Examples

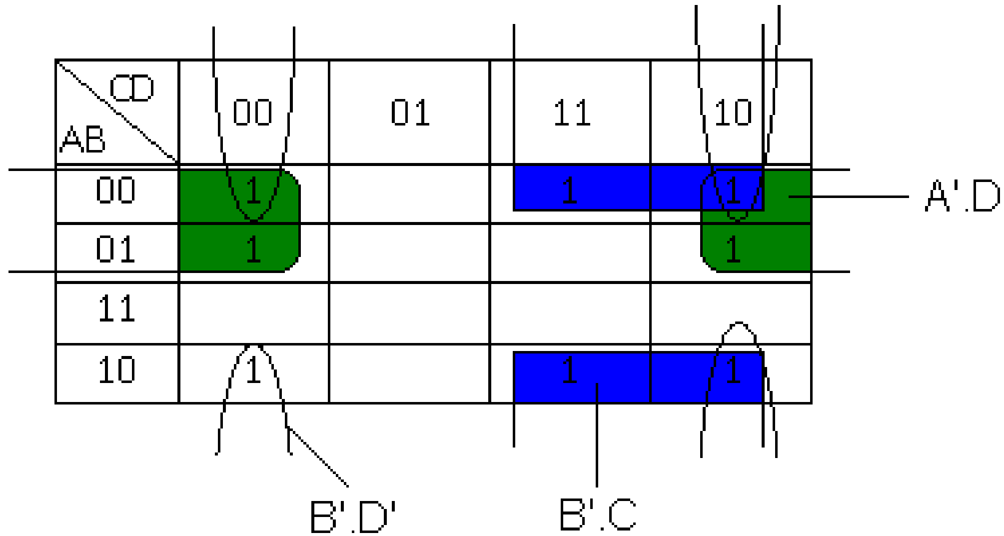
CONTENT:

- ❖ **Karnaugh Maps**
- ❖ **Don’t Care Conditions**
- ❖ **Simplification Using Tabular Method**
- ❖ **Implementing Circuits Using NAND or NOR Gates Only**
- ❖ **Decoders**
- ❖ **Multiplexers**

Example #1

Simplify F(A,B,C,D) = ∑ (0,2,3,4,6,8,10,11) using Karnaugh maps.

