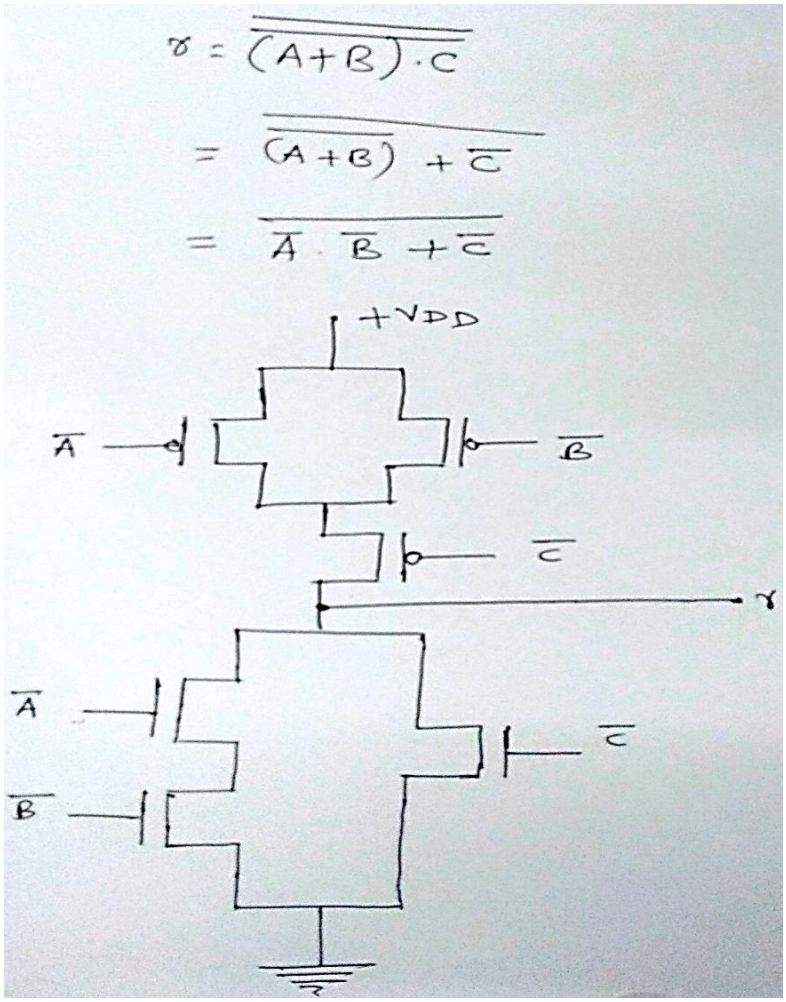
		Step 6 Circuit Diagram- 	
	b)	**Explain how to estimate the channel resistance of CMOS transistor.**	**4 Marks**
	Ans:	Consider a uniform slab of conducting material of resistivity ρ. Let W be the width, t the thickness and Lthe length of the slab.  (Hence, the resistance between A and B terminal is found as, RAB =ρ.L/A ohms. Where A = cross-sectional area. Thus RAB =ρ.L/t.W ohms. …… …… Consider the case in which L = W, that is a square of resistive material then RAB =ρ/t = Rs Where Rs =ρ/t ohm per square or sheet resistance Therefore, Rs = ohm per square Hence Rs is completely independent of the area of the square. ----- ----- ----- ----- ---- Thus to obtain the resistance of a conductor on a layer multiply the sheet resistance Rs, by the ratio of length to width of the conductor as example shown in above diagram	**Diagram 1 Marks,** **1 Marks** **1 Marks** **1Marks for example**
	c)	**Compare BJT and CMOS technology.**	**4 Marks**
	Ans:		

			Any four point 1M each

Sr.	CMOS Technology	Bipolar Technology
1.	Low static power dissipation	High power dissipation
2.	High input impedance	Low input impedance
3.	High packing density	Low packing density
4.	High delay sensitive to load	Low delay sensitive to load
5.	Low output drive current	High output drive current
6.	Bidirectional capability	Essentially unidirectional
7.	It is an ideal switching device.	It is not an ideal switching device.
8.	Voltage driven	Current driven
9.	High power application	Low power application
10.	Unipolar device	Bipolar Device
11.	High current gain	Low current gain
12.	It has less fan out	It has more fan out.

