

MU

es easy-solutions

Mumbai University Paper Solutions

Strictly as per the new revised syllabus of
Mumbai University
w.e.f. academic year 2013-2014

Digital Logic Design and Analysis

Semester III – Computer Engineering

Solved University Question Papers Upto May 2016....

B-40

EMO36C Price ₹ 85/-



easy – solutions

Digital Logic Design and Analysis

As per the new revised syllabus w.e.f. academic year 2013-2014

Semester III – Computer Engineering

- May 2015 8-1 to 8-18
- Dec. 2015 9-1 to 9-17
- May 2016 10-1 to 10-12
- University Question Papers Q-6 to Q-8
- Syllabus
- Index
- Dec 2011 1-1 to 1-21
- May. 2012 2-1 to 2-27
- Dec. 2012 3-1 to 3-13
- May 2013 4-1 to 4-20
- Dec. 2013 5-1 to 5-14
- May 2014 6-1 to 6-24
- Dec. 2014 7-1 to 7-17
- University Question Papers Q-1 to Q-5

Marketed By



EMO36C



Digital Logic Design and Analysis

Statistical Analysis

Chapter No.	May 2015	Dec. 2015	May 2016
Chapter 1	6 Marks	2 Marks	03 Marks
Chapter 2	6 Marks	-	04 Marks
Chapter 3	10 Marks	4 Marks	05 Marks
Chapter 4	9 Marks	-	04 Marks
Chapter 5	24 Marks	22 Marks	10 Marks
Chapter 6	-	2 Marks	05 Marks
Chapter 7	31 Marks	40 Marks	40 Marks
Chapter 8	7 Marks	-	05 Marks
Chapter 9	10 Marks	6 Marks	15 Marks
Chapter 10	10 Marks	22 Marks	15 Marks
Chapter 11	6 Marks	20 Marks	10 Marks
Chapter 12	-	-	-
Chapter 13	7 Marks	-	13 Marks
Chapter 14	-	-	-
Appendix-A		2 Marks	04 Marks
Repeated Questions	20 Marks	25 Marks	20 Marks

May 2015

Chapter 1 : Number Systems [Total Marks 6]

(2 Marks)

Q. 1(a) Convert $(121.2)_3$ into base 10.

Ans. :

Step 1 : Convert $(121.2)_3$ into decimal :

Given number :

1	2	1	.	2
---	---	---	---	---

Decimal :
$$(1 \times 3^2) + (2 \times 3^1) + (1 \times 3^0) + (2 \times 3^{-1})$$

(C-5186)

$$\therefore N = 9 + 6 + 1 + 0.66 = 16.66$$

$$\therefore (121.2)_3 = (16.66)_{10}$$

...Ans.

Q. 1(f) Convert $(126)_{10}$ to Octal, Hexcode.

Ans. :

Step 1 : Convert decimal to octal :

8 126 6		LSB
8 15 7		
8 1 1		
	0	MSB

(C-5189)

$$\therefore (126)_{10} = (176)_8$$

Step 2 : Convert decimal to hexadecimal :

16 126 14	→ E	LSB
16 7 7	→ 7	
	0	MSB

(C-5190)

$$\therefore (126)_{10} = (7E)_{16}$$

...Ans.

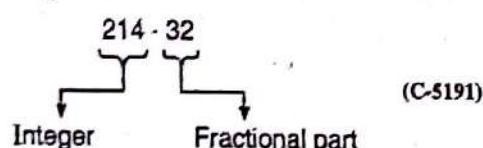
...Ans.

Q. 1(h) Convert $(214.32)_{10}$ to binary.

(2 Marks)

Ans. :

Step 1 : Convert integer and fractional parts :



Step 2 : Convert integer :

2 214 0		LSB
2 107 1		
2 53 1		
2 26 0		
2 13 1		
2 6 0		
2 3 1		
2 1 1		
	0	MSB

Step 3 : Convert fractional part :

Decimal fraction	Base	Product	Carry	
0.32	× 2	= 0.64	0	MSB
0.64	× 2	= 1.28	1	
0.28	× 2	= 0.56	0	
0.56	× 2	= 1.12	1	
0.12	× 2	= 0.24	0	LSB

$$\therefore (214)_{10} = (11010110)_2$$

$$(0.32)_{10} = (0.01010)_2$$

(C-5192)

Step 4 : Combine results of steps 2 and 3 :

$$(214.32)_{10} = (11010110.01010)_2$$

...Ans.

Chapter 2 : Binary Arithmetic [Total Marks 6]

Q. 1(c) Find the one's complement and two's complement of $(57)_{10}$. (2 Marks)

Ans. :

Step 1 : Convert decimal to binary :

$$(57)_{10} \longrightarrow (111001)_2$$

Step 2 : 1's complement of $(57)_{10}$:

$$(111001)_2 \xrightarrow{\text{1's complement}} (000110)_2$$

Step 3 : 2's complement of $(57)_{10}$:

$$(111001)_2 \xrightarrow{\text{1's complement}} (000110)_2 \xrightarrow{\text{add 1}} (000111)_2$$

Q. 1(l) Perform binary subtraction using 2's complement for $(62)_{10}$ and $(99)_{10}$. (4 Marks)

Ans. :

Step 1 : Convert both the numbers to binary :

$$(62)_{10} = (0111110)_2$$

$$(99)_{10} = (1100011)_2$$

Step 2 : Obtain 2's complement of $(99)_{10}$:

Decimal	Binary	1's complement	2's complement
$(99)_{10}$	$\longrightarrow (1100011)_2$	$\longrightarrow (0011100)_2$	$\longrightarrow (0011101)_2$

Step 3 : Add $(62)_{10}$ and 2's complement of $(99)_{10}$:

$$\begin{array}{r}
 (62)_{10} : 0\ 1\ 1\ 1\ 1\ 1\ 0 \\
 + \\
 \text{2's complement of } (99)_{10} : 0\ 0\ 1\ 1\ 1\ 0\ 1 \\
 \hline
 \text{Carry} : 1\ 1\ 1\ 1
 \end{array}$$

$$\boxed{0}\ 1\ 0\ 1\ 1\ 0\ 1\ 1$$

(C-5193)

→ "0" carry indicates that the result
is negative and in its 2's complement form

Step 4 : Convert the answer into its true form :

$$\text{Answer} : 1\ 0\ 1\ 1\ 0\ 1\ 1$$

$$\begin{array}{r}
 \text{Subtract 1} : \quad \quad \quad 1 \\
 \hline
 1\ 0\ 1\ 1\ 0\ 1\ 0 \\
 \downarrow \text{Invert all bits} \\
 0\ 1\ 0\ 0\ 1\ 0\ 1
 \end{array}$$

(C-5194)

$$\therefore (62)_{10} - (99)_{10} = -(0100101)_2 = -(37)_{10}$$

...Ans.

Chapter 3 : Codes [Total Marks 10]

(2 Marks)

Q. 1(b) Represent $(52)_{10}$ into Excess - 3 code and Gray code.

Ans. : Write the BCD equivalent of $(52)_{10}$:

$$\text{Step 1 : } (52)_{10} = (01010010)_{\text{BCD}}$$

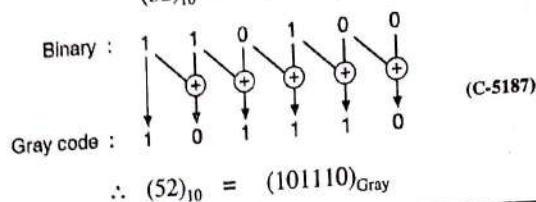
S Step 2 : Convert decimal to excess-3 code :

$$\begin{array}{ll} (52)_{10} & : \quad 0 \ 1 \ 0 \ 1 \quad 0 \ 0 \ 1 \ 0 \\ \text{Excess 3} & : \quad 1 \ 0 \ 0 \ 0 \quad 0 \ 1 \ 0 \ 1 \\ \therefore (52)_{10} & = (1000 \ 0101)_{\text{Excess-3}} \end{array}$$

...Ans

Step 3 : Convert $(52)_{10}$ to gray code :

$$(52)_{10} = (110100)_2$$



...Ans

**Q
At
St**

$$\therefore (52)_{10} = (101110)_{\text{Gray}}$$

(2 Marks)

Q. 1(e) Obtain hamming code for 1011.

Ans. :

Step 1 : The code word format :

The seven bit Hamming code format is shown in Fig. 1-Q. 1(e).

Given bit word = 1 0 1 1

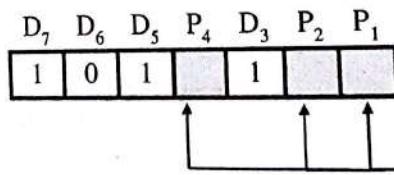


Fig. 1-Q. 1(e)

Step 2 : Decide P_1 :

P_1 sets the parity of bits P_1 , D_3 , D_5 and D_7 . As $D_7 D_5 D_3 = 1 \ 1 \ 1$ we have to set $P_1 = 1$ in order to have the even parity.

Ste

D_7	D_6	D_5	P_4	D_3	P_2	P_1
1	0	1		1		1

Set $P_1 = 1$ to have the even parity of $P_1 D_3 D_5 D_7$

Step 3 : Decide P_2 :

P_2 is set to have the even parity of P_2 , D_3 , D_6 and D_7 . But $D_3, D_6, D_7 = 1 \ 0 \ 1$ hence $P_2 = 0$.

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

→ Set P₂ = 0 to have even parity of P₂, D₃, D₆ and D₇

Step 4 : Decide P₄:

P₄ is set to have the even parity of P₄, D₅, D₆ and D₇. But D₅, D₆, D₇ = 1 0 1, hence set P₄ = 0.

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

↓ ← Complete code word

P₄ = 0 to have even parity of P₄, D₅, D₆, D₇

Q. 6(c) Write short note on gray code and Excess-3 code.

(6 Marks)

Ans. :

1. Excess - 3 Code :

Excess - 3 is also called as XS - 3 code. It is a nonweighted code used to express decimal numbers as shown in Table 1.

The Excess-3 code words are derived from the 8421 BCD code words by adding (0011)₂ or (3)₁₀ to each code word in 8421. The excess - 3 codes are obtained as follows :

Add

Decimal Number → 8421 BCD → Excess - 3 code
0011

Excess - 3 codes for the single digit decimal numbers are listed in Table 1.

Table 1 : Excess - 3 codes

Decimal	BCD				Excess - 3			
	8	4	2	1	BCD + 0011			
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

Note : Excess - 3 is a sequential code, because each succeeding code is one binary number greater than its preceding code.

Excess - 3 is a **sequential code** because we get any code word by adding binary 1 to its previous code word as illustrated in Table 1.

Excess - 3 is a **self complementing code**. This is because in Excess - 3 we get the 9's complement of a number by just complementing each bit that means by replacing a 0 by 1 and 1 by 0.

2. Gray code :

Gray code is another non-weighted code. It is not an arithmetic code. It has a very special feature that only one bit in the gray code will change, each time the decimal number is incremented as shown in Table 2. As only one bit changes at a time, the Gray code is called as a **unit distance code**. The Gray code is a **Cyclic code**.

Table 2

Decimal	Binary	Gray
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 0
11	1 0 1 1	1 1 1 1
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0
16	1 0 0 0	0 0 0 0

- Note :
1. In gray code only the encircled bit changes when the decimal number is incremented by 1.
 2. In binary one, two, three or sometimes all the 4 bits change when the decimal number is incremented by 1.

Chapter 4 : Logic Gates and Boolean Algebra [Total Marks 9]

Q. 1(d) Realize $y = AB + \overline{AB}$ using NAND gates only. (2 Marks)

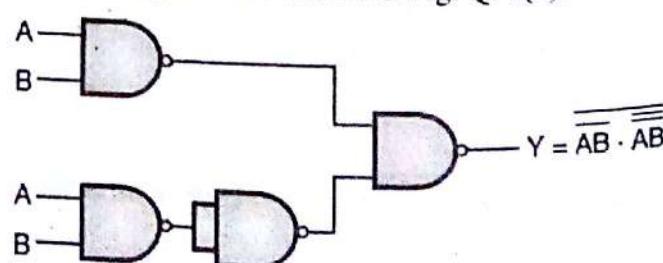
Ans. :

$$Y = A \cdot B + \overline{AB}$$

Taking double inversion of RHS,

$$= \overline{\overline{AB}} + \overline{\overline{AB}} = \overline{\overline{AB} \cdot \overline{AB}}$$

Implementation using NAND gates is as shown in Fig. Q. 1(d).



(C-5188) Fig. Q. 1(d) : Implementation using NAND gates

Q. 1(g) State Demorgans law.**(2 Marks)****Ans. : Please refer Q. 1(a) of Dec. 2013.****Q. 3(b) Simplify $Y = (A + \bar{A}B)(C + \bar{D})$.****(5 Marks)****Ans. :**

$$\begin{aligned}
 Y &= \overline{(A + \bar{A}B)(C + \bar{D})} \\
 &= (A + \bar{A}B) + (C + \bar{D}) \quad \dots(\text{Using De-morgan's theorem}) \\
 &= [\bar{A} \cdot \overline{\bar{A}B}] + [\bar{C} \cdot \overline{\bar{D}}] \quad \dots(\text{Using De-morgan's theorem}) \\
 &= \bar{A} \cdot [A \cdot \bar{B}] + [\bar{C} \cdot D] \quad \dots(\text{Using De-morgan's theorem}) \\
 &= \bar{A} \cdot [\bar{A}\bar{B}] + \bar{C}D \quad \dots(\because \bar{A} \cdot \bar{A} = 0) \\
 &= \bar{A} A \bar{B} + \bar{C}D \quad \dots(\because A\bar{A} = 0) \\
 Y &= \bar{C}D
 \end{aligned}$$

Chapter 5 : Logic Minimization and Reduction Techniques [Total Marks 24]

Q. 2(a) Minimize the logic function using Quine-Mc Cluskey method. $f(A, B, C, D) = \sum m(1, 3, 7, 9, 10, 11, 13, 15)$ **(12 Marks)****Ans. :****Step 1 : Group the minterms according to number of 1's :****Table 1**

Group	Minterm	Binary representation			
		A	B	C	D
1	1	0	0	0	1
2	3	0	0	1	1
	9	1	0	0	1
	10	1	0	1	0
3	7	0	1	1	1
	11	1	0	1	1
	13	1	1	0	1
4	15	1	1	1	1

One 1
 Two 1's
 Three 1's
 Four 1's

Q. 1

Ans

Step

Step 2 : Group the minterms to form the pairs :

Table 2

Group	Minterm pair	Binary representation			
		A	B	C	D
1	1 - 3	0	0	-	1
	1 - 9	-	0	0	1
2	3 - 7	0	-	1	1
	9 - 11	1	0	-	1
	9 - 13	1	-	0	1
	10 - 11	1	0	1	-
3	3 - 11	-	0	1	1
	7 - 15	-	1	1	1
	11 - 15	1	-	1	1
	13 - 15	1	1	-	1

$$A \bar{B} C$$

Q. 1
Ans.

Step

Step 3 : Group the minterms to form groups of four :

Table 3

Group	Minterm quad	Binary representation				Prime implicants
		A	B	C	D	
1	1, 3, 9, 11	-	0	-	1	$\bar{B}D$
	1, 9, 3, 11	-	0	-	1	
2	3, 7, 11, 15	-	-	1	1	CD
	3, 11, 7, 15	-	-	1	1	
	9, 11, 13, 15	1	-	-	1	AD
	9, 13, 11, 15	1	-	-	1	

(C.S.I)

Step

Step 4 : Prepare the table of prime implicants :

Table 4

Prime Implicants	Decimal numbers	Given minterms								
		1	3	7	9	10	11	13	15	
$A\bar{B}C$	10, 11					(X)	X			
$\bar{B}D$	1, 3, 9, 11	(X)	X		X		X		X	
CD	3, 7, 11, 15		X	(X)			X		X	
AD	9, 11, 13, 15				X		X	(X)	X	

