Important Instructions to examiners:
1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

1 (A) Attempt any THREE: 12M

(a) Define the terms:

(i) Metastability.
(ii) Noise Margin.
(iii)Power Fanout.
(iv) Skew

Ans: **(each for 1 marks)**

Definition:

(i) Metastability

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.

(ii) 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).

(iii) 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.

(iv)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.

(b) Compare BJT and CMOS (any 4 points)

Ans:

Comparison :(1 marks for each point any 4 points should be considered)

Sr. No.	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

(c) State the use and syntax of:

(i) Signal

(ii) Variable in VHDL

Ans: (1 marks for USE and 1 marks for Syntax)

(i) Signal:

Use: (Any 1 relevant point should be considered)

- Signal objects are used to connect entities together to form models.
- Signals are the means for communication of dynamic data between entities.

Syntax: SIGNAL signal _name: signal_type [:= initial value];

OR

SIGNAL Name(type)

(ii) Variable in VHDL:

Use: (Any relevant point other than this also should be considered)

Variables are used for local storage in process statements and subprograms.

Syntax: VARIABLE variable_name {,variable_name} : variable_type [:=value];

(d) Compare Software and Hardware description languages.

Ans:

Comparison: (1 mark of each point)

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 hardware and so a hardware language is required.	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 hardware.
4.	We get different results when the order is changed.	The HDL is always concurrent.

(e) Write the VHDL program for 3-bit up-counter.

Ans: **(4 marks for correct program)**

Program:

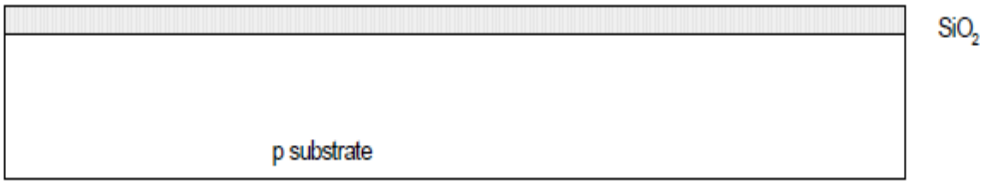
```
library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity counter is
        port(Clock, CLR : in  std_logic;
        Q : out std_logic_vector(2 downto 0));
end counter;
architecture archi of counter is
signal tmp: std_logic_vector(2 downto 0);
begin
process (Clock, CLR)
begin
         if (CLR='1') then
         tmp <= "000";
         elsif (Clock'event and Clock='1') then
         tmp <= tmp + 1;
         end if;
end process;
         Q <= tmp;
end archi;
```

(B) Attempt any ONE: 6M

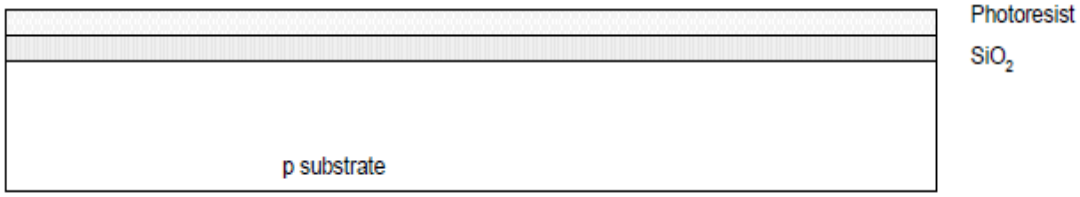
(a) List and explain the main steps carried in typical n-well, CMOS fabrication process with neat sketched.

Ans: (6 Marks for correct answers with neat Diagram)

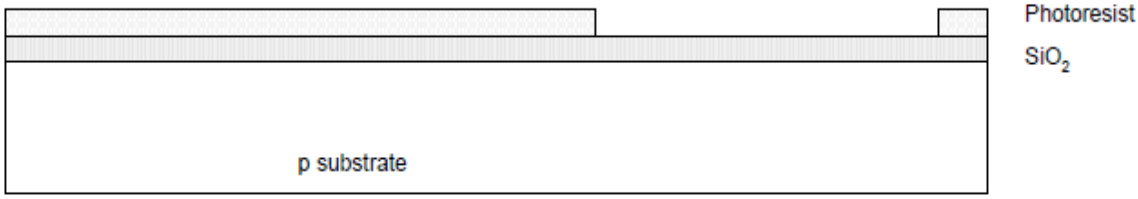
Blank wafer covered with a layer of SiO2 using oxidation



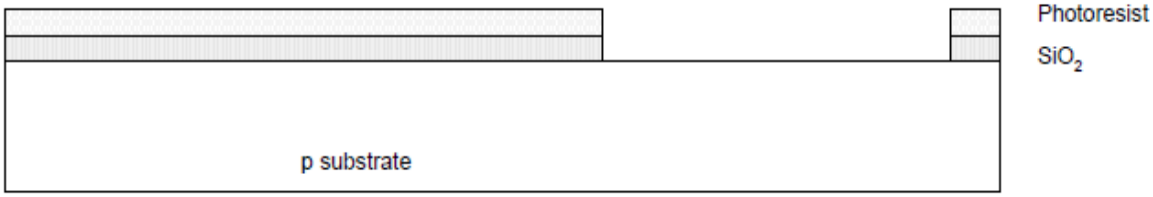
Spin on the photoresist. Exposed to UV light using the n-well mask. (Photolithography)



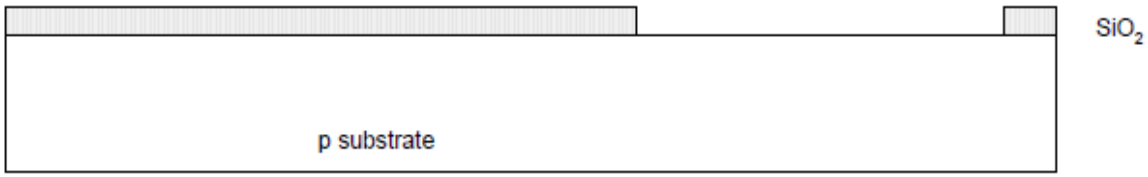
Strip off the exposed photoresist using organic solvents



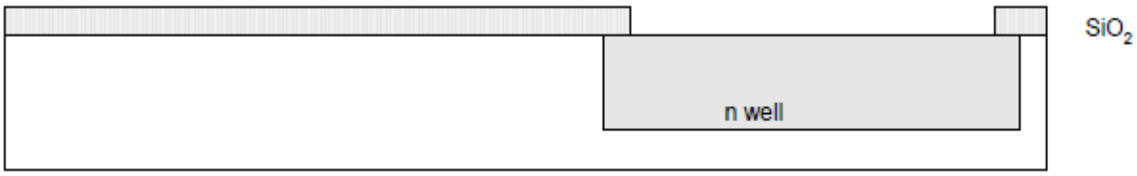
Etch the uncovered oxide using HF (Hydroflouric acid)



