DIGITAL LOGIC DESIGN AND ANALYSIS (DLDA)

NOVEMBER--2018

Q1 (a) Convert decimal number 576.24 into binary, base-9, octal, hexadecimal system. **(04)**

Solution:
(i)Conversion to Binary

2	576	
2	288	0
2	144	0
2	72	0
2	36	0
2	18	0
2	9	0
2	4	1
2	2	0
2	1	0
	0	1

The binary equivalent of 576 is 1001000000.

0.24 x 2 = 0.48 ~ 0
0.48 x 2 = 0.96 ~ 0
0.96 x 2 = 1.92 ~ 1-
0.92 x 2 = 1.84 ~ 1
0.84 x 2 = 1.68 ~ 1

$(576.24)_{10} = (1001000000.00111)_{2}$

(ii)Conversion to Base-9

9	576	
9	64	0
9	7	1
	0	7

The base-9 equivalent of 576 is (710).

0.24 x 9 = 2.16 ~ 2
0.16 x 9 = 1.44 ~ 1
0.44 x 9 = 3.96 ~ 3
0.96 x 9 = 8.64 ~ 8

$(576.24)_{10}=(710.2138)_{9}$

(iii)Conversion to Octal

8	576	
8	72	0
8	9	0
8	1	1
	0	1

The octal equivalent of 576 is (1100).

0.24 x 8 = 1.92 ~ 1
0.92 x 8 = 7.36 ~ 7
0.36 x 8 = 2.88 ~ 2
0.88 x 8 = 7.04 ~ 7

$(576.24)_{10}= (1100.1727)_{8}$

(iv)Conversion to Hexadecimal

16	576	
16	36	0
16	2	4
	0	2

The Hexadecimal equivalent of 576 is (240).

0.24 x 16 = 3.84 ~ 3
0.84 x 16 = 13.44 ~ D
0.44 x 16 = 7.04 ~ 7
0.04 x 16 = 0.64 ~ 0

$(576.24)_{10} = (240.3D70)_{16}$

(b) Construct hamming code for 1010 using odd parity. (04)

Solution:

The given code is 1010 i.e. 4-bits.

∴ 3 Parity bits are required.

1	2	3	4	5	6	7
P1	P2	1	P3	0	1	0

Using Odd - parity,
P1 = (3, 5, 7) = (1, 0, 0) = 0
P2 = (3, 6, 7) = (1, 1, 0) = 1
P3 = (5, 6, 7) = (0, 1, 0) = 0
∴ Hamming code is as follows:

1	2	3	4	5	6	7
0	1	1	0	0	1	0

The Hamming code for the given code is 1010 is 0110010.

(c)Convert $(-89)_{10}$ to its equivalent sign magnitude, 1's Complement and 2's Complement Form. **(04)**

Solution:

The sign magnitude equivalent of $(-89)_{10}$ is as follows:

Sign bit

1	1	0	1	1	0	0	1

MSB magnitude

$(-89)_{10} = (11011001)_2$

The binary equivalent of $(-89)_{10}$ is $(11011001)_2$.
So, 1`s complement of $(-89)_{10}$ is $(00100110)_2$

2's complement of $(-89)_{10}$ is :

```
    0   0   1   0   0   1   1   0
+                               1
_____________________________________
    0   0   1   0   0   1   1   1
```

So, 2's complement of $(-89)_{10}$ is $(00100111)_2$.

(d)Perform $(BC5)_H$ - $(A2B)_H$ without converting to any other base. **(04)**

Solution:

```
    B   ¹²C   ¹⁵5
-   A    2    B
_____________________
```

1 9 A

Here, as 5 < B, so we borrow 16 from the previous value as the given values are in Hexadecimal format. 16 is then added to 5 ,thus it becomes 21 and 21 is subtracted from B. So, we get answer as A.

Since, we have borrowed from the previous value i.e C ,the value of C decreases by 1.So, it becomes B.

Thus, $(BC5)_H - (A2B)_H = (19A)_H$

(e) Prove De Morgan's theorem. **(04)**

Solution:

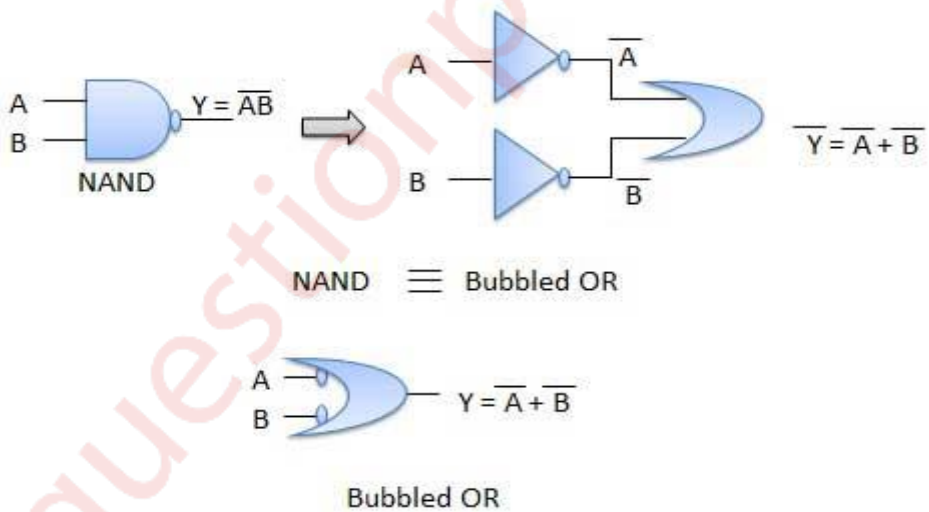
The two theorems suggested by De-Morgan and which are extremely useful in Boolean algebra are stated as follows:

Theorem 1: $\overline{AB} = \overline{A} + \overline{B}$

NAND = Bubbled OR:

This theorem states that the complement of a product is equal to the sum of individual complements.

This theorem can be verified using truth table as follows:



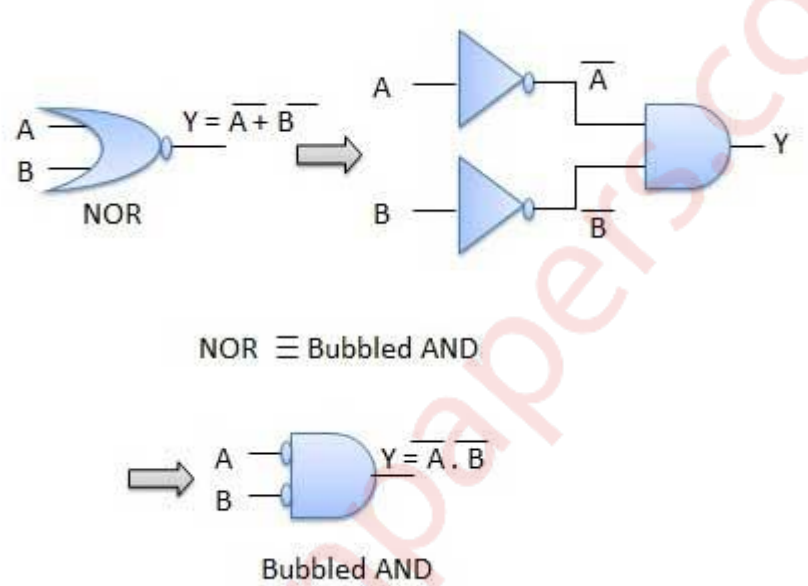
A	B	$\overline{AB}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

Thus we can see that $\overline{AB} = \overline{A} + \overline{B}$.

