Royal Education Society's

COCSIT LATUR

FACULTY OF COMPUTER STUDIES

[CBCS REVISED PATTERN]

B.Sc.(CS)FY (Semester I)

Model Question Paper

Subject – Fundamentals of Digital Electronics (BCS-104B)

Time: *Three Hours* ***Maximum Marks*** *- 75*

Instructions to the candidates:

1. All questions are Compulsory.

2. Figures to the right indicate full marks.

3. Assume suitable data, if required.

Q.1 Attempt any FIVE of the following (3 Marks each) **15**

a) Explain excess-3 code in detail.

b) Explain Delay FF.

c) Explain full adder.

d) Explain 1:2 De-multiplexer.

e) Explain X-OR and X-NOR gates.

f) What is 1's complement? Explain with example.

g) Explain PIPO shift register.

Q. 2 Attempt any three of the following (5 Marks each) **15**

a) State and prove De-Morgan's theorems.

b) Explain NAND and NOR universal gates.

c) Explain AND, OR, NOT basic gates.

d)What is signal? Explain Analog and Digital signals.

e) Explain decimal to BCD encoder.

Q. 3 Attempt any three of the following (5 Marks each) **15**

a) Design 8:1 multiplexer.

b) What is shift register? Explain SISO shift register.

c) Explain clocked SR FF.

d)Explain Digital to Analog converter.

e) Explain Hamming code in detail.

Q. 4 Attempt any three of the following (5 Marks each) **15**

a) What is counter? Explain asynchronous counter in detail.

b) Solve the following.

i) (11011+01011) ii) (110-100)

c) Convert the following.

i) $(A54)_{16} = (?)_2$ ii) $(654.32)_8 = (?)_2$

d) Explain BCD to decimal decoder.

e) Minimize using K-map:

$f(A,B,C,D) = \sum m\,(0,1,4,5,10,11,12) + d(2,3)$

Q .5 Short notes on any three of the following (5 Marks each) **15**

a) BCD code

b) 2's complement.

c) Parity code

d) half adder.

e) Analog to digital converter.

Model Answer Sheet

Q.1 Attempt any FIVE of the following (3 Marks each) **15**

a) Explain excess-3 code in detail.

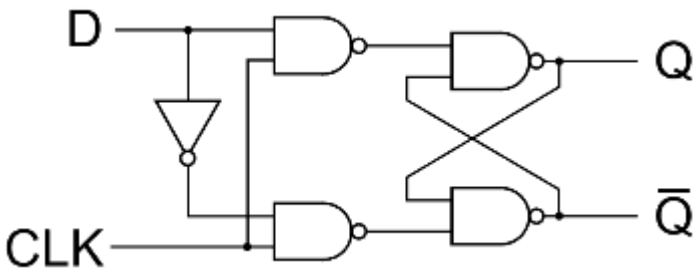
The excess-3 code is also treated as **XS-3 code**. The excess-3 code is a non-weighted and self-complementary BCD code used to represent the decimal numbers. This code has a biased representation. This code plays an important role in arithmetic operations because it resolves deficiencies encountered when we use the 8421 BCD code for adding two decimal digits whose sum is greater than 9. The Excess-3 code uses a special type of algorithm, which differs from the binary positional number system or normal non-biased BCD.

Decimal Digit	BCD Code	Excess-3 Code
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

b) Explain Delay FF.

A D-type flip-flop is a clocked flip-flop which has two stable states. A D-type flip-flop operates with a delay in input by one clock cycle. Thus, by cascading many D-type flip-flops delay circuits can be created, which are used in many applications such as in digital television systems.

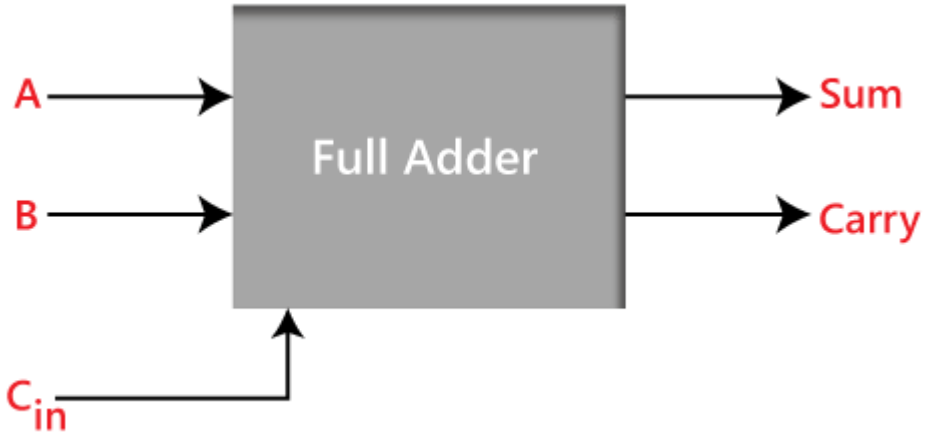
A D-type flip-flop is also known as a D flip-flop or delay flip-flop.



c) Explain full adder.

The half adder is used to add only two numbers. To overcome this problem, the full adder was developed. The full adder is used to add three 1-bit binary numbers A, B, and carry C. The full adder has three input states and two output states i.e., sum and carry.

