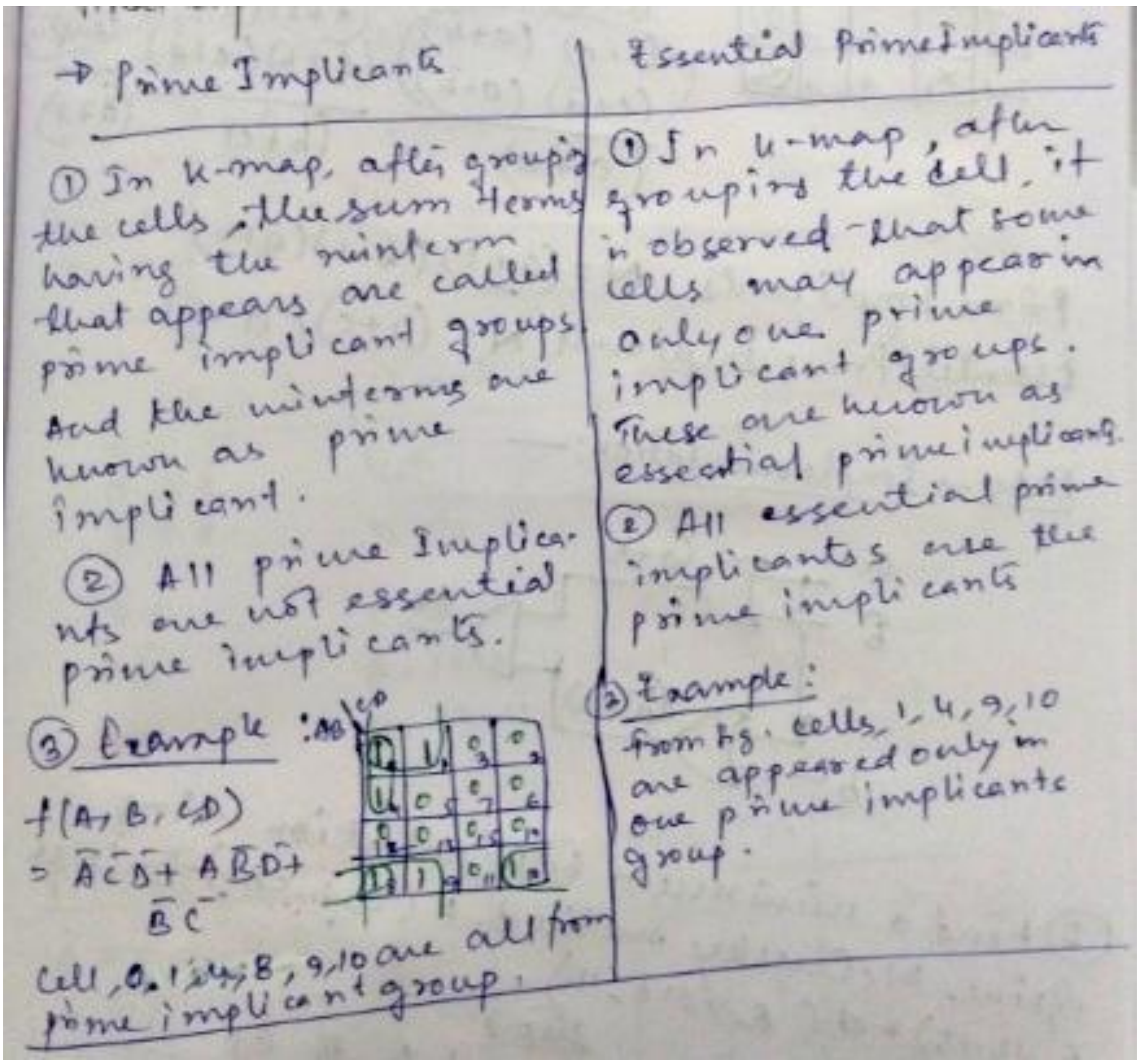
CMR INSTITUTE OF TECHNOLOGY	

Solutons of Internal Assesment Test - I			
Sub:	**DIGITAL SYSTEM DESIGN**	Code:	18EE35
Sem:	3rd	Branch:	EEE

1. Distinguish prime implicants and essential prime implicants. Determine prime implicants and essential prime implicants of Y=f(a,b,c,d)= πM (0,1,4,5,8,9,11)+d(2,10) and obtain the final expression.

