1. Design NAND Gate logical circuit for implementing the SOPs equation.

$Y = AB + BC + A'C + AB'C$

soln

$Y = AB.1 + 1.BC + A'1.C + AB'C$

$Y = AB(C+C') + (A+A')BC + A'(B+B')C + AB'C$

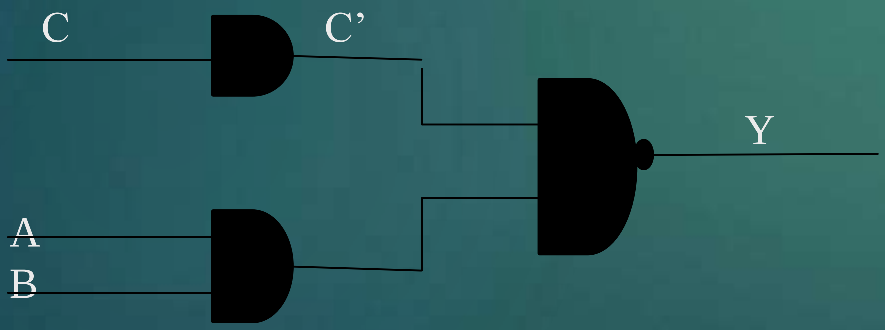
$Y = ABC + ABC' + ABC + A'BC + A'BC + A'B'C + AB'C$

$Y = ABC + ABC' + A'BC + A'B'C + AB'C$

1 1 1   1 1 0   0 1 1   0 0 1   1 0 1

$\Sigma m = (1,3,5,6,7)$

$Y = C'' + A''B''$



BC \ A	0	1
00		
01	1	1
11	1	1
10		1

2. Solve the logical function using k map

f(A,B,C,D) = A’B’C + AD + BD’ +CD’ +AC’ +A’B’

soln

f(A,B,C,D) =A’B’C.1 + AD.1.1 + BD’.1.1 + CD’.1.1 + AC’.1.1 +A’B’.1.1

=A’B’C(D+D’) + AD(B+B’)(C+C’) +BD’(A+A’)(C+C’) +CD’(A+A’)(B+B’) +AC’(B+B’)(D+D’) + A’B’(C+C’)(D+D’)

=A’B’CD + A’B’CD’ + ABCD +ABC’D +AB’CD + AB’C’D + ABCD’ + A’BCD’ +ABC’D’ + A’BC’D’ + ABCD’ + AB’CD’ +A’BCD’ +A’B’CD’ +ABC’D +ABC’D’ +AB’C’D +AB’C’D’ + A’B’CD +A’B’CD’ + A’B’C’D +A’B’C’D’

f(A,B,C,D) = Σm (0,1,2,3,4,6,8,9,10,11,12,13,14,15)

K-Map

CD \ AB	00	01	11	10
00	1	1	1	1
01	1	0	1	1
11	1	0	1	1
10	1	1	1	1

Y= A +D’ +B’

3. Solve the logical function using k-map F (A,B,C,D) = Σm (,1,2,4,7,8,11,13,14)

Solution :

F(A,B,C,D) = Σm (,1,2,4,7,8,11,13,14)

k-map

CD \ AB	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$$= A'BCD + AB'CD + ABC'D + ABCD' + A'B'C'D + A'B'CD' + A'BC'D' + AB'C'D'$$

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

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

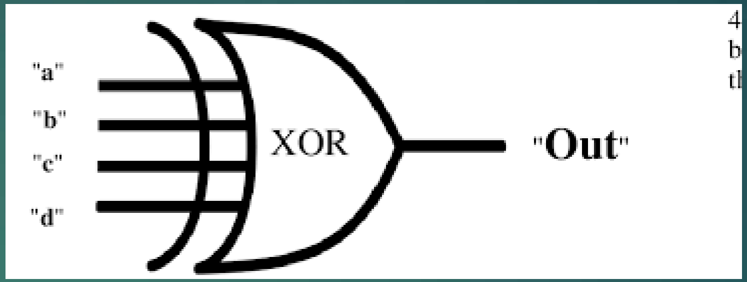
$$= (A \oplus B)(C \odot D) + (A \odot B)(C \oplus D)$$

Let $(A \oplus B) = X$, $(C \oplus D) = Y$

Now $F(A,B,C,D)$

$= XY' + X'Y$

$= (X \oplus Y)$



Now putting the values

$F(A,B,C,D) = (A \oplus B \oplus C \oplus D)$

4. Design Code converter

a) Binary code to gray code converter.

b) Gray code to Binary code converter.

c) BCD Code to Excess-3 code converter.

d) Excess-3 code to BCD code converter.

Solution:-

BINARY to GRAY CODE CONVERTER

*K-map for g_0

b3,b2 \ b1,b0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

The truth table for the conversion is-

Binary				Gray Code			
b_3	b_2	b_1	b_0	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
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

K-map for g1:-

b3,b2 \ b1,b0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

K-map for g2:-

b3,b2 \ b1,b0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

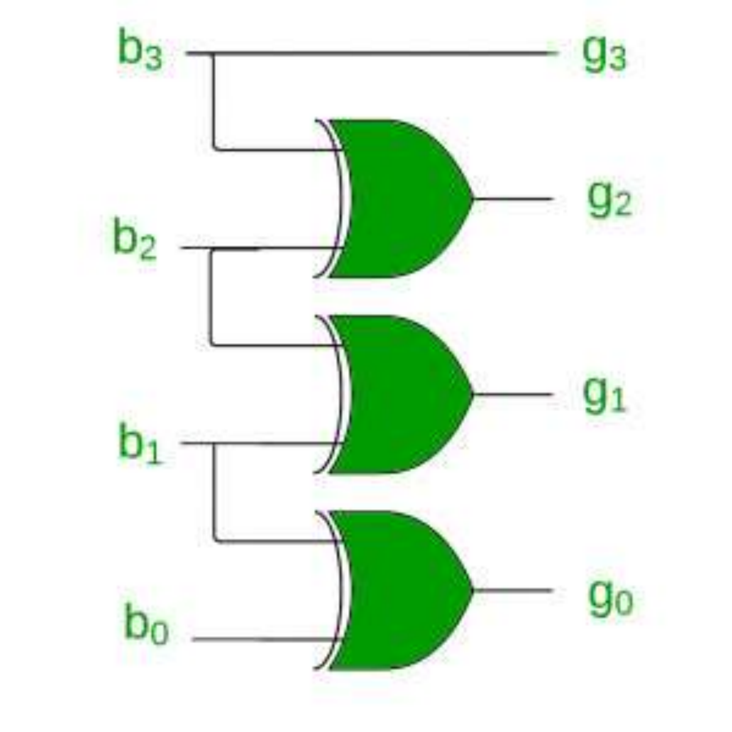
K-map for g3:-

b3,b2 \ b1,b0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Corresponding minimized Boolean expressions for gray code bits

$$g_0 = b_0 b_1' + b_1 b_0' = b_0 \oplus b_1$$
$$g_1 = b_2 b_1' + b_1 b_2' = b_1 \oplus b_2$$
$$g_2 = b_2 b_3' + b_3 b_2' = b_2 \oplus b_3$$
$$g_3 = b_3$$

The corresponding digital circuit –



Converting Gray Code to Binary : –

K-map for b0 :-

g3,g2 \ g1,g0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Truth table-

Gray Code				Binary			
g_3	g_2	g_1	g_0	b_3	b_2	b_1	b_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	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

K-map fro b1 :-

g3,g2 \ g1,g0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

K-map fro b2 :-

g3,g2 \ g1,g0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

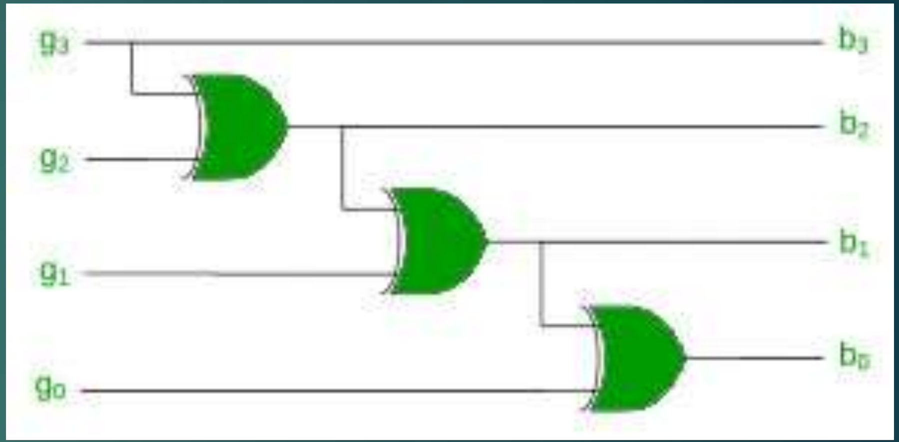
K-map fro b3 :-

g3,g2 \ g1,g0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Corresponding Boolean expressions –

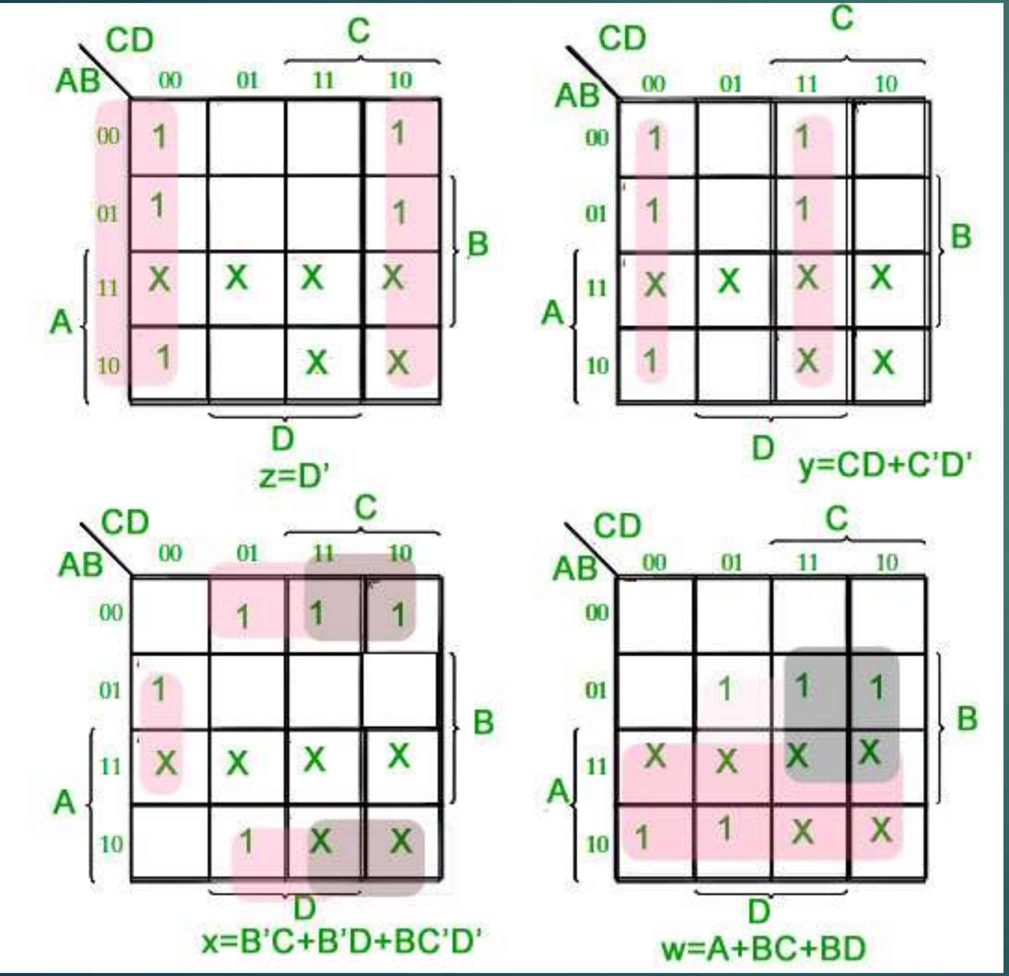
$$
\begin{aligned}
b_0 &= g_3'g_2'g_1'g_0 + g_3'g_2'g_1g_0' + g_3'g_2g_1'g_0' + g_3'g_2g_1g_0 + g_3g_2'g_1'g_0' + g_3g_2'g_1g_0 \\
&\quad + g_3g_2g_1'g_0 + g_3g_2g_1g_0' \\
&= g_3'g_2'(g_1'g_0 + g_1g_0') + g_3'g_2(g_1'g_0' + g_1g_0) + g_3g_2'(g_1'g_0' + g_1g_0) \\
&\quad + g_3g_2(g_1'g_0 + g_1g_0') \\
&= g_3'g_2'(g_0 \oplus g_1) + g_3'g_2(g_0 \odot g_1) + g_3g_2'(g_0 \odot g_1) + g_3g_2(g_0 \oplus g_1) \\
&= (g_0 \oplus g_1)(g_2 \odot g_3) + (g_0 \odot g_1)(g_2 \oplus g_3) \\
&= g_3 \oplus g_2 \oplus g_1 \oplus g_0 \\
b_1 &= g_3'g_2'g_1 + g_3'g_2g_1' + g_3g_2g_1 + g_3g_2'g_1' \\
&= g_3'(g_2'g_1 + g_2g_1') + g_3(g_2g_1 + g_2'g_1') \\
&= g_3'(g_2 \oplus g_1) + g_3(g_2 \odot g_1) \\
&= g_3 \oplus g_2 \oplus g_1 \\
b_2 &= g_3'g_2 + g_3g_2' \\
&= g_3 \oplus g_2 \\
b_3 &= g_3
\end{aligned}
$$

Corresponding digital circuit –



BCD to Excess 3 code converter:-

To find the corresponding digital circuit, we will use the K-Map technique for each of the Excess-3 code bits as output with all of the bits of the BCD number as input.

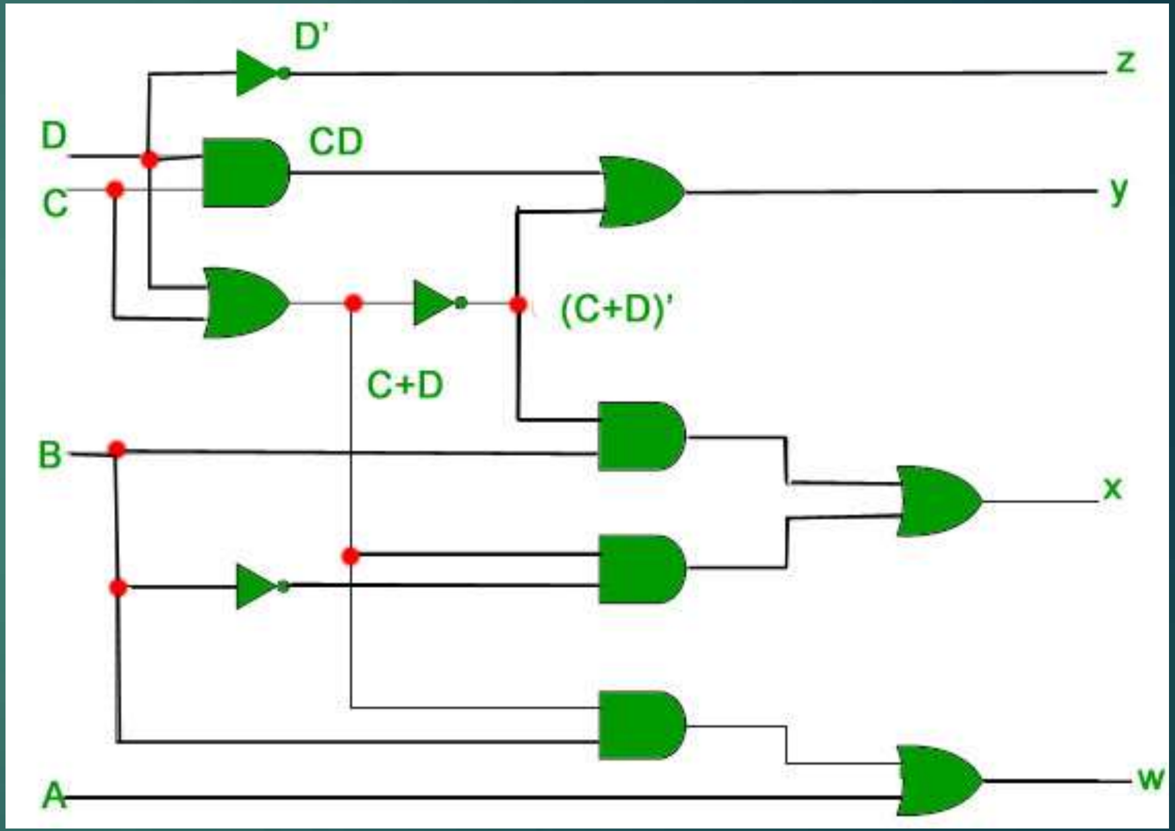


BCD(8421)				Excess-3			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Corresponding minimized Boolean expressions for Excess-3 code bits –

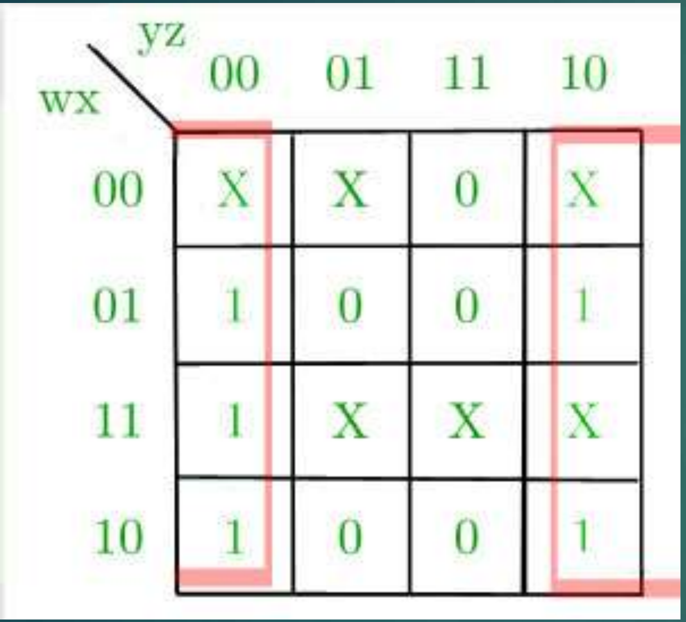
$$w = A + BC + BD$$
$$x = B'C + B'D + BC'D'$$
$$y = CD + C'D'$$
$$z = D'$$

The corresponding digital circuit-



Excess 3 to BCD code converter:-

K-map for D



Excess-3				BCD			
w	x	y	z	A	B	C	D
0	0	0	0	X	X	X	X
0	0	0	1	X	X	X	X
0	0	1	0	X	X	X	X
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
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

K-map for C

wx \ yz	00	01	11	10
00	X	X	0	X
01	0	1	0	1
11	0	X	X	X
10	0	1	0	1

K-map for B

wx \ yz	00	01	11	10
00	X	X	0	X
01	0	0	1	0
11	0	X	X	X
10	1	1	0	1

K-map for A

wx \ yz	00	01	11	10
00	X	X	0	X
01	0	0	0	0
11	1	X	X	X
10	0	0	1	0

Corresponding minimized boolean expressions for Excess-3 code bits –

$A = wx + wyz$

$B = x'y' + x'z' + xyz$

$C = y'z + yz'$

$D = z'$

The corresponding digital circuit –

Here $E_3,\ E_2,\ E_1,\ and\ E_0$ correspond to $w,\ x,\ y,\ and\ z$ and $B_3,\ B_2,\ B_1,\ and\ B_0$ correspond to $A,\ B,\ C,\ and\ D$.