Block diagram

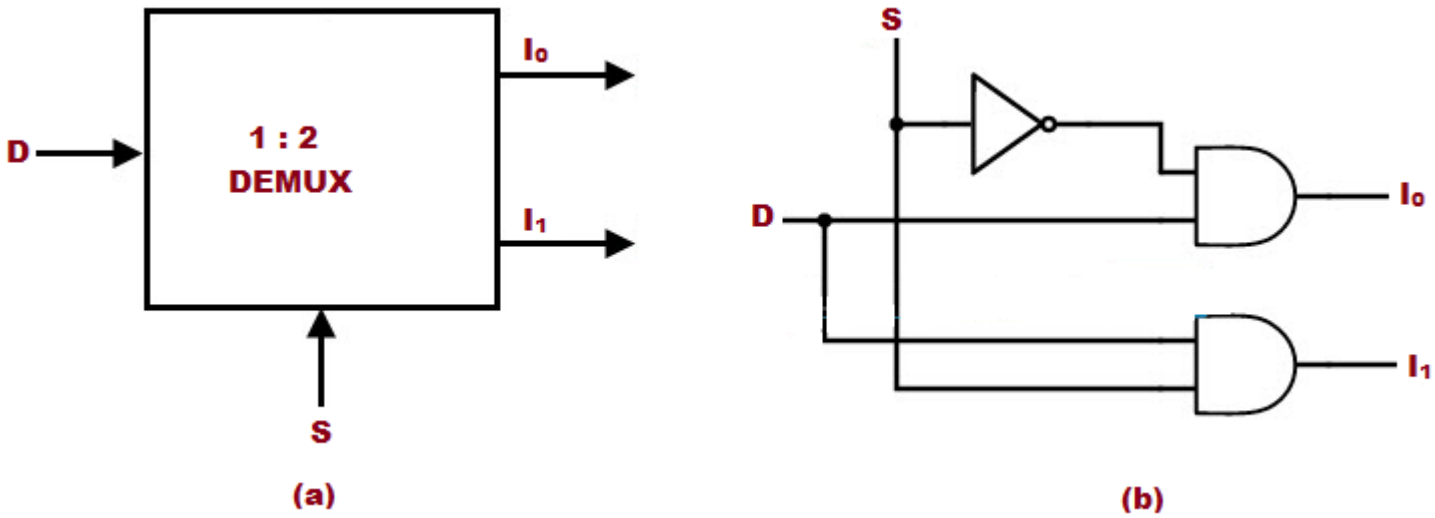


Truth Table

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

d) Explain 1:2 De-multiplexer.

A De-multiplexer is a combinational circuit that has only 1 input line and 2N output lines. Simply, the multiplexer is a single-input and multi-output combinational circuit. The information is received from the single input lines and directed to the output line. On the basis of the values of the selection lines, the input will be connected to one of these outputs. De-multiplexer is opposite to the multiplexer.



e) Explain X-OR and X-NOR gates.

The logic gate performs this modulo sum operation without including carry is known as **XOR gate**. An XOR gate is normally two inputs logic gate where the output is only logical 1 when only one input is logical 1. When both inputs are equal, either are 1 or both are 0, the output will be logical 0.

in the XNOR gate, the inverse is true. Hence the output is 0 when only one input is 0, and the output is 1 when both inputs are the same (i.e. either two 0's or two 1's).

f) What is 1's complement? Explain with example.

1's complement of a binary number is another binary number obtained by toggling all bits in it, i.e., transforming the 0 bit to 1 and the 1 bit to 0.

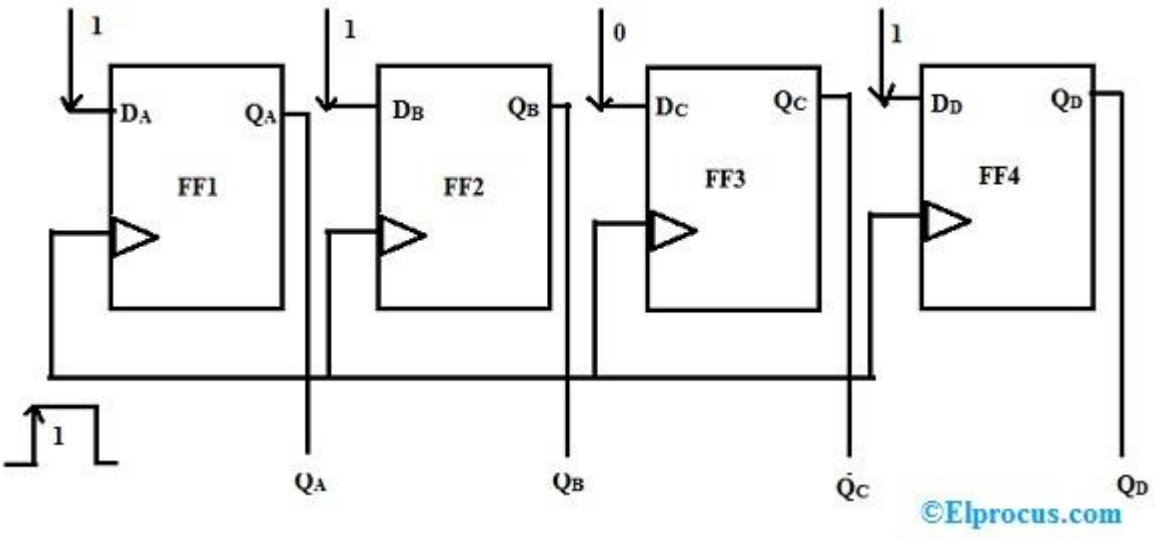
Examples:

1's complement of "0111" is "1000"

1's complement of "1100" is "0011"

g) Explain PIPO shift register.