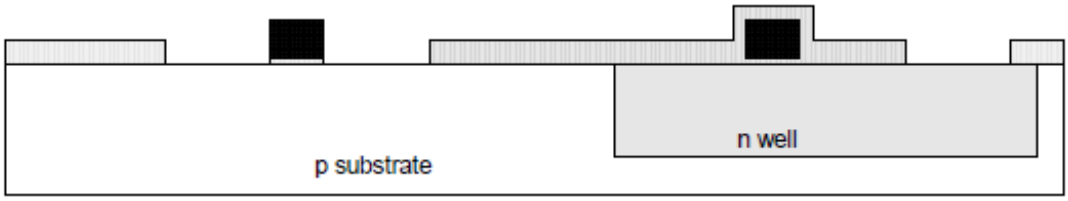
Etch the remaining photoresist using a mixture of acids



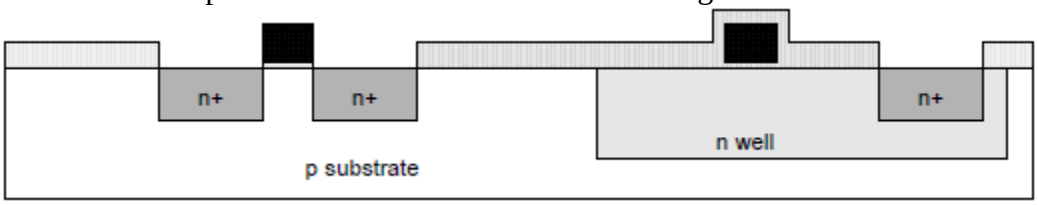
n-well is formed using either diffusion or ion implantation



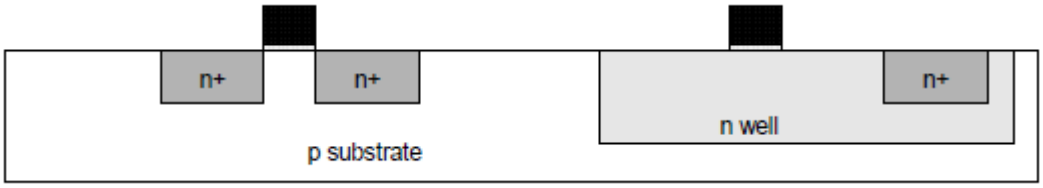
Pattern oxide using n+ active mask to define n diffusion regions



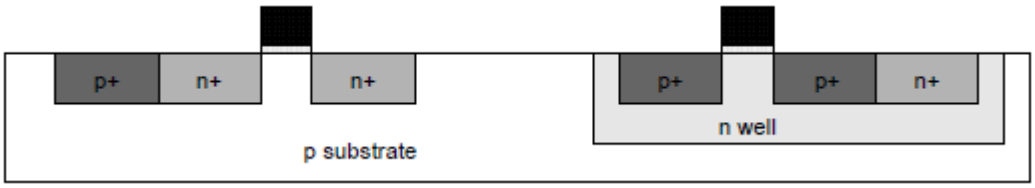
Diffusion or ion implantation used to create n diffusion regions



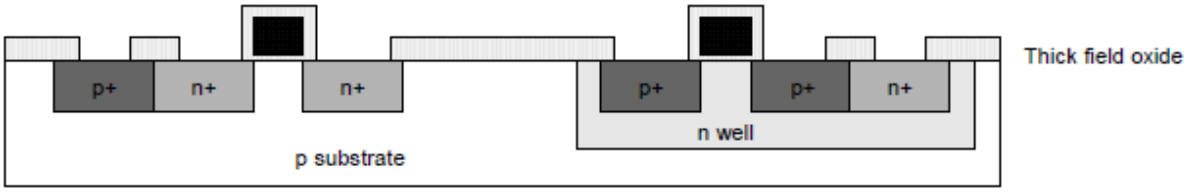
Strip off the oxide to complete patterning step



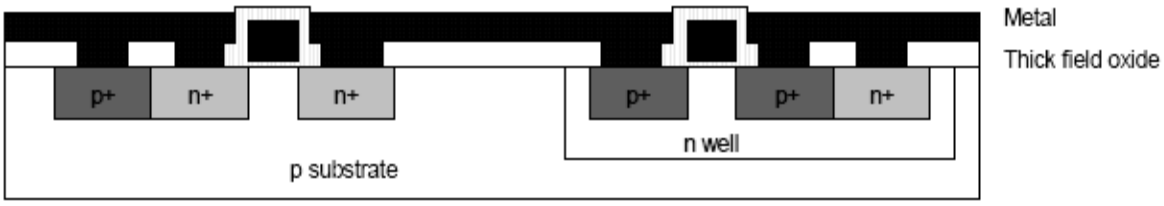
Similar steps used to create p diffusion regions



Cover chip with thick field oxide and etch oxide where contact cuts are needed



Remove excess metal leaving wires



(b) Describe the following statements with example:

(i) assert statement

(ii) wait statement

(iii) case statement

Ans: (2 marks each)

NOTE: (1 marks for Description and 1 marks for eg. Any relevant eg will do & Only the syntax of example is important)

(i) Assert statement:

ASSERT statement is used for exception handling based on self-checking model that reports violations in the design immediately.

E.g. Design of S-R flip-flop

Program:

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
ENTITY rsff IS
PORT(r,s,clk: IN STD_LOGIC;
        q,qbar: OUT STD_LOGIC);
END rsff;
ARCHITECTURE behave OF rsff IS
BEGIN
   P1: PROCESS (clk)
        BEGIN
        ASSERT ( s='1' NAND r='1')
         REPORT "undefined state"
          SEVERITY ERROR;
    IF s='1' THEN
       q<='1';
    ELSE r='1' THEN
       q<='0';
     END IF;
       qbar<= NOT (q);
     END PROCESS P1;
END behaves;
```

(ii) Wait Statement:

Wait Statement is a sequential statement, used to specify timing inputs for simulation or synthesis, or explicitly suspend a process without using the sensitivity list of the process.

E.g. D flip-flop with asynchronous clock and reset**.**

Program:

```
LIBRARY IEEE;
USE IEEE. STD_LOGIC _1164.all;

ENTITY dff  IS
PORT(d, clk,rst: IN STD_ LOGIC;
        q: OUT STD _ LOGIC);
END dff;
ARCHITECTURE behave OF dff IS
BEGIN
        P1: PROCESS
                BEGIN
                  IF rst='0' THEN
                    q <='0';
ELSE clk' EVENT AND clk ='1' THEN
q <=d;
END IF;
WAIT ON clk, rst ;
sEND PROCESS P1;
END behave;
```

(ii) (iii) Case Statement:

A sequence statement selects for execution one of the several alternative sequences is called case statement.

Eg. VHDL program for 4:1 Multiplexer using case statement**.**

Program:

