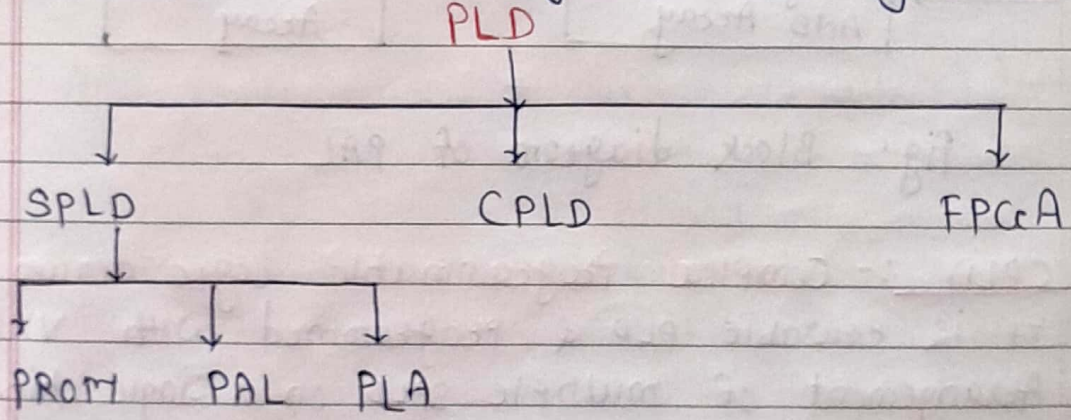
Unit No.5 VHDL

CVHSIC - Very High speed Integrated Circuit Hardware description language

- **Programmable logic Devices :-**
  - PLD are basicaly integrated circuit (IC) with AND array at I/P & OR array at O/P.
  - Logic devices - Fixed & Programmable.
  - PLD can be programmed according to the user requirments.

- **Advantages of PLD's :-**
  - Less design time
  - Low power requirements
  - Reduced space
  - High design security
  - Low production cost.
  - Flexible & can be reprogrammed quickly.
  - High switching speed.

- **Classification of programmable logic devices:-**



- **SPLD (Simple Programmable logic devices) :-**
  - Fixed function & applicable for ckt with limited gates.
  - Low cost & High pin to pin performance.

PROM :-

AND - OR array on single chip.
(fixed) (Programmable)

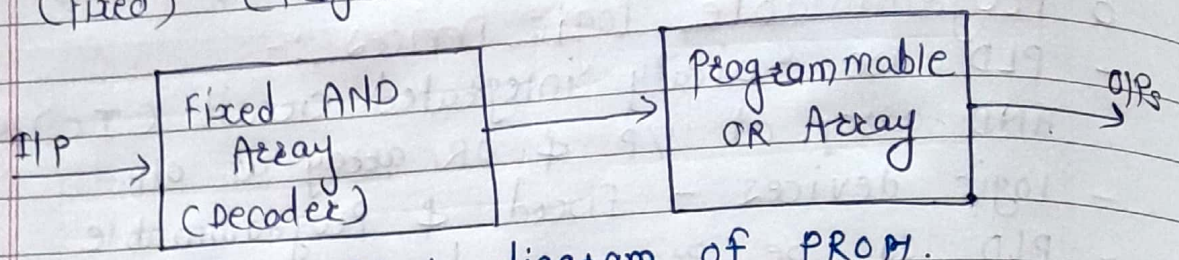


fig :- Block diagram of PROM.

PLA :-

AND array & OR array both are programmable.

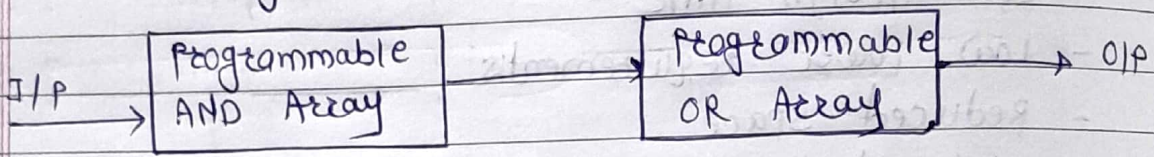


fig :- Block diagram of PLA.

PAL :-

AND array programmable & OR array fixed.

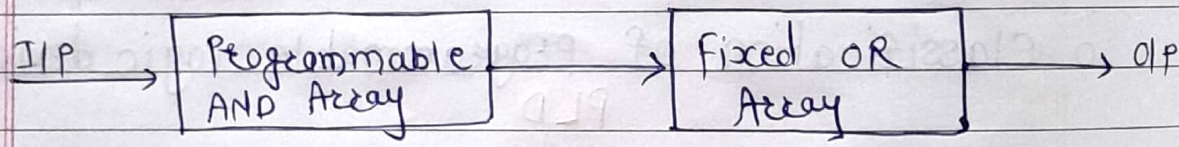


fig :- Block diagram of PAL

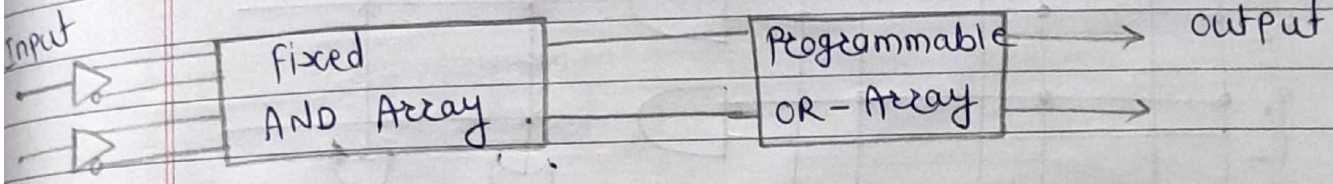
- CPLD :- Complex programmable logic devices
  - It is erasable PLD & programmed with VHDL coding.
  - Arrangement of multiple SPLD on a single chip.
  - EPROM & EEROM Technologies.

- FPGA :- field programmable gate Array :-
  - suitable for complex applications, wide density range.
  - It is very large system integrated chip.

○ Detail architecture, study of PROM, PAL, PLA:-

- A PROM is a memory device in which the AND array is fixed (like decoder) and the OR array is programmable.
- It is one-time programmable; once programmed, the stored information can not be changed.

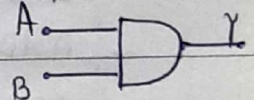
Architecture of PROM :-



Ex :- $F_1 = \Sigma m(0, 1, 2, 5, 7)$ → (5) digits
$F_2 = \Sigma m(1, 2, 4, 6)$ → (4) digits

OR gates = No. of eqn
= 2 output

AND gates = 5 (Max^m bit's select)



