| USN | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|

# CMR INSTITUTE OF TECHNOLOGY

| Internal Assesment Test – I | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Sub:** | **ADVANCED VLSI** | | | | | | | **Code:** | **21EC71** |
| **Date:** | **26.10.2024** | **Duration:** | **90 mins** | **Max Marks:** | **50** | **Sem** | **VII** | **Branch** | **ECE-A.B.C.D** |

**Answer Any Five Questions**

| | Questions | Marks | OBE | |
|---|---|---|---|---|
| | | | CO | RBT |
| 1. | Explain Application Specific Integrated Circuits (ASIC) in detail and outline their various classifications. | [10] | CO1 | L1 |
| 2. | (a) Describe the 4-bit Carry Look-Ahead Adder, including its advantages and disadvantages.<br>(b) Difference between PLA and PAL. | [10] | CO1 | L1 |
| 3. | Explain the steps involved in the ASIC design process with neat labeled diagram. | [10] | CO1 | L2 |

| USN | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 4. | (a) Write short note on carry skip/ carry by-pass adder.<br>(b) Write short note on ASIC cell libraries. | [10] | CO1 | L2 |
| 5. | Explain Booth's Algorithm in detail and illustrate it with an example. | [10] | CO1 | L2 |
| 6. | Discuss the following types of Programmable Logic Devices with suitable diagrams:<br>(a) Simple Programmable Logic Devices (SPLDs)<br>(b) Field Programmable Gate Arrays (FPGAs) | [10] | CO1 | L2 |

CI CCI HOD

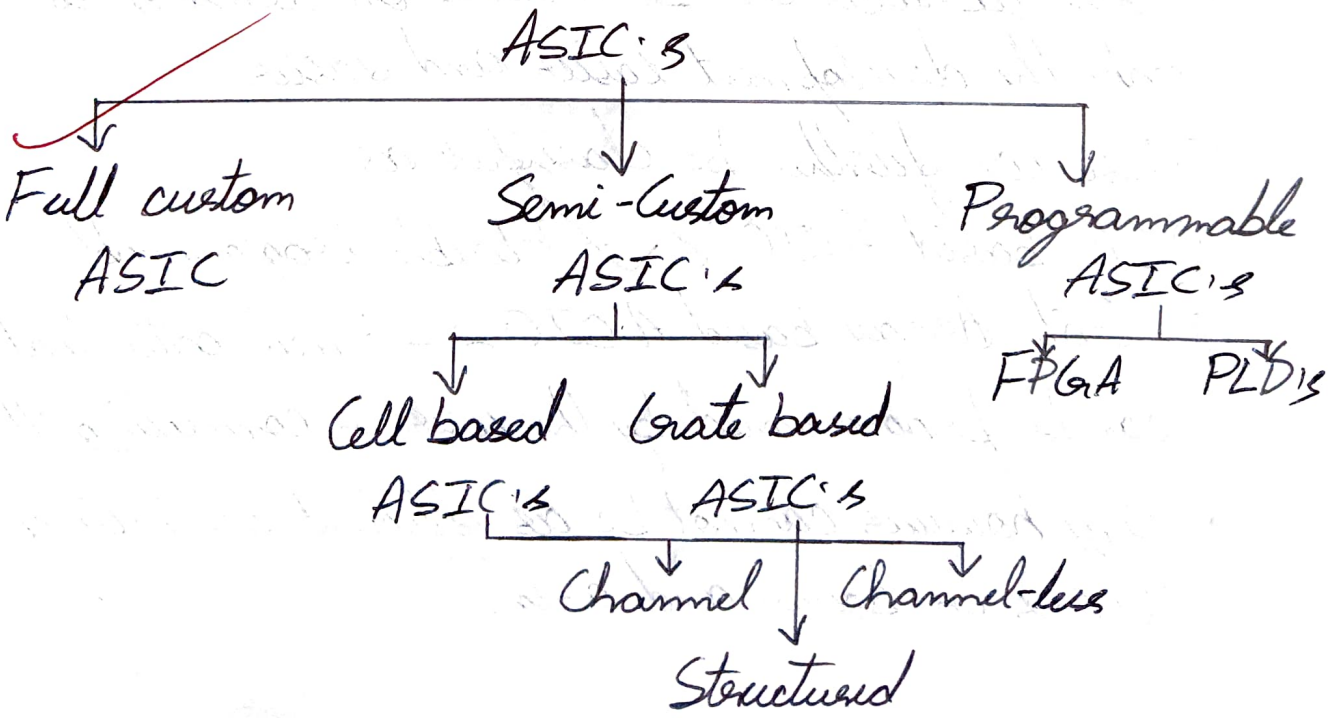
CMR

# INTERNAL ASSESSMENT TEST-I

1. • Application Specific Integrated Circuits (ASIC's) are IC's designed for a specific use-case.
   • They are usually designed for one customer at a time
   • For example, an IC for a bear that talks back, a satellite circuit, for a mobile phone etc...
   • The differ from general purpose Integrated circuits which can be programmed for multiple tasks but cannot be optimized like ASIC's

⇒ ASIC's are classified into 3 types

i) Full-custom ASIC's
ii) Semi-Custom ASIC's
iii) Programmable ASIC's



i) Full Custom IC's

- In full custom ASIC's, the designer design's the IC from scratch as per the specifications and requirements.
- The designer squeezes every inch of the IC's to place the units as densely as possible and make it as energy efficient as possible.
- But to do this would take a person who knows CAED software tool well, and a ton of development cost and time.

ii) Semi - Custom ASIC's

- In semi-custom ASIC's there exists predefined libraries such as (basic gates, MUX, Flip Flops etc) which are preverified by the manufacturer.
- These libraries can be invoked on demand to make the development easier and faster.
- These can further be classified as.

i) Cell based ASIC's - Flexible programmable - Cells

ii) Gate Array based ASIC's - Fixed gates that can be programmed by the user in connecting them.