```
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 downto 0);
    y : out std_logic  );
end multiplexer4_1;
architecture Behavioral of multiplexer4_1 is
```

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

Q2. Attempt any FOUR: 16M

(a) Define the terms:

i. Oxidation
ii. Epitaxy
iii. Deposition
iv. Ion- Implantation

Ans: (1 marks each)

Definition:

(i) Oxidation:

Oxidation is a process by which a layer of silicon dioxide is grown on the surface of a silicon wafer. The oxidation of silicon is necessary throughout the modern integrated circuit fabrication process.

(ii) Epitaxy: Single-crystal film grown on silicon surface with controlled impurities that can have fewer defects than native wafer surface.

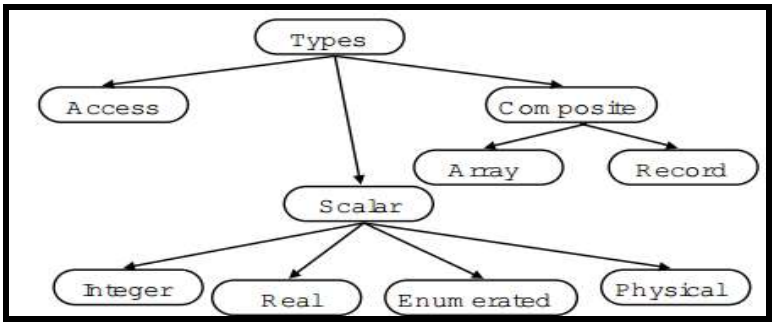
(iii) Deposition: Deposition is a process followed by an implantation step to reduce poly resistance.

(iv) Ion- Implantation: Ion implantation can be defined as the process by which impurity ions are accelerated to high velocity and physically lodged into the target material.

(b) State the Data types used in VHDL.

Ans:

Note: (if listed marks should be given) **4M**



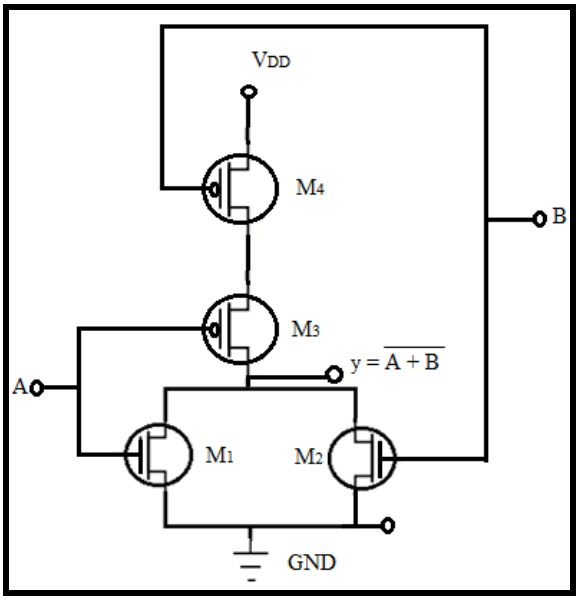
(c) Design the following logic gates using CMOS: (2 marks each)

i. NOR gate

ii. NAND gate with truth table showing ON-OFF action.

Ans:

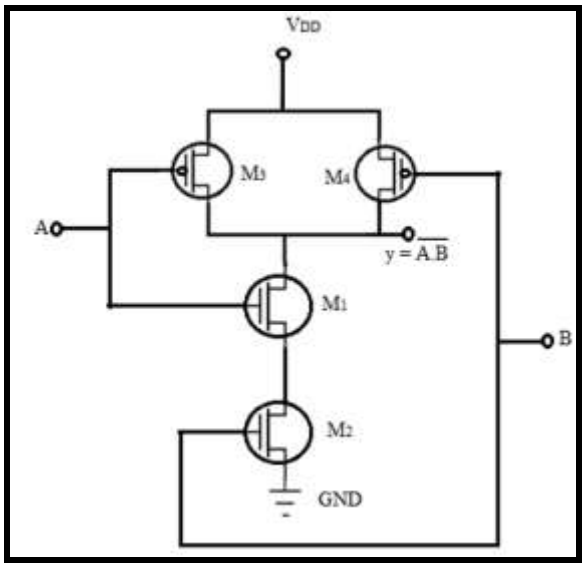
(i) NOR gate: (2 mark for diagram Truth table is optional)



Truth Table:

Input		CMOS				Output
A	**B**	**M1**	**M2**	**M3**	**M4**	
0	0	OFF	OFF	ON	ON	1
0	1	OFF	ON	ON	OFF	0
s1	0	ON	OFF	OFF	ON	0
1	1	ON	OFF	ON	OFF	0

(ii) NAND gate with truth table showing ON-OFF action. (1.5 mark for diagram 0.5 mark for Truth table)



Truth Table:

Input		CMOS				Output
A	**B**	**M1**	**M2**	**M3**	**M4**	
0	0	OFF	OFF	ON	ON	1
0	1	OFF	ON	ON	OFF	1
1	0	ON	OFF	OFF	ON	1
1	1	ON	ON	OFF	OFF	0

(d) Compare concurrent and sequential statement. (Any four points)

Ans:

Comparison: (any four) **4M**

Sr. No.	Concurrent Statement	Sequential Statement
1.	Many of these statements can be active at the same time.	A set of VHDL statements that executes in sequence is called sequential statements.
2.	Inside Architecture	Inside process
3.	Simple signal assignment statement	Sequential signal assignment statement
4.	Conditional signal assignment statement	Variable assignment statement
5.	E.g. Process, Component Instance, concurrent signal assignment.	If, for, switch-case, signal assignment.

(e) State and explain Delta Delay.

Ans:

Explanation:

- The concept of Delta delay is introduced to achieve concurrency and order independence.
- 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.
- Thus, real time and simulation time are different.
- Several logic changes occur simultaneously in a circuit.
- But a simulator cannot process events concurrently. Therefore, time is frozen within the simulator.
- Events are processed and logic values are updated one after the other till no more events takes place. This is known as one simulation cycle.
- The real time that the simulator takes to execute one simulation cycle is known as delta delay for simulation delta with zero simulation time.

Q3) Attempt any FOUR of the following: 16M

a) Explain Moore and mealy m/c with block diagram.

Ans: (1 marks for Explanation & 1M block diagram for moore as well as mealy machine)

Moore machine:

In a Moore machine the output depends on the present state only. Moore machine model is derived from the general basic model shown in the figure above by a process of generation

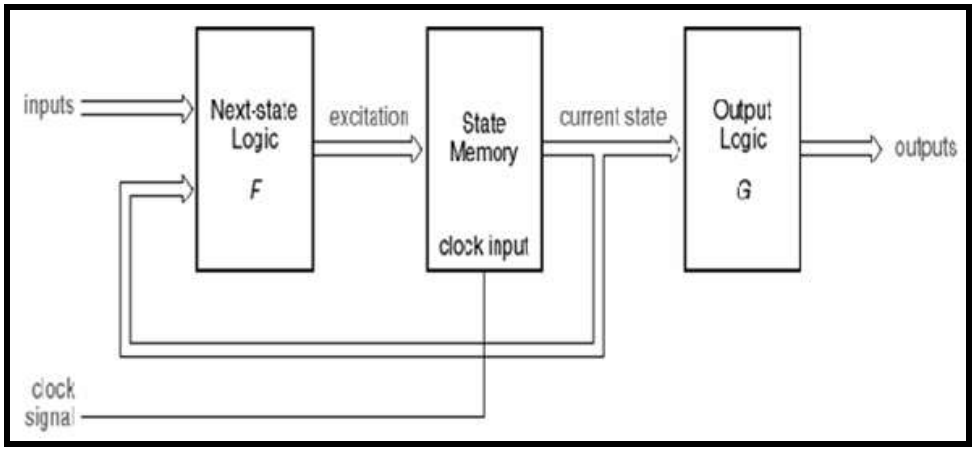


Fig. Moore m/c

Mealy machine:

In a Mealy machine the output depends on the present state and present set of inputs. Mealy machine model is derived from the general basic model shown in the figure below by a process of generation.