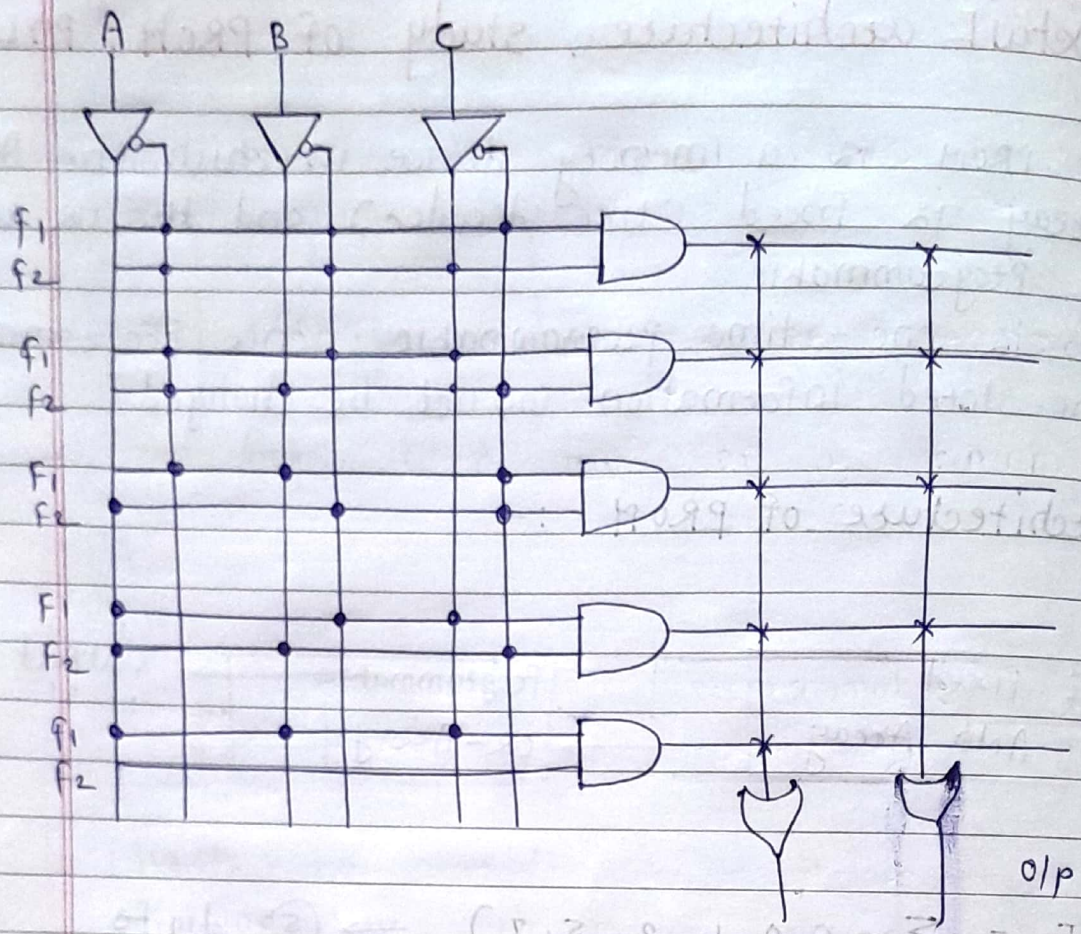
Highest bit = (7) → Binary

8	4	2	1
0	1	1	1

3 bit = Input

	4	2	1
0			
1	0	0	1
2	0	1	0
3	0	1	1
4			

I/P



Advantage :-

- simple, easy to use.
- Fast operation.

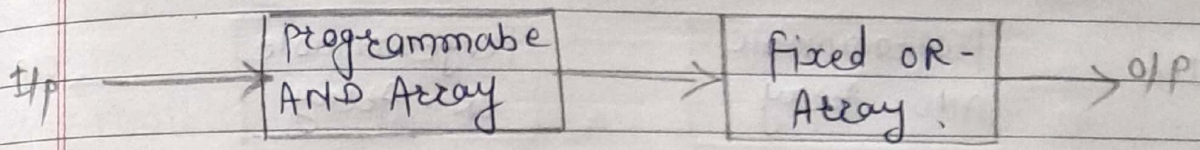
Limitation :-

- AND array is fixed → less flexible.
- one time programmable.

PAL (Programmable Array Logic) :-

- PAL was developed after PROM to provide more flexibility.
- It has a programmable AND array & a fixed OR array.
- This means we can choose the product terms but the number of OR terms is fixed.

Architecture :-



- Inputs are provided to a programmable AND-Array, where fuses determines which input variable form each product term.
- The product terms are connected to a fixed OR Array which sum specific product terms to generate o/ps.
- output may be combinational or registered

Advantages :-

- Faster than PLA (Because OR Array is fixed)
- simple to manufacture.
- High speed operation.

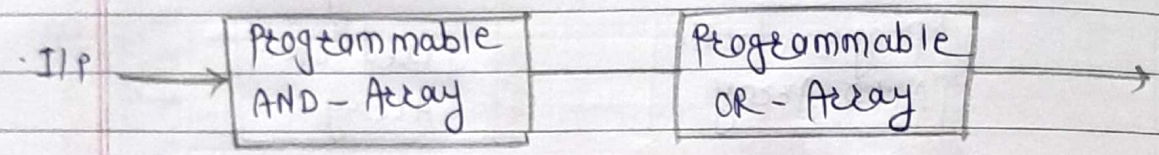
Disadvantages :-

- Limited flexibility.
- Not reprogrammable.

PLA (Programmable logic Array) :-

- PLA provides the highest flexibility among all simple PLD's.
- Both AND & OR array's are programmable.
- Thus any logic fun? can be implemented easily.

Architecture :-



- Inputs are connected to a programmable AND array, allowing the creation of specific product terms.
- These product terms are fed into a programmable OR Array, where the final sum of product expressions are formed.
- The O/P can be directly connected with flfs.

Advantages :-

- Most flexible PLD.
- Reprogrammable
- Suitable for implementing multiple logic fun?

Disadvantages :-

- Slower & more complex than PAL.
- More expensive to produce.

○ Comparison of PROM, PAL & PLA :-

	PROM	PAL	PLA
1.	AND array is fixed & OR array is Programmable.	OR Array is fixed & AND array is Programmable.	Both AND & OR arrays are Programmable.
2.	cheaper	Moderately expensive	Expensive
3.	SOP fun^n in std forms only implement	Any SOP fun^n can be implement	Any SOP fun^n can be implemented.
4.	Possible to decode any minterms	can produced desired minterms	can produced desired minterms.

○ General Architecture of FPGA and CPLD :-

* Field Programmable Gate Arrays (FPGA) :-

- FPGA is an integrated circuit which consists of an array of logic cells with programmable interconnections.
- The logic cells & interconnection can be programmed by user for the desired application.
- FPGA configuration is generally described using a Hardware Description language similar to appl^n specific Integrated circuit (ASIC).
- The general structure of FPGA is shown in below fig. It consists of three major sections such as configurable logic block, programmable interconnect array, & programmable I/O cells.

