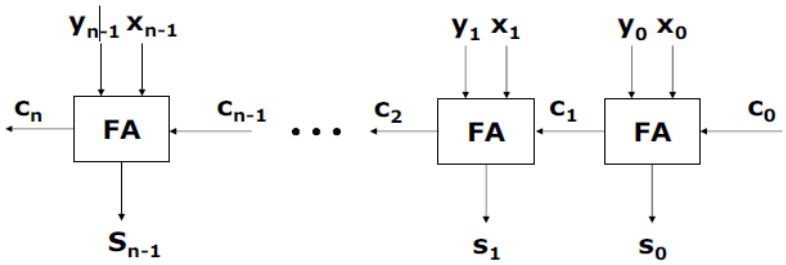
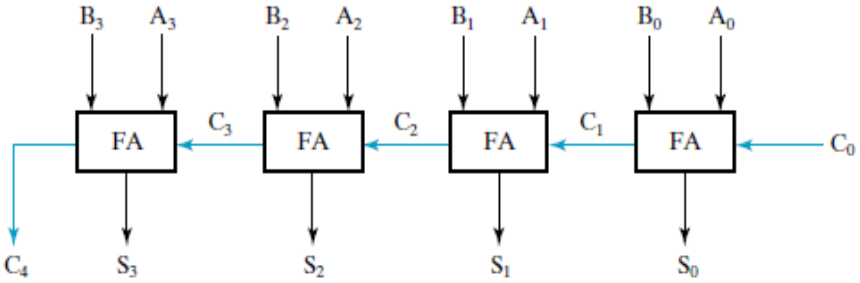
Combinational Logic Circuit

Binary Parallel Adder

Two binary numbers of n bits can be added by using binary parallel adder. A binary parallel adder is a digital function that produces the arithmetic sum of to binary numbers in parallel. It consists of full-adders connected in cascade, with the output carry from one full-adder connect to the input carry of the next full-adder.

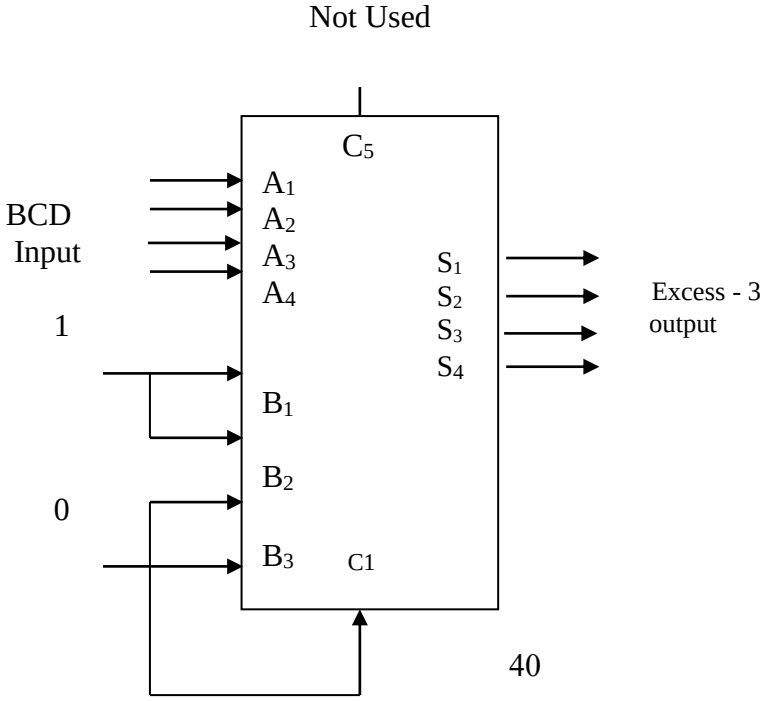


Q Design a binary parallel adder to add two 4-bit binary numbers?