Solution:



→ Prime Implicants

① In k-map, after grouping the cells, the sum terms having the minterm that appears are called prime implicant groups. And the minterms are known as prime implicant.

② All prime Implicants are not essential prime implicants.

③ Example: f(A, B, C, D) = $\bar{A}\bar{C}\bar{D} + A\bar{B}\bar{D} + \bar{B}\bar{C}$

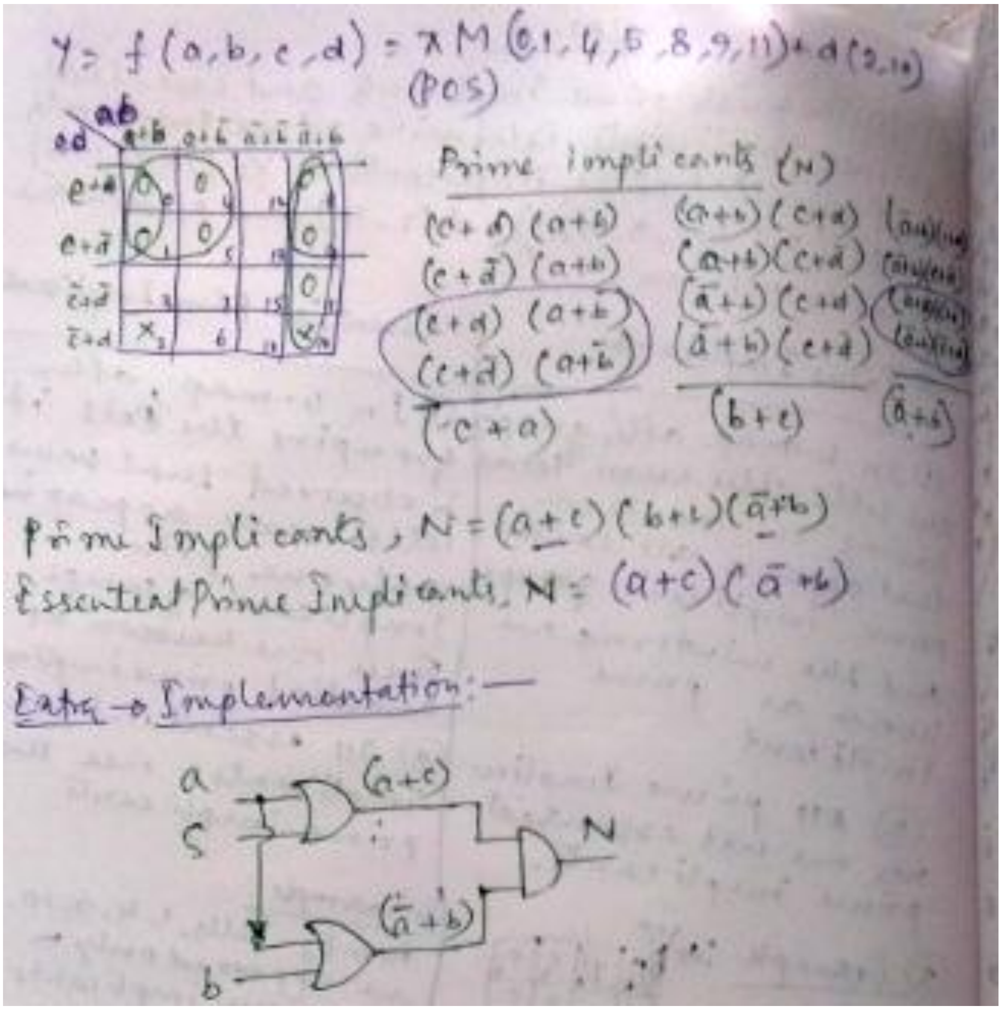
Cell 0, 1, 2, 4, 8, 9, 10 are all from prime implicant group.

