Decoders:-

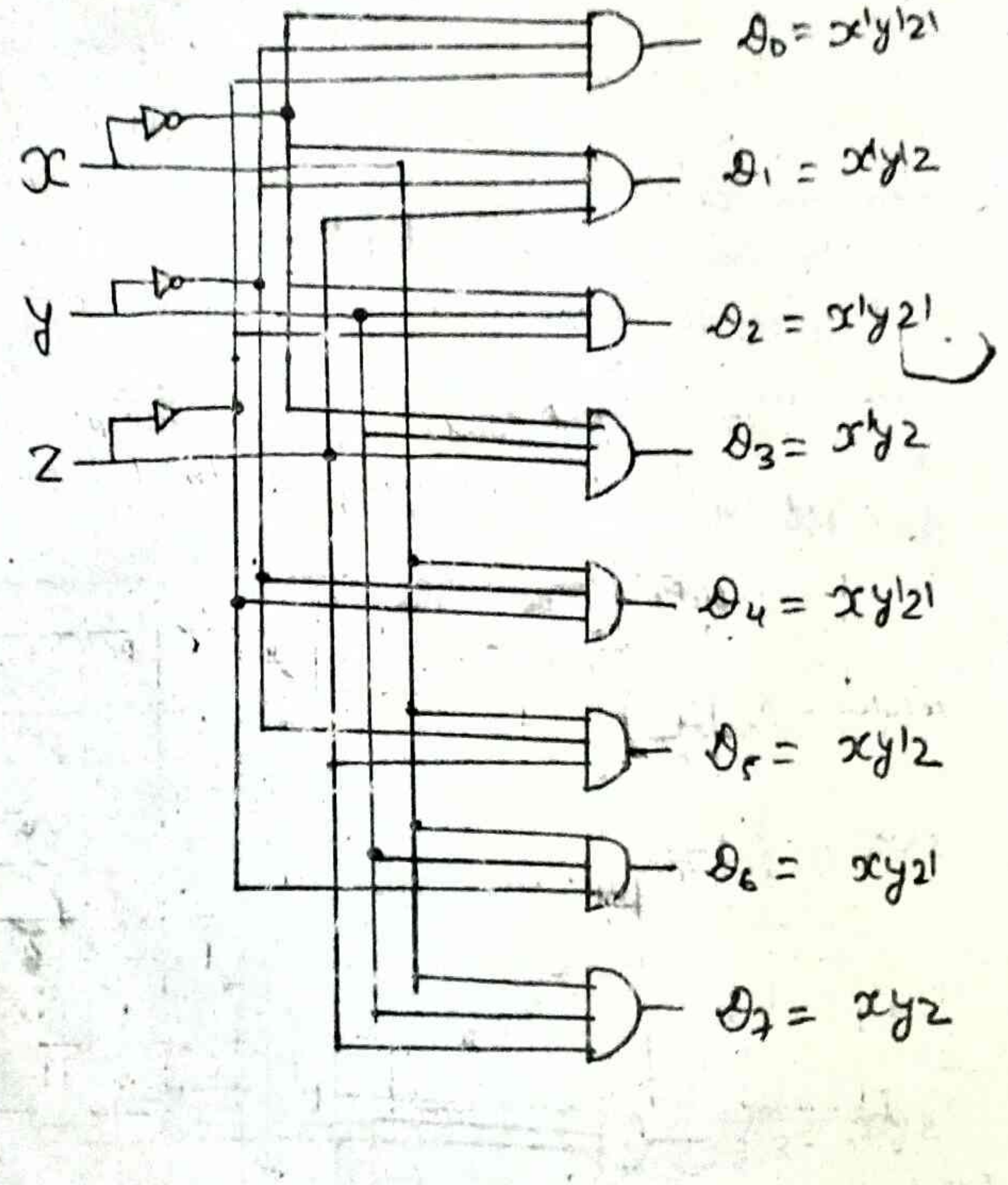
A combinational circuit that converts information of n input lines to a maximum of 2^n unique output lines. If any n-bit decoded information has unused or don't care condition combination the decoder will have less than 2^n outputs.

Here we are presenting n to m line decoder where $m \le 2^n$. Purpose of this decoder is to generate 2^n or less minterms of n input variables. Here the three inputs are converted into 8 outputs that is the reason this decoder is also known as 3 to 8 line decoder. Hence any 3 to 8 line decoder takes 3 inputs and generates 8 output lines. The input output relationship can be easily described by the following truth table-

Inputs			Outputs							
x	y	z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

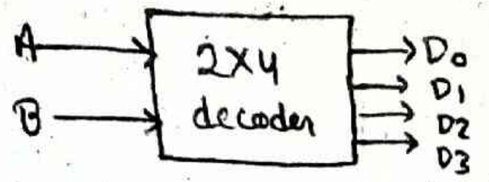
fig:- truth table for 3X8 line decoder.

Here the output is equal to 1 at any time and the valee 1 represents the minterm equivalent of binary number presently available input lines- The logical diagram can be easily drawn by observing the minterms as follows.

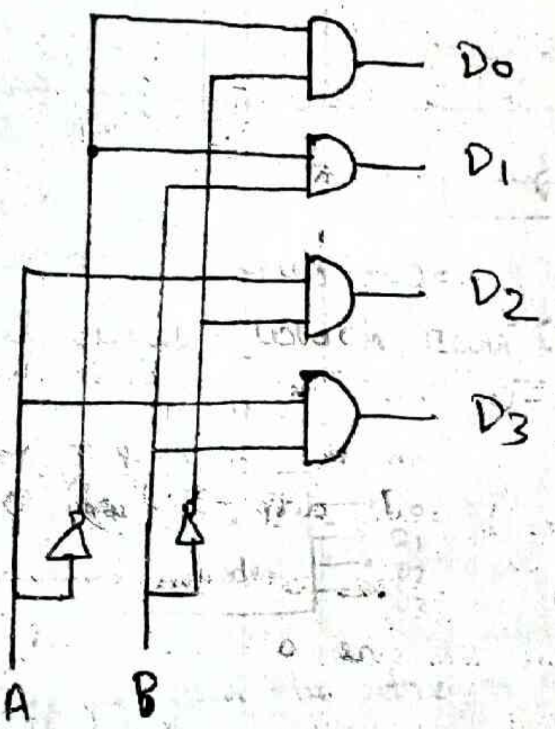


2X4 Decoder :-

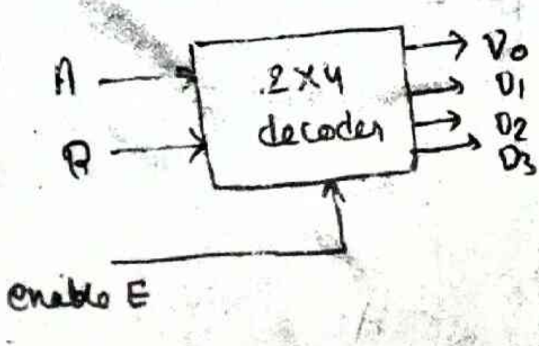
As 2X4 line decoder has two lines as inputs and 4 output lines. which can be shown in the fifure given below