d)	**Write the syntax of entity and architecture in VH-IDL programming.**	**4 Marks**
Ans:	**Entity –** Syntax entity entity_name is Port declaration; end entity_name; **Architecture** Syntax: architecture architecture_name of entity_name is architecture_declarative_part; begin Statements; end architecture_name;	**Entity 2Marks Architecture 2Marks**

	e)	**Explain the sharing of complex operators.**	**4 Marks**
	Ans:	**Sharing of Complex Operators:** • Description of component structure decides efficiency of a synthesized design. • Optimization of individual components made from random logic produces similar results from two very different descriptions. • Concentrate the majority of design effort on the implied component hierarchy rather than on the logical descriptions. • This reduces the gate count and critical path for delay. •	**Each Point 1 Marks**
	f)	**Write the VHDL code for full adder.**	**4 Marks**
	Ans:	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; SUM : out std_logic; CARRY : out std_logic); end Full_Adder; architecture behavioral of Full_Adder is begin SUM <= A XOR B XOR CIN; CARRY<=(A AND B) OR (A AND CIN) OR (B AND CIN); end behavioral; **Note: Anysutable logic using.**	**Declaration 1Marks Entity 1Marks Architectur e 2Marks**
Q. 3		**Attempt any four of the following:**	**16 Marks**
	a)	**State the any four features of VHDL.**	**4 Marks**
	Ans:	• VHDL has powerful construct • VHDL is a Hardware Description language used for design entry and simulation of digital circuits. VHDL is an event-driven language; that is, whenever an event occurs on signals in VHDL it triggers the execution of the statement. • VHDL is technology platform- independent language and portable • VHDL allows both concurrent as well as sequential modeling • It includes advance features of configuration • It is a case sensitive language • VHDL is also said to be strongly typed language • VHDL supports design library	**Any four Features, 1 Mark each**

		• It can handle asynchronous as well as synchronous sequential circuits. • 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.	
	b)	**Design the Boolean expression r = (A + B).C using CMOS logic.**	**4 Marks**
	Ans:		**1Mark** **Design 3Marks**
	c)	**Compare synchronous and asynchronous sequential circuits. (any four points).**	**4 Marks**
	Ans:		

		<table><tr><th>Parameter</th><th>Asynchronous</th><th>Synchronous</th></tr><tr><td>Definition</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><tr><td>Clock required</td><td>It does not use a clock for all flip flop. Only one flip flop is clocked</td><td>It uses a clock pulse</td></tr><tr><td>o/p affected by</td><td>The state of circuit can change immediately when an input change occurs</td><td>A change of state occurs only in response to a synchronizing clock pulse.</td></tr><tr><td>Memory element</td><td>Either latches(unlocked FF) or logic gates</td><td>Clocked FF</td></tr><tr><td></td><td>These circuits are difficult to design</td><td>These circuits are easy to design.</td></tr><tr><td>Speed</td><td>They are slower</td><td>They are faster</td></tr></table>	**Any four point 1Mark each**
	d)	**Write the VHDL code for 3:8 decoder.**	**4 Marks**

	Ans:	Library IEEE ;
		Use IEEE .std _logic_1164.all ;
		
		Entity decoder is
		Port (a,b,c: in std _logic;
		Stb :in std_logic ;
		Y : out std_logic _vector(7 downto 0));
		End decoder;
		
		architecture behavior of decoder is
		signal temp : std_ logic_vector(3 downto 0);
		begin
		temp <=stb& a & b &c ;
		Y <=”01111111” when temp =” 0000” else
			Entity 2 Marks
			
			Architecture 2 Marks

		"10111111" when temp = "0001" else "11011111" when temp =" 0010" else "11101111" when temp = "0011" else "11110111" when temp =" 0100" else "11111011" when temp = "0101" else "11111101" when temp =" 0110" else "11111110" ; end behavior; **Note: Any logic using with-select or case statement or if statement can be used for program.**	
	e)	**Explain efficient coding styles.**	**4 Marks**
	Ans:	**Efficient Coding Styles:-** A coding style is set of rules that a programmer uses for choosing an expressive form to use in the given situation. • 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.	**Each rule carries 1 Mark**
	f)	**Compare FPGA and CPLD.**	**4 Marks**
	Ans:		**Any four points 1Mark each**

Sr. No.	FPGA	CPLD
1	It is field programmable gate arrays.	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 devices.
4	Numbers of input and output pins on FPGA are less than CPLD.	Numbers of input and output pins on CPLD are high.
5	FPGA is suitable for designs with large number of simple blocks with few numbers of inputs.	CPLD are ideal for complex blocks with large number of inputs.
6	FPGA based designs require more board space and layout complexity is more.	CPLD based designs need less board space and less board layout complexity.
7	It is difficult to predict the speed performance of design.	It is easier to predict speed performance of design.
8.	FPGA are available in wide density range.	CPLDs contain fewer registers but have better performance.