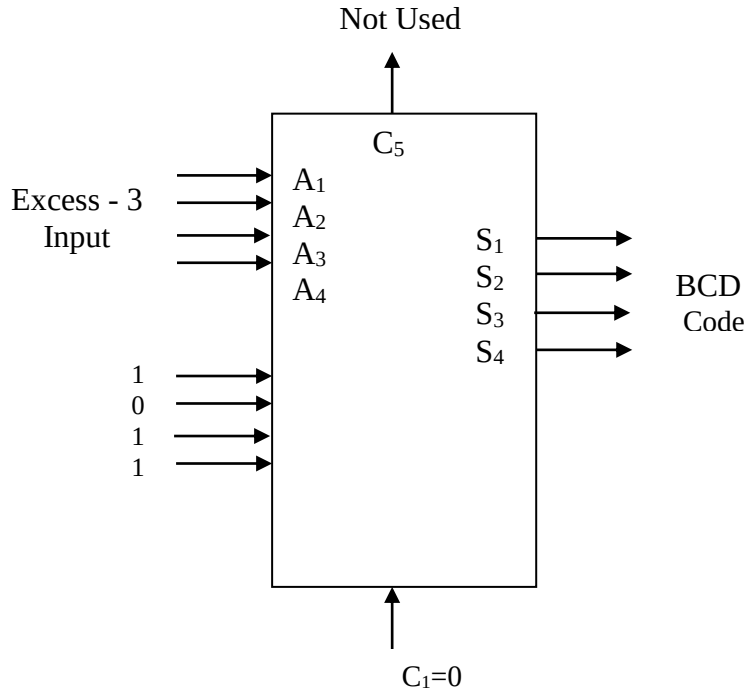


Ex Design a BCD to Excess 3 code converter?

Sol



Ex Design Excess-3 to BCD code converter?



BCD adder

It is a combinational circuit that adds two BCD digits in parallel and produces a sum digit also in BCD. Consider the arithmetic addition of two decimal digits in BCD, together with a possible carry from previous stage. Since each input digit does not exceed 9, the output sum cannot be greater than 9 + 9 + 1 =19, where the 1 in the sum being an input carry.

