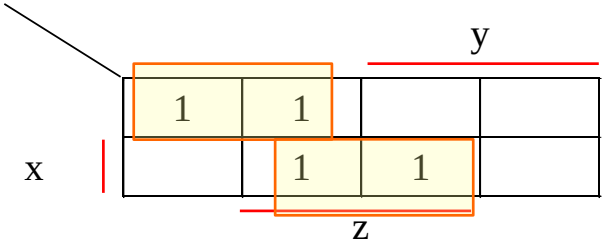
Tutorial 3 – Sample Solution
Logic Circuit Design

CO 2206 Computer Organization

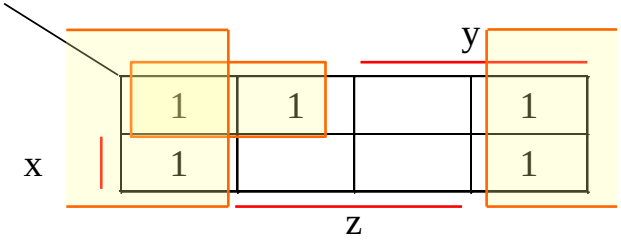
Task 1: Ans.1

- Simplify the following Boolean functions using Karnaugh map:
  - F(x,y,z) = ∑m(0,1,5,7)
  - F(x,y,z) = ∑m(0,1,2,4,6)
  - F(w,x,y,z) = Σm(1,4,6,7,8,9,10,11,15)
  - F(A,B,C,D) = ∑m(0,2,5,6,8,10,13,14,15)

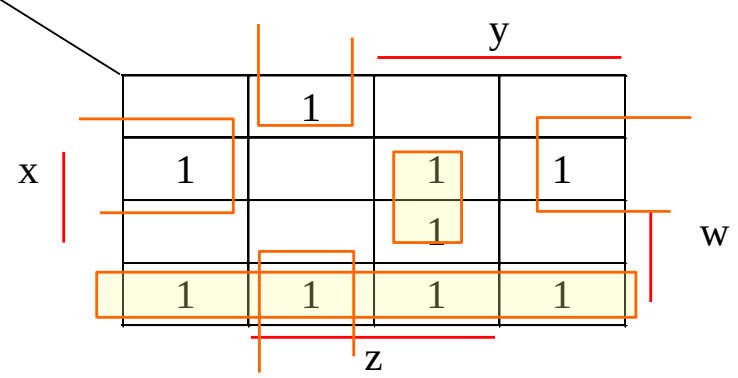
F = x'y + xz



F = z' + x'y'