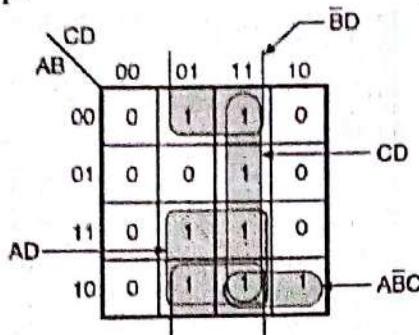
Step

@esl

The encircled crosses represent the EPIs. They cover minterms 1, 3, 7, 9, 10, 11, 13 and 15.

$$\therefore f(A, B, C, D) = A\bar{B}C + \bar{B}D + CD + AD \quad \dots \text{Ans.}$$

Step 5 : Crosscheck using K-map :



$$\therefore f(A, B, C, D) = A\bar{B}C + \bar{B}D + CD + AD$$

(C-5199) Fig. Q. 2(a)

Q. 4(a) Given the logic expression :

$$A + \bar{B}\bar{C} + A\bar{B}\bar{D} + ABCD$$

1. Express in standard SOP
2. Draw K-map for the equation.
3. Minimize and realize using NAND gates only.

(12 Marks)

Ans. :

1. $Y = A + \bar{B}\bar{C} + A\bar{B}\bar{D} + ABCD :$

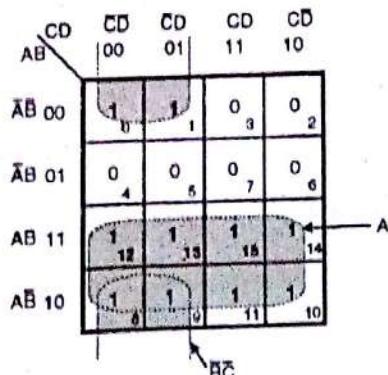
Standard SOP form

$$Y = A(B + \bar{B})(C + \bar{C})(D + \bar{D}) + \bar{B}\bar{C}(A + \bar{A})(D + \bar{D}) + A\bar{B}\bar{D}(C + \bar{C}) + ABCD$$

$$Y = ABCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} \\ + A\bar{B}\bar{C}D + A\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + ABC\bar{D} + A\bar{B}\bar{C}\bar{D} + ABCD$$

$$Y = ABCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} \\ + \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}D$$

2. K-map :



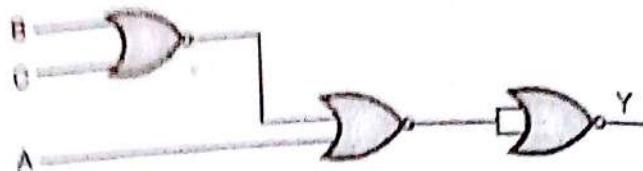
$$\therefore Y = A + \bar{B}\bar{C}$$

(C-3746) Fig. 1-Q. 4(a)

D.L.F.**Q. 1
Ans.
Step****S 1 DUDA (COMP + MU)****Q. 1 Logic diagram using NOT gates only :**

$$Y = A + \bar{B}\bar{C} = A + \bar{B} + \bar{C}$$

$$\therefore Y = A + \bar{B} + \bar{C}$$

**Step**

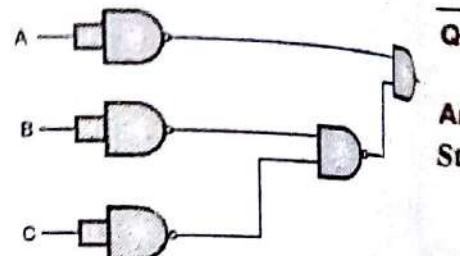
(C-3747) Fig. 2-Q. 4(a)

Q. 4 Logic diagram using NAND gates only :

$$Y = A + \bar{B}\bar{C}$$

According to De-Morgan's theorem

$$Y = \overline{\overline{A + \bar{B}\bar{C}}} = \overline{\overline{A}} \cdot \overline{\overline{\bar{B}\bar{C}}} = \bar{A} \cdot \bar{\bar{B}}\bar{C} = \bar{A} \cdot \bar{B}\bar{C}$$



(C-3748) Fig. 3-Q. 4(a)

S 5 Expression in standard POS form :

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

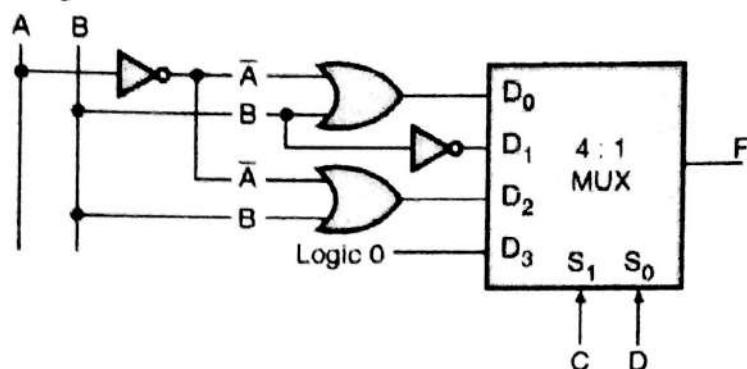
Step**Chapter-7 : Combinational Logic Design [Total Marks 31]****Q. 2(b) Implement the following expression using single 4 : 1 Mux.
 $F(A,B,C,D) = \sum m(0, 1, 2, 4, 6, 9, 12, 14)$** **Ans. :****Step 1 : Write the design table :**

$$D_0 = \bar{A}\bar{B} + \bar{A}B + AB \\ = \bar{A}(\bar{B} + B) + AB \\ = \bar{A} + AB = \bar{A} + B \\ D_1 = \bar{A}\bar{B} + A\bar{B} = \bar{B}(\bar{A} + A) = \bar{B} \\ D_2 = \bar{A}\bar{B} + \bar{A}B + AB \\ = \bar{A}(\bar{B} + B) + AB = \bar{A} + AB = \bar{A} + B = D_3 = 0$$

Inputs	D ₀	D ₁	D ₂	D ₃
$\bar{A}\bar{B}$	0	1	2	3
$\bar{A}B$	4	5	6	7
$A\bar{B}$	8	9	10	11
AB	12	13	14	15

Step**esl****a****easy-solutions**

Step 2 : Implementation using 4 : 1 MUX :



(C-3268) Fig. Q. 2(b)

Q. 3(a) Design a 4-input (A,B,C,D) digital circuit that will give at its output (X) a logic 1 only if the binary number formed at the input is between 2 and 9 (including). **(10 Marks)**

Ans. :

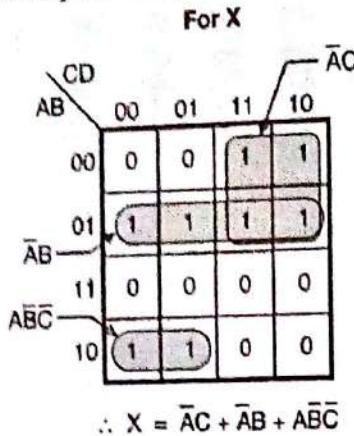
Step 1 : Write the truth table :

Table 1

Decimal	Inputs				Output X
	A	B	C	D	
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

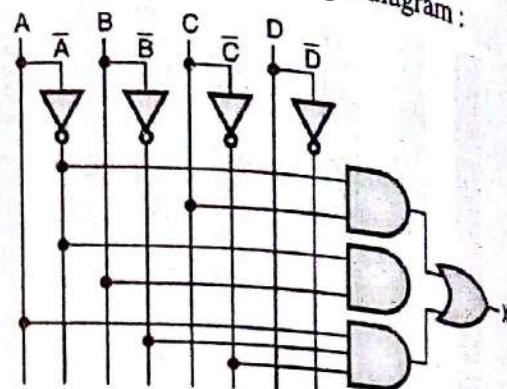
Q. 1
Ans
Step

Step 2 : K-map and simplification :



Step

Step 3 : Draw the logic diagram :



(C-5195) Fig. Q. 3(a) : Logical diagram

Q. 1
Ans.
Step

Q. 3(c) Design 1 bit comparator using logic gates.

Ans. :

1-Bit Binary Comparator :

Truth table of one-bit comparator :

The one-bit comparator is a combinational logic circuit with two inputs A and B and outputs namely $A < B$, $A = B$ and $A > B$. A one bit comparator compares the two single bit A and B and produces an output that indicates the result of the comparison. This is clear from the table as given in Table 1.

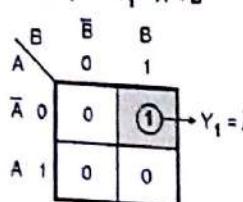
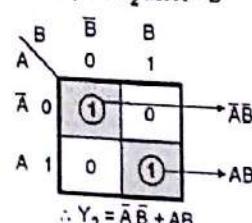
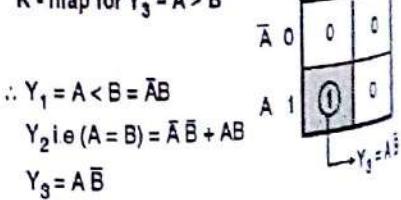
Table 1 : Truth table of a one-bit comparator

Inputs		Outputs		
A	B	$Y_1 = A < B$	$Y_2 = A = B$	$Y_3 = A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Step

K-maps for each output :

The K-maps for the three outputs Y_1 , Y_2 and Y_3 are as shown in Fig. 1-Q. 3(c).

K-map for $Y_1 = A < B$ K-map for Y_2 i.e. $A = B$ K-map for $Y_3 = A > B$ 

$$\therefore Y_1 = A < B = \bar{A}B$$

$$Y_2 \text{ i.e. } (A = B) = \bar{A}\bar{B} + AB$$

$$Y_3 = A\bar{B}$$

Step

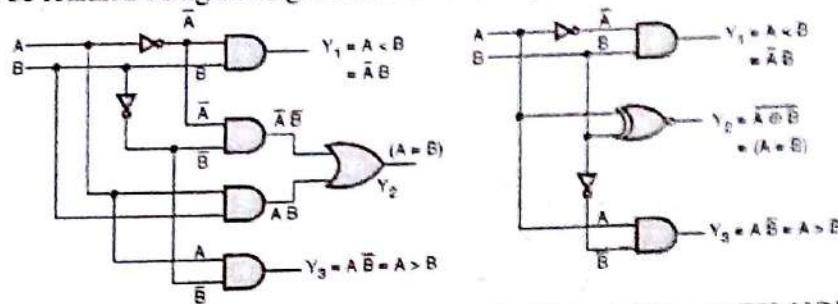
(C-404) Fig. 1-Q. 3(c) : K-maps for the three outputs of a one-bit comparator
From Fig. 1-Q. 3(c) we can write the expressions for the three outputs as,

$$Y_1 = (A < B) = \bar{A}B ,$$

$$Y_2 = (A = B) = \bar{A} \bar{B} + AB = \overline{A \oplus B}$$

$$Y_3 = A \bar{B}$$

The expression for Y_2 is nothing but the expression for an EX-NOR gate. Hence the single bit comparator can be realized using basic gates as shown in Fig. 2-Q. 3(c).



(a) Using basic gates

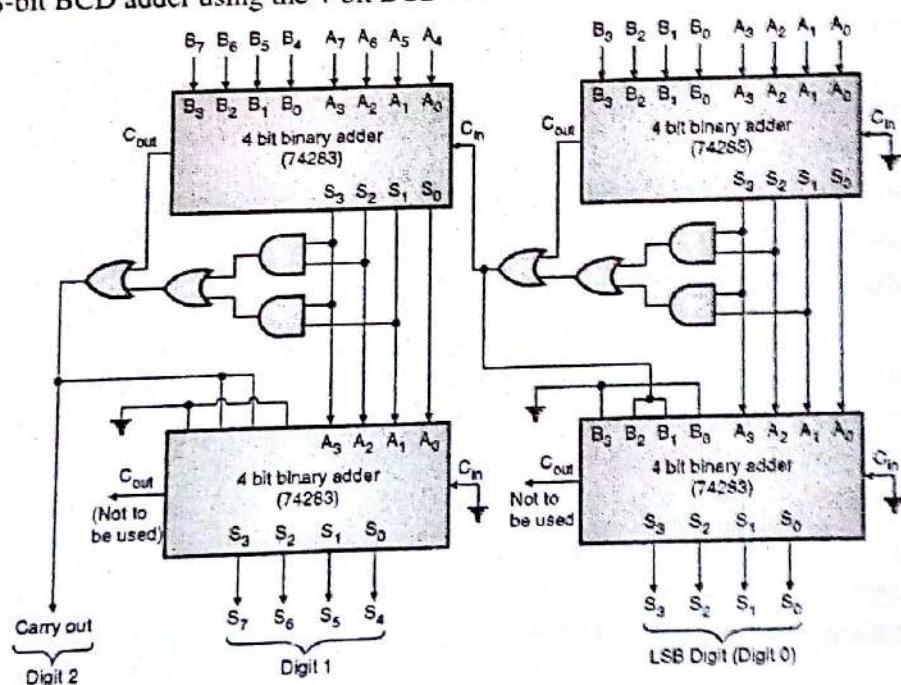
(b) Using AND and EX-NOR gates

(C-405) Fig. 2-Q. 3(c) : Realization of one bit comparator

(S Marks)

Q. 4(b) Design 8 bit BCD adder.**Ans. :****8-Bit BCD Adder:**

The 8-bit BCD adder using the 4-bit BCD adder 74283 is shown in Fig. Q. 4(b).



(C-400) Fig. Q. 4(b) : 8-bit BCD adder using 74283

Chapter 8 : Multivibrators [Total Marks 7]

(7 Marks)

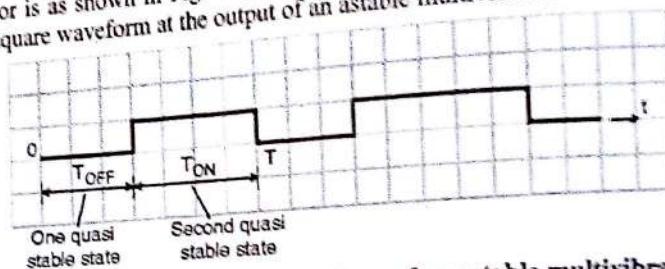
Q. 6(b) Write short note on multivibrators.

Ans. :

A multivibrator is an oscillator that produces non-sinusoidal waveforms such as a square wave. A square wave consists of harmonics or multiple variations of fundamental frequency. Therefore it is called multivibrators.

(1) Astable multivibrator :

An astable multivibrator has two quasi-stable states and no stable state. Its output changes its state from one quasi state to the other repeatedly. It does not require any external trigger input to change its state. Hence it is also called as a free running multivibrator. The output voltage waveform of an astable multivibrator is as shown in Fig. 1-Q. 6(b). It is a rectangular waveform. It is also possible to obtain a perfectly square waveform at the output of an astable multivibrator.



(K-77) Fig. 1-Q. 6(b) : Output waveform of an astable multivibrator

Total time period of one cycle of the output is given by,

$$T = T_{ON} + T_{OFF}$$

$$\text{And frequency } f = 1/T.$$

Applications of astable multivibrator :

1. As a source of "clock" signal for digital circuits.
2. As a square wave oscillator.

(2) Monostable multivibrator :

The monostable multivibrator has one stable state and another quasi stable state. It requires an external trigger input to change its state from stable to quasi stable. After some time it will automatically return back to the stable state. The duration for which the output remains in the quasi-stable state is called as the pulse width or on time.

Types of monostable :

There are two types of monostable multivibrators :

1. Non retriggerable monostable multivibrator.
2. Retriggerable monostable multivibrator.

stable

Retri

circu

pulse

when

durin

show

stabili

nonre

strea

App

1.

2.

3.

4.

(3)

stabili

chan

mult

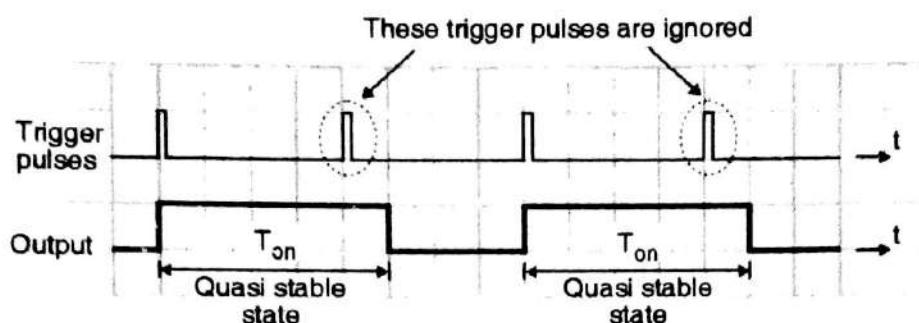
bist

state

wave

Non retriggerable monostable :

This is the normal monostable multivibrator.