Q. 4		**Attempt any four of the following:**	**16 Marks**
	a)	**Write two advantages and disadvantages of VHDL.**	**4 Marks**
	Ans:	**Advantages:** 1. It allows the behaviour of the required system to be described (modelled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires). 2. VHDL allows the description of a concurrent system. 3. VHDL is a dataflow language, unlike procedural computing languages such as BASIC, C, and assembly code, which all run sequentially, one instruction at a time. 4. A VHDL project is multipurpose. Being created once, a calculation block can be used in many other projects. However, many formational and functional block parameters can be tuned (capacity parameters, memory size, element base, block composition and interconnection structure). 5. A VHDL project is portable. Being created for one element base, a computing device project can be ported on another element base, 6. Standard language 7. Fully expressive language 8. Hierarchical 9. Configurable 10. Tool availability 11.	**Any two advantages 2 Marks** **Any two disadvantanges 2 Marks**

		Dis Advantages: 1. Synthesis results of VHDL may vary from one tool to another. 2. Most synthesis tools allow the designer use synthesis directives to some level of control over implementation to make area efficient version speed efficient implementation choices. 3. It has poor implementation due to results of inefficient code which can result in slow execution times or poor memory utilization. 4. Synthesis tools allow designers to specify technology specific gate level implementation, but descriptions of these types are neither high level nor device independent.	
	b)	**Define the following terms related to fabrication process.** **1) Oxidation** **ii) Ion-implantation** **iii) Diffusion** **iv) Deposition.**	**4 Marks**
	Ans:	**1) Oxidation:** Oxidation is a process which converts silicon on the wafer into silicon dioxide. **ii) Ion-implantation:** Ion implantation is the dominant technique to introduce dopant impurities into crystalline silicon. **iii) Diffusion:** Diffusion is the movement or addition of impurity atoms in a silicon substrate at high temperatures. **iv) Deposition:** A multitude of layers of different materials have to be deposited during the IC fabrication process. This the evaporating dopant material on to the silicon surface followed by a thermal cycle which is used to drive the impurities from the surface of silicon into bulk.	**Each 1 mark**
	c)	**Write the VHDL code for D-flip-flop.**	**4 Marks**

	Ans:	library IEEE;
		use IEEE.STD_LOGIC_1164.all;
		entity d_flip_flop is
		port(
		din : in STD_LOGIC;
		clk : in STD_LOGIC;
		reset : in STD_LOGIC;
		dout : out STD_LOGIC
		);
		end d_flip_flop;
		architecture d_flip_flop_arc of d_flip_flop is
		begin
		dff : process (din,clk,reset) is
		begin
		if (reset='1') then
		dout<= '0';
			Entity 2 Marks Architecture 2Marks

```
        elsif (rising_edge (clk)) then
            dout<= din;
        end if;
    end process dff;
end d_flip_flop_arc;
```

d) **Design parity generator using moore machine.** **4 Marks**

Ans: **Condition:** For every three bits that are observed on the input w during the three consecutive clock cycles, the FSM generates the parity bit P=1 if and only if numbers of ones in the three bit sequence is odd.

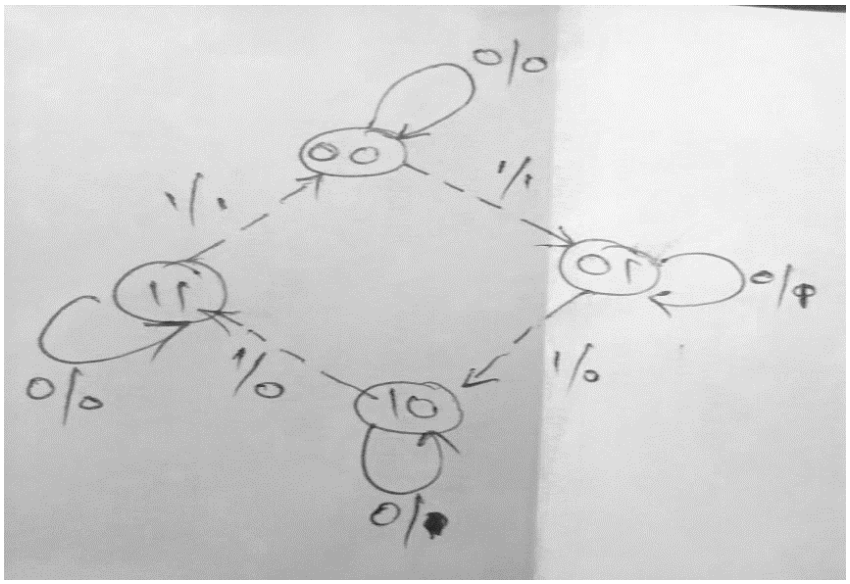
Step 1 1 Marks

Step 2 2 Marks

Step 3 1 Marks

Step 1: **Write the state table :**

Input	Present state		Next state		F F inputs				Output
w	Q_B	Q_A	Q_{B+1}	Q_{A+1}	J_B	K_B	J_A	K_A	P
0	0	0	0	0	0	×	0	×	0
0	0	1	0	1	0	×	×	0	1
0	1	0	1	0	×	0	0	×	1
0	1	1	1	1	×	0	×	0	0
1	0	0	0	1	0	×	1	×	1
1	0	1	1	0	1	×	×	1	0
1	1	0	1	1	×	0	1	×	0
1	1	1	0	0	×	1	×	1	1