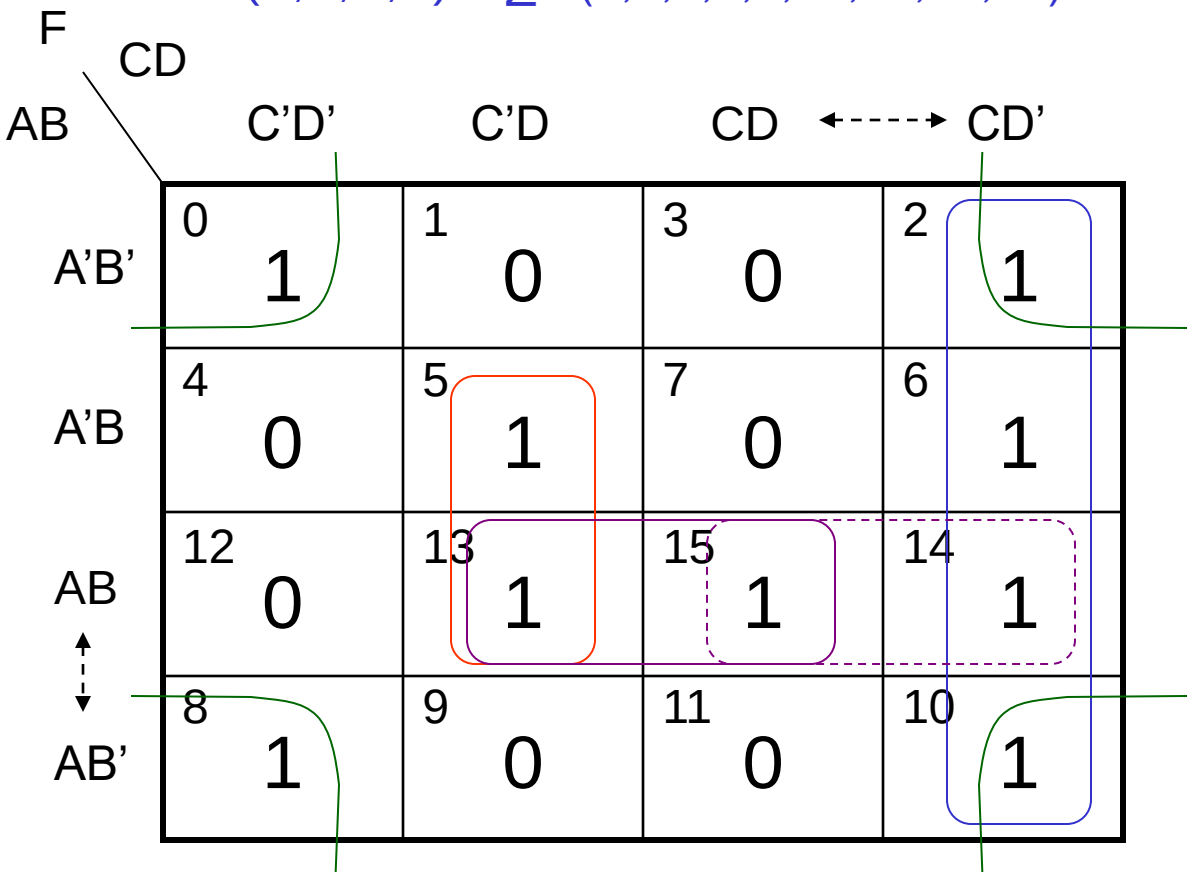


F = x'y'z' + w'xz' + xyz + wx'



Task 1: Ans.2

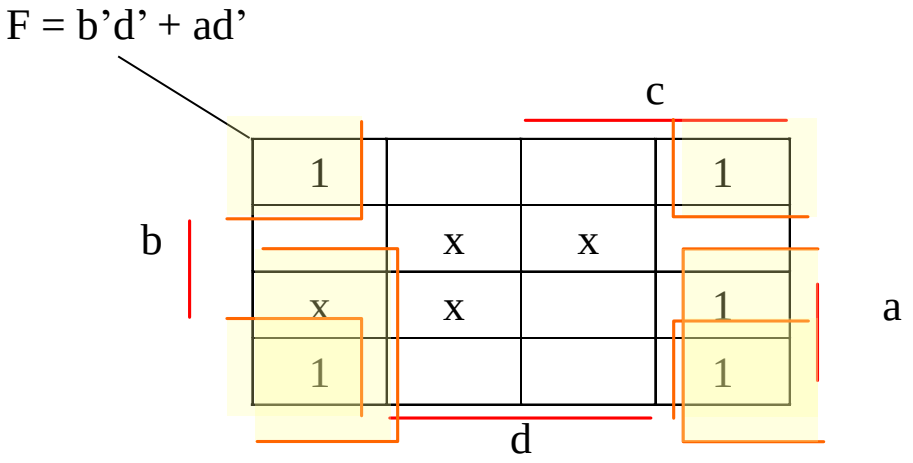
$F(A,B,C,D) = \sum m(0,2,5,6,8,10,13,14,15)$