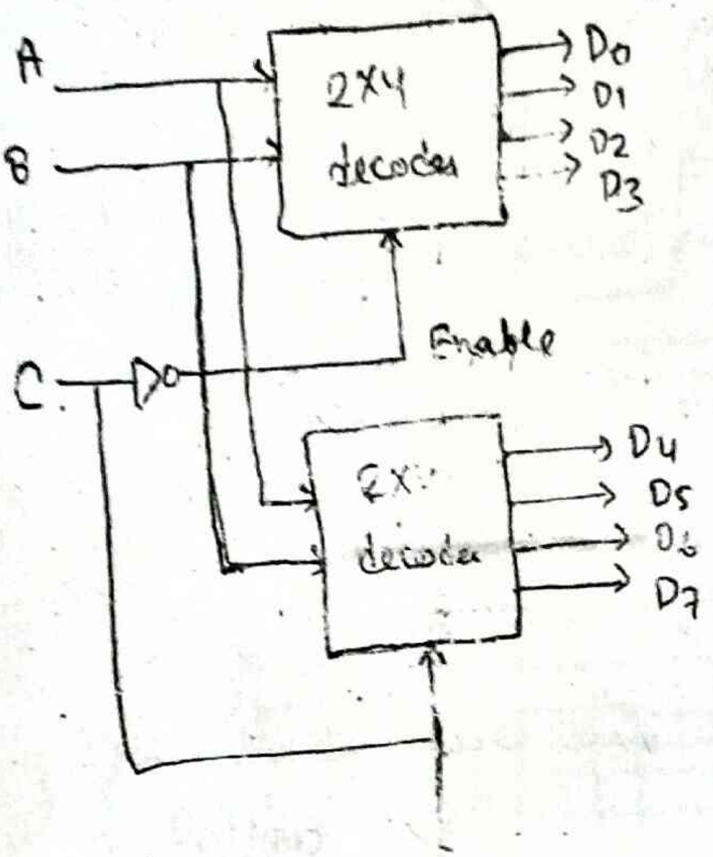
The logic diagram can be drawn as



If the enable input is 0 all the output of the circuit will be 0. when the enable input is 1 the circuit operates as a decoder. The block diagram of 2X4 decoder with enable input is as follow



Using 2X4 decoder we can generate 8X8 decoder as follow



Here when C=0, the top decoder is enabled and other is disabled. The bottom decoder output are all 0's and the top decoder provides outputs from D_0 to D_3 (000 to 011) when C =1, the enable condition are reversed, the output of the top decoder are all 0's and the bottom decoder provides outputs from D_4 to D_7 (100 to 111)

Encoders:-

2^n or less

An encoder is a combinational logic circuit that performs an operation reverse to that of a decoder. Means encoder has 2^n or less inputs lines and generates n output lines. The output lines generates the binary codes for the 2^n input veriables. Example can be given as the octal to binary encoder that consists of eight inputs, one for each of the eight digits and three outputs that generates the corresponding binary number. This encoder is constructed by using three OR gates that can be easily determined by observing the truth table.

Inputs								outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

octal to binary

By observing the truth table we can say that the lowser order output z is 1 in the case of odd input octal digit. Simillarly output y is 1 in the case of 2,3,6&7 and the output x is 1 in the case of 4,5,6&7 and hence the circuit can be drawn as given below—

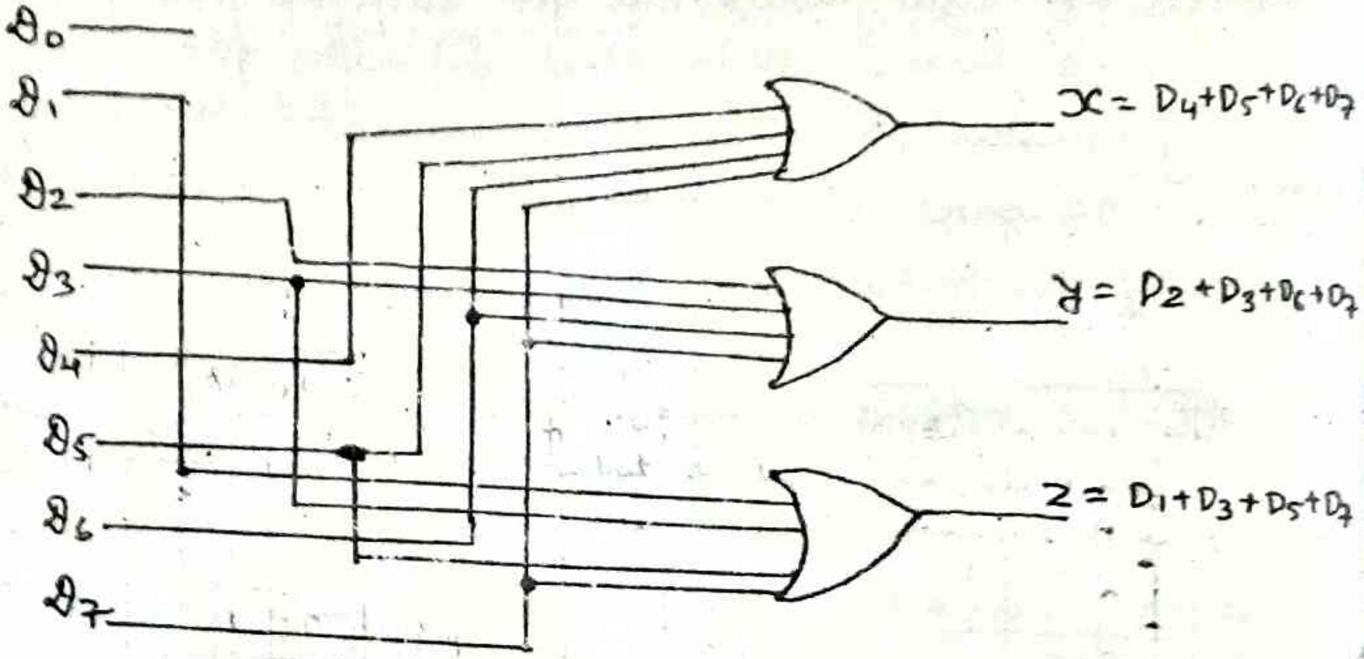


fig:- The logic circuit for octal to binary Encoder

we can also draw the diagram for decimal to BCD encoder, that consists of 10 input lines D_0 to D_9 and 4 output lines w, x, y & z. And are given by the relations

