

UNIT-5 (Digital electronics)

Semiconductor memories & programmable logic devices.

I. Answer the following questions (1M).

1)a Classify the different types of programmable logic device.

Ans There are three major types of combinational PLDs and they differ in the placement of programmable connection in the AND-OR array.

① PALs (programmable array logic)

② PLAs (programmable logic array)

③ PROMs (programmable read only memory).

b what do you mean by volatile and nonvolatile memory?

Ans Volatile memory can retain the data as long as power is applied

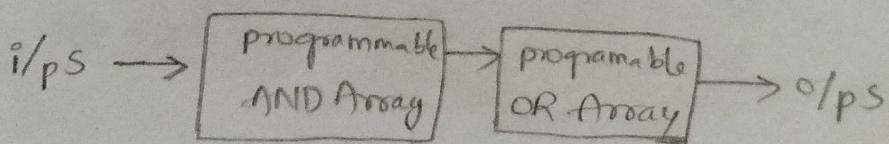
Ex:- RAM

Non volatile memory can hold data even the power blows off.

Ex:- ROM.

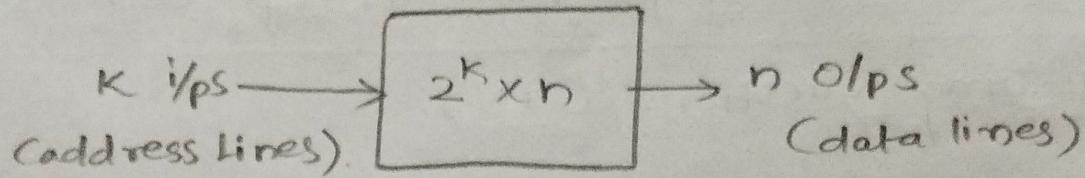
c what is programmable logic array? How it differs from ROM?

Ans the most flexible PLD is PLA, where both the AND array and OR array can be programmed.



The ROM has fixed AND array as decoders and the

programmable OR array. It consists of  $K$  I/ps &  $n$  O/ps.



d State the advantages of FPGA.

FPGAs excel at parallel processing.

- \* They offer superior performance compared to a general CPU.
- \* FPGAs are cost effective
- \* They can accelerate product development, allowing for quick market release.

e Distinguish between PAL and PLA.

| A:- | PAL  | PLA  |
|-----|--|--|
|     | <ul style="list-style-type: none"><li>1. Here OR array is fixed and AND array is programmable.</li><li>2. PAL is cheaper.</li><li>3. PAL is simpler.</li></ul> | <ul style="list-style-type: none"><li>1. Both AND and OR array is programmable.</li><li>2. PLA is costlier</li><li>3. PLA is more complex than PROM.</li></ul> |

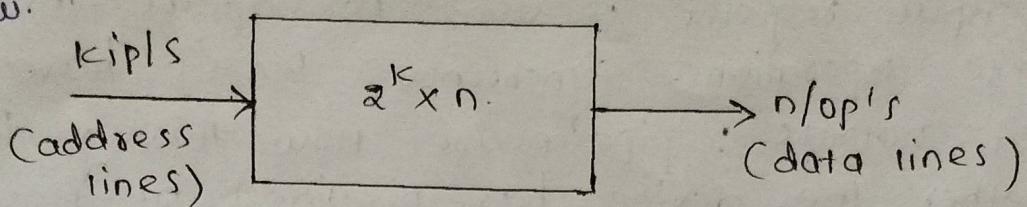
## UNIT - I

3 Marks

2a) Explain PROM

### PROM

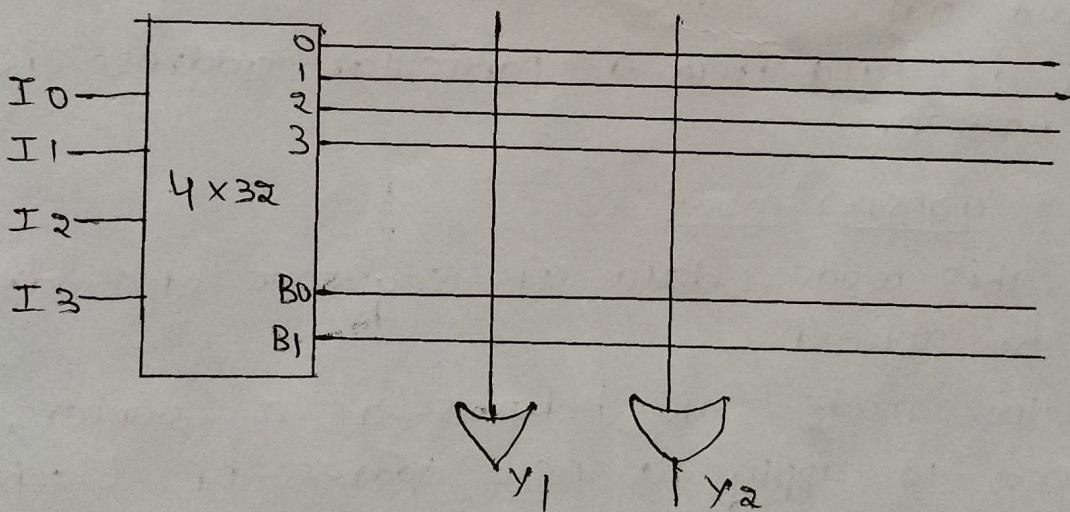
The block diagram of ROM is shown in the figure below.