The BCD adder work as

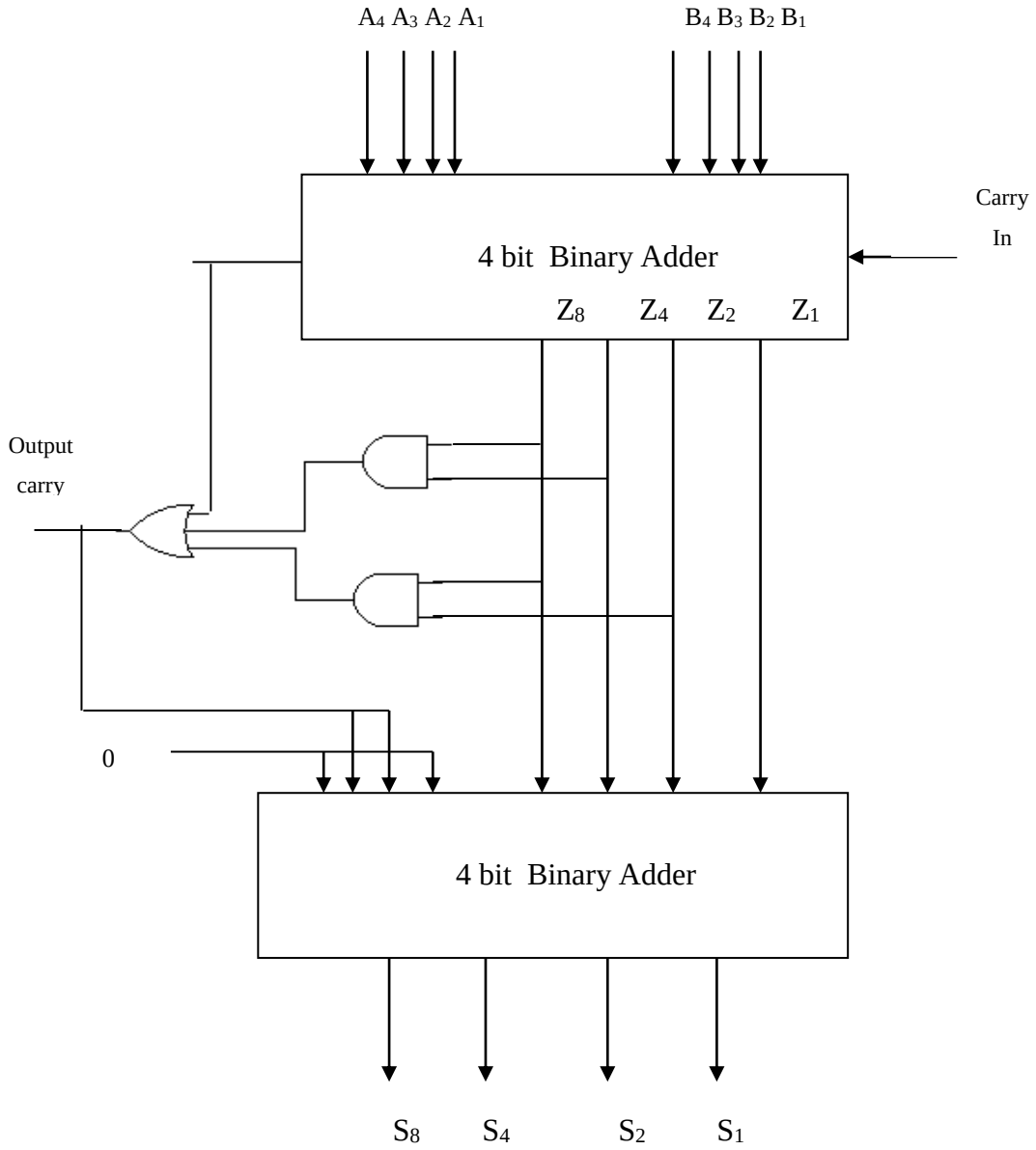
- Suppose two BCD digits are applied to 4 – bit binary adder.
- The adder will form the sum in binary and produce a result in the range 0 to 19.
- The binary sum is listed in the first column in the table, where Z_1, Z_2, Z_4, Z_8 represent the weights 1 , 2 , 4, 8 and K is the carry.
- The output sum of two decimal digits must represent in BCD and listed in the second column in the table.
- In second column (BCD sum) , where the sum is equal to or less than 1001 , the BCD number is identical, and no conversion is needed.
- When the sum is greater than 1001. The BCD adder must include the correction logic in its internal construction. The addition of binary 6 (0110) to binary sum convert it to correct the BCD representation and also produces an output carry.

- From the first column (Binary Sum), the correction is needed when the binary sum has an output carry K=1, or when 1 in positions Z_8 and Z_4, or when 1 in positions Z_8 and Z_2.
- The equation is

$$\mathbf{C = K + Z_8 Z_4 + Z_8 Z_2}$$

When the value of C = 1 the circuit will add binary 6 (0110) for correction.

Binary Sum					**BCD Sum**					**Decimal**
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
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