The shift register which uses parallel input and generates parallel output is known as the parallel input parallel output shift register. This shift register includes three connections only the PI (parallel i/p), PO (parallel o/p) & the CLK signal. This kind of shift register also works like a time delay device or temporary storage device like a SISO shift register with the time delay being changed through the CLK signals frequency.



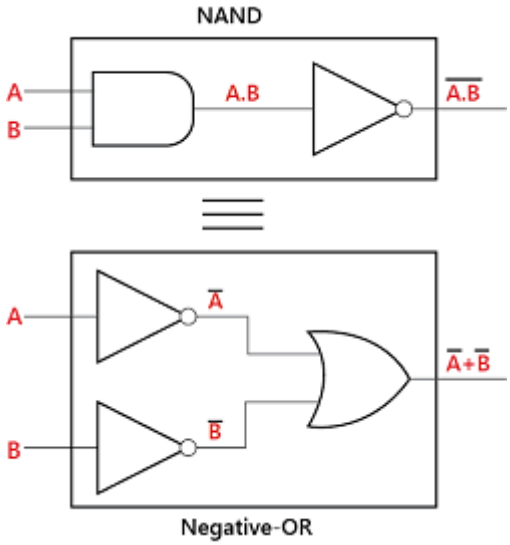
Q. 2 Attempt any three of the following (5 Marks each) **15**

a) State and prove De-Morgan's theorems.

De-Morgan's First Theorem

According to the first theorem, the complement result of the AND operation is equal to the OR operation of the complement of that variable. Thus, it is equivalent to the NAND function and is a negative-OR function proving that (A.B)' = A'+B' and we can show this using the following table.

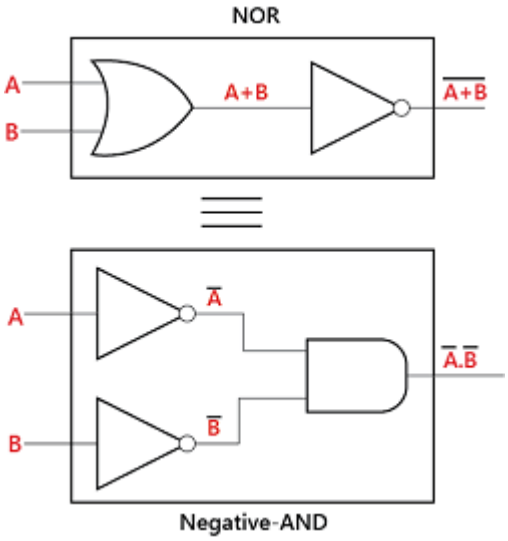
Inputs		Output For Each Term				
A	B	A.B	(A.B)'	A'	B'	A'A+B'
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0



De-Morgan's Second Theorem

According to the second theorem, the complement result of the OR operation is equal to the AND operation of the complement of that variable. Thus, it is the equivalent of the NOR function and is a negative-AND function proving that (A+B)' = A'.B' and we can show this using the following truth table.

Inputs		Output For Each Term				
A	B	A+B	(A+B)'	A'	B'	A'.B'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0



b) Explain NAND and NOR universal gates.

NAND Gate: The NAND gate represents the complement of the AND operation. Its name is an abbreviation of NOT AND. The graphic symbol for the NAND gate consists of an AND symbol with a bubble on the output, denoting that a complement operation is performed on the output of the AND gate. The truth table and the graphic symbol of NAND gate is shown in the figure. The truth table clearly shows that the NAND operation is the complement of the AND.
NOR Gate: The NOR gate represents the complement of the OR operation. Its name is an abbreviation of NOT OR. The graphic symbol for the NOR gate consists of an OR symbol with a bubble on the output, denoting that a complement operation is performed on the output of the OR gate. The truth table and the graphic symbol of NOR gate is-

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

Q = A NAND B

Truth Table

Input A	Input B	Output Q
0	0	1
0	1	1
1	0	1
1	1	0

c) Explain AND, OR, NOT basic gates.

OR Gate

In an OR gate, the output of an OR gate attains state 1 if one or more inputs attain state 1.

The Boolean expression of the OR gate is Y = A + B, read as Y equals A ‘OR’ B.

The truth table of a two-input OR basic gate is given as;

A	**B**	**Y**
0	0	0
0	1	1
1	0	1
1	1	1

AND Gate

In the AND gate, the output of an AND gate attains state 1 if and only if all the inputs are in state 1.

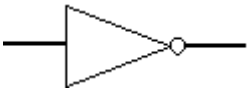
The Boolean expression of AND gate is Y = A.B

The truth table of a two-input AND basic gate is given as;

A	**B**	**Y**
0	0	0
0	1	0
1	0	0
1	1	1

NOT Gate

In a NOT gate, the output of a NOT gate attains state 1 if and only if the input does not attain state 1.



The Boolean expression is:

$Y=\overline{A}$

It is read as Y equals NOT A.

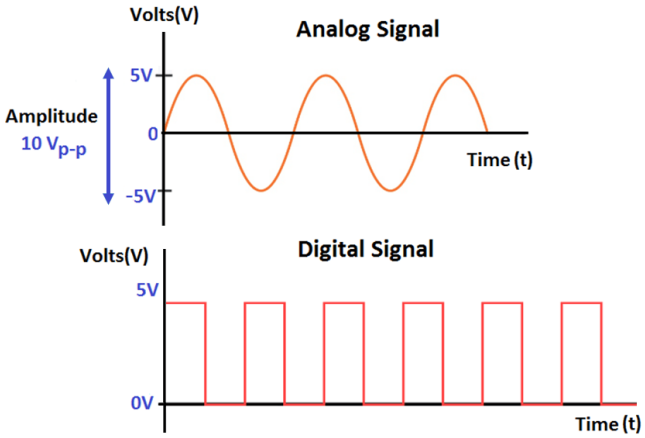
The truth table of NOT gate is as follows;

A	**Y**
0	1
1	0

d) What is signal? Explain Analog and Digital signals.

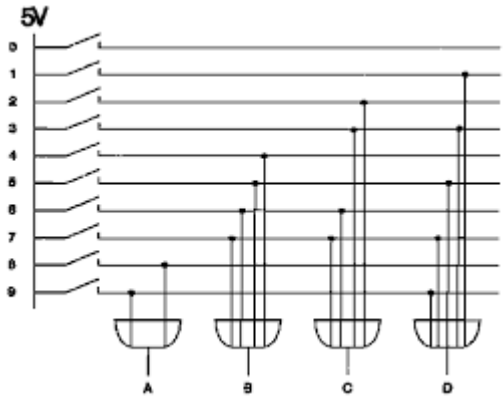
A signal, which is a continuous function of time and used to carry the information is known as an analog signal. An analog signal represents a quantity analogous to another quantity, for example, in case of an analog audio signal, the instantaneous value of signal voltage represents the pressure of the sound wave.

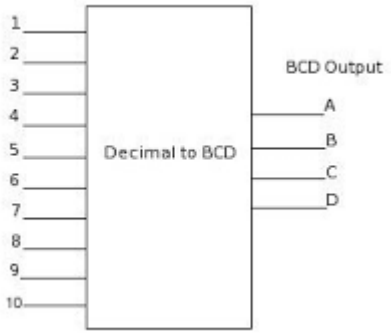
A signal that is discrete function of time, i.e. which is not a continuous signal, is known as a digital signal. The digital signals are represented in the binary form and consist of different values of voltage at discrete instants of time.



e) Explain decimal to BCD encoder.

simple decimal-to-BCD encoder is a digital circuit that has 10 input lines and 4 output lines. The inputs represent the 10 decimal numbers from 0 to 9, where only one input can be active. The outputs indicate the BCD code that represents the active input.

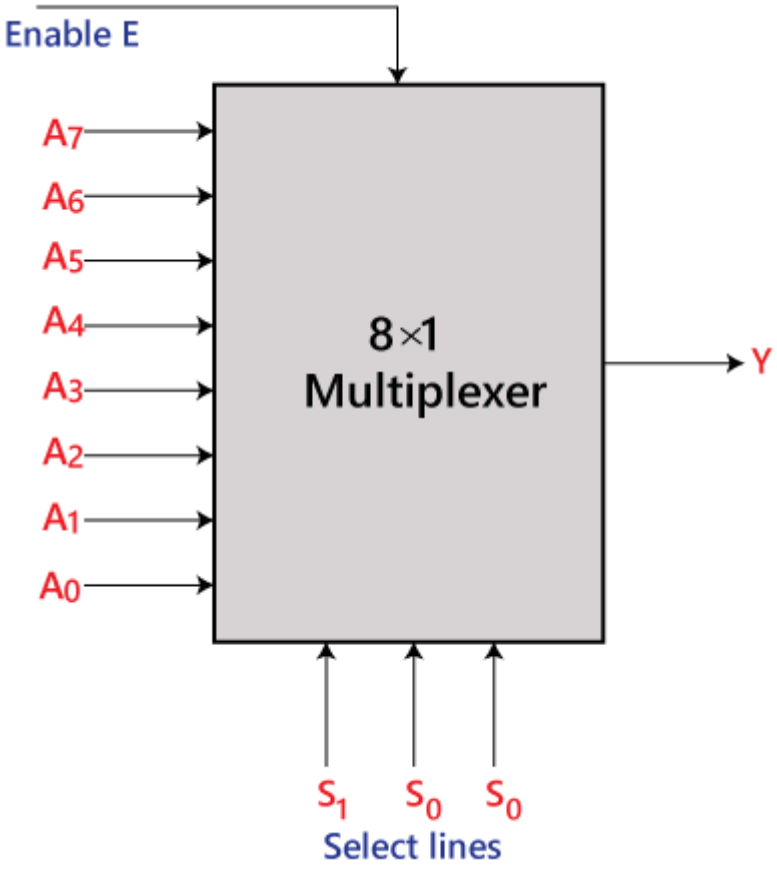




Q. 3 Attempt any three of the following (5 Marks each) **15**

a) Design 8:1 multiplexer.

In the 8 to 1 multiplexer, there are total eight inputs, i.e., A0, A1, A2, A3, A4, A5, A6, and A7, 3 selection lines, i.e., S0, S1and S2 and single output, i.e., Y. On the basis of the combination of inputs that are present at the selection lines S0, S1, and S2, one of these 8 inputs are connected to the output. The block diagram and the truth table of the 8×1 multiplexer are given below.



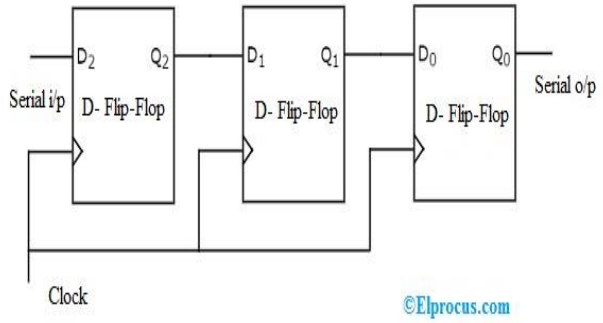
INPUTS			Output
S_2	S_1	S_0	Y
0	0	0	A_0
0	0	1	A_1
0	1	0	A_2
0	1	1	A_3
1	0	0	A_4
1	0	1	A_5
1	1	0	A_6
1	1	1	A_7

The logical expression of the term Y is as follows:

Y=S0'.S1'.S2'.A0+S0.S1'.S2'.A1+S0'.S1.S2'.A2+S0.S1.S2'.A3+S0'.S1'.S2 A4+S0.S1'.S2 A5+S0'.S1.S2 .A6+S0.S1.S3. A7

b) What is shift register? Explain SISO shift register.

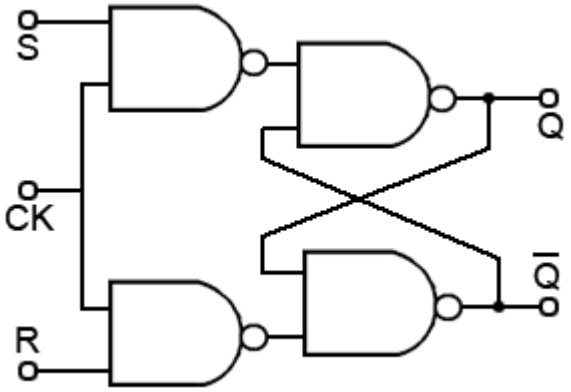
Serial In Serial Out (SISO) shift registers are a kind of shift registers where both data loading as well as data retrieval to/from the shift register occurs in serial-mode. Figure 1 shows a n-bit synchronous SISO shift register sensitive to positive edge of the clock pulse. Here the data word which is to be stored is fed bit-by-bit at the input of the first flip-flop. Further it is seen that the inputs of all other flip-flops (except the first flip-flop FF1) are driven by the outputs of the preceding ones say for example, the input of FF2 is driven by the output of FF1. At last the data stored within the register is obtained at the output pin of the nth flip-flop in serial-fashion.



Serial in – Serial out Shift Register (SISO)

c) Explain clocked SR FF.

The SR flip-flop is also named as RS flip-flop. When both the inputs of the SR flip-flop are high, then the indeterminate state is theirs. In other programming environments, it is required to assign determinate outputs to all flip-flop conditions. Hence, RS and SR flip-flops were designed. The clocked SR flip-flop is shown below.

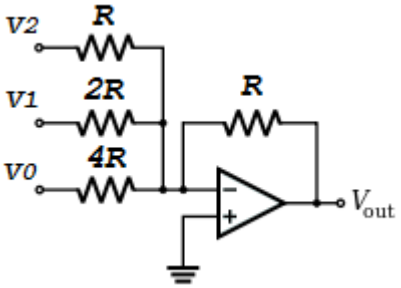


Inputs		*Output*
S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	–

1. Q output is set to logic 1 by applying logic 0 to the S input.
2. Returning the S input to logic 1 has no effect. The 0 pulse (high-low-high) has been ‘remembered’ by the Q.
3. Q is reset to 0 by logic 0 applied to the R input.
4. As R returns to logic 1 the 0 on Q is ‘remembered’ by Q.

d) Explain Digital to Analog converter.

he following diagram shows a 3 bit digital to analog converter implemented using a summing opamp amplifier.



From the summing amplifier circuit (see equation 6), the output voltage is

$$Vo = -R(V2R+V12R+V04R)\ (1)(1)\ Vo = -R(V2R+V12R+V04R)$$

Simplifying, we obtain

$$Vo = -14(4V2+2V1+V0)\ (2)(2)\ Vo = -14(4V2+2V1+V0)$$

Output Table				
V2	V1	V0	Digital Value	Vo
0	0	0	0	0
0	0	1	1	-0.25
0	1	0	2	-0.5
0	1	1	3	-0.75
1	0	0	4	-1.0
1	0	1	5	-1.25
1	1	0	6	-1.5
1	1	1	7	-1.75

From the table, we can conclude the following

- The inputs can be thought of as a binary number, one that can run from 0 to 7.
- V2 is the MSB (most significant bit) and V0 is the LSB (least significant bit).
- The output is a voltage that is proportional to the binary number input.
- The resolution of this DAC is 3 (the number of bits) or -0.25V (the step size).
- To have more bits, add an additional resistor for each additional bit. Note the relationship between adjacent resistor values.

e) Explain Hamming code in detail.

Hamming code is an error correction system that can detect and correct errors when data is stored or transmitted.

Decoding a message in Hamming Code

Once the receiver gets an incoming message, it performs recalculations to detect errors and correct them. The steps for recalculation are −

- Step 1 − Calculation of the number of redundant bits.
- Step 2 − Positioning the redundant bits.
- Step 3 − Parity checking.
- Step 4 − Error detection and correction

Step 1 − Calculation of the number of redundant bits

Using the same formula as in encoding, the number of redundant bits are ascertained.

$2r \geq m + r + 1$ where m is the number of data bits and r is the number of redundant bits.

Step 2 − Positioning the redundant bits

The r redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc.