- \* It consists of  $k$  ips and  $n$  op's
- \* Fixed AND array + programmable OR array

\* size of ROM is  $2^k \times n$

Ex :- if  $k=4$ ,  $n=2$ , then the size of ROM is  $2^4 \times 2$   
i.e decoder of size  $4 \times 32$ . is used and two  
OR gates are used.



2b) Discuss the basic concepts and the principle of operation of Bipolar SRAM cell.

### Static RAM (SRAM) :-

The synchronous SRAM's use clock interface for control, address and data as shown in the figure. It consists of edge triggered registers, address

- Address Registers (AR) and Control Registers (CR) to store the address and control information, respectively. Due to these registers it is possible that an operation that is set up before the rising edges of the clock is performed internally during the subsequent clock period. Input register is used to capture the I/P data for write operations. Synchronous RAM's have two types of outputs : pipelined or flow through. Depending on whether the device has pipelined or flow through O/P's, output register (OR) is not provided to hold the O/P data from a read operation.

Operating Modes :- There are 2 operating modes

1. Single transfer mode
2. Burst transfer mode.

f. Single transfer Mode :-

- In this mode, only one read or write cycle is carried out
- That is data from a particular address is read or written.

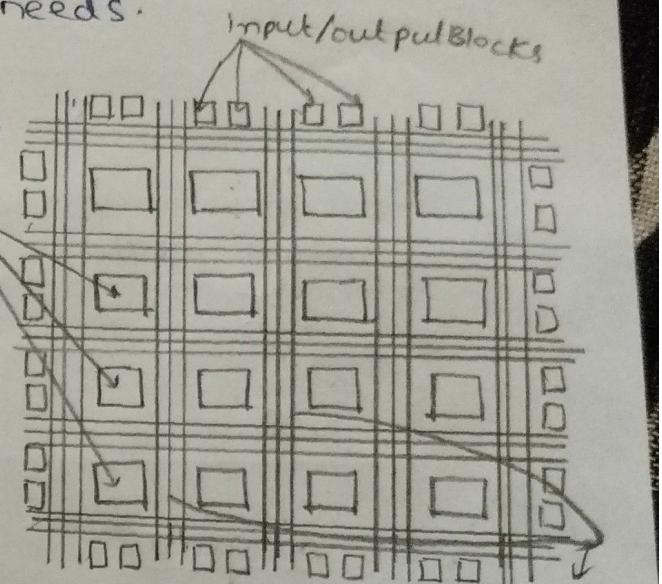
g. Burst transfer mode :-

- In this mode, data at sequence of addresses is read or written
- In this mode, AR behaves as a counter, eliminating the need to apply a new address of each cycle.

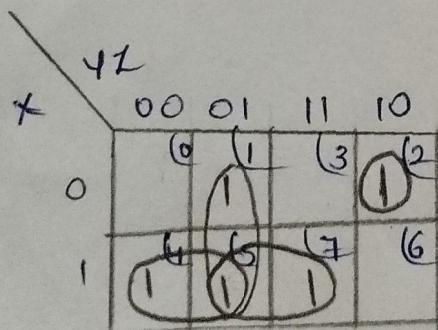
2c) Explain in brief about FPGA

### FPGA

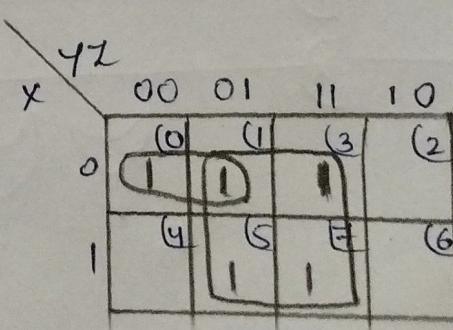
- A Field programmable Gate Array (FPGA) is an integrated circuit (IC) that can be reprogrammed after manufacturing to meet specific needs.
- FPGAs are a subset of programmable logic devices (PLD's). Field programmable gate arrays (FPGAs) are integrated circuits (ICs) that are made up of several components, including:
- Configurable logic blocks (CLBs): The primary component of an FPGA, CLBs contain logic gates, flip-flops, multiplexers, and look up tables (LUTs). Flip-flops are used to store data, while LUT are a crucial component of the FPGA architecture.
- Programmable interconnects: These allow designers to connect and configure blocks to perform various function
- Programmable input/output blocks (IOBs): These are another component of an FPGA.