Essential Prime Implicants

① In k-map, after grouping the cell, it is observed that some cells may appear in only one prime implicant groups. These are known as essential prime implicants.

② All essential prime implicants are the prime implicants.

③ Example: from fig. cells 1, 4, 9, 10 are appeared only in one prime implicants group.



2. Find a minimum SOP solution using Quine-McCluskey method $F(a,b,c,d)=\sum m(2,3,4,5,13,15)+d(8,9,10,11)$.

Solution:

Step 1 : List all minterms in binary form.

Minterms	Binary representation			
m_2	0	0	1	0
m_3	0	0	1	1
m_4	0	1	0	0
m_5	0	1	0	1
m_{13}	1	1	0	1
m_{15}	1	1	1	1
dm_8	1	0	0	0
dm_9	1	0	0	1
dm_{10}	1	0	1	0
dm_{11}	1	0	1	1

Step 2 : Arrange the minterms according to number of 1's.

Minterms	Binary representation			
m_2	0	0	1	0
m_4	0	1	0	0
m_8	1	0	0	0
m_3	0	0	1	1
m_5	0	1	0	1
m_9	1	0	0	1
m_{10}	1	0	1	0
m_{11}	1	0	1	1
m_{13}	1	1	0	1
m_{15}	1	1	1	1

Step 3 :

Minterm		Binary representation			
2, 3	✓	0	0	1	–
2, 10	✓	–	0	1	0
4, 5		0	1	0	–
8, 10	✓	1	0	–	0
8, 9	✓	1	0	0	–
3, 11	✓	–	0	1	1
5, 13		–	1	0	1
9, 13	✓	1	–	0	1
9, 11	✓	1	0	–	1
10, 11	✓	1	0	1	–
13, 15	✓	1	1	–	1
11, 15	✓	1	–	1	1

Step 4 :

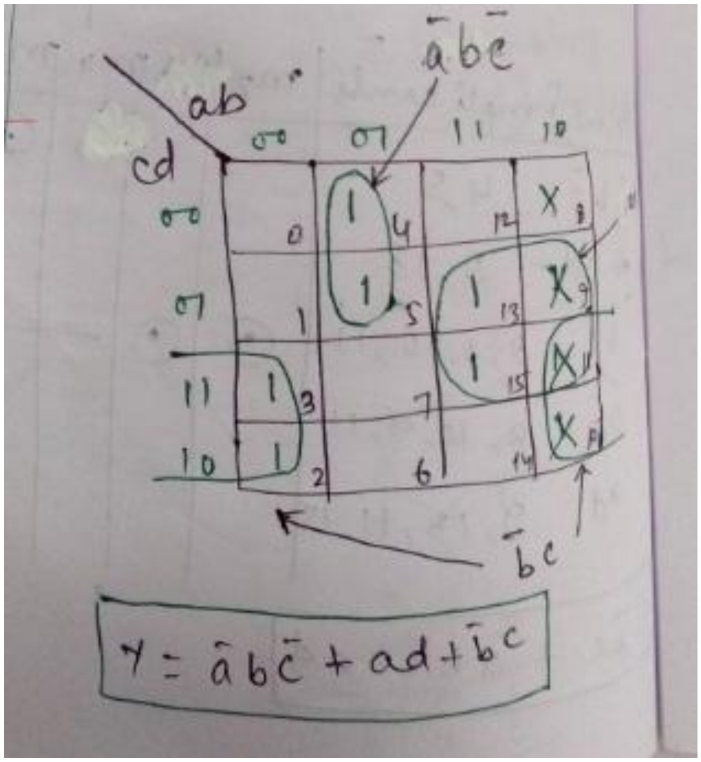
Minterm	Binary representation			
2, 3, 10, 11	–	0	1	–
8, 9, 10, 11	1	0	–	–
9, 11, 13, 15	1	–	–	1

Step 5 :

Prime implicants				Binary representation			
$\bar{a}$	b	$\bar{c}$	4, 5	0	1	0	–
b	$\bar{c}$	d	5, 13	–	1	0	1
$\bar{b}$	c		2, 3, 10, 11	–	0	1	–
a	$\bar{b}$		8, 9, 10, 11	1	0	–	–
a	d		9, 11, 13, 15	1	–	–	1