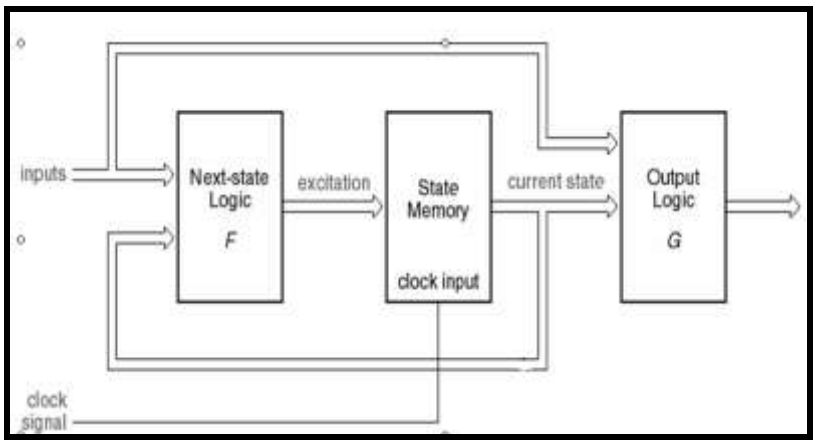


Fig. Mealy m/c

.

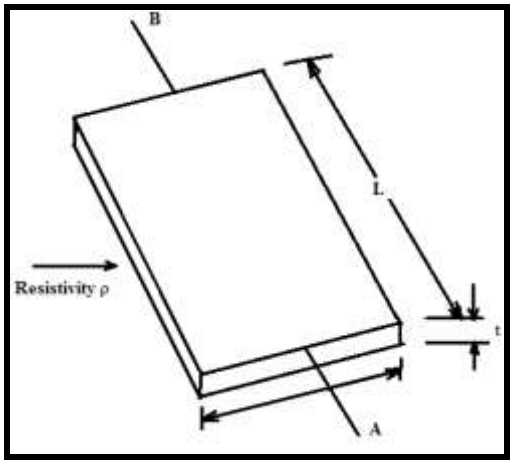
b) Explain process of estimation of channel resistance & how it is calculated?

Ans: (2 marks for explanation & 2 marks for calculation)

A MOS is created by superimposing a number of layers of conducting and insulating materials. It has diffusion polysilicon and metal layers separated by insulating layers. Due to these layers resistances and capacitances are introduced in the circuit, which affects the performance of the circuit. They also have inductance characteristics.

Resistance Estimation:

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.



Hence the resistance between A and B terminal.

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

Where A = cross-sectional area.

Thus $R_{AB} = \rho.L/t.w$ ohms.

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

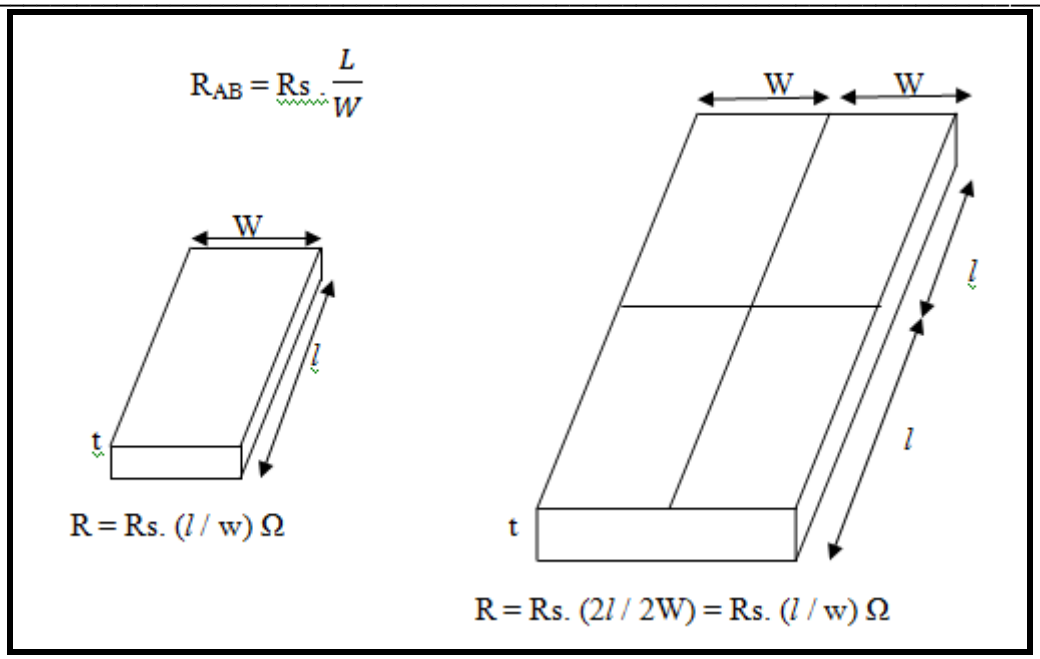
$R_{AB} = \rho/t = Rs$

Where

$Rs = \rho/t$ ohm per square or sheet resistance

Therefore, Rs = 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 non-linear, it is used to approximate its behaviour in terms of a „change resistance" to estimate the performance.

The channel resistance Rc

Rc=K(L/W)

Where K= 1/(μCox(vgs-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 Rc, channel resistance mainly depends on length to width ratio of the channel.

c) Write abbreviation of VLSI & VHDL. What is VHDL.

Ans: (**1 mark each for the full form, 2 marks for explaining what is VHDL)**

VLSI: VLSI stands for VERY LARGE SCALE INTEGRATION.

VHDL: VHDL stands for VHSIC hardware description language and VHSIC is an acronym for very high speed integrated circuit.

VHDL: VHDL is a hardware description language used for modelling digital system mode of interconnection of components.

(Software to design hardware that runs s/w and is controlled by s/w.)

d) Design the Boolean equation using CMOS?

$Y=\overline{A+B}.\overline{A}.\overline{C}$

Ans: (4 marks for correct design)

$Y = \overline{\overline{A+B}} . \overline{A} . \overline{C}$

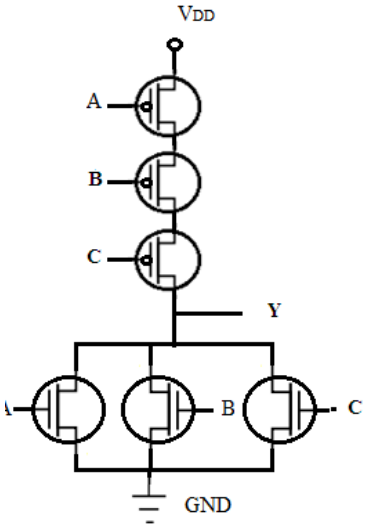
Taking Double bar

$= \overline{\overline{(A+B)} + \overline{A} . \overline{C}}$

$= \overline{(A . B) + (A + C)}$

$= \overline{(A + B) + (A + C)}$

$= \overline{(A + B + C)}$ ..... Since (A+A = A)



e) State & explain coding styles in VHDL.

Ans: (4 marks)- (Any relevant right answer marks should be givens)

Dataflow modelling:
In dataflow modelling, the flow of the data is expressed using concurrent statements for signal assignment.

Structural modelling:
In structural modelling, the architecture is described as a set of interconnected components. That are used in the architecture are described in the declarative part of the architecture.

Behavioural modelling:
Behavioural modelling explains the behaviour of an entity as a group of statements that are executed sequentially in the order in which they are specified. This set of sequential statements has to be specified within a 'process' statement.

Mixed modelling:
A single architecture can have all the three modelling styles that we have studied till now. That is, we can have a combination of dataflow modelling, structural modelling as well as behavioural modelling within a single architecture body.

Q.4 (A) Attempt any THREE: 12M

(a) Explain the term w.r.t VHDL:

(i) Entity

(ii) Architecture (with one example)

Ans:

Explanation:

(i) ENITIY: (2 marks)

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

(ii) ARCHITECTURE: (1 mark for explanation +1 mark for example)

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

Example of an Architecture of AND gate