Step 3 − Parity checking

Parity bits are calculated based upon the data bits and the redundant bits using the same rule as during generation of c1,c2 ,c3 ,c4 etc. Thus

c1 = parity (1, 3, 5, 7, 9, 11 and so on)

c2 = parity (2, 3, 6, 7, 10, 11 and so on)

c3 = parity (4-7, 12-15, 20-23 and so on)

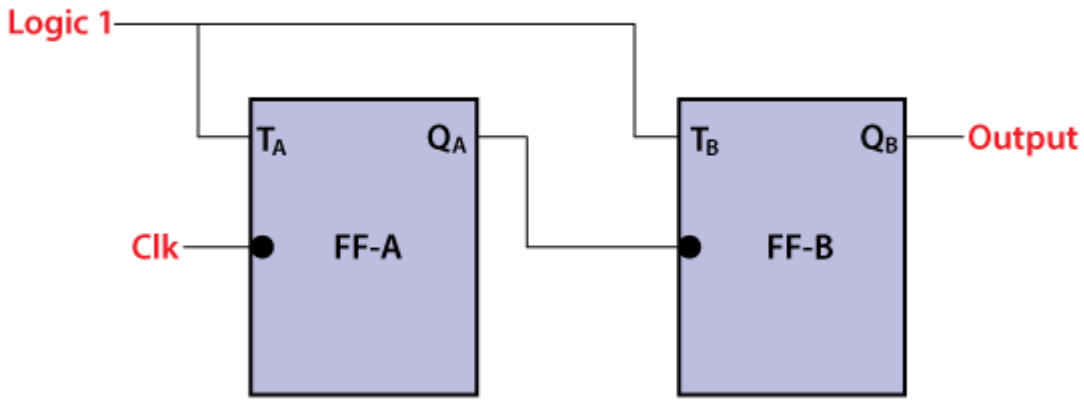
Step 4 − Error detection and correction

The decimal equivalent of the parity bits binary values is calculated. If it is 0, there is no error. Otherwise, the decimal value gives the bit position which has error. For example, if c1c2c3c4 = 1001, it implies that the data bit at position 9, decimal equivalent of 1001, has error. The bit is flipped to get the correct message.

Q. 4 Attempt any three of the following (5 Marks each) **15**

a) What is counter? Explain asynchronous counter in detail.

Definition: Asynchronous counters are those counters which do not operate on simultaneous clocking. In asynchronous counter, only the first flip-flop is externally clocked using clock pulse while the clock input for the successive flip-flops will be the output from a previous flip-flop.



he Asynchronous counter is also known as the ripple counter. Below is a diagram of the 2-bit Asynchronous counter in which we used two T flip-flops. Apart from the T flip flop, we can also use the JK flip flop by setting both of the inputs to 1 permanently. The external clock pass to the clock input of the first flip flop, i.e., FF-A and its output, i.e., is passed to clock input of the next flip flop, i.e., FF-B.

b) Solve the following.

i) (11011+01011) ii) (110-100)

i) 11011+01011 = 100110 ii) 110-100 = 010

c) Convert the following.

i) (A54)16 = (?)2 ii) (654.32)8 = (?)2

i) (A54)16 = (?)2

A= 1010

5= 0101

4= 0100

(A54)16 = (101001010100)2

ii) (654.32)8 = (?)2

6 = 110

5 = 101

4 = 100

3 = 011

2 = 010

(654.32)8 = (110101100.011010)2

d) Explain BCD to decimal decoder.

A BCD code applied to the 4 inputs, A, B, C, and D, results in a high level at the selected 1-of-10 decimal decoded outputs.

This type of decoder is probably the most widely used in all digital systems because it changes the inherent binary codes used within the system to the decimal code used by the human operators. Figure 1 illustrates the function block of a basic BCD-to-decimal decoder. Four input lines represent the decimal numbers 0 through 9. This type of decoder is often used in combination with decade counters and with decimal displays. By using only, the three least significant inputs, a 3-bit binary-to-octal decoder is obtained, with outputs only on terminals 0 through 7.

e) Minimize using K-map:

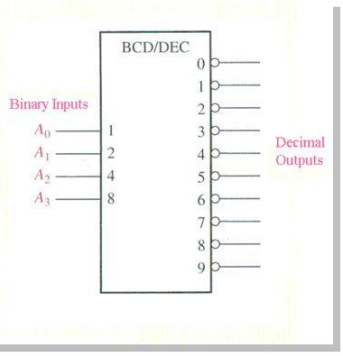
$f(A,B,C,D) = \sum m (0,1,4,5,10,11,12) + d(2,3)$

e) Minimize using K-map:

$f(A,B,C,D) = \sum m (0,1,4,5,10,11,12) + d(2,3)$

- BCD-to-decimal decoder

DECIMAL DIGIT	BCD CODE A_3	A_2	A_1	A_0	DECODING FUNCTION
0	0	0	0	0	$\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$
1	0	0	0	1	$\bar{A}_3\bar{A}_2\bar{A}_1A_0$
2	0	0	1	0	$\bar{A}_3\bar{A}_2A_1\bar{A}_0$
3	0	0	1	1	$\bar{A}_3\bar{A}_2A_1A_0$
4	0	1	0	0	$\bar{A}_3A_2\bar{A}_1\bar{A}_0$
5	0	1	0	1	$\bar{A}_3A_2\bar{A}_1A_0$
6	0	1	1	0	$\bar{A}_3A_2A_1\bar{A}_0$
7	0	1	1	1	$\bar{A}_3A_2A_1A_0$
8	1	0	0	0	$A_3\bar{A}_2\bar{A}_1\bar{A}_0$
9	1	0	0	1	$A_3\bar{A}_2\bar{A}_1A_0$