Block – diagram of a BCD adder

The Decoder

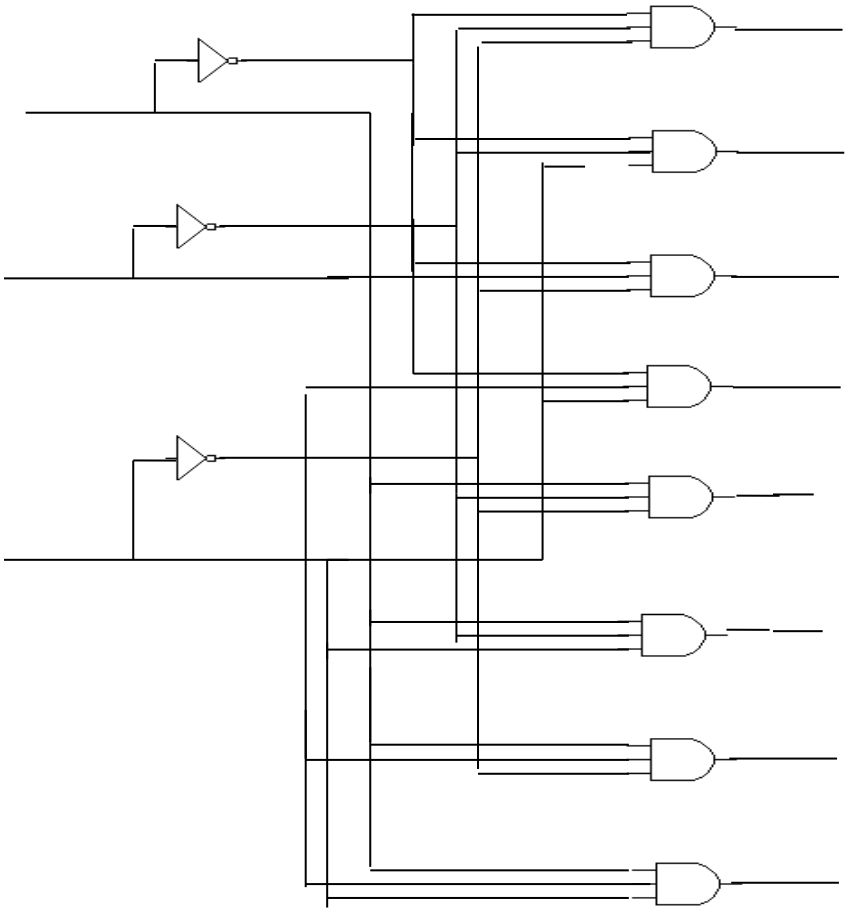
A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n output lines, each output representing one of the minterms of input variables.

Ex Design 3 to 8 decoder?

Sol

The truth table is

X	Y	Z	D 0	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



3 to 8 line decoder

Ex Design BCD to Decimal decoder?

Sol:

The size of the decoder is 4 x 16 and the truth table is

X	Y	Z	W	D_0	D1	D2	D3	D4	D5	D6	D7	D8	D9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	1										
1	1	0	0	Don't Care									
1	1	0	1										
1	1	1	0										

XY \ ZW	0 0	01	11	10
00	D_0	D_1	D_3	D_2
01	D_4	D_5	D_7	D_6
11	X	X	X	X
10	D_8	D_9	X	X

$D_0 = X Y Z W$ $D_1 = X Y Z W$

$D_2 = Y Z W$ $D_3 = Y Z W$

$D_4 = Y Z W$ $D_5 = Y Z W$

$D_6 = Y Z W$ $D_7 = X Z W$

$D_8 = X W$ $D_9 = Y Z W$

Ex Implement the Full- Adder circuit with a Decoder and OR gates?

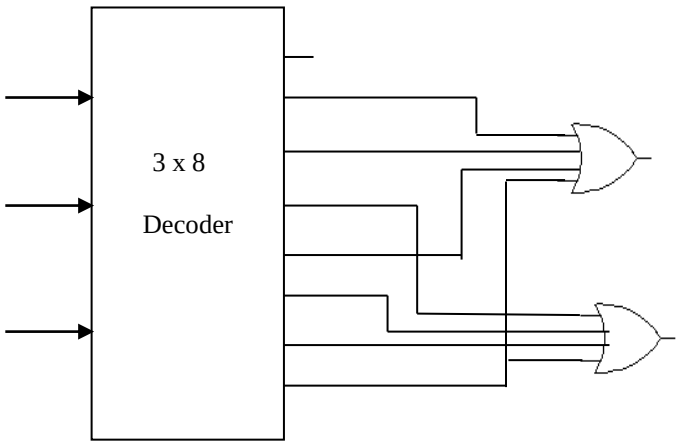
Sol

From the truth table of a Full-Adder the functions of this combinational circuit in Sum of Minterms