Prime Implicants		m_2	m_3	m_4	m_5	m_{13}	m_{15}	
$\bar{a} b \bar{c}$	4, 5			⊙	⊙			→ $\bar{a} b \bar{c}$
$b \bar{c} d$	5, 13				•	•		
$\bar{b} c$	2, 3, 10, 11	⊙	⊙					→ $\bar{b} c$
$a \bar{b}$	8, 10, 9, 11							
$a d$	9, 13, 11, 15					⊙	⊙	→ ad

$$Y = \bar{b}c + \bar{a}b\bar{c} + ad$$

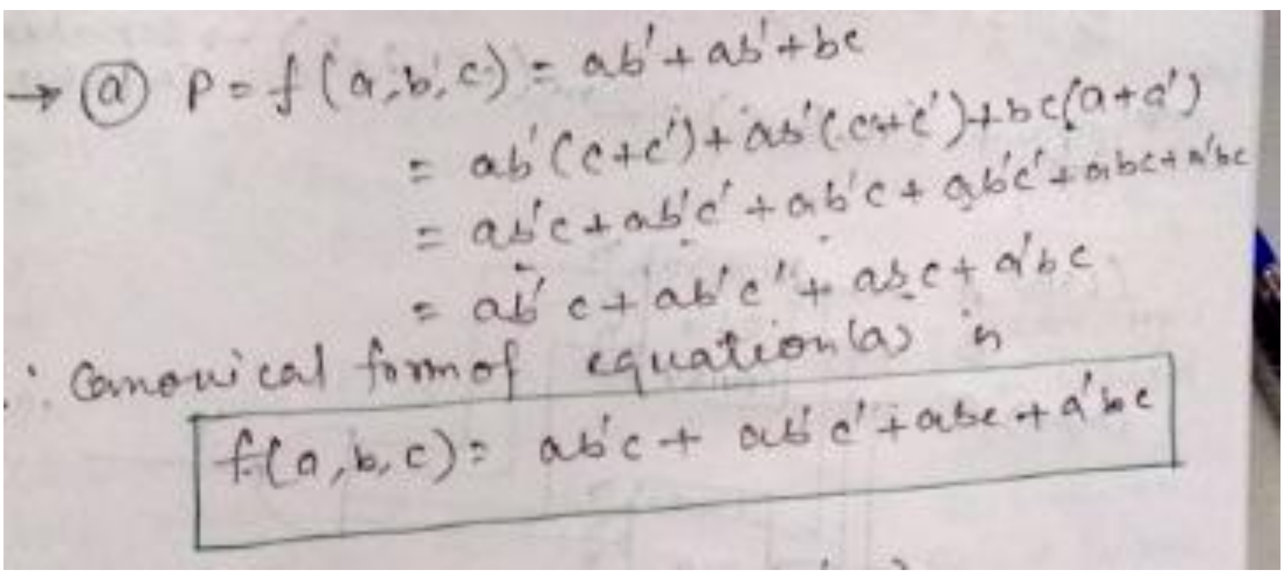


3. Place the following equations into proper canonical forms:

(a) P=f(a,b,c) = ab’+ab’+bc

(b) T= f(a,b,c)= (a+b’)(b’+c)

Solution:



→ (a) $P = f(a,b,c) = ab' + ab' + bc$

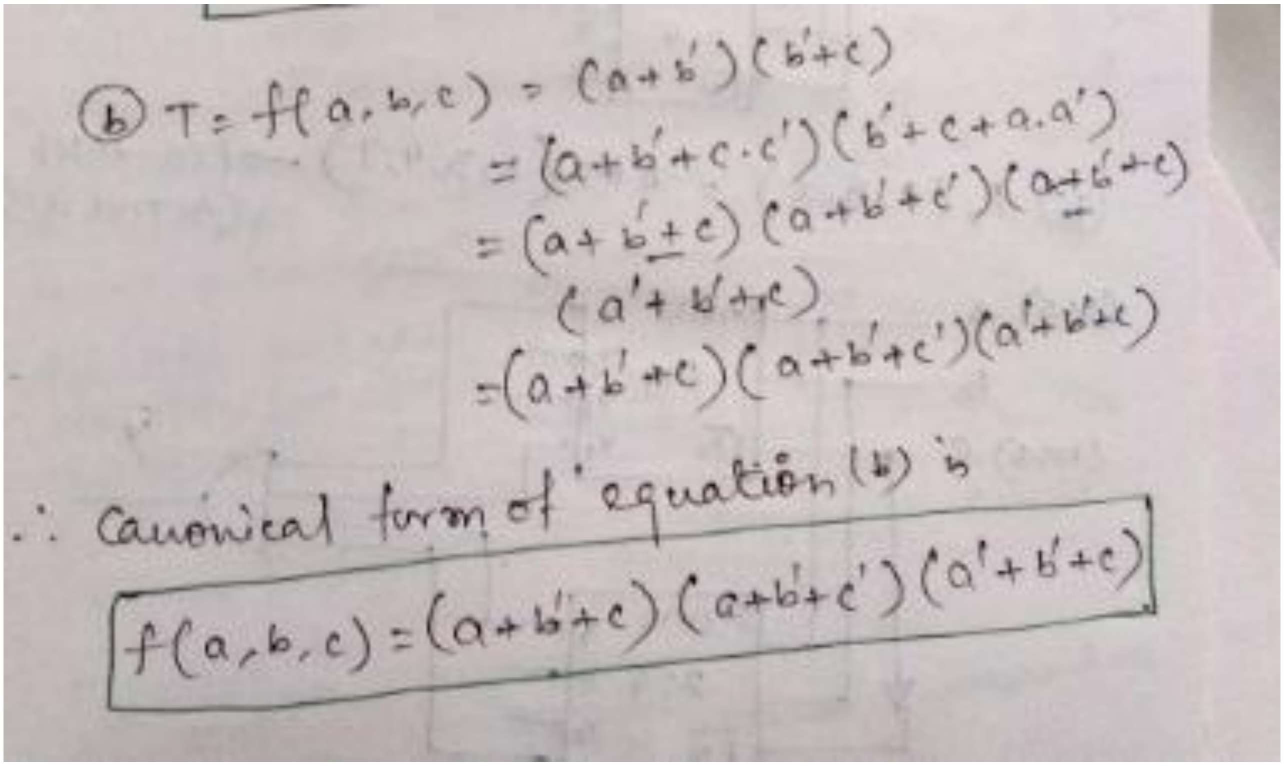
$= ab'(c+c') + ab'(c+c') + bc(a+a')$

$= ab'c + ab'c' + ab'c + ab'c' + abc + a'bc$

$= ab'c + ab'c' + abc + a'bc$

∴ Canonical form of equation (a) is

$f(a,b,c) = ab'c + ab'c' + abc + a'bc$



(b) $T = f(a,b,c) = (a+b')(b'+c)$

$= (a+b'+c\cdot c')(b'+c+a\cdot a')$

$= (a+b'+c)(a+b'+c')(a+b'+c)(a'+b'+c)$

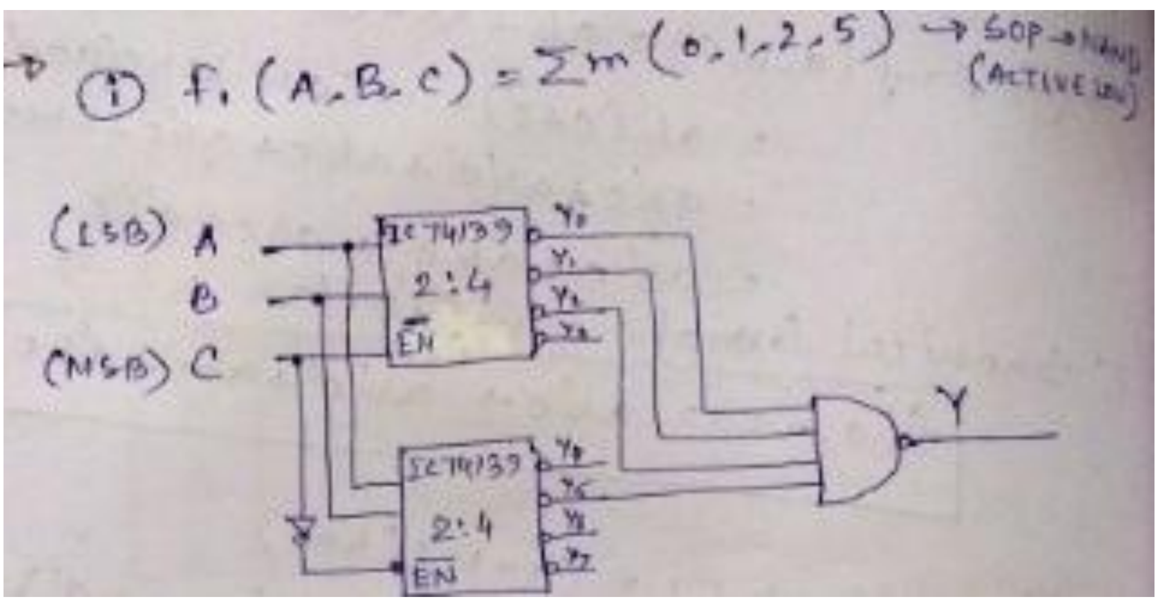
