

UNIT II- COMBINATIONAL CIRCUITS

PART A

1. Write the design procedure of combinational circuits.

The procedure involves the following steps,

- i. The problem is stated
- ii. The number of available input variables and required output variables is determined.
- iii. The input and output variables are assigned letter symbols.
- iv. The truth table that defines the required relationships between inputs and outputs is derived.
- v. The simplified Boolean function for each output is obtained.
- vi. The logic diagram is drawn.

2. Compare serial adder and parallel adder.

S.No	Parallel adder	Serial adder
1	The Parallel adder uses registers with parallel load.	Serial adder uses shift registers.
2	The number of full-adder circuits in the parallel adder is equal to the number of bits.	Serial adder requires only full-adder circuits and a carry flip-flop.
3	Parallel adder is combinational circuit.	Serial adder is a sequential circuit.
4	Parallel adder is faster than serial adder.	Serial adder is slower than parallel adder.

3. Define Half adder & Full adder.

- The logic circuit that performs the addition of two bits is a half adder.
- The logic circuit that performs the addition of three bits is a full adder.

4. Give the logic expressions for sum and carry in Full adder circuit.

$$S = x \oplus y \oplus z$$

$$C = xy + yz + zx$$

5. What is priority encoder?

It is an encoder circuit that includes the priority function. The function of the priority encoder is such that if two or more inputs are equal to 1, the input having the highest priority will take precedence.

6. Give the logic expressions for sum and carry in Full adder circuit.

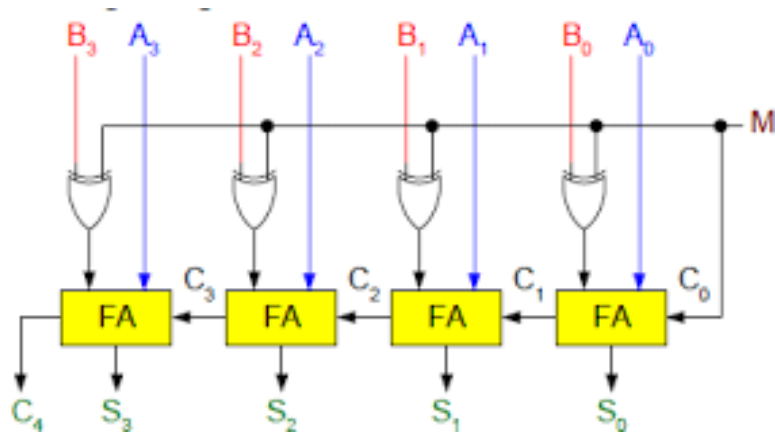
$$S = x \oplus y \oplus z$$

$$C = xy + yz + zx$$

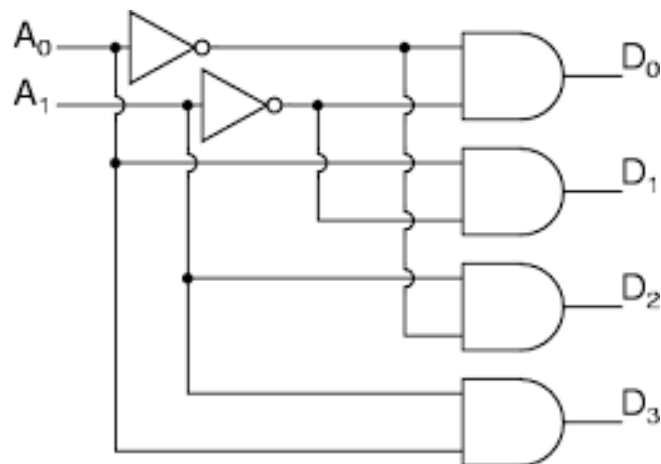
7. Give any four examples for combinational circuits.

- i. Half adder, Full adder
- ii. Magnitude Comparator
- iii. Decoder and Encoder

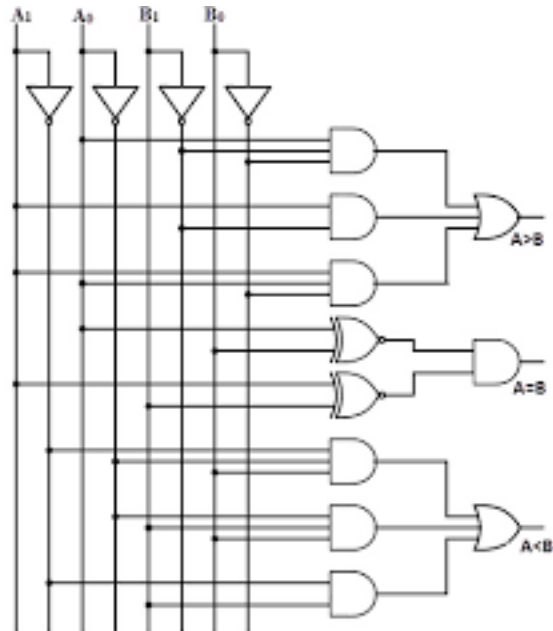
8. Construct 4-bit parallel adder/subtractor using Full adder & X-OR gates.



