Unit – III

Objectives:

- To understand various combinational logic circuits.

Syllabus:

Combinational Logic Circuit Design: Introduction to combinational logic circuits, design of Half adder, full adder, half subtractor, full subtractor, applications of full adders, 4-bit binary subtractor, adder-subtractor circuit, BCD adder circuit, Excess 3 adder circuit, look-a-head adder circuit, Design of decoder, demultiplexer, 7 segment decoder, higher order demultiplexing, encoder, multiplexer, higher order multiplexing, realization of Boolean functions using decoders and multiplexers, priority encoder, 4-bit digital comparator.

Outcomes:

Students will be able to

- Design adders and subtractors.
- Understand 4-bit adders like BCD adder, Excess-3 adder, and look-a-head carry adder.
- Understand other combinational circuits like comparator, decoder, demultiplexer, encoder, multiplexer.
- Realize Boolean functions using decoders and multiplexers.

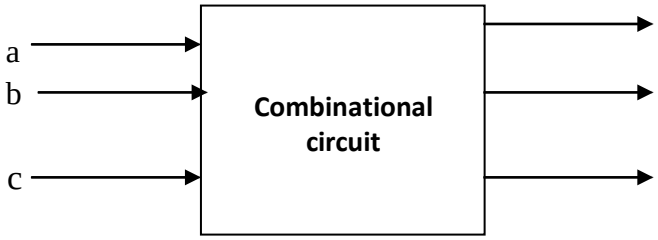
Learning Material

Combinational circuit consists of logic gates whose outputs at anytime are determined directly from the present combination of inputs without regard to previous inputs.
Combinational circuit is a combination of different gates.
For example: encoder, decoder, multiplexer and de-multiplexer etc. are some combinational circuits.

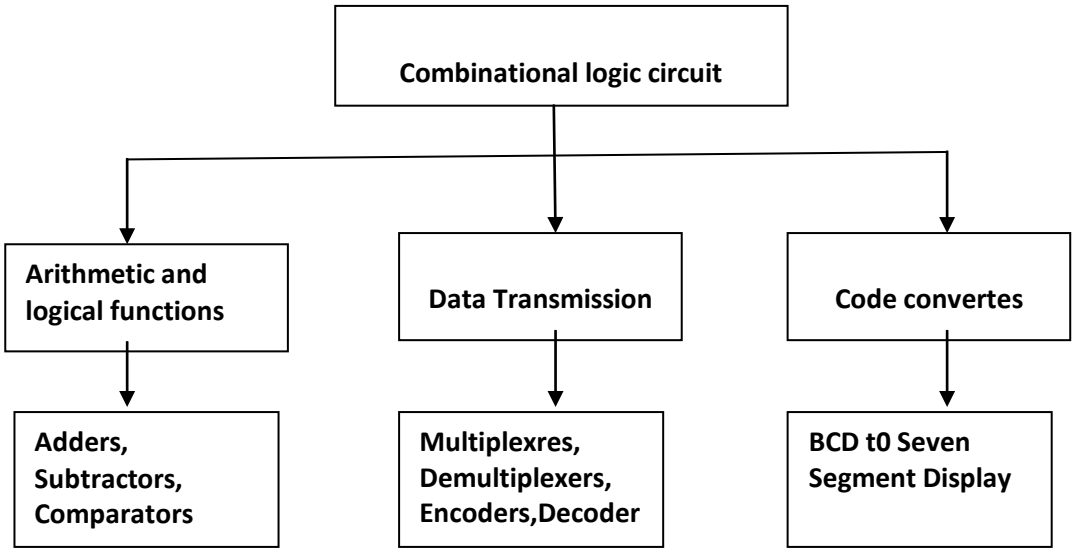
Some of the characteristics of combinational circuits are following:

- The output of combinational circuit at any instant of time depends only on the levels present at input terminals.
- The combinational circuit does not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have a n number of inputs and m number of outputs.

Block diagram



Classification of combinational Logic :



Design procedure of combinational Logic circuit:

1. The problem is stated
2. The number of available input variables and required output variables is determined
3. The input and output variables are assigned letter symbols
4. The truth table that defines the required relationship between inputs and outputs is derived
5. The simplified Boolean function for each output is obtained
6. The logic diagram is drawn.

Adders:

Digital computers perform various arithmetic operations. The most basic arithmetic operation is the addition of two binary digits.

Different types of adders are discussed below:

Half Adder

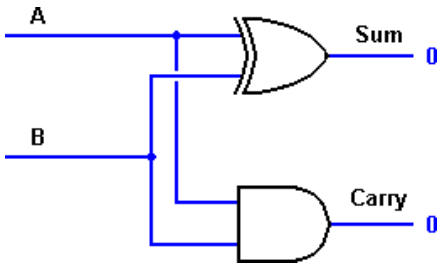
Half adder is a combinational logic circuit that performs the addition of two bits.
Half adder circuit needs two binary inputs and two binary outputs.
The input variables designate the augend and addend bits, the output variables produce the sum and carry.

Block diagram

Truth Table

Inputs		Outputs	
A	**B**	**Sum(s)**	**Carry(c)**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Circuit Diagram:



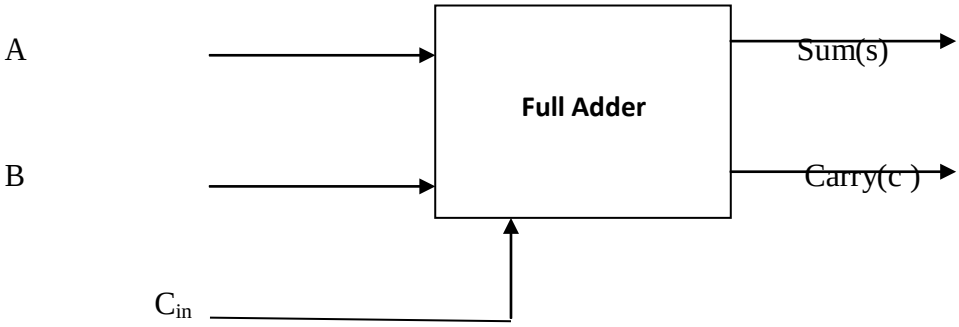
Logic Equations:

Sum(s)= A⊕B; Carry(c)=AB;

Full Adder

The combinational circuit that performs the addition of three bits (two significant bits and previous carry) is called full adder. It consists of three inputs and two outputs. Two significant bits represented as A and B and the third input C_{in} represents the carry from the previous lower significant position. The two outputs are Sum (s) and Carry(c).