Theorem 2 : $A + B = A \cdot B$

NOR = Bubbled AND:

This theorem states that the complement of a sum is equal to the product of individual complements.



This theorem can be verified using truth table as follows:

A	B	$\overline{A + B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

Thus we can say that $\overline{A + B} = \overline{A} \cdot \overline{B}$.

Q.2(a)Given the logic expression $A + \overline{B}\ \overline{C} + A B \overline{D} + A B C D$ (10)

1. **Express it in standard SOP form.**
2. **Draw K-map and simplify**
3. **Draw logic diagram using NOR gates only.**

Solution:

The given expression is :

$A + \overline{B}\ \overline{C} + A B \overline{D} + A B C D$

(i)Expressing the given expression in standard SOP form:

$$A + \overline{B}\ \overline{C} + A B \overline{D} + A B C D$$

$\therefore A (B + \overline{B}) (C + \overline{C}) (D + \overline{D}) + \overline{B}\ \overline{C} (A + \overline{A}) (D + \overline{D}) + A B \overline{D} (C + \overline{C}) + A B C D$

$\therefore (A B + A \overline{B})(C D + \overline{C} D + C \overline{D} + \overline{C}\ \overline{D}) + \overline{B}\ \overline{C} (A D + \overline{A} D + A \overline{D} + \overline{A}\ \overline{D}) + A B C \overline{D} + A B \overline{C}\ \overline{D} + A B C D$

$\therefore A B C D + A B \overline{C} D + A B C \overline{D} + A B \overline{C}\ \overline{D} + A \overline{B} C D + A \overline{B}\ \overline{C} D + A \overline{B} C \overline{D} + A \overline{B}\ \overline{C}\ \overline{D} +$

$+A \overline{B}\ \overline{C} D + \overline{A}\ \overline{B}\ \overline{C} D + A \overline{B}\ \overline{C}\ \overline{D} + \overline{A}\ \overline{B}\ \overline{C}\ \overline{D} + A B C \overline{D} + A B \overline{C}\ \overline{D} + A B C D$

The standard SOP form of the given expression is :

$\therefore \overline{A}\ \overline{B}\ \overline{C}\ \overline{D} + \overline{A}\ \overline{B}\ \overline{C} D + A \overline{B}\ \overline{C}\ \overline{D} + A \overline{B}\ \overline{C} D + A \overline{B} C \overline{D} + A \overline{B} C D + A B \overline{C}\ \overline{D} + A B \overline{C} D +$

$A B C \overline{D} + A B C D$

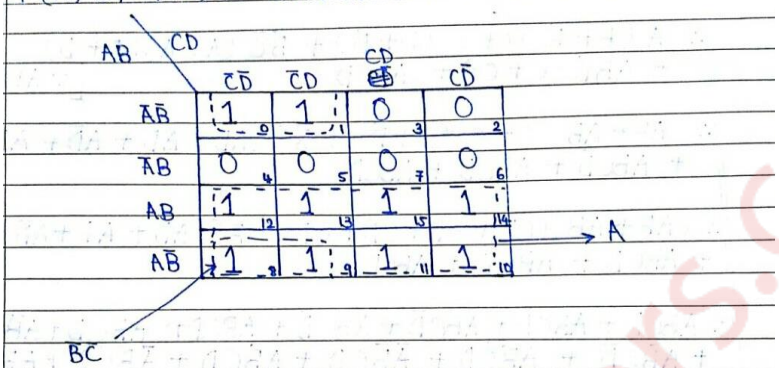
(ii)Using K-map to simplify the given expression :

From standard SOP form we get,
$F(A, B, C, D) = \Sigma m (0, 1, 8, 9, 10, 11, 12, 13, 14, 15)$

2. Using K-map to simplify the given expression.

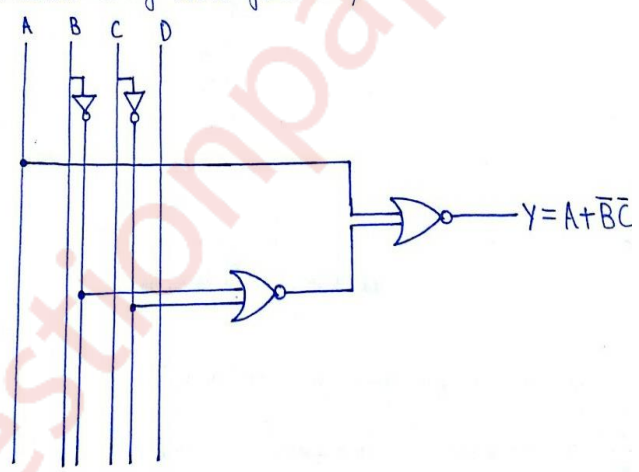
From standard SOP form we get

$F(A, B, C, D) = \Sigma m(0, 1, 8, 9, 10, 11, 12, 13, 14, 15)$



The simplified expression is $Y = A + \bar{B}\bar{C}$

3. Implementation using NOR gates only



Q.2(b)Reduce using Quine McCluskey method and realize the operation using only NAND gates. **(10)**

$F(A, B, C, D) = \pi M(0, 2, 3, 6, 7, 8, 9, 12, 13)$

Solution:

$F(A, B, C, D) = \pi M(0, 2, 3, 6, 7, 8, 9, 12, 13)$

$F(A, B, C, D) = \Sigma m(1, 4, 5, 10, 11, 14, 15)$

Step 1: Grouping the minterms according to the number of 1's

Group	Minterm	Binary representation			
		A	B	C	D
1	1	0	0	0	1
	4	0	1	0	0
2	5	0	1	0	1
	10	1	0	1	0
3	11	1	0	1	1
	14	1	1	1	0
4	15	1	1	1	1

Step 2: Grouping the minterms that form pairs

Group	Minterm	Binary representation			
		A	B	C	D
1	1-5	0	-	0	1
	4-5	0	1	0	-
2	10-11	1	0	1	-
	10-14	1	-	1	0
3	11-15	1	-	1	1
	14-15	1	1	1	-

Step 3: Grouping the minterms to form quad

Group	Minterm	Binary representation			
		A	B	C	D
1	10-11-14-15	1	-	1	-
	10-14-11-15	1	-	1	-

Step 4: Collecting all Prime Implicants

From Step 2, the prime implicants are $\bar{A}\,\bar{C}\,D$ and $\bar{A}\,B\,\bar{C}$.
From Step 3, we get the prime implicants as A C.