Step 2 : K-maps and simplification :

For J_A

w \ $Q_B Q_A$	00	01	11	10
0	0	X	X	0
1	1	X	X	1

$J_A = w$

For K_A

w \ $Q_B Q_A$	00	01	11	10
0	X	0	0	X
1	X	1	1	X

$K_A = w$

For J_B

w \ $Q_B Q_A$	00	01	11	10
0	0	0	X	X
1	0	1	X	X

$J_B = wQ_A$

For K_B

w \ $Q_B Q_A$	00	01	11	10
0	X	X	0	0
1	X	X	1	0

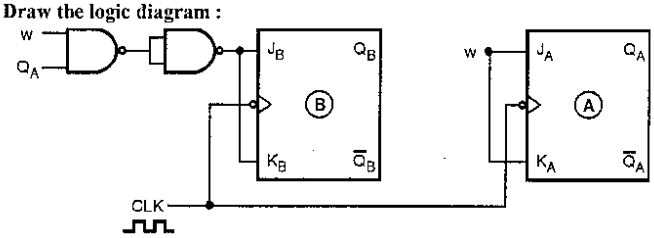
$K_B = wQ_A$

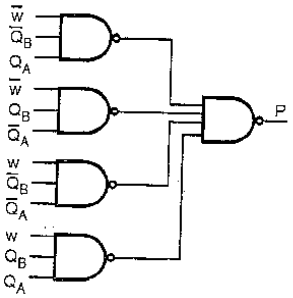
For output P

w \ $Q_B Q_A$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$P = w\bar{Q}_B\bar{Q}_A + \bar{w}\bar{Q}_B Q_A + \bar{w} Q_B \bar{Q}_A + w Q_B Q_A$$
$$= w(\bar{Q}_B\bar{Q}_A + Q_B Q_A) + \bar{w}(\bar{Q}_B Q_A + Q_B \bar{Q}_A)$$
$$= w(Q_B \odot Q_A) + \bar{w}(Q_B \oplus Q_A)$$
$$= w\overline{(Q_B \oplus Q_A)} + \bar{w}(Q_B \oplus Q_A)$$
$$= w \oplus Q_B \oplus Q_A$$

Step 3 : Draw the logic diagram :



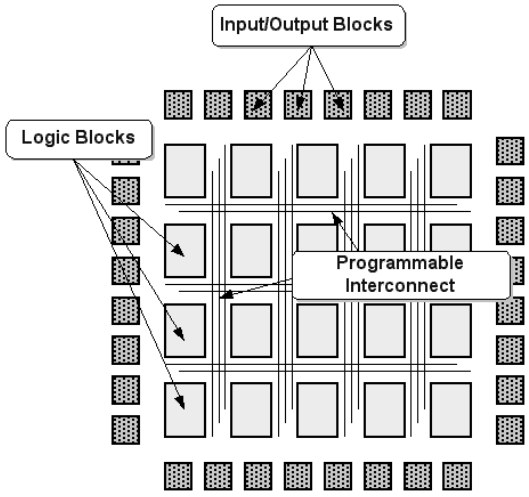


e)	**Compare hardware and software description language.**	**4 Marks**
Ans:		**1mark each**

Sr No.	Software Language	Hardware Description Language
1	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.
2.	A software language cannot be used to describe the hardware and so a hardware language is used.	A hardware language is used to describe the hardware.
3.	In software language, the statements are evaluated sequentially.	In VHDL, concurrent statements are defined to take care of concurrency.
4.	Different results are obtained if the order is changed.	The HDL is always concurrent.

f) Draw FPGA's configurable logic block diagram and write the function of it. — **4 Marks**

Ans: — **Diagram 2 Marks** — **Function 2 Marks**



FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together" – somewhat like many logic gates that can be inter-wired in different configurations.
Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

Q.5 — **Attempt any four of the following:** — **16 Marks**

a) State the applications of test bench and write down the typical format of test bench. — **4 Marks**

Ans: — **Application 2Marks**

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.
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.