```
architecture and_arch  of and_gate is
        BEGIN
                y<= a and b;
        END and_arch;
```

(b) Write VHDL program to implement 4:1 Multiplexer using case statement.

Ans: [ENITITY 1 Mark, PROCESS 1 Mark, Architecture 1 mark, case 1 mark]

Program:

4:1 MUX with CASE STATEMEMT

```
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 downto 0);
   y : out std_logic  );
end multiplexer4_1;
architecture Behavioral of multiplexer4_1 is
begin
process(i0,i1,i2,i3,sel)
begin
case sel is
 when "00" => y <= i0;
 when "01" => y <= i1;
 when "10" => y <= i2;
 when others => y <= i3;
end case;
end process;
end Behavioral;
```

(c) Draw HDL design flow for synthesis and explain.

Ans: (marks to be given to the answer covering just major points)

Explanation:

HDL DESIGN FLOW FOR SYNTHESIS [4 marks]

Synthesis = Translation + Optimization.

Synthesis is an automatic method of converting higher level of abstraction to lower level of abstraction. i.e. 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 netlist. These gate level netlists consist of interconnected gate level macrocells.

The inputs to the synthesis process are RTL (Register Transfer Level) VHDL description, circuit constraints and attributes for the design, and a technology library.

Diagram:

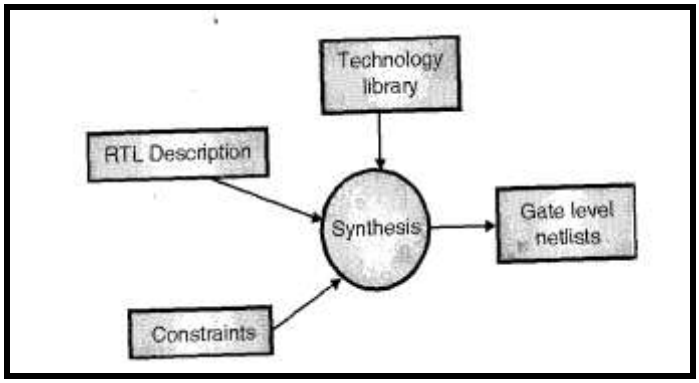


Figure: HDL design flow

The synthesis process produces an optimized gate level netlist from all these inputs. The translation from RTL description to Boolean equivalent description is usually not user controllable.

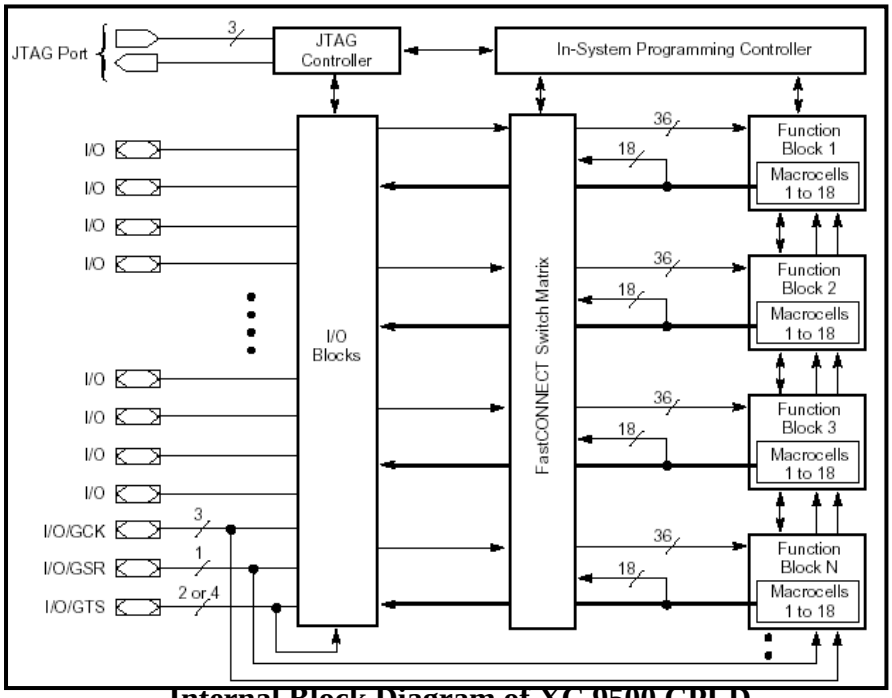
The intermediate form that is generated is a format that is optimized for a particular tool and may not even be viewable by the user. All the conditional signal assignments and selected signal assignment statements are converted to their boolean equivalent in this intermediate form.The optimization process 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 rules of boolean algebra. This removes the redundant logic and reduces the area requirement.

(d) Draw internal block diagram of CPLD Xilinx series and explain the function of each block.

Ans: **[Block Diagram -2marks,Explanation 2 Marks]**

(marks to be given to the answer covering just major points)

Diagram:



Internal Block Diagram of XC 9500 CPLD

Explanation:

ARCHITECTURAL DESCRIPTION

- Each external I/O pin can be used as an input, an output, or a bidirectional pin according to device programming. The I/O pins at the bottom are also used for special purposes.
- Any of the 3 pins can be used as "Global Clocks" (GCK).Each macrocell can be programmed to use a selected clock input.
- One pin can be used as a "Global Set/Reset "(GSR).Each macrocell can use this signal as an asynchronous Preset or Clear.
- Two or Four pins depending on the devices can be used as "Global Three State Controls"(GTS).One of the signals can be selected in each macrocell to output enable the corresponding output driver when the macrocell's output is hooked to an external I/O pin.
- Only four Functional Blocks(FB) are shown but XC9500 scales to accommodate 16 FB's in the XC95288.Regardless of the specific family member each FB programmable receives 36 signals from the switch matrix. The inputs to the switch matrix are the 18 macrocell outputs from each of the functional blocks and the external inputs from the I/O pins.
- Each Functional block also has 18 outputs that run under the switch matrix and connect to the I/O blocks. These are the output-enable signals for the I/O block output drives; they're used when FB macrocells's output is hooked up to an external I/O pin. Each Functional Block has programmable logic capability with 36 inputs and 18 outputs. Fast Connect Switch Matrix connects all Functional Block outputs to the I/O blocks and the input signals from the I/O block to the Functional Block.

Q.4 (B) Attempt any ONE: 6M

a. Write the VHDL program to implement 9:4 encoder.

Ans:

Program:

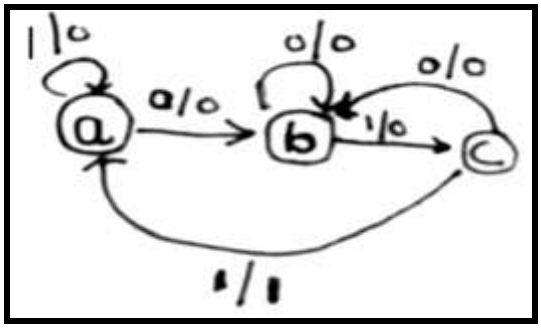
VHDL for 9:4 ENCODERS
[6 marks= 1 mark entity +1mark Architecture +1 mark Process+3marks case]

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

ENTITY enc9_4 IS
   PORT ( d : IN STD_LOGIC_VECTOR(8 DOWNTO 0);
            b : OUT STD_LOGIC_VECTOR(3  DOWNTO 0));
END enc8_3;

ARCHITECTURE behavioral OF enc9_4 IS
BEGIN
PROCESS(d)
BEGIN
CASE d IS
                        WHEN "000000001"=>b<="0000";
                        WHEN "000000010"=>b<="0001";
                        WHEN "000000100"=>b<="0010";
                        WHEN "000001000"=>b<="0011";
                        WHEN "000010000"=>b<="0100";
                        WHEN "000100000"=>b<="0101";
                        WHEN "001000000"=>b<="0110";
                        WHEN "010000000"=>b<="0111"
WHEN OTHERS =>b<="1000";
END CASE;
END PROCESS;
END behavioral;
```