$$w = D_8 + D_9$$
$$x = D_4 + D_5 + D_6 + D_7$$
$$y = D_2 + D_3 + D_6 + D_7$$
$$z = D_1 + D_3 + D_5 + D_7 + D_9$$

16 — 4
— 4

The encoder in this case consists of 4 OR gates.
6

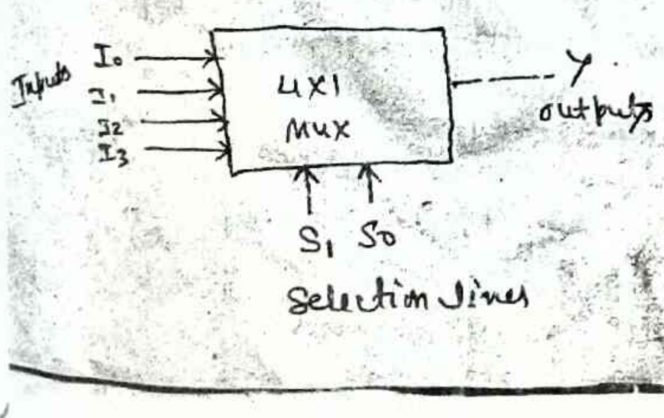
Multiplexers:-

Multiplexers are the combinational circuit that accept input from 2^n lines and give the output in a single output line. The selection of particular input line is controlled by a set of selection lines. Generally there are 2^n input lines and n selection lines whose bit combination determine which input is selected.

consider the following 4 to 1 line multiplexer-

This multiplexer consists of four input lines I_0 to I_3 is applied to one input of AND gate. selection lines S_1 & S_0 are decoded to select a particular AND gate. Consider the case when $S_1S_0 = 10$, the AND gate associated with input I_2 has two of its inputs equal to 1 and the third input connected to I_2. The other three AND gates have at least one input equal to 0 which makes their output equal to 0. The output of OR gate is now equal to the value of I_2, thus providing a path from the selected input to the output. The multiplexer selects one of many inputs and control the binary information to the output line and hence multiplexer is also known as the data selector.