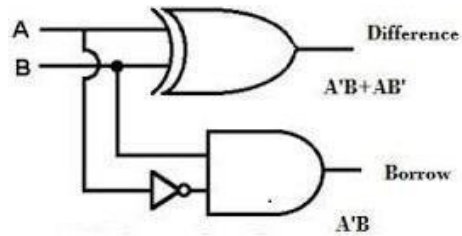
9. Convert a 2 to 4 line decoder with enable input to 1x4 Demultiplexer.



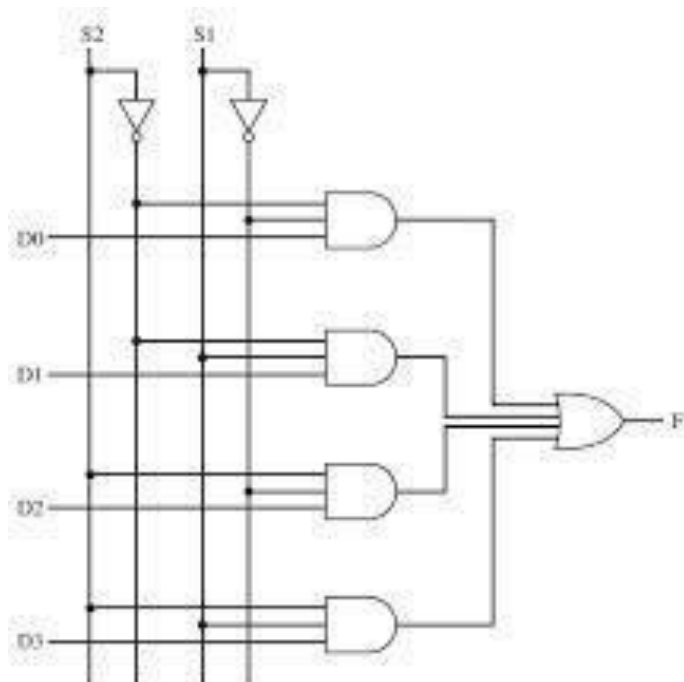
10. Draw the logic circuit of 2-bit comparator.



11. Design a Half Subtractor using basic gates.



12. Draw the logic diagram of 4 line to 1 line multiplexer.



13. List out various application of multiplexer.

Multiplexers are used in data selection, data routing, operations sequencing, parallel to serial conversion, waveform generation and logic function generation.

14. Write the difference between DEMUX and DECODER.

S.NO	DEMUX	DECODER
1	It is a digital circuit that receives information on a single line and transmits the information on one of the several output lines. It consists of a one input line, 'n' selection lines and '2 ⁿ ' output lines.	A decoder is a combinational circuit that converts n number of input lines to a maximum of 2 ⁿ number of output lines.
2	It contains 'n' selection lines.	Selection lines are not present.

15. Write the expression for borrow and difference in Full Subtractor circuit.

$$\text{Borrow } B = \bar{x}y + \bar{x}z + yz$$

$$\text{Difference } D = x \oplus y \oplus z$$

16. Write about the design procedure for combinational circuits (NOV/DEC 2013)

1. From the given specification determine the number of inputs and outputs and assign a symbol to each input and output.
2. Derive the truth table that defines the required relationship between inputs and outputs.
3. Obtain the Boolean function for each output as a function of the input variables using Karnaugh Map
4. Draw the logic diagram based on the Boolean function obtained in step 3.

PART B

1. Explain about Carry look ahead adder

Most other arithmetic operations, e.g. multiplication and division are implemented using several add/subtract steps. Thus, improving the speed of addition will improve the speed of all other arithmetic operations.