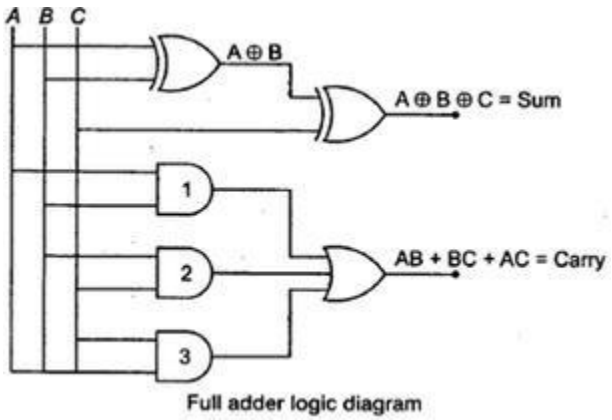
Block diagram:



Truth Table

Inputs			outputs	
A	**B**	**C_{in}**	**Sum**	**Carry**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Circuit Diagram:



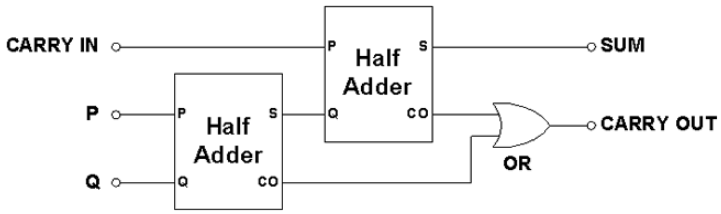
Full adder logic diagram

Logic Equations:

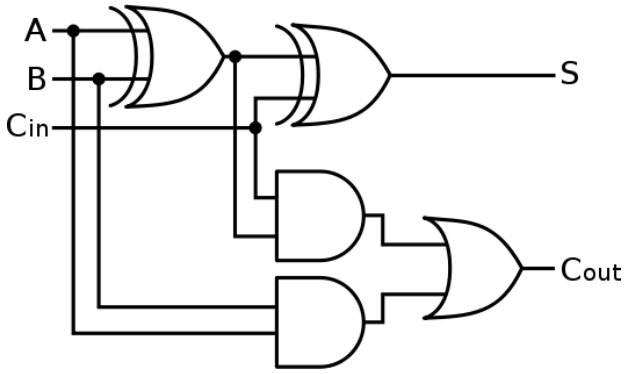
Sum(s)=A $\oplus$B$\oplus C_{in}$

Carry(c)=AB+$BC_{in+}AC_{in}$

Full adder implementation using two half adders and ORGATE:



Logic Diagram:



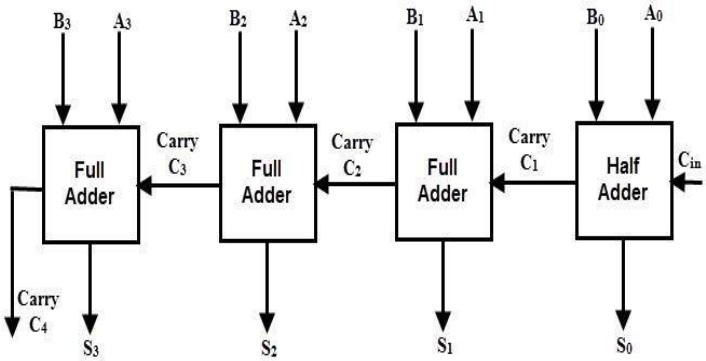
N-Bit Parallel Adder

The Full Adder is capable of adding only two single digit binary number along with a carry input. But in practice we need to add binary numbers which are much longer than just one bit. To add two n-bit binary numbers we need to use the n-bit parallel adder. It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.

4 Bit Parallel Adder

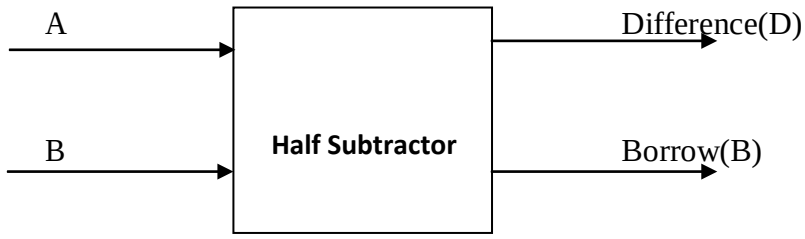
In the block diagram, A0 and B0 represent the LSB of the four bit words A and B. Hence Full Adder-0 is the lowest stage. Hence its Cin has been permanently made 0 .The rest of the connections are exactly same as those of n-bit parallel adder is shown in fig. The four bit parallel adder is a very common logic circuit.

Block diagram:



Half Subtractor

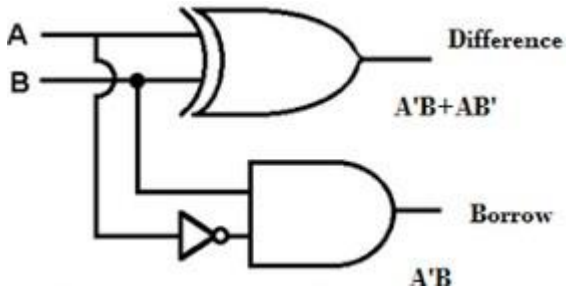
A Half subtractor is a combinational circuit that subtracts two bits and produces their difference. It produces the difference between the two binary bits at the input and also produces an output Borrow to indicate if a1 has been borrowed. In the subtraction A−B, A is called as Minuend bit and B is called as Subtrahend bit.



Truth Table

Inputs		Outputs	
A	**B**	**Difference(D)**	**Borrow(B)**
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Logic Diagram:

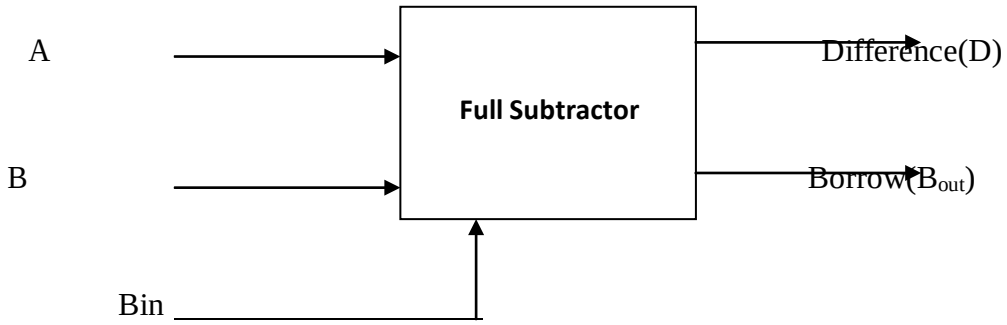


Logic Equations:

Difference(D)= A⊕B; Borrow(B)=A'B;

Full Subtractor:

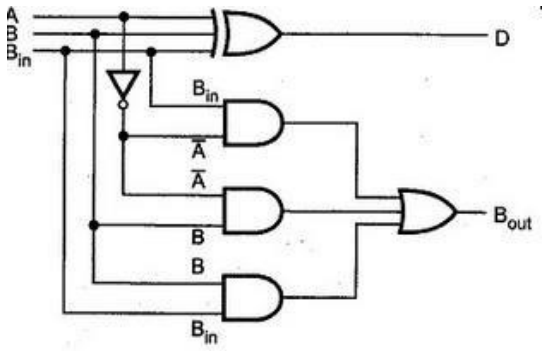
The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A, B, B_{in} and two output D and B_{out}. A is the 'minuend', B is 'subtrahend', C is the 'borrow' produced by the previous stage, D is the difference output and B is the borrow output.



Truth Table

Inputs			outputs	
A	**B**	**B_{in}**	**D**	**B_{out}**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Circuit Diagram



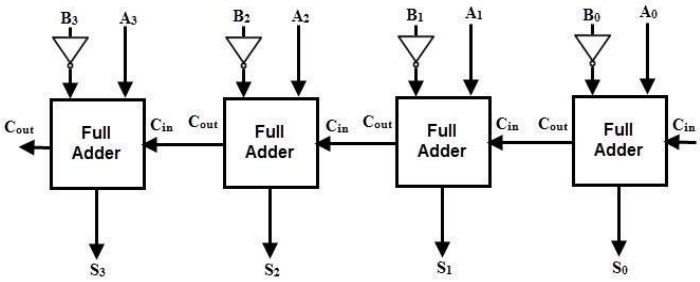
Logic Equations:

Difference(D)= A ⊕B⊕B_{in}

4 Bit Parallel Subtractor

The number to be subtracted B is first passed through inverters to obtain its 1's complement. The 4-bit adder then adds A and 2's complement of B to produce the subtraction. $S_3S_2S_1S_0$ represents the result of binary subtraction A−B and carry output Cout represents the polarity of the result. If A>B then Cout=0 and the result of binary form A−B then Cout=1 and the result is in the 2's complement form.