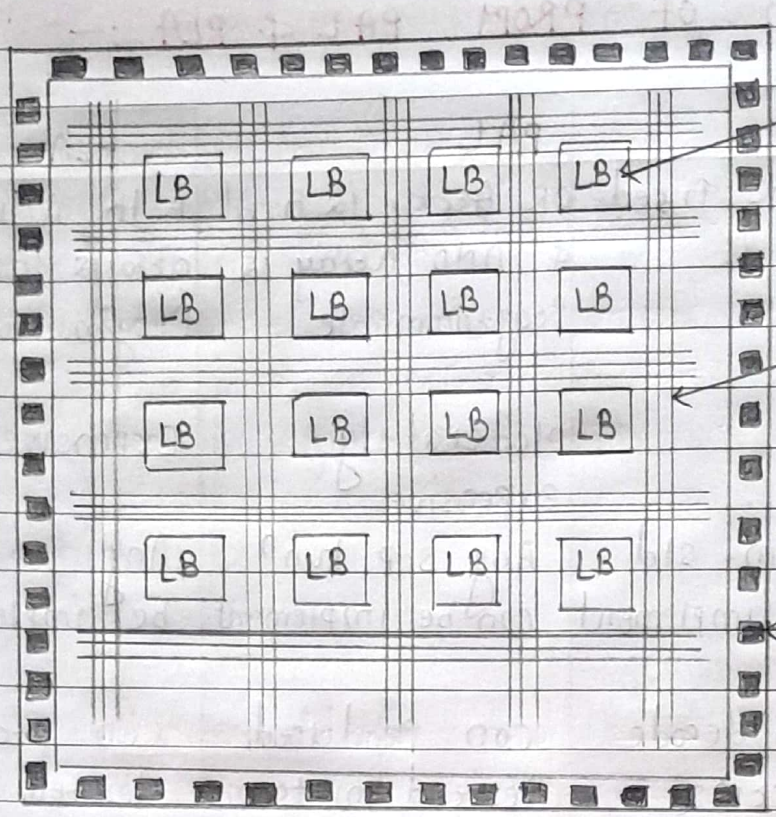


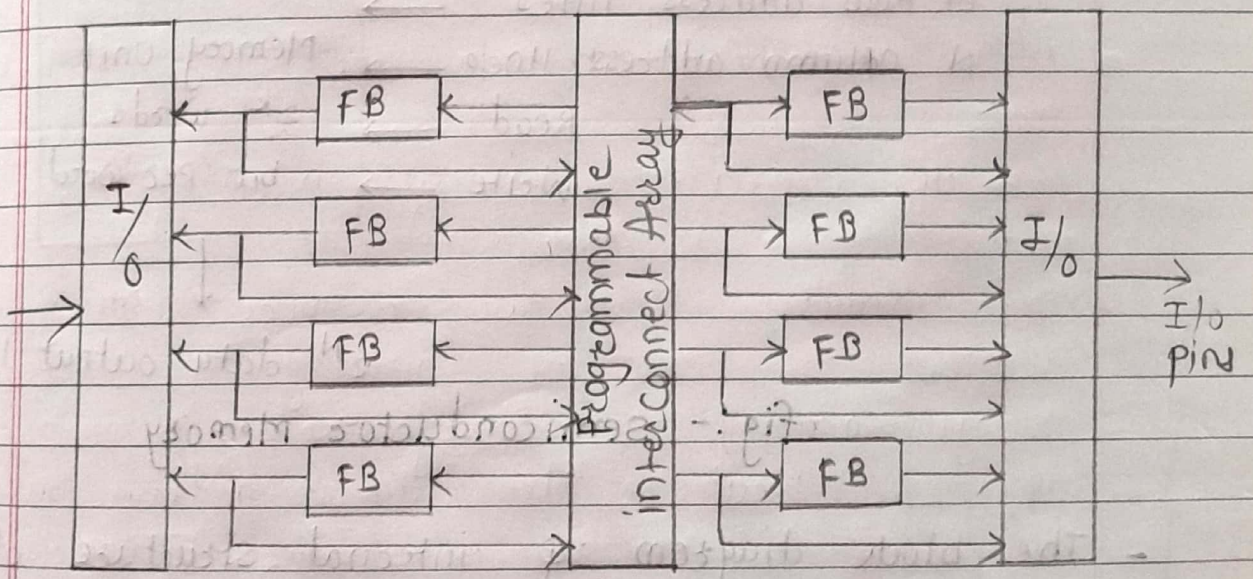
fig :- Block dia. of FPGA

- **Feature of FPGA :-**
  1. Programmability :-
  2. Parallel processing.
  3. Scalability.
  4. High speed operation.
  5. Low latency.

- **Application of FPGAs :-**
  1. Digital signal processing (eg. Audio, video & image processing)
  2. Embedded system (eg. IOT devices, robotics).
  3. Communication systems (eg. 5G network switches).
  4. Machine learning (eg. Neural networks).
  5. Aerospace & defence (eg. Radar).

* Complex Programmable logic Devices (CPLD) :-

- The extented improvement of integrated circuit (IC) technology allows to integrate more number of logic fun^n on a single chip.
- A design of large number of PALs or PLA's together on single silicon chip to perform the large size of Boolean logic functions is called as Complex Programmable Logic devices (CPLD).
- The basic structure of CPLD is shown in fig below. It consists of Input / output port, Program blocks & Interconnect array.



FB = (ROM, PLA & PAL)

○ Application of CPLD :-

1. Glue logic.
2. Control circuits.
3. State machines.
4. Prototyping.
5. Embedded systems.

Semiconductor Memories :-

- Semiconductor memories are the digital storage device that stores the digital information to the digital system.
- To store the binary information the memory arrays of the semiconductor memory unit are constructed by the combination of semiconductor devices transistors or diode called memory cells.

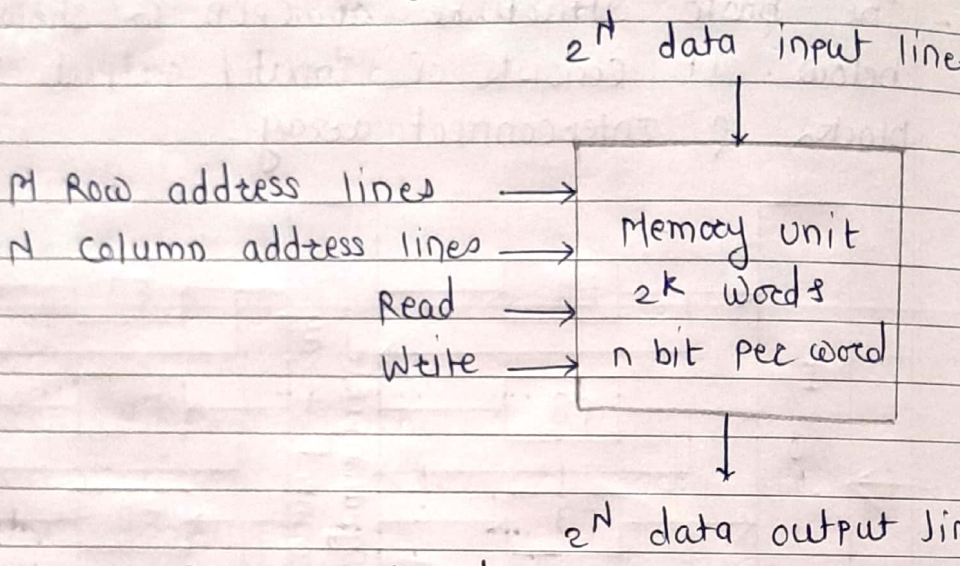


fig:- semiconductor Memory

- The block diagram & internal structure of semiconductor memory device are shown in above fig.
- The circuit consists of M - row address line N - column address lines ; 2^N - data input & output lines, read & write control lines.
- A semiconductor memory device is constructed with two main units such as a decoder and a memory unit.

Memory Organization :-

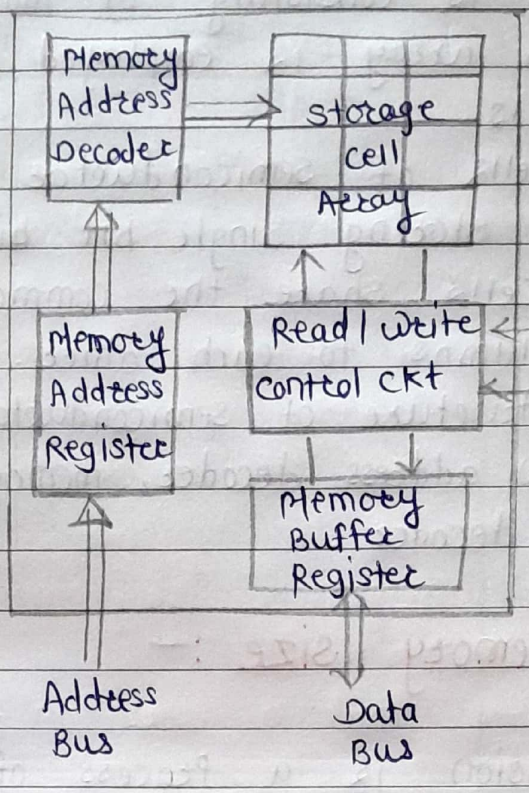


Fig (a) Memory organization.

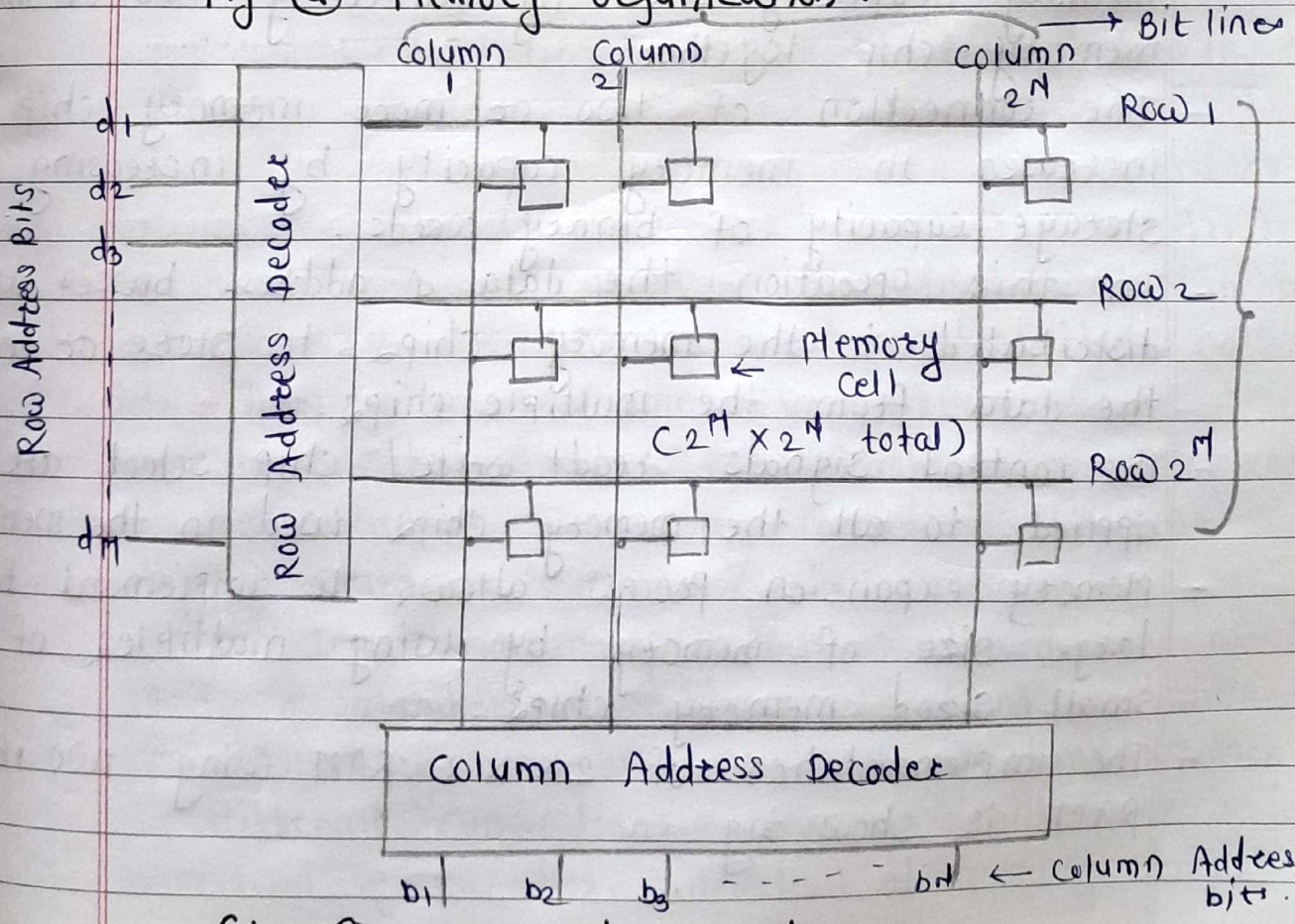


Fig (b) Semiconductor Memory.

- The typical memory organization of semiconductor memory device is consisting of many memory arrays in which each array is employed with more no. of storage cells.
- Each storage cells of semiconductor memory device is capable of storing single bit binary information.
- The memory cells share the common connection of rows & columns to each other.
- The internal structure of semiconductor memory device consists of row address decoder, memory array & column address decoder.

o Expanding memory size :-

- Memory expansion is a process of increasing memory capacity size by connecting two or more memory chip together.
- The connection of two or more memory chip increases the memory capacity by increasing storage capacity of binary words.
- In this operation the data & address busses are distributed to the memory chips to store or read the data from the multiple chips.
- The control signals read/ write/ chip select are applied to all the memory chips used in the memory.
- Memory expansion process allows to implement the large size of memory by using multiples of small sized memory chips.
- The implementation of 2K x 8 RAM using two 1K x 8 RAM is show fig.

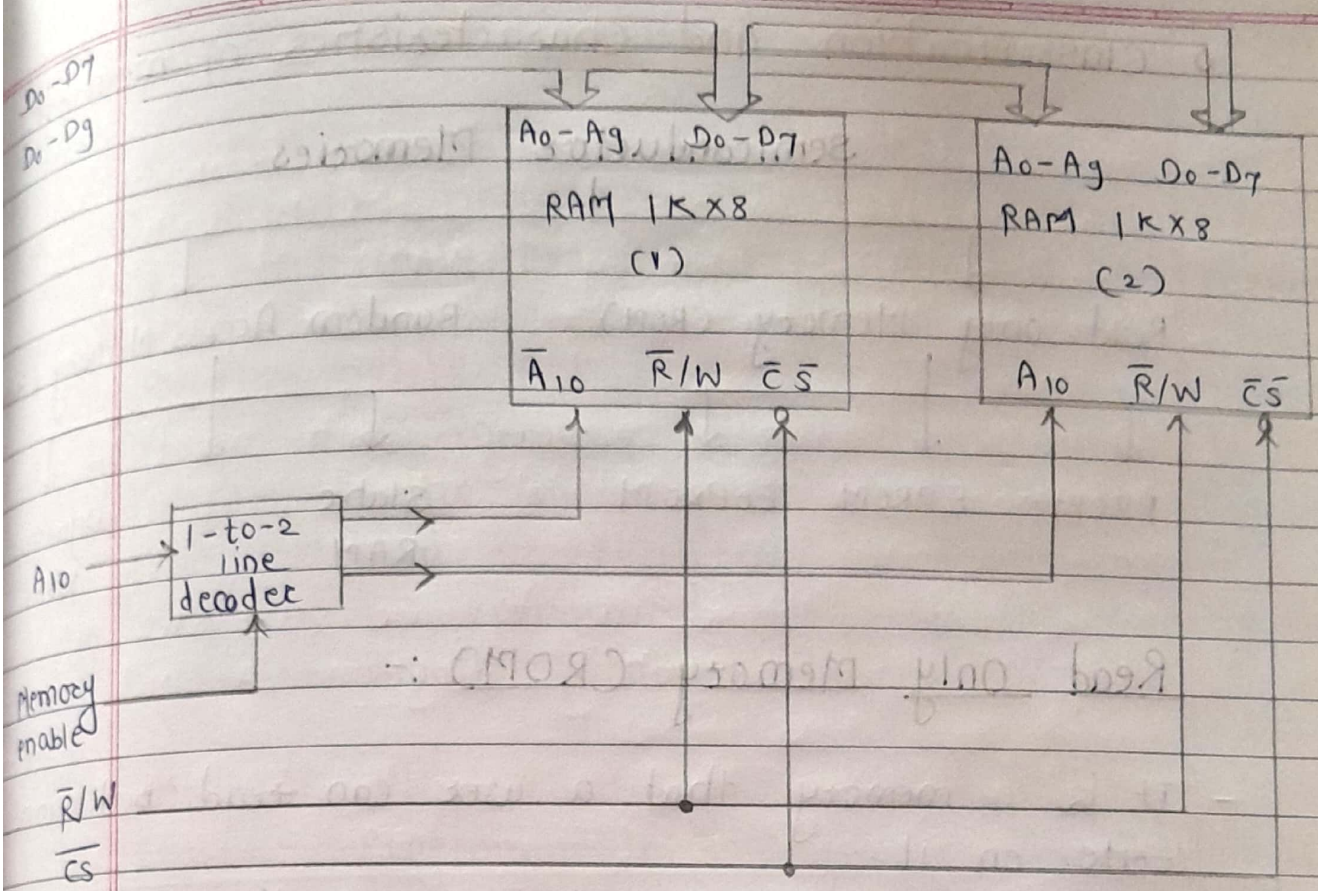
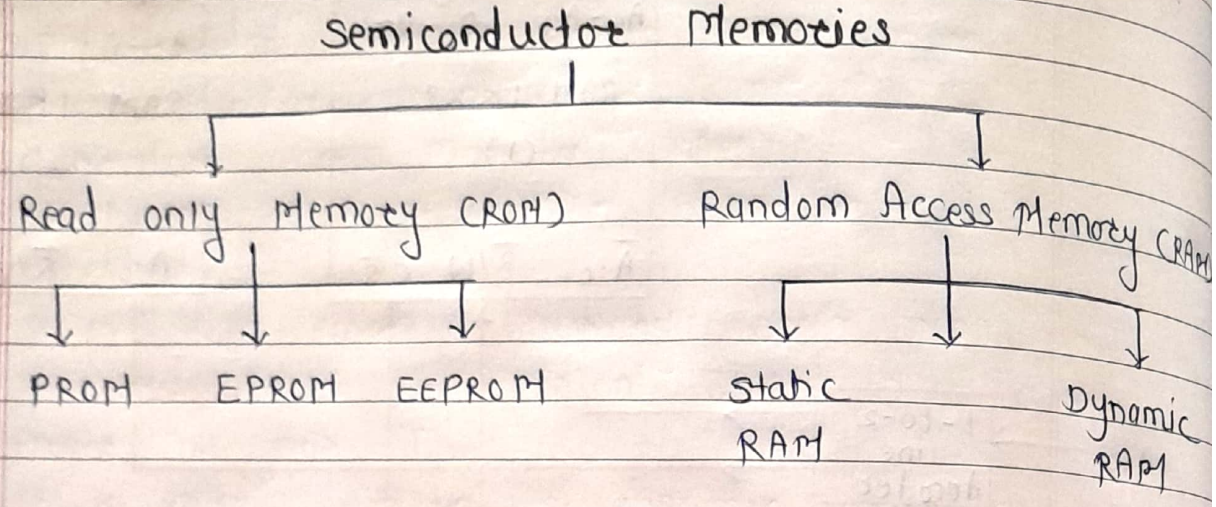


Fig:- Implementation of 2K×8 RAM using two 1K×8 RAMs.

The implementation of 2K×8 RAM requires 11 input address lines ($2^{11} = 2048$) ($A_0 - A_{10}$) & 8 data lines ($D_0 - D_7$). The implementation of 1K×8 RAM requires 10 input address lines ($2^{10} = 1024$) ($A_0 - A_9$) & 8 data lines ($D_0 - D_7$).
The implementation of 2K×8 RAM using two 1K×8 RAM can be obtained easily by cascading the two memory chips.

In the memory expansion the desired word length & number of address locations are obtained by connecting two or more memory chips together. The different connections of memory configⁿ are: series connection & parallel connection of memory chips.

Classification and characteristics of memories:



Read only Memory (ROM) :-

- It is a memory that a user can read but cannot write on it.
- This type of memory is non-volatile (information is stored permanently and cannot be erased).
- The memory in ROM is filled at the time of creation.
- The user can find ROM chips in computers & other electronics products like washing machines, digital cameras, microwave ovens, cars etc.
- Read only memory is a memory device that can store the set of binary information permently for digital system processing.
- In embedded systems to store the configuration parameters, calibration setting etc.
- BIOS - Basic input output system.
- It used in printers, keyboards & communication with the main system.

Basically ROM is classified into different types :-

1. Programmable ROM (PROM)
2. Erasable programmable ROM (EPROM)
3. Electrically Erasable programmable ROM (EEPROM)

1 Programmable ROM (PROM) :-

- PROM is a memory device which allows to programme the memory device as per the programme table designed or required by the customers.
- PROM is an economical way of programming ROM even for small quantities.

2. Erasable Programmable ROM (EPROM) :-

- Erasable programmable ROM is a reprogrammable memory device which allows to delete & rewrite the binary information stored.
- The binary information is stored as a charge on an isolated gate capacitor (floating gate) of Mos transistor's used in the memory cells.
- It allows to erase all the information present in the ROM & do not allow for erasing selective information.

3. Electrically Erasable Programmable ROM (EEPROM) :-

- EPROM is also a reprogrammable memory device which erase all the binary information stored in the memory unit by exposing the ultraviolet light.
- EEPROM consisting of a Mos transistor memory cells.
- It allows to erase the binary infor^n selectively register level rather than erasing all information by using electrical signals.

Random Access Memory (RAM) :-

- The read-write memory array circuit of a semiconductor memory is commonly called as "Random Access Memory" (RAM).
- RAM can be accessed for data read / data write operation.
- RAM is a volatile memory which means that the memory loses the content stored in it when the power is turn off.
- It is used to store the dynamic variables including stack & other supporting programmes that may loaded from disk drives.

Comparison betⁿ SRAM and DRAM :-

	Static RAM	Dynamic RAM
1	SRAM memory is capable of retaining the binary inforⁿ long as power is applied.	DRAM memory that requires a periodic refreshing to avoid the data loss even the power is applied continuly
2	Memory cell size is large	Memory cell size is small
3	Memory Access time is less	Memory Access time is high.
4	SRAM cell circuit is complex	DRAM cell circuit is simple.
5	SRAMs are expensive.	DRAMs are cost effective.
6	It do not require refreshing.	To avoide the data loss periodic refreshing is required
7	It does not require extra harware circuit.	Extra hardware circuit is required.
8	Power consumption is high.	Power consumption is less.

o Difference betⁿ RAM and ROM :

	RAM	ROM
1	RAM stands for Random Access memory	ROM stands for Read only memory.
2	It is a temporary memory	It is a permanent memory.
3	Data stored in RAM can be retrived & altered.	Data stored in ROM can only read.
4	RAM is a volatile memory.	ROM is a non-volatile memory.
5	It is a high speed memory.	It is much slower than RAM.
6	It is expensive.	It is comparatively cheap.
7	store data in MBs.	store data in GBs.
8	The instructions are written into the RAM at the time of execution.	The instruction are written into the ROM at manufacturing time.
9	Types of RAM (i) SRAM (ii) DRAM	Types of ROM (i) PROM (ii) EPROM (iii) EEPROM

Introduction to VHDL :-

- VHDL stands for very high speed integrated circuit (VHSIC) Hardware description language.
- VHDL is used to design digital systems.
- It defines syntax & simulation sematics to design the program
- It is strong typed language.

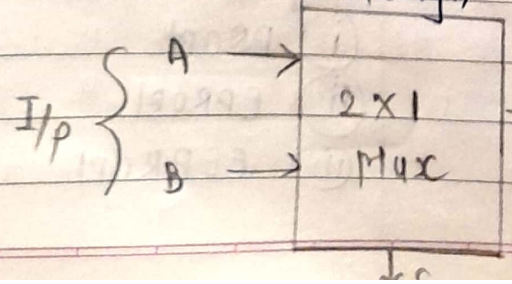
Modeling styles in VHDL

There are three types of modeling styles :-

1. Behavioral
2. Data flow
3. structural

1 Behavioral description :-

- In this modeling type the behavioral of an entity (Input and output relationship) is expressed using sequential statements.
- Details of gate level implementations are not required for this type of modeling.
- In behavioral design method asithmatic of inter-connected symbols are not used.
- This style of modeling needs input specifications or truth table as design specifications.



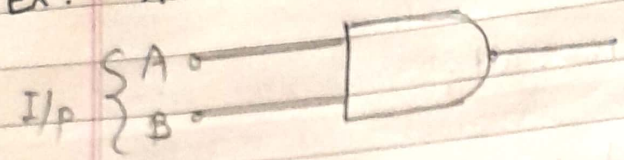
2:1 Multiplexer :-

```
library IEEE;
use IEEE.std_logic_1164.all;
Entity mux 2:1 is
port (A, B, sel : in std_logic; Y : out std_logic);
end mx 2:1;

architecture behavioral of mux 2 to 1 is
begin
  process (A, B, sel)
begin
  if sel = '0' then
     Y <= A;
  else
     Y <= B;
  end if;
  end process;
  end behavioral;
```

2 Data flow Description :-

- Data flow modeling is based on concurrent assignment statement.
- Describes how data flows bet^n components.
- The flow of data through entity is expressed using concurrent signal assignment statement when any input changes its value, each statement gets executed.
- The data flow modeling style needs Boolean eq^n as design specification.
- Easier for combinational circuits like adders & multiplexer's etc.

Ex:- AND Gate

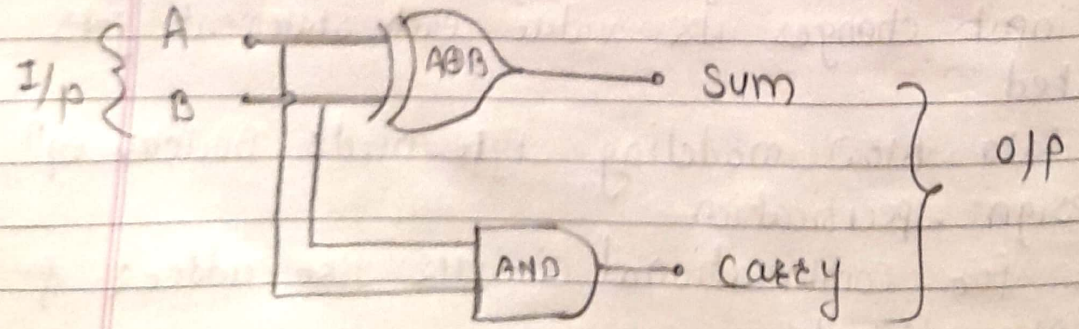



```
library IEEE;
use IEEE.std_logic_1164.all;

Entity AND_gate is
port ( A, B : in std_logic;
       Y : out std_logic);
end AND_gate;

architecture dataflow of AND_gate is
begin
   Y <= A AND B;
  end dataflow;
```

3. Structural Description :-

- Structural modeling is a hierachical design method. Here sub component perform small operations of the complete model.
- Each component can simulated individually.
- describes the interconnection of components.

Ex:- Half Adder :-



```
library IEEE;
use IEEE.std_logic_1164.all;

Entity HalfAdder1 is
port (A, B : in std_logic;
      sum, carry : out std_logic);
end HalfAdder1;

architecture structural of HalfAdder1 is
 component xor_gate
 port (A, B : in std_logic;
       sum : out std_logic);
 end component;

 component AND_gate
 port (A, B : in std_logic;
       carry : out std_logic);
 end component;

begin
 U1 : xor_gate port map (A => A, B => B, Y => sum);

 U2 : AND_gate port map (A => A, B => B, Y => carry);

end structural;
```

o Lexical Elements :-

- Identifiers: names for signals, variables, constants (eg - clk, data-in)
- keywords: reserved words (eg - begin, end, if, then)
- comments: start with
- literals: numbers or characters (eg: '0', '1', "1010")
- Operators: logical and arithmatic (eg. and, or, +, -

o Data Objects in VHDL :-

- Constants: Fixed values, can not be changed during simulation.
- Variables :- can be changed inside a process.
- Signals :- Used to connect components & transfer valu
- Files :- Store and read external data.

o Data Types in VHDL :-

- Bit : '0', '1'
- Bit - Vector : eg :- "1010".
- Std_logic / Std_logic_vector : Supports multiple logic states.
- Integer, Boolean, character, string.

Attributes in VHDL :—

- Attributes provide information about signal & object

Ex :-

- 'event → true if signal changed
- 'last_value → previous value of signal
- 'range → range of array or vector.

Operator's in VHDL :-

- Logical operators :- and, or, nand, nor, xor, xnor & not.
- Relational operators :- =, /=, <, <=, >, >=
- Arithmatic operators :- +, -, *, /, mod, rem
- Concatation :- &

simulation in VHDL :-

- simulation is used to verify the functionality of the VHDL design before hardware implementation
- Steps :- 1. Write VHDL code.
  2. Write a testbench.
  3. Compile & simulate the design.
  4. Check waveforms & outputs.

Example Testbench for AND Gate :-

```
library IEEE;
use IEEE.std_logic_1164.all;

entity tb_and_gate is
end tb_and_gate;
```

```
architecture test of tb_and_gate is
signal a, b, y : std_logic;
begin
  UUT : entity work.and_gate port map (a => a,
                                       b => b, y => y);

process
begin
  a <= '0'; b <= '0'; wait for 10 ns;
  a <= '0'; b <= '1'; wait for 10 ns;
  a <= '1'; b <= '0'; wait for 10 ns;
  a <= '1'; b <= '1'; wait for 10 ns;

wait;
end process;
end test;
```

o VHDL Coding Examples of Combinational circuits

Full Adder :-

```
library IEEE
use IEEE.std_logic_1164.all;

Entity Full_Adder is
 port ( A : in std_logic;
        B : in std_logic;
        cin : in std_logic;
        Sum : out std_logic;
        Cout : out std_logic);
end Full_Adder;
```

```
architecture structural of full-Adder is
signal S1, C1, C2 : std-logic;
component Half_Adder
port ( X : in std-logic; Y : in std-logic;
       Sum : out std-logic; Carry : out std-logic);
end component;
begin
HA1 : Half_Adder port map ( X => A,
                            Y => B,
                          sum => S1
                        carry => C1);
HA2 : Half_Adder port map ( X => S1,
                            Y => Cin,
                          sum => sum
                        carry => C2);

    Cout <= C1 or C2;
end structural;
```

Q.8 Implement given Boolean functions using PLA, PAL & PROM :-

$F_1(A, B, C) = \Sigma m(0, 2, 6, 7)$

$F_2(A, B, C) = \Sigma m(0, 3, 4, 5, 7)$

⟹ Implementation of Boolean fun^n using a PROM :-

Given :-

$F_1(A, B, C) = \Sigma m(0, 2, 6, 7)$

$f_2(A, B, C) = \Sigma m(0, 3, 4, 5, 7)$

The Boolean expressions are consisting of 3 inputs ($N = 3$) & 2 output fun^n ($M = 2$)

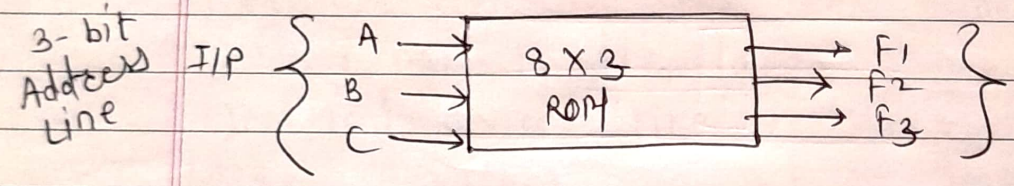


fig :- Block diagram.

- In the ROM structure, the decoder codes the 3-bit binary input address lines into 8 distinct word line.
- All the word lines are connected to each of programmable OR gates through the electronic fuse to provide the o/p fun^n.

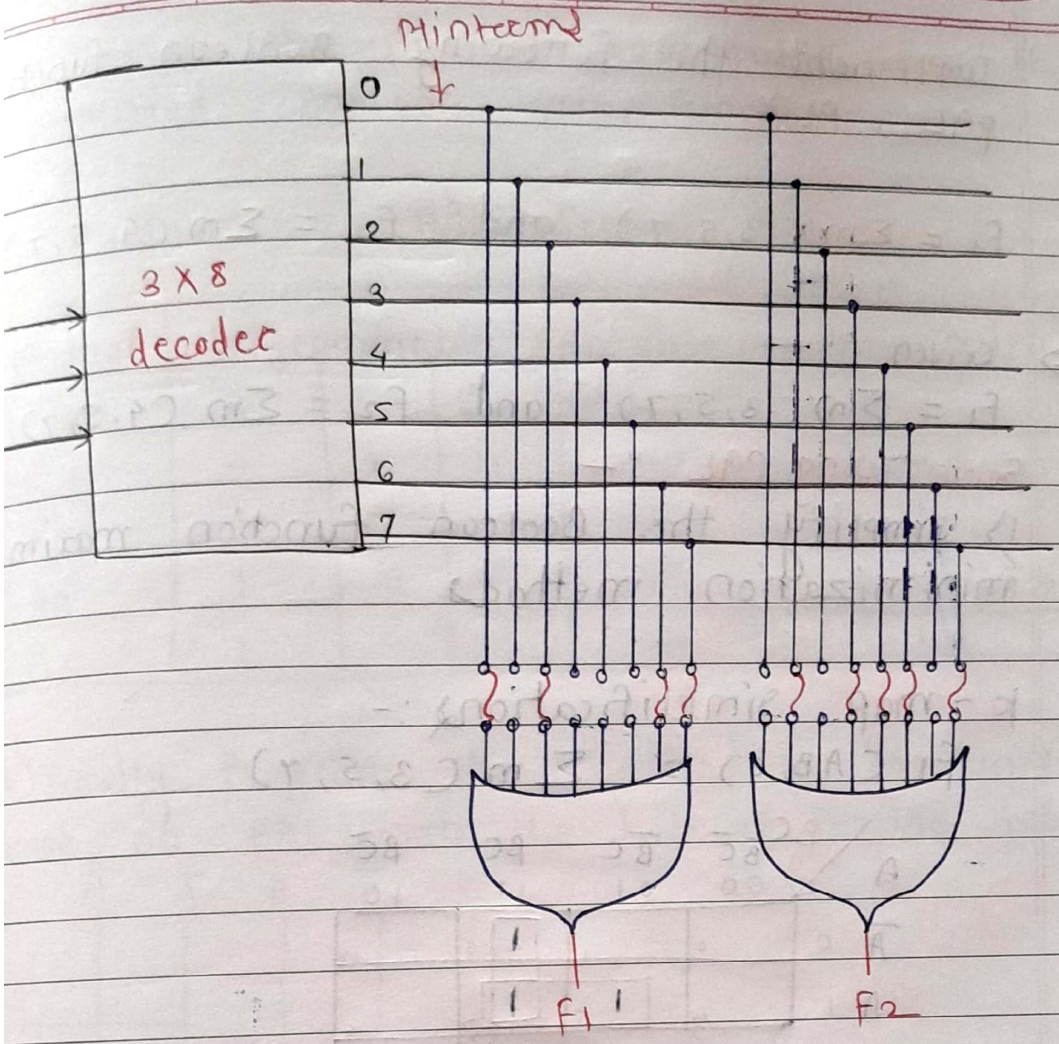


fig:- Implementation of Combinational logic using ~~3X~~ 8X3 ROM.

Implement the following Boolean function using PAL, PLA

$f_1 = \Sigma m(3,5,7)$ and $f_2 = \Sigma m(4,5,7)$

Given :—

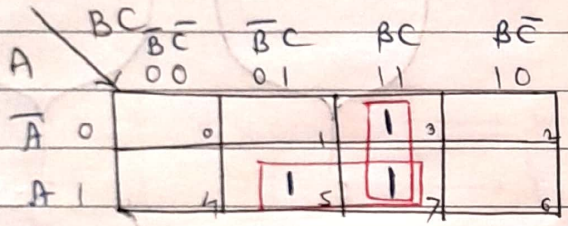
$f_1 = \Sigma m(3,5,7)$ and $f_2 = \Sigma m(4,5,7)$

Solve using PAL :—

1) simplify the Boolean function maximally using minimization methods.

k-map simplifications :-

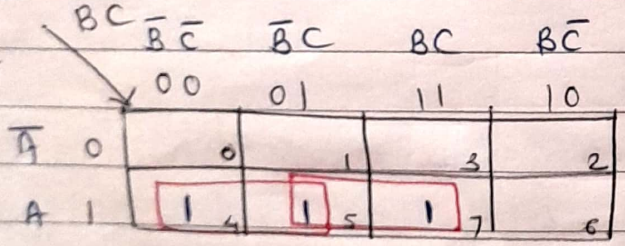
$F_1(A,B,C) = \Sigma m(3,5,7)$



$F_1 = BC + AC$

k-map simplifications :-

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



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

2) Prepare the PAL program table for the simplified Boolean logic function.

Table :- PAL Program Table

Product terms	Programmable AND gate i/p				outputs
	A	B	C	F_1	
BC	–	1	1	–	$F_1 = \bar{A}BC + AC$
AC	1	–	1	–	
$A\bar{B}$	1	0	–	–	$F_2 = A\bar{B} + AC$
AC	1	–	1	–	

3) Finally, program the simplified Boolean functions on PAL with the help of program table.