Format 2Marks

A typical test bench format is

```
entity TEST_ BENCH is
end;
architecture TB_ BEHAVIOR of TEST_ BENCH is
component ENTITY_ UNDER_TEST
port (list- of- ports-their-types-and-modes);
end component;
Local-signal-declarations;
begin
Generate-waveforms-using-behavioral-constructs;
Apply- to-entity-under-test;
EUT: ENTITY_UNDER_TEST port map ( port-associations) ;
Monitor-values-and-compare-with-expected-values;
end TB_ BEHAVIOR;
```

b) Design 2-bit sequential counter using mealy machine. — **4 Marks**

Ans:

State Diagram

State Table:

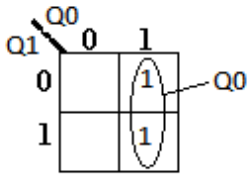
Count	Previous State		Next State		Excitation	
	Q_1	Q_0	Q_1^*	Q_0^*	T_1	T_0
0	0	0	0	1	0	1
1	0	1	1	0	1	1
2	1	0	1	1	0	1
3	1	1	0	0	1	1

Table 1Marks

Kamp:

Kamp 2Marks

Kmap for T1



kmap for T0

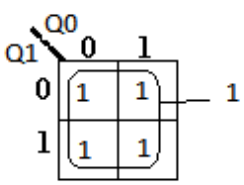
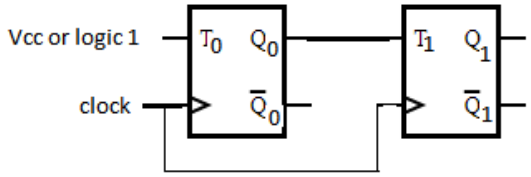


Diagram 1Marks

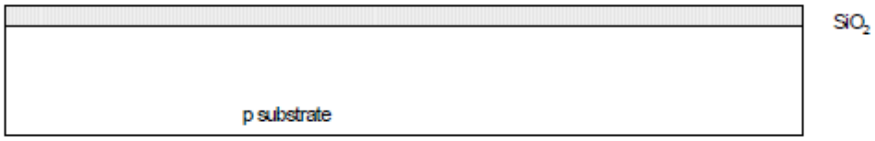
Circuit Diagram:



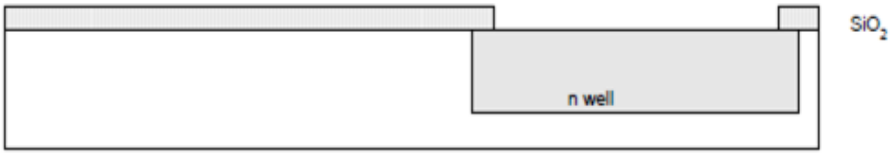
c) Explain n-wel1 CMOS fabrication process with neat sketches. **4 Marks**

Ans: The fabrication steps are as follows:

- Thick SiO2 layer is grown on p-type silicon wafer.

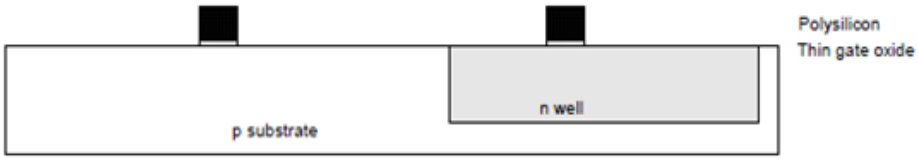


- After defining the area for N-well diffusion, using a mask, the SiO2 layer is etched off and n-well diffusion process is carried out.



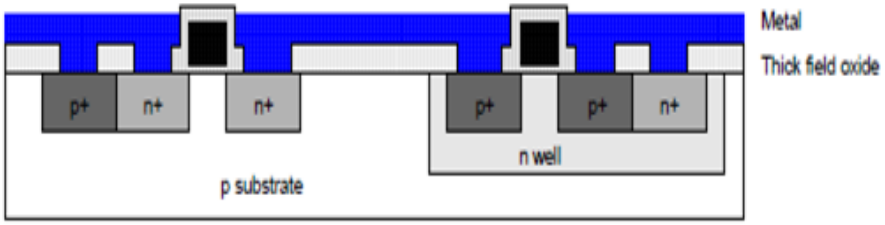
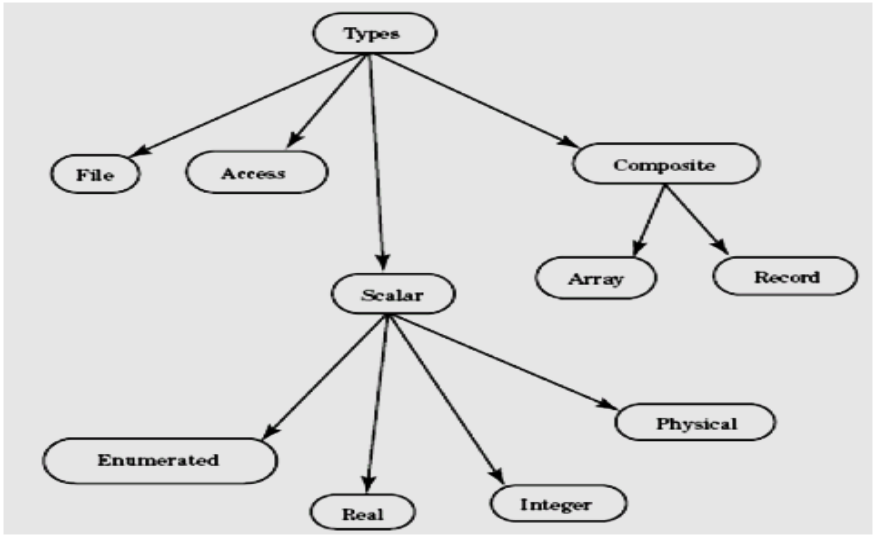
- Oxide in the n transistor region is removed and thin oxide layer is grown all over the surface to insulate gate and substrate.