2b)d) Demonstrate the realization of the following function using PAL  $F(x,y,z) = \Sigma(1,2,4,5,7)$ , and  $F_2(x,y,z) = \Sigma(0,1,3,5,7)$ .



$$F_1 = \bar{X}y\bar{Z} + \bar{Y}z + \bar{X}\bar{Y} + Xz$$



$$F_2 = z + \bar{X}\bar{Y}$$

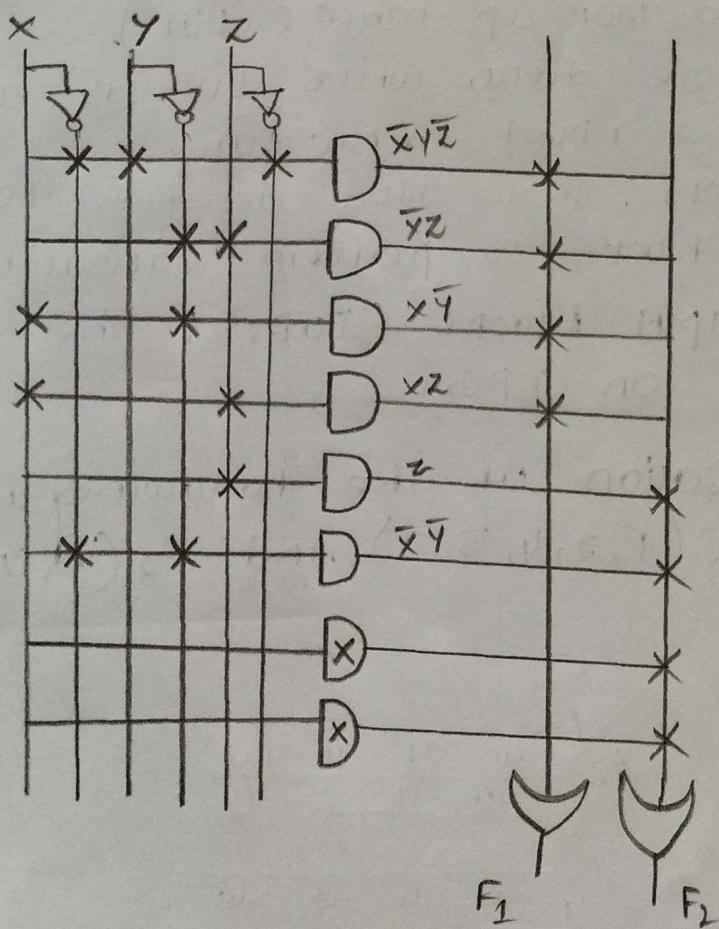
## Programmable Table

| Product term | AND gate i/p |   |   | O/P's (F1)       |
|--------------|--------------|---|---|------------------|
|              | x            | y | z |                  |
| 1            | 0            | 1 | 0 | $\bar{x}yz$      |
| 2            | -            | 0 | 1 | $\bar{y}z$       |
| 3            | 1            | 0 | - | $x\bar{y}$       |
| 4            | 1            | - | 1 | $xz$             |
| Product term | i/p          |   |   | (F2)             |
| 5            | -            | - | 1 | $z$              |
| 6            | 0            | 0 | - | $\bar{x}\bar{y}$ |

} F1

} F2

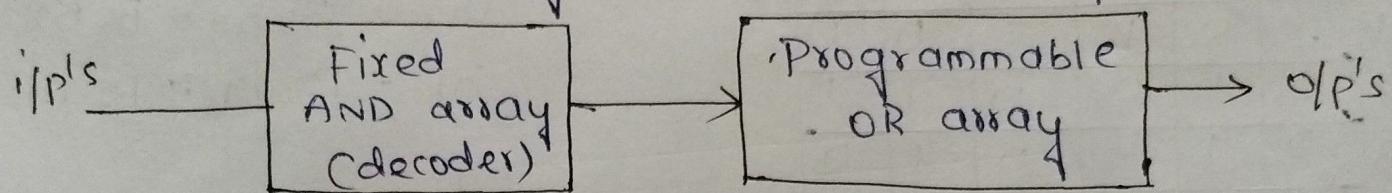
## Circuit diagram



(e) Classify types of PLD's and write notes on PLD's  
There are three major types of combinational PLD's and they differ in the placement of programmable connection in the AND-OR array

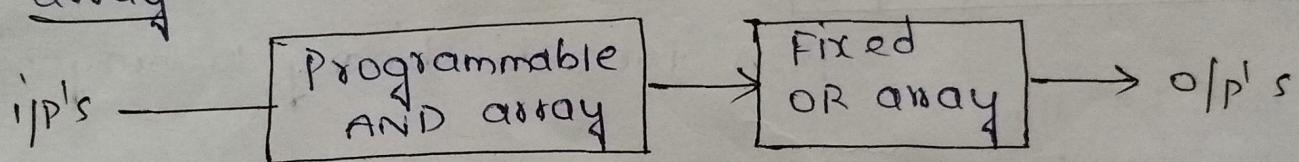
- ① PALs (programmable array logic)
- ② PLAs (programmable logic array)
- ③ PROMs (programmable read only memory)

\* PROM has fixed AND array constructed as a decoder, and programmable OR array



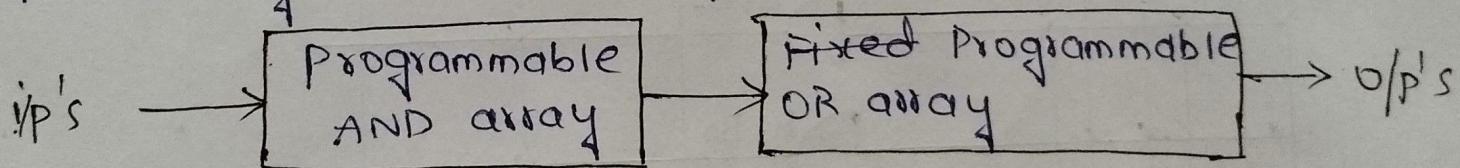
(a) PROM.