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

Step 5: Preparing the Prime Implicants Table

Prime Implicant (PI)	Decimal numbers (minterms)	Given Minterms
		1 4 5 10 11 14 15
$\bar{A}\,\bar{C}\,D$	1,5	X X
$\bar{A}\,B\,\bar{C}$	4,5	X X
A C	10,11,14,15	X X X X

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

Q.3(a)Design a 4-bit binary to gray code converter. (10)

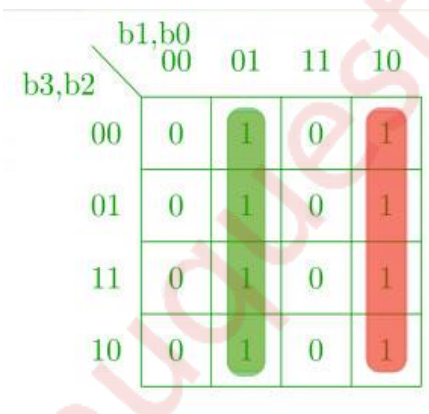
Solution:

The truth table showing binary inputs being converted to gray outputs is as follows:

BINARY INPUT				GRAY CODE OUTPUT			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

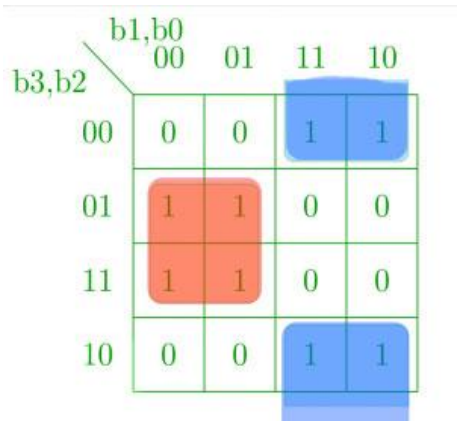
The K-maps for each gray output is as shown below.

K-map for g_0:



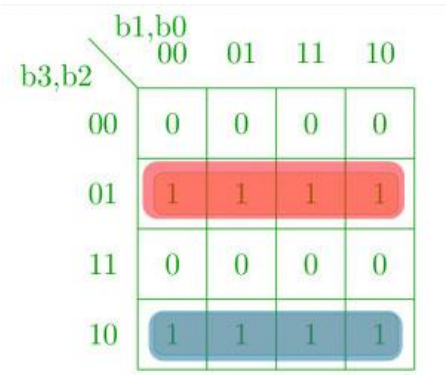
$\therefore g_0 = \overline{b1}\, b0 + b1\, \overline{b0}$

K-map for g_1:

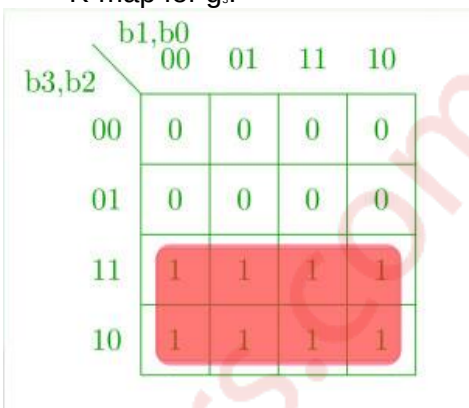


$\therefore g_1 = \overline{b2}\, b3 + b2\, \overline{b3}$

K-map for g_2:



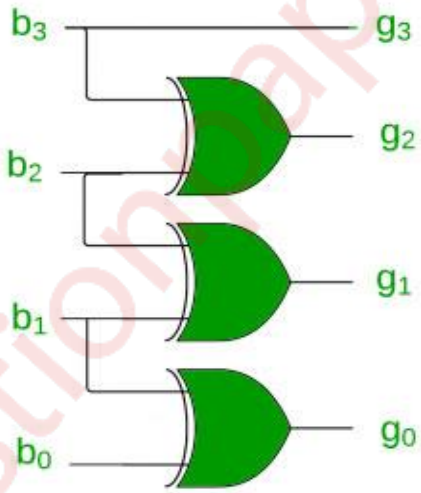
K-map for g_3:



$\therefore g_2 = \overline{b1}\, b2 + b1\, \overline{b2}$

$\therefore g_3 = b3$

The logic diagram of 4-bit Binary to Gray code converter is as shown below.



Q.3(b)Design a 4 bit BCD adder using IC 7483 and necessary gates. (10)

Solution:

1. A BCD adder adds two BCD digits and produces a BCD digit. BCD number cannot be greater than 9.
2. The two given BCD numbers are to be added using the rules of binary addition.
3. If sum is less than or equal to 9 and carry=0 then correction is necessary. The sum is correct and in the true BCD form.
4. But if sum is invalid BCD or carry=1, then the result is wrong and needs correction.
5. The wrong result can be corrected by adding six (0110) to it.
6. The 4 bit binary adder IC 7483 can be used to perform addition of BCD numbers.

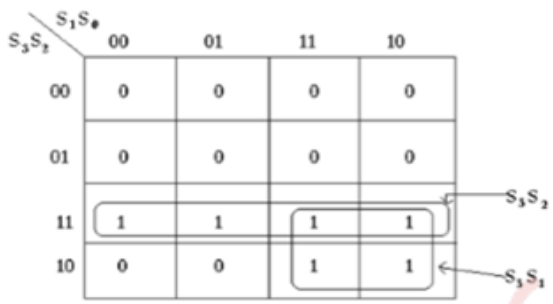
7. In this, if the four-bit sum output is not a valid digit, or if a carry C3 is generated then decimal 6 (0110 binary) is to be added to the sum to get the correct result.
8. Fig1 shows a 1-digit BCD adders can be cascaded to add numbers several digits long by connecting the carry-out of a stage to the carry-in of the next stage.
9. The output of combinational circuit should be 1 if the sum produced by adder 1 is greater than 9 i.e. 1001. The truth table is as follows:

I/P				O/P
S3	S2	S1	S0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Truth table for BCD numbers

Y=1 since the sum obtained is an invalid BCD number.

K-map:



The Boolean expression is
$Y = S_3S_2 + S_3S_1$

- The BCD adder is shown below. The output of the combinational circuit should be 1 if Cout of adder-1 is high. Therefore Y is 0 with Cout of adder 1.
- The output of combinational circuit is connected to B_1B_2 inputs of adder-2 and $B_3 = B_1 + 0$ as they are connected to ground permanently. This makes $B_3B_2B_1B_0 = 0110$ if Y' = 1.
- The sum outputs of adder-1 are applied to $A_3A_2A_1A_0$ of adder-2. The output of combinational circuit is to be used as final output carry and the carry output of adder-2 is to be ignored.

BLOCK DIAGRAM OF BCD ADDER

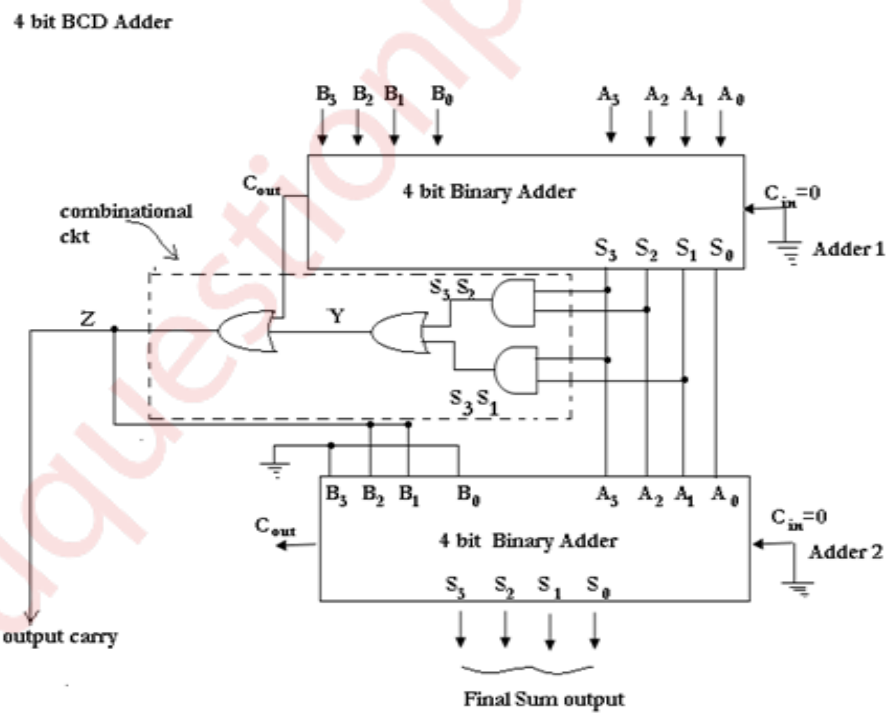


Fig5: 4-bit BCD adder

Operation of BCD Adder is as follows:
Case1: Sum ≤ 9 and carry = 0

- The output of combinational circuit Y' = 0. Hence $B_3B_2B_1B_0$ = 0 0 0 0 for adder-2.