- 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.

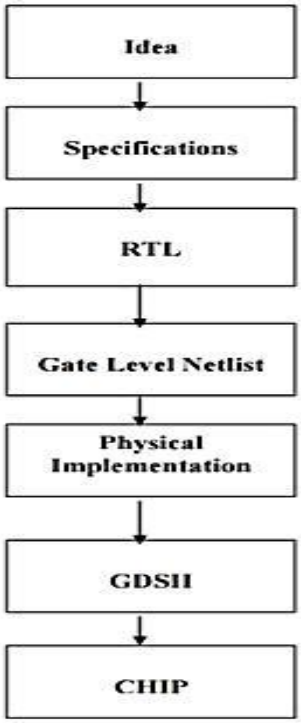


- Using n+ mask and complementary n+ mask, source and drain of both nMOS and pMOS transistors are formed one after another using respective diffusion processes. These same masks also include the VDD and VSS contacts.

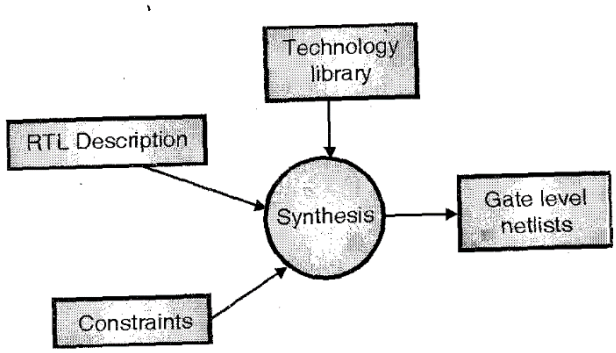
Steps 2Marks
Neat sketches 2Marks

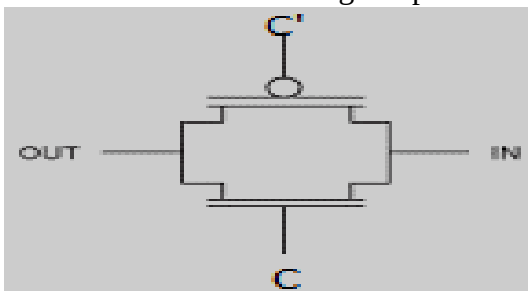
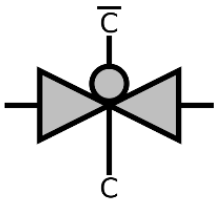
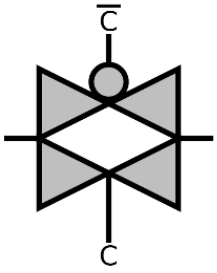
		• The contacts are made using proper masking procedure and metal is deposited and patterned on the entire chip surface. • An overall passivation layer is formed and the openings for accessing bonding pads are defined. 	
	d)	**List the data types used in HDL and explain.**	**4 Marks**
	Ans:	 There are mainly two types: 1. Scalar 2. Composite Scalar Types : 1. Integer: Defines the value with Integer. (Integer Range). 2. Real: Defines the value with number. 3. Enumerated: Defines the set of user defined values consisting of identifiers and character literals 4. Physical :used to represent physical quantities e.g. distance ,time 5. Composite : • Array: Contain many elements of same type. • Record : Contain elements of different types	**1 Mark for List, 1.5 Marks for Scalar and 1.5 Marks for Composite**
	e)	**Explain event scheduling.**	**4 Marks**
	Ans:	**Event Scheduling:** The mechanism for delaying the new value is called scheduling an event. e.g.	**2Marks for event scheduling 2Marks for**