- They however cannot be as efficient as Full custom ASIC's in power and area.

iii) Programmable ASIC's

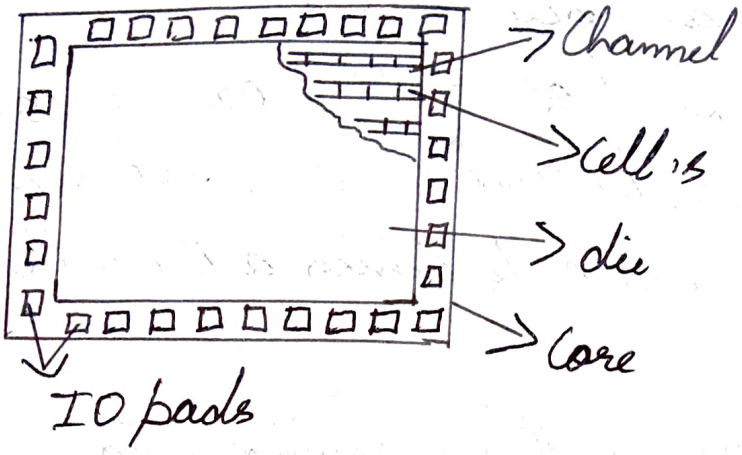
- These kind of ASIC's can be configured by the user easily and in a short duration.
- These have cells which act as the functional unit.
- They are further classified into
  i) FPGA - Field programmable Gate Array's
  ii) PLD's - Programmable logic devices.

a) Cell based ASIC's

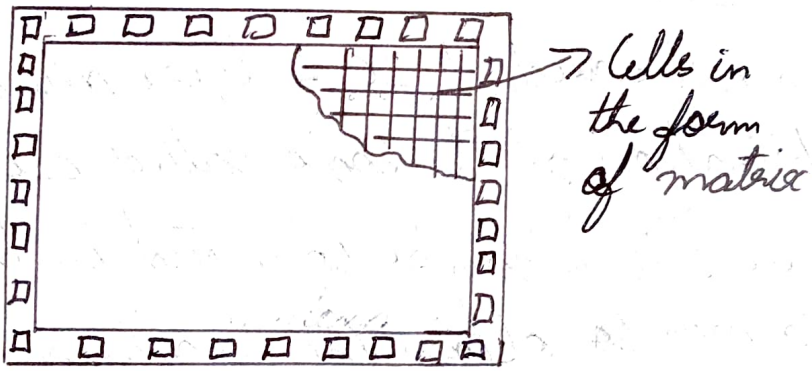
- In cell based ASIC's there are fundamental arbitration of the logic called cell's.
- These cell's can be connected in the required manner to obtain the required functions.
- The cell's contain predefined and pre-verified components like (AND gates, Flip-Flop's and (MUX)

⇒ The Cell based ASIC's may further be classified into three kinds

i) Channeled - Have channels or specified paths for connection of cells.




ii) Channelless: In this redundant or unused cells are used as contact or connections to transmit the data




• Advantages

i) They can be flexible to have on space with flexible blocks of cells.

ii) They have comparitively less congestion as compared to Full custom ASIC's. (Routing path's)

iii) Easily implementable

iv) Saves time in programming

Disadvantages.

i) The cells are pre-programmed and hence the energy consumption is high.

ii) The unused cells contribute to wasted space and hence lower's density.

2 (b) Gate Array based ASICs

- These type of ASICs have a number of fixed Gates (AND (or) OR).
- The connection between these gates are decided by the designer to obtain the required function.
- Advantages

i) They are less energy hungry and consume comparitively less power.

ii) They are easier to work with and program.

