UNIT-3

Minimisation of Switching functions

Introduction :

For simplification of boolean expresions by boolean algebra we need better understanding of boolean laws, rules and theorems. During the process of simplification we have to predict each successive step. For these reasons, we can never be absolutely certain that an expression simplified by Boolean algebra alone is the simplest possible expression.

The map method gives us a systematic approach for simplifying a Boolean expression. The map method, first proposed by Veitch and modified by Karnaugh, hence it is known as Veitch diagram or the Karnaugh map.

The Karnaugh Map :

The basis of this method is a graphical chart known as Karnaugh map (K-map). It contains boxes called cells. Each of the cell represents one of the 2^n possible products that can be formed from n- variables. Thus, a 2- variable map contains $2^2=4$ cells, a 3- variable map contains $2^3=8$ cells and so on.

2- Variable K-map :

2- Variable K-map contains 4 cells. Product terms are assigned to the cells of K-map by labelling each row and column of the map with a variable and its complement. The product term corresponds to a given cell is then the product of all variable in the row and column where the cell is located.

A \ B	$\bar{B}$	B
$\bar{A}$	$\bar{A}\bar{B}$ 0	$\bar{A}B$ 1
A	$A\bar{B}$ 2	AB 3

A \ B	0	1
0	m_0 0	m_1 1
1	m_2 2	m_3 3

3- Variable K- map :

In 3- variable K-map, it contains $2^3 = 8$ cells.

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$\bar{A}BC$ 3	$\bar{A}B\bar{C}$ 2
A	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5	ABC 7	$AB\bar{C}$ 6

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

4- Variable K- map :

In 4- variable K-map, it contains $2^4 = 16$ cells.

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}\bar{B}CD$ 3	$\bar{A}\bar{B}C\bar{D}$ 2
$\bar{A}B$	$\bar{A}B\bar{C}\bar{D}$ 4	$\bar{A}B\bar{C}D$ 5	$\bar{A}BCD$ 7	$\bar{A}BC\bar{D}$ 6
AB	$AB\bar{C}\bar{D}$ 12	$AB\bar{C}D$ 13	$ABCD$ 15	$ABC\bar{D}$ 14
$A\bar{B}$	$A\bar{B}\bar{C}\bar{D}$ 8	$A\bar{B}\bar{C}D$ 9	$A\bar{B}CD$ 11	$A\bar{B}C\bar{D}$ 10

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

Implementing a Boolean Expression & Truth table :

Once a boolean expression is in 'Sum of Product' form then K-map can be constructed by placing a 1 in cell corresponding to a minterm. Similarly from a truth table also a K-map can be constructed by placing a 1 in a cell corresponding combination for output.

Ex:

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

A \ B	$\bar{B}$	B
$\bar{A}$	0 ₀	1 ₁
A	1 ₂	0 ₃

Ex: $Y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + ABC$

The above expression can be represents as $Y = \Sigma m\,(0, 3, 4, 7)$.

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1 (0)	(1)	1 (3)	(2)
A	1 (4)	(5)	1 (7)	(6)

Grouping of the cells of 1's for Simplification :

We can group 1's that are in adjacent cells in one (1), pairs (2), quads (4), octets (8). Each group of 1's should be maximised to include the largest no. of adjacent cells as possible. Every 1 on the map must be included in atleast one group. There can be overlapping groups if they include non common 1's. The grouping should be done in square or rectangle form by convey the minterms of horizontal & vertical. The diagonal grouping is not allowed. The grouping can also be made by rolling the map.

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1		1	1
A	1	1		

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$			1	
A		1		

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$				
A	1	1	1	1

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1			1
$\bar{A}B$		1	1	
AB		1	1	
$A\bar{B}$	1		1	1

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$		1	1	
A		1	1	

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1			1
A	1			1

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$				
$\bar{A}B$				
AB	1	1	1	1
$A\bar{B}$	1	1	1	1

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1			1
$\bar{A}B$	1			1
AB	1			1
$A\bar{B}$	1			1

Simplifying the Expression on the Map :

Construct the K-map for the given expression & group the 1's in the cells. Simplify the expression by following procedure.

i) Each group of 1's create a product term composed of all variables that appear in only one form either uncomplemented or complemented with in the group. Variables that appear both uncomplemented and complemented are eliminated.

Scanned by CamScanner

ii) The final simplified expression is formed by summ the product terms of all the groups.

→ Simplify the following Boolean expressions using K-map metho

1) $F = x'yz + x'yz' + xy'z' + xy'z$

	$y'z'$	$y'z$	yz	yz'
x'			1	1
x	1	1		

$F = xy' + x'y$

2) $F = x'yz + xy'z' + xyz + xyz'$

	$y'z'$	$y'z$	yz	yz'
x'			1	
x	1		1	1

$F = xz' + yz$

3) $F = A'C + A'B + AB'C + BC$

	$B'C'$	$B'C$	BC	BC'
A'		1	1	1
A		1	1	

$F = C + A'B$

4) $F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$

	$y'z'$	$y'z$	yz	yz'
x'	1 $_0$	$_1$	$_3$	1 $_2$
x	1 $_4$	1 $_5$	$_7$	1 $_6$

$F = z' + xy'$

5) $F(w, x, y, z) = \Sigma(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

	$y'z'$	$y'z$	yz	yz'
$w'x'$	1	1		1
$w'x$	1	1		1
wx	1	1		1
wx'	1	1		

$F = y' + w'z' + xz'$

6. $F = A'B'C' + B'CD' + A'BCD' + AB'C'$

$F = A'B'C'(D+D') + (A+A')B'CD' + A'BCD' + AB'C'(D+D')$

$= A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'$

$= 0001 + 0000 + 1010 + 0010 + 0110 + 1001 + 1000$

$= m_0 + m_1 + m_2 + m_6 + m_9 + m_{10} + m_{11}$

	C'D'	C'D	CD	CD'
A'B'	1 (0)	1 (1)	(3)	1 (2)
A'B	(4)	(5)	(7)	1 (6)
AB	(12)	(13)	(15)	(14)
AB'	1 (8)	1 (9)	(11)	1 (10)

$F = B'C' + B'D' + A'CD'$

Simplification of Sum of Products Expressions :

Procedure to simplify boolean expressions as follows :

1) Plot the K-map and place 1's in those cells corresponding to the 1's in the truth table or sum of product expression. Place 0s in other cells.

2) Check the K-map for adjacent 1s and encircle those 1s which are not adjacent to any other 1's. These are called isolated 1's.

3. Check for those 1's which are adjacent to only one other 1 and encircle such pairs.

4. Check for quads and octets of adjacent 1's even if it contains some 1's that have already been encircled. While doing this make sure that there are minimum no. of groups.

5. Combine any pairs necessary to include any 1's that have not yet been grouped.

6. Form the simplified expression by summing product terms of all the groups.

Simplification

→ Minimize following Boolean Expressions.

1) $F = \bar A \bar B \bar C + A \bar B \bar C + \bar A \bar B C + A \bar B C + \bar A B C$

	$\bar B \bar C$	$\bar B C$	BC	$B\bar C$
$\bar A$	1	1	1	
A	1	1		

$F = \bar A C + \bar B$

2) $F = \bar A \bar B C \bar D + \bar A B \bar C \bar D + A B \bar C \bar D + A B \bar C D + A \bar B \bar C D + \bar A B \bar C D$

	$\bar C \bar D$	$\bar C D$	CD	$C\bar D$
$\bar A \bar B$				1
$\bar A B$	1	1		
AB	1	1		
$A\bar B$		1		

$F = \bar A \bar B C \bar D + B \bar C + A \bar C D$

4) $F = \bar A \bar B \bar C D + \bar A B \bar C D + \bar A B C D + \bar A B C \bar D + A B \bar C \bar D + A B \bar C D + ABCD + A\bar B CD$

	$\bar C \bar D$	$\bar C D$	CD	$C\bar D$
$\bar A \bar B$		1		
$\bar A B$		1	1	1
AB	1	1	1	
$A\bar B$			1	

$F = \bar A \bar C D + \bar A B C + A B \bar C + A C D$

5) $f(A,B,C,D) = \Sigma m\,(0,1,4,8,9,10)$

	$\bar C \bar D$	$\bar C D$	CD	$C\bar D$
$\bar A \bar B$	1	1		
$\bar A B$	1			
AB				
$A\bar B$	1	1		1

$F = \bar B \bar C + A \bar B \bar D + \bar A \bar C \bar D$

Simplification of Product of Sums Expressions :

Procedure to simplify POS expressions.

1. Plot the K-map and place 0s in those cells corresponding to the 0s in the truth table & maxterms in the products of sum expression.
2. Check the K-map for adjacent 0s and encircle those 0s which are not adjacent any other 0s. These are called isolated 0s.
3. Check for those 0s which are adjacent to only one other 0 and encircle such pairs.
4. Check for quads and octets of adjacent 0s even if it contains some 0s that have already been encircled. While doing this make sure that there are minimum no. of groups.
5. Combine any pairs necessary to include any 0s that have not yet been grouped.
6. Form the simplified SOP expression for F by summing product terms of all the groups.
7. Use De Morgan's theorem on F to produce the simplified expression in POS form.

→ Minimize the following expressions into POS form.

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

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	0	0	
A	0		0	

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

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

$$= (\overline{\bar{B}\bar{C}})(\overline{BC})(\overline{\bar{A}C})$$

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

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

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

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0		0	0
$\bar{A}B$				
AB	0	0	0	
$A\bar{B}$	0	0		

$F = ABD + \bar{A}\bar{B}C + A\bar{C} + \bar{A}\bar{B}\bar{D}$

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

3) $f(A,B,C,D) = \Pi M(0, 2, 3, 8, 9, 12, 13, 15)$

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

$F = ABD + \bar{A}\bar{B}C + A\bar{C} + \bar{A}\bar{B}\bar{D}$

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

4) $F(A,B,C,D) = \Pi M(5, 6, 7, 12, 13)$

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

$F = AB\bar{C} + B\bar{C}D + \bar{A}BC$

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

Scanned by CamScanner

Five Variable K-map :

A five variable K-map requires $2^5 = 32$ cells, but adjacent cells are difficult to identify on a single 32 cell map. Therefore two 16-cells K-maps are generally used. If the variables are A, B, C, D and E then 5-Variable K-map can be constructed as follows.

AB \ CDE	000	001	011	010
00	0	1	3	2
01	8	9	11	10
11	24	25	27	26
10	16	17	19	18

	110	111	101	100
00	6	7	5	4
01	14	15	13	12
11	30	31	29	28
10	22	23	21	20

Maps of more than four variable are not as simple to use. The no. of cells becomes excessively large and the geometry for combining adjacent cells becomes more involved. The no. of cells is always equal to the no. of minterms.

In above K-map rows and columns are numbered in a reflected - code sequence. The minterm assigned to each square is read from these numbers. In the third row (11) and second column (001), in the five - variable map, is number 11001, the equivalent of decimal 25. Therefore, this square represents minterm m_{25}. The letter symbol of each variable is marked along those cells where the corresponding bit value of the reflected - code number is a 1.

Six Variable K-Map :

A 6-variable K-map requires $2^6 = 64$ Cells. Maps seven & more variables need too many cells. They are impractical to use. The 6-variable K-map is shown below.

ABC \ DEF	000	001	011	010	110	111	101	100
000	0	1	3	2	6	7	5	4
001	8	9	11	10	14	15	13	12
011	24	25	27	26	30	31	29	28
010	16	17	19	18	22	23	21	20
110	48	49	51	50	54	55	53	52
111	56	57	59	58	62	63	61	60
101	40	41	43	42	46	47	45	44
100	32	33	35	34	38	39	37	36

The 6-variable map consists of four four variable maps. Each of these four variable maps is recognised from the double lines in the center of the map; each retains the previously defined adjacency when taken individually.

Don't Care Conditions :

The 1's and 0's in the map signify the combination of variables that makes the function equal to 1 & 0 respectively. The combinations are usually obtained from a truth table that lists the conditions under which the function is a 1. The function is assumed to be 0 under all other conditions.

In some logic circuits, certain input conditions never occur, therefore the corresponding o/p never appears. In such cases the output level is not defined, it can be either HIGH or LOW. These output levels are indicated by 'X' or d or ϕ in the truth tables and are called don't care outputs or don't care conditions. These don't care conditions can be used on a map to provide further simplification of the function.

It should be realized that a don't-care combination cannot be marked with a 1 on the map because it would require that the function always be a 1 for such input combination.

Ex:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	X
1	1	1	X

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1	1	0
A	0	1	X	X

$Y = C$

In above example don't care o/p for cell ABC is taken as 1 to form a quad, & don't care o/p for cell $AB\bar{C}$ is taken as 0, since it is not helping any way to reduce an expression.

→ Find the reduced SOP form for $f(A,B,C,D) = \Sigma m(1,3,7,11,15) + \Sigma d(0,2,4)$

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

$F = \bar{A}\bar{B} + CD$

$\rightarrow f(A,B,C,D) = \Sigma m(5,6,7,12,13) + \Sigma d(4,9,14,15)$

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

$F = B$

$\rightarrow f(A,B,C) = \Sigma(0,1,3,7) + d(2,5)$

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1	1	1	X
A		X	1	

$F = \bar{A} + C$

$\rightarrow F(w,x,y,z) = \Sigma m(0,7,8,9,10,12) + \Sigma d(2,5,13)$

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	1 (0)	(1)	(3)	X (2)
$\bar{w}x$	(4)	X (5)	1 (7)	(6)
wx	1 (12)	X (13)	(15)	(14)
$w\bar{x}$	1 (8)	1 (9)	(11)	1 (10)

$F = \bar{x}\bar{z} + \bar{w}xz + w\bar{y}$

Tabular Method (or) Quine Mc Clusky method :

① Arrange all minterms in groups, such that all terms in the same group have the same number of 1's in their binary representation. Start with least no. of 1's and continue with groups of increasing no. of 1's. The no. of 1's in a term is called index of the term.

② Compare every term of the lowest - index group with each term in the successive group; whenever possible, combine the two terms being compared by means of combining theorem.

Two terms from adjacent groups are combinable if their binary representation differ by just a single digit in the same position, the combined term consists of original fixed representation, with a differend digit is replaced by '–', A Check mark (✓) is placed next to every term which has been combined with atleast one term.

③ The terms generated in step 2 are now compared in the same fashion : a new term is generated by combining two terms which differ by only a single 1 and whose dashes are in same position. The process continues until no further combinations are possible. The remaining unchecked terms constitute the set of prime implicants of the function.

→ Simplify the following boolean function by using Tabulation method.

$$F(A,B,C,D) = \Sigma m\,(0, 2, 3, 6, 7, 8, 10, 12, 13)$$

Step 1 : List all minterms in the binary form.

Step 2 : Arrange the minterms according to no. of 1's.

Step 3 : Compare each binary number with every term in the adjacent next higher category and if they differ only by one position, put a check mark and copy the term in the next column with '–' in the position they differed.

Step 4 : Apply the same process describe in step 3 for the resultant column and continue these cycles until a single pass through cycle yields no further elimination of literals.

Prime Implicants

Step 5: List the prime implicants

Step 6: Select the minimum no. of prime implicants which must cover all the minterms.

Minterm	Binary representation
m_0	0000
m_2	0010
m_3	0011
m_6	0110
m_7	0111
m_8	1000
m_{10}	1010
m_{12}	1100
m_{13}	1101

Minterm	Binary Representation
m_0	0000 ✓
m_2	0010 ✓
m_8	1000 ✓
m_3	0011 ✓
m_6	0110 ✓
m_{10}	1010 ✓
m_{12}	1100 ✓
m_7	0111 ✓
m_{13}	1101 ✓

Minterm	Binary Representation
0, 2	00_0 ✓
0, 8	_000 ✓
2, 3	001_ ✓
2, 6	0_10 ✓
2, 10	_010 ✓
8, 10	10_0 ✓
8, 12	1_00
3, 7	0_11 ✓
6, 7	011_ ✓
12, 13	110_ ✓

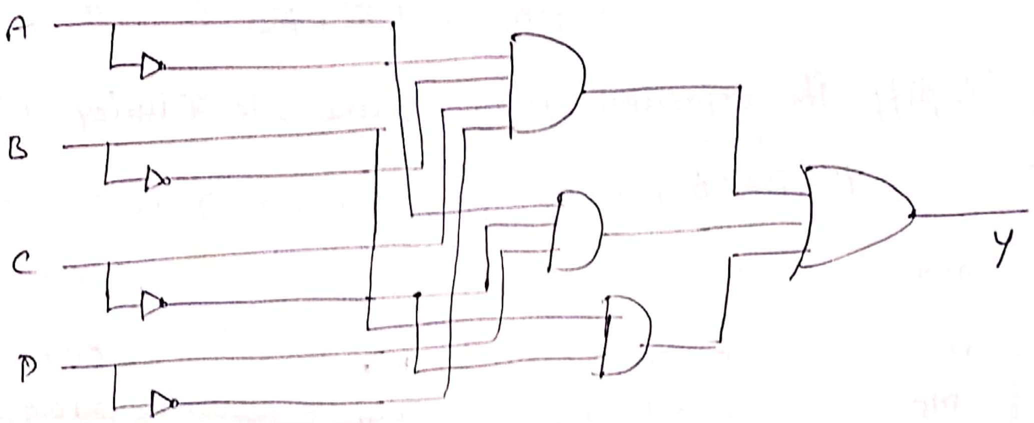
Minterms	Binary Representation
0, 2, 8, 10	_0_0
2, 3, 6, 7	0_1_

Prime Implicants	Binary Representation
8, 12	1_00
12, 13	110_
0, 2, 8, 10	_0_0
2, 3, 6, 7	0_1_

Prime Implicants	2	4	5	9	12	13
2	x					
9, 13				x		x
4, 5, 12, 13		x	x		x	x

$$Y = (0010) + (1-01) + (-10-)$$

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



Problem Solving using K-map :

Code Converters :

There is a wide variety of binary codes used in digital systems. Some of these codes are binary coded decimal (BCD), Excess - 3, Gray and so on. Many times it is required to convert one code to another.

1. Binary to BCD
2. BCD to Binary
3. BCD to Excess - 3
4. Excess - 3 to BCD
5. Binary to Gray
6. Gray to binary
7. ~~Gra~~ BCD to Gray

Binary

Prime Implicants	0	2	3	6	7	8	10	12	13
8, 12						x		x	
✓ 12, 13								x	⊗
✓ 0, 2, 8, 10	⊗	x				x	⊗		
✓ 2, 3, 6, 7		x	⊗	⊗	⊗				

$\therefore F(A,B,C,D) = (110-) + (-0-0) + (0-1-)$

$= AB\bar{C} + \bar{B}\bar{D} + \bar{A}C$

→ Simplify the expresion using Quine Mc Cluskey method.

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

Minterm	Binary representation
m_4	0100
m_5	0101
m_{12}	1100
m_{13}	1101
m_9	1001
m_2	0010

Minterm	Binary representation
m_2	0010
m_4	0100 ✓
m_5	0101 ✓
m_9	1001 ✓
m_{12}	1100 ✓
m_{13}	1101 ✓

Minterm	Binary Representation
4, 5	010–
4, 12	–100
5, 13	–101
9, 13	1–01
12, 13	110–

Minterms	Binary representation
4, 5, 12, 13	–10–

Prime Implicants	Binary Representation
2	0010
9, 13	1–01
4, 5, 12, 13	–10–

Binary to BCD Converter

Binary Code D	C	B	A	BCD Code B_4	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	1	0	0
1	1	1	1	1	0	1	0	1

For B_0

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$	0	1 (1)	1 (3)	2
$\bar{D}C$	4	1 (5)	1 (7)	6
DC	12	1 (13)	1 (15)	14
$D\bar{C}$	8	1 (9)	1 (11)	10

$B_0 = A$

For B_1

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$			1	1
$\bar{D}C$			1	1
DC	1	1		
$D\bar{C}$				

$B_1 = DC\bar{B} + \bar{D}B$

For B_2

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$				
$\bar{D}C$	1	1	1	1
DC			1	1
$D\bar{C}$				

$B_2 = \bar{D}C + CB$

For B_3

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$				
$\bar{D}C$				
DC				
$D\bar{C}$	1	1		

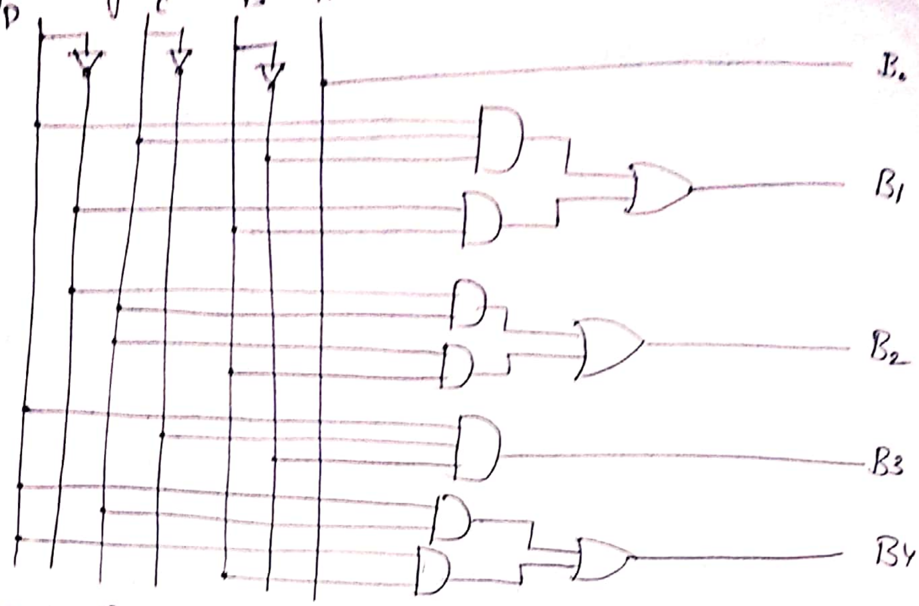
$B_3 = D\bar{C}\bar{B}$

For B_4

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$				
$\bar{D}C$				
DC	1	1	1	1
$D\bar{C}$			1	1

$B_4 = DC + DB$

Logic Diagram :



BCD to Binary Converter :

B_3	B_2	B_1	B_0	D	C	B	A
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1

For A

	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$		1	1	
$\bar{B_3}B_2$		1	1	
B_3B_2	X	X	X	X
$B_3\bar{B_2}$	X	1	X	X

$A = B_0$

	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$			1	1
$\bar{B_3}B_2$			1	1
B_3B_2	X	X	X	X
$B_3\bar{B_2}$			X	X

$B = B_1$

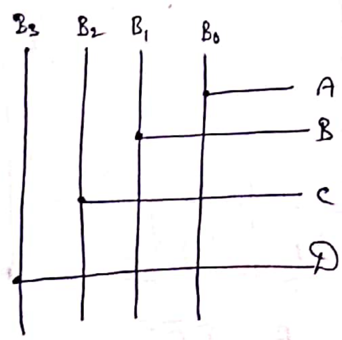
	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$				
$\bar{B_3}B_2$	1	1	1	1
B_3B_2	X	X	X	X
$B_3\bar{B_2}$				

$C = B_2$

For D

	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$	1			1
$\bar{B_3}B_2$	1			1
B_3B_2	X	X	X	X
$B_3\bar{B_2}$	X		X	X

$D = B_3$



BCD to Excess-3 Converter :

Decimal	B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
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

For E_0

	$\bar{B}_1\bar{B}_0$	$\bar{B}_1B_0$	B_1B_0	$B_1\bar{B}_0$
$\bar{B}_3\bar{B}_2$	1			1
$\bar{B}_3B_2$	1			1
B_3B_2	X	X	X	X
$B_3\bar{B}_2$	1		X	X

$E_0 = \bar{B}_0$

For E_1

	$\bar{B}_1\bar{B}_0$	$\bar{B}_1B_0$	B_1B_0	$B_1\bar{B}_0$
$\bar{B}_3\bar{B}_2$	1		1	
$\bar{B}_3B_2$	1		1	
B_3B_2	X	X	X	X
$B_3\bar{B}_2$	X		X	X

$E_1 = \bar{B}_1\bar{B}_0 + B_1B_0 = B_1 \odot B_0$

For E_2

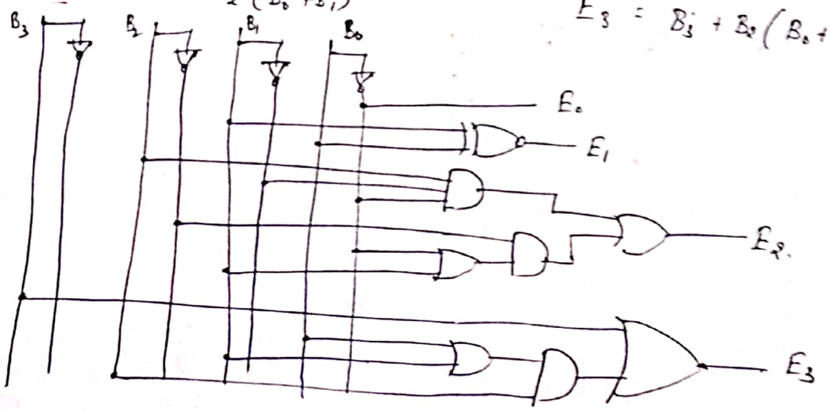
	$\bar{B}_1\bar{B}_0$	$\bar{B}_1B_0$	B_1B_0	$B_1\bar{B}_0$
$\bar{B}_3\bar{B}_2$		1	1	1
$\bar{B}_3B_2$	1			
B_3B_2	X			
$B_3\bar{B}_2$		1	X	X

$E_2 = B_2\bar{B}_1\bar{B}_0 + \bar{B}_2(B_0 + B_1)$

For E_3

	$\bar{B}_1\bar{B}_0$	$\bar{B}_1B_0$	B_1B_0	$B_1\bar{B}_0$
$\bar{B}_3\bar{B}_2$				
$\bar{B}_3B_2$		1	1	1
B_3B_2	X	X	X	X
$B_3\bar{B}_2$	1	1	X	X

$E_3 = B_3 + B_2(B_0 + B_1)$



Excess-3 to BCD code Converter :

E_3	E_2	E_1	E_0	B_3	B_2	B_1	B_0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

For B_0

	$\bar{E}_1\bar{E}_0$	$\bar{E}_1 E_0$	$E_1 E_0$	$E_1\bar{E}_0$
$\bar{E}_3\bar{E}_2$	X	X		X
$\bar{E}_3 E_2$	1			1
$E_3 E_2$	1	X	X	X
$E_3\bar{E}_2$	1			1

$B_0 = \bar{E}_0$

For B_1

	$\bar{E}_1\bar{E}_0$	$\bar{E}_1 E_0$	$E_1 E_0$	$E_1\bar{E}_0$
$\bar{E}_3\bar{E}_2$	X	X		X
$\bar{E}_3 E_2$		1		1
$E_3 E_2$	X	X	X	X
$E_3\bar{E}_2$		1		1

$B_1 = \bar{E}_1 E_0 + E_1\bar{E}_0 = E_1 \oplus E_0$

For B_2

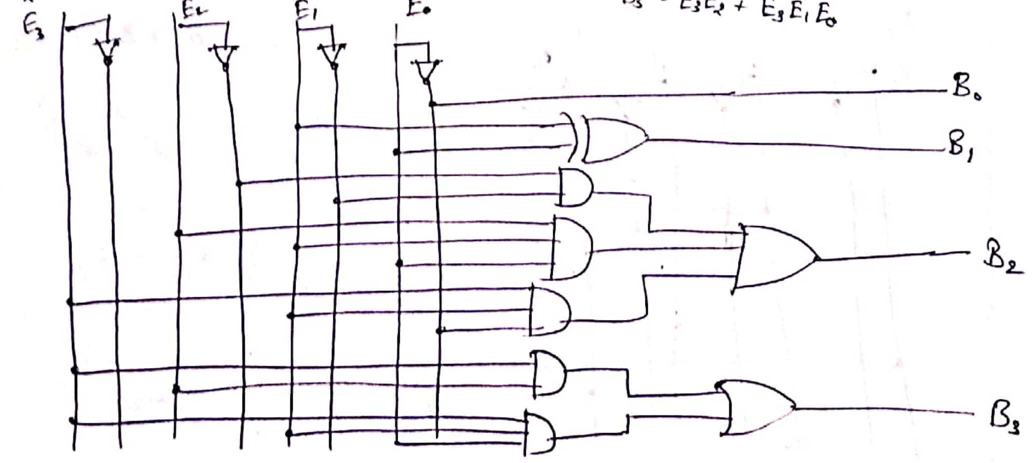
	$\bar{E}_1\bar{E}_0$	$\bar{E}_1 E_0$	$E_1 E_0$	$E_1\bar{E}_0$
$\bar{E}_3\bar{E}_2$	X	X		X
$\bar{E}_3 E_2$			1	
$E_3 E_2$		X	X	X
$E_3\bar{E}_2$	1	1		1

$B_2 = \bar{E}_2\bar{E}_1 + E_2 E_1 E_0 + E_3 E_1 \bar{E}_0$

For B_3

	$\bar{E}_1\bar{E}_0$	$\bar{E}_1 E_0$	$E_1 E_0$	$E_1\bar{E}_0$
$\bar{E}_3\bar{E}_2$	X	X		X
$\bar{E}_3 E_2$				
$E_3 E_2$	1	X	X	X
$E_3\bar{E}_2$			1	

$B_3 = E_3 E_2 + E_3 E_1 E_0$



Binary to Gray Code Converter :

The gray code is often used in digital systems because it has the advantage that only one bit in the numerical representation changes between successive numbers.

Decimal	D	C	B	A	G_3	G_2	G_1	G_0
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	1
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	1
11	1	0	1	1	1	1	1	0
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

For G_0

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$		1		1
$\bar{D}C$		1		1
DC		1		1
$D\bar{C}$		1		1

$G_0 = B \oplus A$

For G_1

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$			1	1
$\bar{D}C$	1	1		
DC	1	1		
$D\bar{C}$			1	1

$G_1 = C \oplus B$

For G_2

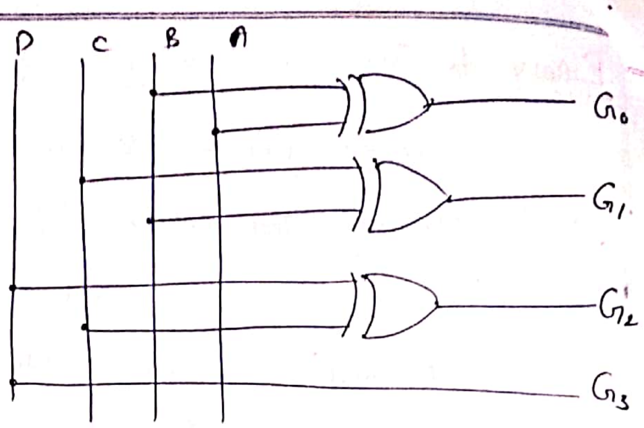
	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$				
$\bar{D}C$	1	1	1	1
DC				
$D\bar{C}$	1	1	1	1

$G_2 = D \oplus C$

For G_3

	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{D}\bar{C}$				
$\bar{D}C$				
DC	1	1	1	1
$D\bar{C}$	1	1	1	1

$G_3 = D$



Gray Code to Binary Converter :

Gray Code G_3	G_2	G_1	G_0	Binary Code D	C	B	A
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

For A

	$\bar G_1\bar G_0$	$\bar G_1 G_0$	$G_1 G_0$	$G_1\bar G_0$
$\bar G_3\bar G_2$		1		1
$\bar G_3 G_2$	1		1	
$G_3 G_2$		1		1
$G_3\bar G_2$	1		1	

For B

	$\bar G_1\bar G_0$	$\bar G_1 G_0$	$G_1 G_0$	$G_1\bar G_0$
$\bar G_3\bar G_2$			1	1
$\bar G_3 G_2$	1	1		
$G_3 G_2$			1	1
$G_3\bar G_2$	1	1		

$$A = (\bar G_3 G_2 + G_3\bar G_2)G_1G_0 + (\bar G_3\bar G_2 + G_3 G_2)\bar G_1 G_0$$
$$+ (\bar G_3 G_2 + G_3\bar G_2)\bar G_1\bar G_0 + (\bar G_3\bar G_2 + G_3G_2)G_1\bar G_0$$
$$= (G_3\oplus G_2)\oplus(G_1\oplus G_0)$$

$$B = (\bar G_3\bar G_2 + G_3G_2)G_1 + (\bar G_3 G_2 + G_3\bar G_2)\bar G_1$$
$$= (G_3\odot G_2)G_1 + (G_3\oplus G_2)\bar G_1$$
$$= (\overline{G_3\oplus G_2})G_1 + (G_3\oplus G_2)\bar G_1$$
$$= G_3\oplus G_2\oplus G_1$$

For C

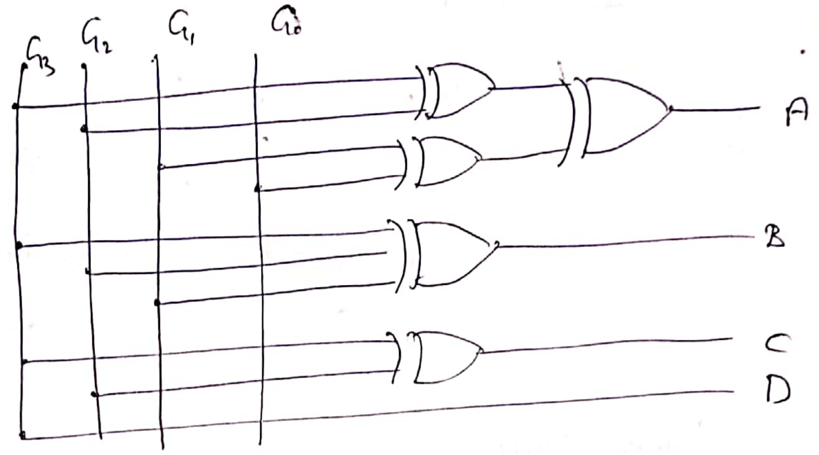
	$\bar{G_1}\bar{G_0}$	$\bar{G_1}G_0$	G_1G_0	$G_1\bar{G_0}$
$\bar{G_3}\bar{G_2}$				
$\bar{G_3}G_2$	1	1	1	1
G_3G_2				
$G_3\bar{G_2}$	1	1	1	1

$$C = \bar{G_3}G_2 + G_3\bar{G_2}$$
$$= G_3 \oplus G_2$$

For D

	$\bar{G_1}\bar{G_0}$	$\bar{G_1}G_0$	G_1G_0	$G_1\bar{G_0}$
$\bar{G_3}\bar{G_2}$				
$\bar{G_3}G_2$				
G_3G_2	1	1	1	1
$G_3\bar{G_2}$	1	1	1	1

$$D = G_3$$



BCD to Gray Code Converter :

BCD Code B_3	B_2	B_1	B_0	Gray Code G_3	G_2	G_1	G_0
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

For G_0

	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$		1		1
$\bar{B_3}B_2$		1		1
B_3B_2		X		X
$B_3\bar{B_2}$		1		X

$$G_0 = \bar{B_1}B_0 + B_1\bar{B_0} = B_1 \oplus B_0$$

For G_1

	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$			1	1
$\bar{B_3}B_2$	1	1		
B_3B_2	X	X		
$B_3\bar{B_2}$			X	X

$$G_1 = B_2\bar{B_1} + \bar{B_2}B_1 = B_2 \oplus B_1$$

Scanned by CamScanner

For G_2

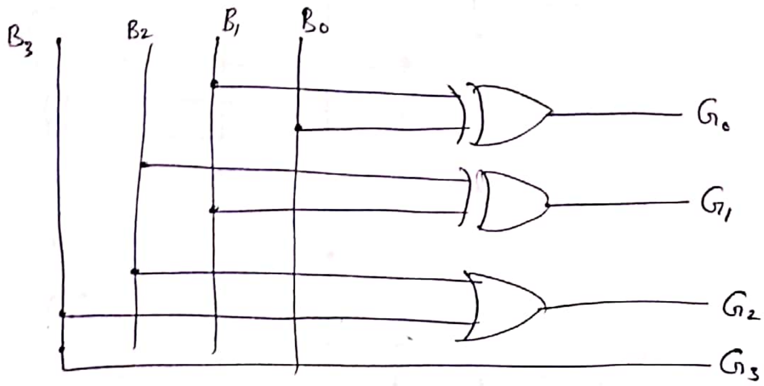
	$\bar{B}_1\bar{B}_0$	$\bar{B}_1 B_0$	$B_1 B_0$	$B_1\bar{B}_0$
$\bar{B}_3\bar{B}_2$		ϕ		ϕ
$\bar{B}_3 B_2$	1	1	1	1
$B_3 B_2$	X	X	X	X
$B_3\bar{B}_2$	1	1	X	X

$G_2 = B_2 + B_3$

For G_3

	$\bar{B}_1\bar{B}_0$	$\bar{B}_1 B_0$	$B_1 B_0$	$B_1\bar{B}_0$
$\bar{B}_3\bar{B}_2$				
$\bar{B}_3 B_2$				
$B_3 B_2$	X	X	X	X
$B_3\bar{B}_2$	1	1	X	X

$G_3 = B_3$



Binary Multiplier :

A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

	$\bar{B}$	B
$\bar{A}$	0	0
A	0	1

$Y = A \cdot B$

Multilevel NAND - NOR realizations :

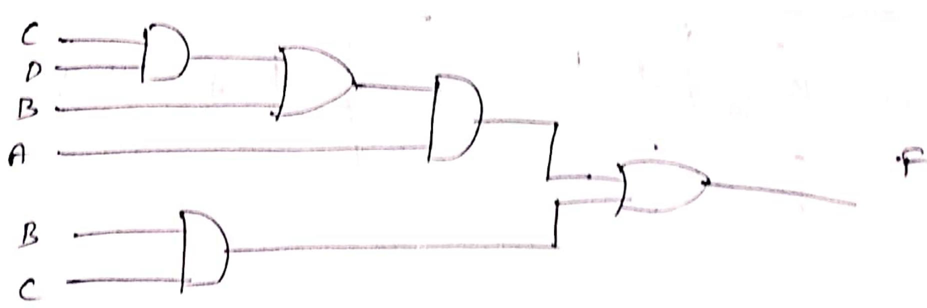
NAND

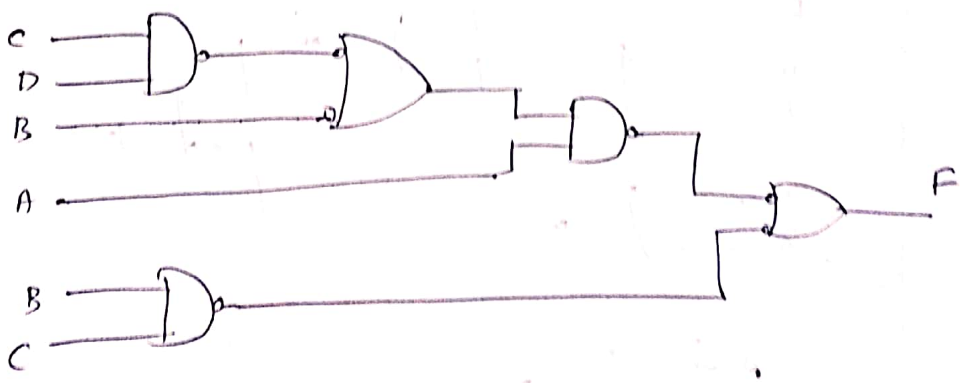
→ Convert all AND gates to NAND gates with AND - invert symbols.

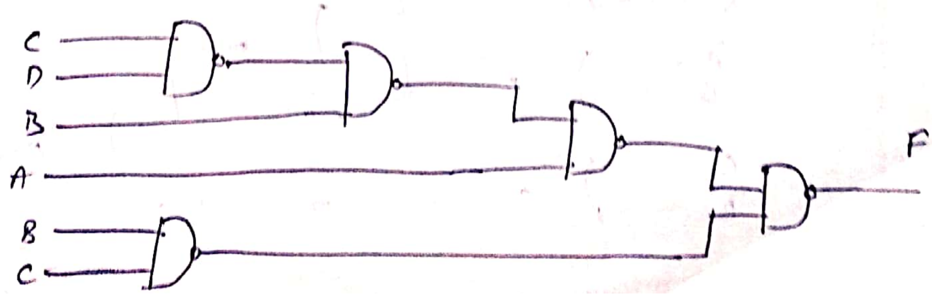
→ Convert all OR gates to NAND gates with invert - OR symbols.

→ Check all the bubbles in diagram. For every bubble that is not compensated by other small circle along the same line, insert an inverter or complement the input literal.

→ $F = A(CD + B) + BC$







NOR

- → Implement the logic function using AND, OR & NOT gate.
- → Convert all AND gates to NOR gates with invert - AND.
- → Convert all OR gates with OR invert.
- → Check all the bubbles in the diagram. For every bubble that is not compensated by another small circle along the same line, insert an inverter as complement the input literal.

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