$F = B'D' + CD' + BCD' + ABD$

Task 1: Ans.3

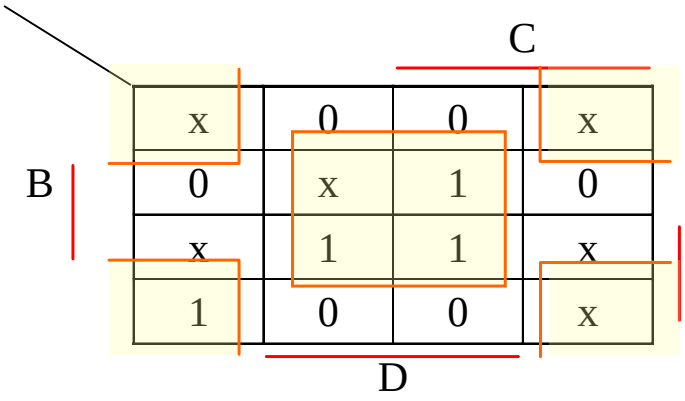
- Find the simplest sum of products form for the function F using the don't care condition G, where
  - F(a,b,c,d) = ∑m(0,2,8,10,14) and
  - G(a,b,c,d) = ∑m(5,7,12,13)



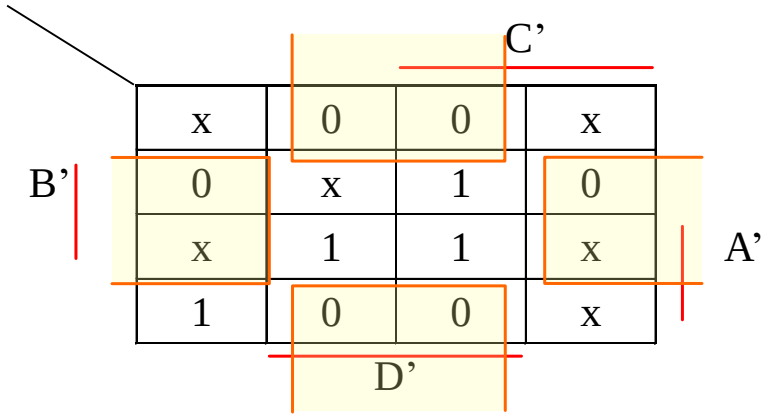
Task 1: Ans.4

- Simplify F(A,B,C,D) = ∏M(1,3,4,6,9,11) together with the don't care conditions d(A,B,C,D) = ∑m(0,2,5,10,12,14) in (i) *sum of products* and (ii) *products of sums*

i) F = B'D' + BD



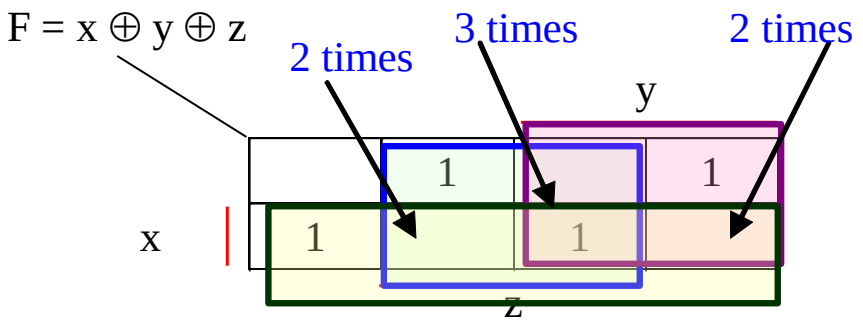
ii) F = (B'+D)(B+D')



Task 1: Ans.5

- Simplify the even parity function using XOR

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



Task 2: Ans.1

- Minimize the following functions using the *Quine-McCluskey* minimization:
  - F(x,y,z) = ∑m(0,1,2,4,6)
  - f1(x1, x2, x3, x4) = Σm(0, 6, 7, 9, 13, 14, 15)
  - f2(x1, x2, x3, x4, x5) = Σm(0, 2, 6, 10, 13, 16, 18, 20, 21, 23, 24, 26, 30, 31)

Task 2: Ans.2

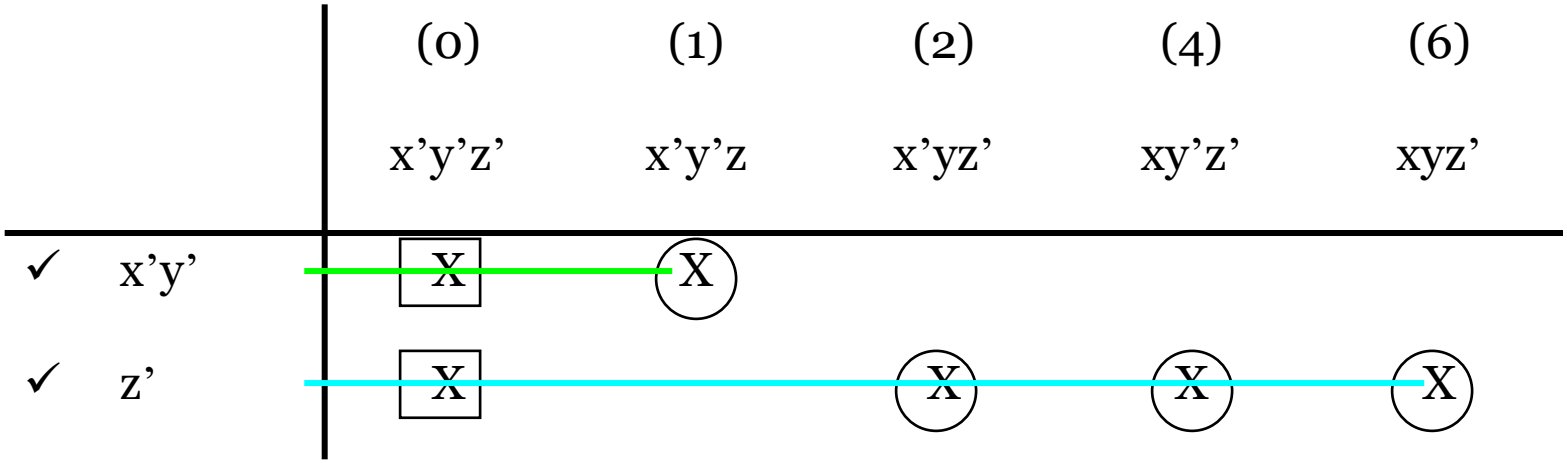
$$F(x,y,z) = \sum m\,(0,1,2,4,6)$$

no.		xyz		min	group
0	=	000	≡	x'y'z'	0
1	=	001	≡	x'y'z	1
2	=	010	≡	x'yz'	1
4	=	100	≡	xy'z'	1
6	=	110	≡	xyz'	2

Task 2: Ans.3

		(a)		(b)			(c)	
Group 0	(0)	x'y'z'	✓	x'y'	(0,1)		z'	(0,2,4,6)
Group 1	(1)	x'y'z	✓	x'z'	(0,2)	✓	~~z'~~	~~(0,4,2,6)~~
	(2)	x'yz'	✓	y'z'	(0,4)	✓		
	(4)	xy'z'	✓	yz'	(2,6)	✓		
Group 2	(6)	xyz'	✓	xz'	(4,6)	✓		

Task 2: Ans.4



✓ = essential prime implicant

F = x'y' + z'

Task 2: Ans.5

$f_1(x_1, x_2, x_3, x_4) = \sum(0, 6, 7, 9, 13, 14, 15)$

Calculation of the tuples:

	$x_4x_3x_2x_1$	2^1-tuple	2^2-tuple
0	0 0 0 0	(6,7) 0 1 1 - ✓	[(6,7)(14,15)] - 1 1 -
6	0 1 1 0 ✓	(6,14) - 1 1 0 ✓	~~[((6,14)(7,15)) - 1 1-~~
9	1 0 0 1 ✓	(9,13) 1 - 0 1	
7	0 1 1 1 ✓	(7,15) - 1 1 1 ✓	
13	1 1 0 1 ✓	(13,15) 1 1 - 1	
14	1 1 1 0 ✓	(14,15) 1 1 1 - ✓	
15	1 1 1 1 ✓		

Determination of the cheapest overlapping:

2^n	0	6	9	7	13	14	15	a
(6,7,14,15)		⊛		⊛		⊛	⊡	*
(9,13)			⊛		⊡			*
(13,15)					*		*	
(0)	⊛							*

(row a: * = essential prime implicant)

In this case these prime implicants already determine a cover
$\Rightarrow f = x_2x_3 + x_1x_2'x_4 + x_1'x_2'x_3'x_4'$

Task 2: Ans.6

b) $f_2(x_1, x_2, x_3, x_4, x_5) = \sum(0, 2, 6, 10, 13, 16, 18, 20, 21, 23, 24, 26, 30, 31)$

Calculation of the tuples:

	$x_5x_4x_3x_2x_1$	2^1-tuple	2^2-tuple	
0	0 0 0 0 0 ✓	(0,2) 0 0 0 - 0 ✓	(0,2,16,18) - 0 0 - 0	(0,16,2,18)
2	0 0 0 1 0 ✓	(0,16) - 0 0 0 0	(2,18,10,26) - - 0 1 0	
16	1 0 0 0 0 ✓	(2,6) 0 0 - 1 0	(16,18,24,26) 1 - 0 - 0	(16,24,18,26)
6	0 0 1 1 0 ✓	(2,10) 0 - 0 1 0		
10	0 1 0 1 0 ✓	(2,18) - 0 0 1 0 ✓		
18	1 0 0 1 0 ✓	(16,18) 1 0 0 - 0 ✓		
20	1 0 1 0 0 ✓	(16,20) 1 0 - 0 0		
24	1 1 0 0 0 ✓	(16,24) 1 - 0 0 0		
		(10,26) - 1 0 1 0 ✓		
13	0 1 1 0 1	(18,26) 1 - 0 1 0		
21	1 0 1 0 1 ✓	(20,21) 1 0 1 0 -		
26	1 1 0 1 0 ✓	(24,26) 1 1 0 - 0 ✓		
23	1 0 1 1 1 ✓	(21,23) 1 0 1 - 1		
30	1 1 1 1 0 ✓	(26,30) 1 1 - 1 0		
31	1 1 1 1 1 ✓	(23,31) 1 - 1 1 1		
		(30,31) 1 1 1 1 -		

Task 2: Ans.7

Determination of the cheapest overlapping:

2^n	0	2	16	6	10	18	20	24	13	21	26	23	30	31	a	b	c
(0,2,16,18)	⊛	⊡	⊡			⊡									*	*	*
(2,18,10,26)		*			*	*					*				**	**	**
(16,18,24,26)			*			⊡		⊛			⊡				*	*	*
(2,6)		⊡		⊛											*	*	*
(2,10)		*			*												
(16,20)			*				*										**
(20,21)							*			*					**	**	
(21,23)										*		*				**	**
(26,30)											*		*		**		
(23,31)												*		*	**		
(30,31)													*	*		**	**
(13)									⊛						*	*	*

(rows a,b,c: * = essential prime implicant, ** = additional implicants selected to get an overlapping)

One possible overlapping (using the implicants from row b)

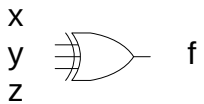
$$f = x'_1x'_3x'_4 + x'_1x_2x'_3 + x'_1x'_3x_5 + x'_1x_2x'_4x'_5 + x'_1x'_2x'_4x_5 + x_1x_3x'_4x_5 + x_2x_3x_4x_5 + x_1x'_2x_3x_4x'_5$$

Another possible overlapping (using the implicants from row c)

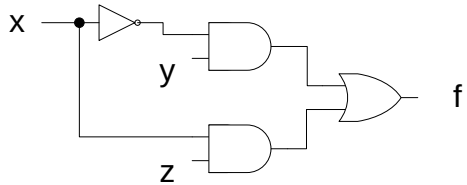
$$f = x'_1x'_3x'_4 + x'_1x_2x'_3 + x'_1x'_3x_5 + x'_1x_2x'_4x'_5 + x'_2x_3x'_4x_5 + x'_1x_2x_4x_5 + x_2x_3x_4x_5 + x_1x'_2x_3x_4$$

Task 3: Ans.1

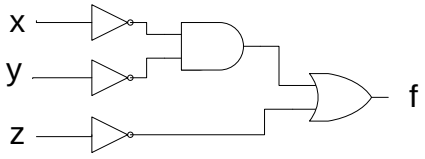
- Ignoring gate-input cost:
  - Implement all functions in Task 1 using any combination of logic gates



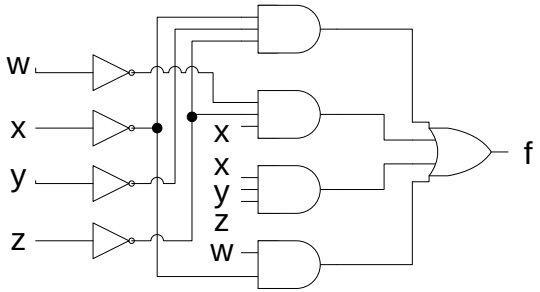
f = x ⊕ y ⊕ z



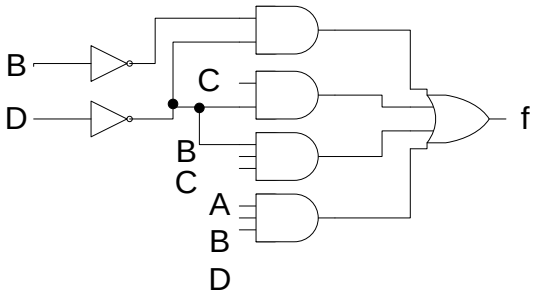
f = x'y + xz



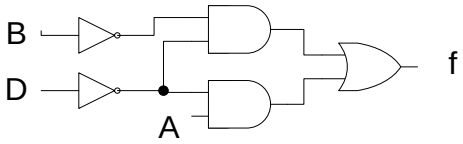
f = z' + x'y'



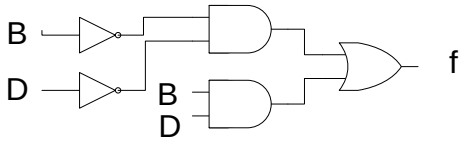
f = x'y'z + w'xz' + xyz + wx



f = B'D' + CD' + BCD' + ABD



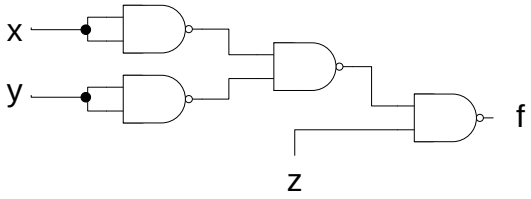
f = B'D' + AD'



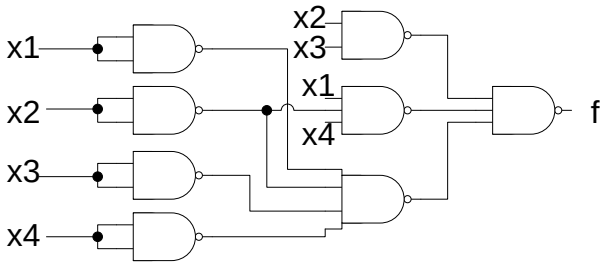
f = B'D' + BD

Task 3: Ans.1

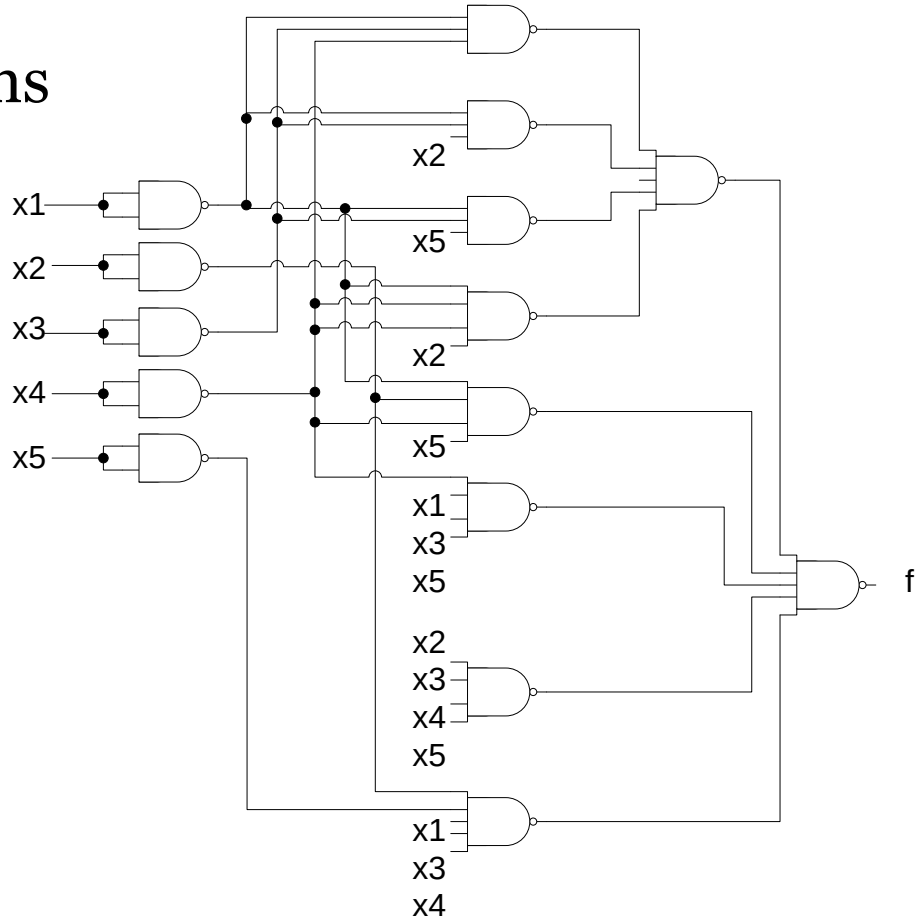
– Implement all functions in Task 2 using *NAND* only



f = x'y' + z



f = x2x3 + x1x2'x4 + x1'x2'x3'x4'

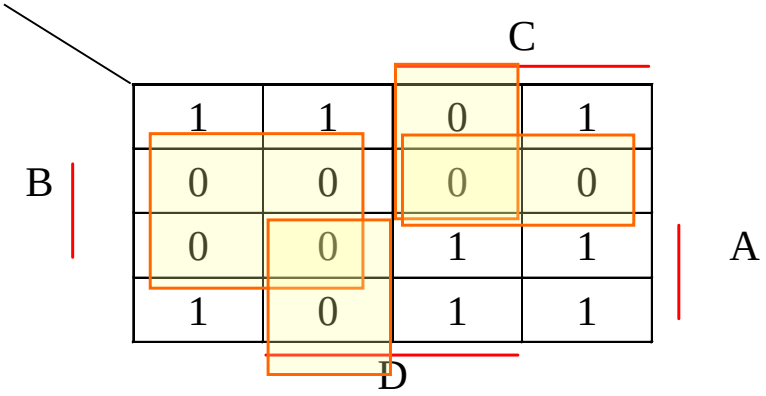


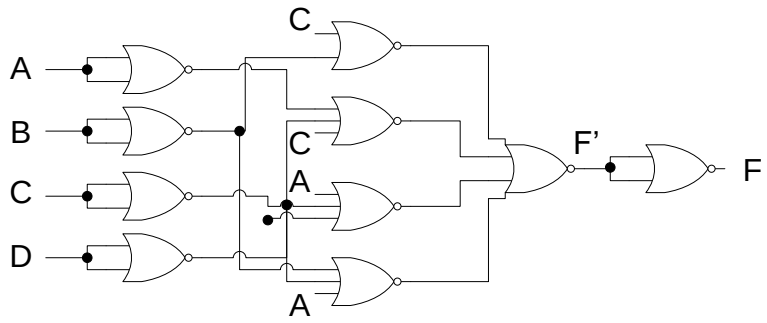
f = x1'x3'x4' + x1'x2x3' + x1'x3'x5 + x1'x2x4'x5' + x1'x2'x4'x5 + x1x3x4'x5 + x2x3x4x5 + x1x2'x3x4x5'

Task 3: Ans.1

– Draw the *NOR* implementation of the function F = Σm(0,1,2,8,10,11,14,15). Hint: Use F'=SOP.

F' = BC' + AC'D +A'CD + A'BC





F' = BC' + AC'D +A'CD + A'BC
= (B'+C)' + (A'+C+D')' + (A+C'+D')' + (A+B'+C')'