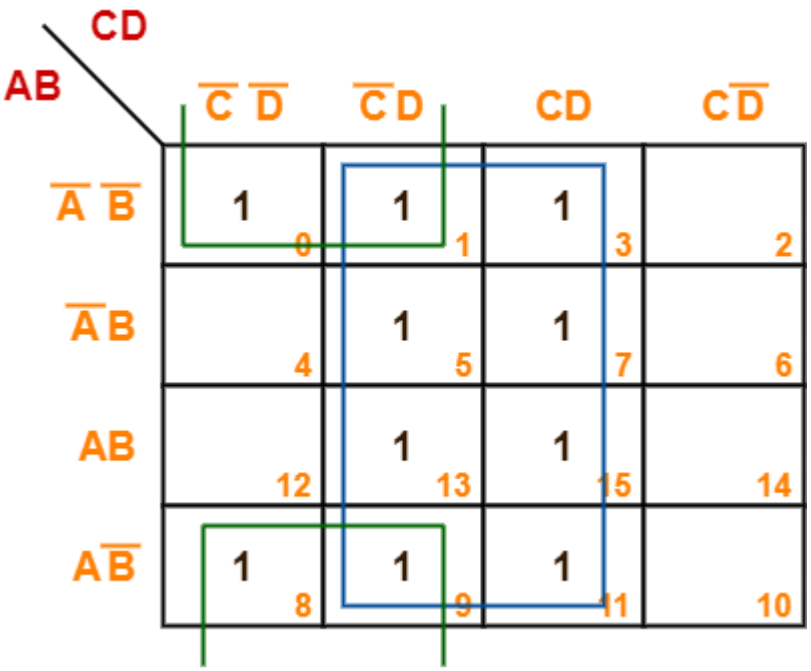
e) Minimize using K-map:

$$F(A, B, C, D) = \Sigma m(0, 1, 3, 5, 7, 8, 9, 11, 13, 15)$$

Solution-

- Since the given Boolean expression has 4 variables, so we draw a 4 x 4 K Map.
- We fill the cells of K Map in accordance with the given Boolean function.
- Then, we form the groups in accordance with the above rules.

Then, we have-



Now,

F (A, B, C, D)

= (A'B' + A'B + AB + AB') (C'D + CD) + (A'B' + AB') (C'D' + C'D)

Q .5 Short notes on any three of the following (5 Marks each) 15

a) BCD code

A binary-coded decimal (BCD) is a type of binary representation for decimal values where each digit is represented by a fixed number of binary bits, usually between four and eight.

The norm is four bits, which effectively represent decimal values 0 to 9. This writing format system is used because there is no limit to the size of a number. Four bits can simply be added as another decimal digit, versus real binary representation

Decimal	Binay (BCD) 8 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

b) Two's complement.

There is a simple algorithm to convert a binary number into 2's complement. To get 2's complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given number

Example-1 − Find 2's complement of binary number 10101110.

Simply invert each bit of given binary number, which will be 01010001. Then add 1 to the LSB of this result, i.e., 01010001+1=01010010 which is answer.

c) Parity code

Parity Bits

The parity check is done by adding an extra bit, called parity bit, to the data to make the number of 1s either even or odd depending upon the type of parity. The parity check is suitable for single bit error detection only.

The two types of parity checking are

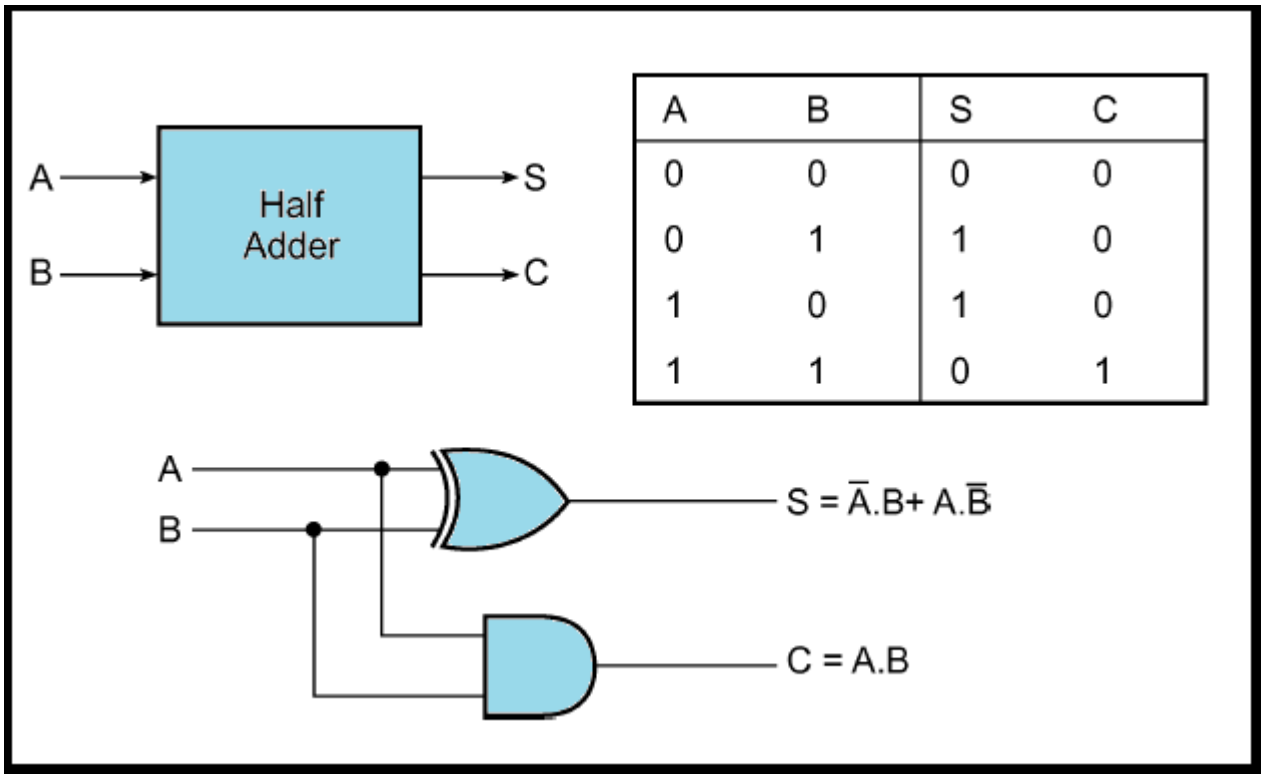
- Even Parity − Here the total number of bits in the message is made even.
- Odd Parity − Here the total number of bits in the message is made odd.