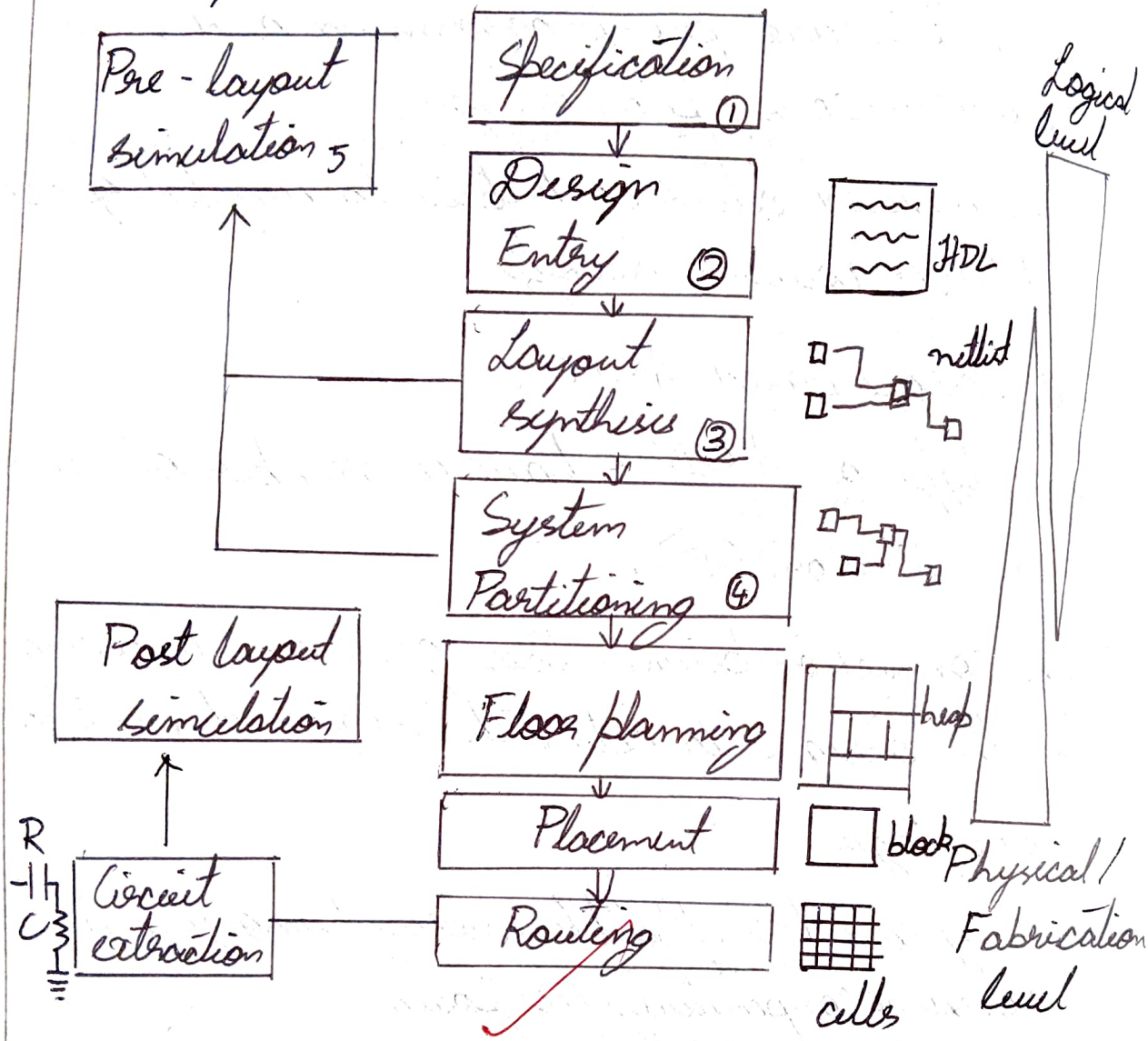
iii) They require less time to program.

- Disadvantages

i) These gates are fixed and hence not properly customisable

ii) They have higher congestion due to the routing of data.

3. The steps involved in ASIC design process is




i) Specification : The specific application and purpose of the IC is is decided upon with additional useage like power or area constraints if any

ii) Design Entry : The design is entered into the design system in the form of a HDL Hardware Definition Language or Verilog

iii) Layout Synthesis : The software interprets these

programs in the form of basic components like (basic gates or flip flops.)

iv) The system Partition: The system is made to the shrink / scale down to the size of ASIC's and allot sufficient resources i.e power and area to it.

v) Floor planning: The efficient location of placing blocks are decided.

vi) Placement: The blocks and cells were placed in the appropriate area's to conserve are and power.

vii) Routing: Decides on the routing or path's of connection's to be made

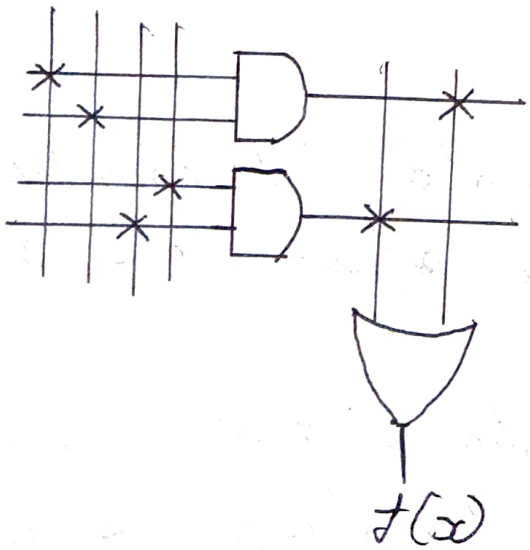
viii) Circuit Extraction: The required values of capacitance and resistance is calculated and displayed.

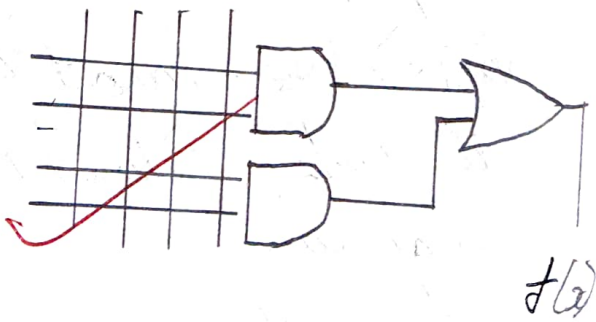
ix) Pre-layout simulation: The calculate of the code - entered in the software works as required.

x) Post layout simulation: It determines if the ASIC's work as intended after fabrication

4 (a)

| PLA | PAL |
|---|---|
| • It stands for Programmable Logic Array. | • It stands for Programmable Array Logic. |
| • It consists of both programmable AND circuits and OR circuits. | • It consists of fixed ~~AND~~ circuits and OR programmable AND circuits |
| • It is more ~~ex~~ expensive | • It is cheaper than PLA |
| • It requires more power. | • It requires less power as compared to PLA |
| • Difficult to programmable the logic comparitively | • Easier to program logic as compared to PLA |



4 (b)

- A 4-bit carry look ahead adder is designed such that it prevents the ripple effect or delay caused by the 4-bit ripple adder
- Instead of propogating the carry, two terms called ~~carry~~ generate and propogate are calculated.