The block diagram and functional table are as follows—



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Functional table

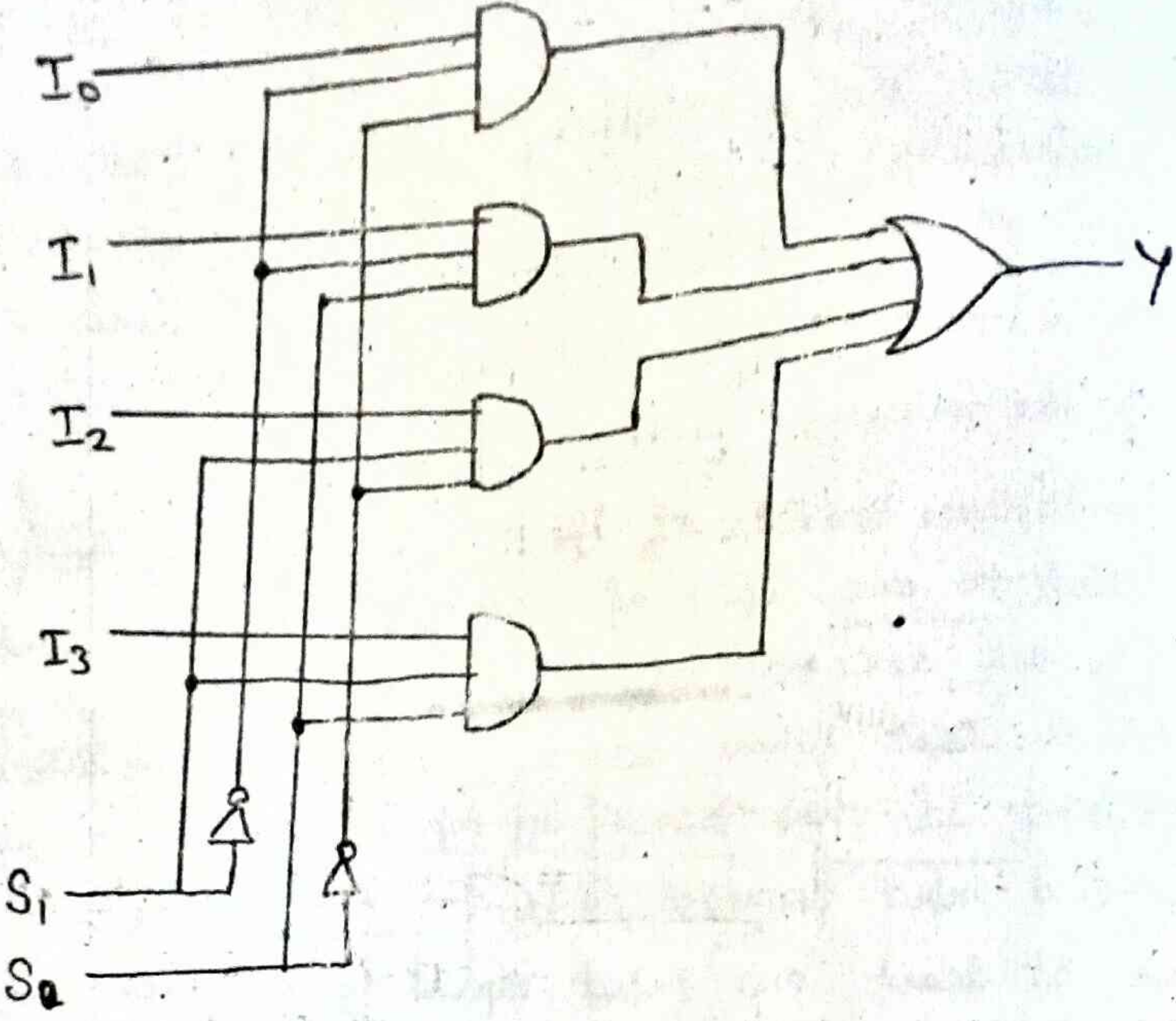


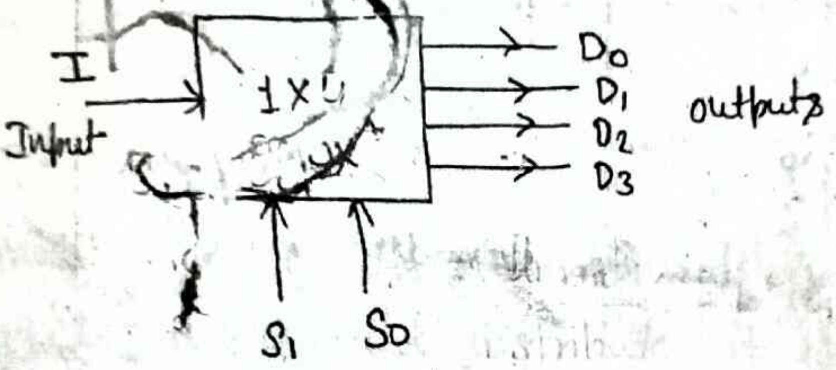
fig:- logic circuit for 4x1 multiplexer

The output Y can be expressed by the function as

$$Y = I_0 S_1' S_0' + I_1 S_1' S_0 + I_2 S_1 S_0' + I_3 S_1 S_0$$

Demultiplexer:

Demultiplexer is the combinational circuit that accepts a single input and distributes it over several outputs. means it works just reverse of the multiplexer. A decorder with an enable input can function as a demultiplexer. If the enable line is taken as a data input line and input lines are taken as the select lines. Consider the following circuit



Here I is taken as the input lines and S_1 & S_0 are taken as the select lines.

Consider the following truth table

Select lines		output lines			
S_1	S_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

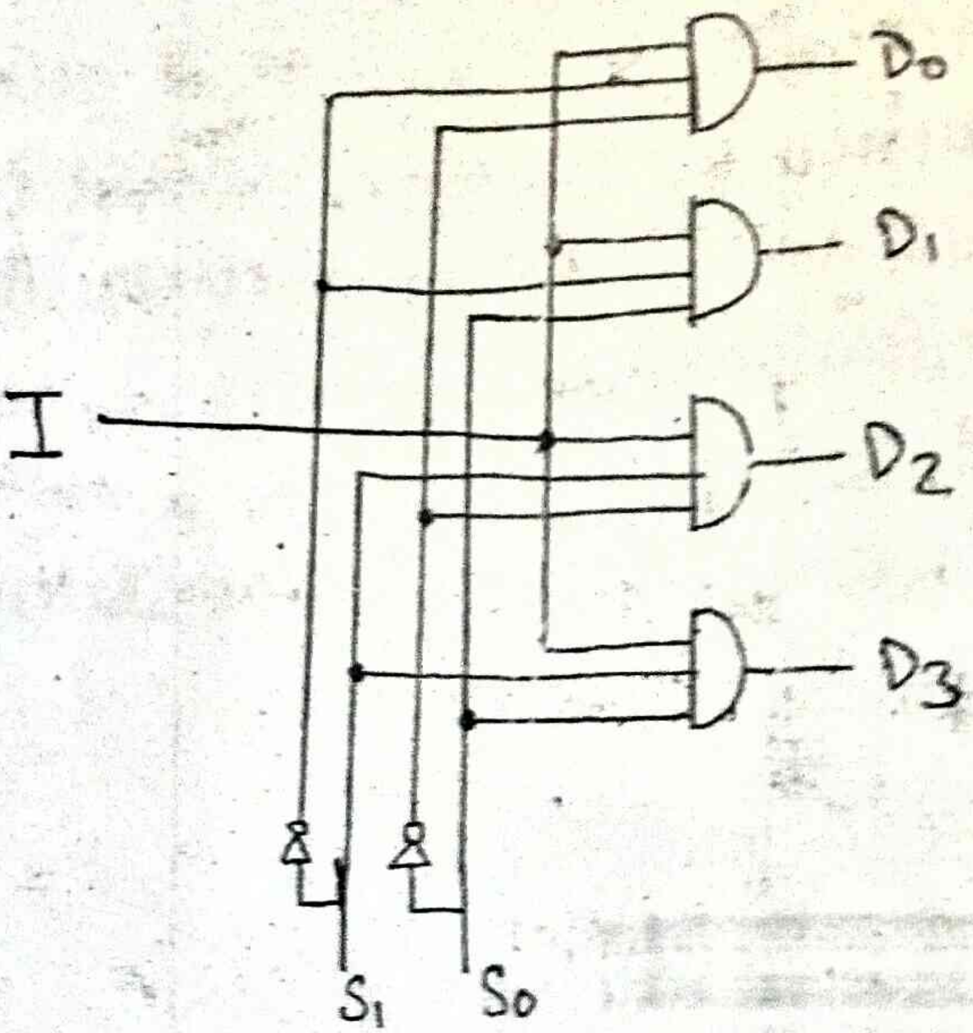


fig:- Logic diagram for demultiplexer.

BCD Adder:- (less imp)

(VI)

A BCD adder is a circuit which is used for adding two BCD digits in parallel and generating the sum digit in BCD. This circuit include the correction logic in its internal construction which is as given below. BCD consists of equivalent 4 bit binary number. since each input digit doesnot exceed 9. And hence the output sum can't be greater than 9+9+1=19 9+9 is for the digits and 1 for the carry. The binary adder produce the sum in binary and produce the result which may range from 0 to 19. These binary numbers can be listed in the table given below and which are labeled by K, Z8, Z4, Z2, Z1 Where K is the carry and the subscripted values represent the weight 8 4 2 1 which can be assigned to the four bits in BCD codes. The first column list the binary sum and the second column list the BCD form sum of decimal numbers. In the table if the sum is equal to 1001 or less than 1001 both binary sum and the BCD sums are equal there is no difficulty for the conversion. Means there is no need of conversion. If the sum is greater than 1001 we get the invalid BCD representation. During such case we perform the addition of binary 0110 to the binary sum that convert the correct BCD representation. Also this produce an output carry as required.