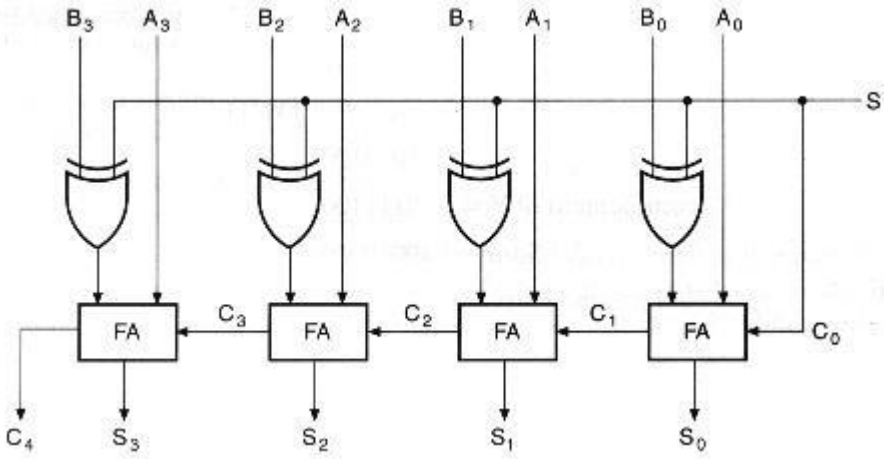
Block diagram



N-Bit Parallel Subtractor

The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted. For example we can perform the subtraction A−B by adding either 1's or 2's complement of B to A. That means we can use a binary adder to perform the binary subtraction.

4-Bit Adder-Subtractor Circuit



This figure represents a 4-bit adder-subtractor circuit. Here the addition and subtraction operations are combined in to one circuit with one common binary adder. The mode input S controls the operation.

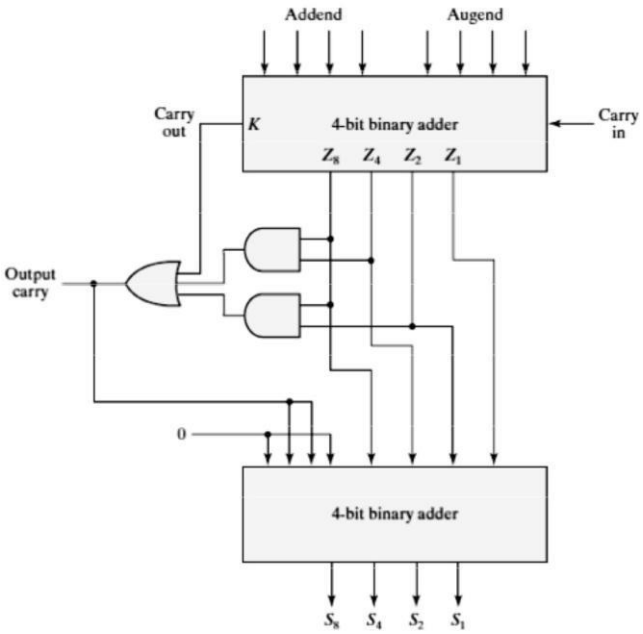
When S=0, the circuit is an adder

When S=1, the circuit is a subtractor

Each XOR gate receives input S and one of the inputs of B. when S=0, we have B $\oplus$ 0 = B. the full adder receives the value of B, the input carry is 0 and the circuit performs A+B.

When S=1, we have B $\oplus$ 1 = B‟ and C1=1. The B inputs are complemented and a 1 is added through the input carry. The circuit performs the operation A plus the 2‟s complement of B.

BCD Adder



The logic circuit that detects the necessary correction can be derived from the entries in the table. It is obvious that a correction is needed when the binary sum has an output carry K = 1. The other six combinations from 1010 through 1111 that need a correction have a 1 in position Z8. To distinguish them from binary 1000 and 1001, which also have a 1 in position Z8, we specify further that either Z 4 or Z2 must have a 1. The condition for a correction and an output carry can be expressed by the Boolean function

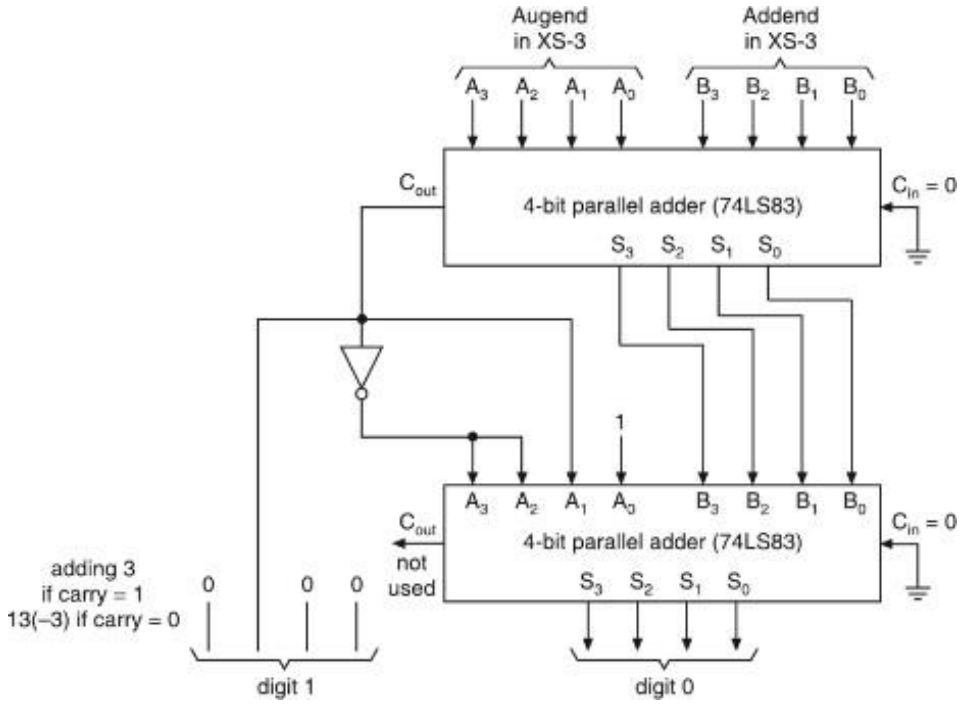
$$C = K + Z_8Z_4 + Z_8Z_2$$

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

Excess-3 Adder Circuit

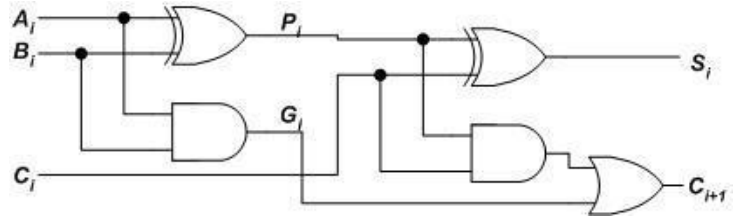
To perform Excess-3 addition, add two XS-3 code groups

- If carry=1, add 0011 (3) to the sum of those two code groups
- If carry=0, subtract 0011 (3), i.e, add 1101 (13 in decimal) to the sum of those two code groups.
- Implementation of XS-3 adder by using 4-bit binary adders is shown in the figure. The Augend and Addend in XS-3 are added using the 4-bit parallel adder. If the carry is a 1, then 0011 (3) is added to the sum bits of the upper adder in the lower 4 bit parallel adder. If the carry is a 0, then 1101 (13) is added to the sum bits (this is equivalent to subtracting 0011 (3) from the sum bits.)



Look Ahead Carry Adder

The Figure shows the full adder circuit used to add the operand bits in the ith column; namely Ai & Bi and the carry bit coming from the previous column (Ci).



In this circuit, the 2 internal signals Pi and Gi are given by:

$$Pi = Ai \oplus Bi \quad \text{.....(1)}$$

$$Gi = Ai\ Bi \quad \text{.....(2)}$$

The output sum and carry can be defined as :

$$Si = Pi \oplus Ci \quad \text{..... (3)}$$

$$C\,i+1 = Gi \oplus Pi\ C\,i \quad \text{..... (4)}$$

G_i is known as the ***carry Generate*** signal since a carry (C_{i+1}) is generated whenever G_i =1, regardless of the input carry (C_i). P_i is known as the ***carry propagate*** signal since whenever P_i =1, the input carry is propagated to the output carry, i.e., C_{i+1}. = C_i (note that whenever P_i =1, G_i =0). Computing the values of P_i and Gi only depend on the input operand bits (Ai & Bi) as clear from the Figure and equations. Thus, these signals settle to their ***steady-state value*** after the propagation through their respective gates.

$C_1 = G_0 + P_0C_0$

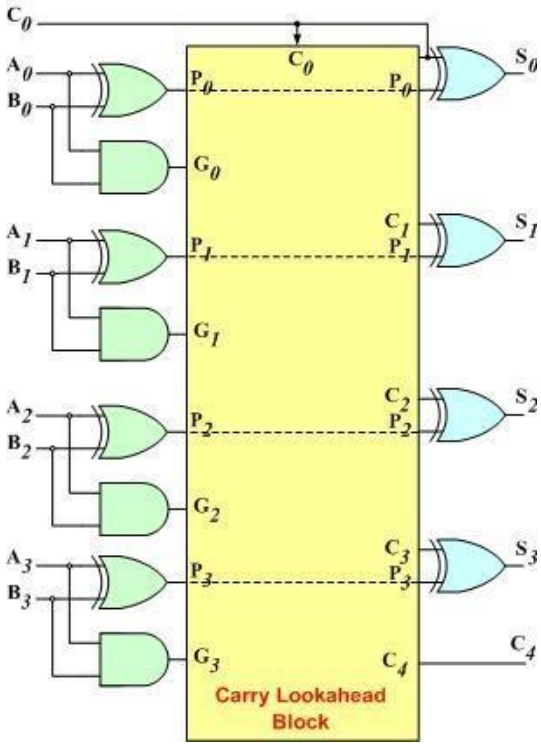
$C_2 = G_1 + P_1C_1 = G_1 + P_1 (G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$

$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$ C_4

$= G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$

In general, the i^{th} carry output is expressed in the form $\mathbf{C_i} = F_i$ (P‟s, G‟s , $\mathbf{C_0}$).
In other words, each carry signal is expressed as a direct SOP function of C_0 rather than its preceding carry signal.
Since the Boolean expression for each output carry is expressed in SOP form, it can be implemented in two-level circuits.
The 2-level implementation of the carry signals has a propagation delay of 2 gates,

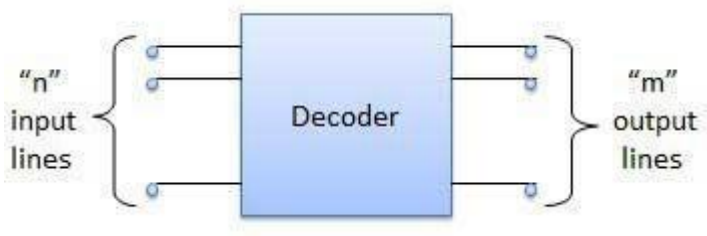
The 4-bit carry look-ahead (CLA) adder consists of 3 levels of logic:



Decoder

A decoder is a combinational circuit. It has n input and to a maximum= 2n outputs. Decoder is identical to a demultiplexer without any data input. It performs operations which are exactly opposite to those of an encoder.

Block diagram



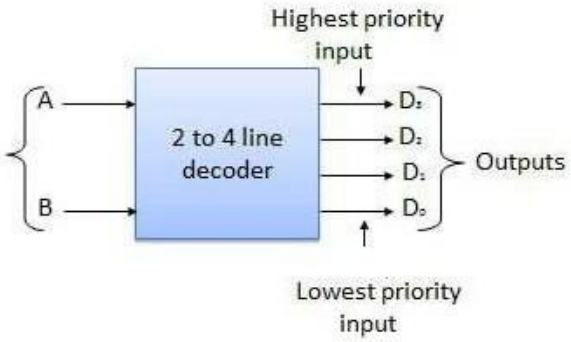
Examples of Decoders are following.

- Code converters
- BCD to seven segment decoders Nixie tube decoders
- Relay actuator

2 to 4 Line Decoder

The block diagram of 2 to 4 line decoder is shown in the fig. A and B are the two inputs where D through D are the four outputs. Truth table explains the operations of a decoder. It shows that each output is 1 for only a specific combination of inputs.

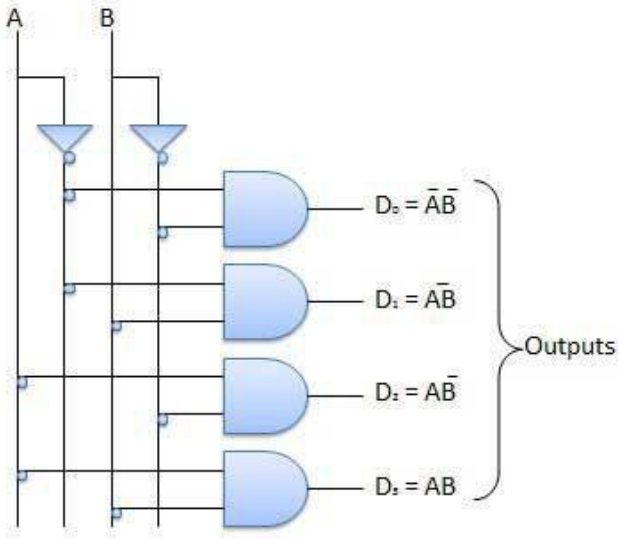
Block diagram



Truth Table

Inputs		Output			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

Logic Circuit



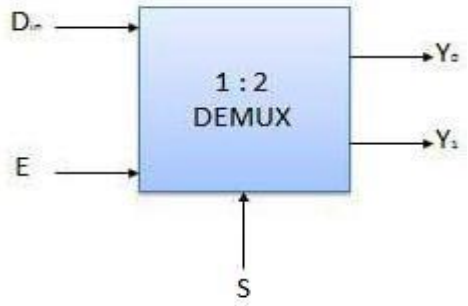
Demultiplexers

A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. It has only one input, n outputs, m select input. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line.

Demultiplexer comes in multiple variations.