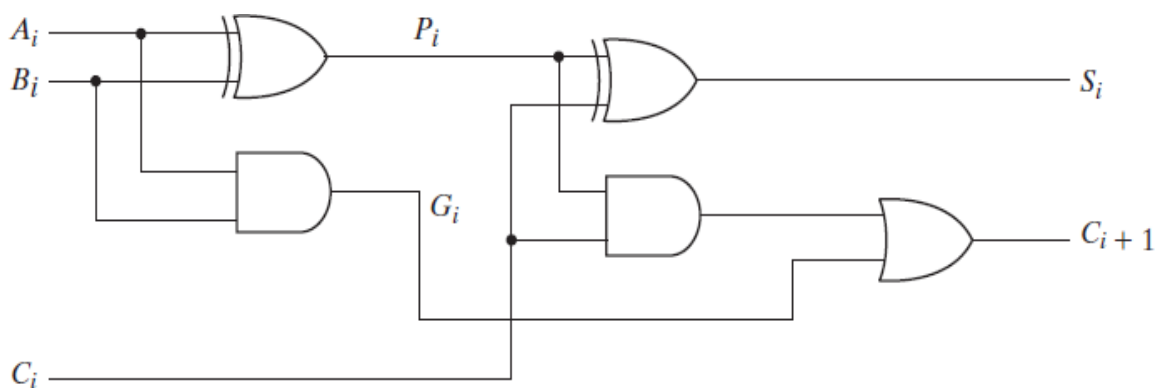
Accordingly, reducing the carry propagation delay of adders is of great importance. Different logic design approaches have been employed to overcome the carry propagation problem.

One widely used approach employs the principle of carry look-ahead solves this problem by calculating the carry signals in advance, based on the input signals. This type of adder circuit is called as **carry look-ahead adder** (CLA adder). It is based on the fact that a carry signal will be generated in two cases:

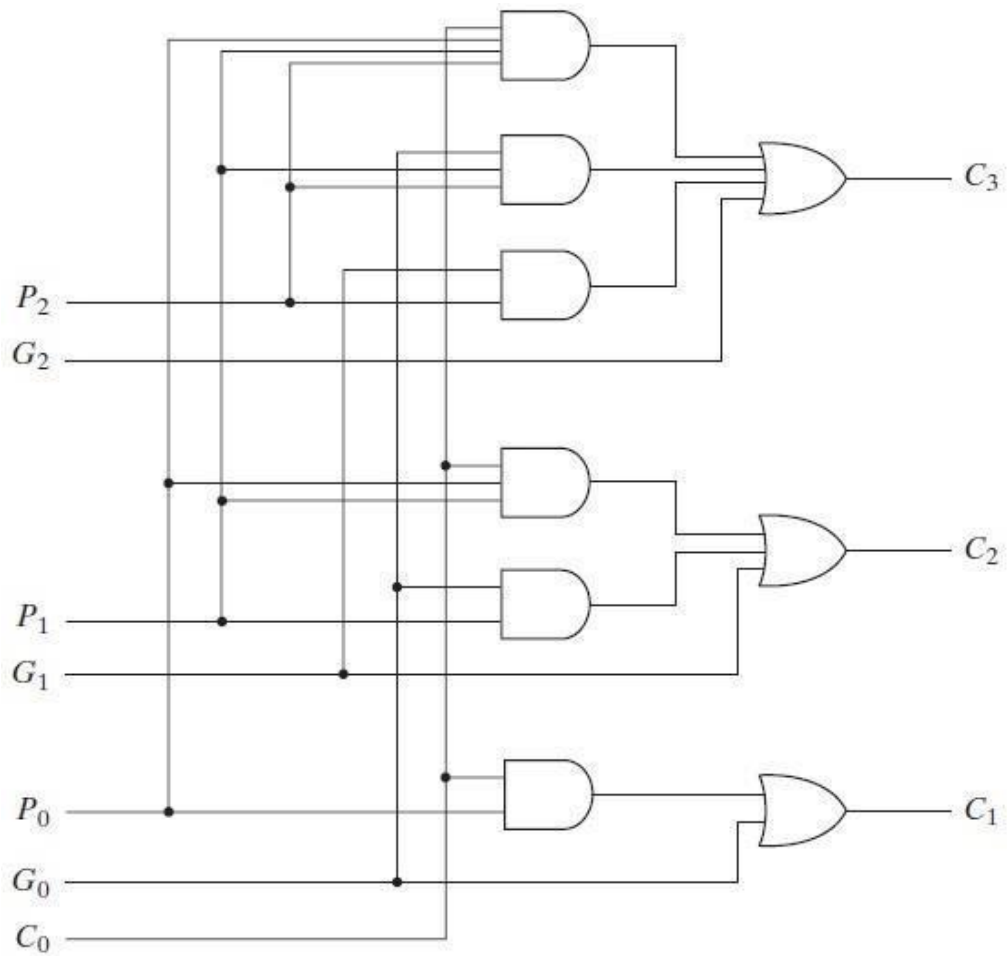
- (1) when both bits A_i and B_i are 1, or
- (2) when one of the two bits is 1 and the carry-in (carry of the previous stage) is 1

Carry look ahead adder is the most widely used technique for reducing the propagation time in parallel adder. The solution for reducing carry propagation delay time is to employ faster gates with reduced delays.