The logic circuit that detects the required correction can be derived from the table given below.

Binary Sum					BCD Sum					Decimal
K	Z8	Z4	Z2	Z1	C	S8	S4	S2	S1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

The correction is needed when the binary sum has an output carry K = 1. The other six combination from 1010 to 1111 that need the correction have 1 in the position Z8 To distinguish them from binary 1000 and 1001 which also have 1 in Z8 position we specify further that by either Z4 or Z2 must have 1. The final condition for a correction is given by boolean function

$$C = K + Z_8 Z_4 + Z_8 Z_2$$

when C=1, it is necessary to add 0110 to the binary sum and give the output carry for the next stage.

The logic circuit of BCD adder for such operation is given as follow-

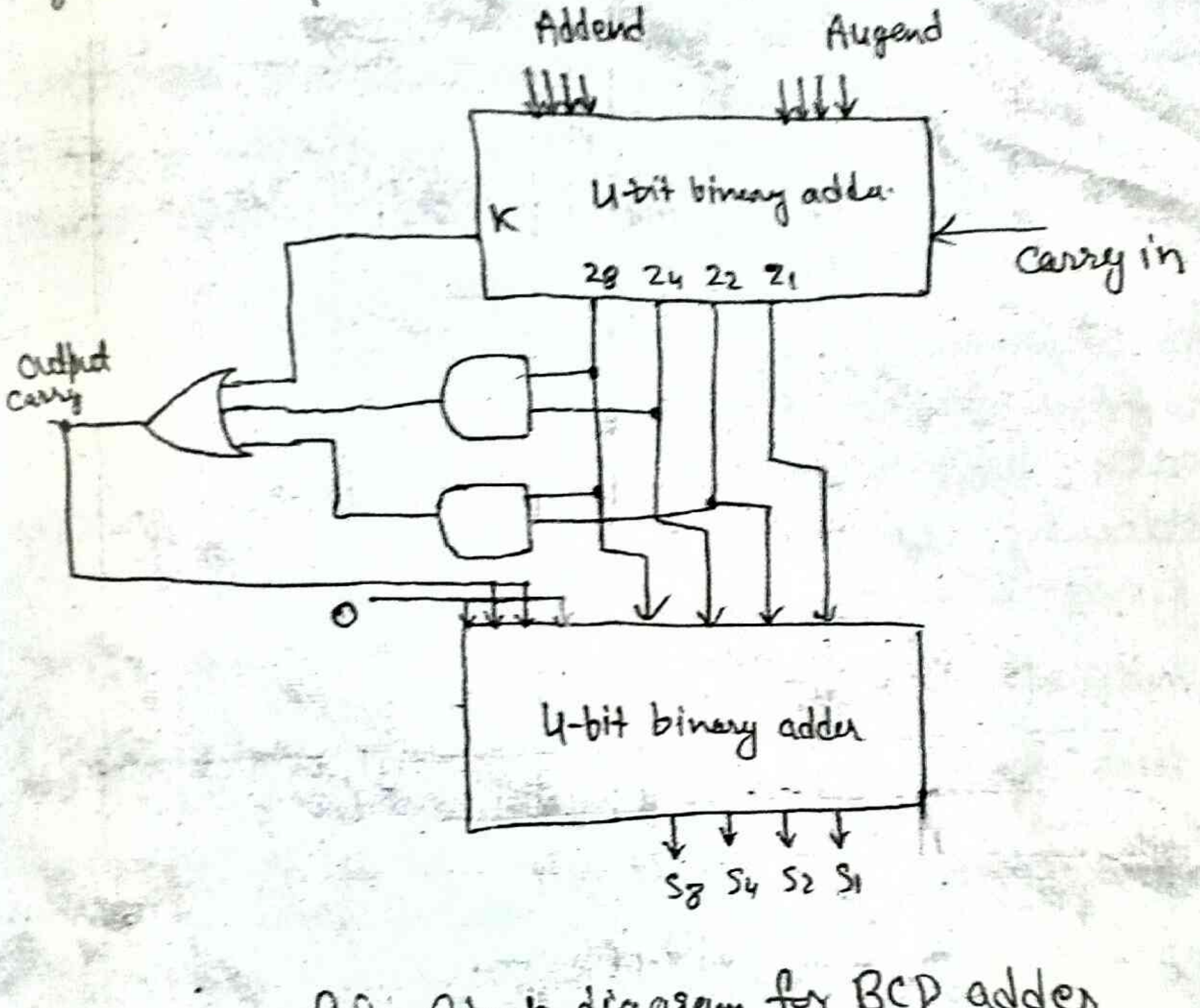


fig:- Block diagram for BCD adder

Magnitude Comparator ;- V.V.I

The magnitude comparator is a combinational circuit that is used for comparing two numbers and find their magnitudes. The outcomes of the comparision is specified by the three binary variables that indicate whether $A>B$, $A=B$ or $A<B$.

An algorithm helps us to find the magnitude of the numbers. Consider two numbers A & B with four digits each. And representing the coefficients of numbers with descending significance as follows.

$$A = A_3A_2A_1A_0$$
$$B = B_3B_2B_1B_0$$

Here each subscripted digit represents one of the digits in the number. The two numbers are equal if all pairs of significant digits are equal ie $A_3=B_3$, $A_2=B_2$, $A_1=B_1$ & $A_0=B_0$. Also when the numbers are binary the digits are either 1 or 0 and the equality relation of each pair of bits can be expressed logically

$$x_i = A_iB_i + A_i'B_i' \quad i=0,1,2,3$$

where $x_i=1$ only if the pair of bits in position i are equal ie if both are 1'S or both are 0'S. Hence for the equality condition to exists all x_i variables must be equal to 1. The binary variable $A=B$ is equal to 1 only if all pairs of digits of the two numbers are equal

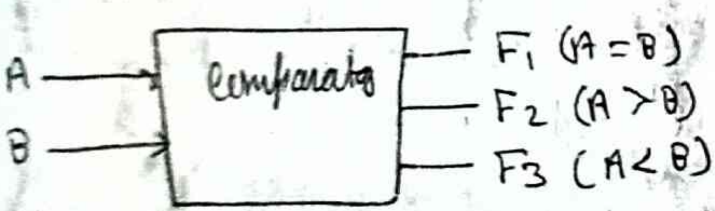
$$(A=B) = x_3x_2x_1x_0$$

For finding whether A is greater than or less than B we inspect the magnitude of pairs of significant digits starting from the most significant position. If two digits are equal, we compare the next lower significant pair of digits. This comparision continues until a pair of unequal digits is reached. If corresponding digit of A is 1 and that of B is 0 we conclude that A>B. If the corresponding digit of A is 0 and that of B is 1 we have A<B. The sequential comparision can be expressed logically by the following boolean function.