- 1:2 demultiplexer
- 1:4 demultiplexer
- 1:16 demultiplexer
- 1:32 demultiplexer

Block diagram



Truth Table

Enable	Select	Output	
E	S	Y0	Y1
0	x	0	0
1	0	0	Din
1	1	Din	0

x = Don't care

Multiplexers:

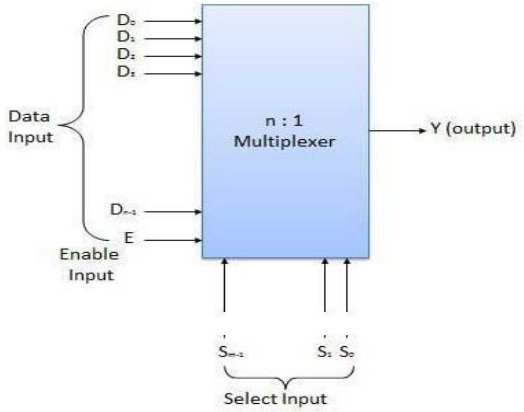
A digital multiplexer is a combinational circuit that selects binary information fron one of many input lines and directs it to a single output line.

The selection of a particular input line is controlled by a set of selection lines.

Normally there are 2^n input lines and an selection lines whose bit combinations determine which input is selected.

E is called the strobe or enable input which is useful for the cascading. It is generally an active low terminal that means it will perform the required operation when it is low.

Block diagram



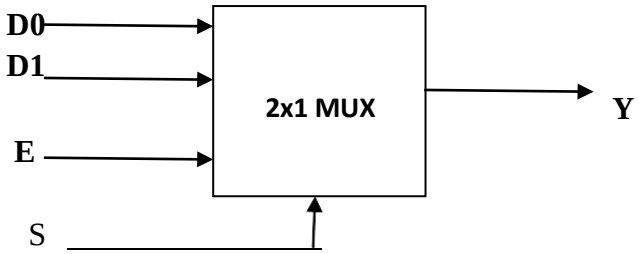
Multiplexers come in multiple variations

- 2:1 multiplexer
- 4:1 multiplexer
- 16:1 multiplexer
- 32:1 multiplexer

2X1 Multiplexer:

A 2 to 1 line multiplexer consists of 2 input lines and one select line and single output line.

Block Diagram:



Truth Table:

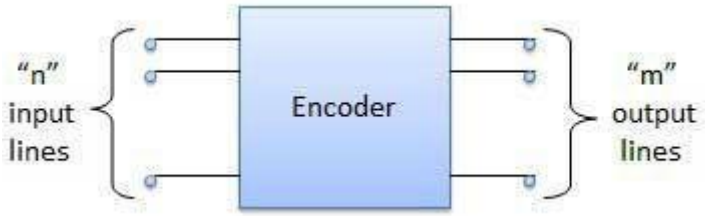
Enable	Select	Output
E	**S**	**Y**
0	X	0
1	0	D_0
1	1	D_1

X=Don't Care

Encoder

Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. An encoder has n number of input lines and m number of output lines. An encoder produces an m bit binary code corresponding to the digital input number. The encoder accepts an n input digital word and converts it into an m bit another digital word.

Block diagram



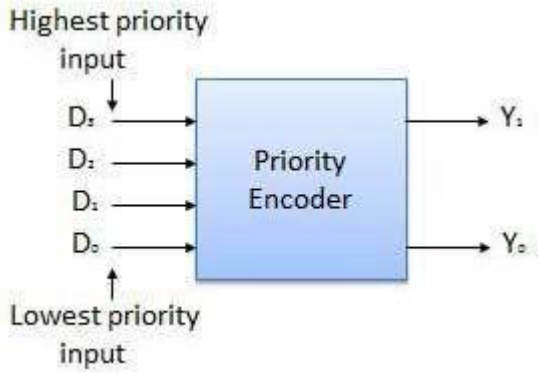
Examples of Encoders are following.

- Priority encoders
- Decimal to BCD encoder Octal to binary encoder
- Hexadecimal to binary encoder

Priority Encoder

This is a special type of encoder. Priority is given to the input lines. If two or more input line are 1 at the same time, then the input line with highest priority will be considered. There are four input D0, D1, D2, D3 and two output Y0, Y1. Out of the four input D3 has the highest priority and D0 has the lowest priority. That means if D3=1 then Y1Y1=11 irrespective of the other inputs. Similarly if D3= 0 and D2=1 then Y1Y0=10 irrespective of the other inputs.

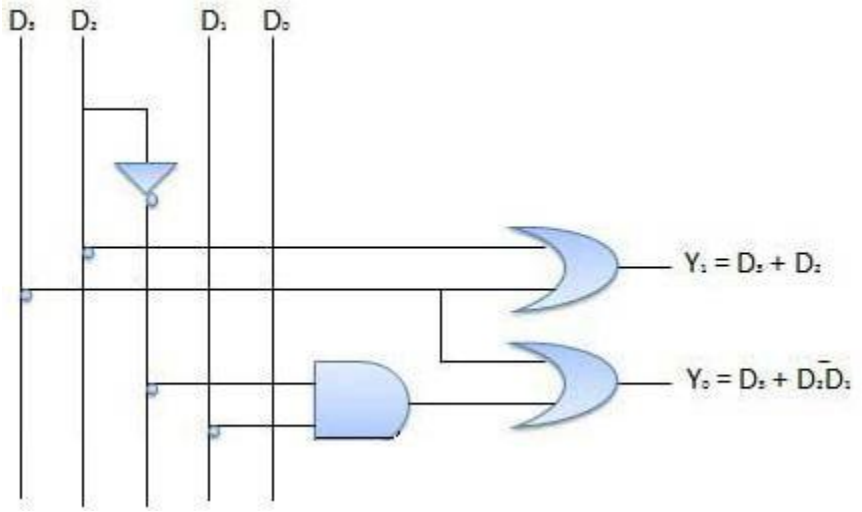
Block diagram



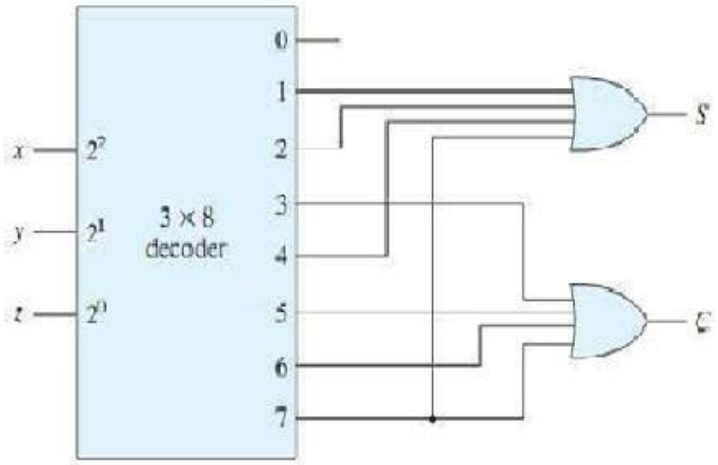
Truth Table

Highest	Inputs		Lowest	Outputs	
D_3	D_2	D_1	D_0	Y_0	Y_1
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

Logic Circuit



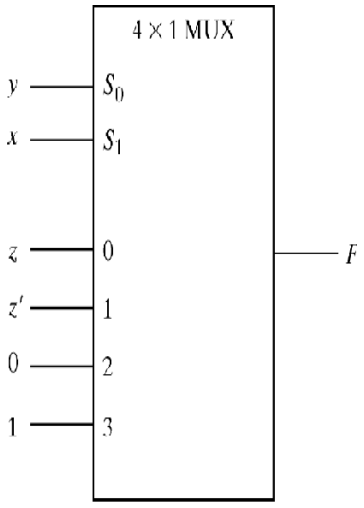
- **Implement full adder circuit whose outputs are given as : S (x,y,z) = Σ (1,2,4,7) C(x,y,z) = Σ(3,5,6,7) with a suitable decoder and external gates**