Full adder Circuit:



Logic Diagram of a look ahead carry generator



The output sum and carry can be defined as :

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

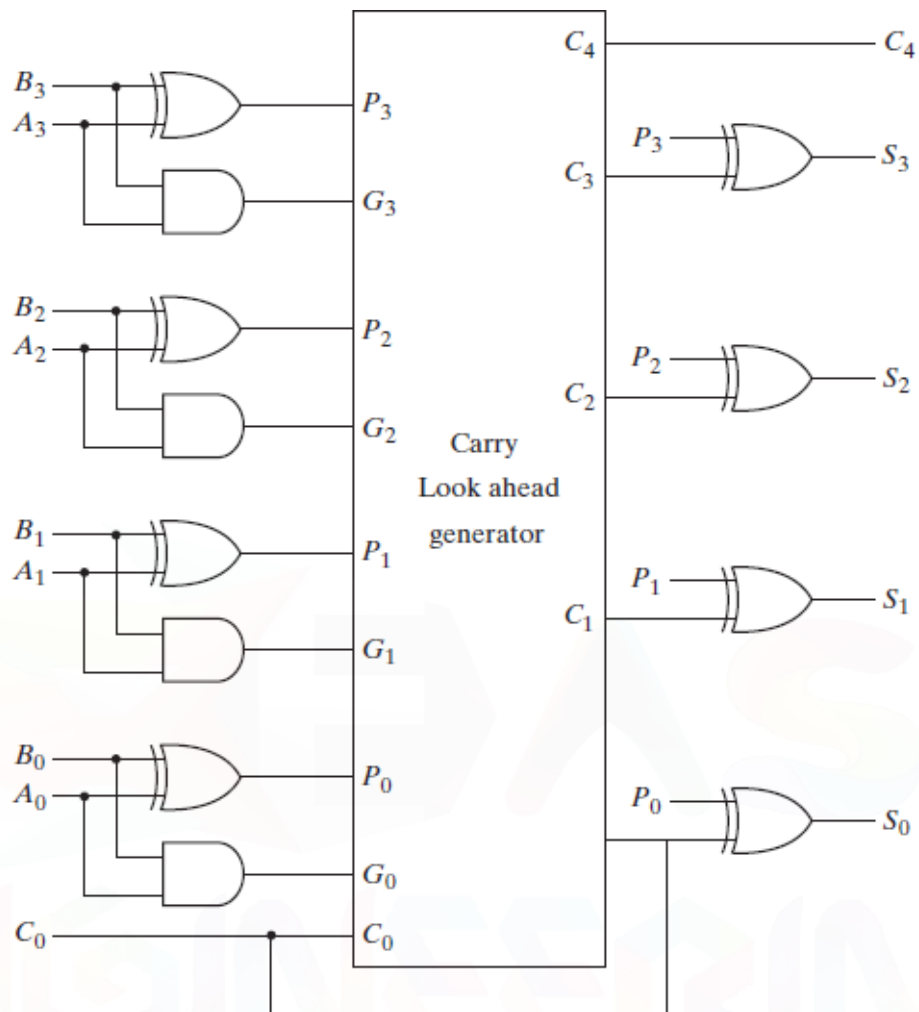
$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0)$$

$$= G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

4 bit full adders with look ahead carry



2. Design a 4 bit Magnitude Comparator and draw the circuit.

Definition:

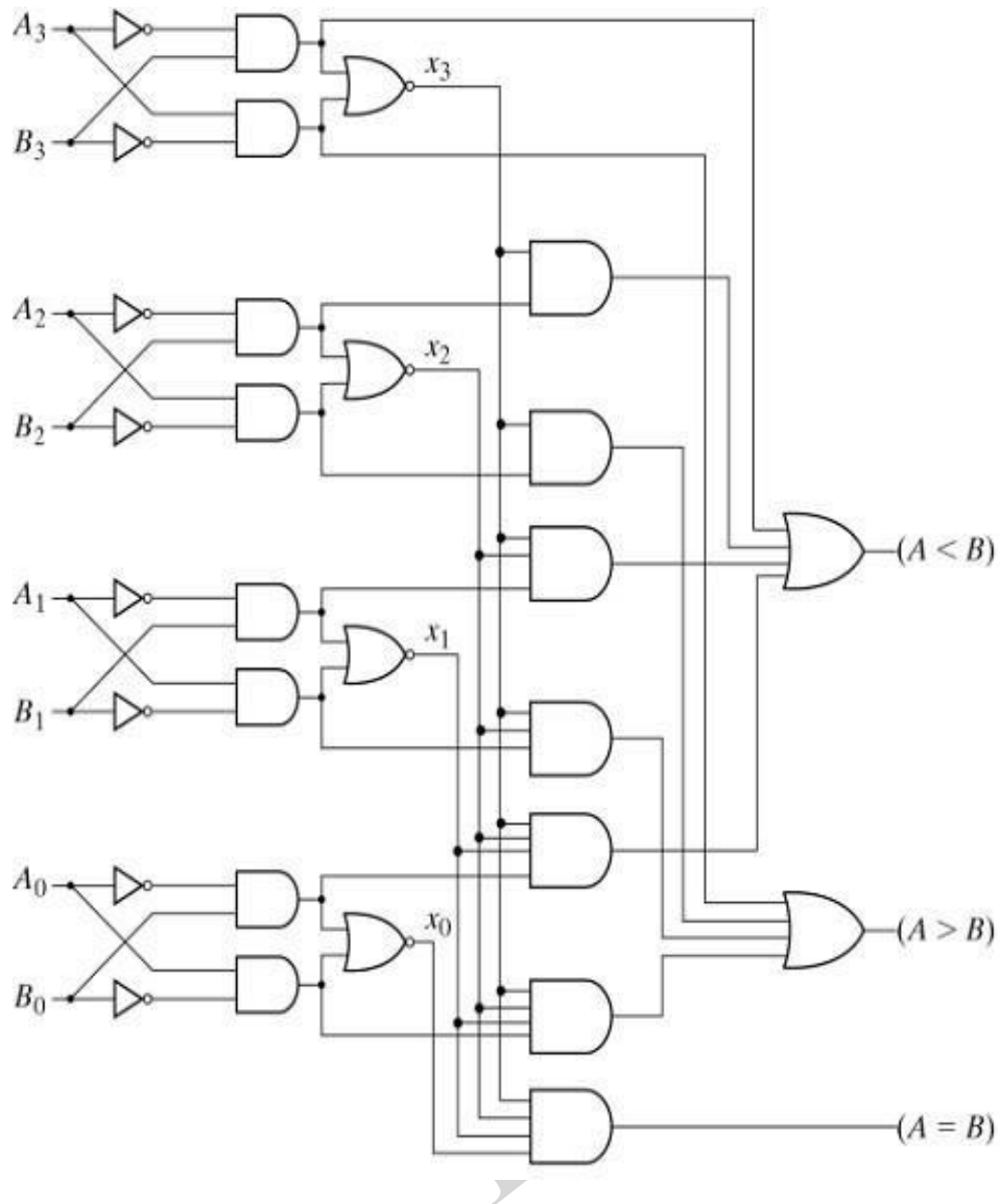
Magnitude comparator is a combinational circuit that compares TWO numbers, A and B, and then determines their relative magnitudes.

$$A > B$$

$$A = B$$

$$A < B$$

Logic Diagram



Logic Expression

$$(A=B) = x_3x_2x_1x_0$$

$$(A>B) = A_3B'_3 + x_3A_2B'_2 + x_3x_2A_1B'_1 + x_3x_2x_1A_0B'_0$$

$$(A<B) = A'_3B_3 + x_3A'_2B_2 + x_3x_2A'_1B_1 + x_3x_2x_1A'_0B_0$$

Explanation

The comparison of two numbers is an operation that determines if one number is greater than, less than, or equal to the other number.