$$(A>B) = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$$

$$(A<B) = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$$

The logical circuit of the single bit magnitude comparator can be drawn as



The truth table for this comparator

Inputs		outputs		
A	B	A = B F1	A > B F2	A < B F3
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

The logical circuit for the 4-bit magnitude comparator is as follows -

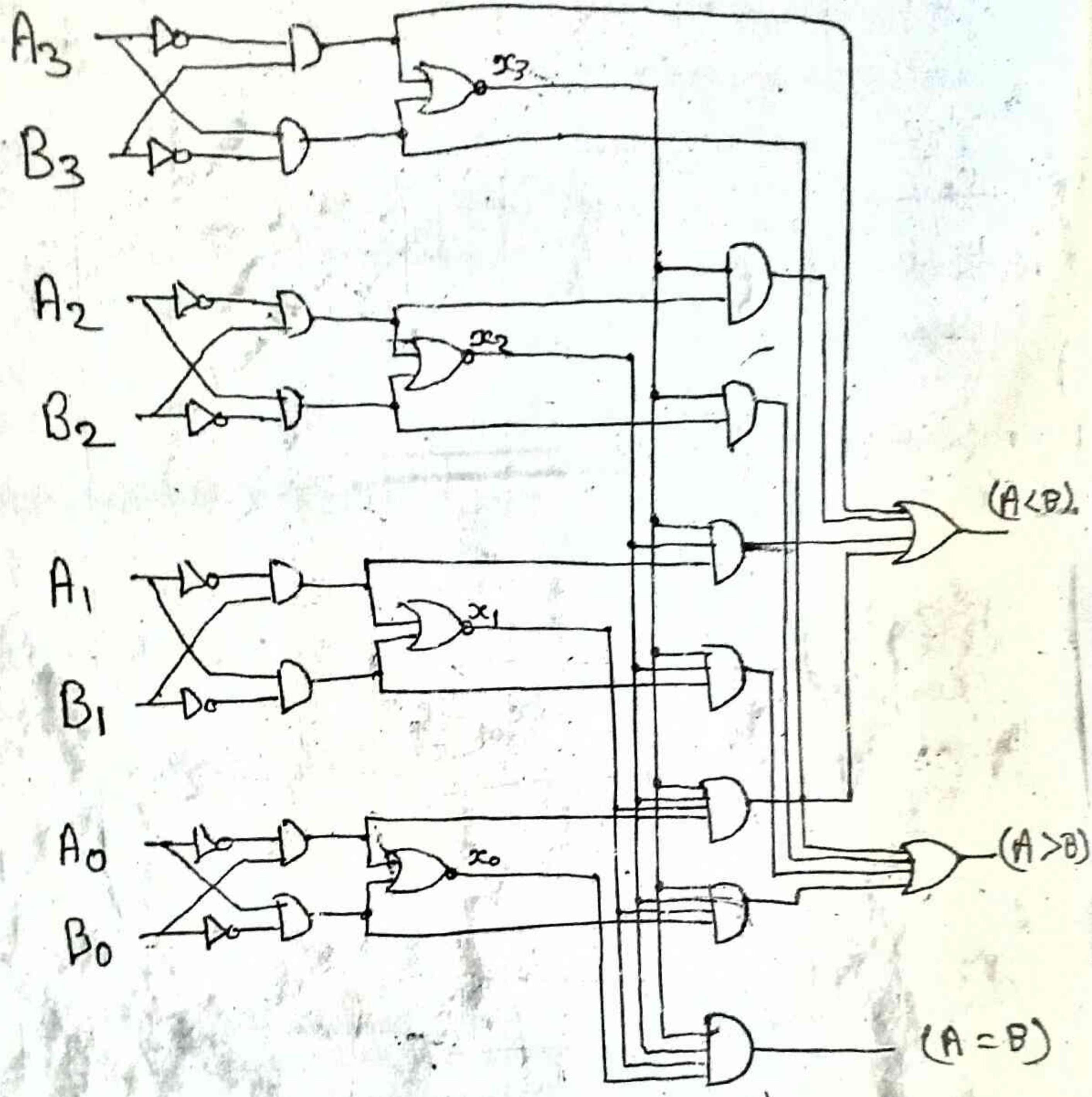


fig:- 4-bit magnitude comparator

Read Only Memory (ROM) Floyd

⑧

ROM is considered as the permenent memory which is used for storing the binary information. Generally the information stored by the Rom is about the information of the manufacturing company and the BIOS. This information must be specified by the desifner of ROM and embedded in the unit to form the required interconnection pattern. ROM comes with special electronic fuses that can be programmed for specific configuration. once the structure is established it stays within the unit even when the power is turned off and on again. This is the reason Rom is permanent memory.

The ROM is a device which consists of decoder and the OR gates within a single IC package. The connection between the outputs of the decoder and the inputs of the OR gates can be specified for each particular configuration. ROM is used to implement complex combinational circuits within one IC package or as permanent storage for binary information. The block diagram for ROM can be given as

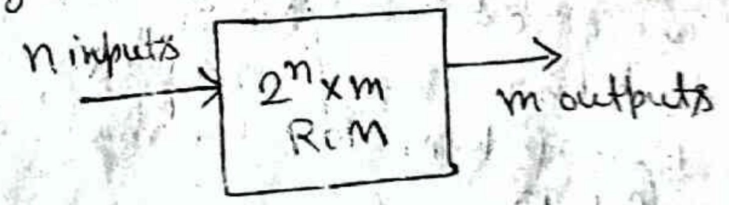


fig:- block diagram of ROM

It consists of n input lines and m output lines each bit combination of the input variables is called an address.