- Hence output of adder-2 is same as that of adder-2

Case 2: Sum >9 and carry = 0

- If S3S2S1S0 of adder -1 is greater than 9, then output Y' of combinational circuits becomes 1.
- Therefore $B_3B_2B_1B_0$ = 0 1 1 0 (of adder-2).
- Hence six (0 1 1 0) will be added to the sum output of adder-1.
- We get the corrected BCD result at the output of adder-2.

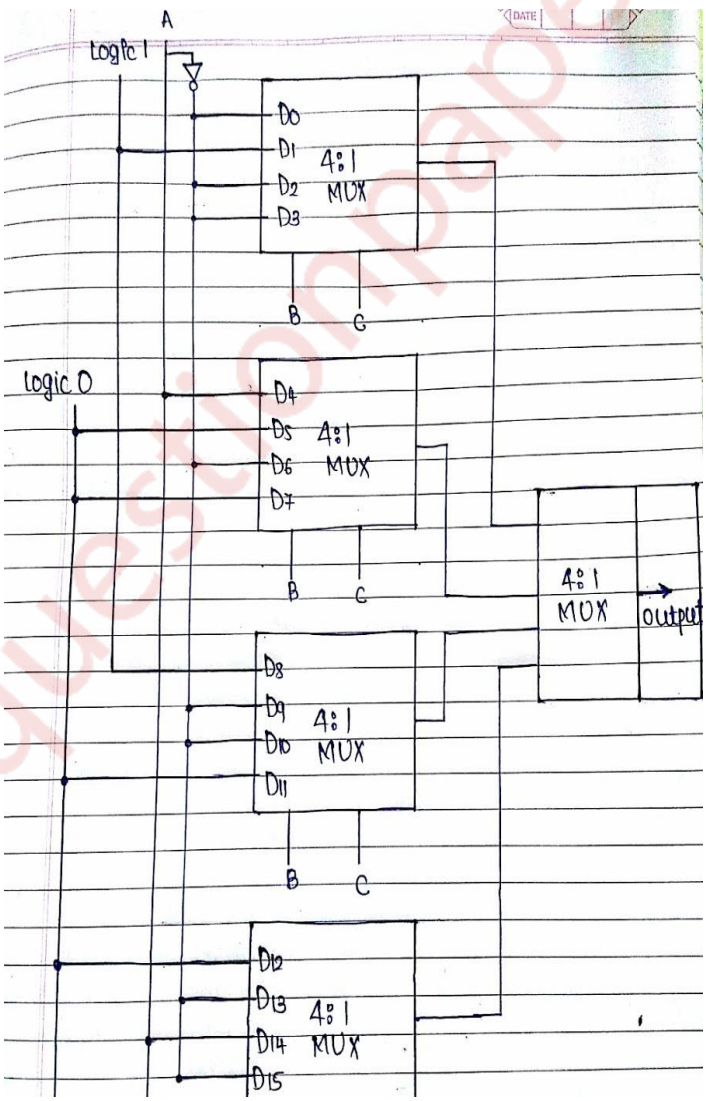
Case 3: Sum ≤ 9 but carry = 1

- As carry output of addere-1 is high, Y' = 1.
- Therefore $B_3B_2B_1B_0$ = 0 1 1 0 (of adder-2).
- Hence six (0 1 1 0) will be added to the sum output of adder-1.
- We get the corrected BCD result at the output of adder-2. Thus the Four bit BCD addition can be carried out using the binary adder.

Q.4(a)Implement the following logic function using all 4 : 1 multiplexers with the select inputs as 'B', 'C', 'D', 'E' only.
F(A, B, C, D, E) = Σ(0,1,2,3,6,8,9,10,13,15,17,20,24,30) (10)

Solution:

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
$\overline{A}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	$\overline{A}$	1	$\overline{A}$	$\overline{A}$	A	0	$\overline{A}$	0	1	$\overline{A}$	$\overline{A}$	0	0	$\overline{A}$	A	$\overline{A}$



Q.4(b)Convert a SR flip flop to JK flip flop. **(10)**

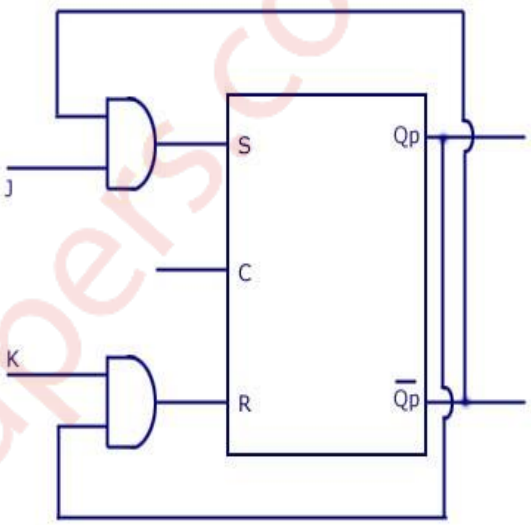
Solution:

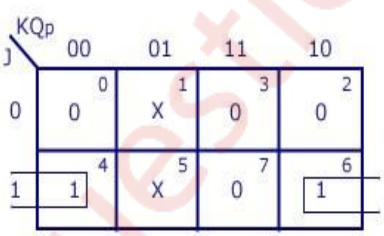
S-R Flip Flop to J-K Flip Flop

Conversion Table

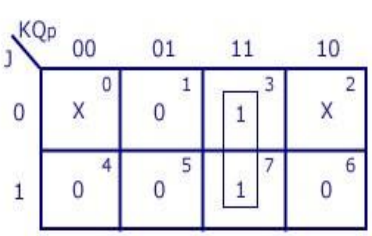
J-K Inputs		Outputs		S-R Inputs	
J	K	Qp	Qp+1	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Logic Diagram





$S = \overline{J}Q_p$

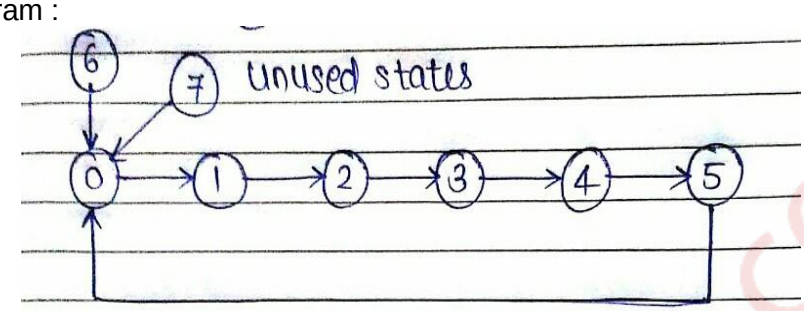


$R = KQ_p$

K-Map

Q.5(a)Design a mod-6 synchronous counter using T flip flop. (10)

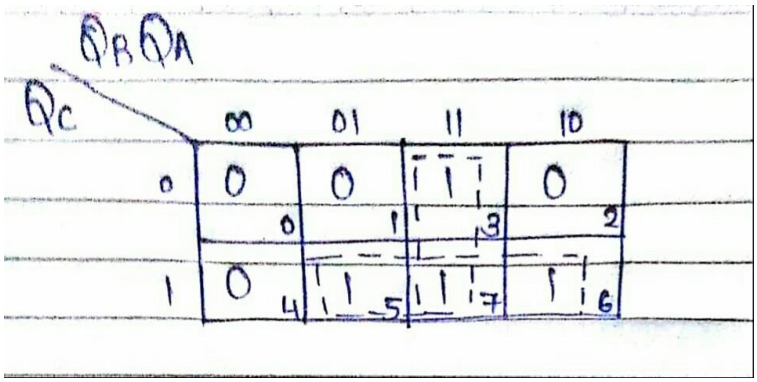
Solution:
(i)State Diagram :



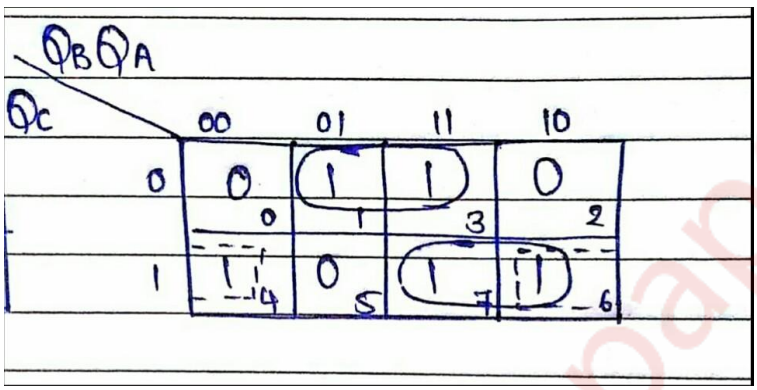
(ii)State Table :

Present State			Next State			Flip-flop inputs		
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_c	T_B	T_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	1	1
1	0	1	0	0	0	1	0	1
1	1	0	0	0	0	1	1	0
1	1	1	0	0	0	1	1	1

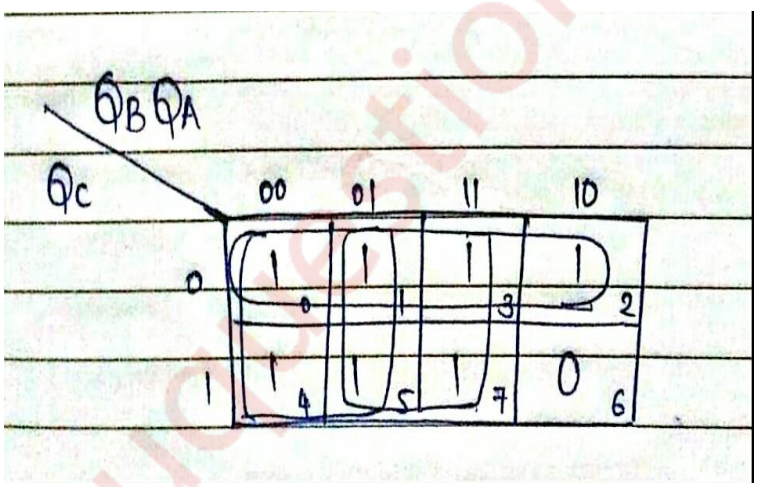
(iii)K map and simplification :