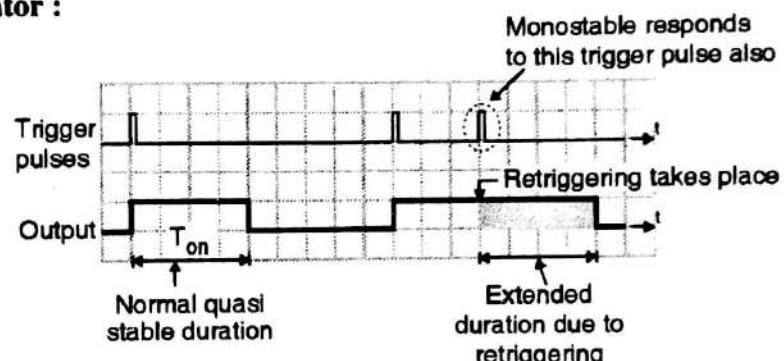
(K-778) Fig. 2-Q. 6(b) : Non retriggerable monostable multivibrator

In this mode, the circuit ignores any trigger pulse applied at its trigger input during the quasi stable period, as shown in Fig. 2-Q. 6(b).

Thus T_{on} in Fig. 2-Q. 6(b) does not get extended by the trigger pulses applied during T_{on} .

Retriggerable monostable multivibrator :

This type of monostable circuit, responds to every trigger pulse applied at its inputs even when the trigger pulse appears during the quasi stable period as shown in Fig. 3-Q. 6(b).



(K-779) Fig. 3-Q. 6(b) : Waveforms of retriggerable monostable multivibrators

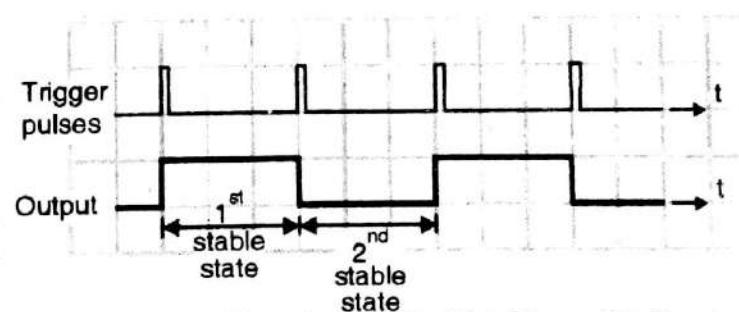
The quasi stable duration goes on extending due to the trigger pulses arriving inbetween the quasi stable period. The monostable multivibrator is said to be retrigged by every trigger input pulse. The normal monostable multivibrators are also called as "one shot". They are also called as "pulse stretchers".

Applications of monostable multivibrator :

1. Monostable multivibrator can be used basically as a timer.
2. In the ON-OFF applications.
3. Light sensitive relays.
4. Sleep timers in TV, AC etc.

(3) Bistable multivibrator :

A bistable multivibrator has two stable states. It need external trigger input to change the existing state. Bistable multivibrator is basically a flip-flop. The bistable multivibrator has no quasi stable state. Fig. 4-Q. 6(b) illustrates the output waveform of a bistable multivibrator circuit.



(K-1180) Fig. 4-Q. 6(b) : Waveforms of a bistable multivibrator

D LDA (COMP - MU)**Q Applications :**

1. As a memory cell to store 1 bit of information.
2. As a basic unit to build registers and counters.

Chapter 9 : Flip Flops [Total Marks 10]

(10 Marks)

Q. 5(b) Convert SR FF to TFF and JK FF.**Ans. :****1. SR Flip Flop to T Flip Flop :**

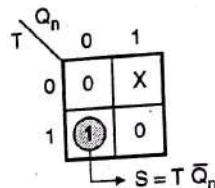
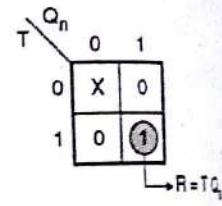
The stepwise conversion process is as follows :

S Step 1 : Write the truth table :

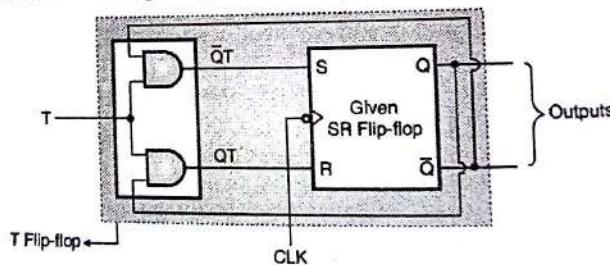
Table 1 : Truth table for SR FF to T FF

Inputs			Outputs	
T	Present state Q_n	Next state Q_{n+1}	S	R
0	0	0	0	X
1	0	1	1	0
1	1	0	0	1
0	1	1	X	0

Step 2 : Write the K maps and obtain expressions for S and R :

For S output**For R output****(a) K map for S (b) K map for R****(C-617) Fig. 1-Q. 5(b)****Step 3 : Draw the logic diagram :**

The logic diagram is shown in Fig. 2-Q. 5(b).

**(C-618) Fig. 2-Q. 5(b) : Conversion from SR flip flop to T flip flop****2. SR Flip Flop to JK Flip Flop : Please refer Q. 6(c) of May 2014.****Chapter 10 : Counters [Total Marks 10]****Q. 5(c) Design a mod 5 synchronous up counter using JK FF.**

(10 Marks)

Ans. :

S Step 1 : Determine the desired number of FFs :
 From the given sequence the number of FFs is equal to 3. This is a MOD-5 synchronous counter since the number of states is 5.

Step 2 : Write the excitation table and state table :

The type of FF used is JK flip-flop. The excitation table for a JK FF is as shown in Table 1. We have already seen how to write the excitation table for JK FF.

Excitation table of JK FF :

Table 1 exhibits the excitation table of a JK FF.

Circuit excitation table :

The circuit excitation table is as shown in Table 2.

Table 1 : Excitation table of JK FF

Present state Q_a	Next state Q_{a+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

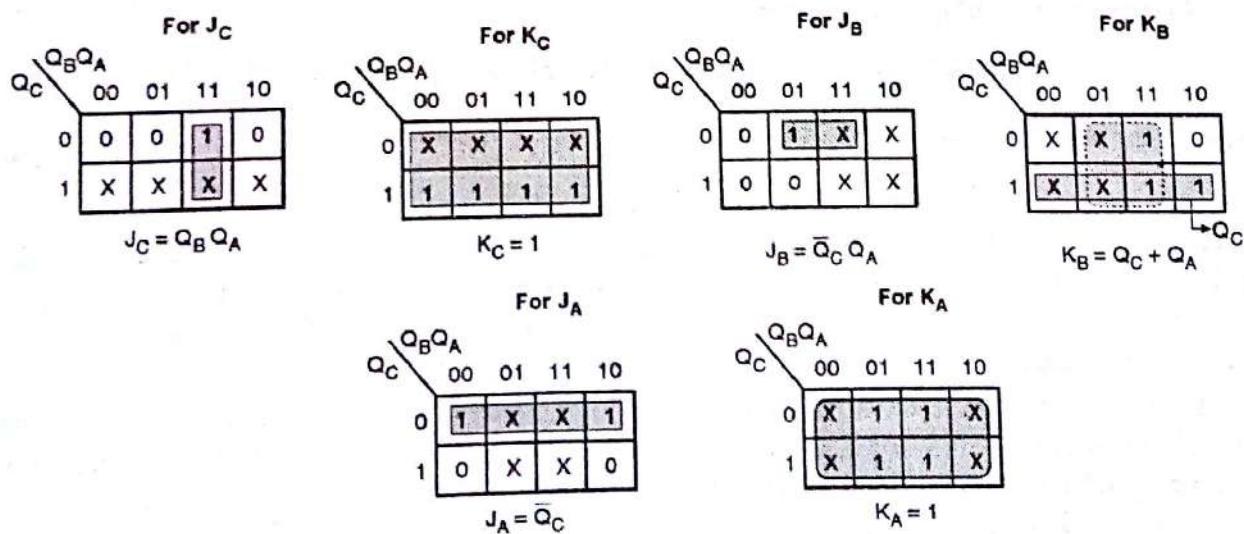
Table 2 : Circuit excitation table

Present state			Next state			Flip-flop inputs					
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	0	0	0	x	1	0	x	0	x
1	0	1	0	0	0	x	1	0	x	x	1
1	1	0	0	0	0	x	1	x	1	0	x
1	1	1	0	0	0	x	1	x	1	x	1

Refer to the shaded portion of the circuit excitation table. This is nothing but the excitation table of FF-C. The J_C and K_C values have been decided based on Q_C and Q_{C+1} . Similarly the entries for J_B and K_B are based on Q_B and Q_{B+1} whereas those for J_A and K_A are based on Q_A and Q_{A+1} .

Step 3 : K-maps and simplifications :

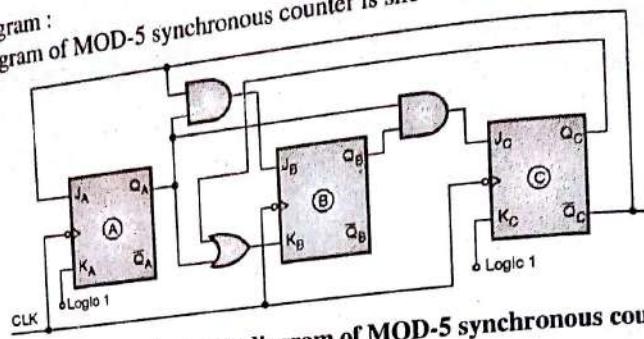
K-maps for the J and K inputs of all the FFs and the corresponding simplified equations are shown in Fig. 1-Q. 5(a). Note that the inputs to this combinational circuit are Q_A , Q_B , Q_C and outputs are J_A , K_A through J_C , K_C .



(C-850) Fig. 1-Q. 5(a) : K-maps and simplifications

Step 4 : Logic diagram:

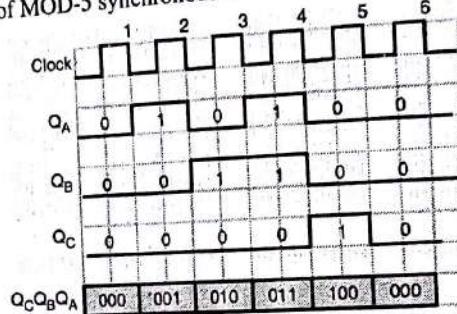
The logic diagram of MOD-5 synchronous counter is shown in Fig. 2-Q. 5(a).



(C-851) Fig. 2-Q. 5(a) : Logic diagram of MOD-5 synchronous counter

Step 5 : Draw the timing diagram:

The timing diagram of MOD-5 synchronous counter is shown in Fig. 3-Q. 5(a).



(C-852) Fig. 3-Q. 5(a) : Timing diagram of MOD-5 counter

Chapter 11 : Registers [Total Marks 6]

Q. 6(d) Write short note on Johnson's ring counter.

(6 Marks)

Ans. : Please refer Q. 5(b) of May 2014.

Chapter 13 : Introduction to VHDL [Total Marks 7]

Q. 6(a) Write short note on VHDL.

(7 Marks)

Ans. :

VHDL : VHDL is a Hardware Description Language. This standard language is used for modelling digital systems made of interconnection of various components. The digital system being designed can be as simple as a simple gate or it can be an extremely complex system. VHDL is an industry standard language used to describe the hardware i.e. it is a hardware language.

It has now become a universal communication medium of design. It is used for specifying the input and output from various design tools used for the digital system design. VHDL can describe the behaviour or structure of a digital system irrespective of its complexity. It is the most widely used hardware description language. Hence it is used for documenting the digital circuits. Such circuits can be understood by the other designers easily. VHDL is a very commonly used and convenient HDL for computers. The hierarchical modelling of a system from top to bottom or vice versa is well supported by the VHDL.

Dec. 2015

Chapter 1 : Number Systems [Total Marks 2]

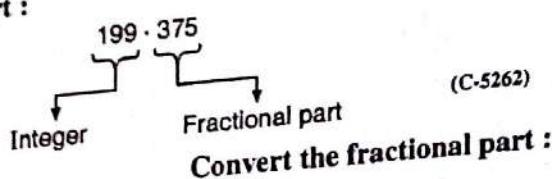
(2 Marks)

Q. 1(a) Convert decimal number 199.375 into binary, octal, hexadecimal system.

Ans. :

$$(199.375)_{10} = (?)_2 = (?)_8 = (?)_{16}$$

Step 1 : Convert decimal to binary :
Separate integer and fractional part :



Convert the integer :

2	199	1
2	99	1
2	49	1
2	24	0
2	12	0
2	6	0
2	3	1
2	1	1
	0	

↑ LSB MSB

Decimal	Base	Product	Carry
0.375	× 2	= 0.75	0
0.75	× 2	= 1.5	1
0.5	× 2	= 1.0	1
0.0	× 2	= 0.0	0

↑ MSB ↓ LSB

(C-5263)

$$\therefore (199)_{10} = (11000111)_2$$

$$\therefore (0.375)_{10} = (0.0110)_2$$

...Ans.

Step 2 : Convert decimal to octal :

Convert the integer :

8	199	7
8	24	0
8	3	3
	0	

↑ LSB MSB

Convert the fractional part :

Decimal	Base	Product	Carry
0.375	× 8	= 3.0	3
0.0	× 8	= 0.0	0

↑ MSB ↓ LSB

(C-5264)

$$\therefore (199)_{10} = (307)_8$$

$$\therefore (0.375)_{10} = (0.30)_8$$

...Ans.

$$\therefore (199.375)_{10} = (307.30)_8$$

DLLA (COMP - MU)

Step 3 : Convert decimal to hexadecimal :
Convert the integer :

Hex		
16	199	7 → 7
16	12	C → C
0		MSB ↑ LSB

$$\therefore (199)_{10} = (C7)_{16}$$

$$\therefore (199.375)_{10} = (C7.60)_{16}$$

Convert the fractional part :

Decimal	Base	Product	Carry
0.375	16	= 6.0	6 MSB
0.0	16	= 0.0	0 LSB

(C-5265)

...Ans.

Chapter 3 : Codes [Total Marks 4]

(2 Marks)

Q. 1(c) Convert binary data 1010 into 7 bit even parity hamming code.

Ans. : Please refer Q. 2(b) of May 2014.

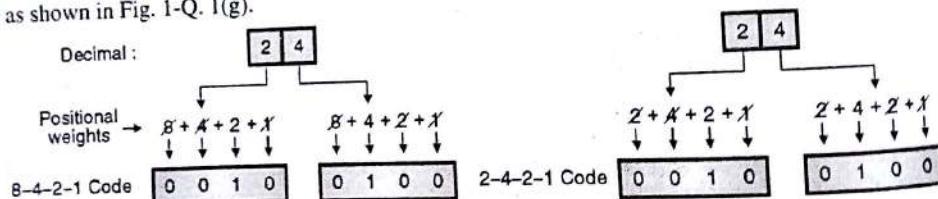
Q. 1(g) Explain in brief weighted and non-weighted codes with one example each.

(2 Marks)

Ans. :

Weighted binary codes :

Weighted binary codes are those codes which are based on the principle of positional weight. Each position of a number represents a specific weight. Several systems of codes are used to express the decimal digits 0 through 9. These codes have been listed in Fig. 1-Q. 1(g). The codes 8421, 2421, 3321 all are the weighted codes. In these codes each decimal digit is represented by a group of four bits as shown in Fig. 1-Q. 1(g).



(C-73) Fig. 1-Q. 1(g)

Non weighted codes :

The codes in which the positional weights are not assigned, are known as non weighted codes. The examples of non-weighted codes are excess-3 and gray codes.