1 – S (X, Y ,Z) = Σ (1, 2, 4, 7)

2 – C (X, Y, Z) = Σ (3, 5, 6, 7)

Therefore the number of the inputs in the decoder is 3 (X, Y, Z) and the output is 8, so the size of the decoder (3 x 8)



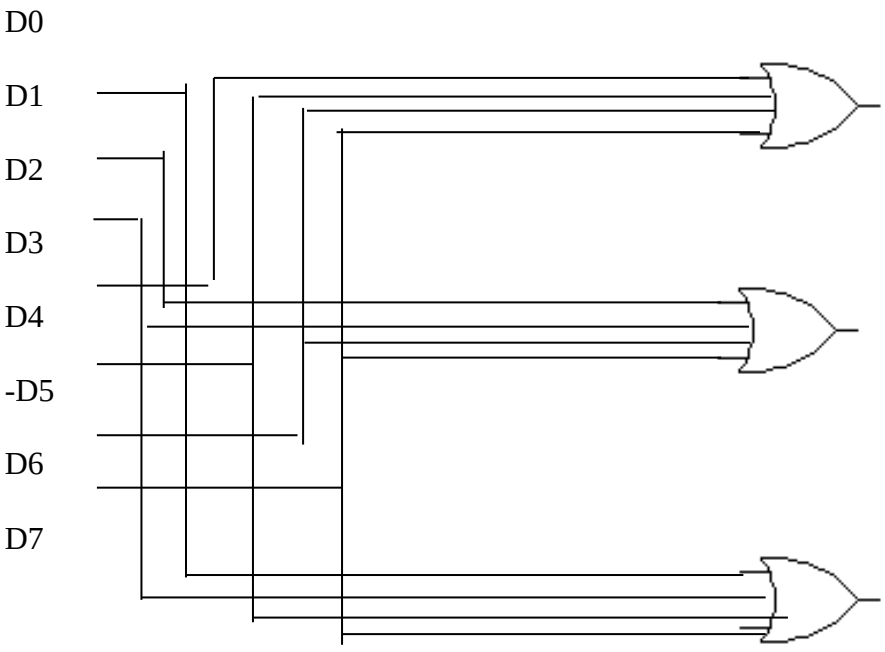
The Encoder

An encoder is a digital function produces a reverse operation from that of a decoder. An encoder has 2^n inputs lines and n output lines generate the binary code for the 2^n input variables.

Ex Design the Octal to Binary Encoder

Sol The truth table is

D_0	D1	D2	D3	D4	D5	D6	D7	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

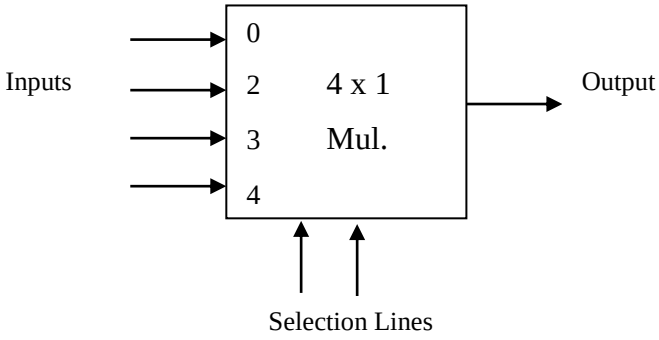


Octal to Binary Encoder

Multiplexer

Multiplexer means transmition a large number of information units over a small number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information one of many inputs lines, there are 2^n inputs lines and n selection lines.