$$G_i = A_i \cdot B_i$$

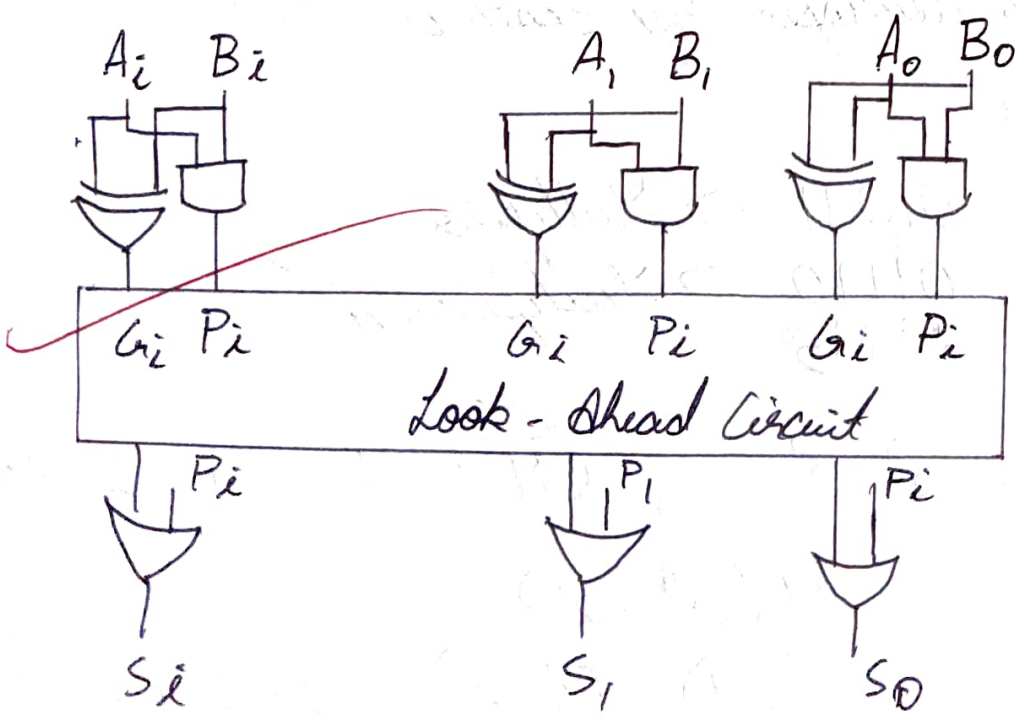
$$P_i = A_i \oplus B_i$$

- These terms are then used to calculate the sum

$$C_1 = G_0 + (P_0 C_0)$$

$$C_2 = G_1 + (P_1 C_1) = G_1 + (P_1 (G_0 + (P_0 C_0)))$$

$$\therefore C_{i+1} = G_i + (P_i C_i)$$



Advantages:

i) It cuts down on the ripple effect and prevents delay

ii) It is scalable and can accomonate n bits

iii) Easy circuit for 4 bits adder look ahead

Disadvantages

i) It becomes complex as the number of bits increases

ii) A large number of gates need to be connected or implemented.

10

5 Booth's Algorithm is a method for multiplying two binary numbers if they have consecutive One's

It was developed by Booth in 1959

Consider

(M) = 01011 → Multiplier

(Q) = 01110 → Multiplicand

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$Q = 2^4 - 2^1$ $\quad (\overline{M}) = 10101$ (2's complement)

$M \times Q = M(2^4 - 2^1)$

$= 2^4(M) - M(2^1) = 2^4(M) + 2^1(\overline{M})$

$2^4(M) = 010110000 \rightarrow$ 4 0's ($2^4$)

$2^1(\bar{M}) = 101010 \rightarrow 1^{\bullet}0$ ($2^1$)

$$
\begin{array}{rl}
 & 2^4(M) = 0\,1\,0\,1\,1\,0\,0\,0\,0 \\
+ & 2^1(\bar{M}) = \boxed{1\,1\,1}\,\boxed{1\,0\,1\,0\,1\,0} \rightarrow \text{Pushed to right} \\
 & \Rightarrow \text{MSB 1 copied.}
\end{array}
$$

~~0 1 0 1 1 0 1 0~~

$\times$ 0 1 0 0 1 1 0 1 0

$\therefore$ The final answer is 0 1 0 0 1 1 0 1 0

i) First the number with the consicutive 1's are made the Multiplicand

ii) The negative sign is made the 2's complement

The number

$$M = 2^{i+1} - 2^j$$

where i is the position of MSB 1 and j is the position of LSB 1

iii) The power multiplied with M is the number of zeros to be added

iv) The number with the least number of bits is moved to the extreame right or LSB position

v) The remaining left bits are the MSB of the number copied to all positions

vi) The final answer is the sum of the two numbers.

Q6

Ans:

a) Simple Programmable Logic Device (SPLDs)

→ The IC which has number of logic Gates, Flipflop which are configured to users to implement function is Programmable IC's

→ The connections can be altered using Programming Process.

→ Simple Programming Process is to use fuse

→ Initially, All device has fuse intact.

→ Programming the device by blowing the fuse along the path to be removing for creating connections.

→ Simple Programmable device is ROM. Simplest ROM is PROM. An Electrically PROM uses MOS transistors, and changes can be made using high Voltage. It can be erased using another EPROM/ Exposing to UV light.

