T.Y. Diploma : Sem. VI
[EJ/ET/EX/EN/EI/IE]
Very Large Scale Integration

Time : 3 Hrs. Prelim Question Paper Solution **Marks : 100**

Q.1(a) Attempt any THREE questions. **[12]**

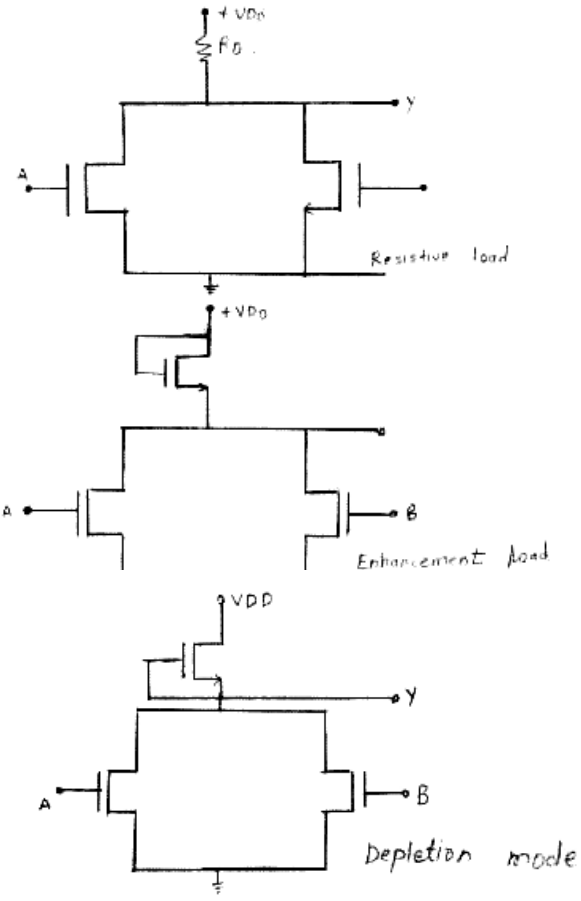
Q.1(a)(i) Differentiate between Xiliv and Atmel series architecture of CPLD. (four points) **[4]**

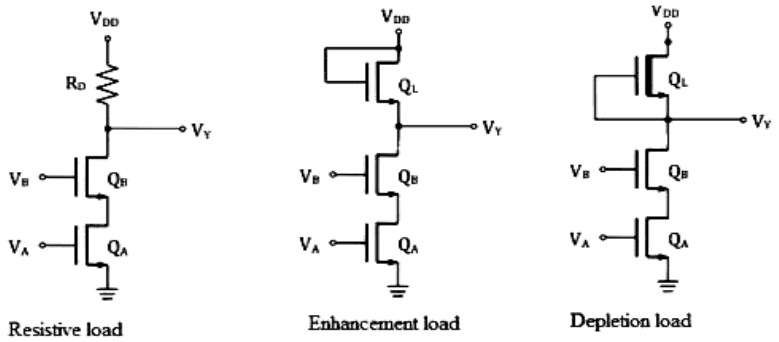
Ans.: [Four points 1 mark each]

	Xilinx CPLD	**ATMEL CPLD**
1.	XC9536, XC9572, XC95108 these have 36, 72 and 108 macro-cells.	ATF1502, ATF1504, ATF1508 these have 32, 64 and 128 macro- cells.
2.	Available in variety of packages but 44 and 84 pins PLCC or J-lead packages are more popular.	Available in variety of packages but 44, 68 and 84 pins PLCC or J-lead packages are more popular.
3.	Xilinx offers their web packs free download.	ATMEL offers their WinCUPL PLD compiler for free download.
4.	The most current version only works on windows XP.	The most current version only works on Windows XP.
5.	In conversion application XC4000 FPGAs the equivalents are 1. XC4002,XC4003 XC5200 SERIES XC5202 AND XC5204 XC505((SPARTAN SERIES) XC 10.	In conversion application atmel series FPGAs equivalent are 1. AT40K05,AT40K05/10 2. AT40K05,AT40K05/10 3. AT40K05,AT40K05/20

Q.1(a)(ii) Draw NAND and NOR gates using NMOS. **[4]**

Ans.: [2 mark each Any one diagram for NAND and NOR each]





Q.1(a) (iii) Write any two pro's and any two con's of VHDL. **[4]**

Ans.: Pros : [2 mark]

- 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 : [2 mark]

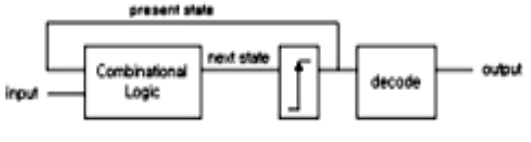
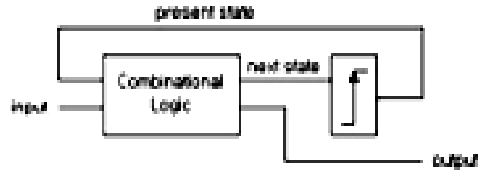
- Extremely verbose coding
- VHDL modules must be defined by a prototype and declared before they're used, causing you to change code in at least 3 places if you want to make a change to the interface.
- The use of the keyword "downto" in every bit vector definition is tedious.
- Sensitivity lists: Missing 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.