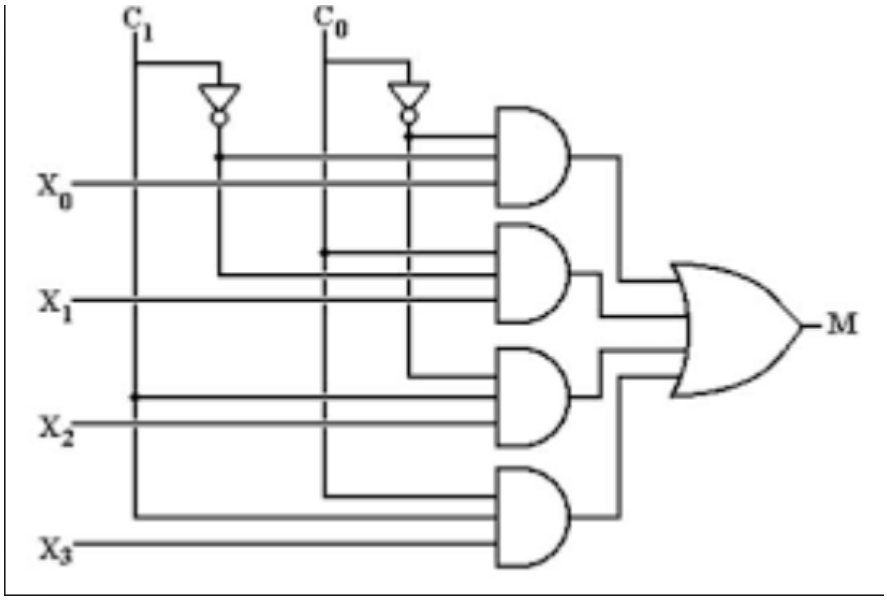
Ex 4 x 1 multiplexer , the input lines is 4 (2n) and the output is one line and the selection lines is 2.



Block Diagram

C_1	C_0	Y
0	0	X_0
0	1	X_1
1	0	X_2
1	1	X_3

Function Table



4 x 1 multiplexer logic circuit

Boolean Function Implementation

Boolean function can implemented with a multiplexer by following these steps:

1- the variables in the function applied as n+1

2- n is the number of selection lines.

3- The reminder variables taken as input variable

4- The minterms is applied in a table

5- The inputs applied as 0 and 1 as primed and not primed

Ex Implement the following function with a multiplexer.

$$F(A, B, C) = \Sigma (1, 3, 5, 6)$$

Sol

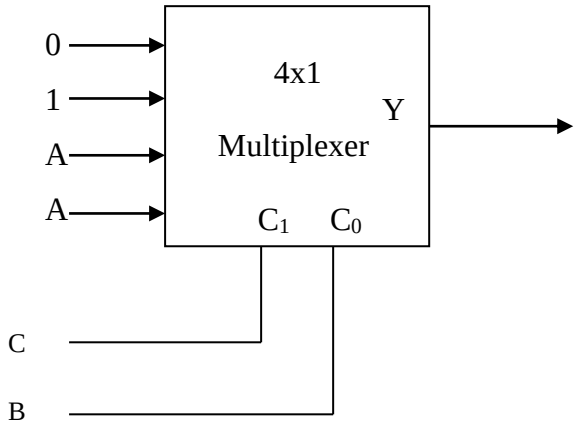
1– The variables in the function are 3 so n+ 1 = 3

2 – The selection lines are n = 2 (B,C)

3– The tired variable A is applied as input to the multiplexer.

	I_0	I_1	I_2	I_3
A	0	(1)	2	(3)
A	4	(5)	(6)	7

4 – The minterms is applied on the table and the block diagram of the multiplexer is drown



H.W.

1- Implement the Boolean function $F = \Sigma (0, 1, 3, 4, 8, 9, 15)$

2 – Implement the following functions using a decoder and OR gates?