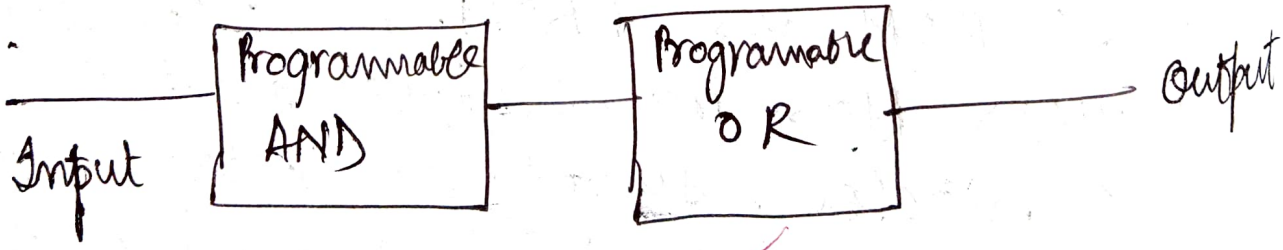
→ Another ROM is Masked ROM which can be Implemented on ASIC

## SPLD's

1) PROM → Programmable Read only Memory
2) PLA → Programmable Logic Array
3) PAL → Programmable Array Logic
4) GAL → Generic Array Logic

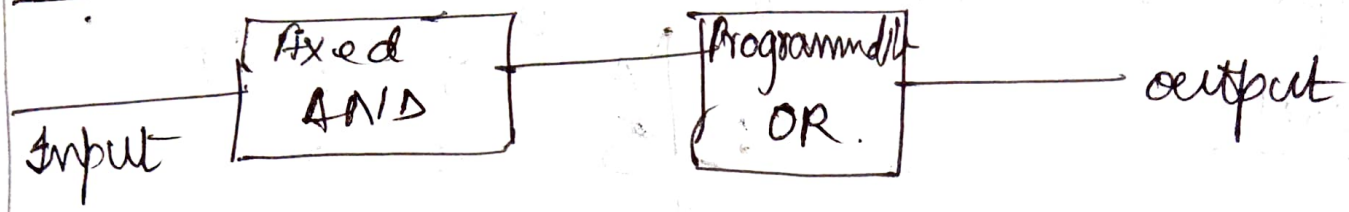
| Device | AND Gate | OR- Gate |
|---|---|---|
| PROM | Fixed | Programmable |
| PLA | Programmable | Programmable |
| PAL | Programmable | Fixed |
| GAL | Programmable | Fixed |