Chapter 5 : Logic Minimization and Reduction Techniques [Total Marks 22]

Q. 1(d) Express the equation in standard POS form : $F(A, B, C) = \sum m(0, 2, 5, 7)$.

(2 Marks)

Ans. :

$$F(A, B, C) = \sum m(0, 2, 5, 7)$$

$$\therefore \text{Equation in standard POS form is } \pi M(1, 3, 4, 6)$$

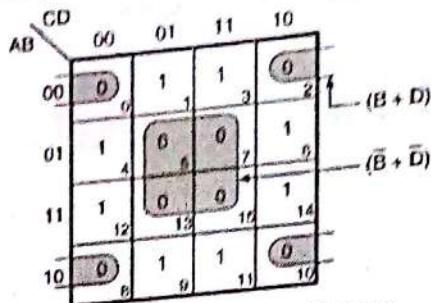
... (Given standard SOP form)
...Ans.

DLLA (COMP - MU)

- Q. 2(a)** Simplify the following equation using K-map to obtain minimum POS equation and realize the minimum equation using only NOR gates.
 $F(A, B, C, D) = \sum m(1, 3, 4, 6, 9, 11, 12, 14)$

(10 Marks)

Ans. :
Step 1: Simplification using K-map :



(C-5266) Fig. 1-Q. 2(a) : K-map

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

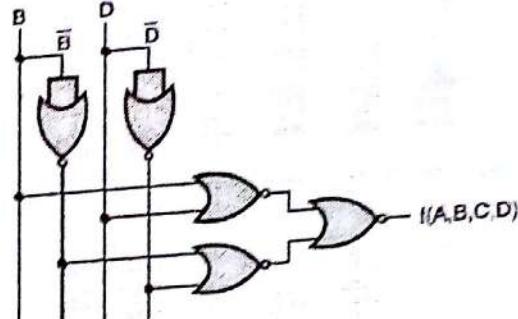
This is required expression in POS form.

- Step 2 :** Implementation using NOR gates :
 Taking double inversion of R.H.S.

$$\begin{aligned} f(A, B, C, D) &= \overline{\overline{(B + D)(\bar{B} + \bar{D})}} \\ &= \overline{\overline{B + D} \cdot \overline{\bar{B} + \bar{D}}} \end{aligned}$$

...(Using De-Morgan's theorem)

Implementation using NOR gates is as shown in
 Fig. 2-Q. 2(a).



(C-5267) Fig. 2-Q. 2(a) : Implementation using NOR gates

- Q. 3(a)** Reduce using Quine Mc-Cluskey method and realize the equation using only NAND gates.
 $F(P, Q, R, S) = \sum m(0, 1, 2, 8, 10, 11, 14, 15)$

(10 Marks)

- Ans. :**
Step 1: Group the minterms according to number of 1's :

Table 1

Group	Minterm	Binary representation			
		P	Q	R	S
0	0	0	0	0	0
	1	0	0	0	1
1	2	0	0	1	0
	8	1	0	0	0
2	10	1	0	1	0
	11	1	0	1	1
3	14	1	1	1	0
	15	1	1	1	1

No 1's

One 1

Two 1's

Three 1's

Four 1's

Step 2 : Group the minterms to form pairs :

Table 2

Group	Minterm pair	Binary representation			
		P	Q	R	S
0	0, 1	0	0	0	-
	0, 2	0	0	-	0
	0, 8	-	0	0	0
1	2, 10	-	0	1	0
	8, 10	1	0	-	0
2	10, 11	1	0	1	-
	10, 14	1	-	1	0
3	11, 15	1	-	1	1
	14, 15	1	1	1	-

$\bar{P} \bar{Q} \bar{R}$
✓
✓
✓
✓
✓
✓
✓
✓

Step 3 : Group the minterms to form quads :

Table 3

Group	Minterm Quad	Binary representation			
		P	Q	R	S
0	0, 2, 8, 10	-	0	-	0
	0, 8, 2, 10	-	0	-	0
2	10, 11, 14, 15	1	-	1	-
	10, 14, 11, 15	1	-	1	-

$\bar{Q} \bar{S}$
PR

Step 4 : Prepare the table of prime implicants :

(C-5268) Table 4

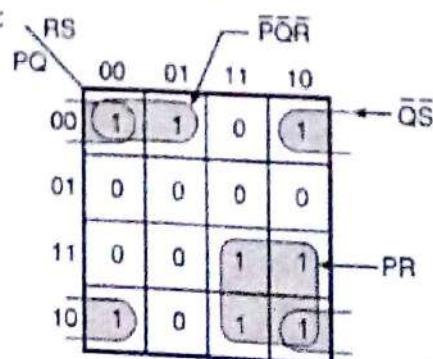
Prime Implicant	Decimal numbers	Given minterms							
		0	1	2	8	10	11	14	15
$\bar{P} \bar{Q} \bar{R}$	0, 1	X	(X)						
$\bar{Q} \bar{S}$	0, 2, 8, 10	X		(X)	(X)	X			
PR	10, 11, 14, 15					X	(X)	(X)	(X)

The encircled crosses represent the EPs. So the terms $\bar{P} \bar{Q} \bar{R}$, $\bar{Q} \bar{S}$ and PR are EPs. They cover all the minterms 0, 1, 2, 8, 10, 11, 14 and 15.

$$\therefore f(P, Q, R, S) = \bar{P} \bar{Q} \bar{R} + \bar{Q} \bar{S} + PR$$

Ans

Step 5 : Crosscheck using K-map :



(C-5269) Fig. 1-Q. 3(a) : K-map

$$\therefore f = (P, Q, R, S) = \bar{P} \bar{Q} \bar{R} + \bar{Q} \bar{S} + PR$$

This is same as the result obtained with Quine Mc-Cluskey method.

Chapter 6 : Logic Families [Total Marks 2]

Q. 1(f) Compare TTL and CMOS with respect to speed, power dissipation, fan-in and fan-out. (2 Marks)

Ans. : Fan-in : Please refer Q. 6(b) of Dec. 2013.

Sr. No.	Parameter	CMOS	TTL
1.	Propagation delay	105 nS (Metal gate CMOS)	10 nS. (Standard TTL)
2	Switching speed	Less than TTL.	Faster than CMOS
3	Power dissipation per gate.	$P_D = 0.1$ mW. Hence used for battery backup applications	10 mW
4	Fan out	Typically 50.	10

Chapter 7 : Combinational Logic Design [Total Marks 40]

Q. 2(b) What is multiplexer tree ? Construct 32 : 1 multiplexer using 8 : 1 multiplexers only. Explain how the logic on particular data line is steered to the output in this design with example ? (10 Marks)

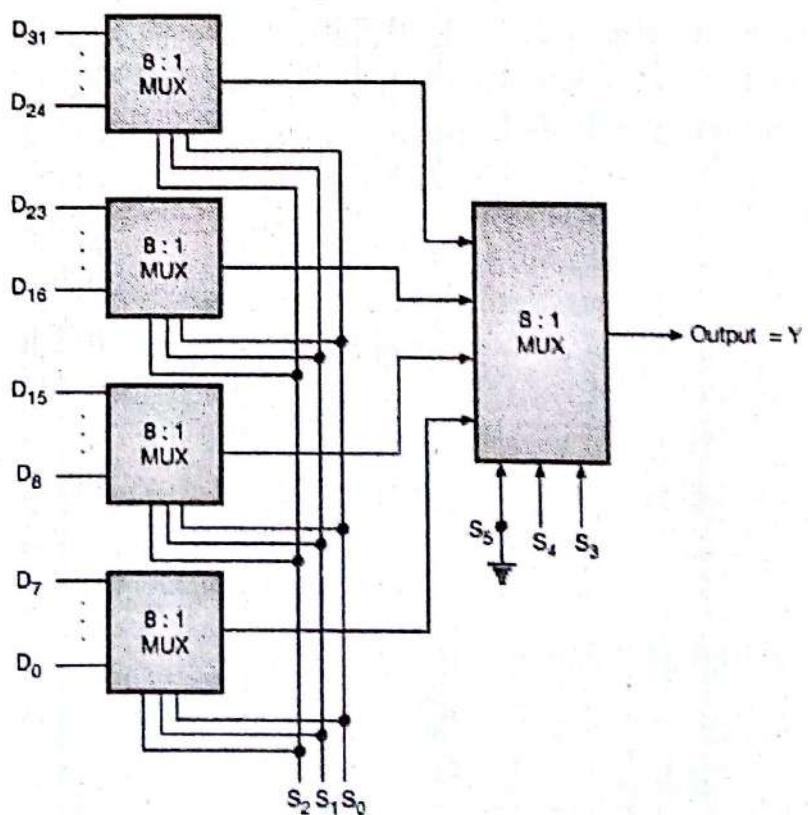
Ans. : Multiplexer tree :

The multiplexers having more number of inputs can be obtained by cascading two or more multiplexers with less number of inputs. This is called as a multiplexes tree.

The truth table is as follows :

Table 1

Select inputs					Output Y
S_4	S_3	S_2	S_1	S_0	
0	0	0	0	0	D_0
⋮					⋮
0	0	1	1	1	D_7
0	1	0	0	0	D_8
⋮					⋮
0	1	1	1	1	D_{15}
1	0	0	0	0	D_{16}
⋮					⋮
1	0	1	1	1	D_{23}
1	1	0	0	0	D_{24}
⋮					⋮
1	1	1	1	1	D_{31}



(C-5272) Fig. Q. 2(b) : 32 : 1 MUX using only 8 : 1 multiplexers

Q. 3(b) Implement single digit BCD adder using 4-bit binary adder IC 7483. Show the design procedure and explain its operation.

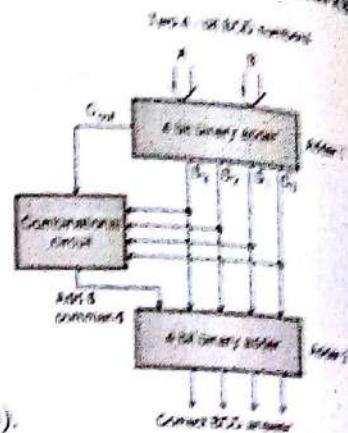
Ans. :

1. Block Diagram of BCD Adder :

From the points stated above, we understand that the 4-bit, BCD adder should consist of the following blocks :

1. A 4-bit binary adder to add the given two 4-bit BCD numbers A and B.
2. A combinational circuit to check if sum is greater than 9 or carry = 1.
3. Another 4-bit binary adder to add six (0110) to the incorrect sum if sum > 9 or carry = 1.

The block diagram of such a BCD adder is shown in Fig. 1-Q. 3(b).



(C-397) Fig. 1-Q. 3(b) : Block diagram of BCD adder

So we have to design the combinational circuit that finds out whether the sum is greater than 9 or Carry = 1.

2. Design of combinational circuit : Please refer Q. 4(b) of Dec. 2014.

Q. 4(a) Explain the concept of comparator. Develop the truth table for 2-bit binary comparator and design it using a suitable decoder and additional gates. (10 Marks)

Ans. :

1. Magnitude comparators :

The block diagram of an n bit digital comparator is shown in Fig. 1-Q. 4(a). Digital comparator is a combinational circuit, which compares the two n-bit binary words applied at its input.

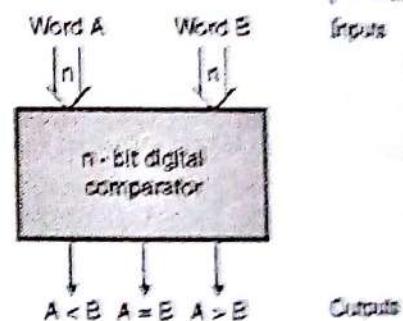
The comparator has three outputs namely $A > B$, $A = B$ and $A < B$. Depending on the result of comparison, one of these outputs will go high.

2. A 2-Bit comparator :

For a 2-bit comparator, each input word A and B is 2 bit long. The truth table of a 2-bit comparator is shown in Table 1.

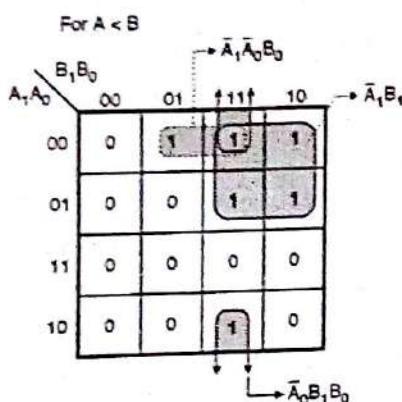
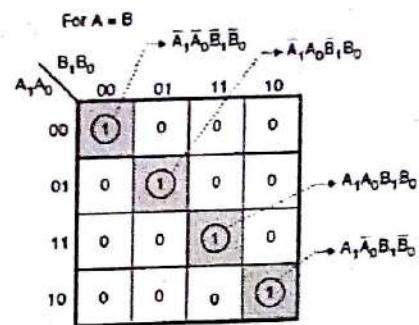
Table 1 : Truth table for a 2-bit comparator

Inputs				Outputs		
A_1	A_0	B_1	B_0	$A < B$	$A = B$	$A > B$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0



(C-403) Fig. 1-Q. 4(a) : Block diagram of an n-bit comparator

Inputs				Outputs		
A_1	A_0	B_1	B_0	$A < B$	$A = B$	$A > B$
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

K-maps :(a) K-map for output $A < B$ (b) K-map for output $A = B$

(C-406) Fig. 2-Q. 4(a)

The K-maps for the three outputs and corresponding simplified expressions are shown in Figs. 2-Q. 4(a)(a), (b) and (c). From these K-maps we get the simplified expressions for the three outputs of comparator are as follows :

$$A < B = \bar{A}_1\bar{A}_0B_0 + \bar{A}_1B_1 + \bar{A}_0B_1B_0$$

For $A > B$: Simplified expression :

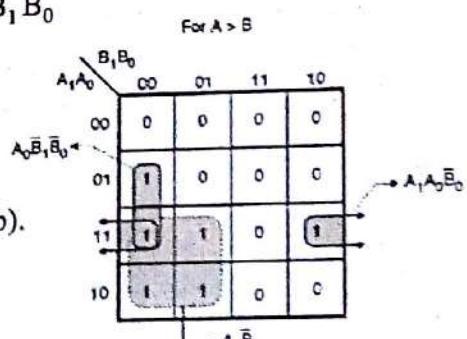
$$A > B = A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0 + A_1\bar{B}_1$$

Simplification for output $A = B$:

Refer the K-map for output $A = B$ shown in Fig. 2-Q. 4(a)(b).

The expression for $A = B$ is given by

$$\begin{aligned} (A = B) &= \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 \\ &\quad + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0 \\ &= \bar{A}_0\bar{B}_0(\bar{A}_1\bar{B}_1 + A_1B_1) + A_0B_0(\bar{A}_1\bar{B}_1 + A_1B_1) \\ &= (\bar{A}_1\bar{B}_1 + A_1B_1)(\bar{A}_0\bar{B}_0 + A_0B_0) \\ \therefore (A = B) &= (A_1 \odot B_1)(A_0 \odot B_0) \end{aligned}$$

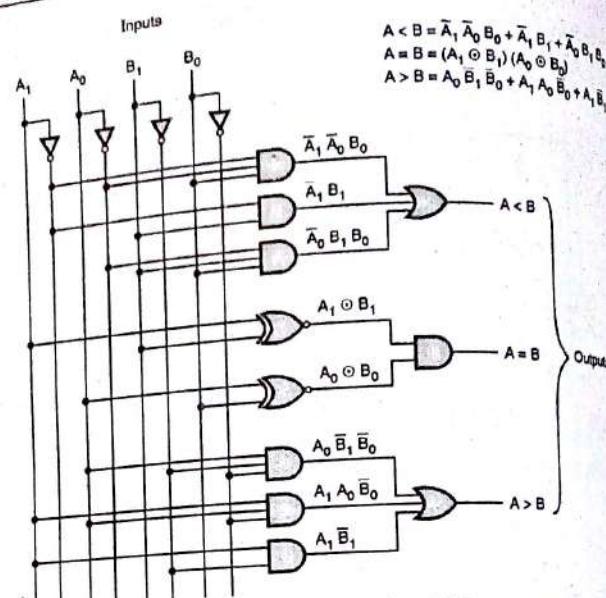
(C-407) Fig. 2-Q. 4(a)(c) : K-map for output $A > B$

where \odot = EX-NOR

DLLA (COMP - MU)

C Logic diagram for a two bit
S comparator :

The logic diagram for the 2-bit digital comparator is shown in Fig. 3-Q. 4(a). This diagram is drawn by referring to the simplified expressions of the outputs.



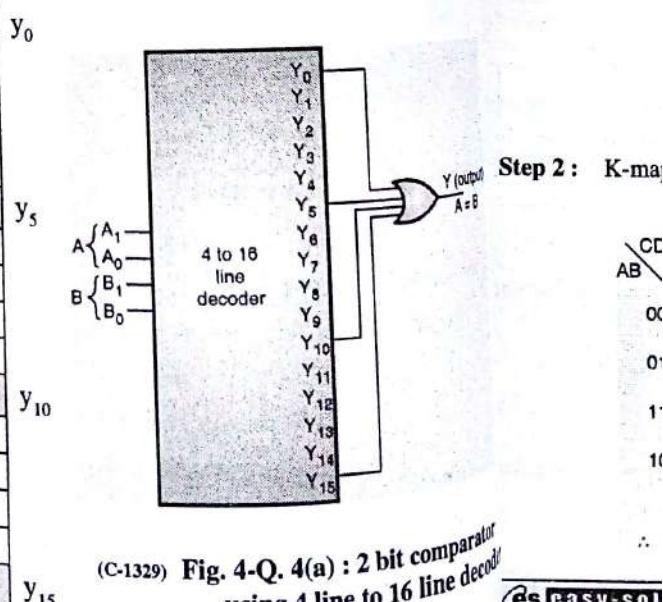
(C-408) Fig. 3-Q. 4(a) : Logic diagram for 2-bit comparator

3. 2 bit comparator using 4 : 16 line decoder :

E The truth table of a 2 bit comparator is as follows :

Table 2 : Truth table of a 2 bit comparator

A		B		Y = equality output
A_1	A_0	B_1	B_0	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
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	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



(C-1329) Fig. 4-Q. 4(a) : 2 bit comparator using 4 line to 16 line decoder

From the truth table the output can be expressed in the SOP form as follows :

$$Y = f(A_1, A_0, B_1, B_0) = \Sigma m(0, 5, 10, 15)$$

The implementation is shown in Fig. 4-Q. 4(a).

- Q. 5(a)** Input to a combinational circuit is a 4-bit binary number. Design the circuit with minimum hardware for the following :

1. Output P = 1 if the number is prime.
2. Output Q = 1 if the number is divisible by 3.

(10 Marks)

Ans. :

Step 1 : Write the truth table :

Prime numbers are : 1, 2, 3, 5, 7, 11 and 13.

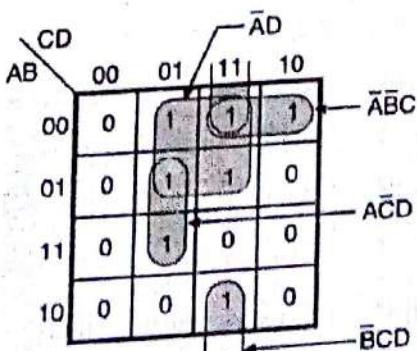
Numbers divisible by 3 are : 0, 3, 6, 9, 12, 15.

Table 1

Decimal	Inputs				Outputs	
	A	B	C	D	P	Q
0	0	0	0	0	0	0
1	0	0	0	1	1	0
2	0	0	1	0	1	0
3	0	0	1	1	1	1
4	0	1	0	0	0	0
5	0	1	0	1	1	0
6	0	1	1	0	0	1
7	0	1	1	1	1	0
8	1	0	0	0	0	0
9	1	0	0	1	0	1
10	1	0	1	0	0	0
11	1	0	1	1	1	0
12	1	1	0	0	0	1
13	1	1	0	1	1	0
14	1	1	1	0	0	0
15	1	1	1	1	0	1

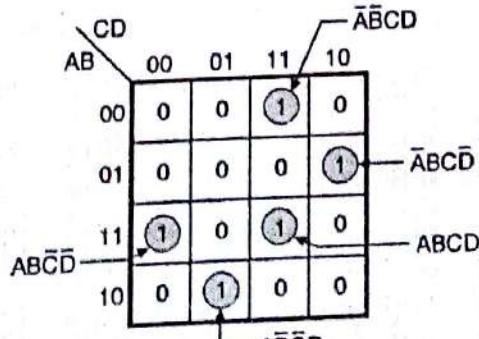
Step 2 : K-maps and simplification :

For output P



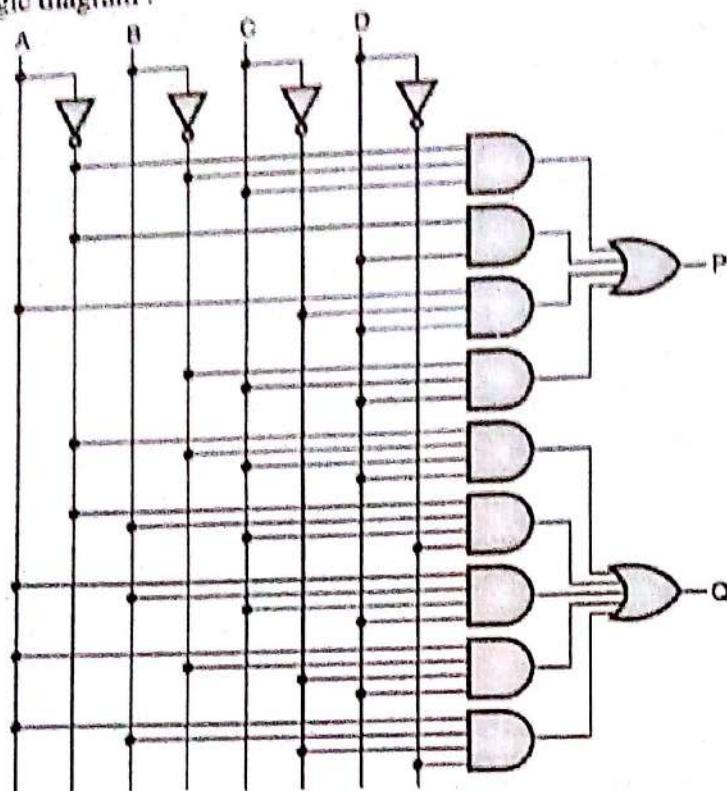
$$\therefore P = \bar{A}\bar{B}C + \bar{A}D + A\bar{C}D + \bar{B}CD$$

For output Q



$$\therefore Q = \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + ABCD + A\bar{B}CD + AB\bar{C}D$$

Step 3 : Draw the logic diagram :



(C-5271) Fig. Q. 5(a) : Logic diagram

Chapter-9 : Flip Flops [Total Marks 6]

Q. 1(e) Differentiate in brief between combinational and sequential circuits.

(2 Marks)

Ans. :

Comparison of combinational and sequential circuits :

Sr. No.	Parameter	Combinational circuits	Sequential circuits
1.	Output depends on	Inputs present at that instant of time.	Present inputs and past inputs/outputs.
2.	Memory	Not necessary	Necessary
3.	Clock input	Not necessary	Necessary
4.	Examples	Adders, subtractors, code converters	Flip flops, shift registers, counters

Q. 1(h) Explain the race around condition in JK flip-flop. State various methods to overcome it.

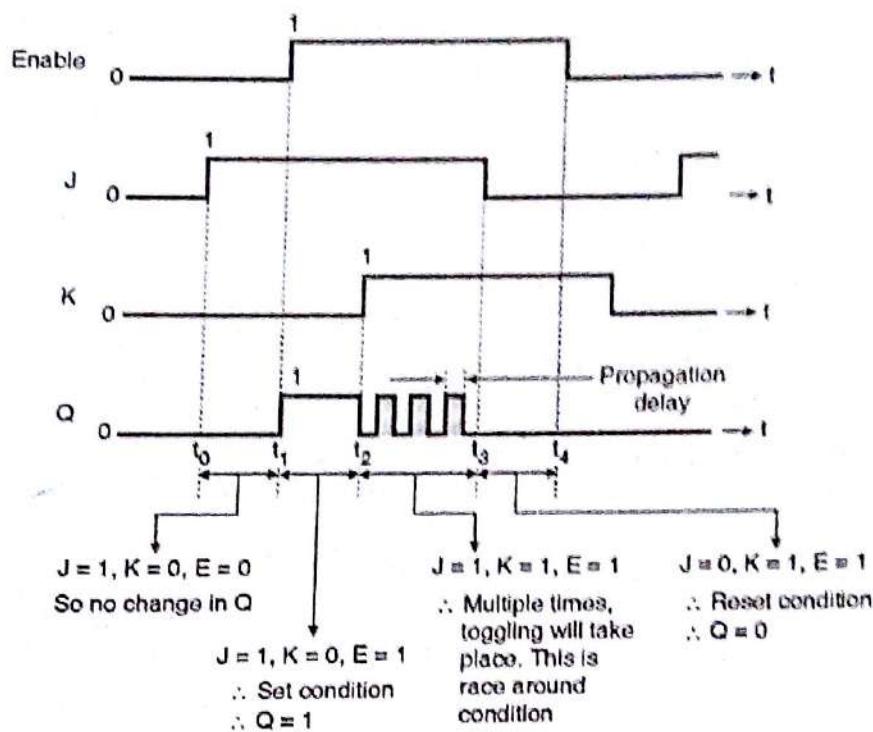
(2 Marks)

Ans. :

Race around condition :

The "Race Around Condition" that we are going to explain occurs when $J = K = 1$ i.e. when the latch is in the toggle mode.

Refer Fig. Q. 1(h) which shows the waveforms for the various modes, when a rectangular waveform is applied to the "Enable" input.



(C-586) Fig. Q. 1(h) : Waveforms for various modes of a JK latch

Interval $t_0 - t_1$

During this interval $J = 1, K = 0$ and $E = 0$.

Hence the latch is disabled and there is no change in Q.

Interval $t_1 - t_2$

During this interval $J = 1, K = 0$ and $E = 1$.

Hence this is a set condition and Q becomes 1.

Interval $t_2 - t_3$: Race around

At instant t_2 , $J = K = 1$ and $E = 1$. Hence the JK latch is in the toggle mode and Q becomes low (0) and $\bar{Q} = 1$.

These changed outputs get applied at the inputs of NAND gates 3 and 4 of the JK latch. Thus the new inputs to Gates 3 and 4 are :

NAND - 3 : $J = 1, E = 1, \bar{Q} = 1$.

NAND - 4 : $K = 1, E = 1, Q = 0$.

Hence R' will become 0 and S' will become 1.

Therefore after a time period corresponding to the propagation delay, the Q and \bar{Q} outputs will change to, $Q = 1$ and $\bar{Q} = 0$.

These changed output again get applied to the inputs of NAND-3 and 4 and the outputs will toggle again.

Thus as long as $J = K = 1$ and $E = 1$, the outputs will keep toggling indefinitely as shown in Fig. Q. 1(h). This multiple toggling in the J-K latch is called as Race Around condition. It must be avoided.

Interval $t_3 - t_4$

During this interval $J = 0$, $K = 1$ and $E = 1$. Hence it is the reset condition.
So Q becomes zero.

How to avoid race around condition ?

The race around condition in JK latch can be avoided by :

Using the edge triggered JK flip flop.

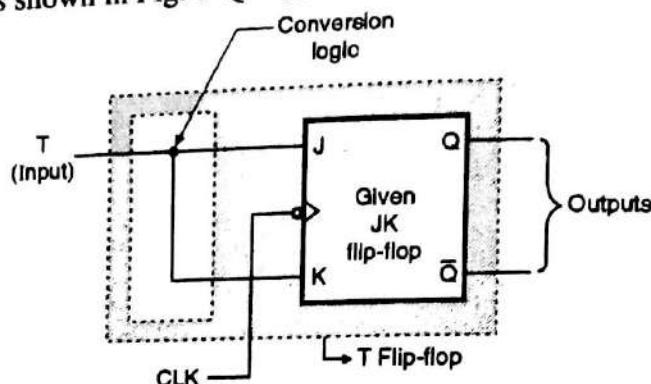
Using the master slave JK flip flop.

Q. 1(i) Convert JK flip-flop into D-flip-flop and T-flip-flop (show only the design without steps)

Ans. :

1. Conversion of JK FF to T FF :

The logic diagram is shown in Fig. 1-Q. 1(i).



(c-616) Fig. 1-Q. 1(i) : Logic diagram for conversion of JK FF to T FF

2. JK Flip Flop to D Flip Flop Conversion : Please refer Q. 6(c)(iii) of May 2014.**Chapter 10 : Counters [Total Marks 22]**

Q. 1(j) What is modulus of the counter ? For MOD-6 counter how many flip-flops are needed

Ans. :

Modulus of the counter (MOD-N counter) :

The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So in general, an n -bit ripple counter is called as modulo- N counter where

$$\text{MOD number} = 2^n$$

So we can conclude that **modulus** of a counter represents the **number of states** through which the counter progresses during its operation.

We can design such modulo counters with the help of the basic ripple counter structure and a combinational logic called reset logic.

Table 1 shows the relation between 2, 3 and 4 bit counters and their modulus.

Table 1

Counter type	Modulus
2 bit up or down	MOD-4
3 bit up or down	MOD-8
4 bit up or down	MOD-16

In MOD-6 number of states is 6 so the number of FFs required is 3.

Q. 4(b) Design MOD-5 synchronous up counter using JK flip-flops with all the design steps. (10 Marks)

Ans. : Please refer Q. 5(a) of May 2015.

Q. 5(b) Draw a circuit diagram for 3-bit asynchronous binary down counter using master-slave JK flip-flops. Show the output of each flip-flop with reference to the clock and justify that the down counting action. Also prove from the timing diagram that the counter is "divide by 8" counter. (10 Marks)

Ans. : 3- Bit asynchronous down counter :

All the counters discussed so far have counted upwards from zero. So they can be called as **up counters**. But the counters which can count in the **downward** direction i.e. from the maximum count to zero are called **down counters**.

The countdown sequence for a 3-bit asynchronous down counter is as follows :

Decimal	Q _C	Q _B	Q _A	Flip-flop outputs
7	1	1	1	
6	1	1	0	
5	1	0	1	
4	1	0	0	
3	0	1	1	
2	0	1	0	
1	0	0	1	
0	0	0	0	

Maximum count

Direction of counting

Recycle

(C-
4871)

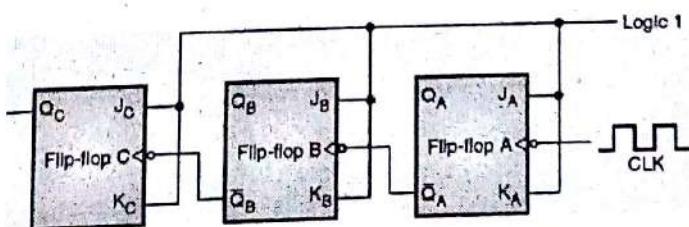
Thus counting takes place as follows :

$$Q_C \ Q_B \ Q_A = 111, 110, 101, 100, 011, 010, 001, 000.$$

From this sequence it is evident that FF-A should toggle at every negative going clock edge but FF-B should change its state only at those instants when Q_A changes from LOW (0) to HIGH (1) and Q_C should change only when Q_B changes from LOW to HIGH.

Thus in a down counter, each FF except the first one (FF-A) should toggle when the output of its preceding flip-flop changes from LOW to HIGH. If all the FFs are negative edge triggered i.e. responding to the negative CLK edge, then we can place an inverter in front of every CLK input or we can drive the CLK input of next FF from the \bar{Q} output of the preceding FF and not from the Q outputs as shown in Fig. 1-Q. 5(b).

A 3-bit asynchronous down counter is shown in Fig. 1-Q. 5(b). The clock input is applied directly to the clock input of FF-A. But \bar{Q}_A is connected to clock of FF-B, \bar{Q}_B to clock of FF-C and so on.



(C-781) Fig. 1-Q. 5(b) : A 3-bit asynchronous down counter

Operation :

Initially let all the flip-flops be in the reset condition.

$$\therefore Q_C \ Q_B \ Q_A = 0 \ 0 \ 0$$

DLDA (COMP - MU)

As soon as the first falling clock pulse arrives, FF-A toggles. So Q_A becomes 1 and \bar{Q}_A changes from 1 to 0. The negative going change in \bar{Q}_A acts as a clock to FF-B. Hence FF-B will change its state. So Q_B becomes 1 and \bar{Q}_B changes from 1 to 0.