1 – $F1 (A, B) = A + B$

2 – $F2 (A, B) = \Sigma (0, 1)$

3 – $F 3 (A, B) = \Pi (1, 3)$

Read Only Memory (ROM)

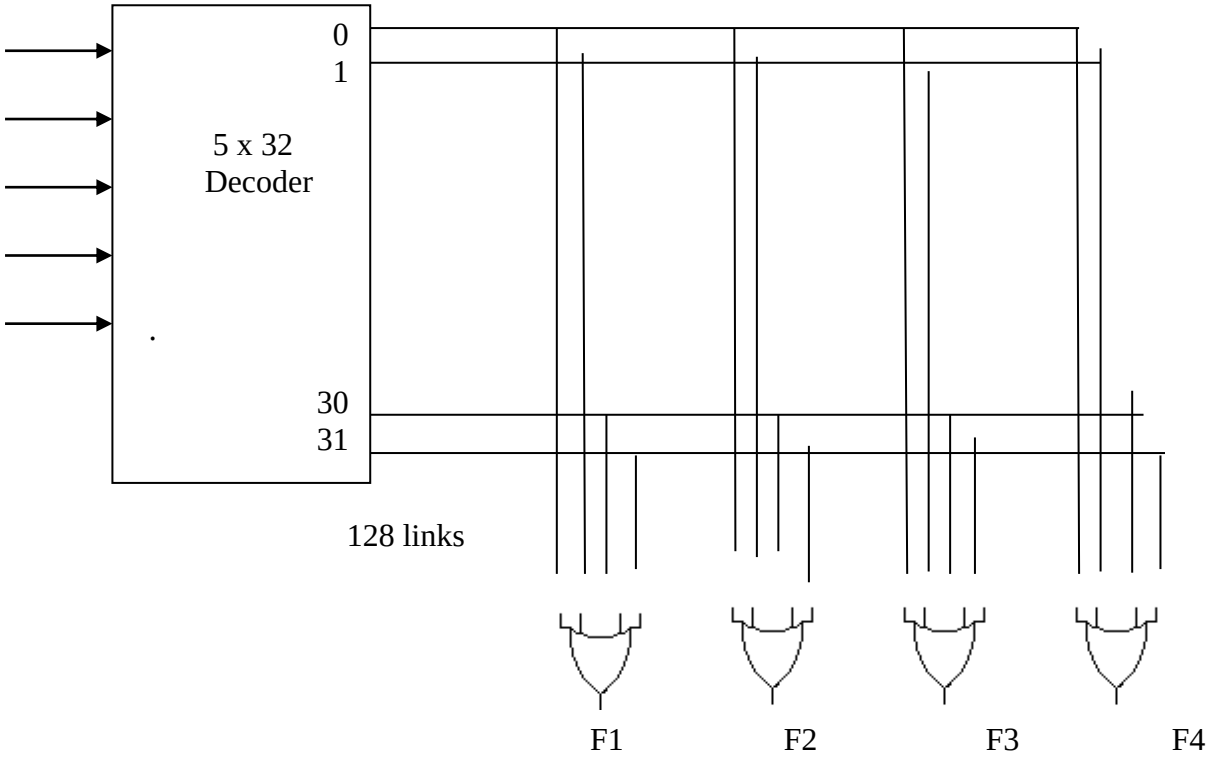
A read only memory (ROM) is a device that includes both the decoder and OR gates with a single IC package. The ROM is a combinational circuit constructed with a decoder and number of OR gates equal to the number of outputs in the unit. ROM consists of 2^n x m, where n is the number of inputs lines and m is the number of outputs lines

Ex consider 32 x 4 ROM

Sol

4 = m = outputs lines and $32 = 2^n = 2^5 \longrightarrow$ n = 5 = inputs lines

minterm



Logic Construction of 32 x 4 ROM

Ex Implement the functions using ROM

F1 (A1 , A0) = Σ (1, 2, 3)

F2 (A1 , A0) = Σ (0, 2)

Ex Design a combinational circuit using ROM. The circuit accepts a 3-bit number and generates an output binary number equal to the square of the input numbers.

Sol

The truth table is

Input			Output					
A2	A1	A0	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

In the output D_0 is equal to A_0 and D1 is equal zero therefore the truth table can derived as follow

Input			Output			
A2	A1	A0	D_5	D_4	D_3	D_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

And the block diagram of this ROM