- **Implement the function F(x,y,z) = m(1,2,6,7) x, and y should be connected with the same order to S1and S0respectively**

x	y	z	F	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	

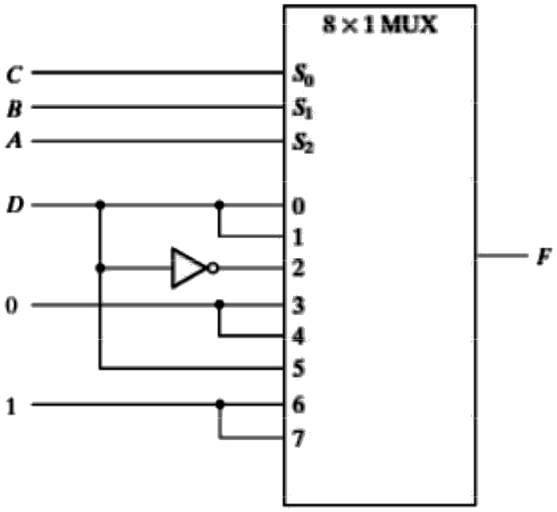
(a) Truth table



(b) Multiplexer implementation

- **Implementation of the Boolean function F (A, B, C, D) = ∑(1, 3, 4, 11, 12, 13, 14, 15) using 8 X 1 MUX**

A	B	C	D	F	
0	0	0	0	0	F = D
0	0	0	1	1	
0	0	1	0	0	F = D
0	0	1	1	1	
0	1	0	0	1	F = D′
0	1	0	1	0	
0	1	1	0	0	F = 0
0	1	1	1	0	
1	0	0	0	0	F = 0
1	0	0	1	0	
1	0	1	0	0	F = D
1	0	1	1	1	
1	1	0	0	1	F = 1
1	1	0	1	1	
1	1	1	0	1	F = 1
1	1	1	1	1	

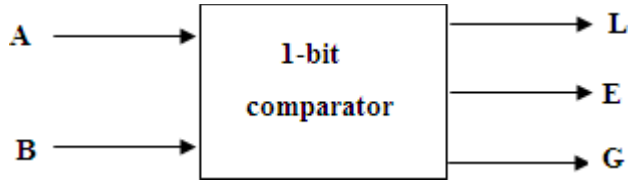


Magnitude Comparator

- A magnitude comparator is a combinational logic circuit that compares the two numbers, A and B and determines their relative magnitudes.
- The outcome of the comparison is specified by the three binary variables that indicate whether A>B,A=B,A<B.

The X-NOR gate (coincidence gate) is a basic comparator, because its output is a 1 only if its two input bits are equal, i.e. the output is a 1 if and only if the input bits coincide.

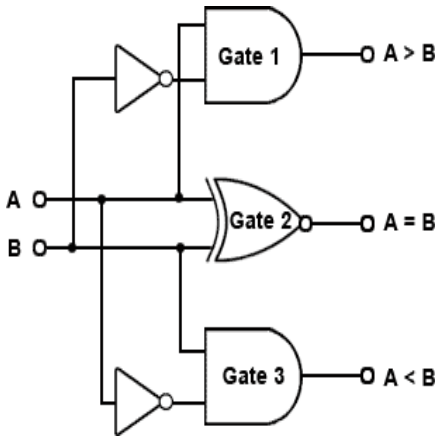
1-Bit Magnitude Comparator



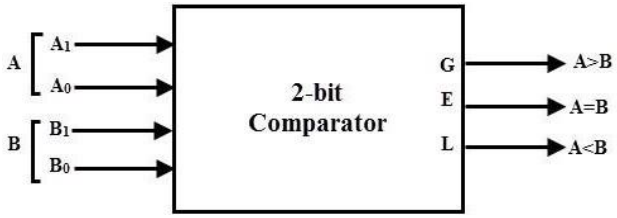
Truth Table

A	B	L A< B	E A=B	G A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Logic diagram of 1-bit comparator



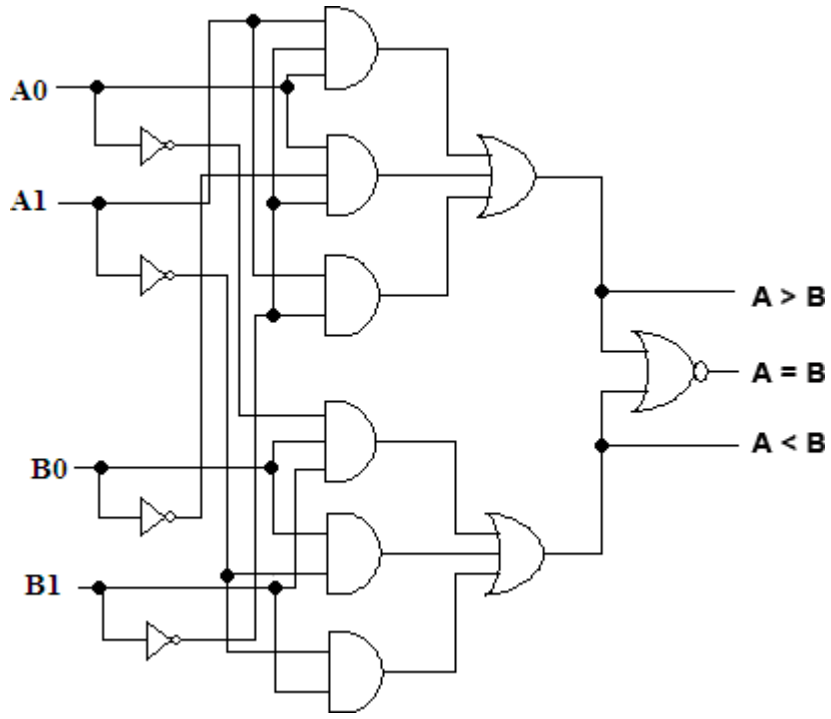
2-bit magnitude comparator:



Truth Table:

INPUT				OUTPUT		
A1	A0	B1	B0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Logic diagram of 2-bit comparator



Assignment-Cum-Tutorial Questions

A. Questions testing the remembering / understanding level of students

I) Multiple Choice Questions:

1. A full subtractor circuit requires_____.

   A. Two inputs and two outputs B. Two inputs and three outputs

   C. Three inputs and one output D. Three inputs and two outputs

2. A de multiplexer has_____.

   A. One data input and a number of selection inputs, and they have several outputs

   B. One input and one output

   C. Several inputs and several outputs

   D. Several inputs and one output

3. How many outputs are on a BCD decoder?

   A.4 B.16 C.8 D.10

4. A decoder converts_____.

   A. Non coded information into coded form B. Coded information into non coded form

   C. HIGHs to LOWs D. LOWs to HIGHs

5. Parallel Adders are

   A. Combinational logic circuits B. Sequential logic circuits

   C. Both of the above D. None of the above

6. A Full Adder can be realized using

   A. One half adder, two OR gates B. Two half adders, one OR gate

   C. Two half adders, two OR gates D. Two half adders, one AND gate

7. In which of the following adder circuits is the carry ripple delay is eliminated?

   A. Half adder B. Full adder C. Parallel adder D. Carry-look-ahead-adder

8. Which logic gate is a basic comparator?

   A. NOR gate B. NAND gate C. X-OR gate D. X-NOR gate

9. A multiplexer is also known as

   A. A data accumulator B. A data restorer C. A data selector D. A data distributor

10. Which logic device is called a distributor?

    A. Multiplexer B. Demultiplexer C. Encoder D. Decoder

II) Descriptive Questions

1. Define combinational logic? Write the design procedure for combinational circuits.
2. Explain the operation of half adder? Realize full adder using logic gates.
3. Explain the operation of half subtractor? Realize full subtractor using logic gates.
4. Discuss the functional principle of 4-bit ripple carry adder. what is its major disadvantage?
5. What is decoder? Draw the logic diagram of 3 to 8 line decoder and explain its operation.