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



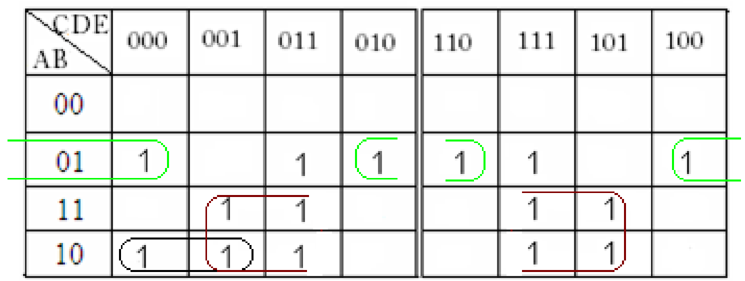
F(A,B,C,D) = A'D'+B'D'+B'C

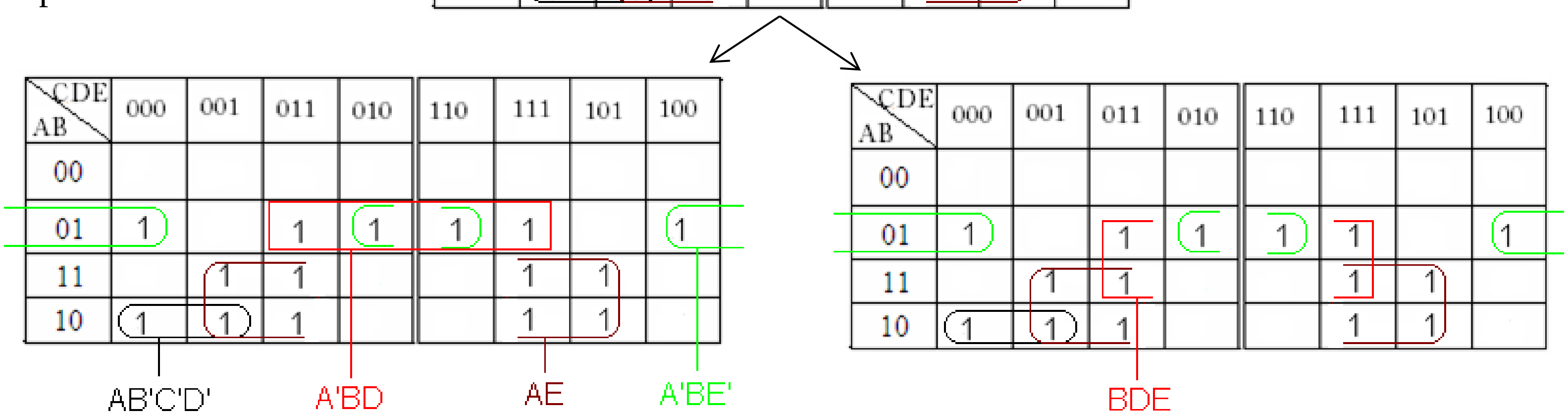
Example #2

The given Karnaugh map can be simplified in two different ways. Find both results.

AB \ CDE	000	001	011	010	110	111	101	100
00								
01	1		1	1	1	1		1
11		1	1			1	1	
10	1	1	1			1	1	

We need to group certain terms as shown on the right, as no other arrangement is possible for them. However, for the ungrouped terms on the right, we have two possible options.





Example #3

A combinational circuit design is required for an alarm system. The system has three inputs: signals from sensors indicating the door is locked (C), the door is open (B), and the window is open (A). As long as the door is not locked, the window and door can be opened. The alarm system is expected to generate an alarm signal (S) when either the window or door is open.

A: The window is open if it is 1.

B: The door is open if it is 1.

C: The door is locked if it is 1.

Since the door cannot be opened if it is locked, there are don't care conditions. Find the logical expression for output S.

C \ AB	00	01	11	10
0	0	1	1	1
1	0	x	x	x

C \ AB	00	01	11	10
0	0	1	1	1
1	0	x	x	x

S = A+B

Example #4

f(a,b,c,d) = ∑(1,4,6,7,8,9,10,11,15)

Determine the prime implicants and the essential prime implicants of f using the tabular (Quine-McCluskey) method.

Determining prime implicants:

	a	b	c	d			a	b	c	d			a	b	c	d
1	0	0	0	1	√	1-9	-	0	0	1		8-9-10-11	1	0	-	-
4	0	1	0	0	√	4-6	0	1	-	0						
8	1	0	0	0	√	8-9	1	0	0	-	√					
6	0	1	1	0	√	8-10	1	0	-	0	√					
9	1	0	0	1	√	6-7	0	1	1	-						
10	1	0	1	0	√	9-11	1	0	-	1	√					
7	0	1	1	1	√	10-11	1	0	1	-	√					
11	1	0	1	1	√	7-15	-	1	1	1						
15	1	1	1	1	√	11-15	1	-	1	1						

Example #4 (continued)

Determining essential prime implicants:

Minterms / Prime Implicants	**1** √	**9** √	**4** √	**6** √	**7**	**15**	**11** √	**8** √	**10** √	
b’c’d m(1,9)	**x**	x								√
a’bd’ m(4,6)			**x**	x						√
a’bc m(6,7)				x	x					
bcd m(7,15)					x	x				
acd m(11,15)						x	x			
ab’ m(8,9,10,11)		x					x	**x**	**x**	√

f (a,b,c,d) = b’c’d + a’bd’ + ab’ + bcd

Example #5

Implement f(a,b,c) = a'c + bc + a'b using NOR gates only.

The function f is in the form of a sum of products. If it were to be implemented using NAND gates, it would be sufficient to take the negation of the entire expression twice. However, since it is required to be implemented with NOR gates, the negation of each product term must be taken twice.