$T_C = Q_A Q_B + Q_B Q_C + Q_A Q_C$

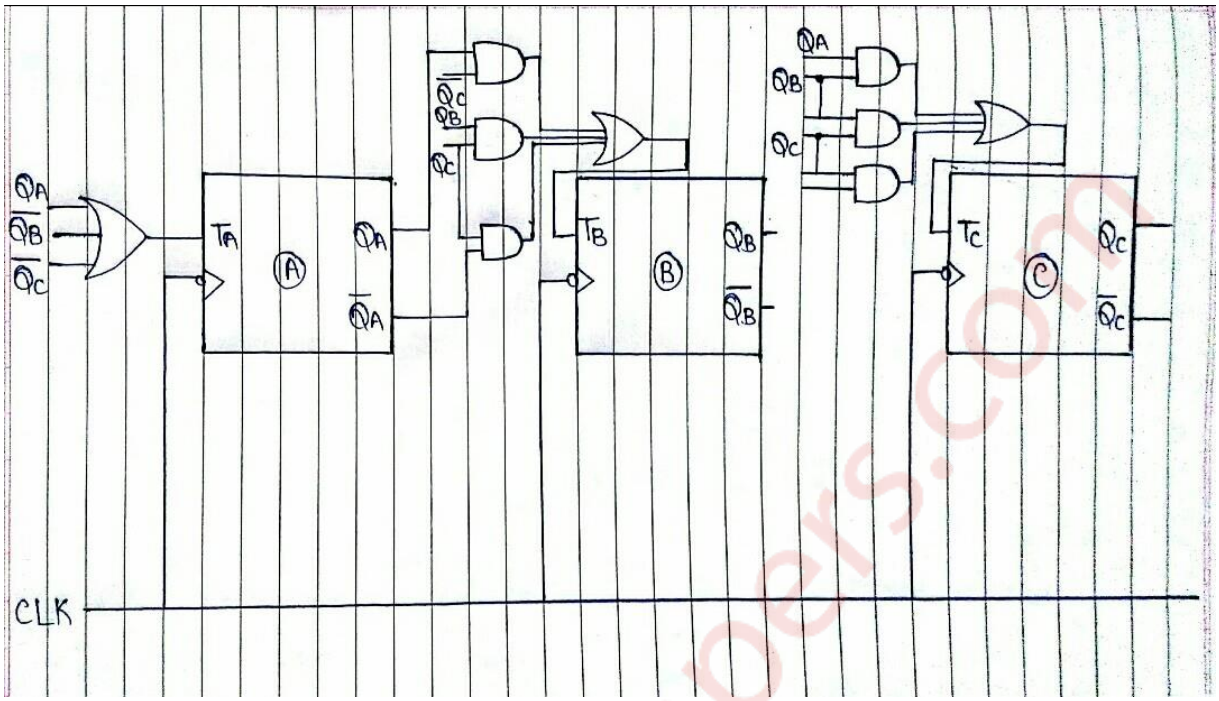


$T_B = Q_A \overline{Q_C} + Q_B Q_C + Q_C \overline{Q_A}$



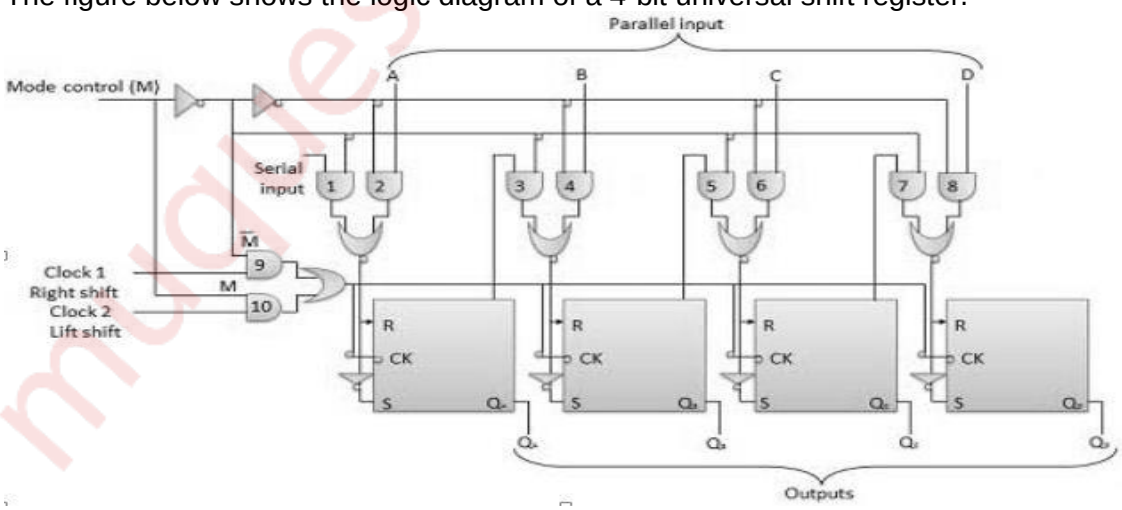
$T_A = \overline{Q_C} + \overline{Q_B} + Q_A$

(iv)Logic Diagram:



Q.5(b)Explain the operation of a 4-bit universal shift register. (10)

Solution:
A shift register which can shift the data in only one direction is called as a unidirectional shift register. A shift register which can shift the data in both directions is called as bi-directional shift register. Similarly, a shift register which can shift data in both directions i.e shift left or right as well as load it parallelly, is called as a universal shift register.
The figure below shows the logic diagram of a 4-bit universal shift register.



This shift register is capable of performing the following operations:
1.Parallel loading (parallel input parallel output)
2.Left shifting

3.Right shifting

The mode control input is connected to Logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting.When mode pin is connected to ground, the universal shift register acts as a bi-directional register.For serial left operation, the input is applied to serial input which goes into AND gate-1 of the above figure.For serial right operation, the serial input is applied to D input (input of AND gate-8).The well known example of universal shift register is in the form of IC 7495.

Universal Shift Register IC 7495:

It is a 4 bit shift register with serial and parallel synchronous operating modes.Because of its capability to operate in all possible modes, it is called a universal shift register.

Some features of this chip are:

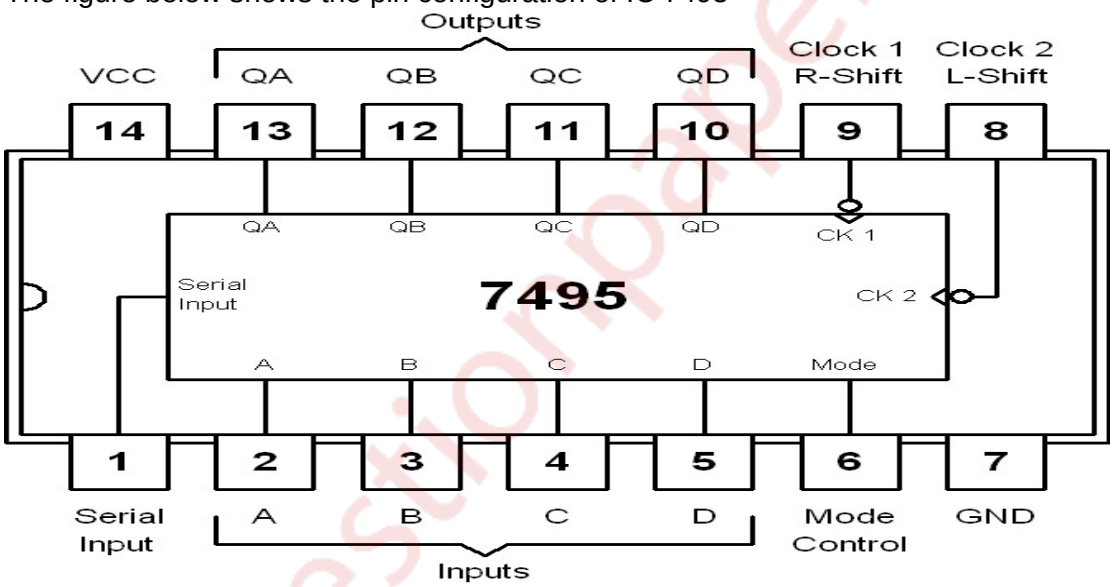
1.Synchronous shift left capacity.

2.Synchronous parallel loading is possible.

3.It has separate clock inputs,one for shift operation and the other for load operation.

4.The cascading of two or more 7495 ICs for more than 4-bits is possible.

The figure below shows the pin configuration of IC 7495



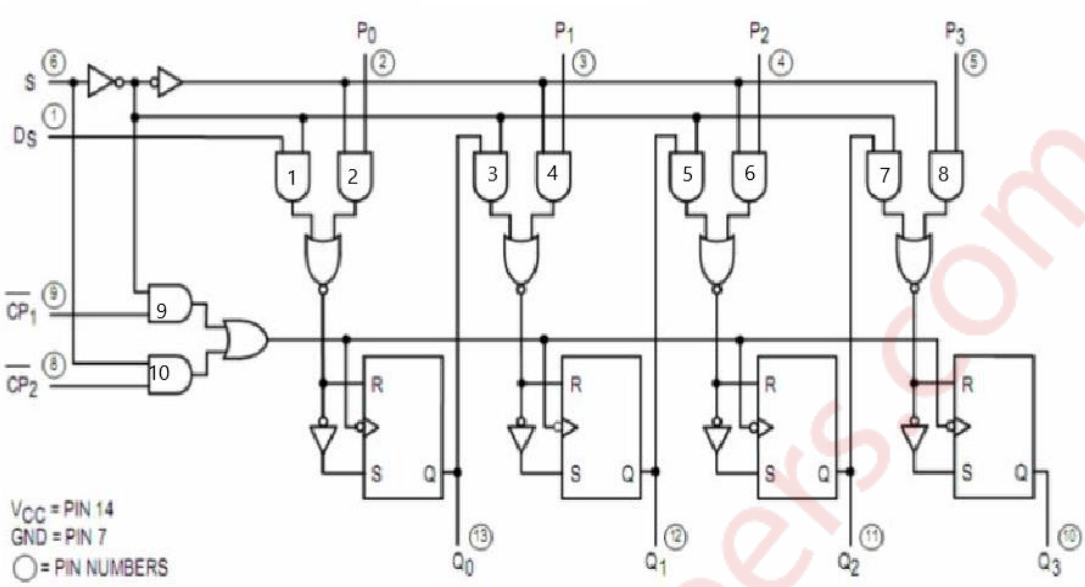
A,B,C,D are the inputs to four internal flip flops with A acting as LSB and D as MSB. QA through QD are the corresponding outputs.