6. What is the difference between encoder and priority encoder? Give the implementation of a 4-bit priority encoder?
7. Discuss how four bit excess - 3 adder circuit is designed. Explain its operation.
8. Discuss how four bit BCD adder circuit is designed. Explain its operation.
9. Briefly describe the concept of look-ahead carry generation with respect to its use in adder circuits.
10. Draw the circuit diagram of a 4-bit adder/subtractor and briefly describe its functional principle.
11. Implement the following function with 8 to 1 multiplexer:

$$f(w, x, y, z) = w\bar{x}\bar{y}z + w\bar{x}y\bar{z} + wy\bar{}$$

12. Implement the three-variable Boolean function using an 8-to-1multiplexer

$$F(A, B, C) = \bar{A}C + \bar{A}BC + A\bar{B}\bar{C}$$

B. Question testing the ability of students in applying the concepts.

I) Multiple Choice Questions

1. How many outputs are on a BCD decoder?

   A.4 B.16 C.8 D.10

2. Which digital system translates coded characters into a more useful form?

   A. Encoder B. Display C. Counter D. Decoder

3. The expansion inputs to a comparator are used for expansion to a (n):

   A. 4-bit system B. 8-bit system C. BCD system D. counter system

4. What control signals may be necessary to operate a 1-line-to-16 line decoder?

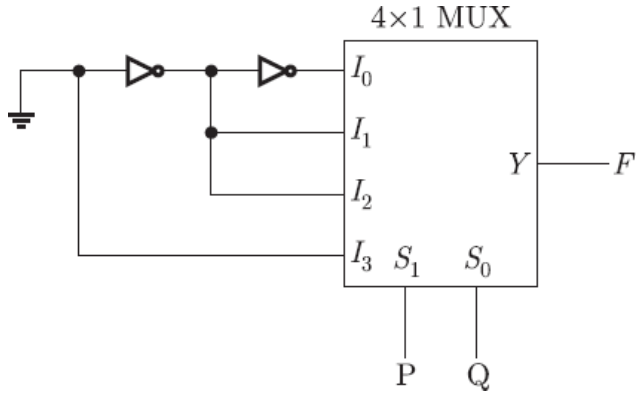
   A. Flasher circuit control signal B. A LOW on all gate enable inputs

   C. Input from a hexadecimal counter D. A HIGH on all gate enable circuits

5. How many exclusive-NOR gates would be required for an 8-bit comparator circuit?

   A.4 B.6 C.8 D.10

6. The logic function implemented by the circuit below is (ground implies a logic "0")



A. F= AND (P, Q) B. F= OR (P, Q) C. F= XNOR (P, Q) D. F= XOR (P, Q)

7. A digital system is required to amplify a binary-encoded audio signal. The user should be

able to control the gain of the amplifier from minimum to a maximum in 100 increments. The minimum number of bits required to encode, in straight binary, is

A. 8 B. 6 C. 5 D. 7

8. The minimum number of 2-input NAND/NOR gates required to realize a half adder is

A. 3 B. 4 C. 5 D. 6

9. The minimum number of 2-input NAND gates required to realize a full adder / full subtractor is

A. 8 B. 9 C. 10 D. 12

10. BCD subtraction is performed using

A. 1‟s complement representation B. 2‟s complement representation

C. 5‟s complement representation D. 9‟s complement representation

II) Problems

1. Realize the logic expression given below using a (i) 8:1 MUX (ii) 16:1 MUX

   $f=\sum m(0,1,3,5,8,11,12,14,15)$

2. Design a 32:1 multiplexer using two 16:1 and 2:1 multiplexers.
3. Implement the following multiple output combinational logic circuit using a 4 to 16 decoder:

   $F1=\sum m(0,1,4,7,12,14,15)$ $F2=\sum m(1,3,6,9,12)$ $F3=\sum m(2,3,7,8,10)$

   $F4=\sum m(1,3,5)$
4. Implement the full adder sum and carry functions with decoder and multiplexers.
5. Implement the following 2:1 multiplexer

   a) and gate b) or gate c) not gate d) ex-or gate
6. Develop a 3-to-8 line decoder using NOR gates only, and draw its logic diagram.
7. A combinational circuit is defined by the equations

   f l = AB + A‟B‟C‟

   f2 = A + B + C‟

   f3 = A‟B + AB‟

   Design a circuit which will implement these three equations using a decoder and NAND gates external to the decoder.

C. Questions testing the analyzing / evaluating ability of students

1. Design a combinational circuit that detects an error in the representation of a decimal digit in BCD. The output of the circuit must be equal to logic 1 when the inputs contain any one of the six unused bit combinations in the BCD code.

2. A combinational circuit is defined by the following three functions $F_1 = x'y'+xyz'$, $F_2 = x'+y$, $F_3=xy+x'y'$. Design the circuit with a decoder and external gates.

3. A logic function has four inputs A, B. C and D that will produce output 1 whenever two adjacent input variables are 1‟s. Treat A and D are also adjacent. Implement this logic function using 8 x 1 and 4 x 1 multiplexers.

4. Obtain logical functions to design decimal to octal using priority encoder.

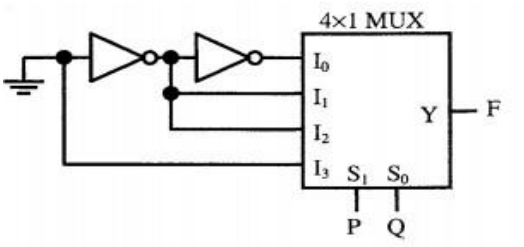
D. GATE/IES Questions

1) The output Y of a 2 bit comparator is logic 1whenever 2-bit input A is greater than 2-bit input B. The no. of combinations for which the output is logic 1 is **GATE-2012**

A. 4 B. 6 C. 8 D.10

2) The logic function implemented by the circuit below is (ground implies a logic „0“)