\* PAL has programmable AND array and a fixed OR array



(b) PAL

\* The most flexible PLD is PLA, whose both the AND array and OR array can be programmed.



(c) PLA.

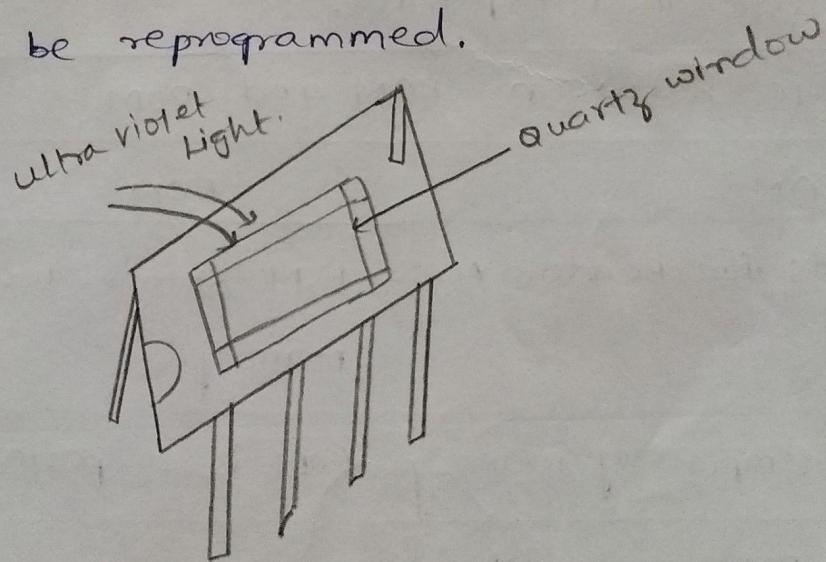
III  
Answer the following questions ? (5M).

3)a

write a short note on (i) EEPROM (ii) EEPROM.

(i) EEPROM : (Erasable programmable Read only memory) :

- \* EEPROMs use MOS circuitry. they store 1s and 0s as packet of charge in buried layer of IC chip.
- \* EEPROMs can be programmed by the user with a special EEPROM programmer.
- \* We can erase the stored data in the EEPROM by exposing the chip to ultraviolet light through the quartz window for 15 to 20 min.
- \* It is not possible to erase selective information, when erased the entire information is lost.
- \* the chip can be reprogrammed.



- \* This memory is suitable for product development experimental projects and college laboratories , since it can be reused many times.

(ii) EEPROM: (Electrically erasable programmable ROM);

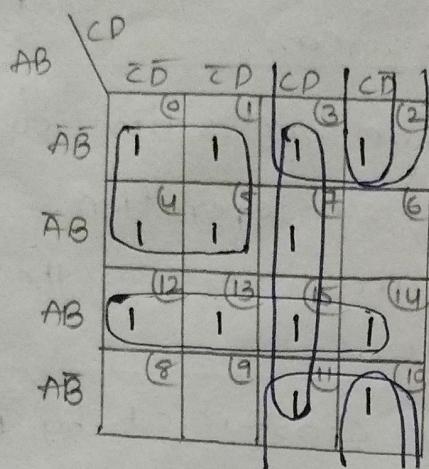
- \* EEPROMs also use MOS circuitry very similar to that of EPROM.
- \* The data is stored as charge (or) no charge on an insulated layer (or) an insulated floating gate in the device.
- \* The insulated layer is made very thin ( $< 200\text{A}$ )
- \* EEPROM allows selective erasing at the register level rather than erasing all information since the information can be changed by using electric signals.
- \* EEPROM also has a special chip erase mode by which entire chip can be erased in 10ms.
- \* EEPROMs are most expensive.

3 b) Differentiate between RAM and ROM.

| Ans | RAM   | ROM   |
|-----|---|---|
|     | RAM stands for Random Access Memory                                   | ROM stands for Read only Memory.                                |
|     | RAM is a temporary memory   | ROM is a permanent memory                                       |
|     | The data in RAM can be changed or deleted                             | The instructions written in ROM cannot be changed or deleted.   |
|     | RAM is a volatile memory  | ROM is a non-volatile memory                                    |
|     | The instructions are written in to the RAM at the time of Execution . | The instructions are written into the ROM at Manufacturing time |

3C) Implement the following function using PAL when the function  $F = \sum m(0, 1, 2, 3, 4, 5, 7, 10, 11, 12, 13, 14, 15)$ .