## PLA

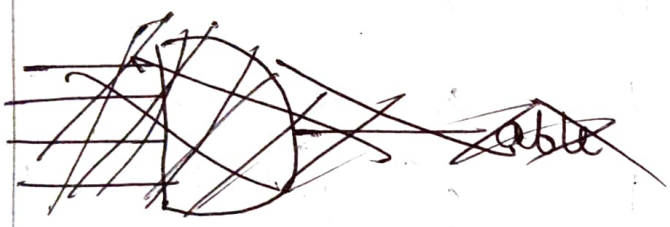


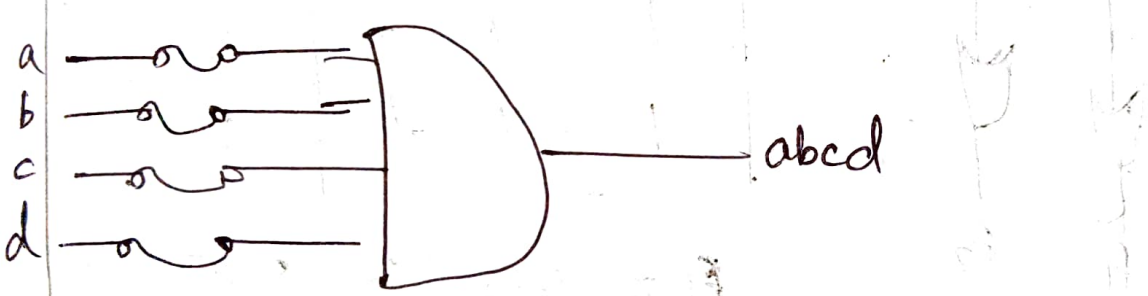

## PAL

PROM

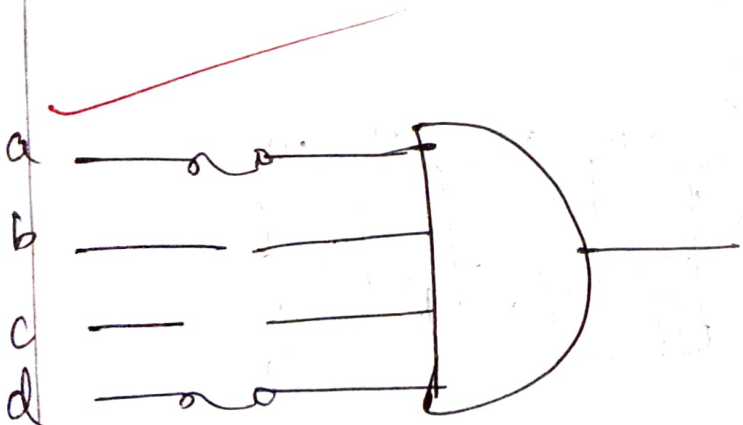



⇒ Programming Process



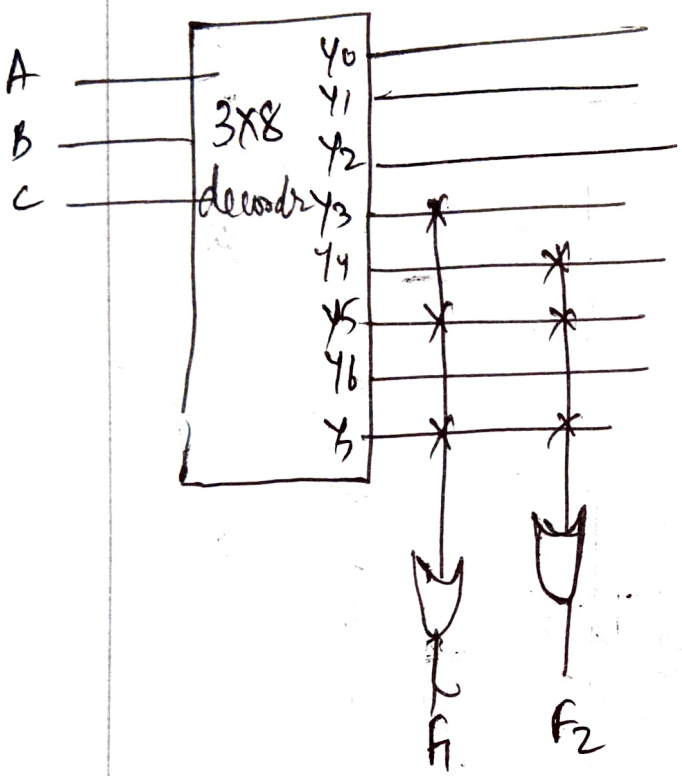

a) Before Programming




b) After Programming

Implement the following using PROM, PLA ?