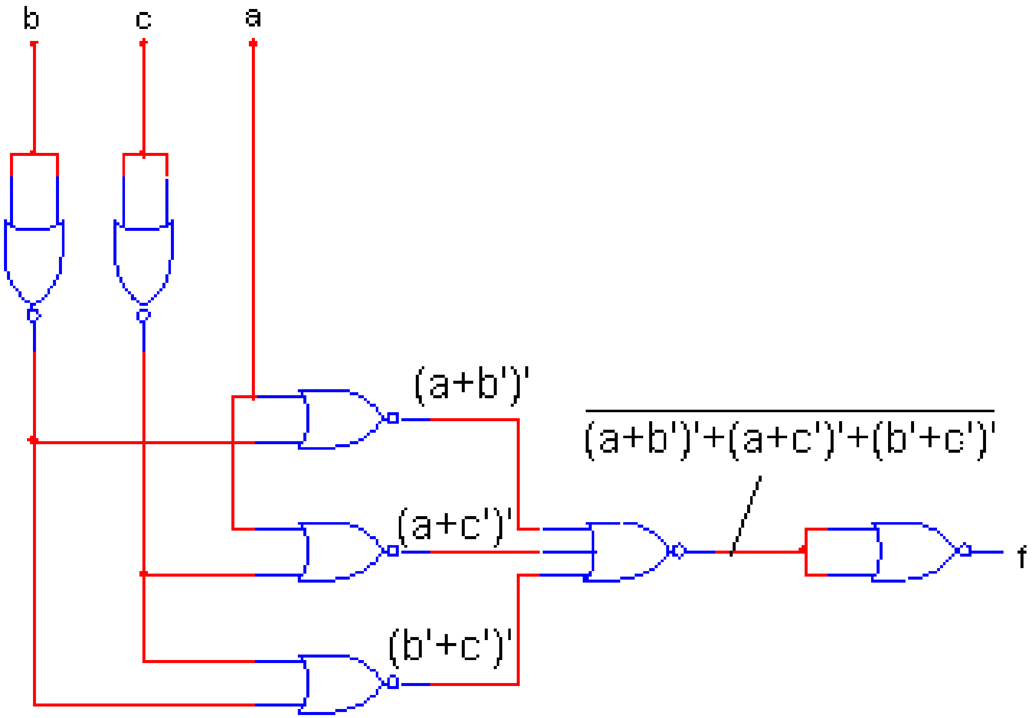
$f(a,b,c) = [(a'c)']' + [(bc)']' + [(a'b)']' = (a+c')' + (b'+c')' + (a+b')'$

Each term in the function f is in NOR form. However, to bring the entire function f into NOR form, it is necessary to take the negation of the entire expression twice.

$$f(a,b,c) = \overline{\overline{(a + c')' + (b' + c')' + (a + b')'}}$$

Example #5 (continued)

$$f(a,b,c) = \overline{(a + c')' + (b' + c')' + (a + b')'}$$

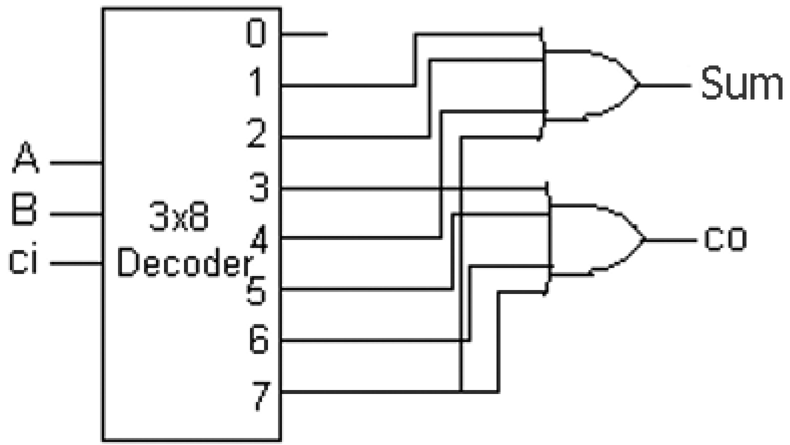


Example #6

Implement a full adder using a decoder and OR gates.

The truth table for a full adder is as below.

Inputs			Outputs	
A	B	ci	Sum	co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

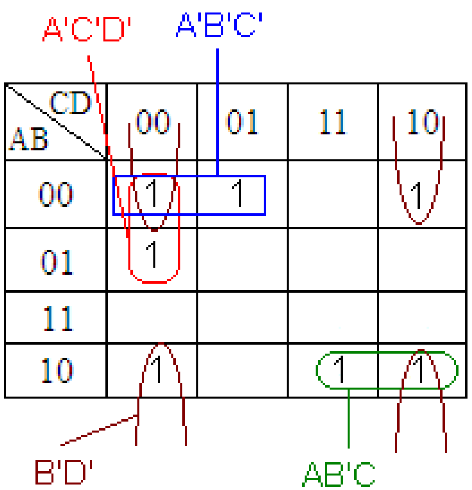


$$\text{Sum} = \sum(1,2,4,7)$$
$$\text{co} = \sum(3,5,6,7)$$

Example #7

Function f(A,B,C,D) is given on the Karnaugh map below. Implement this function using a 4×1 MUX and logic gates. Let the selection lines of the MUX be $S_1\ S_0$ = A B.