This negative going change in \bar{Q}_B acts as a clock to FF-C. Hence FF-C will change its state. Q_C becomes 1 and \bar{Q}_C becomes 0.

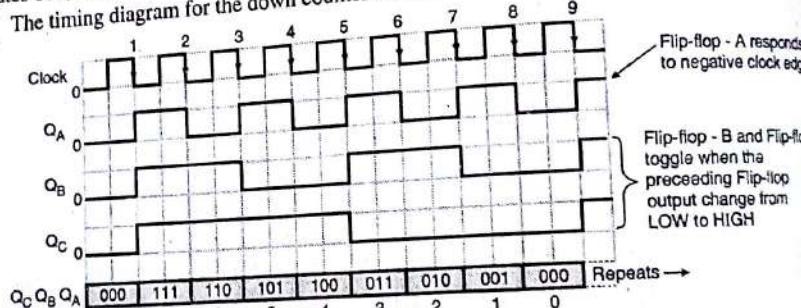
Thus after the first clock pulse the output of counter are,
 $Q_C \ Q_B \ Q_A = 1 \ 1 \ 1$ After the 1st CLK pulse

Corresponding to the second falling clock edge, FF-A toggles. Q_A becomes 0 and \bar{Q}_A becomes 1. This positive going change in \bar{Q}_A does not alter the state of FF-B. So Q_B remains 1 and \bar{Q}_B remains 0. There is no change in the state of FF-C. Hence after the second clock pulse the counter outputs are,

$Q_C \ Q_B \ Q_A = 1 \ 1 \ 0$ After the 2nd CLK pulse

The down counting will thus take place. Similarly the counter will count down to pass through the states 101, 100, 011, 010, 001 and 000. The operation repeats itself thereafter.

The timing diagram for the down counter is shown in Fig. 2-Q. 5(b).

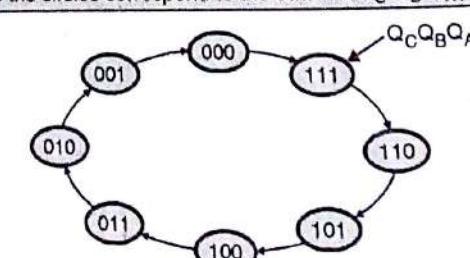


(C-782) Fig. 2-Q. 5(b) : Timing diagram of a 3-bit down counter

State diagram :

The state diagram of the 3-bit down counter is shown in Fig. 3-Q. 5(b) :

Note : The numbers inside the circles correspond to the state of $Q_C \ Q_B \ Q_A$.



(C-783) Fig. 3-Q. 5(b) : State diagram of a 3 bit ripple down counter

Chapter 11 : Registers [Total Marks 20]

Q. 6(a) What is shift register ? Explain 4-bit bidirectional shift register. (10 Marks)

Ans. : Please refer Q. 5(a) of Dec. 2014.

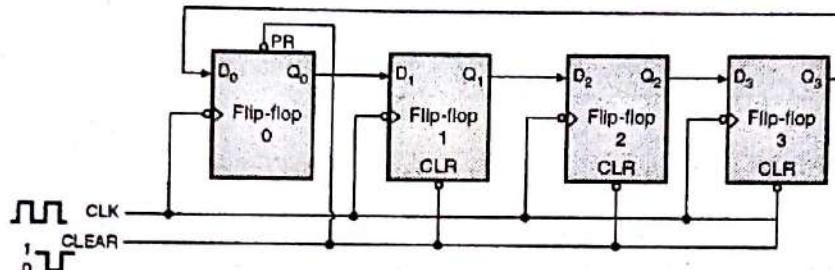
Q. 6(b) Draw and explain the working of 4-bit ring counter with timing diagram. (10 Marks)

Ans. : Ring counter :

Fig. 1-Q. 6(b) shows a typical application of shift registers called Ring Counter.

The connections reveal that they are similar to the connections for shift right operation, except for one change.

Output of FF-3 is connected to data input D_0 of FF-0. Ring counter is a special type of shift register.



(C-871) Fig. 1-Q. 6(b) : A four bit ring counter

Operation :

Initially a low clear.(CLR) pulse is applied to all the flip-flops. Hence FF-3, FF-2 and FF-1 will reset but FF-0 will be preset. So the outputs of the shift register are :

$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 0 \ 0 \ 0 \ 1.$$

Now the clear terminal is made inactive by applying a high level to it. The clock signal is then applied to all the flip-flops simultaneously. Note that all the flip-flops are negative edge triggered.

On the first negative going CLK edge :

As soon as the first falling edge of the clock hits, only FF-1 will be set because $Q_0 = D_1 = 1$.

The FF-0 will reset because $D_0 = Q_3 = 0$ and there is no change in the status of FF-2 and FF-3.

Hence after the first clock pulse the outputs are

$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 0 \ 0 \ 1 \ 0.$$

On the second falling edge of clock :

At the second falling edge of the clock, only FF-2 will be set because $D_2 = Q_1 = 1$.

FF-1 will reset since $D_1 = Q_0 = 0$. There is no change in status of FF-3 and FF-0.

So after the second clock pulse the outputs are,

$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 0 \ 1 \ 0 \ 0.$$

Similarly after the third clock pulse the outputs are,

$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 1 \ 0 \ 0 \ 0.$$

And after the fourth one the outputs are,

$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 0 \ 0 \ 0 \ 1.$$

These are the outputs from where we started. Hence the operation repeats from this point onwards.

Dec 20

Number of output states :

The number of output states for a ring counter will always be equal to the number of flip-flops. So for a 4-bit ring counter the number of states is equal to 4. The operation of a four bit ring counter is summarized in Table 1.

(C-1310) Table 1 : Summary of operation of a ring counter

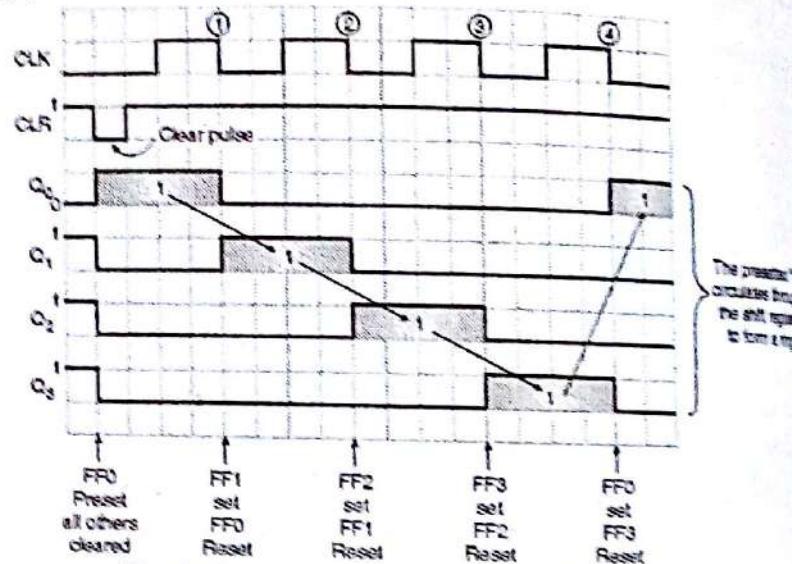
CLR	CLK	Q_0	Q_1	Q_2	Q_3
1	X	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0

← FF-0 Preset, others cleared

The presetted 1 follows a circular path to form a ring

Waveforms for the ring counter :

The waveforms for the 4-bit ring counter are as shown in Fig. 2-Q. 6(b). These waveforms clearly show that the presetted "1" shifts one bit per clock cycle and forms a ring. Hence the name ring counter.



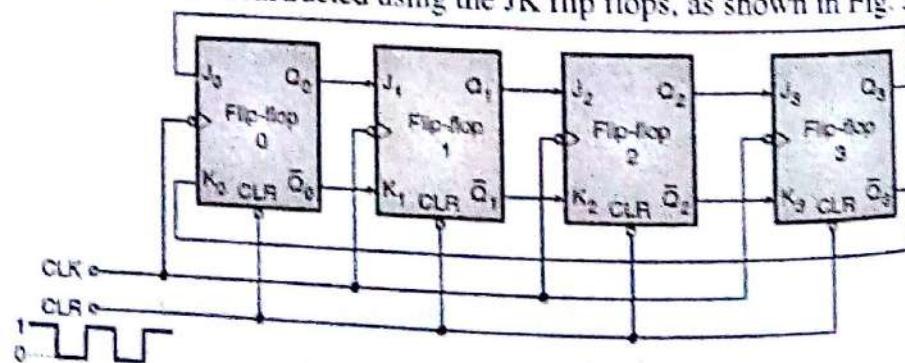
(C-1310) Fig. 2-Q. 6(b) : Waveforms of a four bit ring counter

Applications of ring counter :

Ring counters are used in those applications in which several operations are to be controlled sequentially. For example in resistance welding the operations called squeeze, hold, weld etc. are to be performed sequentially. We can use a ring counter to initiate these operations.

Ring counter using JK Flip Flop :

A ring counter can also be constructed using the JK flip flops, as shown in Fig. 3-Q. 6(b).



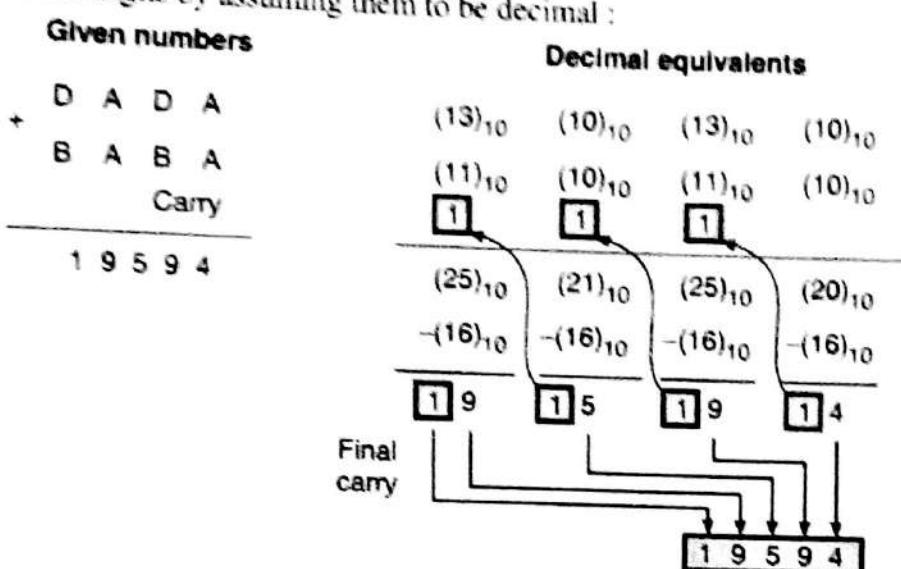
(C-1361) Fig. 3-Q. 6(b): Ring counter using JK flip flops

Q. 1(b) Perform hexadecimal arithmetic operation : DADA + BABA

Ans. :

(Appendix A, 2 Marks)

Step 1 : Add the digits by assuming them to be decimal :



(C-5261)

$$\therefore (DADA)_{16} + (BABA)_{16} = (19594)_{16}$$

...Ans.



Chapter 1 : Number Systems [Total Marks - 03]

Q. 1(a) Convert $(532.125)_8$ into decimal, binary and hexadecimal.

(3 M)

Ans. : Conversion to decimal :

1. Conversion to decimal :

$$\begin{aligned}
 N = & \boxed{5} \quad \boxed{3} \quad \boxed{2} \quad \cdot \quad \boxed{1} \quad \boxed{2} \quad \boxed{5} \\
 & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
 & (5 \times 8^2) + (3 \times 8^1) + (2 \times 8^0) + (1 \times 8^{-1}) + (2 \times 8^{-2}) + (5 \times 8^{-3}) \quad (\text{C.M.D})
 \end{aligned}$$

$$= 320 + 24 + 2 + 0.125 + 0.03125 + 0.009765 = 346.166015$$

$$\therefore (532.125)_8 = (346.166015)_{10}$$

2. Conversion to binary :

$$\begin{aligned}
 N = & \boxed{5} \quad \boxed{3} \quad \boxed{2} \quad \cdot \quad \boxed{1} \quad \boxed{2} \quad \boxed{5} \\
 & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
 & 101 \quad 011 \quad 010 \quad \cdot \quad 001 \quad 010 \quad 101
 \end{aligned}$$

$$\therefore (532.125)_8 = (101011010 \cdot 001010101)_2$$

3. Conversion to hexadecimal :

Obtained binary number : 101011010 · 001010101

Add 3 zeros to extreme left and right side.

$$\begin{array}{c}
 \text{Group of 4 bits : } \boxed{0001} \quad \boxed{0101} \quad \boxed{1010} \quad \cdot \quad \boxed{0010} \quad \boxed{1010} \quad \boxed{1000} \\
 \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow
 \end{array}$$

$$\begin{array}{c}
 \text{Hex number : } 1 \quad 5 \quad A \quad \cdot \quad 2 \quad A \quad 8
 \end{array}$$

$$\therefore (532.125)_8 = (15A \cdot 2A8)_{16}$$

Chapter 2 : Binary Arithmetic [Total Marks : 04]

Q. 1(c) Subtract using 1's and 2's complement method $(56)_{10} - (76)_{10}$.

(4 M)

Ans. :

Using 1's complement method :

Step 1 : Obtain 1's complement of $(76)_{10}$:

$$\begin{array}{r}
 \text{Binary} \quad (76)_{10} \xrightarrow{\text{1's complement}} (1001100)_2
 \end{array}$$

Step 2 : Add $(56)_{10}$ and 1's complement of $(76)_{10}$:

$$\begin{array}{r}
 (56)_{10} : \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \\
 1's \text{ complement of } (76)_{10} : \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \\
 \text{Carry : } \quad 1 \quad 1 \\
 \hline
 \text{Final carry} \rightarrow \boxed{0} \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1
 \end{array}$$

Q. 1(b)

Ans. :

Step 1 :

Step 2 :

es easy

As the final carry is 0, the answer is negative and in its 1's complement form. So convert answer into its true form as follows :

$$\text{Answer : } (1101011)_2 \xrightarrow{\text{Invert}} (0010100)_2 = (20)_{10}$$

$$\therefore (56)_{10} - (76)_{10} = -(20)_{10}$$

...Ans.

Using 2's complement form :

Step 1 : Obtain 2's complement of $(76)_{10}$:

$$(76)_{10} \xrightarrow{\text{Binary}} (1001100)_2 \xrightarrow{\text{1's complement}} (0110011)_2 \xrightarrow{\text{add 1}} (0110100)_2$$

Step 2 : Add $(56)_{10}$ and 2's complement of $(76)_{10}$:

$(56)_{10}$:	0 1 1 1 0 0 0	(C-5518)
2's complement of $(76)_{10}$:	0 1 1 0 1 0 0	
Carry :	1 1	
Final carry →	0 1 1 0 1 1 0 0	

As the final carry is 0, the answer is negative and in its 2's complement form.

Step 3 : Convert the answer into its true form :

$$\begin{array}{r} \text{Answer : } 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \\ \text{Subtract 1 : } \quad \quad \quad \quad \quad \quad \quad \quad 1 \\ \hline \underbrace{1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1} \\ \text{Invert} \\ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 = (20)_{10} \end{array} \quad \text{(C-5519)}$$

$$\therefore (56)_{10} - (76)_{10} = -(20)_{10}$$

...Ans.

Chapter 3 : Codes [Total Marks : 05]

Q. 1(b) Convert $(47.3)_7$, into BCD, excess-3 and gray code. (3 Marks)

Ans. :

Step 1 : Convert base 7 to decimal :

$$\begin{array}{l} \text{Given number = } \boxed{4} \ \boxed{7} \ . \ \boxed{3} \\ \text{Decimal number = } (4 \times 7^1) + (7 \times 7^0) + (3 \times 7^{-1}) \\ = 28 + 7 + 0.4285 = 35.4285 \\ \therefore (47.3)_7 = (35.4285)_{10} \end{array} \quad \text{(C-5512)}$$

Step 2 : Conversion to BCD :

$$\begin{array}{l} \text{Decimal : } \boxed{3} \ \boxed{5} \ . \ \boxed{4} \ \boxed{2} \ \boxed{8} \ \boxed{5} \\ \downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow \\ \text{BCD : } 0011 \ 0101 \ . \ 0100 \ 0010 \ 1000 \ 0101 \\ \therefore (47.3)_7 = (0011 \ 0101 \ . \ 0100 \ 0010 \ 1000 \ 0101)_{\text{BCD}} \end{array} \quad \text{...Ans.}$$

Step 3 : Conversion to Excess - 3 :

Decimal

↓

↓

↓

↓

↓

↓

Excess-3 equivalent

↓

↓

↓

↓

↓

↓

3	5	.	4	2	8	5
↓	↓	.	↓	↓	↓	↓
6	8	.	7	5	11	8
0110	1000	.	0111	0101	1011	1000

$$\therefore (47.3)_4 = (0110 \ 1000 \cdot 0111 \ 0101 \ 1011 \ 1000)_{\text{Excess-3}}$$

Step 4 : Conversion to binary :

Convert the integer :

2	35	1
2	17	1
2	8	0
2	4	0
2	2	0
2	1	1
0		

Convert the fractional part :

Base Product Carry

0.4285	$\times 2 = 0.857$	0
0.857	$\times 2 = 1.714$	1
0.714	$\times 2 = 1.428$	1
0.428	$\times 2 = 0.856$	0
0.856	$\times 2 = 1.712$	1

MSB ↓ LSB

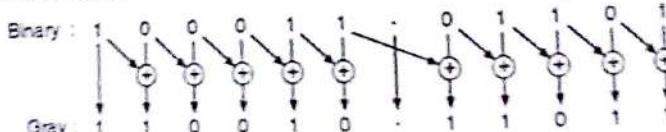
$$\therefore (35)_{10} = (100011)_2$$

$$\therefore (0.4285)_{10} = (0.01101)_2$$

(C-5513)

$$\therefore (35.4285)_{10} = (100011.01101)_2$$

Step 5 : Conversion to gray :



$$\therefore (47.3)_7 = (100011-01101)_2 = (110010 \cdot 11011)_{\text{gray}}$$

Q. 1(d) Obtain odd parity Hamming code for 1011.

(2 Marks)

Ans. :

Data word : 1011

Take

Format of 7-bit hamming code is as follows :

7	6	5	4	3	2	1
1	0	1	P ₄	1	P ₂	P ₁

This
gates.

Step 1 : Decide P₁ :

Consider bits 1, 3, 5 and 7. They are 111 P₁
So for odd parity set P₁ = 0

$$\therefore P_1 = 0$$

Step 2 : Decide P_2 :

Consider bits 2, 3, 6 and 7. They are 101 P₂

So for odd parity set $P_2 = 1$

$$\therefore P_2 = 1$$

Step 3 : Decide P_4 :

Consider bits 4, 5, 6 and 7. They are 101 P₄

So for odd parity set $P_4 = 1$

$$\therefore P_4 = 1$$

Obtained codeword is as follows :

$$\begin{array}{ccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ \text{Codeword} & = & \boxed{1} & 0 & 1 & 1 & 1 & 0 \end{array}$$

Chapter 4 : Logic Gates and Boolean Algebra [Total Marks : 04]

Q. 1(e) Implement Ex-OR gate using NOR gate only.

(2 Marks)

Ans. :

Ex-OR using only NOR :

$$Y = A \oplus B = \bar{A}\bar{B} + A\bar{B} \quad \dots \text{EX-OR gate}$$

$$\text{Let } X = \bar{A}\bar{B} \text{ and } Z = A\bar{B}$$

$$\therefore Y = X + Z$$

$$Y = \overline{\overline{X} + Z} \quad \dots \text{Double inversion.}$$

$$\therefore Y = \overline{\overline{X}} \cdot \overline{\overline{Z}} \quad \dots \text{De Morgan's theorem}$$

$$= (\overline{\overline{A}\bar{B}}) \cdot (\overline{\overline{A}\bar{B}})$$

$$\text{But } \overline{\overline{AB}} = (A + \bar{B}) \text{ and } \overline{\overline{AB}} = (\bar{A} + B) \quad \dots \text{De Morgan's theorem}$$

$$\therefore Y = (A + \bar{B}) \cdot (\bar{A} + B)$$

$$\therefore Y = (\overline{\overline{A + \bar{B}}}) + (\overline{\overline{\bar{A} + B}}) \quad \dots \text{De Morgan's theorem}$$

Take double inversion of RHS we get,

$$Y = \overline{\overline{(A + \bar{B}) + (\bar{A} + B)}}$$

This is the required expression. Fig. 1 shows the realization of EX-OR gate using only NOR gates.

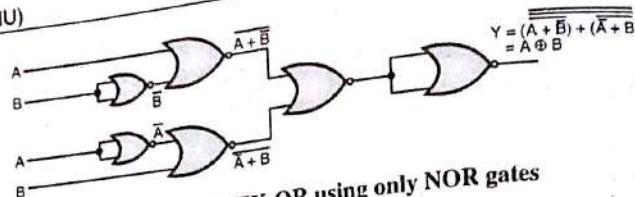
DLDA (COMP - MU)

10-5

May 2015

DLDA (COMP)

Step 2 : V



(B-515) Fig. 1 : EX-OR using only NOR gates

Q. 1(g) State and prove De-Morgan's theorem.
Ans. : Please refer Q. 1(a) of Dec. 2013.

(2 Marks)

Chapter 5 : Logic Minimization and Reduction Techniques [Total Marks : 10]

Q. 2(a) Reduce equation using Quine McCluskey method and realize circuit using basic gates.
 $F(A, B, C, D) = \sum m(1, 3, 7, 9, 10, 11, 13, 15)$
Ans. : Please refer Q. 2(a) of May 2015.

(10 Marks)

Chapter 6 : Logic Families [Total Marks : 05]

Q. 4(a) Compare TTL and CMOS logic.
Ans. : Please refer Q. 1(f) of Dec. 2015.

Step 3 : Re
The
Fig. 2(a).

Chapter 7 : Combinational Logic Design [Total Marks : 40]

Q. 2(b) Design 8 bit BCD adder.
Ans. : Please refer Q. 4(b) of May 2015.

(10 Marks)
Q. 3(b) Irr
F
Ans. :
Step 1 : W

Q. 3(a) Design a logic circuit to convert gray to BCD code.

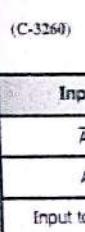
(10 Marks)

Ans. :

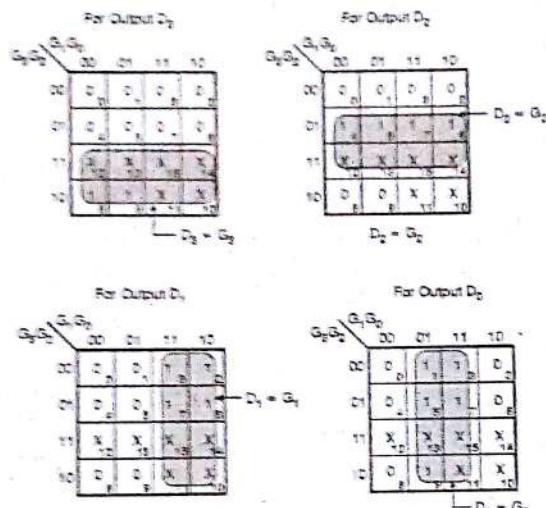
Step 1 : Write the truth table relating gray and BCD codes :

Decimal	Gray inputs				BCD outputs			
	G ₃	G ₂	G ₁	G ₀	D ₃	D ₂	D ₁	D ₀
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	1	0	0	1	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	1	1	0	0	1	0	0	0
9	1	1	0	1	1	0	0	1

Inp
A
B
Input to



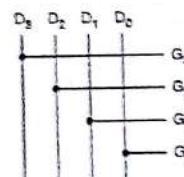
Step 2 : Write K-maps for each output and get simplified equation :



(C-1789) Fig. 2

Step 3 : Realization :

The gray to BCD converter is shown in Fig. 2(a).



(C-1710) Fig. 2(a)

Q. 3(b) Implement the following using only one 8 : 1 MUX and few gates.

$$F(A, B, C, D) = \sum m(0, 3, 5, 7, 9, 13, 15)$$

(5 Marks)

Ans. :

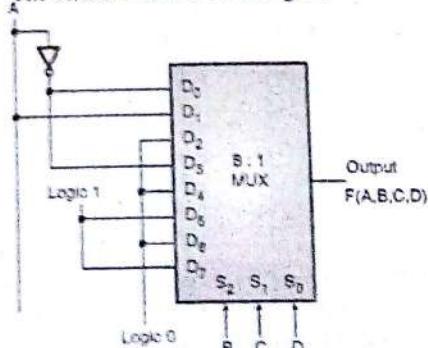
Step 1 : Write the design table :

(C-3246)

Inputs	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
A	0	1	2	3	4	5	6	7
B	0	1	10	11	12	13	14	15
Input to MUX	A	A	0	A	0	1	0	1

Step 2 : Implement using an 8 : 1 multiplexer :

The circuit is as shown in Fig. 3.



(C-3281) Fig. 3

DLDA (COMP - MU)

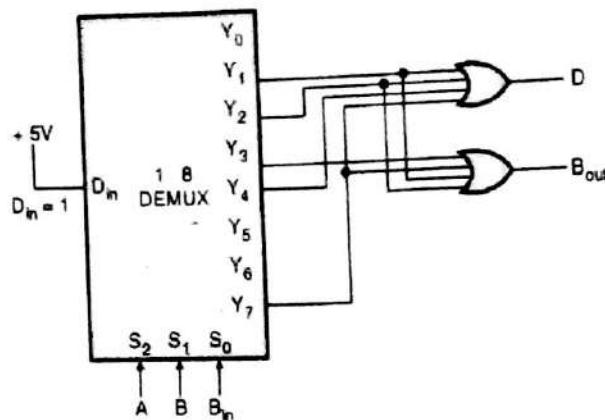
Q. 3(c) Design a full adder circuit using half adders and some gates.
Ans. : Please refer Q. 1(c) of May 2014.

Q. 4(b) Implement full subtractor using demultiplexer.
Ans. :

The full subtractor has three inputs A, B and B_{in} and two outputs namely difference D and borrow out (B_{out}). The truth table of a full subtractor is as follows :

Table 1 : Truth table of a full subtractor

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



(C-464) Fig. 4 : Full subtractor using 1 : 8 demux

From the truth table we can express the difference and borrow out outputs in the standard forms as,

$$D = f(A, B, B_{in}) = \sum m(1, 2, 4, 7)$$

$$\text{and } B_{out} = f(A, B, B_{in}) = \sum m(1, 2, 3, 7)$$

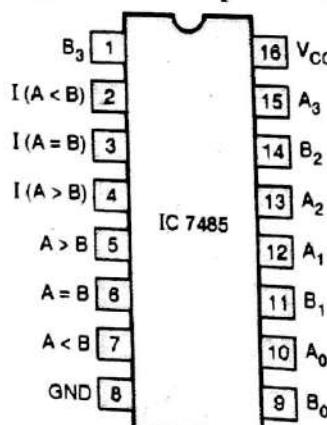
Connect D_{in} to logic 1 permanently and connect A, B and B_{in} to the select inputs S_2, S_1, S_0 respectively as shown in Fig. 4. As D_{in} is connected to 1, we get the required minterms at the Dem outputs. We have to OR the required minterms to obtain the D and B_{out} outputs as shown in Fig. 4.

Q. 6(d) Write short note on 4-bit magnitude comparator

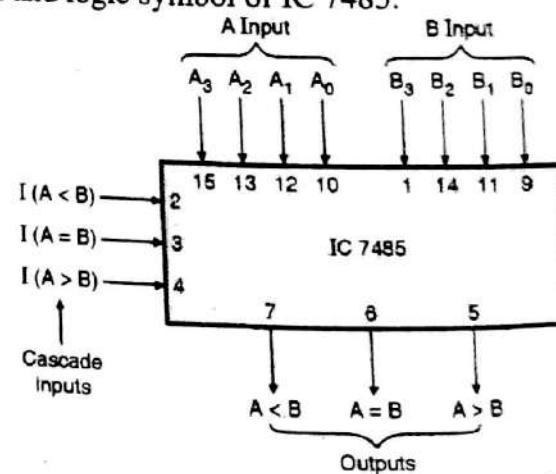
(5 Min)

Ans. :

IC 7485 is a four bit comparator in the integrated circuit form. It compares two 4-bit words A ($A_3 - A_0$) and B ($B_3 - B_0$). It is possible to cascade more than one IC 7485 to compare words of any length. Fig. 5 shows the pin configuration and logic symbol of IC 7485.



(a) Pin configuration



(b) Logic diagram of IC 7485

(C-409) Fig. 5

Pin names and their functions :

The pin names and their functions are as listed in Table 2.

Table 2

Pin name	Pin number	Function
A ₀ to A ₃	10, 12, 13, 15	Binary input (operand 1) Active high
B ₀ to B ₃	9, 11, 14, 1	Binary input (operand 2) Active high
I(A < B)	2	
I(A = B)	3	
I(A > B)	4	
A < B	7	
A = B	6	
A > B	5	

Truth table of IC 7485 :

Table 3

Comparing inputs				Cascading inputs			Outputs		
A ₃ , B ₃	A ₂ , B ₂	A ₁ , B ₁	A ₀ , B ₀	I _{A > B}	I _{A < B}	I _{A = B}	Y _{A > B}	Y _{A < B}	Y _{A = B}
A ₃ > B ₃	X	X	X	X	X	X	H	L	L
A ₃ < B ₃	X	X	X	X	X	X	L	H	L
A ₃ = B ₃	A ₂ > B ₂	X	X	X	X	X	H	L	L
A ₃ = B ₃	A ₂ < B ₂	X	X	X	X	X	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ > B ₁	X	X	X	X	H	L	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ < B ₁	X	X	X	X	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ > B ₀	X	X	X	H	L	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ < B ₀	X	X	X	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	H	L	L	H	L	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	L	H	L	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	L	L	L	H	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	X	X	H	L	L	H
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	H	H	L	L	L	L

Function table of IC 7485 : Table 4 shows the function table for IC 7485.

Table 4

Comparing inputs	Cascading inputs			Outputs		
	I _{A < B}	I _{A = B}	I _{A > B}	Y _{A < B}	Y _{A = B}	Y _{A > B}
A < B	X	X	X	1	0	0
A = B	0	0	0	1	0	1
	0	0	1	0	0	1
	1	0	0	1	0	0
	1	0	1	0	0	0
	X	1	X	0	1	0
A > B	X	X	X	0	0	1

The function table shown in Table 4 tells us that if $I(A < B)$ or $I(A > B)$ is equal to 1 and other two cascading inputs are at logic 0 then the corresponding output ($A < B$ or $A > B$) will be active. But if $A = B$, then the corresponding output will be active if and only if $I(A = B)$ is at logic 1 irrespective of the other two inputs.

General description :

As shown in the truth table (Table 3); the MSB are compared first i.e. A_3 is compared with B_3 . Depending on whether $A_3 > B_3$ or $A_3 < B_3$ the outputs $Y(A > B)$ or $Y(A < B)$ are activated. If $A_3 = B_3$, then the next MSB bits B_2 and A_2 are compared. If $A_2 = B_2$ then comparison of A_1 and B_1 is performed and so on. If $A_3 A_2 A_1 A_0 = B_3 B_2 B_1 B_0$ then the IC 7485 will check the cascading inputs. Decision making will then take place as follows:

Case 1 : If $I(A > B) = 1$ and $I(A = B) = I(A < B) = 0$

This indicates that at the previous stage LSB of A is greater than LSB of B.

Case 2 : If $I(A = B) = 1$

If $I(A = B) = 1$ then the chip will not check the status of $I(A < B)$ and $I(A > B)$ inputs. It will understand that the LSB of A is equal to LSB of B.