A 32X4 Rom

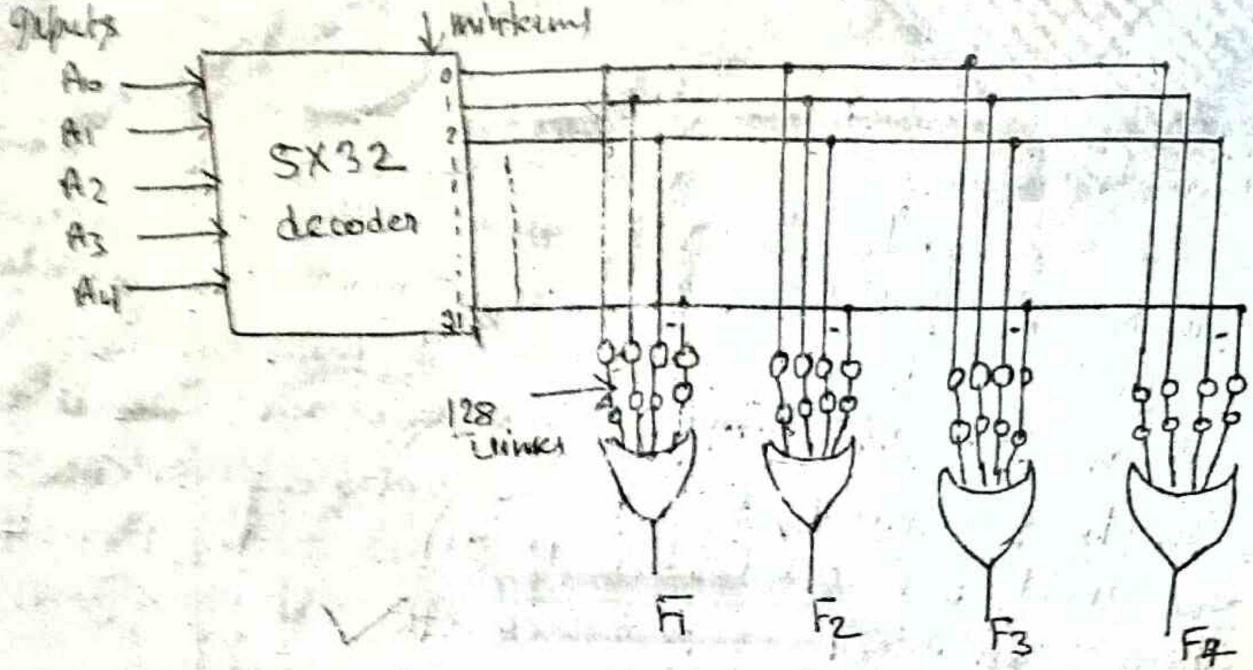


fig:- logic diagram of 32X4 Rom

Here the five inputs are decoded into 32 lines. Each output of the decoder represents one of the minterms of a function of five variables. Each one of the 32 addresses selects one and only one output from the decoder. The 32 outputs of the decoder are connected through fuses to each OR gates. Here only four fuses are shown in the figure.

Combinational logic implementation:-

From the diagram of Rom it is clear that each output provides the sum of all minterms of the n input variables. Any boolean function can be expressed in sum of minterms form. Each output of Rom can be made to represent the boolean function.

→ For n input m output combinational circuit we need $2^n \times m$ Rom

→ the blowing of the fuses is reffered to as programming the Rom

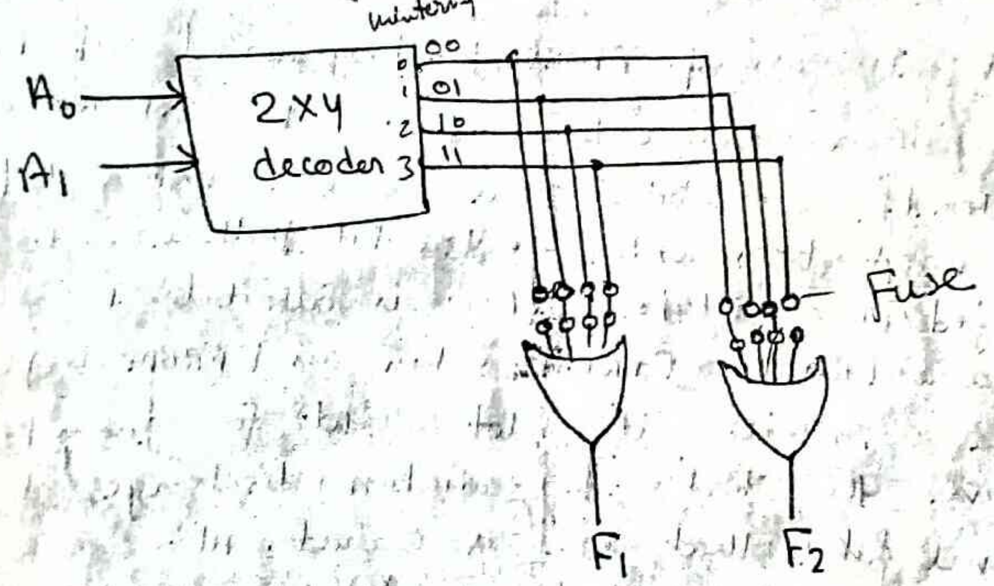
→ The information for the required path in Rom is given by the program table.

consider the following truth table

A_1	A_0	F_1	F_2
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	0

Here are two inputs & two outputs and the boolean function is $F_1(A_1, A_0) = \Sigma(1,2,3)$
$F_2(A_1,A_0) = \Sigma(0,2)$

The logical diagram for this RoM is



This example demonastrate the general procedure for
combinational circuit with a RoM.
implementing any combinational circuit with a RoM. in the combinational
From the number of inputs and outputs in the combinational
From the number of ~~~~ size of Rom required.
circuit, we first determine the size of RoM required.
Then we must obtain the programming truth table of the
RoM. The 0's or 1's in the output functions of
truth table directly specify those fuses that
must be blown to provide the required combinational
Circuit in Sum of minterms form.

Types of ROM

The content of the ROM is fabricated by the ROM designer at the time of manufacturing. Also depending upon the procedure of storing data and the technique of erasing that information the ROMs are classified into different categories-

PROM The programable read only memory (PROM) consists of the technique by which user allows to program the chip in this own laboratory to achieve the desired relationship between input addresses and stored words. PROM programmers are commercially to facilate this procedure. Means these types of PROM chips are programmed once.

EPROM ;- In case of PROM the programmed chip has the fixed pattern which is permanent and can't be erased or altered. Once a bit pattern has been established, the unit must be discarded if the bit pattern is to be changed. A next type of Rom is available & is called erasable PROM. or EPROM. when an EPROM is placed under a special ultraviolet light for given period of time the short wave radiation discharges the internal gates that serve as contact. Also some PROM can be erased with electrical signals instead of ultraviolet light to return ROM to its initial State and can be reprogrammed.