IC 7495 is capable of performing the following operations:

1.Parallel loading (parallel input parallel output)

2.Left shifting

3.Right shifting

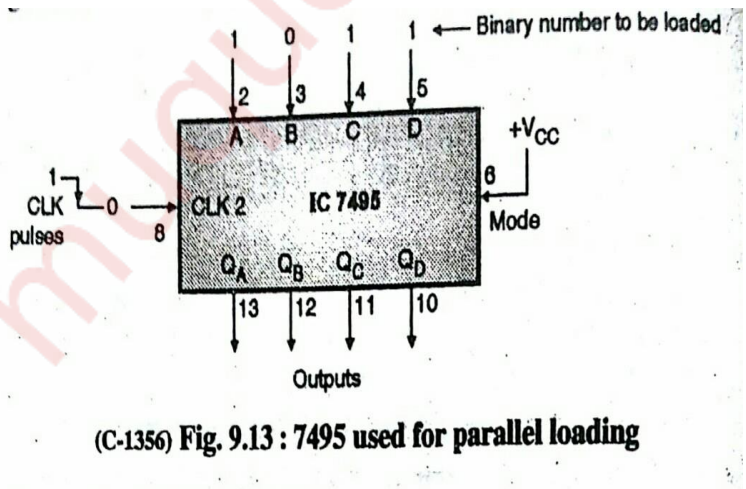


INTERNAL LOGIC DIAGRAM OF IC 7495

1.Parallel loading:

The connection diagram for IC 7495 in parallel input parallel output mode is as shown in the figure below.The mode control (M) is connected to logic 1. This will enable the AND gates 2, 4, 6, 8 as shown in internal logic diagram. The AND gates 1, 3, 5, 7 are disabled.This allows data transfer from the inputs A, B, C, D to the flip flops and disable
s the serial transfer of data. The 4-bit binary number which is to be loaded parallely is applied to the A, B, C, D inputs.

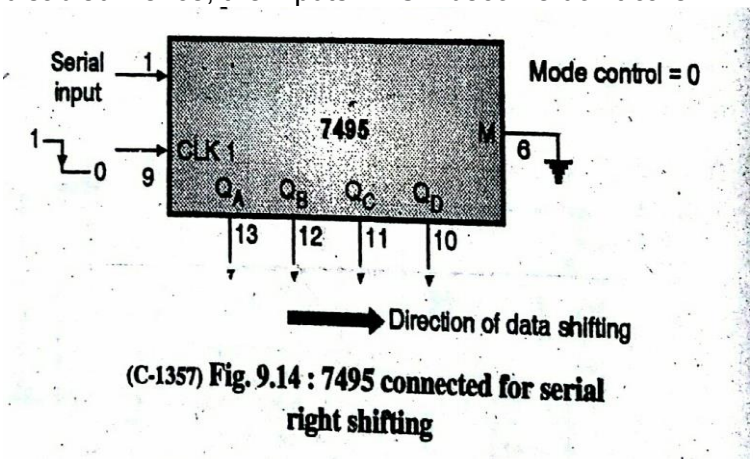
The clock applied at clock-2 input only will be paused through the flip flops because with M=1 and AND gate-10 is enabled and gate-9 is disabled. As soon as a falling edge of clock is applied, all flip-flops will change their status simultaneously and binary number applied to ABCD inputs will be loaded into the shift register. The unused inputs such as serial input and clock-1 can be left open or connected to ground because they are the don't care options for this mode.



(C-1356) Fig. 9.13 : 7495 used for parallel loading

2.Serial Shift Right Operation:

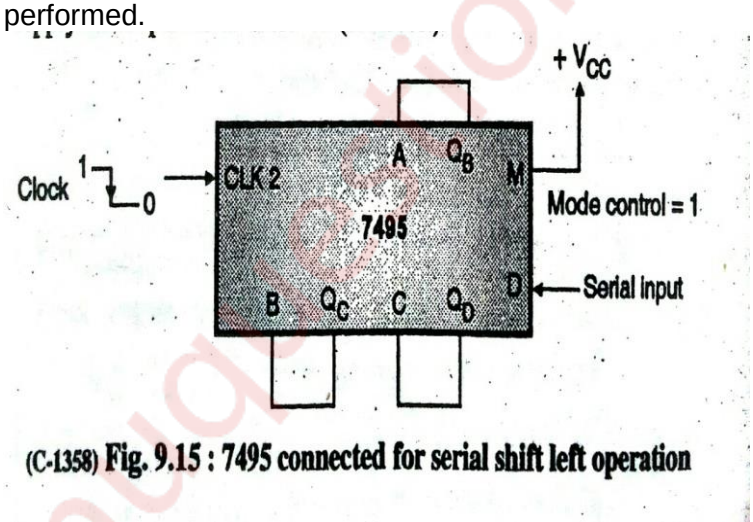
The connection for serial shift right mode is as shown in the figure below. Make mode control = 0, therefore AND gates 1, 3, 5, 7 will be enabled and AND gate 2, 4, 6, 8 will get disabled.Hence, the inputs ABCD become don't care.



(C-1357) **Fig. 9.14 : 7495 connected for serial right shifting**

3.Serial Shift Left Operation:

The connection diagram of 7495 for the shift left operation is as shown in the figure. Q_D is connected to C, Q_C to B and Q_B to A and the serial data is applied at input D.Mode control is connected to 1. Hence, the AND gates 2, 4, 6, 8 are enabled whereas 1, 3, 5, 7 are disabled. This will make the serial input (pin no. 1) a don't care input. The serial data is applied to D which will be routed through the enabled AND gates 2, 4, 6, 8 to facilitate the right shifting operation. As M = 1, AND gate - 10 is enabled and gate - 9 is disabled. So clock - 1 becomes a don't care input. Apply clock pulses to CLK - 2(shift left).Each high to low transition of clock will transfer data from D to Q_D, Q_D to Q_C , Q_C to Q_B ,Q_B to Q_A.Thus the shift left operation is performed.



(C-1358) **Fig. 9.15 : 7495 connected for serial shift left operation**

Q.6 Write Short notes on (any 2) **(20)**

(a)VHDL

Solution:

The long form of VHDL is Very High Speed Integrated Circuit (VHSIC) hardware description language.

VHDL is used to form a digital system at many levels of ideas ranging from algorithmic level to the gate level.

This language defines the syntax as well as simulation semantics for each language. It is a strong typed language which contains too many words to write.

VHDL is difficult to understand because it provides wide ranging of modelling capabilities but without learning the more complex features it is possible to incorporate a core subset of language which is simple and easy to understand.

Some Features of VHDL are:

1. Strongly typed language: Only LHS and RHS operators of the same type are allowed in VHDL.
2. Support hierarchies: Using VHDL, hierarchy can be represented. For example, full adder.In this case it is composed of half adder and OR gate.
3. VHDL supports for test and simulation of programs.
4. Concurrency: VHDL is a concurrent language which executes statements simultaneously in parallel.
5. VHDL supports different types of data modelling
   (i)Structural
   (ii)Data flow
   (iii)Behavioural
   (iv)Mixed

6. Supports sequential statement: VHDL can execute only one statement at a time in sequence only.
7. VHDL supports synchronous and asynchronous models.
8. VHDL can be used as a communication model between different CAD and CAE models.

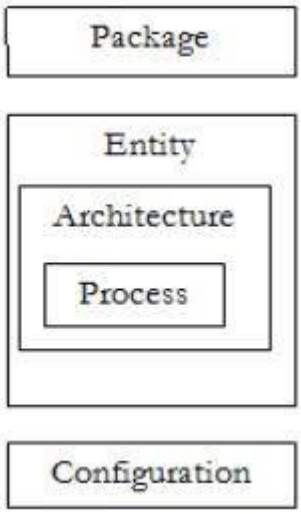
Structure of VHDL module:

Design units of VHDL code are independent components which are separately combined and stored in the library.