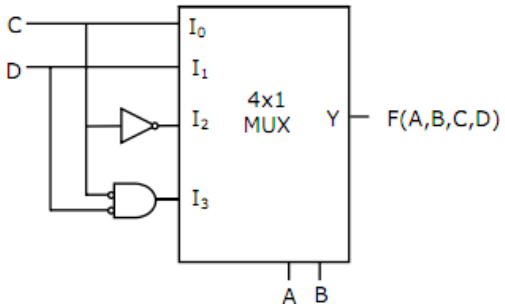
GATE-2011



A. F= AND (P,Q) B. F= OR (P,Q) C. F= XNOR (P,Q) D. F= XOR (P,Q)

3) The Boolean function realized by the logic circuit shown is

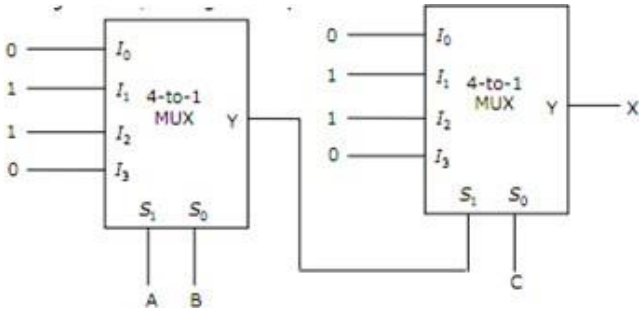
GATE-2010



A. F= Σm(0,1,3,5,9,10,14) B. F= Σm(2,3,5,7,8,12,13)

C. F= Σm(1,2,4,5,11,14,15) D. F= Σm(2,3,5,7,8,9,12)

4) In the following circuit X is given by **GATE-2007**

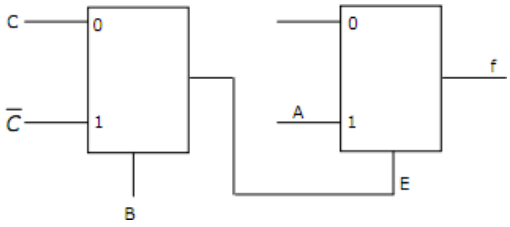


A.

X=AB“C“+A“BC“+A“B“C+ABC. B. X= AB“C“+A“BC“+A“B“C+ABC

C. X=AB +BC+AC D. X=A“B“+B“C“+A“C“

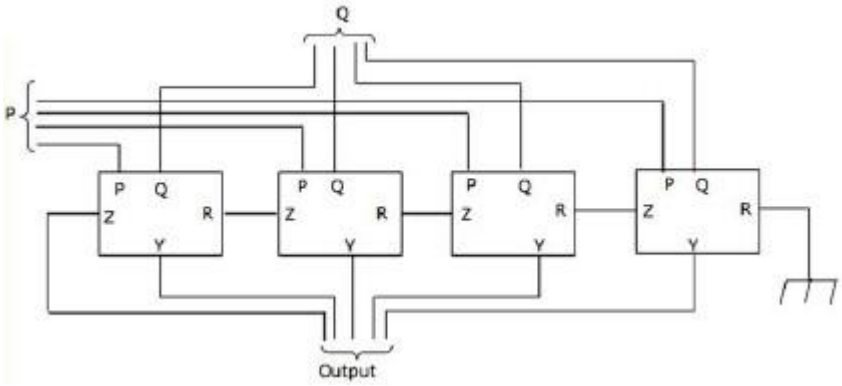
5) The Boolean function f implemented in figure using two input multiplexers is **GATE-2005**



A. AB‟C + ABC‟ B. ABC + AB‟C‟ C. A‟BC + A‟B‟C‟ D. A‟B‟C + A‟BC‟

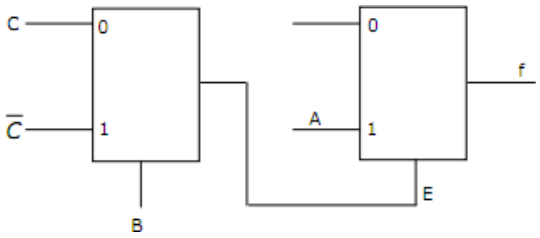
6) The minimum no. of 2:1 multiplexers required to realize a 4:1multiplexer is **GATE-2004**
A. 1 B. 2 C. 3 D. 4

7) The circuit shown in figure below has 4 boxes each described by inputs P, Q, R and outputs Y, Z with Y= P ⊕ Q ⊕ R; Z=RQ+ P‟R+Q P‟. The circuit acts as a
GATE-2003



A. 4 bit adder giving P+Q B. 4 bit subtractor giving P-Q
C. 4 bit subtractor giving Q-P D. 4 bit adder giving P+Q+R

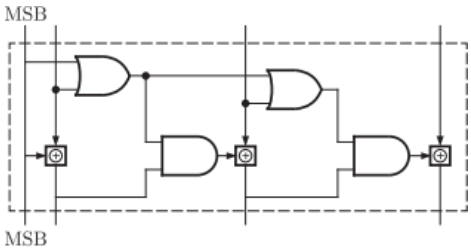
8) The Boolean function f implemented in figure using 2 input multiplexers is **GATE 2005**



A. AB‟C+ABC‟ B. ABC+AB‟C‟ C. A‟BC+A‟B‟C‟ D. A‟B‟C+A‟BC‟

9) The minimum number of 2 to 1MUX requires to realize a 4 to 1 MUX are **GATE 2004**
A. 1 B. 2 C. 3 D. 4

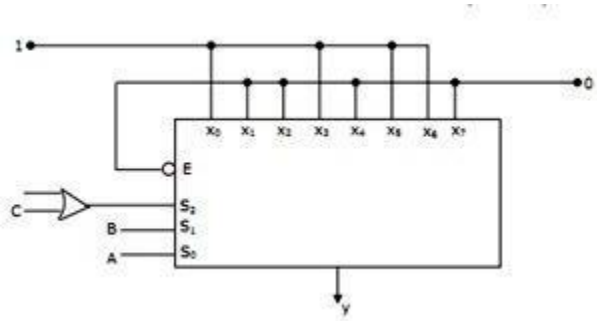
10) The circuit shown in figure converts **GATE 2003**



A. BCD to binary code B. Binary to Excess-3 C. Excess-3 to gray D. Gray to binary

11) Without any additional circuitry, an 8:1 MUX can be used to obtain **GATE 2003**
A. Some but not all Boolean functions of 3 Variables
B. All functions of 3 variables but not of 4 variables
C. All functions of 3 variables and some but not all functions of 4 variables
D. All functions of 4 variables

12) In the TTL circuit in figure below s_2 to s_0 are select lines and x_7 and x_0 are input lines . s_0 and x_0 are LSB‟s. The output Y is **GATE-2001**
A. indeterminate B. A⊕B C. (A⊕B)‟ D. C‟(A⊕B) + C(A⊕B)‟



13) For a binary halfsubtractor having inputs A and B ,The correct set of logical expressions for the outputs D and X are

A. D=AB+A‟B, X=A‟B B. D=A‟B+AB‟, X=A‟B **GATE-1999**

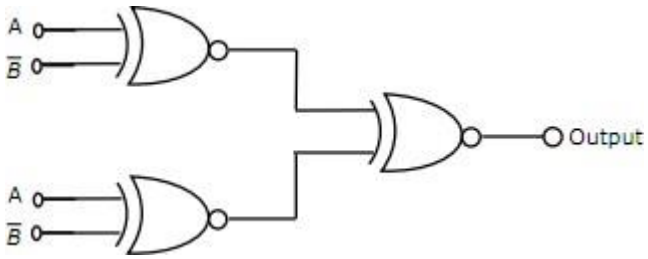
C. D=A‟B+AB‟, X=AB‟ D. D=AB+A‟B‟, X=AB‟

14) A 2 bit binary multipler can be implemented using **GATE-1997**

A.2 inputs AND only B.2 input XORS and 4 input AND gates only

C.Two 2 inputs NORS and one XOR gate D.XOR gates and shift registers

15) The output of the circuit shown in figure is equal to **GATE-1995**



A.0 B.1 C. A‟B+AB‟ D. (AB)‟.(AB)‟

16) The logic realized by the circuit shown in figure is: **GATE-1992**

A. F=A.C B. F=A+C C. F=B.C D. F=B+C