Sol:- K-map :-

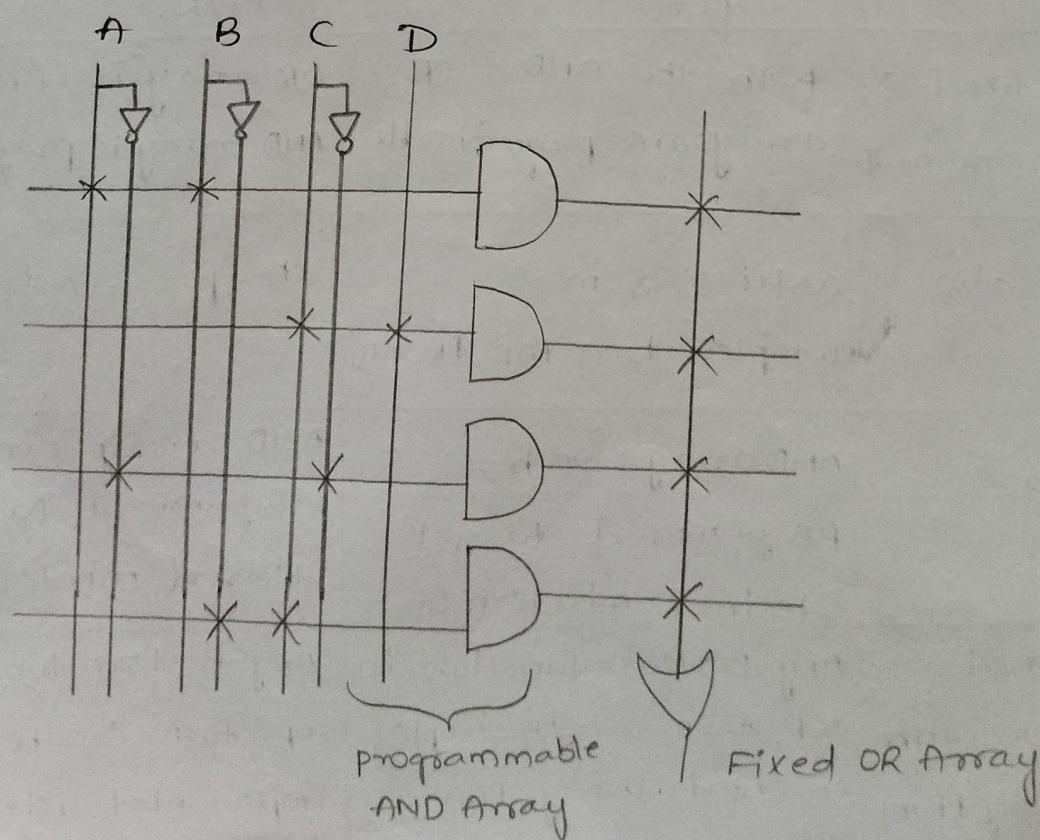


$$F = AB + CD + \overline{AC} + \overline{BC}$$

Programming table :-

| Product term. | AND gate I/p's     | outputs   |
|---------------|--------------------|---|
| 1             | A B C D<br>1 1 - - |   |
| 2             | - - - 1            |   |
| 3             | 0 - 0 -            | $F = AB + CD + \overline{AC} + \overline{BC}$ . |
| 4             | - 0 1 -            |   |

Realization of the given function using PAL :



3d) Differentiate static and Dynamic RAM.

| A: | Static RAM   | Dynamic RAM.   |
|----|--|--|
| ①  | this contains less memory cells per unit area.   | This contains more memory cells per unit area.   |
| ②  | It has less access time. so these are faster (time required to access the data is less). | It has more access time so, these are slower in operation. (time required to access the data is more). |
| ③  | These are constructed by flip-flops.   | These are constructed by MOS transistors.  |
| ④  | cost is more.  | cost is less.  |
| ⑤  | It requires no refreshment   | it requires continual refreshment  |
| ⑥  | package density is low   | package density is high.   |
| ⑦  | HW complexity is less  | HW complexity is high.   |

3e) write the comparison between PROM, PLA, PAL?

|    | PROM  | PLA  | PAL.   |
|----|---|--|--|
| 1. | AND array is fixed & OR array is programmable                             | Both the AND & OR arrays are programmable                      | OR array is fixed & AND array is programmable                  |
| 2. | cheaper and also simple to use  | costliest & more complex than PAL, PROMs.                      | cheaper and simpler.   |
| 3. | All minterms are decoded,   | AND array can be programmed to get desired minterms.           | AND array can be programmed to get desired minterms.           |
| 4. | only Boolean functions in standard SOP form can be implemented using PROM | Any Boolean function in SOP form can be implemented using PLA. | Any Boolean function in SOP form can be implemented using PAL. |

4a) Explain the functions and applications of PLAs in memory addressing and implement the following two Boolean functions with a PLA.