$= (a+b'+c)(a+b'+c')(a'+b'+c)$

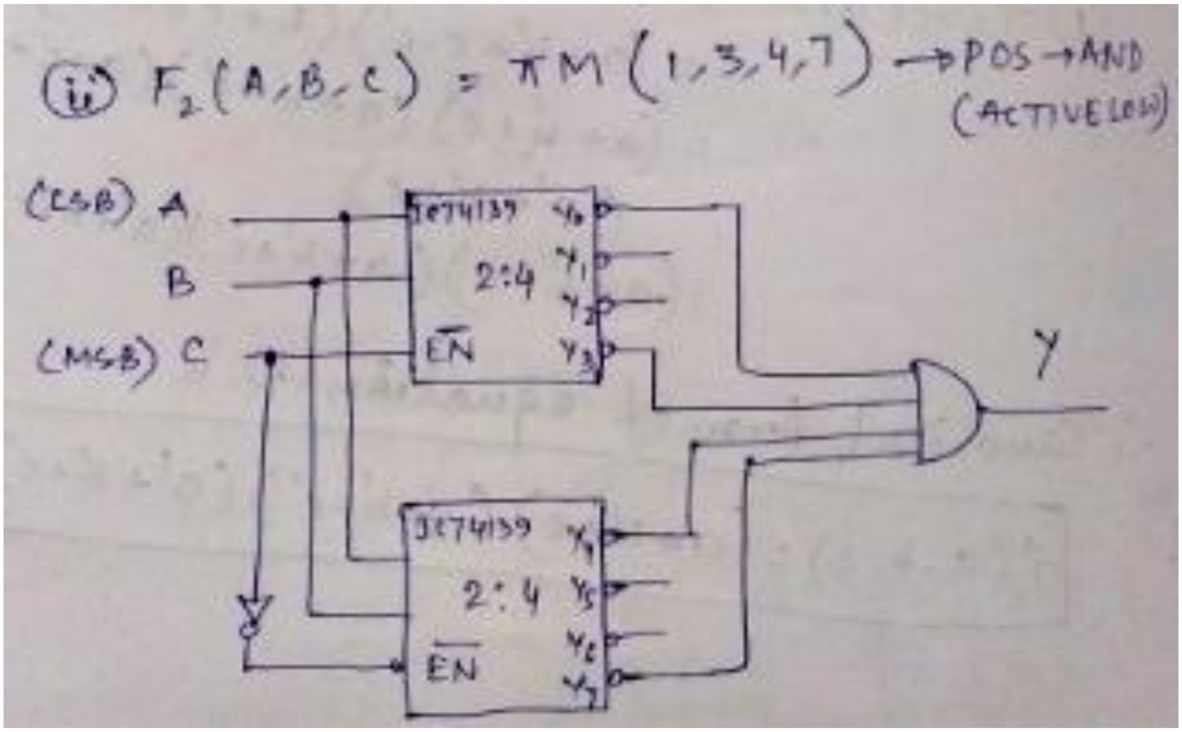
∴ Canonical form of equation (b) is

$f(a,b,c) = (a+b'+c)(a+b'+c')(a'+b'+c)$

4. Implement the function using active low output dual 2:4 decoder line decoder IC 74139
i)$F_1(A,B,C)=\sum m(0,1,2,5)$ ii)$F_2(A,B,C)=\pi M(1,3,4,7)$

Solution:





5. What is the problem associated with the parallel adder? Explain the method of correcting it, with suitable circuit equations.