Original Data	Even Parity	Odd Parity
0 0 0 0 0 0 0 0	0	1
0 1 0 1 1 0 1 1	1	0
0 1 0 1 0 1 0 1	0	1
1 1 1 1 1 1 1 1	0	1
1 0 0 0 0 0 0 0	1	0
0 1 0 0 1 0 0 1	1	0

d) half adder.

A Half-adder circuit needs two binary inputs and two binary outputs. The input variable shows the augend and addend bits whereas the output variable produces the sum and carry. We can understand the function of a half-adder by formulating a truth table. The truth table for a half-adder is:

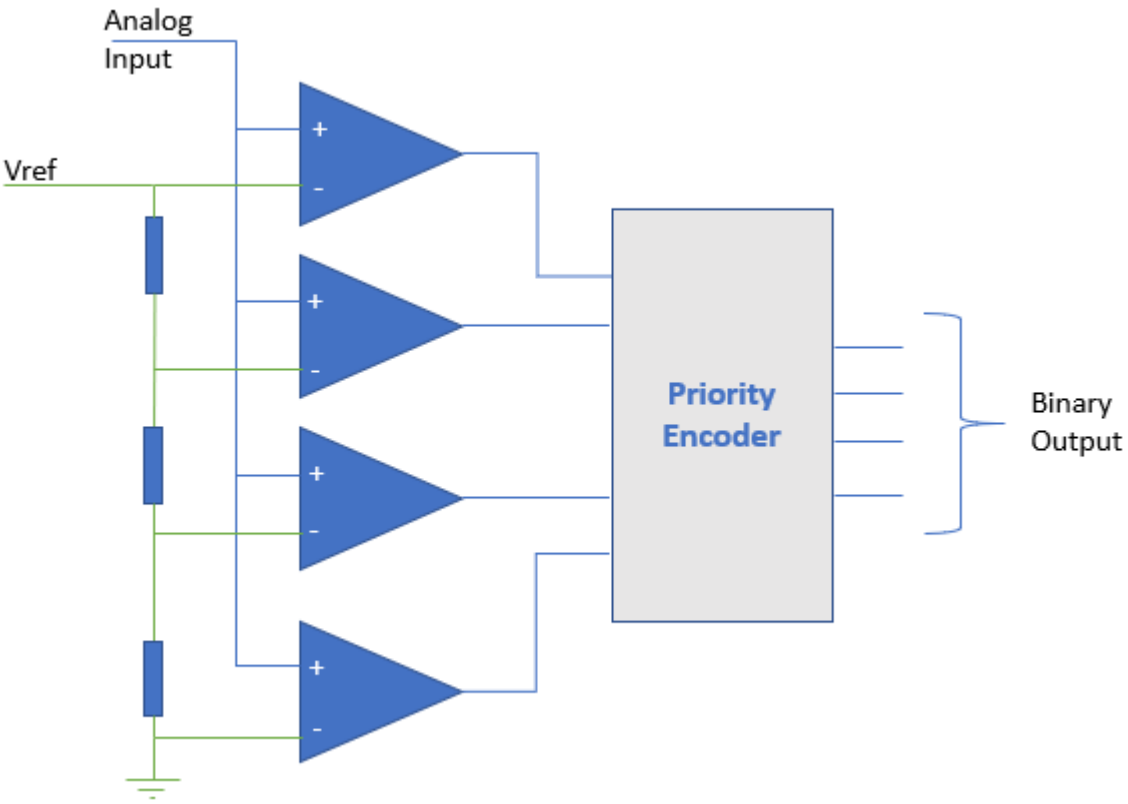
.



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

e)Analog to digital converter.

From the name itself it is clear that it is a converter which converts the analog (continuously variable) signal to digital signal. This is really an electronic integrated circuit which directly converts the continuous form of signal to discrete form. It can be expressed as A/D or A-to-D or A-D or ADC. The input (analog) to this system can have any value in a range and are directly measured. But for output (digital) of an N-bit A/D converter, it should have only 2N discrete values. This **A/D converter** is a linkage between the analog (linear) world of transducers and discreet world of processing the signal and handling the data. The digital to analog converter (DAC) carry out the inverse function of the ADC. The schematic representation of ADC is shown below.


Analog
Input
Vref
+
-
+
-
+
-
+
-
Priority
Encoder
Binary
Output