$$F_1(A, B, C) = \sum(0, 1, 3, 5) \text{ and } F_2(A, B, C) = \sum(1, 2, 4, 7)$$

PLA

$$F_1(A, B, C) = \sum(0, 1, 3, 5)$$

$$F_2(A, B, C) = \sum(1, 2, 4, 7)$$

K map for  $F_1$

|  |   | $\bar{B}C$ | $\bar{B}\bar{C}$ | $B\bar{C}$ | $BC$ | $B\bar{C}$ |
|--|---|------------|------------------|------------|------|------------|
|  |   | 00         | 01               | 11         | 10   | 11         |
|  |   | A          | 0                | 1          | 1    | 0          |
|  | 0 | 1          | 0                | 1          | 1    | 0          |
|  | 1 | 0          | 1                | 0          | 0    | 1          |

$$\bar{A}\bar{B} + \bar{A}\bar{C} + \bar{B}\bar{C}$$

K-map for  $F_2$

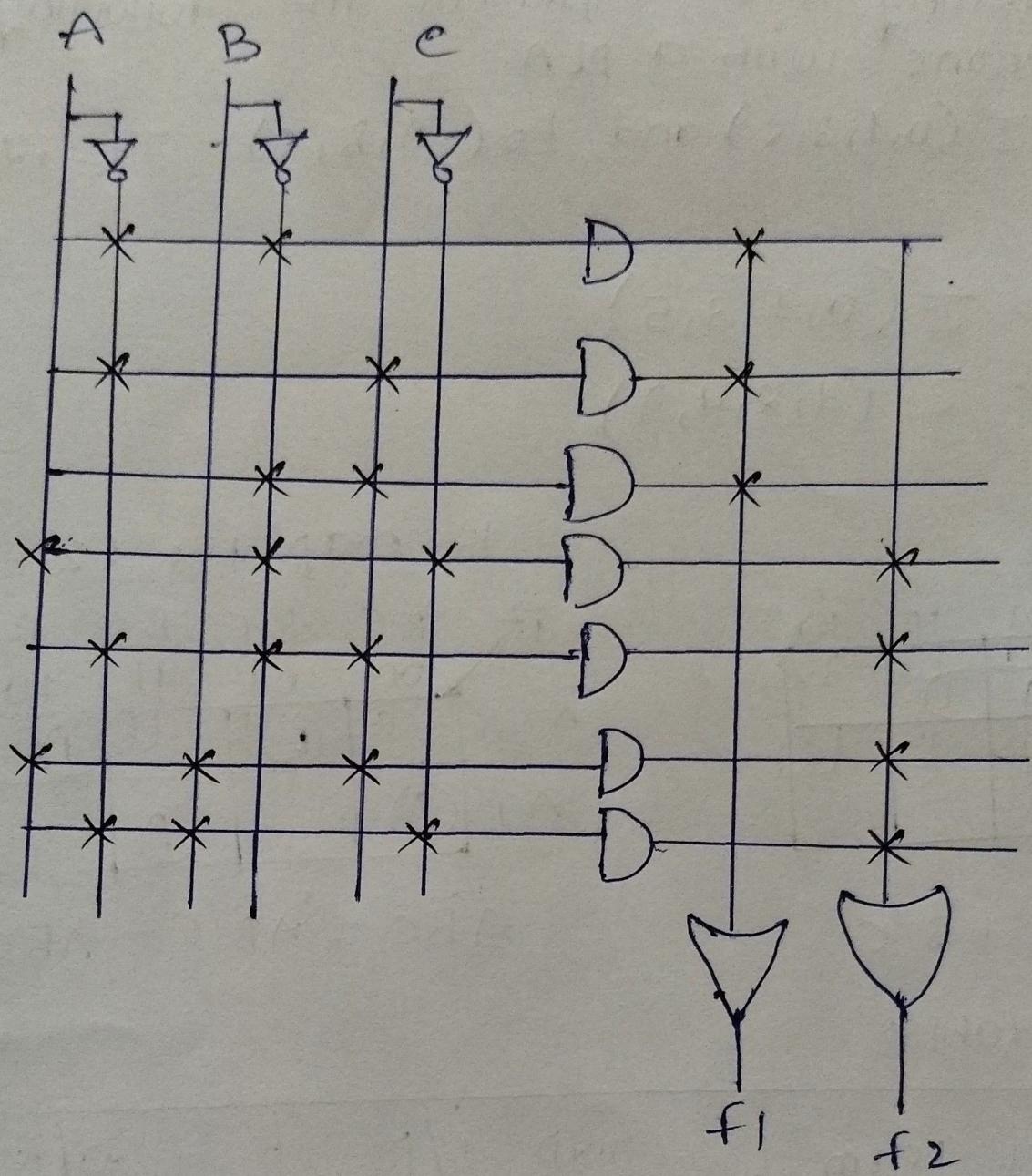
|  |   | $\bar{B}\bar{C}$ | $\bar{B}C$ | $BC$ | $B\bar{C}$ |   |
|--|---|------------------|------------|------|------------|---|
|  |   | 00               | 01         | 11   | 10         |   |
|  |   | A                | 0          | 1    | 1          | 0 |
|  | 0 | 0                | 1          | 1    | 0          | 1 |
|  | 1 | 1                | 0          | 0    | 1          | 0 |

$$A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}BC$$

Programming table

| Product term | AND i/p |   |   | O/P   |
|--------------|---------|---|---|---|
|              | A       | B | C |   |
| 1            | 0       | 0 | - | $f_1 = \bar{A}\bar{B} + \bar{A}C + \bar{B}C$                |
| 2            | 0       | - | 1 |   |
| 3            | -       | 0 | 1 | $f_2 = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}BC$ |
| 4            | 1       | 0 | 0 |   |
| 5            | 0       | 0 | 1 |   |
| 6            | 1       | 1 | 1 |   |
| 7            | 0       | 1 | 0 |   |

logic diagram :-



10 Marks

4b)

Design 3 bit binary to Gray code converter using PLA?