b. Design a sequence detector of 011 using JK flip-flop. Use mealy machine.

Ans: Sequence Detector 011 for Mealy using JK flip-flop

STATE DIAGRAM **[1 mark]**



STATE TABLE **[2marks]**

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

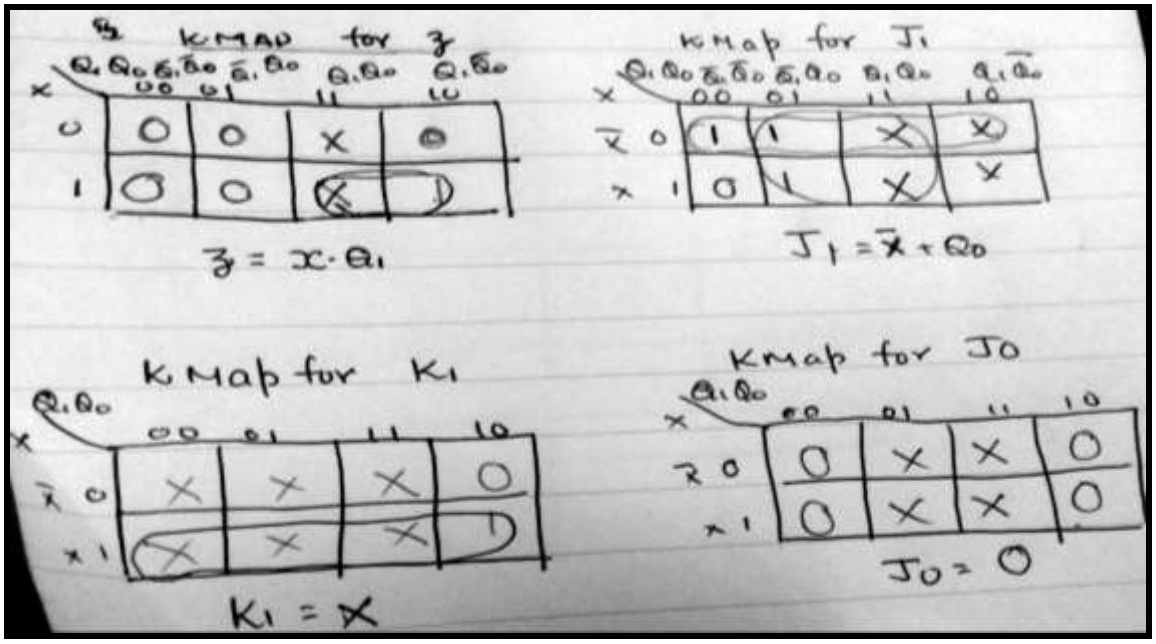
EXCITATION TABLE of JK Flip Flop

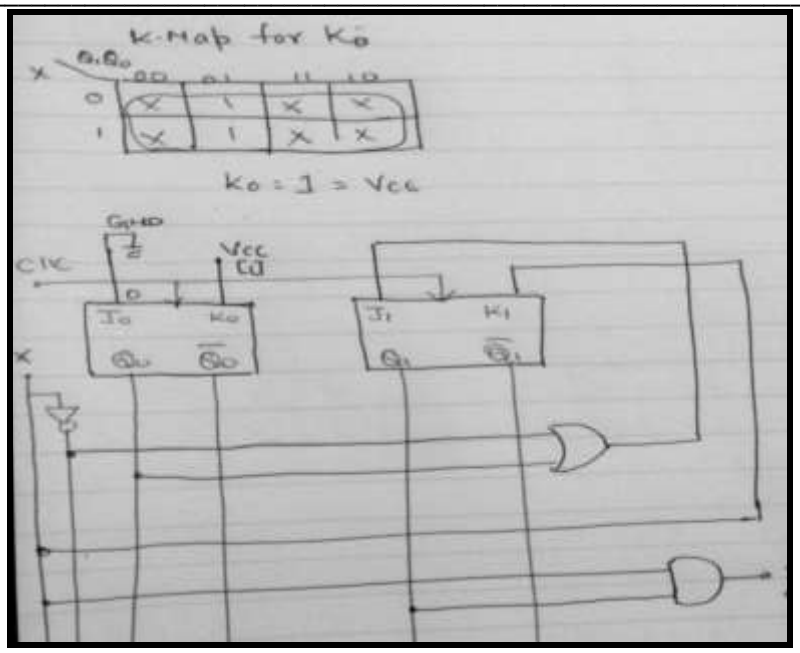
EXCITATION		INPUTS	
Q	Q*	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