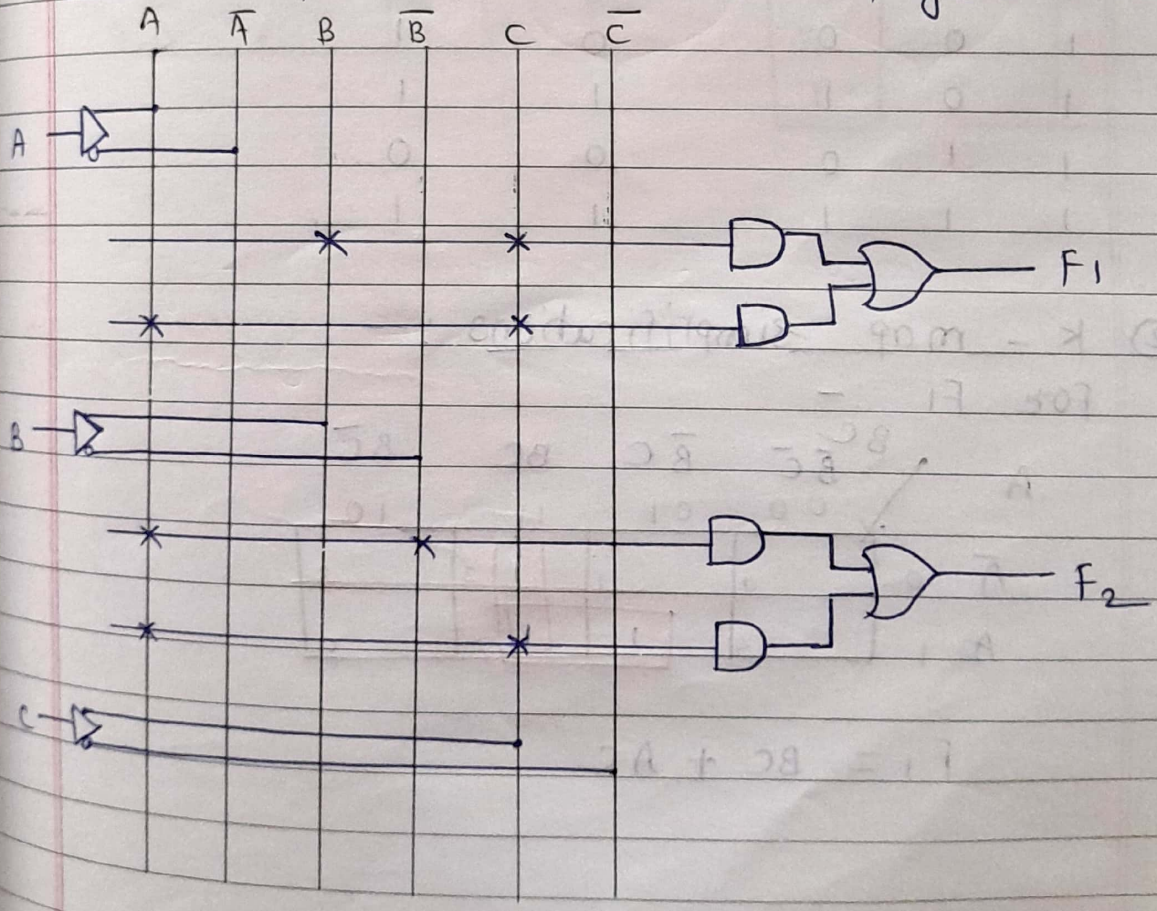


fig :- Implementation of PAL.

Solve using PLA :-

Given :- $F_1(A, B, C) = \Sigma m(3, 5, 7)$

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

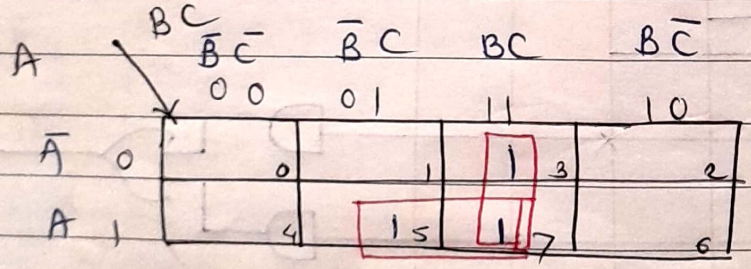
① The truth table of the given Boolean function is shown in table.

Truth table :-

Inputs			Outputs	
A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

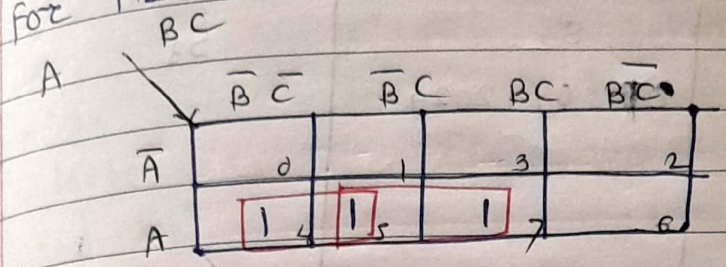
② K-map simplifications :-

for F_1 :-



$F_1 = BC + AC$

for f_2 :-



$$f_2 = A\bar{B} + AC$$

programmable table of PLA :-

product terms	Inputs			outputs		
	A	B	C	f_1	f_2	
BC	–	1	1	1	–	
AC	1	–	1	1	1	
$A\bar{B}$	1	0	–	–	1	
				T	T	T/C

Q. Implement the following Boolean function using suitable PLA.
$F(A, B, C, D) = \Sigma m(3,4,5,7,10,14,15)$ give me proper solution. ~~give~~

→ Given :-

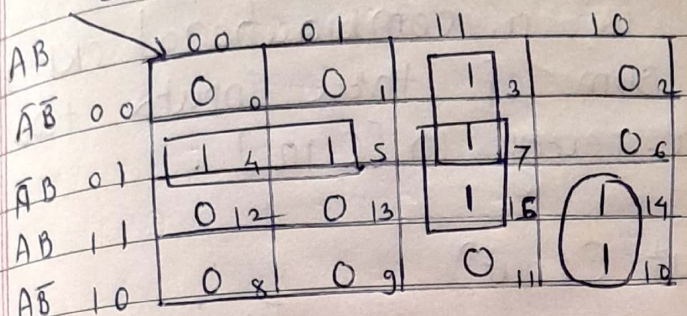
$F(A, B, C, D) = \Sigma m(3,4,5,7,10,14,15)$

The logic truth table of given 4-input function are shown in table :-

Truth table of 4-input logic functions :-

Inputs				Outputs
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

K - map for F



$F = \bar{A}B\bar{C} + \bar{A}CD + BCD + AC\bar{D}$

Programmable table of PLA :-

Product terms	Input A B C D	outputs F
$\bar{A}B\bar{C}$	0 1 0 -	1
$\bar{A}CD$	0 - 1 1	1
BCD	- 1 1 1	1
$AC\bar{D}$	1 - 1 0	1

T	T/C

Assignment No. 5

Q.1 Describe the architecture and operation of a programmable Logic Array (PLA). How does it differ from a PAL?

Q.2 Distinguish between SRAM and DRAM.

Q.3 Write a VHDL code for a full adder using structural description.

Q.4 Explain General Architecture of FPGA in detail.

Q.5 Implement the following Boolean function using PAL, PLA.

$F_1 = \sum m(3,5,7)$ and $F_2 = \sum m(4,5,7)$

Q.6 Write VHDL code for Mux 4:1 using dataflow & behavioural architecture style.

Q.7 Classify memories RAM and ROM.

Q.8 Implement given Boolean functions using PLA, PAL and PROM.

$F1(A,B,C) = \sum m(0,2,6,7)$

$F2(A,B,C) = \sum m(1,3,4,5,7)$

Q.9 Explain General Architecture of CPLD in detail.

Q.10 List the various advantages and feature of VHDL.