Sol

Three Binary Bits  $B_3 \ B_2 \ B_1$

Three Gray Bits  $G_3 \ G_2 \ G_1$

Truth table for 3-bit binary to gray conversion is

| Binary |       |       | Gray  |       |       |
|--------|-------|-------|-------|-------|-------|
| $B_3$  | $B_2$ | $B_1$ | $G_3$ | $G_2$ | $G_1$ |
| 0      | 0     | 0     | 0     | 0     | 0     |
| 1      | 0     | 0     | 0     | 0     | 1     |
| 2      | 0     | 1     | 0     | 0     | 1     |
| 3      | 0     | 1     | 1     | 0     | 1     |
| 4      | 1     | 0     | 0     | 1     | 0     |
| 5      | 1     | 0     | 1     | 1     | 1     |
| 6      | 1     | 1     | 0     | 1     | 0     |
| 7      | 1     | 1     | 1     | 1     | 0     |

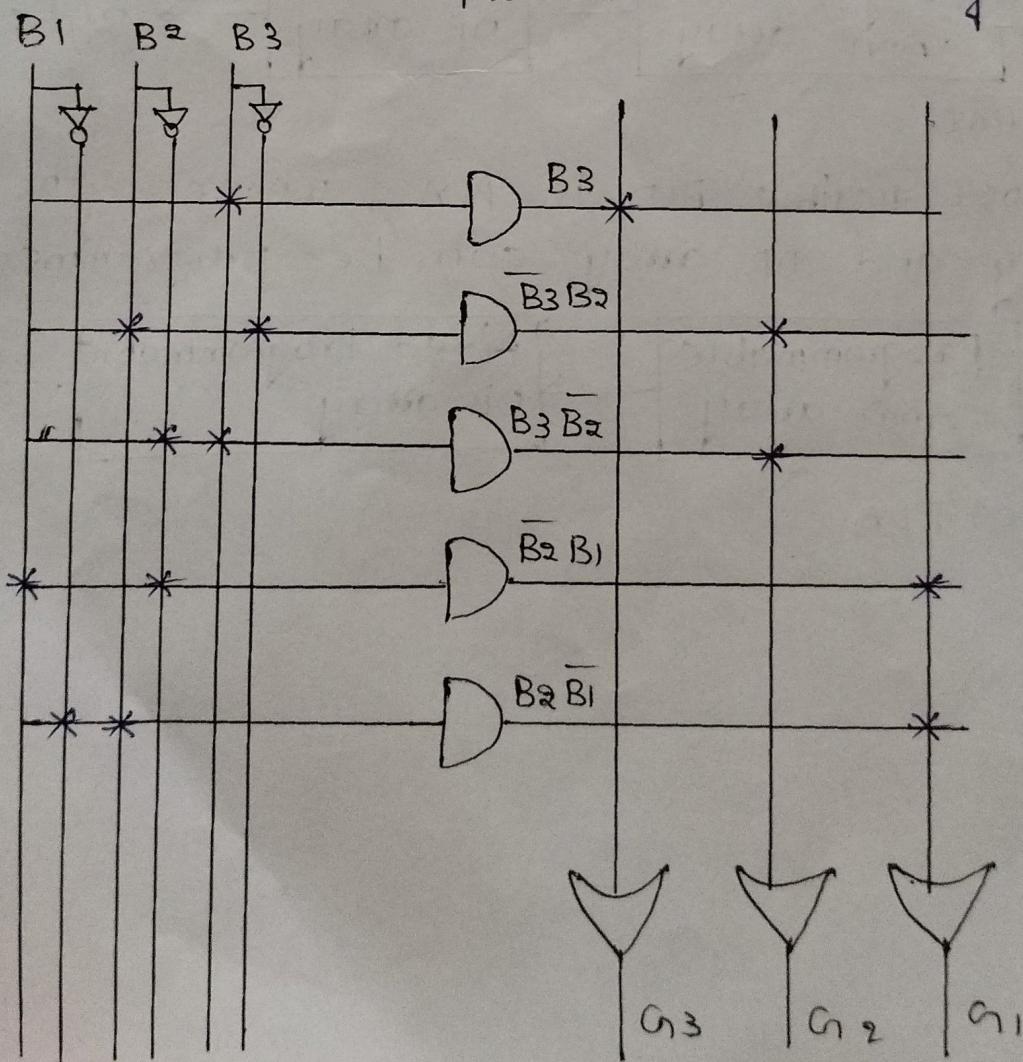
on simplification,

$$G_3 = B_3$$

$$G_2 = B_3 \bar{B}_2 + \bar{B}_3 B_2$$

$$G_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$$

PLA for 3bit binary to Graycode.