EEPROM / EAPROM These are the PROM which are used again by erasing / altering the content present on the chip by the help of electric method. The Content of the Rom chip is erased by the electric signal instead of ultraviolet signals.

Programmable Logic Array (PLA)

Programmable logic array (PLA) is a LSI component that can be used in economically as an alternative to Rom chip where the number of don't care conditions is excessive. Means when Rom consists of more don't care conditions, then the don't care condition becomes an address input that will never occur. The word at the don't care address need not be programmed and may left in their original state (all 0's or all 1's). The result is that not all the bit patterns available in the Rom are used which may be considered as waste of available equipment.

The block diagram of PLA is given as follows-

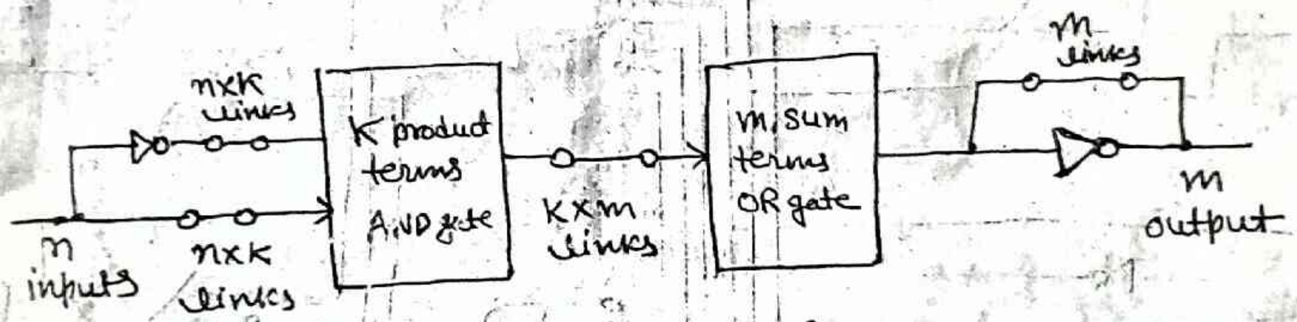


fig:- block diagram of PLA

The diagram shows that PLA consists of n inputs m outputs, K product terms & m sum terms. The product term consists of a group of K AND gates and the sum term constitute a group of m OR gates. Links are inserted between the n inputs and their complement values to each of the AND gates. Also the links are provided between the outputs of the AND gates and inputs of the OR gates.

The PLA program truth table and the boolean function implementation is given as.

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	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

A \ BC	00	01	11	10
0				
1	1	1	1	

$F_1 = AB' + AC$

A \ BC	00	01	11	10
0			1	
1		1	1	

$F_2 = AC + BC$

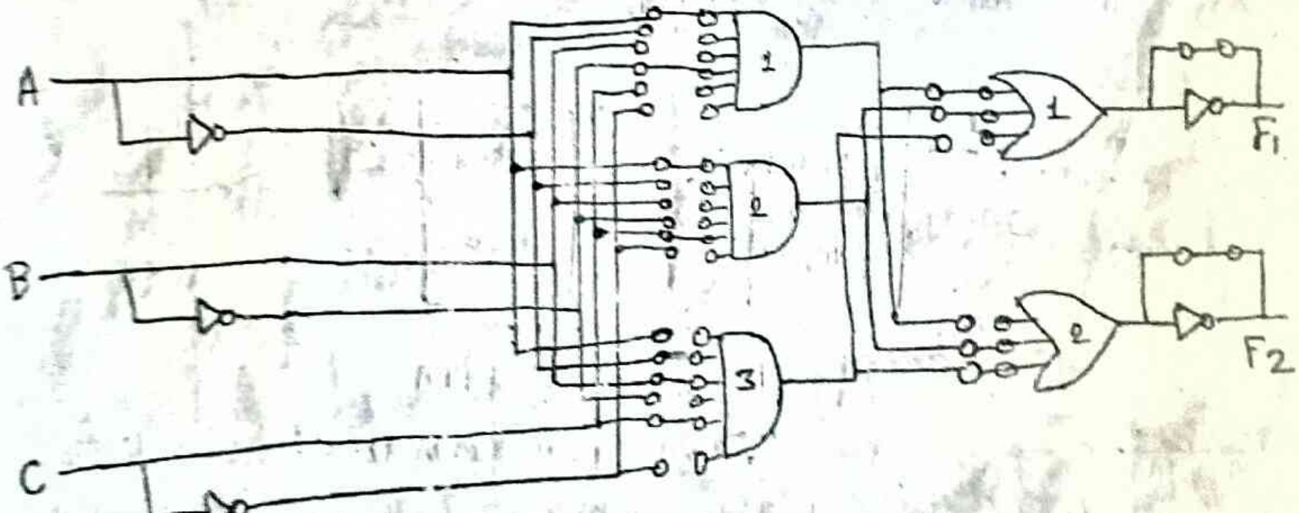


fig:- PLA with 3 inputs, 2 outputs.

In PLA programming we specify paths in its AND-OR-NOT pattern. Typical PLA program consists of three columns in the table as given below-

1st column → list the product terms
2nd " → list the required path betn inputs and AND gates
3rd " → list the required path betn AND gates and OR gates

	Product term	Inputs A	B	C	outputs F1	F2
AB'	1	1	0	—	1	—
AC	2	1	—	1	1	1
BC	3	—	1	1	—	1
					T	T	T/C

Here for each product term, the inputs are marked with 1, 0, or —

If variable in the product term is in its true form the input variable is marked with 1

If variable in the product term is in its complement form the input variable is marked with 0

Ily

If variable in the product term is absent input variable is marked with —

Difference betn Rom & PLA

Rom

- It Generates all the minterms as an output of decoder
- It uses decoder
- Size of Rom is specified by no. of inputs (n) & no. of outputs (m).
- No of programed links = $2^n * m$

PLA

- It does not provide full decoding of the variables
- Decoder is replaced by group of AND Gates.
- Size of PLA is specified by no of inputs (n) no. of product terms (k) number of outputs (m) & number of sum terms
- No. of programed links = $2n * K + K * m + m$

24 $2n * K + K * m + m$

Finally a T (true) output indicates that
the link across the output inverter remains
in the place and C (complement) specifies that
the corresponding link be broken.