UNIT – VII

SEMI CONDUCTOR MEMORIES

INTRODUCTION

Earlier memories used were mostly of magnetic type. With unprecedented developments in semiconductor technology, it has become possible to make semiconductor memories of various types and sizes. These memories have become more popular due to their small size, low cost, high speed, high reliability and ease of expansion of memory size. The fig (7.1) shows the block diagram of a memory device.

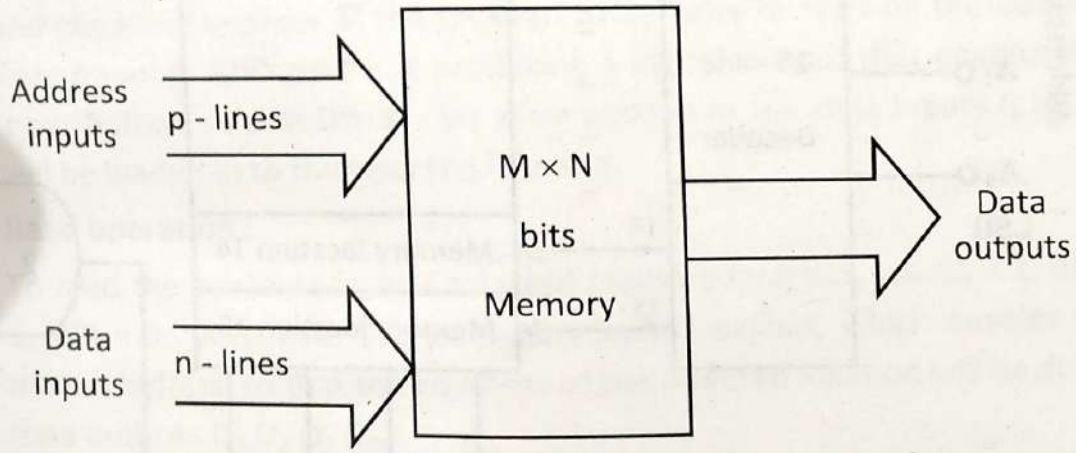


Fig. (7.1) : Block diagram of memory device.

There are number of locations in the memory chip, each location being meant for one word of digital information. The size of the memory chip is specified by two numbers M and N as MXN bits. The number M specifies the number of locations available in the memory, and N is the number of bits in each location. The commonly used values of the number of words per chip are 64, 256, 512, 1024, 2048, 4096 etc, where as common values for the word size are 1, 4, 8 etc.

Each of the M locations in the above memory is defined by a unique address and therefore, for accessing any one of the M locations, P inputs are required, where $2^P = M$. These set of lines are referred to as address inputs or address bus.

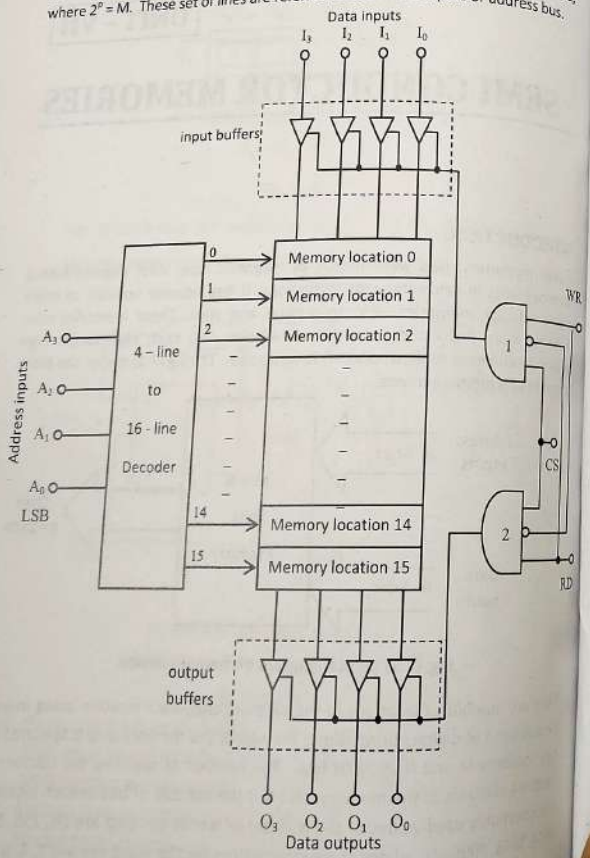


Fig. (7.2) : Internal organization of 16 × 4 memory.

The number of inputs required to store the data into or read the data from each memory location is N. One set of N lines is required for storing data into the memory, referred to as data inputs; and another set of N lines is required for reading data already stored in the memory, which is referred to as data outputs. In some memories same set of lines is used for data inputs as well as data outputs, and is referred to as data bus. This means that data bus is time multiplexed. It is used as input for some specific time, and as output for some time. This offers saving of pins on the IC chip.

A number of control inputs are required to give command to the device to perform the desired operation, for example, a command signal is required to tell the memory whether to write or read operation is desired. Other commanded inputs include chip enable (CE), chip select (CS) etc.

7.1 INTERNAL ORGANIZATION OF A 16 × 4 MEMORY CHIP

The internal organization of a 16 ×4 memory chip is shown in fig (7.2). The write and read operations are carried out as given below.

Write operation:

To write a word into the selected memory location, put CS=1, WR=1, RD = 0, and place the address of the location to be write or read on the address lines (inputs). AND gate 1 is producing a logic 1 output that enables the input buffers so that the 4 – bit word applied to the data inputs I_3 I_2 I_1 I_0 will be loaded in to the selected location.

Read operation :

To read the contents from a selected memory location, put CS = 1, RD = 1, WR = 0. AND gate 2 is producing logic 1 output, which enables the output buffers, so that the contents of the selected location will be at the data outputs O_3 O_2 O_1 O_0.

7.2 CLASSIFICATION OF MEMORIES

Memories are classified according to their principle of operation. The most commonly used memories are.

1. Sequential access memory (SAM)
2. Random access memory (RAM)
3. Read only memory (ROM)
4. Content addressable memory (CAM)

In sequential accessed memories, the memory locations are accessed, for reading from or writing into in a sequential fashion. Therefore, the time required for accessing a memory location for writing into or reading from is different for different locations. There are two types of sequential accessed memories

1. Shift registers
2. Charge coupled devices

Random access memory is one in which the access time is same for each memory location. This is also called as read – and – write memory referred to as RAM. They can be static or dynamic.

Read only memories, as name suggests, is meant only for reading the information from it. The process of entering the information into the memory is done outside the system where it is used. Therefore, it is called as read – only – memory. It is used to store information which is fixed, such as tables for various functions, fixed data and instructions. The ROM is a random- access memory.

Memories are classified according to their physical characteristics as

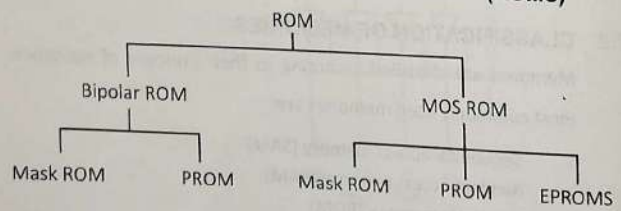
1. Erasable and non – erasable memories
2. Volatile and non – volatile memories

A memory in which the information stored can be erased and new information is stored, is referred to erasable memory. In a non – erasable memory information stored can't be erased, for example ROM.

In a volatile memory, the information stored is lost when the electrical power is switched off.

In a non volatile memory the information remain unchanged even if the power is switched off.

7.3 CLASSIFICATION OF READ ONLY MEMORIES (ROMS)



Mask ROM is a type of ROM in which data are permanently stored in the memory during the manufacturing process.

PROM (programmable ROM) is the type in which the data are electrically stored by the user with the aid of specialized equipment.

Mask ROM and PROM can be either bipolar or MOS Technology. The EPROM (Erasable programmable ROM) is strictly MOS technology.

EPROM is electrically programmable by the user but the stored data can be erased either by exposure to ultraviolet light or by electrical means. The later type is called as electrically erasable PROM (EEPROM) (or) electrically alterable PROM (EAPROM)

Mask ROM

The basic cell in a MOS ROM is a bipolar transistor as shown in fig (7.3)

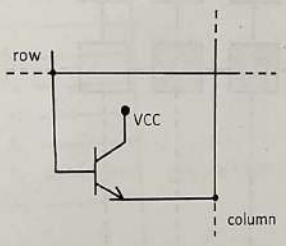


Fig. (7.3) : Basic Mask ROM Cell

Storing a 1

The presence of a connection from a row line to the base of transistor represents a 1 at that location, because when the row line takes a HIGH, all transistors with a base connection to that row line turn on and connected the HIGH (1) to the associated column lines.

Storing a 0

At row / column junctions where there are no base connections, the column lines remains low (0).

16 × 8 bit mask ROM

The configuration of a 16×8 mask ROM is shown in fig.(7.4). The shaded rectangle corresponds to a transistor with a base connection and unshaded rectangle corresponds to a transistor without base connection.

Referring to fig (7.4), when a binary address code is applied to the address input, the corresponding row line goes HIGH.

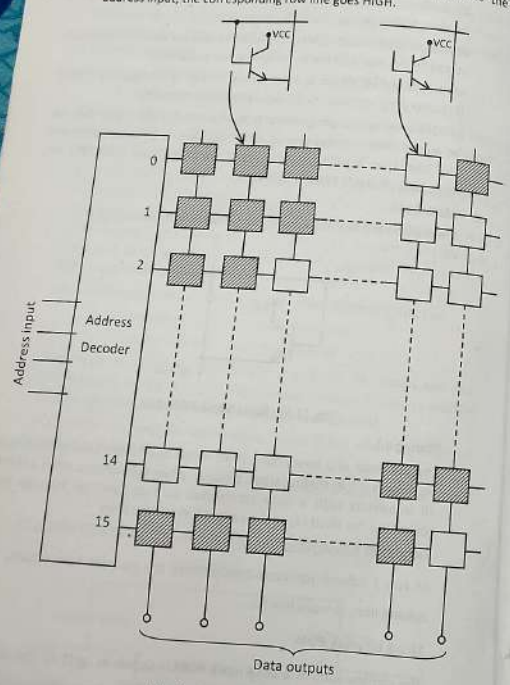


Fig. (7.4) : 16 × 8 Mask ROM

This HIGH is connected to the column lines through the transistors at each junction where 1 is stored. At each cell where 0 is stored, the column line stays low because of the terminating transistor. The column lines forms the data outputs. The eight data bits stored in the selected row appear on the output lines.

Total capacity of the ROM is 16 × 8 = 128 bits.

The fig (7.5) shows the logic symbol for 256 × 4 ROM

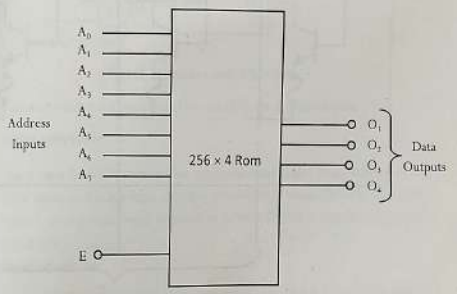


Fig. (7.5) : Logic symbol for 256 × 4 ROM

PROMS:

They use some type of fusing process to store bits, where by a memory link is fused open or left intact to represent 0 or 1. The fig (7.6) shows a typical PROM using bipolar transistors.

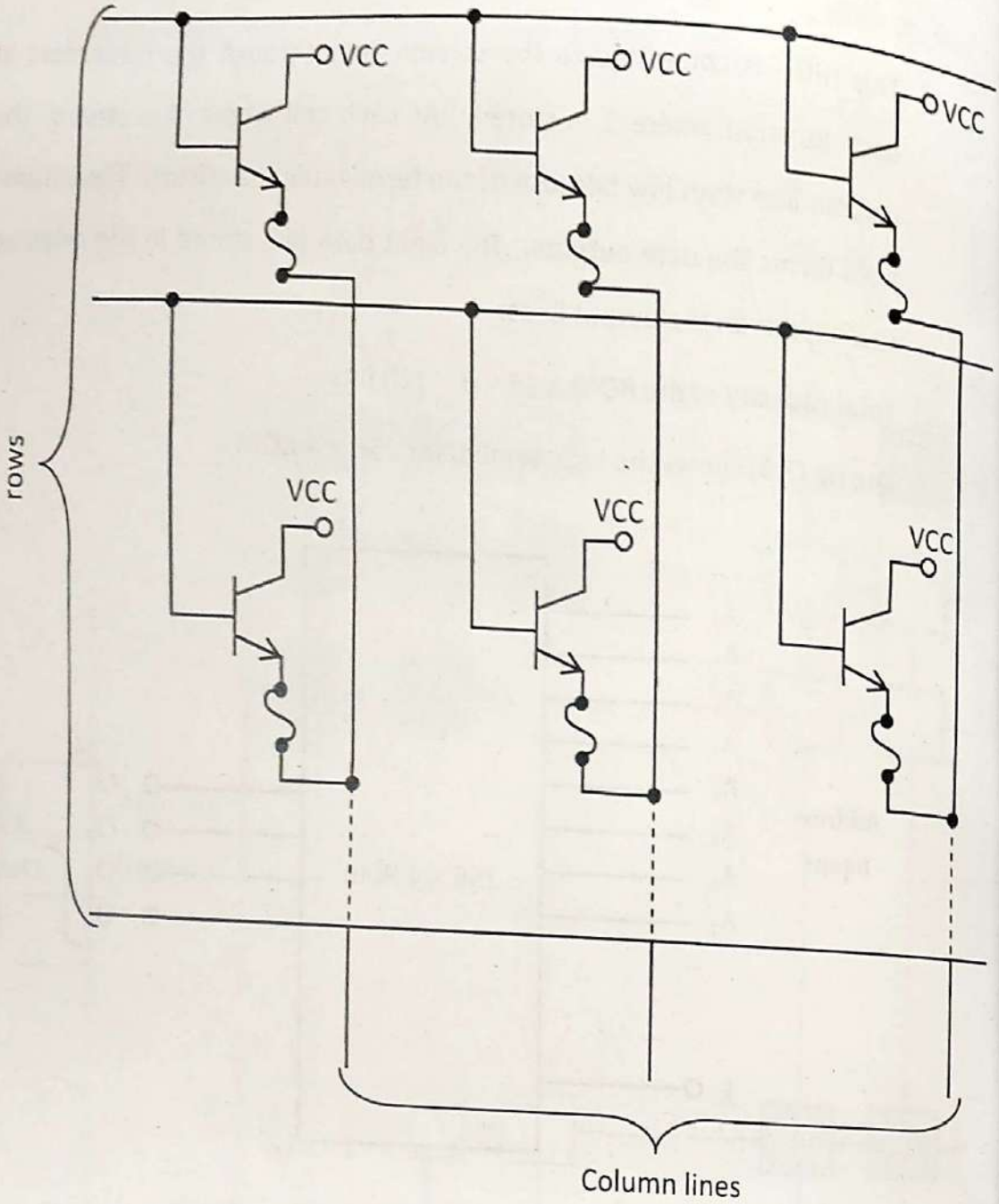


Fig. (7.6) : Typical PROM

The fuse links are manufactured into PROM between the emitter of each cell's transistor and its column line. In the programming process, a sufficient current is injected through a fuse link to burn it open to create a stored O. The link is left infact for a stored 1.

Fig.(7.8) : RAM Example

...c bipolar cell consists of two transistors which are cross coupled to ...onversional flip – flop. Each transistor has two emitters, however, one of these is used to select the cell in an array, the other emitter is used to sense the state of the flip – flop, and to write in to the flip – flop. The connections to the top emitters in Q_1 and Q_2 from what is called a digit line. It is used to both sense the state of the flip – flop and write into it.

The read and write operations are performed as follows.

Read operation:

Make the select line high to select the cell from which we want to read.

If suppose Q_1 is ON and Q_2 is OFF current flows through on transistor to ground. No current flows through OFF transistor.

Measure voltage at points P_1 and P_2. No voltage at point P_2 indicate a stored 1 ($\therefore Q_2$ = off) if there is a voltage at point P_2, it indicate a stored O ($\therefore Q_2$=on).

Write operation:

To write a 0, make the select line high. Force P_1 high w.r.t P_2

(i.e, $P_1 = +5v$, P_2 = GND).

Then Q_2 = ON ($\therefore$ its base is connected to VCC through ON and emitter is at O V) and Q_1 = OFF. As Q_2 = ON, stored bit is a O.

Two write a 1, make the select lines high.

Force P_2 HIGH w.r.t. P_1 (i.e, $P_2 = +5V$, P_1 = GND)

Then Q_2 = OFF and Q_1 is ON. As Q_2 is OFF stored bit is a 1.

7.6 DYNAMIC RAM MEMORIES

Dynamic Ram cell store data in a small capacitor. The basic dynamic RAM cell is shown in fig (7.11).

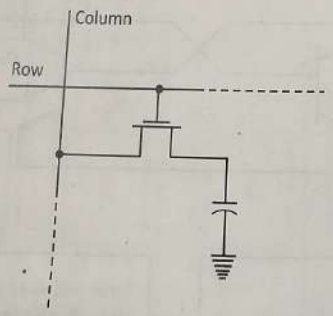


Fig. (7.11) : Basic dynamic RAM cell

Fig. (7.12) shows the complete arrangement to write and read from a dynamic RAM cell.

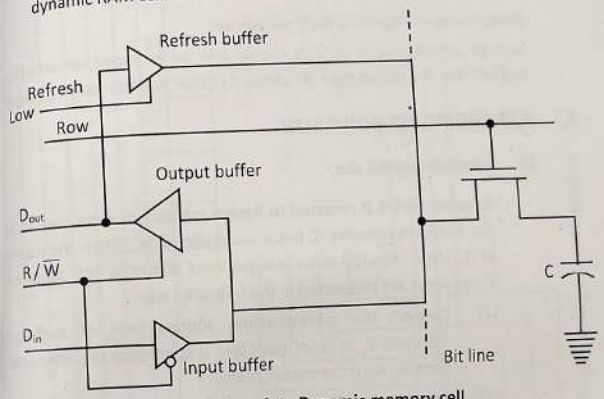


Fig. (7.12) : Complete Dynamic memory cell.

To write a 1 into the cell,

Make R/ $\overline{w}$ = 0, this enables input buffer and disables output buffer.

Put D_{in} = 1

Put row line = 1 (this turns ON the FET)

This FET which is ON acts as a closed switch connecting capacitor to the bit time. As D_{in} = 1, bit line = 1, and the capacitor charges to that positive voltage

To write a O in to the cell,

Make R/ $\overline{w}$ = 0

Put Row line = 1.

As D_{in} = 0, the capacitor remains uncharged, or if it is already storing a 1, it discharges through D_{in} line.

To read from the cell,

Make R/$\overline{w}$ = 1, which enables the output buffer and disables the input buffer.

Put row line = 1, now the FET is on and connecting the capacitor to the bit line and thus to the output buffer. So the data bit appear on the data – output line D_{out}.

Advantages of dynamic RAM memories:

Very simple, thus allowing very large memory arrays to be constructed on a chip at a lower cost per bit.

Disadvantage of dynamic RAM memories:

Storage capacitor can't hold its charge over an extended period of time and will lose the stored data bit unless it charge is refreshed periodically.

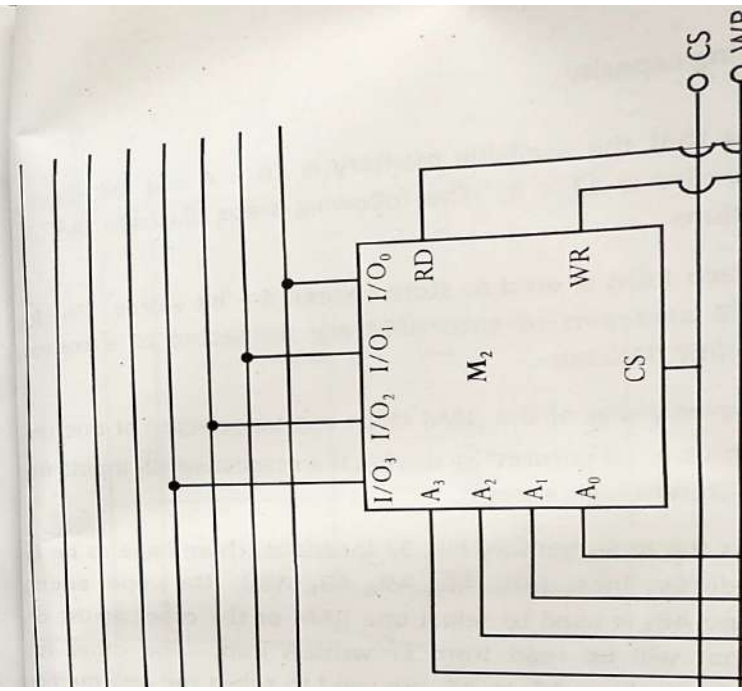
7.7 EXPANDING MEMORY SIZE

(i) **Expanding word size.**

Suppose that it is required to have a memory of word size 'n' and the available memory IC has a word size of 'N'. Then the number of IC chips required is an integer, next higher to the value n/N. These chips are connected in the following way:

(1) Connect the corresponding address lines of each chip individually, ie, A_0 of each chip is connected together and it become A_0 of the overall memory

(2) Connect the RD input of each IC together and it become the



(ii) Expanding capacity:

Suppose that the available memory is 16 × 4 and the required memory size is 32 × 4. The following steps illustrate the inter connections

(1) Each RAM is used to store sixteen 4 – bit words. The four I/0 data pins of each RAM are connected to a common 4-line data bus.

(2) As only one of the RAM chips can be selected at one time, this can be ensured by driving the respective $\overline{CS}$ inputs from different logic signals.

As the new memory has 32 locations, there have to be five address lines. (AB_4, AB_3, AB_2, AB_1, AB_0). The upper address line AB_4 is used to select one RAM or the other as the one that will be read from or written into. The other four address lines AB_0 to AB_3 are used to select the one memory location out of 16 locations from the selected memory chip. The fig (7.14) shows the inter connections to make 32 × 4 memory using two 16 × 4 memory chips.

When $AB_4 = 0$, RAM 1 is enabled ($\therefore \overline{CS} = 0$) for read / write. Then, any address location in RAM1 can be accessed by $AB_3 - AB_0$ (0000 – 1111).

Thus, the range of addresses representing locations in RAMI is

00000 – 01111.

Note that when $AB_4 = 0$, the $\overline{CS}$ of RAM2 is high, so that its I/0 lines are disabled and can't communicate with data bus.

When $AB_4 = 1$, RAM2 is enabled ($\therefore \overline{CS} = 0$) and the lines AB_3 to AB_0 select one of its locations. Thus, the range of addresses located in RAM2 are

10000 – 11111.

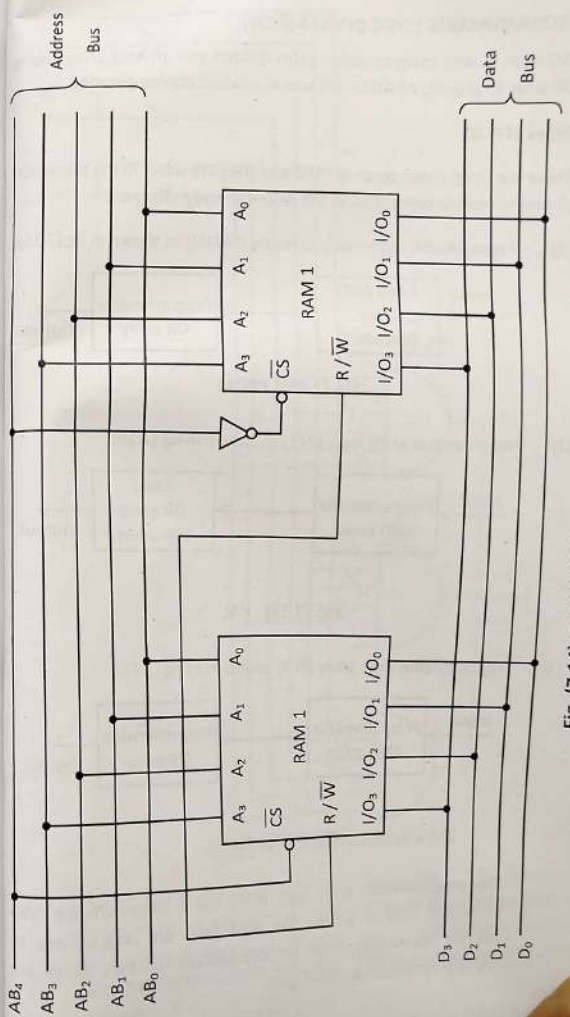


Fig. (7.14) : 16 × 8 Memory using 16 × 4 memory

7.8 PROGRAMMABLE LOGIC DEVICES (PLDS)

PLD is an IC with programmable gates divided into an AND array and an OR array to provide an AND – OR sum of product implementation.

Types of PLDS

There are three major types of PLDS and they are differ in the placement of programmable connection in the AND–OR array. They are

(1) Programmable read – only memory (PROM) as shown in fig (7.15)

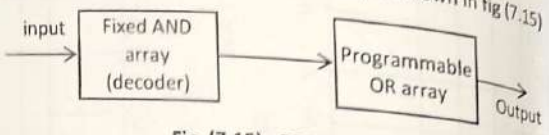


Fig. (7.15) : PROM

(2) Programmable array logic (PAL) as shown in fig (7.16)

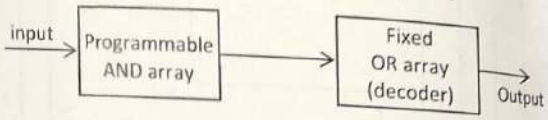


Fig. (7.16) : PAL

(3) Programmable logic array (PLA) as shown in fig (7.17)

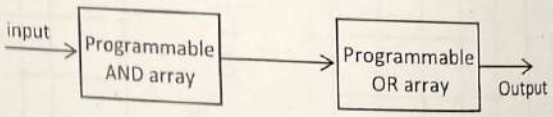


Fig. (7.17) : PLA

The programmable array logic (PAL) has a programmable AND array and fixed or array. The AND gates are programmed to provide the product terms for the Boolean function, which are logically summed in each OR gate.

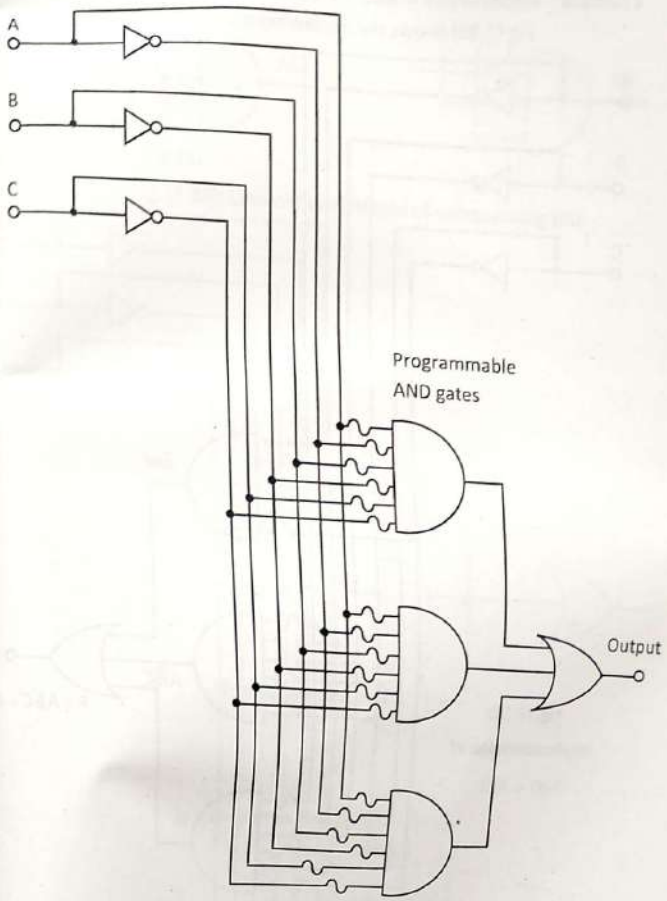


Fig. (7.18): Basic structure of a PAL

The fig (7.18) shows an example of basic PAL. The fuse symbols represent fusible links that can be burned open using equipment. Every input variable and its complement can be left either connected or disconnected from every AND gate. We can say that the AND gates are programmed.

Example : Implement $F = \bar{A}BC + A\bar{B}C$ using PAL..

Fig (7.19) shows the implementation of $\bar{A}BC + A\bar{B}C$.

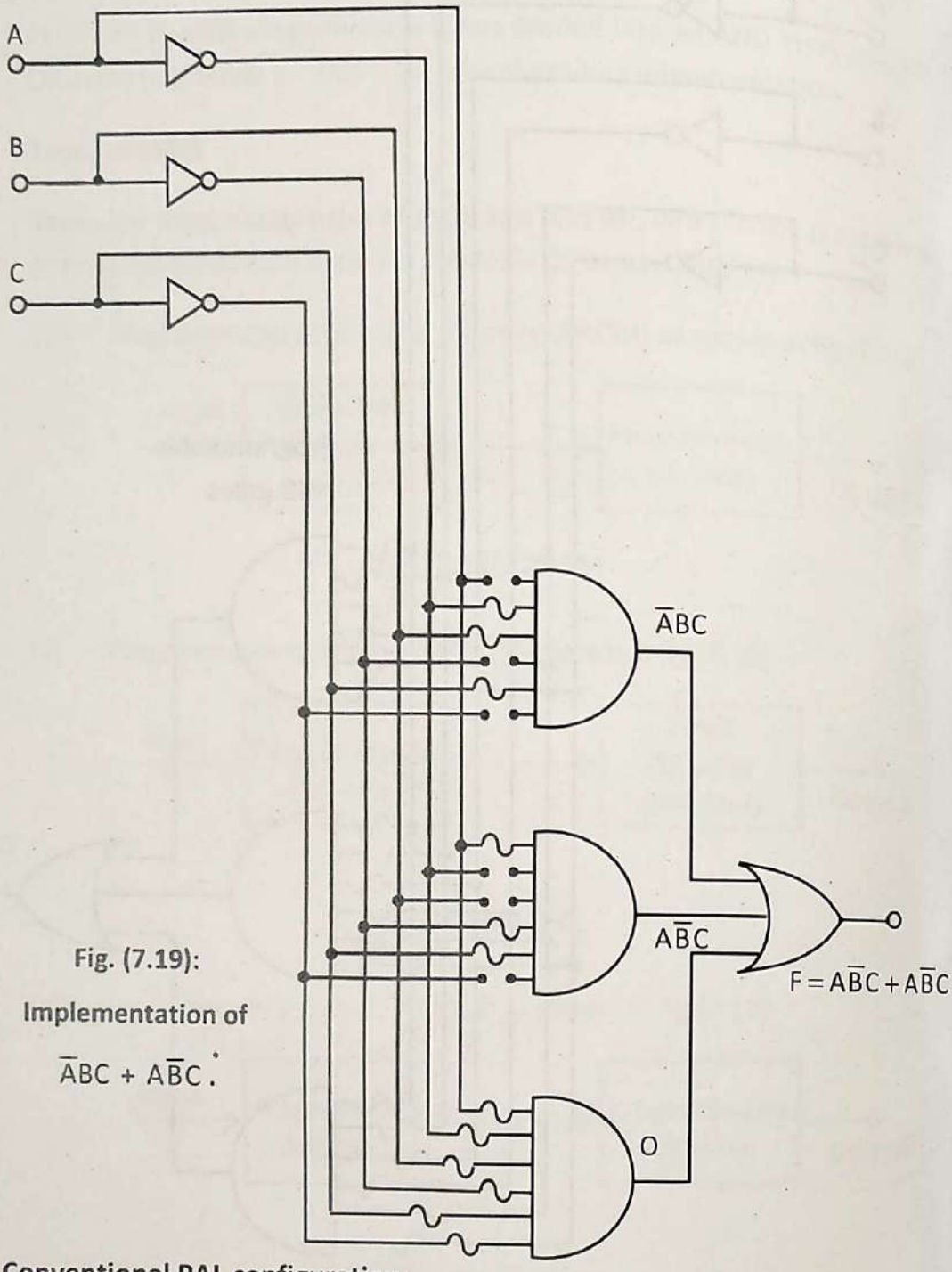


Fig. (7.19): Implementation of $\bar{A}BC + A\bar{B}C$.

Conventional PAL configuration:

The fig (7.20) shows a conventional means of representing PAL. And gate is drown with a single input line, where as in reality, it has three inputs. An X denotes a connection through an in fact fusible link and a dot sign represents a permanent connection. The absence of any symbol represents an open or no connection by virtue of burned – open link.

Example

Implement $F_1 = \bar{A}BC + A\bar{C} + A\bar{B}C$ and $F_2 = \overline{A\bar{B}\bar{C}} + BC$ using conventional PAL

Fig (7.22) shows the implementation of F_1 and F_2.

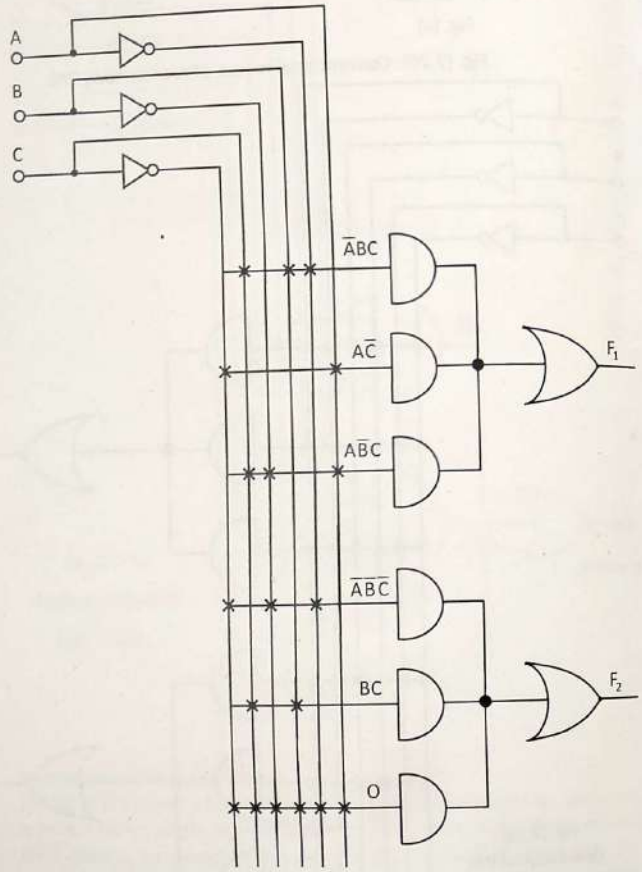


Fig. (7.22): implementation of F_1 and F_2

7.9 PROGRAMMABLE LOGIC ARRAY (PLA)

It consists of both programmed AND and OR arrays.

Types of PLA'S

There are two types of PLA'S according to their programming.

1. Mask- programmable PLA
2. Field – programmable PLA

In mask – programmable PLA, the customer must submit a PLA program table to the manufacturer. This table is used by the vendor to produce PLA.

The field programmable PLA can be programmed by means of certain procedures. Every individual user can program their application in the FPLA independently.

The most flexible logic array is the programmable logic array (PLA). Where both the AND and OR arrays can be programmed. The product term in AND array may be shared by any OR gate to provide the required sum of products implementation.

The internal logic of a PLA with three inputs and two outputs is shown in fig (7.23).

Inputs and their complements are available using buffer and inverter. Each input and its complement are connected to the inputs of each AND gate as indicated by the intersection between the vertical and horizontal lines. The outputs of AND gates are connected to the inputs of each OR gate.

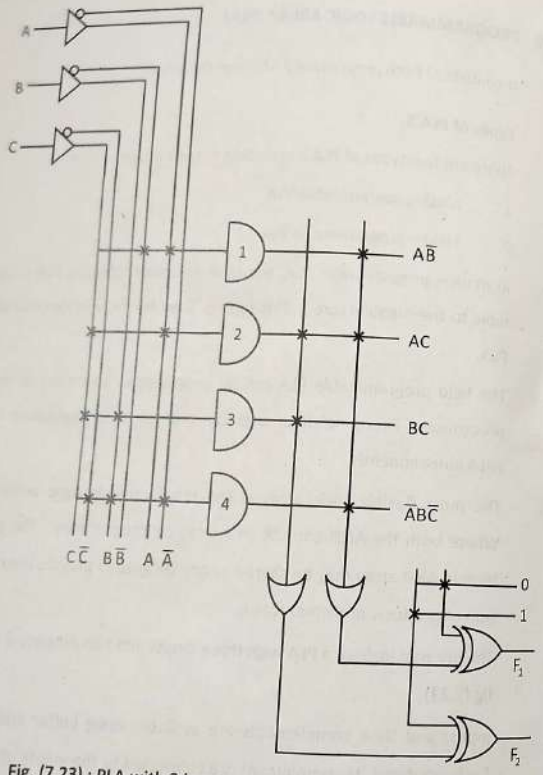


Fig. (7.23) : PLA with 3 inputs, two outputs and 4 product terms.

The output of the OR gate goes to an XOR gate where the other input can be programmed to receive a signal equal to either 1 or 0. The output is inverted when the XOR input is connected to 1. The output does not change when the XOR input is connected to 0. The particular Boolean functions implemented in PLA of fig (7.23) are

$$F_1 = A\bar{B} + AC + \bar{A}B\bar{C}$$

$$F_2 = \overline{(AC + BC)}$$

The product terms generated in each AND gate are listed along the output of the gate in the diagram. The product term is determined from the inputs whose cross points are connected and marked with X. The output of the OR gate gives the logic sum of the selected product terms. The output may be complemented or left in its true form.

PLA program table

The PLA program table contain product terms, Inputs appearing (complement, un-complement or not present, outputs interms of the product terms. This has to be submitted to the manufacturer to get the user – made PLA.

Example 1 :

Implement the combinational circuit for the functions using 3 × 3 × 2 PLA

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

$$F_1 = (A, B, C) = \sum m\,(4, 5, 7)$$

The simplified expressions for F_1 and F_2 are obtained using k–maps as shown in fig (7.24)

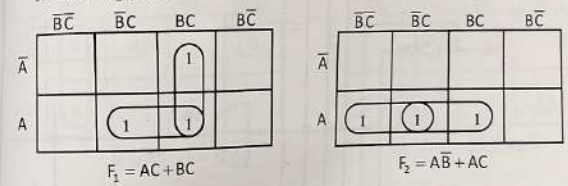


Fig. (7.24) : K-maps

The output equations are

$$F_1 = BC + AC$$

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

There are three distinct product terms in this combinational circuit, they are $A\bar{B}$, AC and BC. The circuit has 3 inputs. 3 –product terms and two outputs. The PLA program table is shown in table (7.1)

Table (7.1) : PLA program table

	Product term	Inputs A	B	C	Output F_1	F_2
$A\bar{B}$	1	1	0	-	-	1
AC	2	1	-	1	1	1
BC	3	-	1	1	1	-

It consists of 3 columns. First column lists the product terms, second column specifies the required paths between inputs and AND gates, third column specific the path between AND gates and OR gates.

For each product term, the input is marked with 1, 0 or – (dash). If a variable in the product term appears in the normal form, the corresponding input variable is marked with a 1. If it appears in its complement form, the corresponding input variable is marked with a 0. If the variable is absent in the product term, it is marked with a dash.

The logic diagram is shown in fig (7.25)

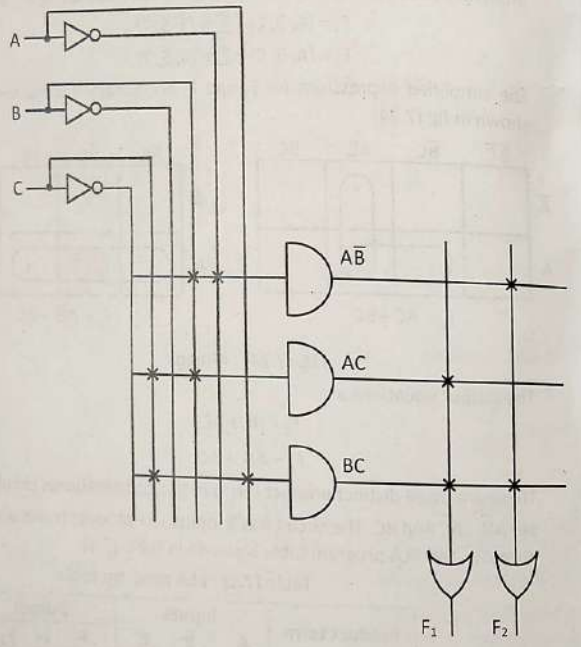


Fig. (7.25) : Implementation of PLA for example (1)

Example (2)

* Implement the combinational logic circuit defined by the functions $f(A, B, C) = \sum(3, 5, 6, 7)$, $F_2 = \sum(0, 2, 4, 7)$ using $3 \times 4 \times 2$ PLA.

The simplified expressions for F_1 and F_2 are obtained using k–maps as shown in fig (7.26)

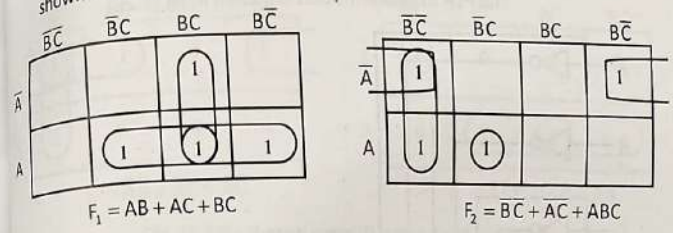


Fig. (7.26) : K-map simplifications for F_1 and F_2

The output functions contain totally 6 – product terms, but given PLA has only 4 product terms. So, it is not possible to implement directly. To reduce the total number of product terms. Let us consider the complement of the functions as shown in fig (7.27).

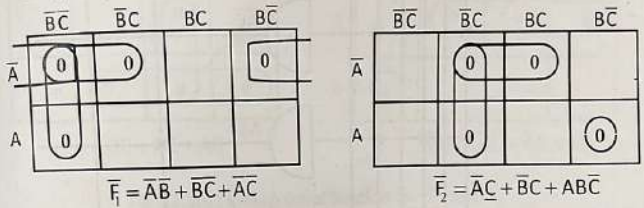


Fig. (7.27) : K-map simplification for $\bar{F}_1$ and $\bar{F}_2$

By taking $\bar{F}_1$ and F_2, we get only 4 product terms. They are $\bar{B}\bar{C}, \bar{A}\bar{B}, \bar{A}\bar{C}$ and ABC.

The program table is shown in table (7.2)

Table (7.2)

	Product term	Inputs A	B	C	Output F_1	F_2
$\bar{B}\bar{C}$	1	-	0	0	1	1
$\bar{A}\bar{C}$	2	0	-	0	1	1
$\bar{A}\bar{B}$	3	0	0	-	1	-
ABC	4	1	1	1	-	1

C	T

The PLA implementation is shown in fig (7.28)

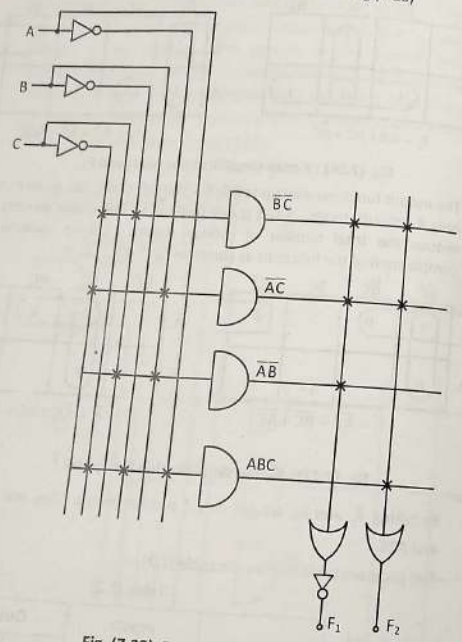


Fig. (7.28): PLA implementation

Example (3)

Implement the following two Boolean functions with a PLA.

$F_1 (A, B, C) = \sum (0, 1, 2, 4)$

$F_2 (A, B, C) = \sum (0, 5, 6, 7)$

The two given functions are simplified using k–map as shown in fig (7.29)

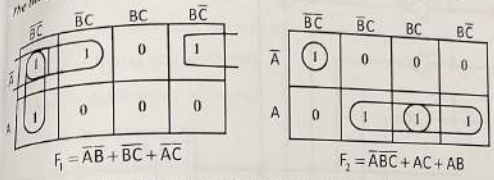


Fig. (7.29) : K-map simplification for F_1 and F_2.

The output functions, these contain totally six product terms, these can be reduced by simplifying the complements of F_1 and F_2, as shown in fig (7.30)

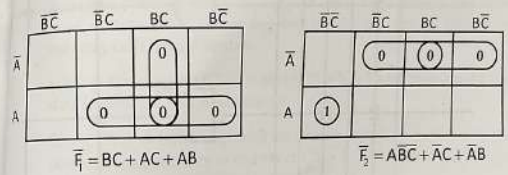


Fig. (7.30) : k-map simplification for $\overline{F}_1$ and $\overline{F}_2$

To reduce the total number of product terms consider $\overline{F}_1$ and $\overline{F}_2$ outputs which contain only four product terms. They are AB, AC, BC, and $\overline{ABC}$. The program table is as shown in fig (7.31)

Product term	Inputs			Output	
	A	B	C	F_1	F_2
AB	1	1	-	1	1
AC	1	-	1	1	1
BC	-	1	1	1	-
$\overline{ABC}$	0	0	0	-	1
				C	T

Fig (7.31) program table for example 3

The PLA implementation is shown in fig (7.32)

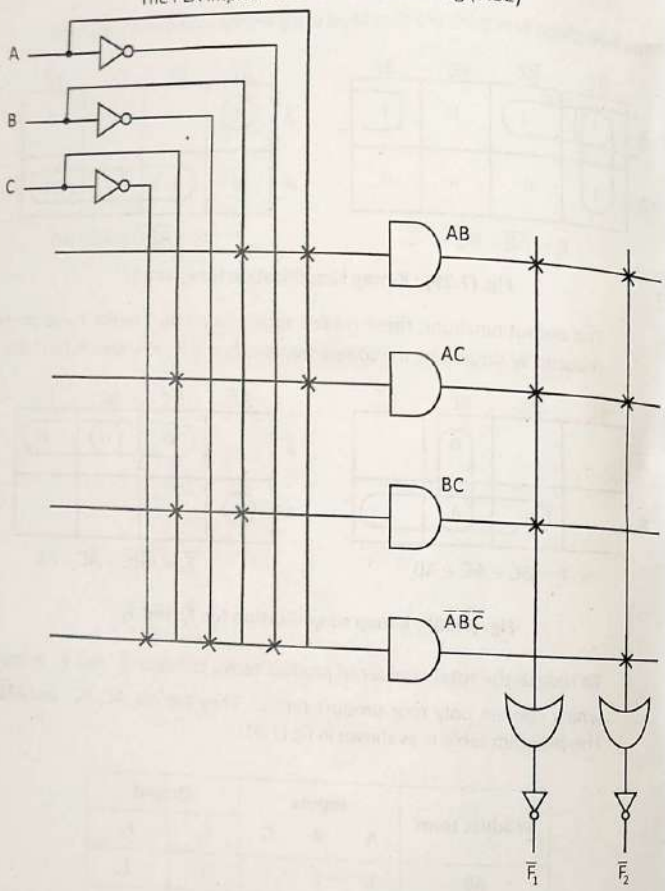


Fig. (7.32) : PLA implementation for Example (3)

SUMMARY

In this chapter, you will be studying

1. Internal organization of memory chip
2. Classification of RMS and ROMS
3. Bipolar and Dynamic memories
4. Expanding memory size
5. Programmable logic devices.

QUESTION & ANSWERS

1. A certain memory has a capacity of 16 k × 32. How many words does if store what is the number of bits per word. How many memory cells does it contain.
2. What is the capacity of the memory that has 1b address inputs, four data inputs and four data outputs.
3. Given 32 × 8 ROM chip with an enable input, show the external connections necessary to construct 128 × 8 ROM with four chips and a decoder.
4. Design a RAM system of 4k × 8 bit capacity using 2k × 4 RAM devices with read/write, chip enable (CE) and output enable (OE) controls.
5. Draw the structure of PLA for designing a BCD to seven segment decoder.
6. Draw the circuit diagram of a PLA to implement the following functions.

   $f_1 (A, B, C) = \Sigma_m (1, 4, 6, 7)$

   $f_2 (x, y, z) = \Sigma_m (0, 2, 4, 6)$

UNIT – VIII

LOGIC FAMILIES

INTRODUCTION

An integrated circuit (abbreviated IC) is a silicon semiconductor crystal, called a chip, containing the electronic components for constructing gates. The various gates are interconnected inside the chip to form the required circuit. The chip is mounted in a ceramic or plastic container, and connections are welded to external pins to form integrated circuit. The number of pins range from 14 on a small IC to several thousands on a large IC. Each IC has a number designation printed on the package for identification.

8.1 CLASSIFICATION OF ICs

Digital ICs are often categorized according to their complexity as measured by number of logic gates in a single IC. They are classified as, small scale integration devices (SSI): These ICs contains number of gates usually fewer than 10. ex: Basic gates and flip–flops.

Medium scale integration (MSI) devices: These ICs have complexity of 10 to 1000 gates in a single package.

Examples of MSI devices are, adders, multiplexers, registers and counters.

Large – scale integration (LSI) devices: These ICs contain thousands of gates in a single package.

Examples of LSI devices are processors, memory chips, and programmable logic devices.

Very – large scale integration (VLSI) devices: These ICs contains hundreds of thousands of gates with in a single package.

Examples of VLSI devices are large memory arrays, and complex microcomputer chips. Because of their small size and low cost, VLSI devices have revolutionized the computer design technology.

8.2 DIGITAL LOGIC FAMILIES

Digital ICS are classified not only by their complexity but also by specific circuit technology to which they belong. The circuit technology is referred to as a digital logic family. Each logic family has its own basic electronic circuit, which may NAND, NOR or NOT gate. Based on technology they are broadly classified as Bipolar logic families and unipolar logic families. Bipolar logic families are again classified in to two categories depending upon whether the transistors used were driven into saturation or not as saturated bipolar logic families and non saturated bipolar logic families. The various bipolar logic families are

1. Resistor – transistor logic (RTL)
2. Diode – transistor logic (DTL)
3. Transistor – transistor logic (TTL)
4. High – threshold logic (HTL)
5. Integrated – Injection logic (I^2 L)

and the various Non – saturated logic families are

1. Schottky TTL
2. Emitter – coupled logic (ECL)

MOS devices are unipolar devices and only MOSFETS are employed in MOS logic circuits. The various MOS logic families are

1. PMOS
2. NMOS
3. CMOS

While in PMOS only p – channel MOSFETS are used and in NMOS only n–channel MOSFETS are used; in complementary MOS (CMOS), both P and n – channel MOSFETS are used.

The basic circuit in each IC logic family is either a NAND or NOR gate. Thus basic circuit is primary building block from which all other more complex digital components are obtained.

8.3 CHARACTERISTICS OF DIGITAL ICs

With the widespread use of ICs in digital systems and with the development of various technologies for fabrication of ICs, it has become necessary to be familier with the characteristics of IC families. The various characteristics are

1. Speed of operation
2. Power dissipation
3. Figure of merit
4. Fan out
5. Current and voltage parameters
6. Noise immunity.
7. Operating temperature range
8. Power supply requirement
9. Flexibilities available.

Speed of operation (propagation delay)

The propagation delay of a gate is the average transition – delay time for the signal to propagate from input to output when the binary signal changes in value. The signals through a gate take a certain amount of time to propagate from the inputs to the output. This interval of time is defined as the propagation delay of the gate. Propagation delay is measured in nano seconds (ns)

The signals that travel from the inputs of a digital circuit to its outputs pass through a series of gates. The sum of the propagation delays through the gate is the total delay of the circuit.

The average propagation delay time of a gate is calculated from the input and output wave forms, as shown in fig (8.1)

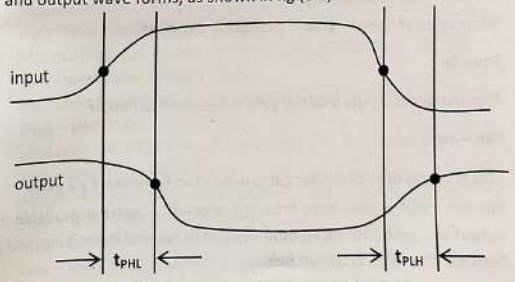


Fig (8.1) : Measurement of propagation delay

The signal – delay time between the input and output when the output changes from high to low level referred to t_{PHL}. Similarly, when the output goes from low to high level, the delay is t_{PLH}. It is customary to measure the time between the 50% point on the input and output transitions. The average propagation delay time is calculated as average of the two delays.

$$\therefore t_{pd} = \frac{t_{PLH} + t_{PHL}}{2}$$

Power dissipation

This is the amount of power dissipated in an IC. It is determined by the current, I_{cc} that it draws from the supply V_{CC} and is given by

$$P_D = V_{CC} \times I_{CC}$$

Where I_{CC} is the average value of I_{CC} (0) and I_{cc} (1)

Where I_{CC} (0) is the current drawn by the IC when all gates in the IC are in low state

I_{CC} (1) is the current drawn by the IC when all the gates in the IC are in high state.

It is measured in milliamperes. Power consumed by an IC is the product of power dissipated by each gate and number of gates.

Figure of merit:

Figure of merit of a digital IC is defined as the product of speed and power

∴ Figure of merit = propagation delay (ns) × power (mw)

It is specified in pico joules.

A low value of speed – power product is desirable.

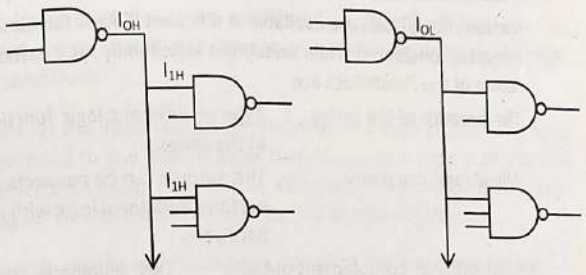
Fan – in

The number of inputs that the gate is designed to handle.

Fan – out:

This is the number of similar gates which can be driven by a gate.

The fan – out is calculated from the amount of current available in the output of a gate and the amount of current needed in each input of gate. Consider the fig (8.2) shown below.



(a) High-level output (b) Low-level output

Fig (8.2) fan – out calculation.

The output of one gate is connected to one or more inputs of other gates. The output of the gate is in the high voltage level in fig (a). It provides a current source I_{OH} to all gate inputs connected to it. Each gate input requires a current I_{IH} for proper operation. Similarly, the output of the gate is in the low voltage level in fig (b). It provides a current of I_{OL} for all the gate inputs connected to it. Each gate requires a current I_{IL}. The fan out of the gate is calculated from the ratio I_{OH} / I_{IH} or I_{OL} / I_{IL} which ever is smaller.

Operating temperatures

Being made with semiconductor devices, ICs are temperature sensitive. The accepted temperature ranges are 0^0c to 70^0c for consumer and industrial applications, and - 55^0c to + 25^0c for military purpose.

Current and voltage parameters:

High level input voltage (V_{IH}) :
Minimum input voltage recognized by the gate as logic 1.

Low - level input voltage (V_{IL}) :
Maximum input voltage recognized by the gate as logic 0.

High - level input voltage (V_{OH}):
Minimum output voltage corresponding to logic 1.

Low - level input voltage (V_{OL}) :
Maximum output voltage corresponding to logic 1.

High – level input current (I_{IH}) :
Minimum input current corresponding to 1 level voltage

Low – level i/p current (I_{IL}) :
Minimum input current corresponding to 0 level voltage.

High – level output current (I_{OH}):
Maximum current which the gate can sink in 1 level.

Low – level output current (I_{OL}) :
The Maximum current which the gate sink in 0 level.

Flexibilities available

Various flexibilities are available in different IC logic families and these must be considered while selecting a logic family for a particular job. Some of the flexibilities are

The breadth of the series : Type of different logic functions available in the series.

Wired logic capability : The outputs can be connected together to perform additional logic with out any extra hardware.

Availability of complement outputs : This eliminates the need for additional inverters.

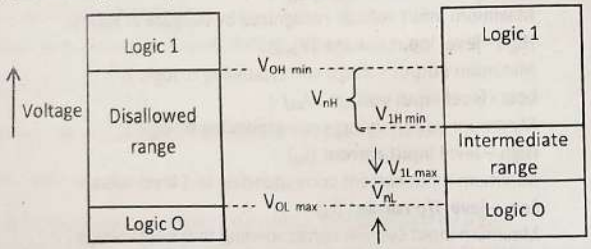
Type of output : Passive pullup, active pullup open collector, and tri – state.

Threshold voltage:

Threshold voltage is defined as that voltage at the input of gate which causes a change in the state of output from one logic level to other.

Noise Margin:

When digital ICS operate in noisy environment the gates may malfunction if the noise is beyond certain limits. The noise immunity of a logic circuit refers to the circuit's ability to tolerate noise voltages at its inputs. A quantitative measure of noise immunity is called noise margin. Noise may be dc noise or ac noise. A drift in the voltage levels of signals is called dc noise. The ac noise is a random pulse caused by other switching signals. Noise margin is expressed in volts and represents the maximum noise signal that can be added to the input signal of a digital circuit without causing an undesirable change in the circuit output. Fig (8.3) shows the range of output and input voltage that can occur in a logic circuit.



(a) Output voltage ranges (b) Input voltage ranges

Fig (8.3) : DC noise margins

In fig (a) voltages greater than $V_{OH(min)}$ are considered as logic 1 and voltages lower than $V_{OL(max)}$ are considered as logic 0. Voltages in the disallowed range should not appear at a logic circuit output under normal conditions.

Fig (b) shows the input voltage requirement of a logic circuit. The logic circuit respond to any input greater than $V_{IH(min)}$ as a logic 1 and to any input lower than $V_{IL(max)}$ as a logic 0. Voltages in the intermediate range will produce an unpredictable response and should not be used.

Noise margin may be high state noise margin or low state noise margin.

High state noise margin (NM_H) is

$$V_{NH} = V_{OH}\ (min) - V_{IH}\ (min)$$

Low state noise margin (NM_L) is

$$V_{NL} = V_{IL}\ (max) - V_{OL}\ (max).$$

8.4 DIODE TRANSISTOR LOGIC (DTL)

DTL is some what more complex than RTL but because of greater fan out and improved noise margins it has replaced RTL. Its main draw back is slower speed, because of this it was modified and emerged as transistor – transistor logic (TTL), which is the most popular logic family today.

The voltage levels for DTL are 0.2v for logic 0 and between 4 – 5v for logic 1. The circuit for DTL Nand gate is shown in fig (8.4)

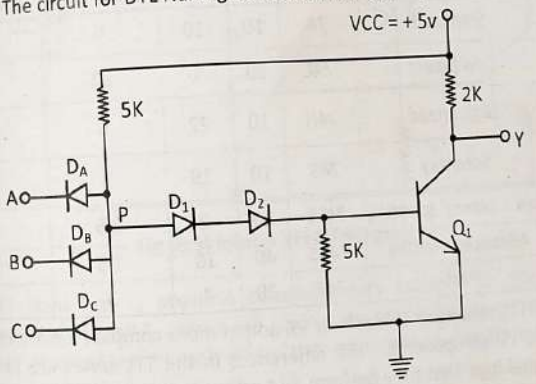


Fig. (8.4) : DTL Nand gate

The operation of DTL Nand gate is explained as follows. If any input diode conducts current through V_{CC} and the 5kΩ resistor into the input diode. The voltage at point P equals to the input voltage plus a diode drop (0.7v), for a total of 0.9v. In order for the transistor Q_1 to conduct, the voltage at point P must overcome a potential of one V_{BE} drop plus two drops across D_1 and D_2 (3 × 0.7 = 2.1v). Since the voltage at point p is maintained at 0.9v by the input conducting diode; the transistor is cutoff and the output voltage is High (+ 5v).

If all the inputs of the gate are High, the inputs diodes D_A, D_B and D_C are reverse biased. The voltage at point P is now equal to V_{BE} plus the two diode drops across D_1 and D_2 (3 × 0.7 = 2.1v). The base current is equal to the difference of currents flowing in the two 5kΩ. Resistors and is sufficient to drive the transistor into saturation. With transistor in saturation, the output voltage drops to V_{CEsat} = 0.2v; which is low level for the gate.

8.5 TRANSISTOR – TRANSISTOR LOGIC (TTL)

This family is so named because of its dependence on transistors alone to perform basic logic operations. It uses transistors operating in saturation mode. There are several subfamilies or series of the TTL technology. The following table illustrate the subfamilies.

TTL series name	Prefix	Fan out	PD (min)	Propagation delay (ns)	Speed power product
Standard	74	10	10	9	90
Low power	74L	20	1	33	33
High speed	74H	10	22	6	132
Schottky	74S	10	19	3	57
Low - power schottky	74LS	20	2	9.5	19
Advanced shottky	74AS	40	10	1.5	15
Fast	74F	20	4	3	12

All TTL series are available in SSI and in more complex form such as MSI and LSI components. The differences in the TTL series are not in the digital logic that they perform, but rather in the internal construction of the basic NAND gate. In any case, TTL logic gates in all the available series come in three different types of output configuration.

1. Open – collector output
2. Totem – pole output
3. Three – state (Tri state) output.

Totem – pole output:

The basic circuit of TTL NAND gate is given in fig (8.5). The input transistor has multi – emitter input, each emitter acting like a diode and, therefore, along with R_1, acts as 2 – input AND gate. The rest of the circuit functions as an inverter. A transistor is used in the input stage instead of diodes to achieve higher speed of operation.

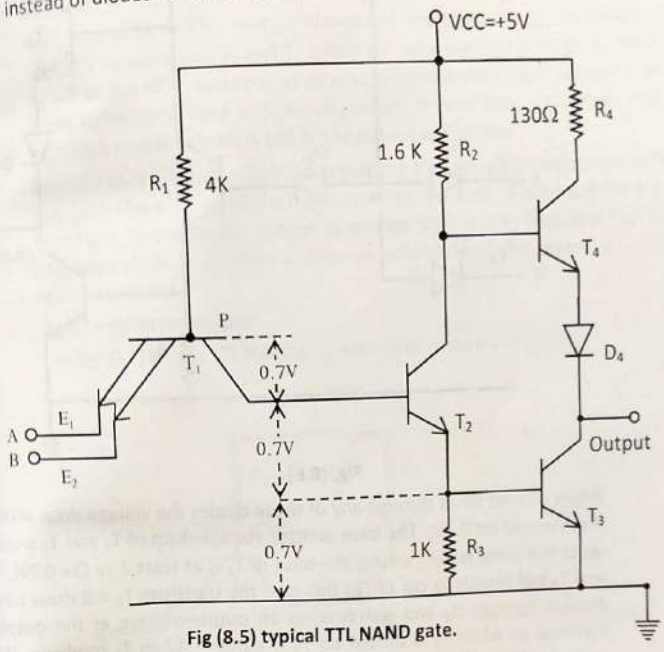


Fig (8.5) typical TTL NAND gate.

The transistor T_2 provides complementary voltages for the output transistors T_3 and T_4, which are stacked one above the other in a totem – pole arrangement in such a way that while one of these conducts the other is cutoff.

To analyze the circuit replace the transistor T_1 with three diodes which represents the three junction as shown in fig (8.6)

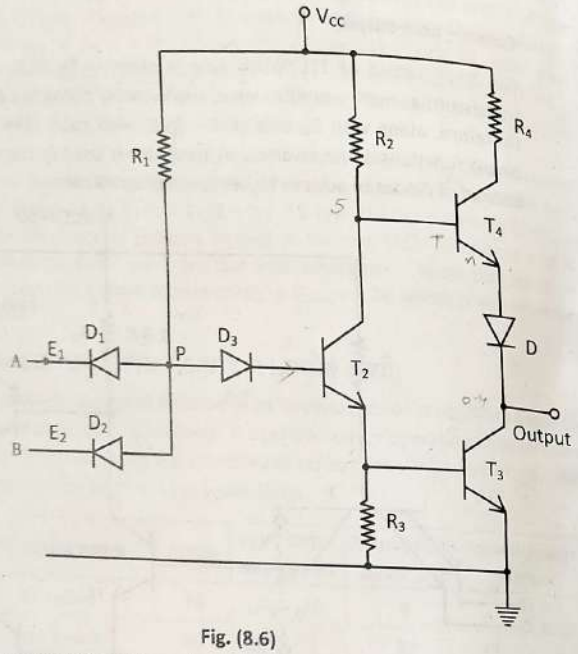


Fig. (8.6)

When current flows through any of these diodes the voltage drop across them would be 0.7v. The base emitter voltage drop of T_2 and T_3 would be of the same order. Unless the base of T_1 is at least 2.1v (3× 0.7v), T_2 and T_3 will remain in cut off. In this case the transistor T_4 will draw base current through R_2 and will develop an output voltage at the output terminal as while T_4 is turned on T_3 is cutoff. When T_4 conducts, the output voltage will be about 3.9v instead of full supply voltage, as there is voltage drop across R_4 and D_4.

When the base of T_1 is 2.1v or more, T_2 and T_3 begin to conduct and the later saturates reducing the output voltage to about 0.4v. Diode D_4 is used in the circuit to develop a voltage drop across it, so that T_4 does not conduct when T_3 is conducting. Now the NAND gate operation is explained as follows.

Case (i) When both A and B inputs are low (0.4v)

In this case, both the input junctions will conduct and the voltage at the base of T_1 (point p) will be the input voltage plus the voltage drop across D_1 and D_2. If the input is 0.4v, the voltage at the base of T_1 will be 0.4 + 0.7 = 1.1v. Current will now flow through R_1 and E_1 or E_2, which ever has a lower input voltage. Since the voltage at the base of T_1 is less then required to forward–bias T_2 and T_3, which is 2.1v, T_2 and T_3 will not conduct and, therefore, the output voltage will be high (2.4–3.9V). We conclude that a low input at either E_1 or E_2 or both will produce high out put.

Case (ii) : When the input voltage at both A and B is High.

Both D_1 and D_2 will be reverse biased and therefore T_1 will not conduct. This will forward bias T_2 and T_3 which will now conduct through R_1 while T_4 will be cut off T_3 will now be saturated making the output voltage, that is the collector voltage of T_3 approximately 0.4v or less. This shows that with high inputs at both A and B, the output will be low.

Because of the totem – pole arrangement the output impedance is very low both when the output is high and when it is Low. A low impedance is necessary to enable the output to change from low to high and high to low logic state in as short a time as possible and this means a faster switching speed

Open – collector output:

The fig (8.7) shows TTL Nand gate with open – collector output.

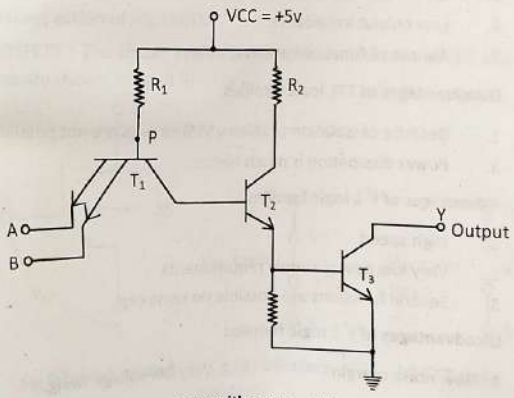


Fig. (8.7) : TTL Nand gate with open – collector output.

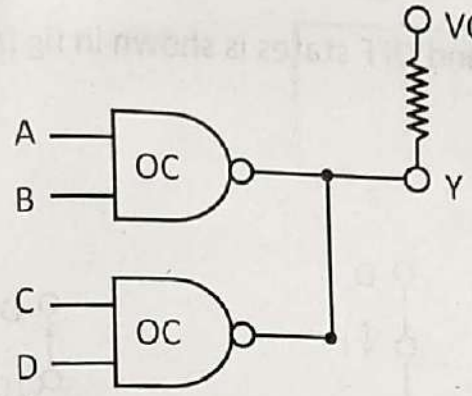


Fig. (8.8a) : Physical connection

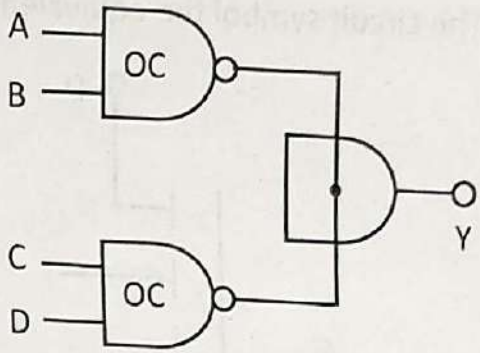


Fig. (8.8b) : Graphic symbol

The output Y = $\overline{AB}.\overline{CD} = \overline{AB + CD}$

8.6 MOS LOGIC FAMILIES

For very large integrated circuits we require very small logic gates. Metal oxide semiconductor FETS meet this requirement to a very large extent and at present these MOSFETS are extensively used in pocket calculators, digital watches and in many areas. They not only meet the requirement of small size, but they are also very economical in cost, and consume very little power, although they are not as fast as TTL.

Mos logic families are categorized into three different families, namely,

PMOS (P – channel NOS) logic families.
NMOS (N – channel MOS) logic families
CMOS (Complementary MOS) logic families.

Switching action of MOSFETSs

NMOSFETS : The circuit symbol, the equivalent circuits in ON and OFF states are shown in fig (8.9)

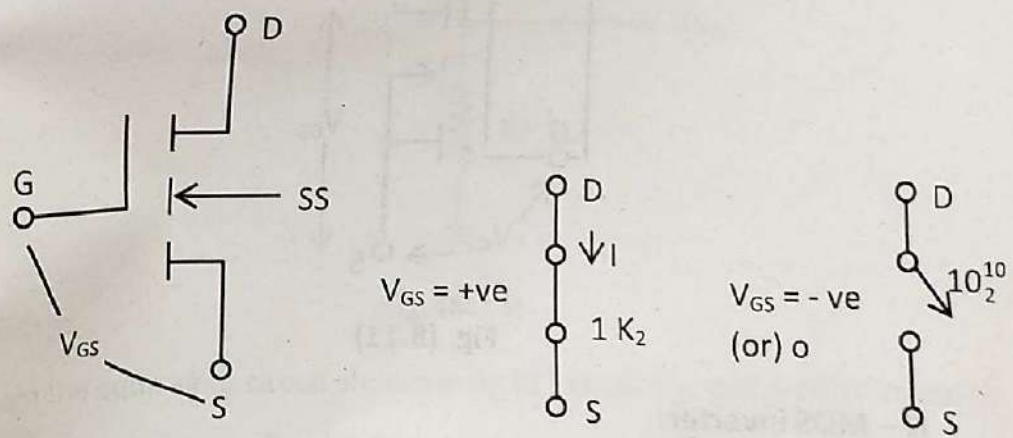
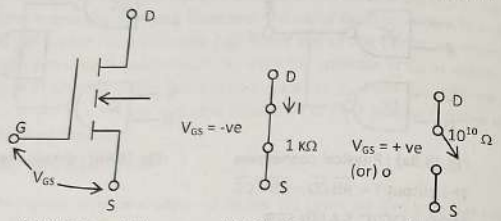


Fig (a) circuit symbol **(b) ON state** **(c) OFF state**

Fig. (8.9)

PMOSFET :

The circuit symbol the equivalent in ON and OFF states is shown in fig (8.10)



(a) Circuit symbol (b) ON state (c) OFF state

Fig. (8.10)

MOSFET as a resistor

The Gate is permanently connected to Drain, so it is always in the ON state and the MOSFET acts as a resistor of value R_{ON}. MOSFETS used as load resistor is designed to have a narrow channel, so its R_{ON} is much greater than the R_{ON} of the switching MOSFET. Typically its R_{ON} = 100KΩ as shown in fig (8.11)

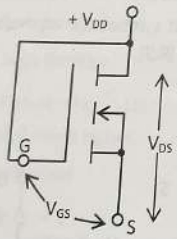


Fig. (8.11)

N – MOS inverter:

The construction of N – MOS inverter is shown in figure (8.12) below

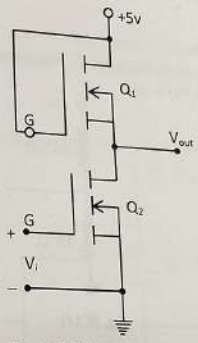


Fig (8.12) N – MOS inverter

In the above figure (Q_1) is a load MOSFET and Q_2 is a switching MOSFET. The following conditions illustrate the operation of N – MOS inverter.

Case (i) When V_{in} = 0v,

Q_1 is always in ON state and for

Vin = 0, Q_2 will be off. The equivalent circuit for case (i) is shown in fig (8.13)

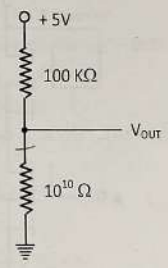


Fig. (8.13)

From the equivalent circuit shown in fig (8.14) the out put is determined as

$$V_{our} = V_{DD} \cdot \frac{R_{OFF}(Q_2)}{R_{ON}(Q_1) + R_{OFF}(Q_2)} = \frac{5 \times 10^{10}}{100 \times 10^3 + 10^{10}} \approx 5V.$$

Case (ii) $V_{in} = +5V$,

Q_2 is ON, so its R_{ON} is 1 K Ω. The equivalent circuit for case (ii) is shown in fig (8.14)

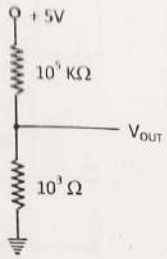


Fig. (8.14)

From the above fig (8.14), the output is determined as

$$V_{out} = \frac{V_{DD} \times R_{ON}(Q_2)}{R_{ON}(Q_1) + R_{ON}(Q_2)} = \frac{5 \times 10^3}{10^5 + 10^3} = 0.05V. \approx 0V.$$

N – MOS NAND gate

The figure (8.15) shows the construction of N – MOS NAND gate. It can be analyzed in the same way as N – MOS inverter is analyzed.

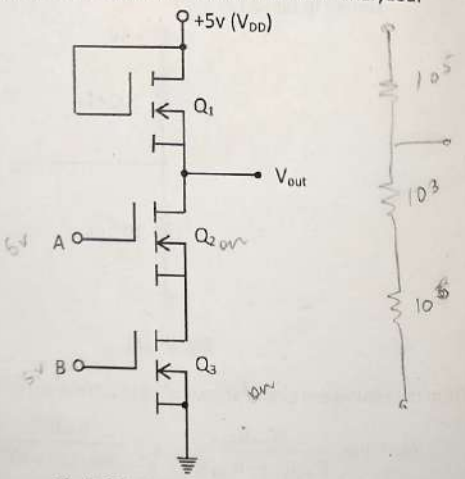


Fig (8.15) N – MOS NAND gate

CMOS logic:

It uses both P – MOS and N – MOS devices in the same circuit.

CMOS logic family is faster and consumes less power than other MOS logic families.

C – MOS inverter:

The following figure (8.16) shows the construction of C – MOS inverter.

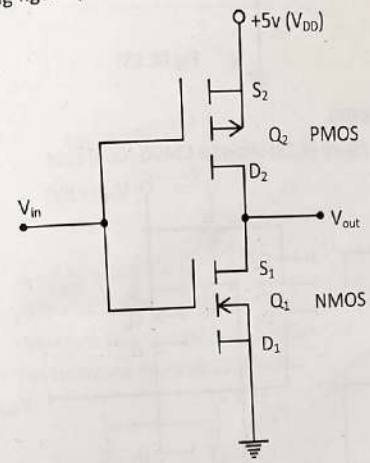


Fig (8.16) C – MOS inverter

The following cases illustrate the operation of C – MOS inverter.

Case (i) V_{in} = Ov (low)

For $V_{in} = 0$, $VGS_1 = 0$ and $VGS_2 = 5$ v

Therefore Q_2 is ON and Q_1 is OFF. The equivalent circuit is as shown in fig (8.17)

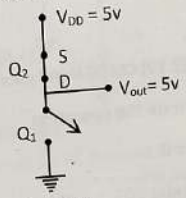


Fig (8.17)

Case (ii) When $VGS_2 = 0v$, and $VGS_1 = +5v$, Q_2 is OFF and Q_1 is ON, the equivalent circuit is as shown in fig (8.18)

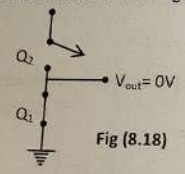


Fig (8.18)

CMOS NAND gate:

The following figure (8.19) shows a CMOS Nand gate

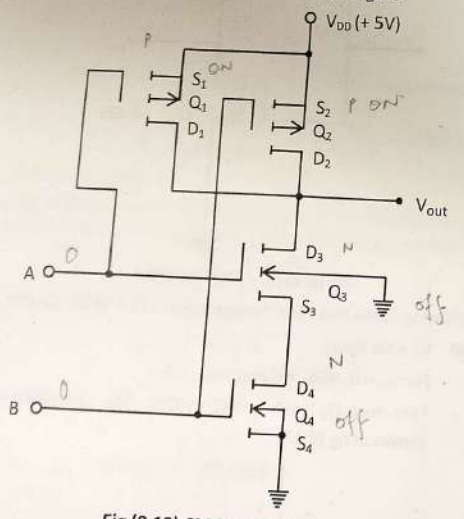


Fig (8.19) CMOS Nand gate

The following cases illustrate the operation

Case (i) When A = 0, B = 0

$VGS_1 = VGS_2 = -5v$ and $VGS_3 = VGS_4 = 0v$

Therefore Q_1 is ON, Q_2 is ON, Q_3 and Q_4 are OFF. The following figure (8.20) results

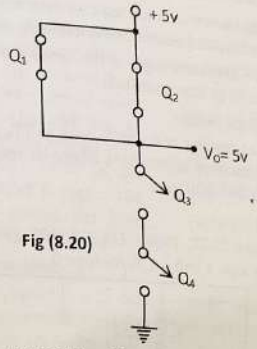


Fig (8.20)

Case (ii) A = +5v and B = +5v

$VGS_1 = VGS_2 = 0v$ and $VGS_3 = VGS_4 = +5v$

Therefore $Q_3 = Q_4$ = off and $Q_3 = Q_4$ = ON.

The following figure (8.21) results.

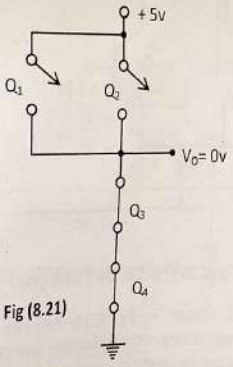


Fig (8.21)

Applications of CMOS logic

(1) Used to construct small, medium and large scale ICS for wide variety of applications ranging from general purpose logic to Micro processors.

(2) Because of its extremely small power consumption it is useful for applications in watches and calculators.

Case (ii) When $VGS_2 = 0v$, and $VGS_1 = +5v$, Q_2 is OFF and Q_1 is ON, the equivalent circuit is as shown in fig (8.18)

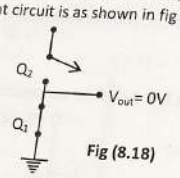


Fig (8.18)

CMOS NAND gate:

The following figure (8.19) shows a CMOS Nand gate

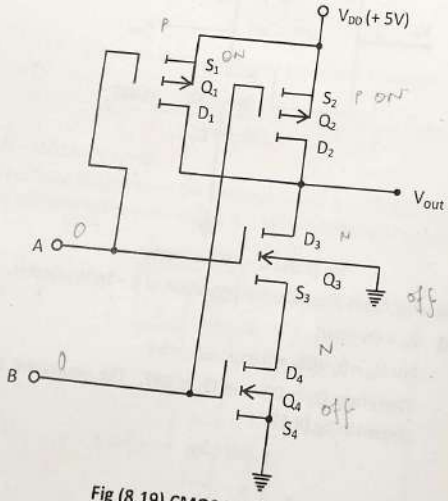


Fig (8.19) CMOS Nand gate

The following cases illustrate the operation

Case (i) When $A = 0$, $B = 0$

$VGS_1 = VGS_2 = -5v$ and $VGS_3 = VGS_4 = 0v$

Therefore Q_1 is ON, Q_2 is ON, Q_3 and Q_4 are OFF. The following figure (8.20) results

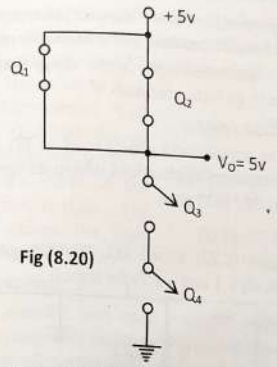


Fig (8.20)

Case (ii) A = + 5v and B = + 5v

$VGS_1 = VGS_2 = 0v$ and $VGS_3 = VGS_4 = +5v$

Therefore $Q_1 = Q_2$ = off and $Q_3 = Q_4$ = ON.

The following figure (8.21) results.

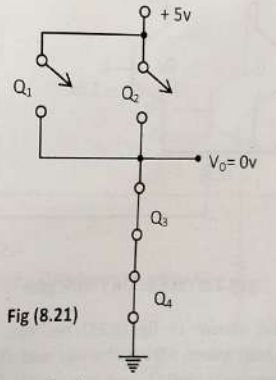


Fig (8.21)

Applications of CMOS logic

(1) Used to construct small, medium and large scale ICS for wide variety of applications ranging from general purpose logic to Micro processors.

(2) Because of its extremely small power consumption it is useful for applications in watches and calculators.

8.7 EMITTER – COUPLED LOGIC (ECL)

It is also called as current – mode logic (or) current steering logic. It is fastest of all logic families because of the following reasons.

1. It is a non – saturated logic, in the sense that the transistors are not allowed to go into saturation.
2. Limited voltage swing.

ECL is not as popular and widely used as TTL and MOS, except in very high frequency applications where its speed is superior. The basic gate is OR/ NOR gate.

ECL OR/NOR gate

The following figure (8.22) shows ECL OR/NOR gate. Logic levels are negative, - 0.9v for logic 1 and – 1.7v for logic 0.

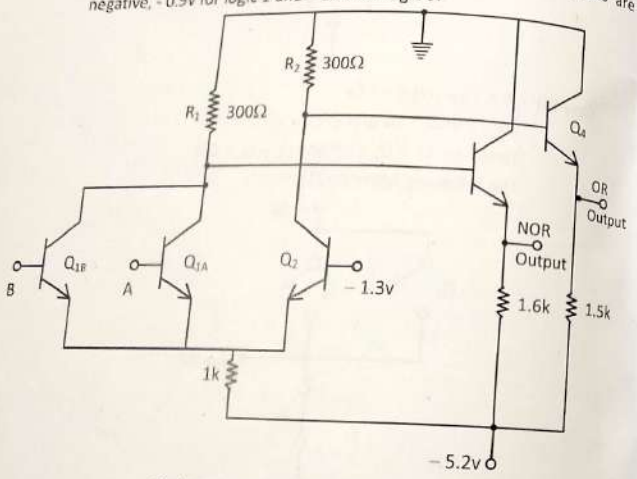


Fig (8.22) : ECL or OR / NOR gate

The above circuit shown in fig (8.23) has two outputs which are complement of each other. Transistor Q_{1A} and Q_2 from a differential amplifier. Transistors Q_{1A} and Q_{1B} are in parallel. Transistors Q_3 and Q_4 are emitter followers whose emitter voltages are the same as the base voltages (Less than 0.8v base to emitter voltage drops).

In fig (8.22) when the inputs A and B are low, i.e, – 1.7v, Q_2 is more forward biased than Q_{1A} and Q_{1B}, and so Q_2 is ON and Q_{1A} and Q_{1B} are OFF.

The value of R_2 is such that current flowing through Q_2 puts the collector at about – 0.9v. Therefore, the emitter of Q_4 is at – 0.9- 0.8 = - 1.7v and so, the OR gate output is low. The base current of Q_3 passing through R_1 if very small. The value of R_1 is such that this current puts collectors of Q_{1A} and Q_{1B} at about – 0.1v so the emitter of Q_3 is at – 0.1 – 0.8 = – 0.9v. i.e, the NOR gate output is High. The figure (8.23) shows the logic symbol of ECL OR/NOR gate.

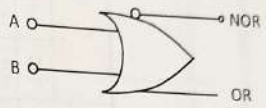


Fig (8.23) : logic symbol of ECL

Wired or connection:

ECL gates are available with open – emitter outputs. The open emitter outputs can be connected together directly, and the common emitter output terminals may be connected through an external resistor to a negative supply (-5.2v) to perform a wired – OR operation as shown in fig (8.24) below.

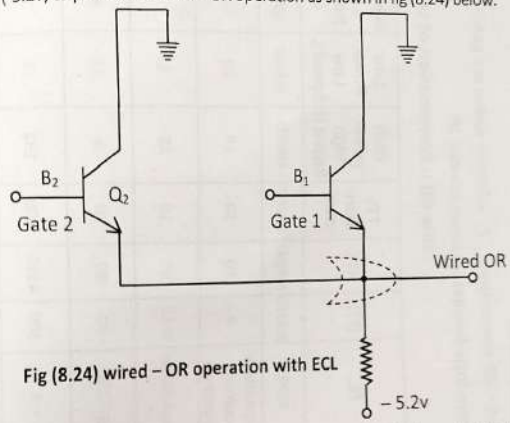


Fig (8.24) wired – OR operation with ECL

The truth table for the above wired OR operation is shown in fig (8.25)

B_1	B_2	Wired Or out put
-0.9v	-0.9v	-1.7v
-0.9v	-0.1v	-0.9v
-0.1v	-0.9v	-0.9v
-0.1v	-0.1v	-0.9v

Fig (8.25) truth table for wired – OR operation.

ECL logic circuits can be interfaced with other type of logic circuits using special level shifter circuits called level translators.

Advantages of ECL logic circuits:

1. Availability of wired – OR connection 2. No noise spikes are generated 3. Complementary outputs are also available

Comparison of different logic families is shown in table (8)

Table (8) :: Comparision of Logic Families

Parameter	RTL	I^2L	DTL	HTL	TTL Stan-dard	High power High Speed H	Low Power Low Speed L	Schottky Low Power LS	Advanced Low Power Schottky ALS	ECL	CMOS			
											74C	74HC	74HCT	74 AC
Basic gate	NOR	NOR	NAND	NAND	NAND	NAND	NAND	NAND	NAND	OR-NOR	NOR / NAND	NOR / NAND	NOR / NAND	NOR / NAND
Fan – out	5	Depends on current injection	8	10	10	10	20	20	10	40-55	50	20	20	50
Power dissipation / gate (mw)	12	6nw-70nw	8-12	55	10	22	1	2	19	0.75	0.01	0.0025	0.0025	0.005
Propagation deley (ns) / gate)	12	25 – 250	30	90	10	6	33	9.5	3	40	70	18	18	5.25
Speed – power product	144	< 1	300	4950	100	132	33	19	57		70.7	10.8	10.8	4.0
Noise immunity	Nomi nal	Poor	Good	Excell-ent	Very good	Very good	Very good	Very good	Very good	Poor	Very good	Very good	Very good	Very good