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



$$
\begin{aligned}
f(A,B,C,D) &= A'B'C' + AB'C + A'C'D' + B'D' \\
&= A'B'C' + AB'C + A'(B+B')C'D' + (A+A')B'D' \\
&= \underline{A'B'C'} + \underline{\underline{AB'C}} + A'BC'D' + \underline{A'B'C'D'} + \underline{\underline{AB'D'}} + \underline{A'B'D'} \\
&= A'B'(C' + \underline{C'D'} + \underline{D'}) + AB'(C+D') + A'BC'D' \\
&= A'B'(C'+D') + AB'(C+D') + A'BC'D'
\end{aligned}
$$

Output expression of a 4×1 MUX is $A'B'I_0 + A'BI_1 + AB'I_2 + ABI_3$

$I_0 = (C'+D') = (C.D)'$

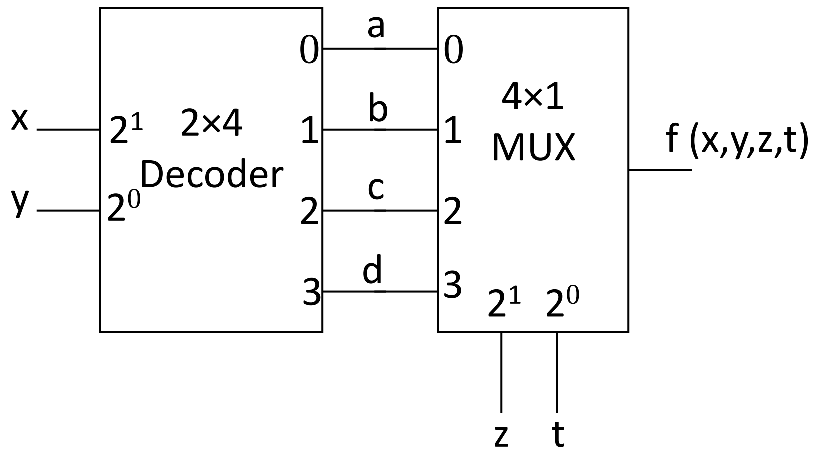
$I_1 = C'D' = (C+D)'$

$I_2 = C+D'$

$I_3 = 0$

Example #8

Find the output function f in terms of minterms.



The output expression of the MUX is:

$f = z't'.a + z't.b + zt'.c + zt.d$

Decoder's output expressions are:

$a = x'y' \quad b = x'y \quad c = xy' \quad d = xy$

If these expressions are substituted into the definition of the MUX, then...

$f = z't'.\, x'y' + z't.\, x'y + zt'.\, xy' + zt.\, xy$

When we rearrange the f function, we get:

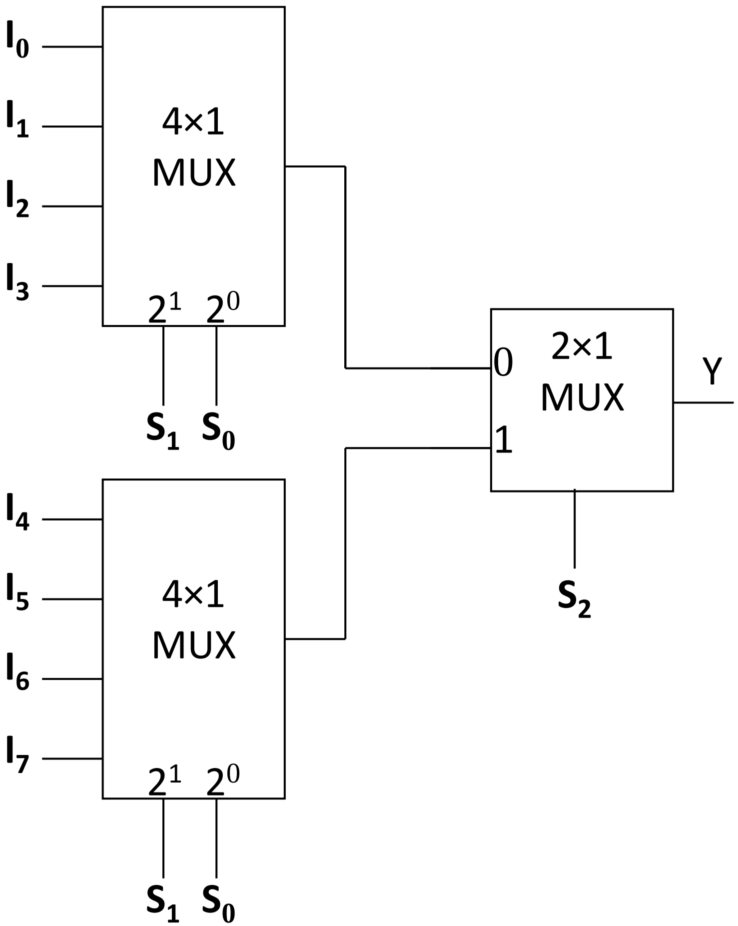
$f(x,y,z,t) = x'y'z't' + x'yz't + xy'zt' + xyzt = \sum(0,5,10,15)$

Example #9

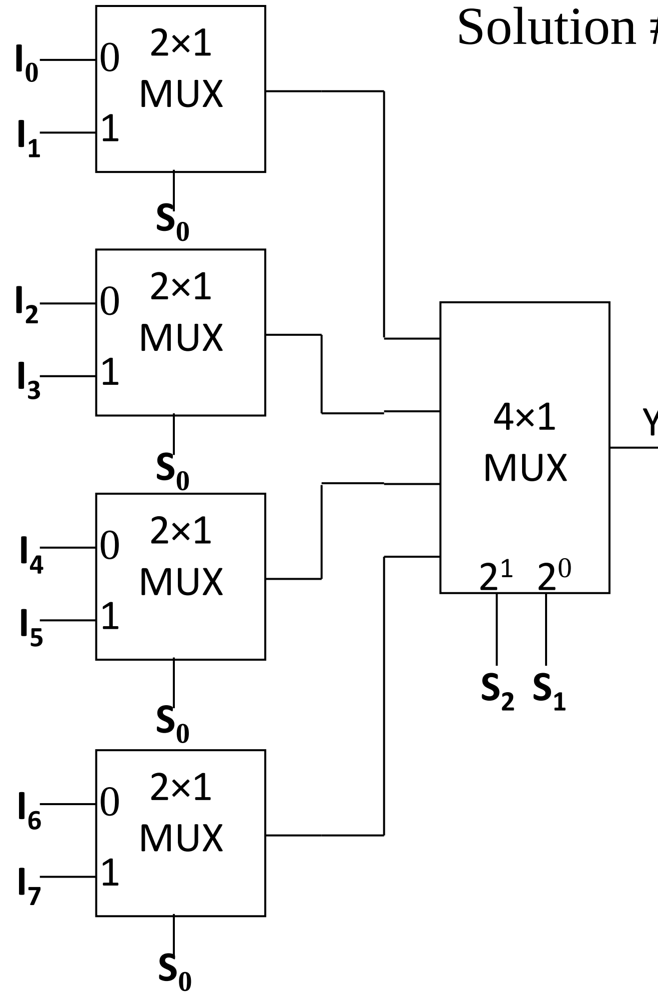
Implement an 8×1 MUX using 4×1 and 2×1 MUXs only.

Let the select lines be $S_2S_1S_0$. MUXs don't have enable inputs.

Solution #1



Solution #2



Example #10

$f(a,b,c,d) = \sum (1,2,4,7,8,11,13,14)$

Implement function f(a,b,c,d) using an 8×1 MUX.

Let the select lines be a, b, and c.

d \ abc	000	001	010	011	100	101	110	111
0	m_0	m_2	m_4	m_6	m_8	m_{10}	m_{12}	m_{14}
1	m_1	m_3	m_5	m_7	m_9	m_{11}	m_{13}	m_{15}
Inputs	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
Connections	**d**	**d’**	**d'**	**d**	**d'**	**d**	**d**	**d'**