PLA Table for the above is

| Product term                     | i/p's          |                |                | o/p's          |                |                |
|----------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|
|                                  | B <sub>3</sub> | B <sub>2</sub> | B <sub>1</sub> | C <sub>3</sub> | C <sub>2</sub> | C <sub>1</sub> |
| 1. B <sub>3</sub>                | 1              | -              | -              | 1              | -              | -              |
| 2. B <sub>3</sub> B <sub>2</sub> | 1              | 0              | -              | -              | 1              | -              |
| 3. B <sub>3</sub> B <sub>2</sub> | 0              | 1              | -              | -              | 1              | -              |
| 4. B <sub>2</sub> B <sub>1</sub> | -              | 1              | 0              | -              | -              | 1              |
| 5. B <sub>2</sub> B <sub>1</sub> | -              | 0              | 1              | -              | -              | 1              |

4c) Analyze a combinational circuit using ROM. The circuit accepts a three-bit number and outputs a binary number equal to the square of the input number.

Step 1 :

| Inputs         |                |                | Outputs        |                |                |                |                |                | Decimal |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---------|
| A <sub>2</sub> | A <sub>1</sub> | A <sub>0</sub> | B <sub>5</sub> | B <sub>4</sub> | B <sub>3</sub> | B <sub>2</sub> | B <sub>1</sub> | B <sub>0</sub> |         |
| 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0       |
| 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 1              | 1       |
| 0              | 1              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 4       |
| 0              | 1              | 1              | 0              | 0              | 1              | 0              | 0              | 1              | 9       |
| 1              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 16      |
| 1              | 0              | 1              | 0              | 1              | 1              | 0              | 0              | 1              | 25      |
| 1              | 1              | 0              | 0              | 1              | 0              | 1              | 0              | 0              | 36      |
| 1              | 1              | 1              | 1              | 1              | 0              | 0              | 0              | 1              | 49      |

Step - 2 :-

$$B_5 = \sum m(6,7)$$

$$B_1 = 0$$

$$B_4 = \sum m(4,5,7)$$

$$B_0 = \sum m(1,3,5,7)$$

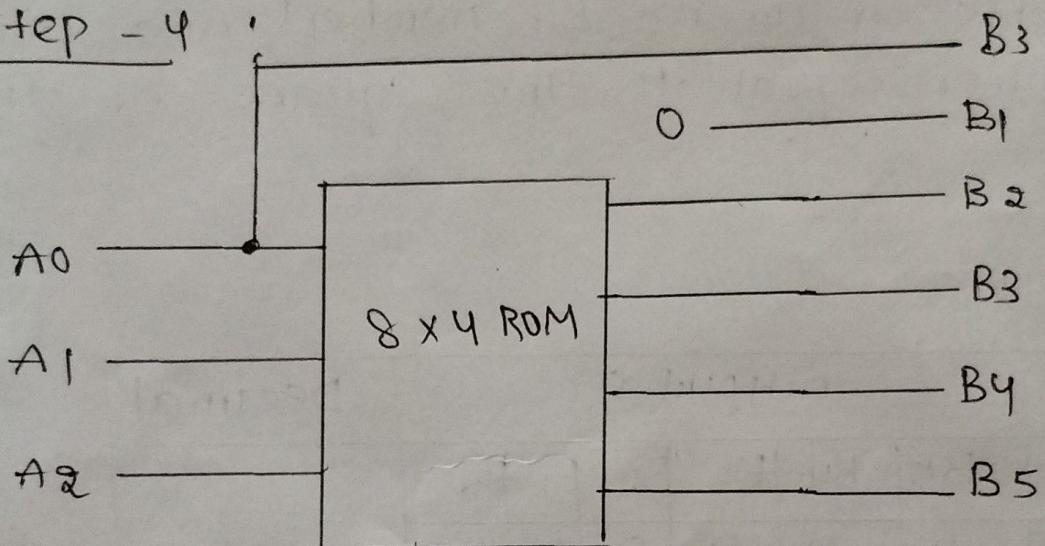
$$B_3 = \sum m(3,5)$$

$$B_2 = \sum m(2,6)$$

Step - 3 : ROM Truth Table

| A <sub>2</sub> | A <sub>1</sub> | A <sub>0</sub> | B <sub>5</sub> | B <sub>4</sub> | B <sub>3</sub> | B <sub>2</sub> |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0              | 0              | 1              | 0              | 0              | 0              | 0              |
| 0              | 1              | 0              | 0              | 0              | 0              | 1              |
| 0              | 1              | 1              | 0              | 0              | 1              | 0              |
| 1              | 0              | 0              | 0              | 1              | 0              | 0              |
| 1              | 0              | 1              | 0              | 1              | 1              | 0              |
| 1              | 1              | 0              | 1              | 0              | 0              | 1              |
| 1              | 1              | 1              | 1              | 1              | 0              | 0              |

Step - 4



Block diagram