Solution:

The parallel adder is ripple carry adder in which the carry output of each full-adder stage is connected to the carry input of the next higher-order stage. Therefore, the sum and carry outputs of any stage cannot be produced until the input carry occurs; this leads to a time delay in the addition process. This delay is known as **carry propagation delay**.

One method of speeding up this process by eliminating inter stage carry delay is called **look ahead-carry addition.** This method utilizes logic gates to look at the lower-order bits of the augend and addend to see if a higher-order carry is to be generated. It uses two functions : carry generate and carry propagate.

Consider the circuit of the full-adder shown in Fig. 2.11.1. Here, we define two functions : carry generate and carry propagate.

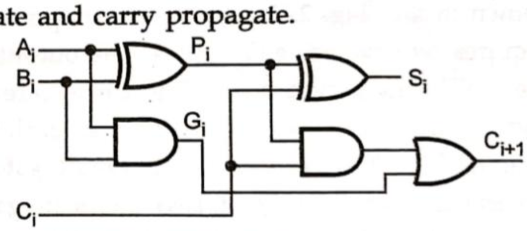


Fig. 2.11.1 Full-adder circuit

$$P_i = A_i \oplus B_i$$
$$G_i = A_i B_i$$

The output sum and carry can be expressed as

$$S_i = P_i \oplus C_i$$
$$C_{i+1} = G_i + P_i C_i$$

G_i is called a ca[...] oth A_i and B_i are one, regardless of the in[...] e it is term associated with the propagatio[...]

Now the Boole[...] ge can be written as follows.

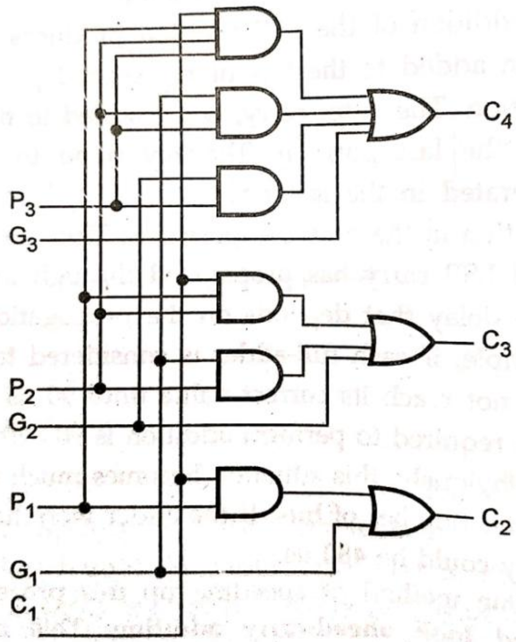


Fig. 2.11.2 Logic diagram of a look ahead carry generator

$$C_2 = ($$

$$C_3 = G_2 + P_2C_2 = G_2 + P_2\,(G_1 + P_1C_1) = G_2 + P_2G_1 + P_2\,P_1C_1$$

$$C_4 = G_3 + P_3\,C_3 = G_3 + P_3\,(G_2 + P_2\,G_1 + P_2\,P_1C_1)$$

$$= G_3 + P_3G_2 + P_3\,P_2\,G_1 + P_3\,P_2\,P_1\,C_1$$

From the above Boolean function it can be seen that C_4 does not have to wait for C_3 and C_2 to propagate; in fact C_4 is propagated at the same time as C_2 and C_3.

6. Realize the following Boolean function $Y=f(w,x,y,z)=\sum(0,1,5,6,7,10,15)$ using :

i) 16 to 1 MUX ii)8:1 MUX iii)4:1 MUX.

Solution: