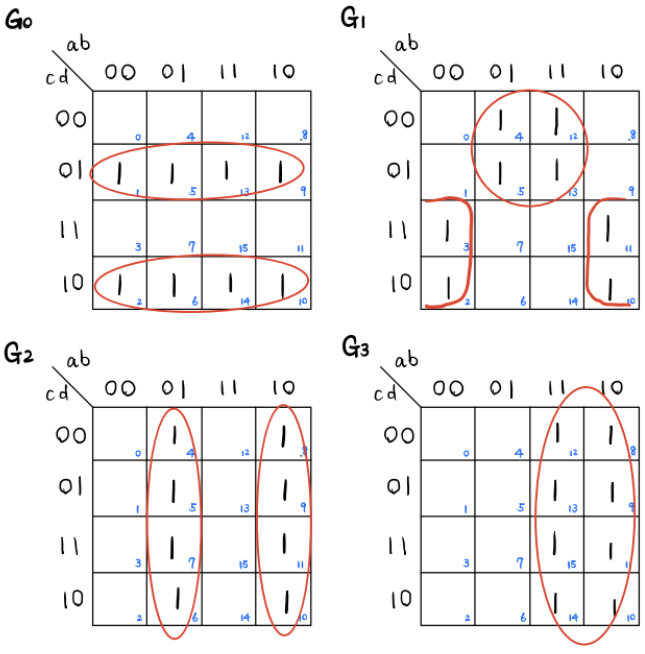
Logic Design HW3 Answer

1.

	4bit binary code				4bit Gray code			
	a	b	c	d	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

K map:



Simplify by Kmap

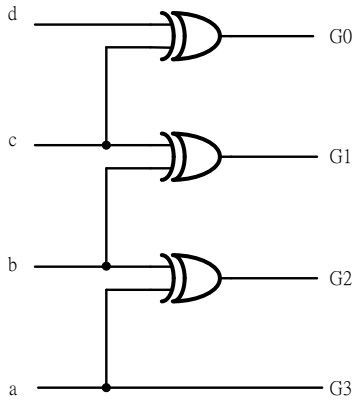
$G_0 = c'd + cd'$

$G_1 = b'c + bc'$

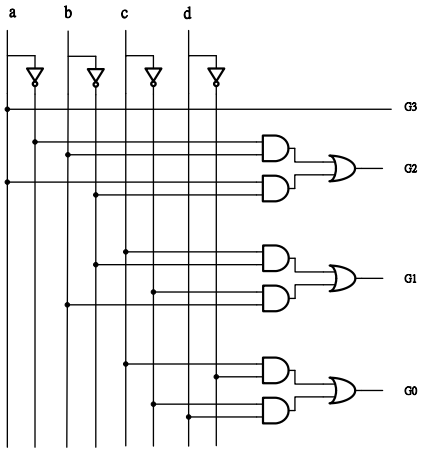
$G_2 = a'b + ab'$

$G_3 = a$

Logic circuit:



or



2.

(a) algebraic manipulation

x+y=1 , xy=0;

xz + x'y + yz = z(x+y) + x'y + xy = z (x+y) + y (x+x') = y+z

(b) truth table

Because xy=0, x+y=1

X	y	z	xz + x'y + yz	y + z
~~0~~	~~0~~	~~0~~	x	x
~~0~~	~~0~~	~~1~~	x	x
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
~~1~~	~~1~~	~~0~~	x	x
~~1~~	~~1~~	~~1~~	x	x

3.

(a)

	minterms
f	(a'bc), (ab'c), (abc'), (abc)
g	(a'b'c), (a'bc'), (ab'c'), (abc)
f'	(a'b'c'), (a'b'c), a'bc'), (ab'c')
g'	(a'b'c'),(a'bc), (ab'c), (abc')

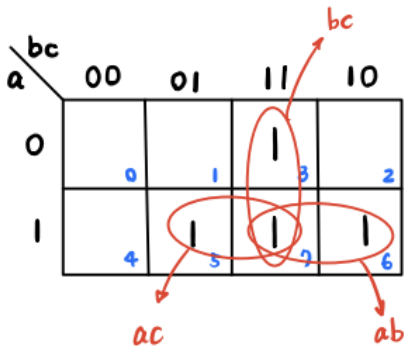
	maxtems
f	(a+b+c), (a+b+c'), (a+b'+c), (a'+b+c)
g	(a+b+c), (a+b'+c'), (a'+b+c'), (a'+b'+c)
f'	(a+b'+c'), (a'+b+c'), (a'+b'+c), (a'+b'+c')
g'	(a+b+c'), (a+b'+c), (a'+b+c), (a'+b'+c')

(b)

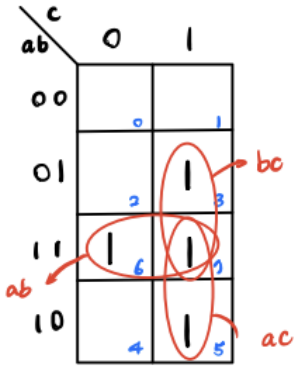
f in sum of minterms (som) algebraic form

f =a'bc + ab'c + abc'+ abc = $\sum(3,5,6,7)$

K map simplify **f = ac + bc + ab**

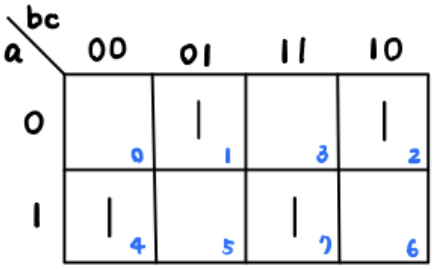


or

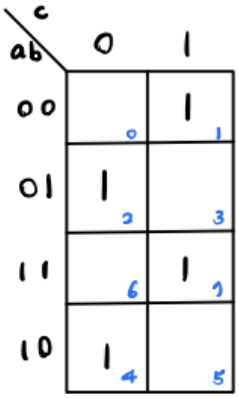


g in sum of minterms algebraic form

g = a'b'c + a'bc' + ab'c' + abc = $\sum(1,2,4,7)$



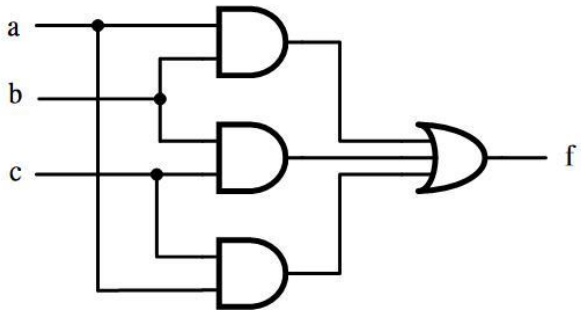
or

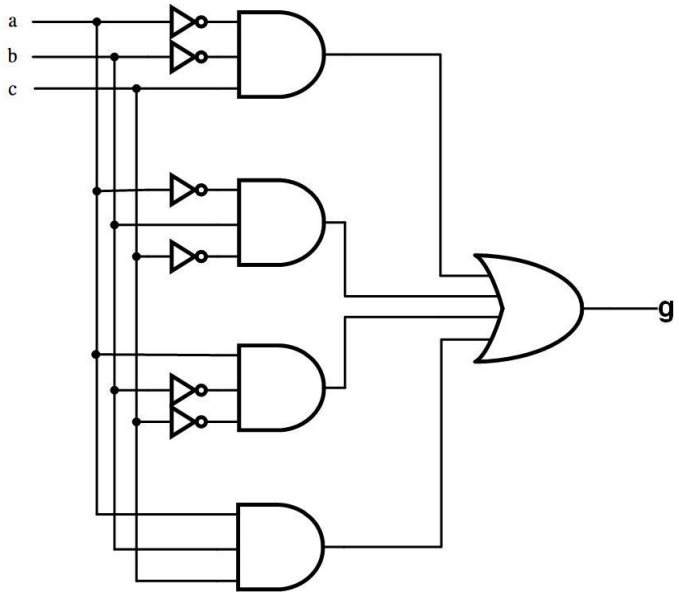


(C)

Sop form f = ac + bc + ab, g = a'b'c + a'bc' + ab'c' + abc

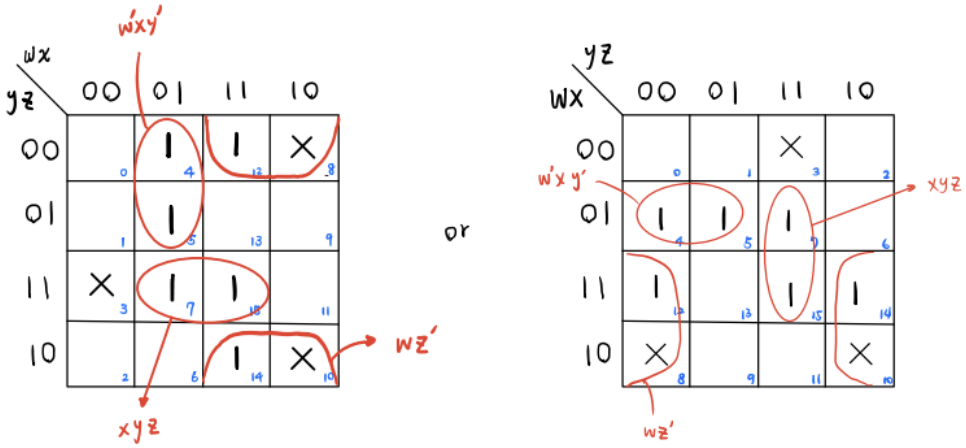
Logic diagrams





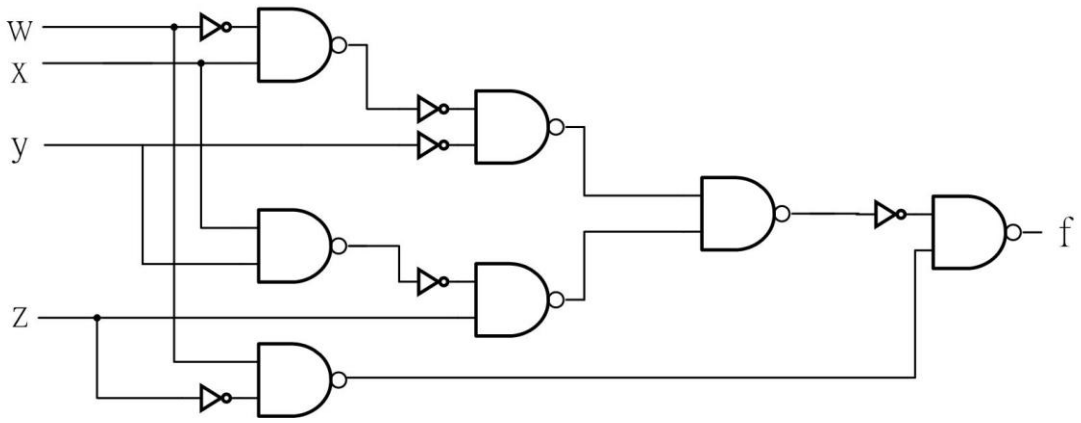
4. f(w, x, y, z) = ∑ (4, 5, 7, 12, 14, 15) + ∑ d(3, 8, 10)

K map:



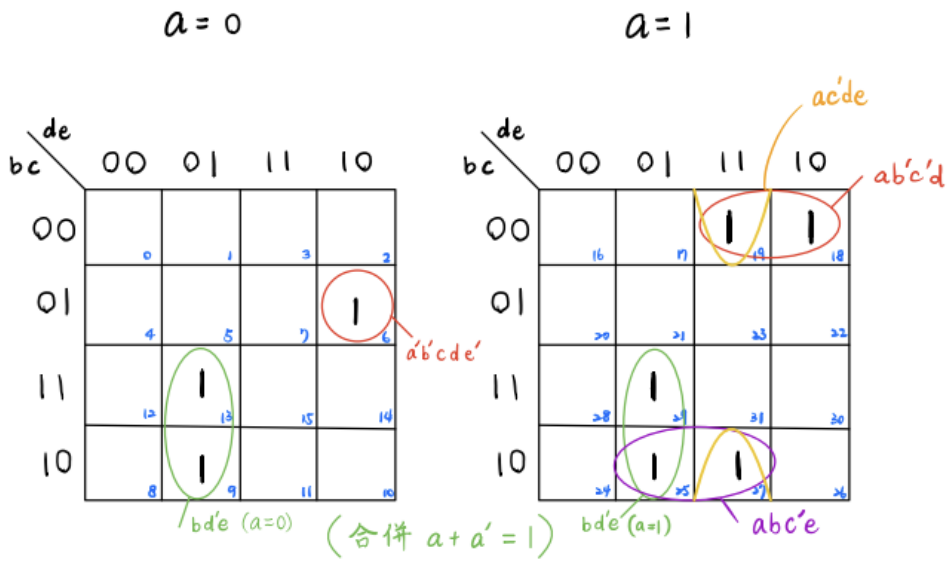
f(w,x,y,z)=wz' + w'xy'+ xyz

Logic diagram:

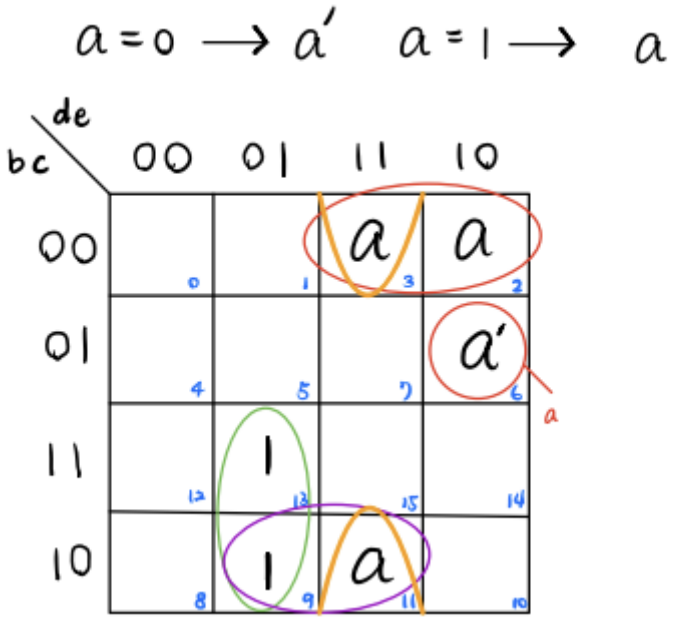


5.

Method1:



Method2:



PIs

There are 5 PIs: a'b'cde', bd'e, ab'c'd, abc'e, ac'de

EPIs

There are 3 EPIs: bd'e, a'b'cde', ab'c'd

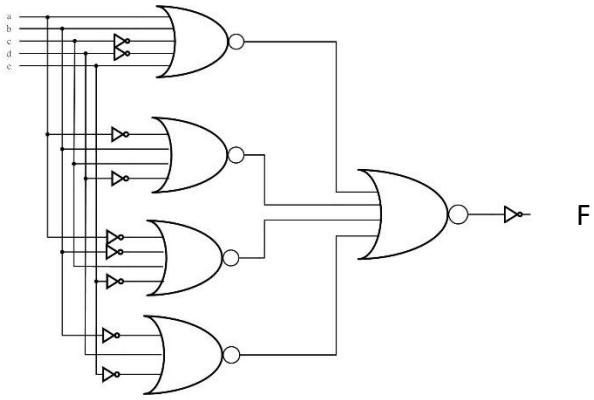
minimal covers:

1. bd'e + a'b'cde' + ab'c'd + abc'e or 2. bd'e + a'b'cde' + ab'c'd + ac'de

two-level NOR-NOR circuit:

1. F = bd'e + a'b'cde' + ab'c'd + abc'e

   =[(b'+d+e')(a+b+c'+d'+e)(a'+b+c+d')(a'+b'+c+e')]'



or

2. F = bd'e + a'b'cde' + ab'c'd + ac'de

   = (b'+d+e') + (a+b+c'+d'+e) + (a'+b+c+d') + (a'+c+d'+e')