- PROM

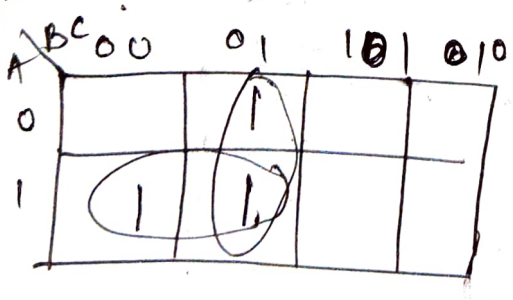



| | A | B | C | $F_1$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 |

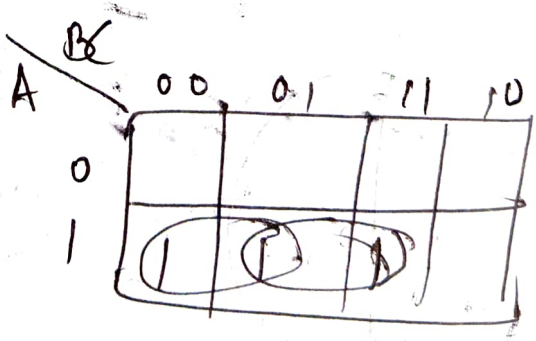
$F_1 = \Sigma m(3, 5, 7)$

$F_2 = \Sigma m(4, 5, 7)$

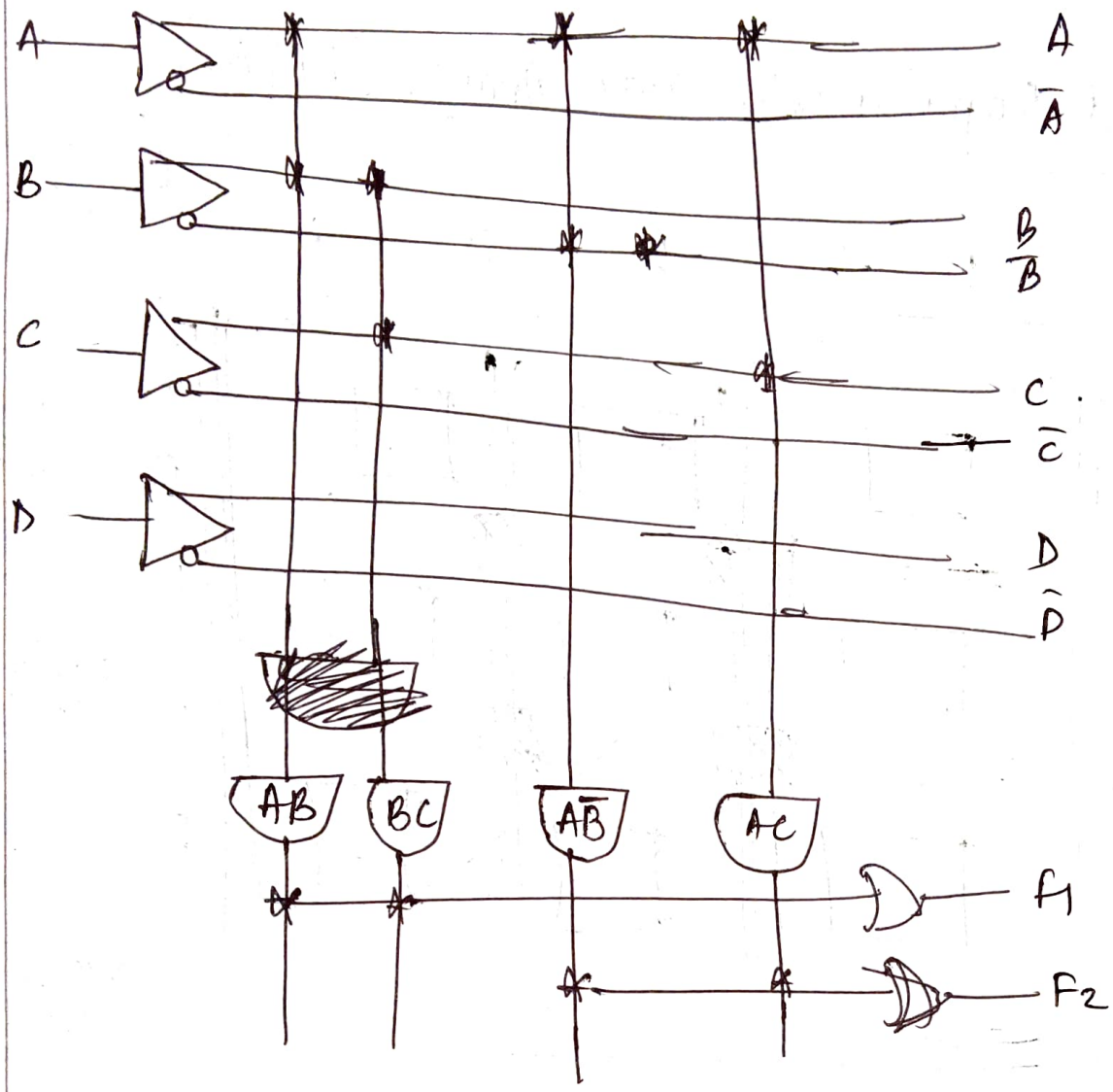
- PLA



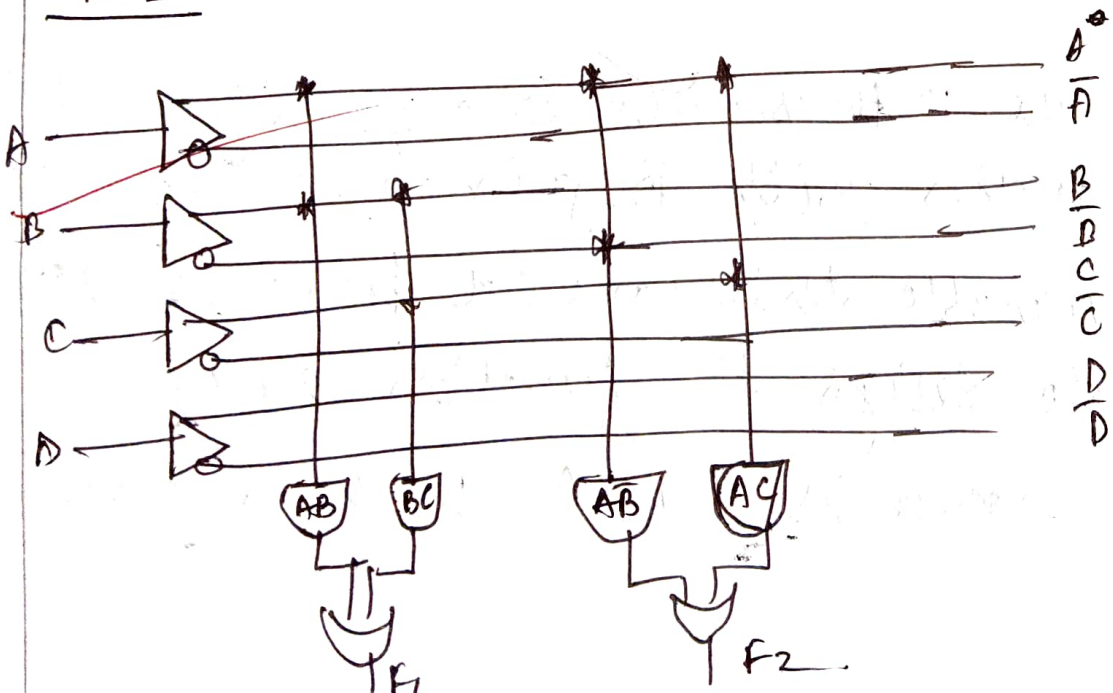

$F_1 = AB + BC$




$F_2 = A\bar{B} + AC$

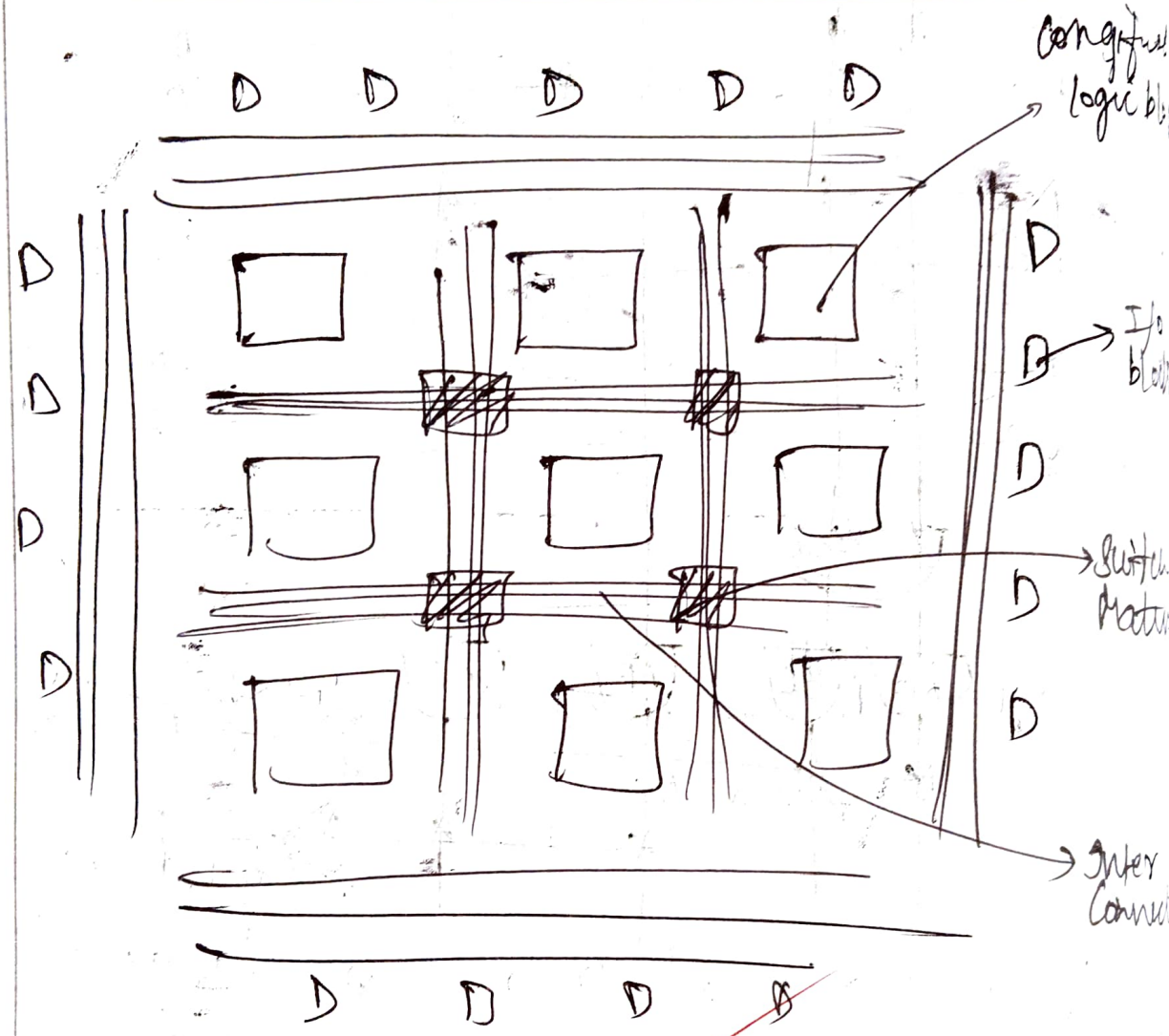

A
B
C
D
A
$\bar{A}$
B
$\bar{B}$
C
$\bar{C}$
D
$\bar{D}$
AB
BC
$A\bar{B}$
AC
$f_1$
$F_2$

PAL


A
B
C
D
A
$\bar{A}$
B
$\bar{B}$
C
$\bar{C}$
D
$\bar{D}$
AB
BC
AB
AC
$F_2$

Q6)

b) Field Programmable Gate Arrays (FPGAs)




→ FPGA's are 10 thousands to more than million logic devices with Programmable Interconnects

→ Programmable Interconnect provides users with to design the function easily

→ Above shown is FPGA block diagram

It has:

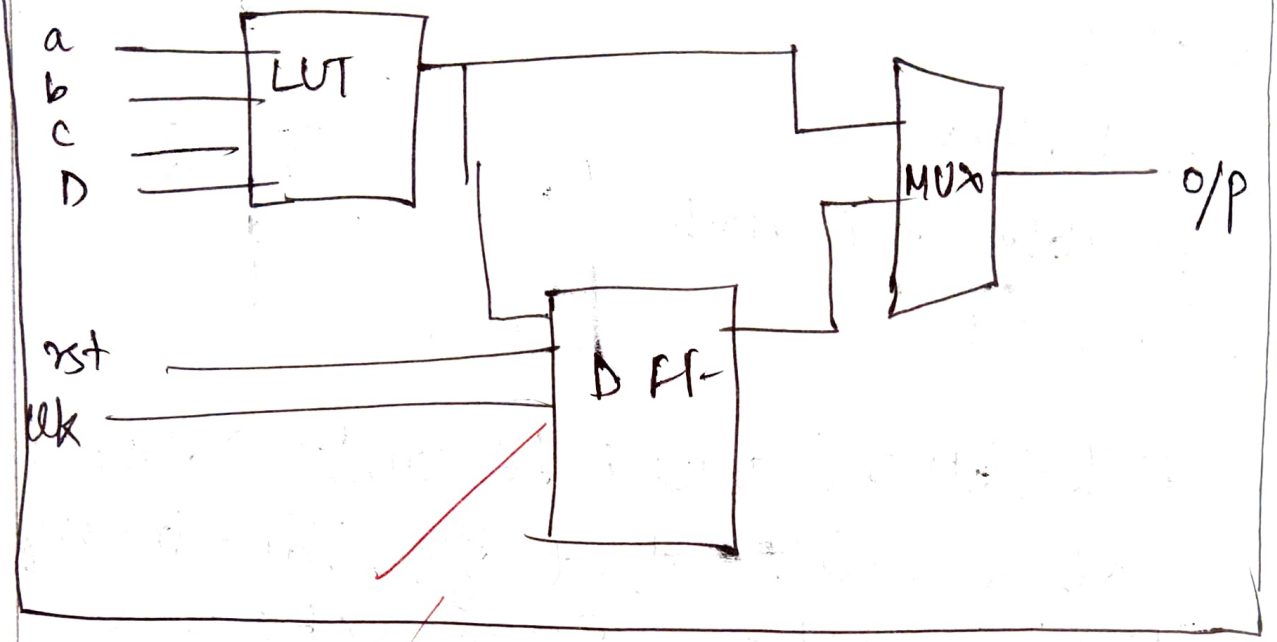
1) I/O blocks
2) Switch Matrix
3) Configurable logic block
4) Interconnect

1> I/O Block: It is used to communicate with external device. Used for data transfer / or gathering data from external device

2> Switch Matrix: Used to make path with Interconnect

3> Configurable logic block: It is a fundamental block of FPGA and called as 'Heart of FPGA'. Because All the functions are implemented here.

It has 3 parts: 1) Look up table
2> D-FF
3> MUX.




CLB

→ Look-up table : It is a 4-i/p device which functions are implemented based on Truth table

→ D-FF : It gives delayed version of o/p of LUT

→ MUX : It selects either the direct o/p or delay o/p of LUT.

4) Interconnect : It is used to connect to CLB's, horizontally and vertically. It is spread over entire board for easy transfer of data.