A square
the name mu

Q. 5(b) C

Ans. : Please

Q. 6(c) V

Ans. : Please

Q. 5(a) D

Ans. :

Step 1 : D

(5) The

Chapter 8 : Multivibrators [Total Marks - 05]

Q. 6(e) Write short note on : Multivibrators

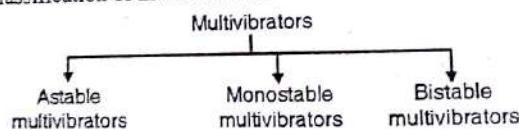
Ans. :

Definition :

Multivibrator is an electronic circuit which can have no, one or two stable states of operation. Depending on the number of stable states we can classify the multivibrators as shown in Fig. 6.

Classification of Multivibrators :

Fig. 6 shows the classification of multivibrators.



(N-197) Fig. 6 : Classification of multivibrators

Astable multivibrator :

The astable multivibrator output does not have any stable state. Its output changes its state from high to low and low to high repeatedly. No external trigger pulses are required to be used.

Monostable multivibrator :

The monostable multivibrator output has only one stable state. Mono means one. External trigger pulse needs to be used for monostable multivibrator.

Bistable multivibrator :

It is a multivibrator circuit, the output of which has two stable states. We have to apply external inputs in order to change the existing state of output. Bistable multivibrator is also called as a flip-flop. Bistable multivibrator is an oscillator that produces non-sinusoidal waveforms such as a square wave.

Step 2 : W

Table 5 sh

Table 5 : T

State	Flip-flop
0	0
1	0
2	0
3	0
4	1
5	1
6	1
7	1

A square wave consists of harmonics or multiple variations of fundamental frequency. Therefore the name multivibrators.

Chapter 9 : Flip Flops [Total Marks : 15]

Q. 5(b) Convert SR flip flop to JK flip flop and D flip flop.

(10 Marks)

Ans. : Please refer Q. 6(c) of May 2014.

Q. 6(c) Write short note on : State table

(5 Marks)

Ans. : Please refer Q. 6(a) of Dec. 2014.

Chapter 10 : Counters [Total Marks - 15]

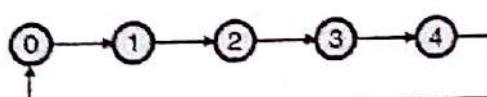
Q. 5(a) Design mod 5 asynchronous up counter.

(10 Marks)

Ans. :

Step 1 : Draw the state diagram :

The state diagram of MOD-5 ripple counter is as shown in Fig. 7 (a).



(C-794) Fig. 7(a) : State diagram of a MOD-5 ripple counter

Step 2 : Write truth table for the reset logic :

Table 5 shows the truth table for the reset logic.

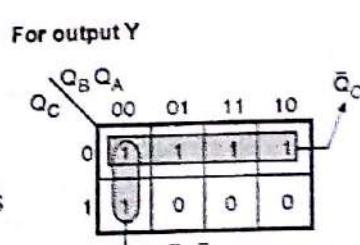
Table 5 : Truth table for the reset logic

State	Flip-flop outputs			Output Y of reset logic
	Q _C	Q _B	Q _A	
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

Valid states

Invalid states

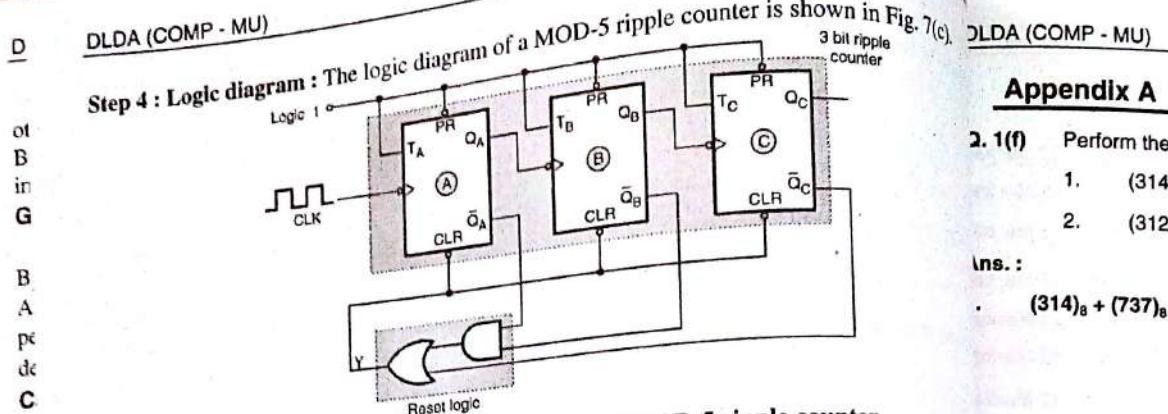
Step 3 : K-map :



Expression for Y
 $Y = \bar{Q}_C + \bar{Q}_B \bar{Q}_A$

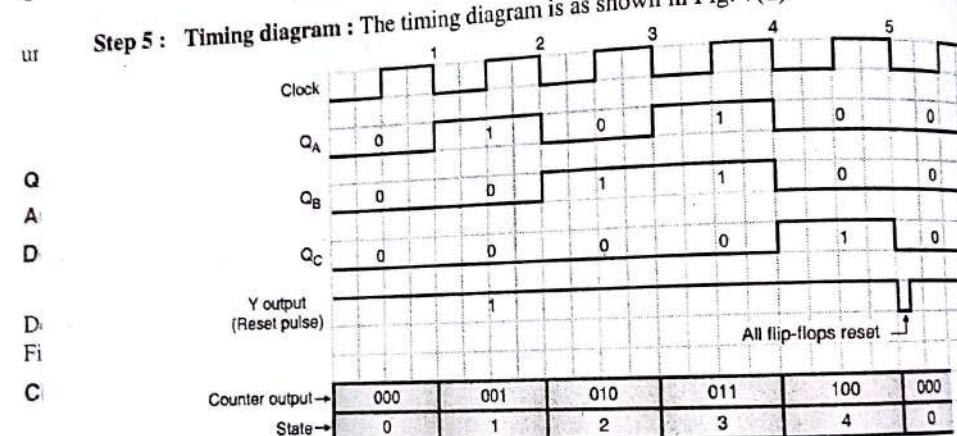
(C-795) Fig. 7(b) : K-map
and simplification

The K map is as shown in Fig. 7(b). The states 0 through 4 are valid states and the output Y of reset logic (Y) is inactive (1) for them. The states 5, 6 and 7 are invalid states. If counter enters into any one of these states that Y = 0 (active) and will reset all the flip-flops.



(C-76) Fig. 7(c) : Logic diagram of MOD-5 ripple counter

Step 5 : Timing diagram : The timing diagram is as shown in Fig. 7(d).



(C-77) Fig. 7(d) : Timing diagram of MOD-5 ripple counter

Q. 6(b) Write short note on : Decade counter

Ans. : Please refer Q. 6(d) of Dec. 2013.

Chapter 11 : Registers [Total Marks - 10]

A: **Q. 4(c)** Explain 4 bit universal shift register.

Ans. : Please refer Q. 3(a) of May 2014.

Chapter 13 : Introduction to VHDL [Total Marks - 05]

P: **Q. 6(a)** Write short note on : VHDL

Ans. : Please refer Q. 6(b) of Dec. 2014.

Appendix A

2. 1(f) Perform the

1. (314)

2. (312).

Ins. :

(314)₈ + (737)₈

∴ (314)₈ +

(312.40)₅ +

∴ (312.40)₅ + (21)

Appendix A : Octal and Hexadecimal Arithmetic [Total Marks - 04]

Q. 1(f) Perform the following operations without changing the base : (4 Marks)

- $$\begin{array}{r} 1. \quad (314)_8 + (737)_8 \\ 2. \quad (312.40)_5 + (214.33)_5 \end{array}$$

Ans. :

$$1. \quad (314)_8 + (737)_8 :$$

$$\begin{array}{r}
 & 3 & 1 & 4 \\
 + & 7 & 3 & 7 \\
 \hline
 \text{Carry} & 1 & 1 & \\
 & \downarrow & \downarrow & \\
 & 10 & 5 & 11 \\
 & -8 & & -8 \\
 \hline
 & 12 & & 13 \\
 & \circled{1} & & \circled{1} \\
 & \downarrow & & \downarrow \\
 & 1 & 2 & 5 & 3 \\
 & \circled{1} & & & \circled{1} \\
 & \downarrow & & \downarrow & \downarrow \\
 & 1 & 2 & 5 & 3
 \end{array}$$

$$\therefore (314)_8 + (737)_8 = (1253)_8$$

ANS

$$2. \quad (312.40)_5 + (214.33)_5 :$$

$$\begin{array}{r}
 & + & 3 & 1 & 2 & \cdot & 4 & 0 \\
 & 2 & 1 & 4 & \cdot & 3 & 3 \\
 \text{Carry : } & \boxed{1} & \boxed{1} & \boxed{1} & & & \\
 \hline
 & 5 & 3 & 7 & \cdot & 7 & 3 \\
 & -5 & & -5 & & -5 & \\
 \hline
 & \underline{\boxed{1}0} & & \underline{\boxed{1}2} & & \underline{\boxed{1}2} & \\
 & 1 & 0 & 3 & 2 & \cdot & 2 & 3
 \end{array}$$

(C-5510)

$$\therefore (312.40)_5 + (214.33)_5 = (1032.23)_5$$

175



May 2015

- Q. 1 (a) Convert $(121.2)_3$ into base 10.
 (b) Represent $(52)_{10}$ into Excess - 3 code and Gray code.
 (c) Find the one's complement and two's complement of $(57)_{10}$.
 (d) Realize $y = AB + \bar{A}\bar{B}$ using NAND gates only.
 (e) Obtain hamming code for 1011.
 (f) Convert $(126)_{10}$ to Octal, Hexcode.
 (g) State De-Morgans law.
 (h) Convert $(214.32)_{10}$ to binary.
 (i) Perform binary subtraction using 2's complement for $(62)_{10}$ and $(99)_{10}$.

Q. 2. (a) Minimize the logic function using Quine-Mc Cluskey method.

$$f(A,B,C,D) = \Sigma m(1, 3, 7, 9, 10, 11, 13, 15)$$

(b) Implement the following expression using single 4 : 1 Mux.
 $f = (A, B, C, D) = \Sigma m(0, 1, 2, 4, 6, 9, 12, 14)$

Q. 3 (a) Design a 4-input (A, B, C, D) digital circuit that will give at its output (X) a logic 1 if the binary number formed at the input is between 2 and 9 (including).

(b) Simplify $Y = (A + \bar{A}B)(C + \bar{D})$

(c) Design 1 bit comparator using logic gates.

Q. 4 (a) Given the logic expression

$$A + \bar{B}C + A\bar{B}\bar{D} + ABCD$$

(i) Express in standard SOP

(ii) Draw K-map for the equation.

(iii) Minimize and realize using NAND gates only.

(b) Design 8 bit BCD adder.

Q. 5 (a) Design a mod 5 synchronous up counter using JK FF.

(b) Convert SR FF to TFF and JK FF.

Q. 6 Write short note on (any three) :

(a) VHDL

(b) Multivibrators

(c) Gray code Excess-3 code

(d) Johnson Ring Counter

- (2) Q. 1 (a) C
 (2) (b) F
 (2) (c) C
 (2) (d) E
 (2) (e) I
 (2) (f) C
 (2) (g) E
 (2) (h) E
 (1) (i) C

Q. 2 (a) S

r
F(5) (b) V
t

Q. 3 (a) I

I

Q. 4 (a) I

I

(10) (b) I

Q. 5 (a) I

(10) I

(20) (b) I

(b)

Q. 6 (a)

(b)

Dec. 2015

- Q. 1 (a)** Convert decimal number 199.375 into binary, octal, hexadecimal system. (2 Marks)
- (b)** Perform hexadecimal arithmetic operation: DADA + BABA. (2 Marks)
- (c)** Convert binary data 1010 into 7 bit even parity hamming code. (2 Marks)
- (d)** Express the equation in standard POS form: $F(A, B, C) = \sum m(0, 2, 5, 7)$. (2 Marks)
- (e)** Differentiate in brief between combinational and sequential circuits. (2 Marks)
- (f)** Compare TTL and CMOS with respect to speed, power dissipation, fan-in and fan-out. (2 Marks)
- (g)** Explain in brief weighted and non-weighted codes with one example each. (2 Marks)
- (h)** Explain the race around condition in JK flip-flop. State various methods to overcome it. (2 Marks)
- (i)** Convert JK flip-flop into D-flip-flop and T-flip-flop (show only the design without steps). (2 Marks)
- (j)** What is Modulus of the counter ? For MOD-6 counter how many flip-flops are needed ? (2 Marks)
- Q. 2 (a)** Simplify the following equation using K-map to obtain minimum POS equation and realize the minimum equation using only NOR gates. (10 Marks)

$$F(A, B, C, D) = \sum m(1, 3, 4, 6, 9, 11, 12, 14)$$
- (b)** What is multiplexer tree ? Construct 32:1 multiplexer using 8:1 multiplexers only. Explain how the logic on particular data line is steered to the output in this design with example. (10 Marks)
- Q. 3 (a)** Reduce using Quine Mc Cluskey method and realize the equation using only NAND gates. (10 Marks)

$$F(P, Q, R, S) = \sum m(0, 1, 2, 8, 10, 11, 14, 15)$$
- (b)** Implement single digit BCD adder using 4-bit binary adder IC 7483. Show the design procedure and explain its operation. (10 Marks)
- Q. 4 (a)** Explain the concept of comparator. Develop the truth table for 2-bit binary comparator and design it using a suitable decoder and additional gates. (10 Marks)
- (b)** Design MOD-5 synchronous up-counter using JK flip-flops with all the design steps. (10 Marks)
- Q. 5 (a)** Input to a combinational circuit is a 4-bit binary number. Design the circuit with minimum hardware for the following: (10 Marks)
 - Output P = 1 if the number is prime.
 - Output Q = 1 if number is divisible by 3.
- (b)** Draw a circuit diagram for 3-bit asynchronous binary down counter using master-slave JK flip-flops. Show the output of each flip-flop with reference to the clock and justify that the down counting action. Also prove from the timing diagram that the counter is "divide by 8" counter. (10 Marks)
- Q. 6 (a)** What is shift register ? Explain 4-bit bidirectional shift register. (10 Marks)
- (b)** Draw and explain the working of 4-bit ring counter with timing diagram. (10 Marks)

□ □ □

May 2016

- Q. 1** (a) Convert $(532.125)_8$ into decimal, binary and hexadecimal. (3 Marks)
 (b) Convert $(47.3)_7$ into BCD, excess-3 and gray code. (3 Marks)
 (c) Subtract using 1's and 2's complement method $(56)_{10} - (76)_{10}$. (4 Marks)
 (d) Obtain odd parity Hamming code for 1011. (2 Marks)
 (e) Implement Ex-OR gate using NOR gate only. (2 Marks)
 (f) Perform the following operations without changing the base : (4 Marks)
 - 1. $(314)_8 + (737)_8$
 - 2. $(312.40)_5 + (214.33)_5$
 (g) State and prove De Morgan's theorem. (2 Marks)
- Q. 2** (a) Reduce equation using Quine Mc Cluskey method and realize circuit using basic gates. (10 Marks)
 $F(A, B, C, D) = \sum m(1, 3, 7, 9, 10, 11, 13, 15)$
 (b) Design 8 bit BCD adder. (10 Marks)
- Q. 3** (a) Design a logic circuit to convert gray to BCD code. (10 Marks)
 (b) Implement the following using only one 8 : 1 MUX and few gates. (5 Marks)
 $F(A, B, C, D) = \sum m(0, 3, 5, 7, 9, 13, 15)$
 (c) Design a full adder circuit using half adders and some gates. (5 Marks)
- Q. 4** (a) Compare TTL and CMOS logic. (5 Marks)
 (b) Implement full subtractor using demultiplexer. (5 Marks)
 (c) Explain 4 bit universal shift register. (10 Marks)
- Q. 5** (a) Design mod 5 asynchronous up counter. (10 Marks) Q. 1(e)
 (b) Convert SR flipflop to JK flipflop and D flipflop. (10 Marks)
- Q. 6** Write short notes on (any four) : (20 Marks)
 - (a) VHDL
 - (b) Decade counter
 - (c) State table
 - (d) 4-bit magnitude comparator
 - (e) Multivibrators

Ans.

(i)

Step



Step