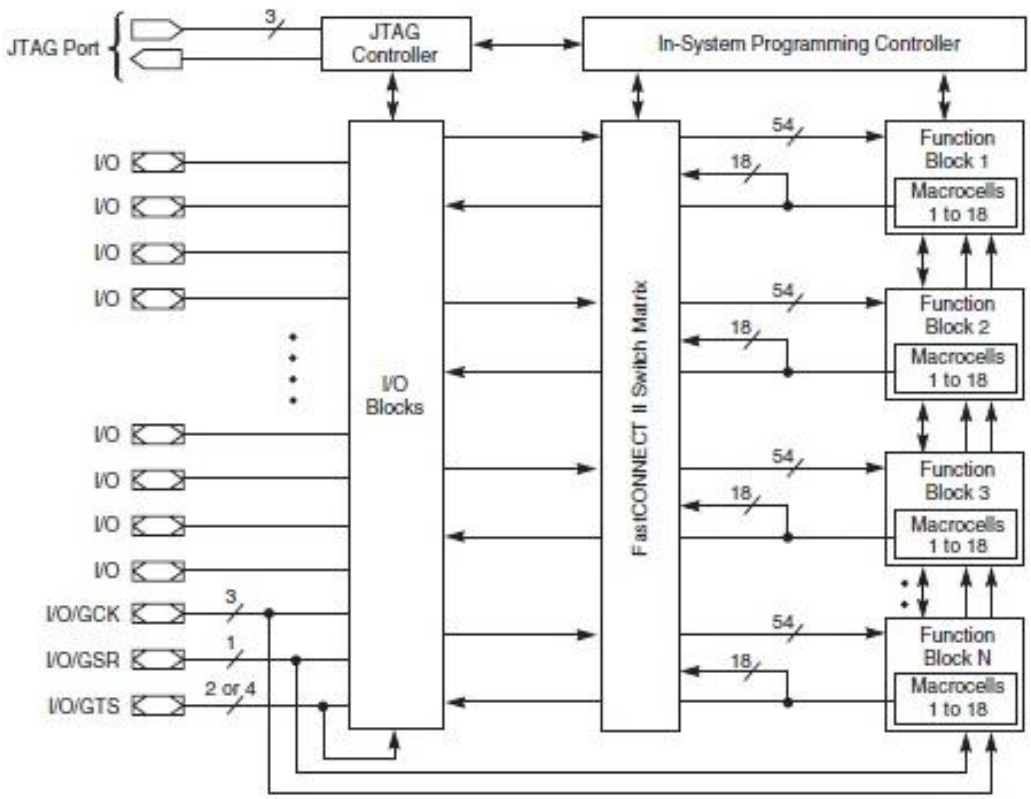
		X<= a after 0.5ns when select=0 else
		X<= b after 0.5ns
		The assignment to signal x does not happen instantly. Each of the values assigned to x contain an after clause.
		By assigning port x a new value, an event was scheduled 0.5ns in the future that contains the new value for signal x. when the event matures (0.5 nanoseconds in the future), signal receives a new value.
		`ARCHITECTURE dataflow OF mux IS`
		`  SIGNAL select : INTEGER;`
		`BEGIN`
		`  select <= 0 WHEN s0 = '0' AND s1 = '0' ELSE`
		`  1 WHEN s0 = '1' AND s1 = '0' ELSE`
		`  2 WHEN s0 = '0' AND s1 = '1' ELSE`
		`  3;`
		`  x <= a AFTER 0.5 NS WHEN select = 0 ELSE`
		`  b AFTER 0.5 NS WHEN select = 1 ELSE`
		`  c AFTER 0.5 NS WHEN select = 2 ELSE`
		`  d AFTER 0.5 NS;`
		`END dataflow;`
			example or justification

	f)	**Draw the design flow of ASIC and explain.**	**4 Marks**
	Ans:	**ASIC DESIGN FLOW:** **Note: Any other relevant diagram and explanation can be consider.** 	**Design flow diagram 2Marks Explaination 2Marks**

		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 Physical Implementation of the Gate Level Netlist. The Gate level Netlist is converted into geometric representation. The geometric representation is nothing but the layout of the design. The layout is designed according to guidelines based on the limitations of the fabrication 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.	
Q.6		**Attempt any four of the following:**	**16 Marks**
	a)	**Write the VHDL code for 3-bit right shift register.**	**4 Marks**
	Ans:	**Library IEEE;** Use IEEE, Std logic_1164.all; **Entity shift_siso is** port (CLK, Sin: in std_logic); Sout: out std_logic); **End shift_siso;**	**Entity 2 Marks** **Architectur e 2Marks**