VHDL program is composed of the following design units:

1. Package (optional)
2. Entity
3. Architecture
4. Configuration (optional)

The diagram below shows the design units of VHDL.



Advantages of VHDL

1. VHDL allows designers to quickly develop designs requiring tens of thousands of logic gates.
2. VHDL supports multiple level of hierarchy and modular design methods.
3. VHDL allows user to pick any synthesis tool.
4. VHDL is multipurpose i.e. once calculation block is created then it can be used in many other projects.
5. For describing complex logic, VHDL provides powerful high level constructs.

Disadvantages of VHDL

VHDL is not a low level language i.e. gate level program. It is not suitable for verification of basic objects like gates. Because in VHDL these objects are readily available.

Applications of VHDL

1. It is used in electronic design automation to describe mixed signal system such as FPGA (Field Programmable Gate Arrays) and Integrated circuits.
2. VHDL can be used as a general purpose parallel programming language.
3. VHDL can also be used for design and simulation purposes.

(b)TTL and CMOS logic families:

TTL family:

The long form of TTL is Transistor Transistor Logic. It is a logic family consisting completely of transistors. It employs transistors with multiple emitters. The digital ICs in the TTL family use only transistors as their basic building block. TTL ICs were first developed in 1965 and they were known as “Standard TTL”. This version of TTL is not practically used now due to availability of advanced versions. Some characteristics of TTL family are as follows:

1. TTL is made up of BJTs (Bipolar Junction Transistor).
2. It has a high level noise margin of 0.4 V i.e. V_{NH} = 0.4 V.
3. It has a low level noise margin of 0.4 V i.e. V_{NL} = 0.4 V.
4. TTL family has less noise immunity than CMOS family.
5. The propagation delay of a standard TTL is 10nS.
6. It has a comparatively faster switching speed as compared to CMOS family.
7. It has a power dissipation of 10mW per gate i.e. P_D= 10mW.

8. TTL family has a speed power product of 100 pJ.
9. P_D of TTL does not depend on frequency.
10. TTL has a Fan Out capacity of 10.
11. In TTL, inputs can remain floating. The floating inputs are treated as logic 1s.
12. Since BJTs require more space, TTLs have a comparatively low component density.
13. Transistors are operated in saturated or cut off regions.
14. Power supply voltage of TTL is fixed which is equal to 5V.

CMOS family:

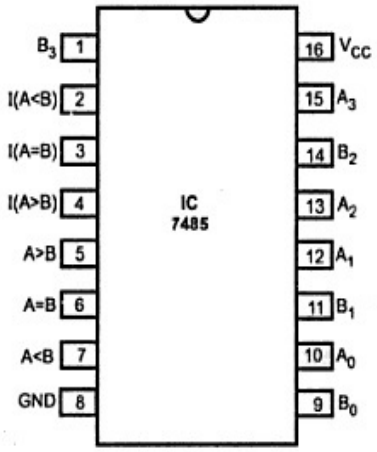
Complementary Metal Oxide Semiconductor (CMOS) is a technology for constructing integrated circuits, employing MOSFET transistors. CMOS technology is used in microprocessors, microcontrollers, static RAM, and other digital logic circuits. CMOS technology is also used for several analog circuits such as image sensors (CMOS sensor), data converters, and highly integrated transceivers for many types of communication. Some characteristics of CMOS logic family are as follows:

1. CMOS devices are made up of N-channel MOSFET and P-channel MOSFET.
2. It has a high level noise margin of 1.45 V i.e. V_{NH} = 1.45 V.
3. It has a low level noise margin of 1.45 V i.e. V_{NL} = 1.45 V.
4. CMOS family has better noise immunity than TTL family.
5. The propagation delay of a metal gate CMOS is 105nS.
6. It has a comparatively slower switching speed as compared to TTL family.
7. It has a power dissipation of 0.1 mW per gate i.e. P_D= 0.1 mW. Hence, it is used for battery backup applications.
8. CMOS family has a speed power product of 10.5 pJ.
9. P_D of CMOS depends on frequency.
10. CMOS has a Fan Out capacity of 50.
11. The unused inputs should be returned to GND or V_{DD}.They should never be left floating.
12. CMOS usually have a high component density than TTL as MOSFETs need less space while fabricating an IC.
13. MOSFETs are operated as switches i.e. in the ohmic region or cut off regions.
14. Power supply voltage of CMOS is flexible ranging from 3V to 15V.

(c)4-bit magnitude comparator

Solution:

A 4-bit comparator is used to compare two 4-bit words A (A_3- A_0) and B (B_3- B_0). IC 7485 is a four bit comparator in the integrated circuit form. It is possible to cascade more than one IC 7485 to compare words of almost any size.The figure below shows the pin configuration and the logic symbol of IC 7485.



(a) Pin diagram (IC 7485)

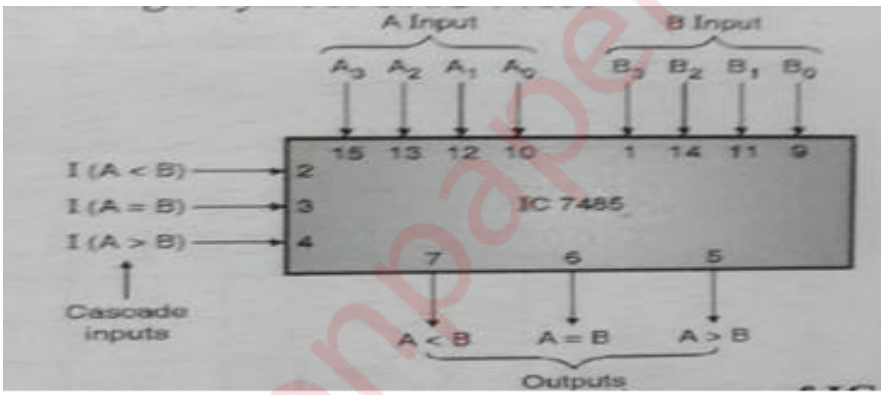


Fig13. Logic diagram of IC7485

The Pin names and their functions are as shown in the table below:

Pin name	Pin number	Function
A_0 to A_3	10,12,13,15	Binary input (opearand 1) Active high
B_0 to B_3	9,11,14,1	Binary input (opearand 2) Active high
I (A < B) I (A = B) I (A > B)	2 3 4	These lines are used for cascading a number of IC 7485 outputs of the previous stage are fed as inputs to this stage.
A < B A = B A > B	7 6 5	These are the outputs. When ICs 7485 are cascaded, these outputs are applied to cascading inputs of the next stage.

Comparing inputs				Cascading inputs			Outputs		
A3,B3	A2,B2	A1,B1	A0,B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	X	X	X	X	X	X	H	L	L
A3<B3	X	X	X	X	X	X	L	H	L
A3=B3	A2>B2	X	X	X	X	X	H	L	L
A3=B3	A2<B2	X	X	X	X	X	L	H	L
A3=B3	A2=B2	A1>B1	X	X	X	X	H	L	L
A3=B3	A2=B2	A1<B1	X	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	H	L	L	H	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	H	L	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	X	X	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	L	L	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	L	H	H	L