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 **[3marks]**





Q.5) Attempt any FOUR:

(a) Explain the architecture of SPARTAN 3 FPGA family with neat diagram.
Ans: (Diagram 2 marks, 2 marks Explanation)

Diagram:

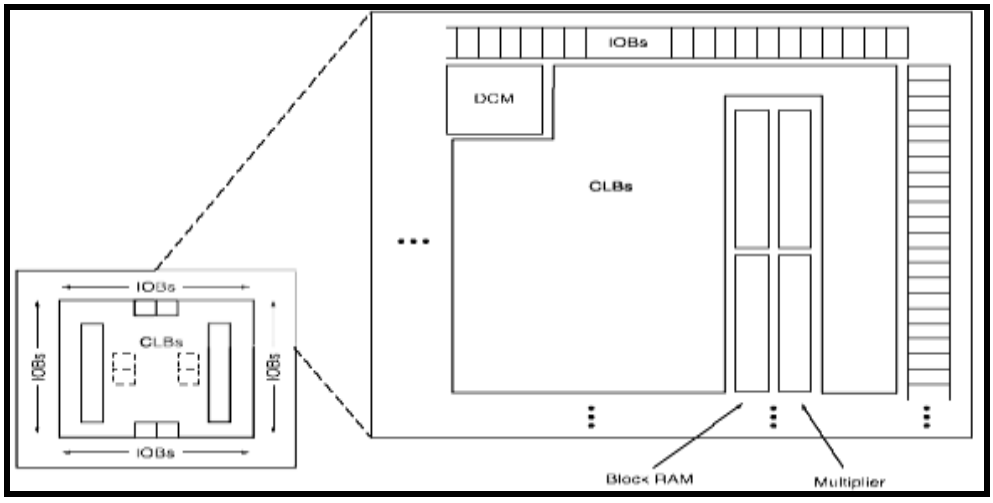


Fig. SPARTAN 3 FPGA

Explanation:

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.

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.

b) Compare FPGA and CPLD

Ans:

Comparison: (4 marks 4 points)

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.

c) Compare synchronous and asynchronous sequential circuits on the basis of:

Ans:

Comparison: (4 marks 4 points)

Parameter	Asynchronous	Synchronous
Definition	It is sequential circuit whose behaviour depends upon the sequence in which the input signals change	It is a sequential circuit whose behaviour can be defined from the knowledge of its signal at discrete instants of time.
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

d) List operators & their operations used in VHDL.

Ans: (Any 4 operator styles 4 marks)

The predefined operators in VHDL array of bit and

1) **Logical operators**: and ,or, not, nand, nor, xor, xnor.
   Logical operators are defined for **type bit** and Boolean, one dimensional array of bit and Boolean type.
2) **Relational operators:**
   i) = equality
   ii) /= inequality
   iii) < less than
   iv) < = less than or equal to
   v) > greater than
   vi) >= greater than or equal
3) **Shift operators:**
   sll shift left logical
   srl shift right logical
   sla shift left arithmetic
   sra shift right arithmetic
   rll right left logical
   rrl right right logical
4) **Adding operators**
   '+' operator is used for addition
   '-' operator is used for subtraction
   '+' and '-' operators are predefined in VHDL for all integer operands.
   & is the concatenation operator. It works on vector only.
5) **Multiplying operators**
   Multiplying operators are predefined in VHDL for all integer types.

'*' is used for integer multiplication
'/' is used for integer division
'*' and '/' are used for floating point numbers and are defined only for integers.

6) Miscellaneous operators

The two misc operators are abs and **
Abs means absolute
** means exponential.

e) State event scheduling & sensitivity list.

Ans: (2 Marks for Event Scheduling 2 Marks for Sensitivity list)

Event scheduling:

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.

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. 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.

f) Explain CMOS transmission gate with diagram.

Ans: (2 Marks Diagram 2 Marks Explanation)

Diagram:

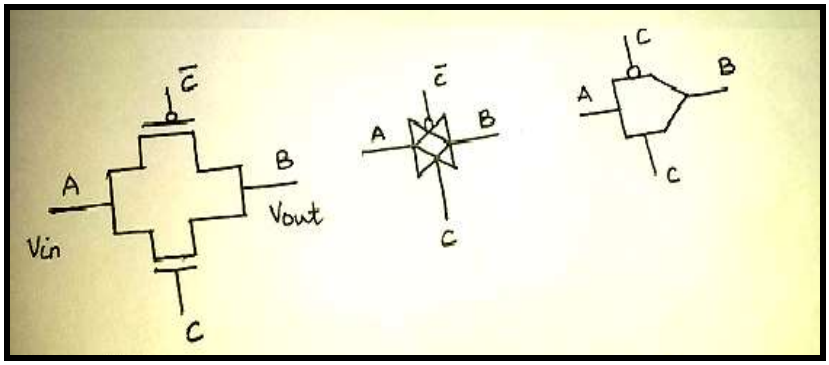


Figure: CMOS transmission gate

Explanation:

It 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 & B which is controlled by C.

If the control signal C is logic high, VDD, then both the transistors are turned ON and provides a low resistance current path between the nodes A & B. If C is low, then both the transistors are off & path between A & B is open circuit. This condition is called high impedance state.

Q.6 Attempt any FOUR: 16M

(a) Explain with neat diagram wafer processing by C-Z method.