The circuit for comparing two n-bit numbers has 2^n entries in the truth table.

$$A = A_3A_2A_1A_0$$

$$B = B_3B_2B_1B_0$$

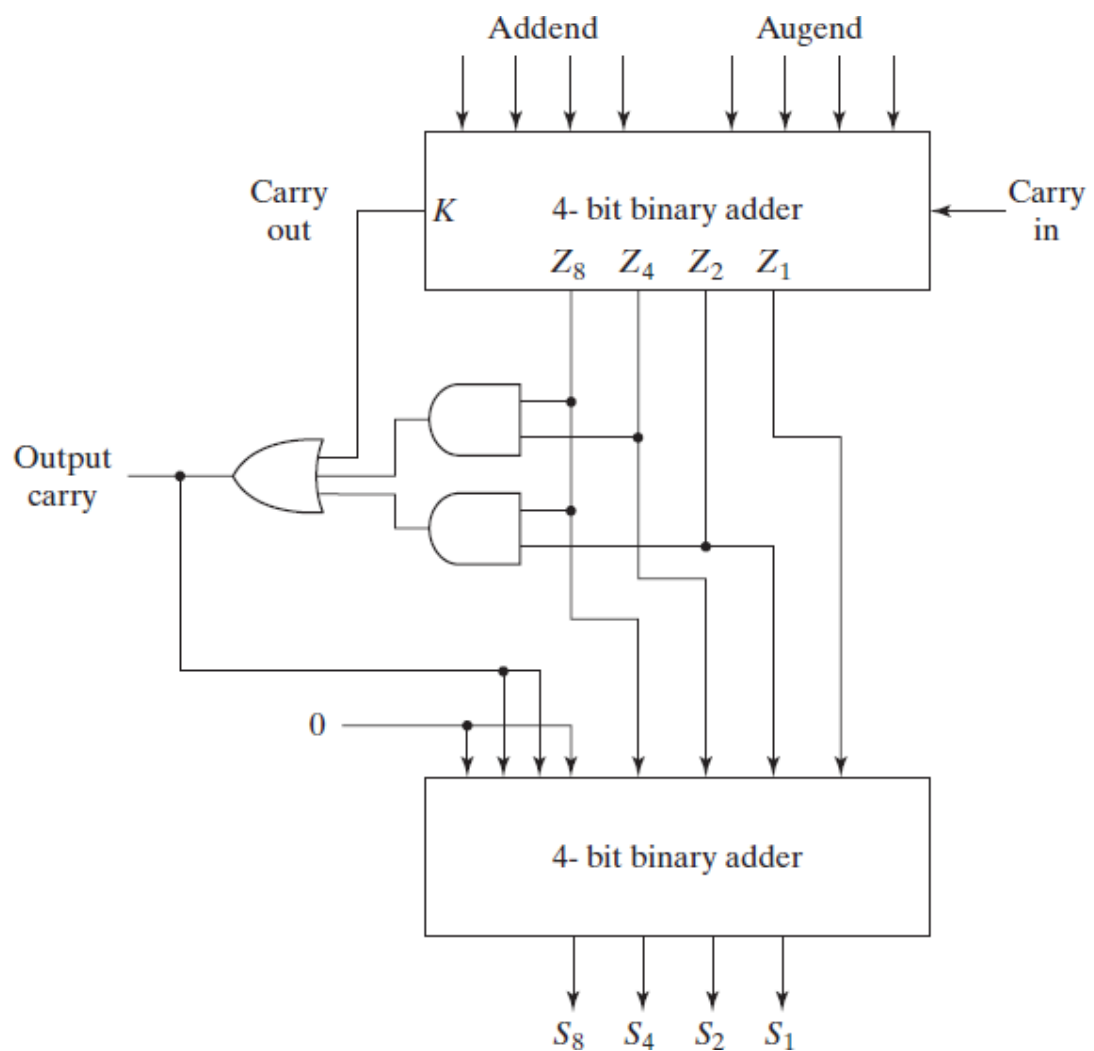
$$x_i = A_iB_i + A_i'B_i'$$

3. Design a BCD Adder and explain its working with necessary circuit diagram.

Definition

- BCD adder that adds two BCD digits and produces sum digit in BCD.
- In BCD each number is defined by a binary code of 4 bits.

Block Diagram



Expression:

$$C = K + Z_8Z_4 + Z_8Z_2$$

Truth Table:

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	0	0	2
0	0	0	1	1	0	0	0	0	1	3
0	0	1	0	0	0	0	0	1	0	4
0	0	1	0	1	0	0	0	1	1	5
0	0	1	1	0	0	0	0	1	0	6
0	0	1	1	1	0	0	0	1	1	7
0	1	0	0	0	0	1	1	0	0	8
0	1	0	0	1	0	1	1	0	1	9
0	1	0	1	0	1	1	0	0	0	10
0	1	0	1	1	1	1	0	0	1	11
0	1	1	0	0	1	1	0	1	0	12
0	1	1	0	1	1	1	0	1	1	13
0	1	1	1	0	1	1	0	1	0	14
0	1	1	1	1	1	1	0	1	1	15
1	0	0	0	0	1	1	0	0	0	16
1	0	0	0	1	1	1	0	0	1	17
1	0	0	1	0	1	1	1	0	0	18
1	0	0	1	1	1	1	1	0	1	19

Explanation:

- The adder will form the sum in binary and produce a result which may range from 0 to 19. These binary numbers are listed in the truth table and are labeled by the symbols K, Z₈, Z₄, Z₂ and Z₁. K is the carry .
- Z represent the weights 8,4,2 and 1 that can be assigned to the 4 bits in the BCD adder.
- The output sum of two decimal digits must be represented in BCD.

4. Explain about Binary Multiplier.

- Multiplication of binary numbers is performed in the same way as with decimal numbers.
- The multiplicand is multiplied by each bit of the multiplier, starting from the least significant bit. The result of each such multiplication forms a partial product. Successive partial products are shifted one bit to the left.
- The product is obtained by adding these shifted partial products.

Example 1:

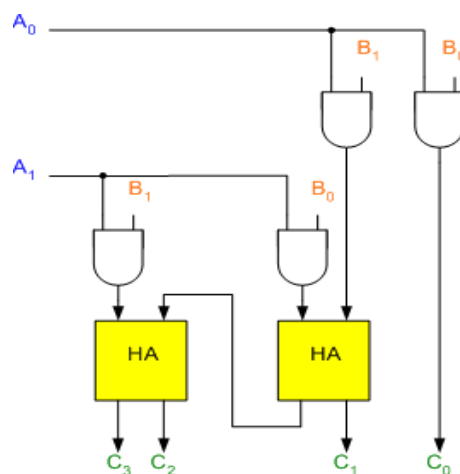
Consider an example of multiplication of two numbers, say A and B (2 bits each), $C = A \times B$.

The first partial product is formed by multiplying the B1B0 by A0. The multiplication of two bits such as A0 and B0 produces a 1 if both bits are 1; otherwise it produces a 0 like an AND operation. So the partial products can be implemented with AND gates.

The second partial product is formed by multiplying the B1B0 by A1 and is shifted one position to the left.

$$\begin{array}{r} B_1 \\ B_1 \\ \times A_1 \\ \hline A_0 B_1 \\ A_0 B_0 \\ \hline A_1 B_1 \\ A_1 B_0 \\ \hline C_3 C_2 C_1 \end{array}$$

The two partial products are added with two half adders (HA). Usually there are more bits in the partial products, and then it will be necessary to use FAs.



The least significant bit of the product does not have to go through an adder, since it is formed by the output of the first AND gate as shown in the Figure.

A binary multiplier with more bits can be constructed in a similar manner.

Example 2:

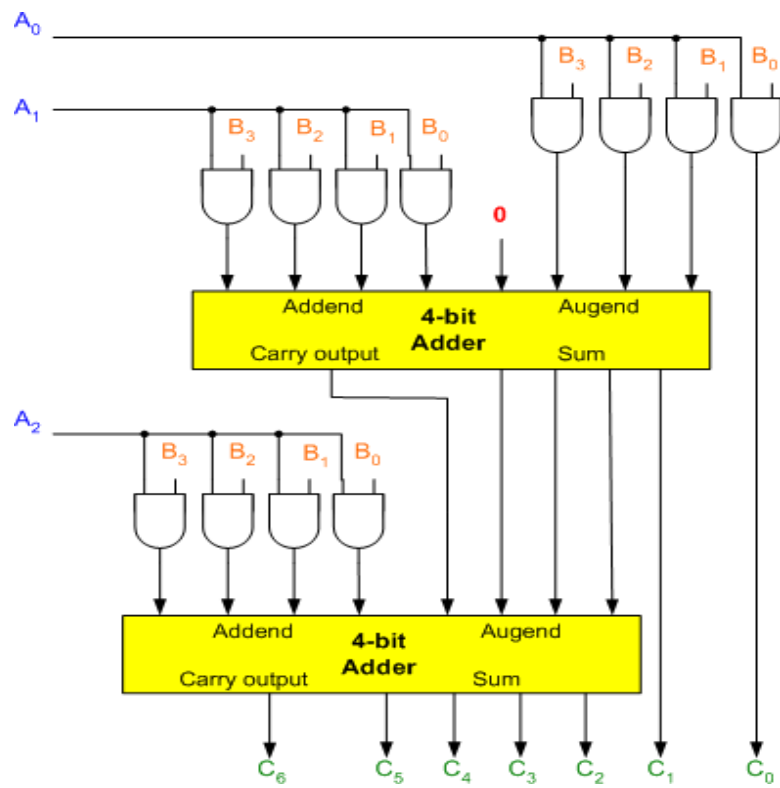
Consider the example of multiplying two numbers, say

A (3-bit number) and B (4-bit number).

Each bit of A (the multiplier) is ANDed with each bit of B (the multiplicand) as shown in the Figure.

$$\begin{array}{r}
 B_3 B_2 B_1 B_0 \\
 \times A_2 A_1 A_0 \\
 \hline
 A_0 B_3 A_0 B_2 A_0 B_1 A_0 B_0 \\
 A_1 B_3 A_1 B_2 A_1 B_1 A_1 B_0 \\
 A_2 B_3 A_2 B_2 A_2 B_1 A_2 B_0 \\
 \hline
 C_6 C_5 C_4 C_3 C_2 C_1 C_0
 \end{array}$$

The binary output in each level of AND gates are added in parallel with the partial product of the previous level to form a new partial product. The last level produces the final product



Since $J = 3$ and $K = 4$, 12 ($J \times K$) AND gates and two 4-bit ($(J - 1) K$ -bit) adders are needed to produce a product of seven ($J + K$) bits. Its circuit is shown in the Figure.

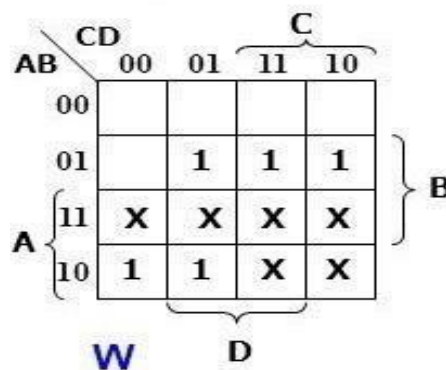
Note that 0 is applied at the most significant bit of augend of first 4-bit adder because the least significant bit of the product does not have to go through an adder.

5. Design a 4 bit BCD to EXCESS 3 code converter. Draw the logic diagram.

Truth Table:-

	BCD				Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0
10	1	0	1	0	X	X	X	X
11	1	0	1	1	X	X	X	X
12	1	1	0	0	X	X	X	X
13	1	1	0	1	X	X	X	X
14	1	1	1	0	X	X	X	X
15	1	1	1	1	X	X	X	X

K-map simplification:-



$$W = A + BC + BD$$

AB \ CD		C			
		00	01	11	10
A	00		1	1	1
	01	1			
	11	X	X	X	X
	10		1	X	X

X
D

$$X = B'C + BC'D' + BD$$

AB \ CD		C			
		00	01	11	10
A	00	1		1	
	01	1		1	
	11	X	X	X	X
	10	1		X	X

Y
D

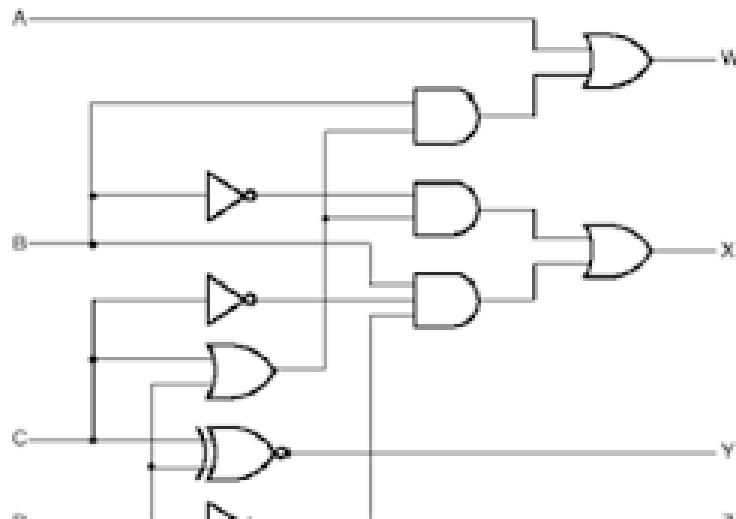
$$Y = CD + C'D'$$

AB \ CD		C			
		00	01	11	10
A	00	1			1
	01	1			1
	11	X	X	X	X
	10	1		X	X

Z
D

$$Z = D'$$

Logic Diagram:-



6. Implement $F(A,B,C,D) = \sum_m(0,2,3,6,8,10,11,12,13,14)$ using 8×1 multiplexer.

Implementation table:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
\bar{A}	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	0	1	1	A	A	1	0

Logic Diagram:-