Q.1(a)(iv) Give the difference between Moore and Mealy machines (four points). **[4]**

Ans.: [Four points 1 mark each]

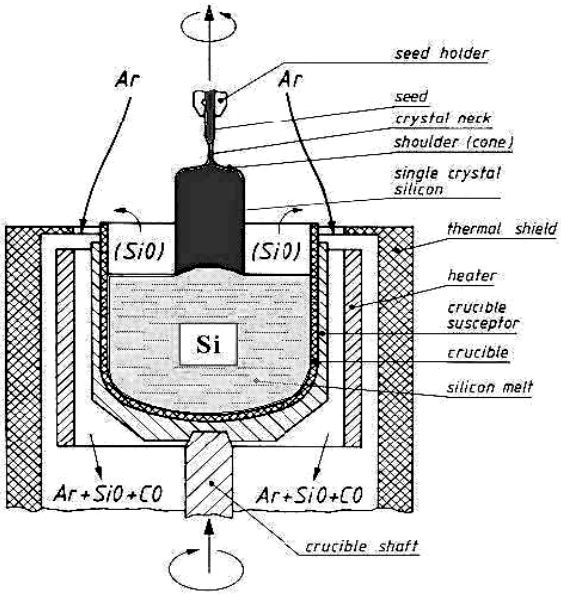
	Moore Machine	Mealy Machine
1.	Output is function of state of machine.	Output is function of state of machine and present input condition.
2.	Requires more number of states.	Requires less number of states.
3.	Faster	Slower
4.	Simple design	Complex design
5.	Output in state	Output is at the time of state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

Q.1(b) Attempt any ONE questions. [6]

Q.1(b)(i) Explain CZ process for wafer fabrication, with neat diagram. [6]

Ans.: [Diagram 3 marks, Explanation 3 marks]

Czochralski (CZ) Process:



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. 14250C), the atmosphere just above the polysilicon melt is typically helium or orgon for freezing.

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

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

The produced crystalline silicon rod is then slicing into wafers using cutting tools like diamond blades. Following slicing at least one face of the wafer is polished to flat scratch free mirror finish surface.

Q.1(b) (ii) Write the VHDL program to implement 4 bit adder. [6]

Ans.: [1 mark Package Declaration, 1 mark Entity and 4 marks Architecture]

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

entity adder is
    Port ( a : in std_logic_vector(3 downto 0);
           b : in std_logic_vector(3 downto 0);
           s : out std_logic_vector(3 downto 0);
           c : out std_logic;
           cin : in std_logic);
end adder;
architecture behavior of adder is
```

```
begin
process(a,b,cin)
variable u:std_logic;
begin
u:=cin;
for i in 0 to 3 loop
s(i)<=a(i) xor b(i) xor u;
u:=(a(i) and b(i))or(b(i) and u) or(u and a(i));
end loop;
c<=u;
end process;

end behavior;
```

Q.2 Attempt any FOUR questions. [16]

Q.2(a) Define the following terms : [4]

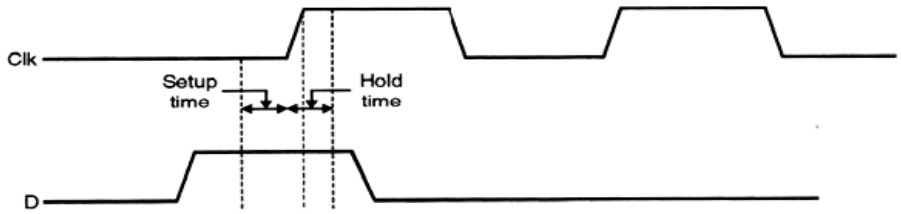
(i) Meta stability (ii) Set-up time (iii) Hold time (iv) Fan-out

Ans.: [1 mark each definition]

(i) **Metastability**: Metastability is the ability of digital system to persist for an unbounded time in an unstable equilibrium or metastable state. In this state the circuit may be unable to settle into a stable 0 or 1 logic level within the time required for proper circuit operation.

(ii) **Set-up time:** Set up time during which the input must be stable before the clock transition takes place.

(iii)



(iv) **Hold time:** Hold time is the minimum time for which the input must be held constant after the transistion occurs.

(v) **Fan-out:** Fan-out of an output 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 capacity. It is the greatest number of inputs of gates of the same type to which the output can be safely connected.

Q.2(b) Compare FPGA and CPLD. [4]

Ans.: [Four points 1 mark each]

	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 fo 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.2(c) Explain oxidation and diffusion process in fabrication process. [4]

Ans.: Oxidation: [2 marks]

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. The oxidation of silicon is achieved by heating silicon wafers in an oxidizing atmosphere such as oxygen or water vapor .There are two types

(i) Wet oxidation

(ii) Dry oxidation

Diffusion : [2 marks]

The process of junction formation, which is transition from p to n type or vice versa. Diffusion of impurity atoms into silicon crystal takes place only at elevated temperature, typically 900 to 1100°C.The following material are used

P type : Borane(B2H6)

N type: Phosphine(PH3)

Q.2(d) What are the advantage of twin-tub process of CMOS fabrication? [4]

Ans.: [Any 4 advantage 1 marks each]

Advantages of Twin-tub process are :

1. Separate optimized wells are available.
2. Balanced performance is obtained for n and p transistors.
3. Make it possible to optimize "Vt", "Body effect", and the "Gain" of n, p devices, independently.
4. The parasitic transistor is not likely to be formed.
5. No latch-up.

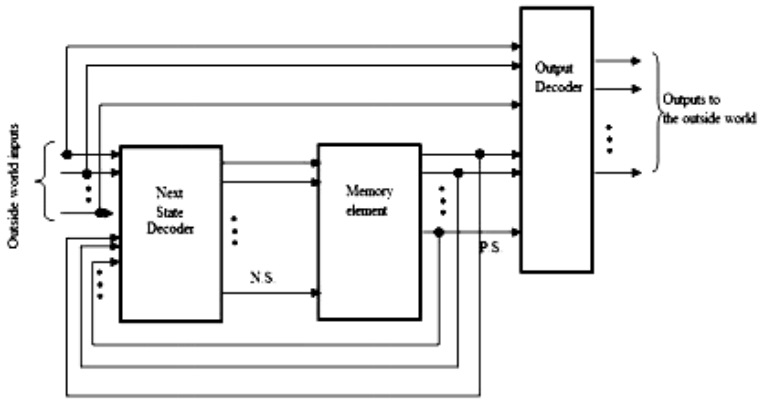
Q.2(e) List the types of FSM. Draw labelled diagram of each. [2]

Ans.: [2 marks each]

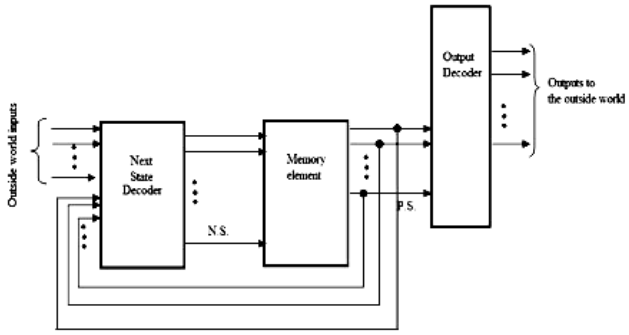
Types of FSM:

- **Melay Machine**
- **Moore Machine**

- **Melay Machine**

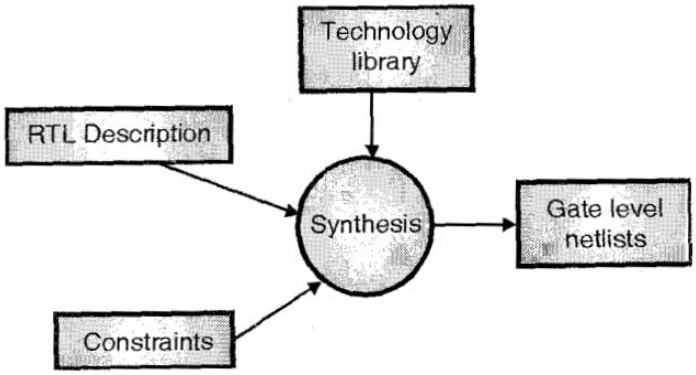


- **Moore Machine**



Q.2(f) Draw the HDL design flow for synthesis. Write the steps in the flow. **[4]**

Ans.: Diagram : [2 marks]



Explanation : [2 marks]

Synthesis is an automatic method of converting higher level of abstraction to lower level of abstraction.

- 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.
- 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 un optimized 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.

OR

Simple steps :

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)

Q.3 Attempt any FOUR questions. [16]

Q.3(a) Design clocked sequential circuit using Toggle flip-flop to count from 00 to 11 (2 bit counter). [4]

Ans.:

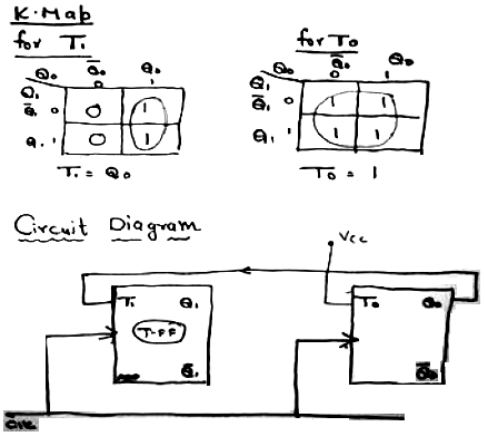
- Number of bits= 2 (i.e. 00 – 11)
- Previous State = Q_1Q_0
- Next State = $Q_1^*Q_0^*$
- **Excitation Table for T- Flip Flop**

Transition		Toggle input
Qn	**Qn***	T
0	**0**	0
0	**1**	1
1	**0**	1
1	**1**	0

- **State Table** [2 marks]

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

[2 marks]



Q.3(b) Compare BJT and CMOS (any four). [4]

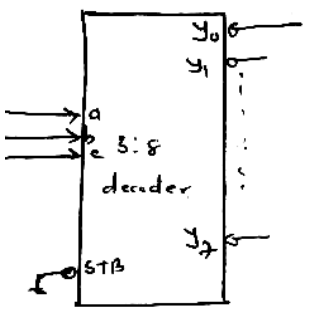
Ans.: [Four points 1 mark each]

	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

Q.3(c) Write the VHDL code for 3:8 decoder. [4]

Ans.: [Entity 1 mark and Architecture 3 marks]

VHDL code for 3:8 decoder



```
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 beh of decoder is
signal temp : std_ logic_vector(3 downto 0);
    begin
temp <=stb & a & b &c ;
Y <="01111111" when temp =" 0000" else
    "10111111" when temp = "0001" else
    ( and so on for each input )

    "11111110" when temp="0111"else
    "zzzzzzzz;
End beh;
```

(Any logic using with ----select or case statement or if statement can be used for program. Give marks to entity declaration and interpretation of syntax)

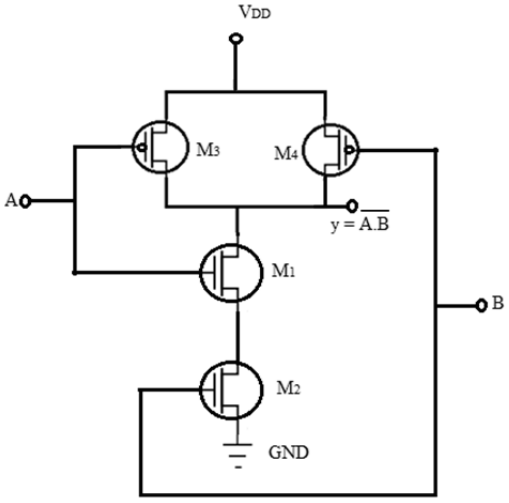
Q.3(d) Draw NAND gate using CMOS transistors. **[4]**

Ans.: [Diagram 2 marks and Explanation 2 marks]

Explanation:

A general CMOS circuit consists of an n-MOS logic block between the output and ground and a pMOS logic block between the output and VDD.

- In CMOS, Number of nMOS transistors = Number of pMOS transistors. For two input NAND gate, there are two nMOS transistors in pull down, and hence two pMOS transistors in pull up. Since pMOS and nMOS are complementary to each other, pMOS transistors are connected in parallel between output and VDD.
- Two input CMOS NAND gate is shown in figure. The pull down sections has transistors M_1 and M_2 in series and pull up section has transistors M3 and M4 are in parallel.
- Transistors M_1 and M_3 from one CMOS with A as input and transistors M_2 and M_4 from another CMOS with B as a input.



Truth Table (Optional):

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

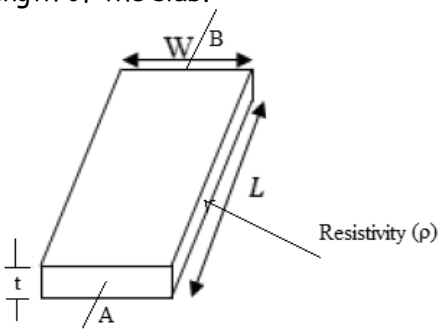
Q.3(e) Explain the estimation of resistance of channel for MOSFET (Sheet Resistance) [4]

Ans.: [4 marks]

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 /the length of the slab.



$$R_{AB} = Rs \cdot \frac{L}{W} \Omega$$

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

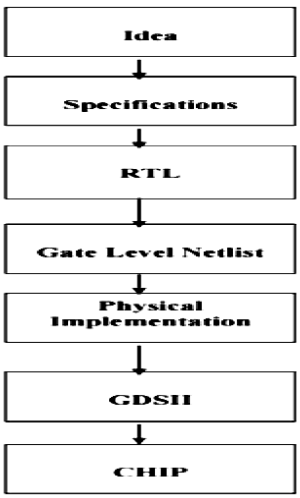
Therefore, Rs = ohm per square

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

Q.3(f) Draw the ASIC design flow and explain it. [4]

Ans.: ASIC Design Flow:

Diagram : [2 marks]



Explanation : [2 marks]

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 Netlist. 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 netlist as output. The resulting gate-level netlist 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.4(a) Attempt any THREE questions. **[12]**

Q.4(a) (i) Explain with syntax. **[4]**

(1) Entity **(2) Architecture**

Ans.: Entity: [2 marks]

- A VHDL Entity specifies the name of the entity, the ports of the entity, and other entity related information.
- All designs are created using one or more entities.
- **Syntax:**

  Entity <entity_name> **is**
  Generic (<generic_list>);
  Port (<port_list>);
  End <entity_name>
- The keyword entity signifies that this is the start of the entity statement. The standard type provided is BIT.

- **Example :**

```
ENTITY mux is
PORT (a, b, c, d: IN_BIT;
s0, s1: IN_BIT;
Y: OUT_BIT);
END mux;
```

- Name of the user created object is mux. The name of the entity is mux.
- The entity has seven ports in the PORT Clause.
- Six of them are input ports and one is output port which is notified as IN and OUT respectively.
- The four data input ports (a, b, c, d) and two select input ports (s0, s1) and one output port (y) are of type BIT.
- The entity describes the interface to the outside world. It specifies the number of ports, direction of the ports, type of ports, etc.

Architecture: [2 marks]

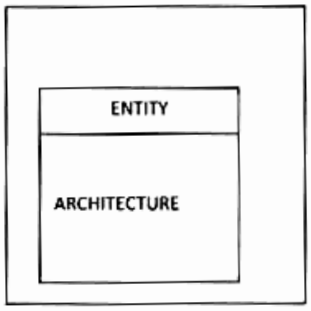
- The entity describes the interface to the VHDL model.
- The architecture specifies behaviour, function, interconnection or relation between input and output of an entity.
- **Syntax:**

```
Architecture architecture_name of entity_name is
architecture_declarations
Begin
concurrent_statements
End [architecture] [architecture_name];
```

- It describes the contents of an entity. The reason for connection between the architecture and the entity is that an entity can have multiple architectures describing the behaviour of the entity.
- The keyword ARCHITECTURE signifies that this statement describes architecture for an entity.
- The statement of the architecture starts with the keyword BEGIN and ends with the END netlist statement.
- **Example:**

```
Architecture AND1 of ANDGATE is
--declarations
Begin
--statements
Y <= A AND B;
End architecture AND1;
```

(OR)



VHDL program file structure

Entity:

- All designs are expressed in terms of entities.
- An Entity is the most basic building block in a design.

- Without communication there is no system. In other words it must get some input data from environment & should output some data.
- Without and interface, the system would be useless.
- In VHDL, the systems external interface is described by its entity. (black box view)

Port:

- It is the list of interface pins (signals) of the entity along with their directions (mode) & type.
- Most frequently used types are:
  - Bit
  - Boolean
  - Integer
  - Real
  - Std_logic (is same as BIT with few more advantages)
- Each interface port can have one of the following modes :
  - In : value can only be read within the entity model and can't be written.
  - Out : value can only be updated within the entity model; & can't be read.
  - Inout : value can be read and updated within the entity model.
  - Buffer : value can be read and updated within the entity model but it can't have more than one source.
- entity_name : It is an identifier and defined by the user to the entity name. The identifier for the entity must start with letter followed by arbitrary combination of letters, digits and underscore symbols. It is possible to write an entity without any generics, ports & passive statements (it is used for test benches).

Syntax of an Entity

entity <entity name> is
generic (<generic_list>);
port (<port_list>);
end <entity_name>;

Architecture :

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

Syntax of an Architecture :

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

Example of a VHDL Program

```
LIBRARY IEEE;
Use IEEE.std logic_1164.all;
entity D_FF is
port (D, CLK : in BIT);
Q : out BIT = '0';
NQ : out BIT = '1');
end entity D_FF;

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

Q.4(a) (ii) Explain the following terms : [4]

(a) Event scheduling (b) Simulation cycle

Ans.: [2 marks]

- Events are changes to the wire or registers. Statements can schedule event to occur at particular time or to be triggered by other events in current time slot or at later simulation time.
- The events queue is segmented into five different regions. Each event will be added to one of the five regions in the queue but are only removed from the active region.
  1. **Active events:** This event occurs at current simulation time and can be processed in any order.
  2. **Inactive events:** This event occurs at current simulation time but shall be processed after all active events are processed.
  3. **Non-blocking assign update event:** This event is evaluated during some previous simulation time, but shall be assigned at this simulation time after all active and inactive events are processed.
  4. **Monitor event:** This event is processed after all active events, inactive events and non-blocking assign update events are processed.
  5. **Future events:** This event occurs at some future simulation time. Future events are divided into future inactive event and future non-blocking assign update event.
- Processing all active events is called simulation cycle.

Simulation cycle : [2 marks]

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 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.

Q.4(a) (iii) List the any four logical operators in VHDL. **[4]**

Ans.: [1 mark each]

Logical Operators: These are defined for type bit and Boolean, one dimensional array of bit and Boolean type.

The logical operators are:

- AND
- OR
- NAND
- NOR
- XOR
- XNOR
- NOT

Q.4(a) (iv) List the advantages and disadvantages of VHDL. **[4]**

Ans.: Advantages of VHDL: [2 marks]

(1) Standard language.
(2) Fully expressive language.
(3) Hierarchical.
(4) Configurable.
(5) Tool availability.
(6) Consistency and completeness checks.
(7) Tight coupling to lower levels of design.
(8) Supports hybrid modeling.