Ans: (2 mark for diagram + 2 marks description)

Diagram:

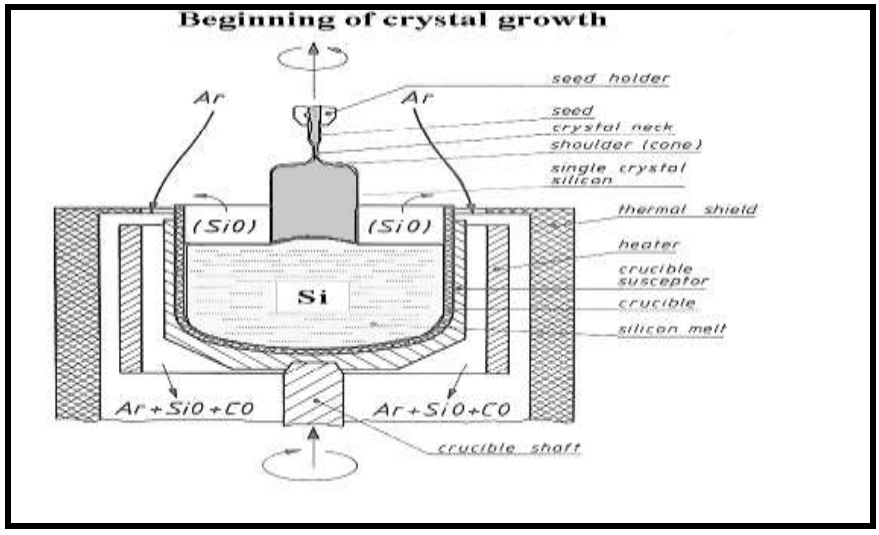


Figure: Wafer processing using CZ process

Explanation:

- The silicon is first melted through heat. A rotative block (the *crystal puller*) with a small silicon crystal (the crystal seed) is introduced very slowly into the melted silicon.
- The temperature is decreased as soon as the seed touches the silicon surface. Gradually, the silicon attaches the seed. Very slowly one lifts up the rotative block. The crystal continues to grow, forming a cylindrical silicon ingot.
- The silicon ingots are then sliced to obtain the circular wafers. The wafer surface is polished through chemical and mechanical techniques to obtain a high-quality substrate in which the semiconductor devices will be fabricated.
- The pure silicon of the wafer is then doped. That is, small impurities are added in a controlled way to define the substrate resistivity. The charge carrier type is also defined within the doping process. Hence, one can have n-type or p-type substrates.

(b) State and explain :

(i) Zero Modeling

(ii) Simulation Cycle

Ans: (marks to be given to the answer covering just major points)

Explanation:

(i) Zero Modeling: (2 marks)

The ordering of zero delay events is handled with a fictitious unit called delta time. Delta time represents the execution of 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 time step. „ Delta time steps continue until there is no activity for the same instant of simulated time.

(ii) Simulation Cycle (2marks)

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 specification in HDL that plays the role of a complete simulation environment for the analyzed system.

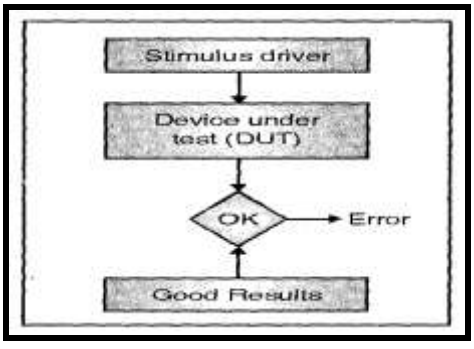
A test bench is at the highest level in the hierarchy of the de&gn. It instantiates the design under test (DUT) and provides the necessary input stimulus to DUT and examines the output from DUT.

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. Test bench block diagram:



(c) State the test bench and write its appliction.

Ans: (Test Bench -2marks, Applications-2 marks)

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:

- 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.
- It is a 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 de&gn. It instantiates the design under test (DUT) and provides the necessary input stimulus to DUT and examines the output from DUT.

(d) What is simulation? Explain event based and cycle based simulator.

Ans: (Simulator 2 marks, Event Based Simulator 2 marks)

(marks to be given to the answer covering just major points)

Simulation:

- Simulation is functional emulation of a circuit design through software programs that use models to replicate how a device will perform in term of timing and results. Simulation and testing being
- important steps in same in details.. A VHDL simulation serves as a basis for testing complex designs and validating the design prior to fabrication. If the VHDL code conforms to all the rules, the compiler generates intermediate code, which can be used by a simulator or by a synthesizer.
- A circuit represented in the form of logic expressions can be simulated to verify that it will function as expected. The tool that performs the task is called a functional simulator.
- There are two types of simulators as under:

1. Event based simulator. 2. Cycle based simulator

Event Based simulator:

- 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.

Cycle-Based simulator :

- Cycle-based simulation ignores intra-cycle state transitions, i.e. they check the status of target signals periodically irrespective of any events. This can boost performance by 10 to 50 times compared to traditional event-driven simulators.
- Cycle-based technology offers greater memory efficiency and faster simulation run-time than traditional pure event-based simulators.
- Cycle-based simulators work best with synchronous design but give less timing accuracy with asynchronous design.
- Signals are treated as variables. Functions such as AND, OR etc. are directly converted to program statements.
- Signal level functions such as memory blocks, adders, multipliers etc. are modeled as subroutines.
- For every input vector, the code is repeatedly executed until all variables have attained steady value.
- Compiled code simulator is efficient when used for high-level design verification. Inefficiency is incurred by the evaluation of the design when only few inputs are changing.

(e) Draw & explain ASIC Design flow.

Ans: **(Any suitable diagram 2marks, Description 2 marks)**

Diagram:

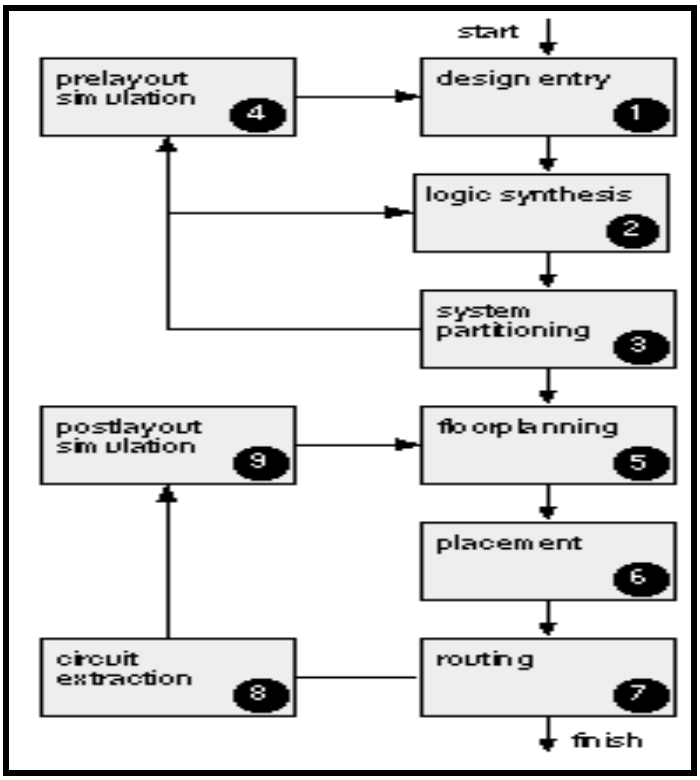


Figure: ASIC design flow

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

(f) Write VHDL program for D-flip flop.
Ans: **(Entity 1 mark, Architecture -3 marks)**

Program:

```
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)
                begin
                if (C'event and C='1') then
                Q <= D;
                end if;
        end process;
end archi;
```