		Architecture **behave** of **shift_siso**is **Signal Temp:** Std_logic_vector (**2 down to 0**); **Begin** **Process** (clk) **Begin** **If** (clk`event and clk = 1)**then** **For** i in 0 to 2 loop Temp (i + 1)<= Temp (i); **End** loop; Temp (0) <= Sin; **End** if**;** **End** process; Sout<=Temp (2); **End behave;** **Note any other suitable logic.**	
	b)	**Draw HDL design flow for synthesis and explain.**	**4 Marks**
	Ans:	 • The process that converts user, hardware description into structural logic description. Synthesis is a means of converting HDL into real world hardware. It generates a gate level net list for the target technology. The synthesis tool converts register transfer level (RTL) description to gate level net list. These gate level net lists consist of interconnected gate level macro cells. • The inputs to the synthesis process are RTL (register transfer level) VHDL description, circuit constraints and attributes for the design, and a technology library. • The synthesis process produces an optimized gate level net list from all these inputs. The translation from RTL description to Boolean equivalent description is usually not user controllable. The intermediate form that is generated is a format that is optimized for a particular tool and may not even be viewable by the user. All the conditional signal assignments and selected signal assignment statements are converted to their boolean equivalent in this intermediate form. The optimization process takes an unoptimized Boolean description and converts it to an optimized Boolean description. For this it uses number of algorithm and rules. This process aims to improve structure of Boolean equations by applying	**Design flow 2 Marks. Explanation 2 Marks**

		rules of Boolean algebra. This removes the redundant logic and reduces the area requirement	
	c)	**Explain CMOS transmission gate with neat diagram.**	**4 Marks**
	Ans:	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.    A detailed analysis of working of transmission gates follows: • When input node A is connected to VDD and control logic C is also high, C = 1 : The output node B may be connected to capacitor. Let us say, voltage at output node is Vout. For PMOS, Source of is at higher voltage than drain. For NMOS, drain is at higher voltage than Source terminal. Hence, node A will act as source terminal for pMOS and as drain terminal for nMOS. Drain to Source and gate to source voltages for nMOS are as: $V_{DS,n} = VDD - Vout$ $V_{GS,n} = VDD - Vout$ For nMOS to be turned off, $V_{GS,n} < V_{th,n}$ $VDD - Vout < V_{th,n}$ $Vout > VDD - V_{th,n}$ (Cut off region) For Vout< VDD – $V_{th,n}$ $V_{DS,n} > V_{GS,n} - V_{th,n}$	**Diagram-1 mark,Explanation-3 mark**
	d)	**Draw the architecture of XC950O CPLD.**	**4 Marks**
	Ans:		**Correct labelled diagram-4 marks**



	e)	**Explain event based simulator.**	**4 Marks**
	Ans:	• Event driven signal keeps track Of any change in the signal in the event queue. • The simulator starts simulation as soon as any signal in event list changes its value. • For this the simulator has to keep record of all the scheduled events in future. This causes a large memory overload but gives high accuracy for asynchronous design. It simulates events only. • Gates whose inputs have events are called active and are placed in activity list. • The simulation proceeds by removing a gate from the activity list. The process Of evaluation stops when the activity list becomes empty.	**Each Point carries 1 Marks**

<table>
<tr><td></td><td>f)</td><td colspan="2">Differentiate between Xilinx and Atmel series architecture of CPLD (four points).</td><td>4 Marks</td></tr>
<tr><td></td><td>Ans:</td><td colspan="2"><table>
<tr><th>Sr no.</th><th>Xilinx CPLD</th><th>Atmel CPLD</th></tr>
<tr><td>1.</td><td>XC 9536, XC 9572, XC 95108 these have 36, 72, 108 microcells.</td><td>ATF 1502, ATF 1504, ATF 1508 these
have 32, 64 and 128 micro cells.</td></tr>
<tr><td>2</td><td>Available in variety of pacakages but 44 and 64 pins PLCC of J lead packages are more popular.</td><td>Available in variety of packages but 44, 68
and 84 pins PLCC or J lead packages are more popular.</td></tr>
<tr><td>3.</td><td>Xilinx offers their web packs free downloads.</td><td>ATMEL offers their WinCUPL PLD compiler for free download.</td></tr>
<tr><td>4.</td><td>The most current version only works on windows XP.</td><td>The most current version only works on windows XP.</td></tr>
<tr><td>5.</td><td>In conversion application XC 4000 FPGAs the equivalents are:
• XC 4002, XC 4003
• XC 5200 Series XC 5202 and XC 5204
• XC 505 (Spartan series) XC 10.</td><td>In conversion application atmel series FPGAs equivalent are:
• AT 40K05/10
• AT40K05/20</td></tr>
</table></td><td>Any four points 1 mark each</td></tr>
</table>