Disadvantages of VHDL: [2 marks]

(1) Extreme verbose coding
- VHDL modules must be defined by a prototype and declared before they are used, causing you to change code in atleast three places if you want to make a change to the interface.

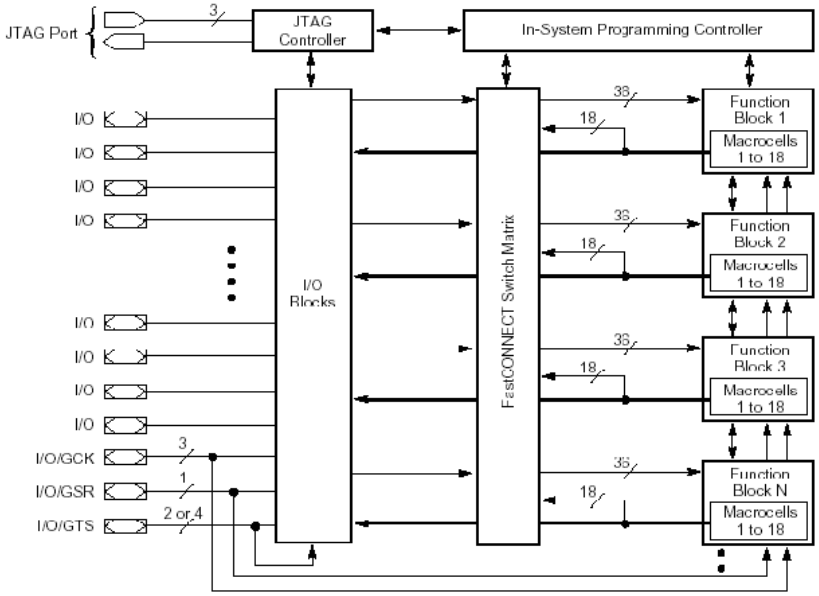
(2) Sensitivity lists
- Missing a single signal in the sensitivity list can cause major differences between simulation and synthesis.
- Each process must have a sensitivity list that may sometimes be very long.

(3) Type conversions
- Signal types that are clearly related (e.g. std_logic and Std_logic_vectors) cannot be simply used together and must be converted into another type.

Q.4(b) Attempt any ONE questions. **[6]**

Q.4(b) (i) Draw the complete block diagram of CPLD and explain the same. **[6]**

Ans.: Architecture of CPLD [2 mark]



Architectural Description: [4 marks]

- 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 macro cell can be programmed to use a selected clock input.
- One pin can be used as a "Global Set/Reset "(GSR).Each macro cell 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 macro cell to output enable the corresponding output driver when the macro cell'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 macro cell 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 macro cells 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) (ii) List and explain the main steps carried in typical n-well, CMOS fabrication process with neat sketches. **[6]**

Ans.: n-well, CMOS fabrication process : [6 marks]

The fabrication steps are as follows :

Step 1 : Formation of n-well regions.

Step 2 : Define nMOS and pMOS active areas.

Step 3 : Field and gate oxidation (thinox)

Step 4 : Form and pattern polysilicon.

Step 5 : p^+ diffusion.

Step 6 : n^+ diffusion.

Step 7 : Contact cuts.

Step 8 : Deposit and pattern metallization.

Step 9 : Over glass with cuts for bonding pads.

The first mask defines the n-well regions. This is followed by a low dose phosphorous implant driver 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. 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 metallize.

Q.5 Attempt any FOUR questions. [16]

Q.5(a) Differentiate between asynchronous and synchronous logical circuit. [4]

Ans.: [1 mark each]

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
Output 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

Q.5(b) Explain with the syntax : (i) Signal (ii) Variable. [4]

Ans.: 1. Signal: [2 marks]

- Signal objects are used to connect entities together to form models.
- A signal declaration looks like;

Syntax :

SIGNAL signal_name : signal_type [:= initial value];

2. Variables: [2 marks]

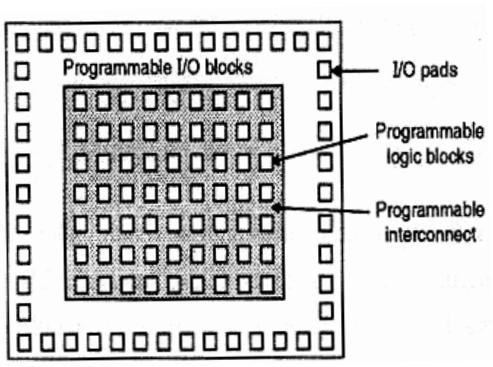
- Variables are used for local storage in process statements and subprograms.
- A variable declaration looks like this

Syntax :

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

Q.5(c) Draw the general FPGA chip architecture and explain the same. [4]

Ans.: [2 marks]

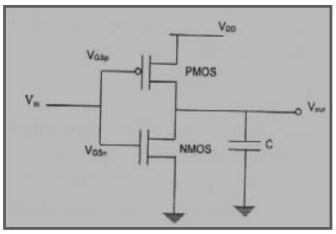


Field programmable Gate Arrays (FPGA): [2 marks]

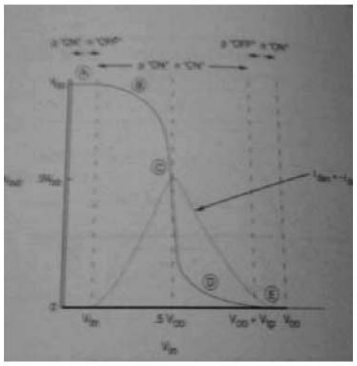
- A field programmable gate array (FPGA) ha large number of programmable logic blocks that are individually smaller than a PLD. The basic structure of a FPGA is sown in figure.
- The programmable logic blocks are arrange in the matrix form with programmable interconnections and the entire array is surrounded by programmable I/O blocks. Each logic block is less capable than a typical PLD, but it has lot more logic blocks than a CPLD of the same size.

Q.5(d) Draw the CMOS inverter characteristic and explain it. **[4]**

Ans.: [Description 2 Marks and Characteristics 2 Marks]



CMOS Inverter



DC Transfer Characteristic

Fig shows the characteristic of CMOS inverter .

Region A

When input voltage 0 ≤ Vin ≤ Vin

N device is cut off and p device in linear range. Vout = Vdd.

Region B

Vin ≤ Vin ≤ Vdd/2 Vout is as per voltage equation

P device is in non saturation region n device is in saturation

Region C

N and P devices in saturation Vout = vin

Region D

Vdd/2 < Vin ≤ Vdd + Vtp

P device is in saturation and n device is in operating in non saturation

Region E

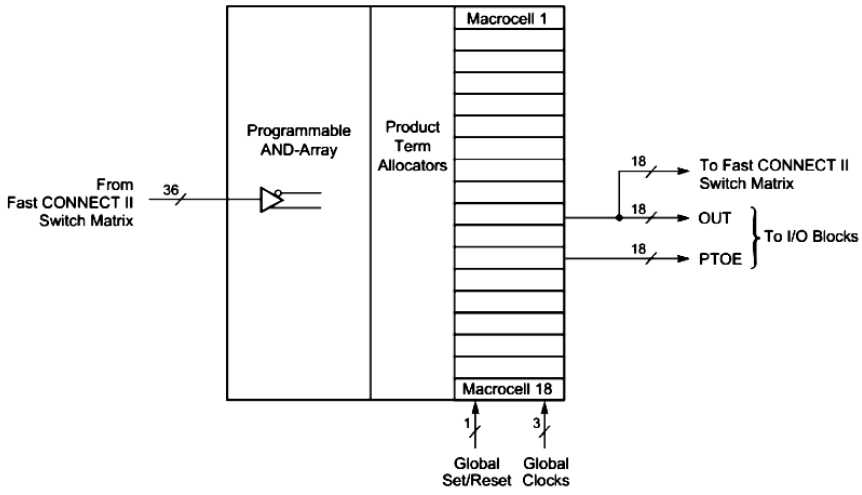
Vin ≥ Vdd-Vtp

P device cutoff n device in linear mode

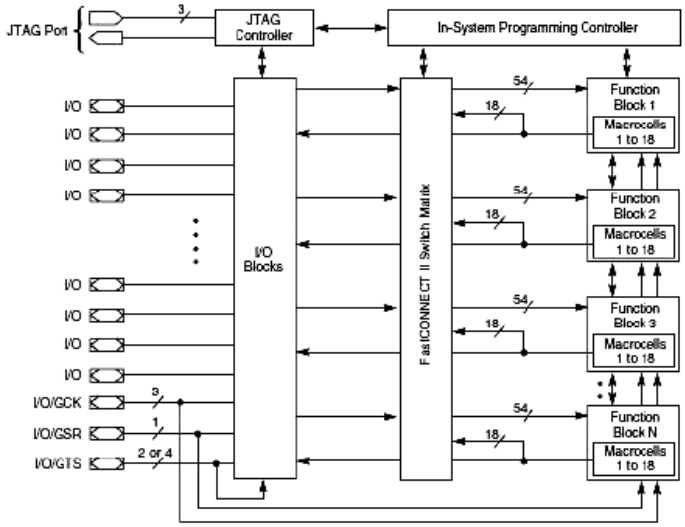
Vout = 0

Q.5(e) Draw the functional block architecture of Xilinx CPLD. **[4]**

Ans.: Diagram : [4 marks]



OR



Q.5(f) What is event scheduling and zero modelling. **[4]**

Ans.: Event scheduling: [2 marks]

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.

Zero Modelling : [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.

OR

In VHDL zero delay circuits and designs that depends on zero delay components can never be built. 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.

Q.6 Attempt any FOUR questions. **[16]**

Q.6(a) Write VHDL code for FULL ADDER. **[4]**

Ans.: [Entity 1 mark and Architecture 3 marks]

FULL ADDER:

A	B	CIN	SUM	COUT
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

SUM	00	01	11	10
0		1		1
1	1		1	

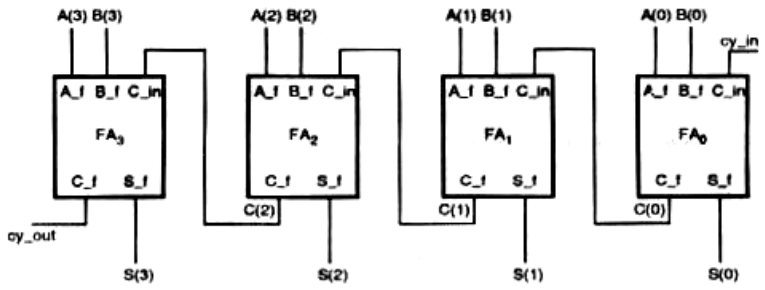
CARRY	00	01	11	10
0			1	
1		1	1	1

SUM = A XOR B XOR C; CARRY = AB + AC + BC;

VHDL code for DATA FLOW model of Full Adder:

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
Entity FA_DF is
Port (A, B, C: in BIT;
SUM, CARRY: out BIT);
End FA_DF;
Architecture FA_dataflow of FA_DF is
Begin
SUM < = A XOR B XOR C;
CARRY < = (A AND B) OR (B AND C) OR (A AND C);
End FA_dataflow;
```

OR




```
LIBRARY IEEE;
use IEEE. Std_logic_1164.all;
entity adder4 is
port (A, B : in bit_vector (3 downto 0) :
       cy_in : in bit;
       S : out bit_vector (3 downto 0) :
     Cy_out bit) :
end adder 4 ;
architecture add4_struct of adder4 is
component full_add
     port (A_f, B_f, C_in : in bit;
           sum_f, carry_f : out bit);
end component;
signal C : bit_vector (2 downto 0);
begin
     FA0 : full_add port map (A(0), B(0), cy_in, S(0), C(0));
     FA1 :  full_add port map (A(1), B(1), C (0), S(1), C(1));
     FA2 :  full_add port map (A(2), B(2), C (1), S(2), C(2));
     FA3 : full_add port map (A(3), B(3), C(2), S(3), cy_out);
end add 4_struct;
```

Q.6(b) Define sensitivity list. State any two VHDL syntax in which sensitivity list is defined. **[4]**

Ans.: [2marks each]

Definition :

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.

Syntax :

Ex. F<=a and b;

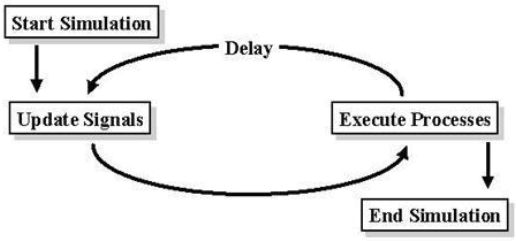
A and b are in the sensitivity list of f. the statement will execute only if one of these will change.

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.

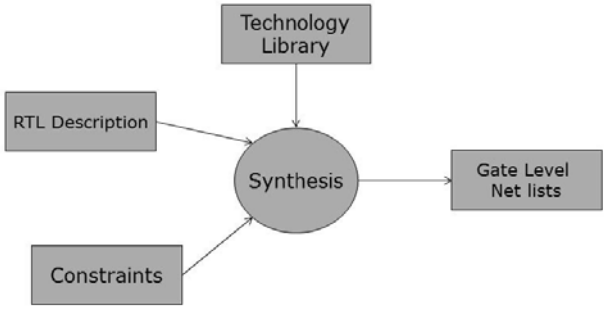
Q.6(c) Draw the simulation cycle and label it. **[4]**

Ans.: Diagram : [4 Marks]



Q.6(d) Draw HDL design flow for synthesis. **[4]**

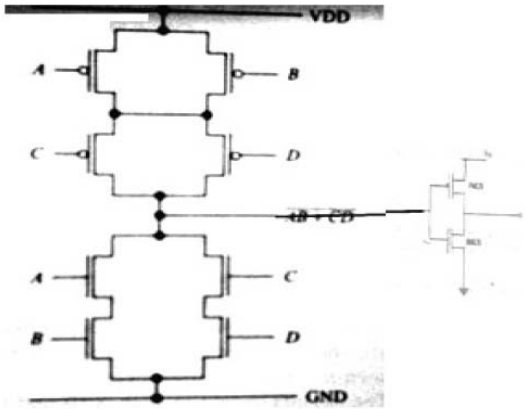
Ans.: Synthesis = Translation + Optimization. [4 marks]



Q.6(e) Execute the following equation by the circuit with CMOS logic. **[4]**

D = [(A·B) + (C·D)]

Ans.: [4 marks]



❑ ❑ ❑ ❑ ❑