Name: ____________________

Problem	Points	Score
1a	10	
1b	10	
1c	10	
1d	10	
2a	10	
2b	10	
3a	10	
3b	10	
3c	10	
3d	10	
Total	**100**	

Notes:

1. The exam is closed book / closed notes. Students are allowed a copy sheet — only **one** side of **one** standard US-size (8.5” x 11”) sheet of paper — on which they can write relevant information such as theorems.

2. Please show ALL work. Incorrect answers with no supporting explanations or work will be given no partial credit.

3. If I can’t read or follow your solution, it is wrong, and no partial credit will be given — PLEASE BE NEAT!

4. Please indicate clearly your answer to every problem.

5. There is sufficient space after each problem to write your solution. In case you need extra paper please see the instructor.

6. Calculators of any kind are not allowed.

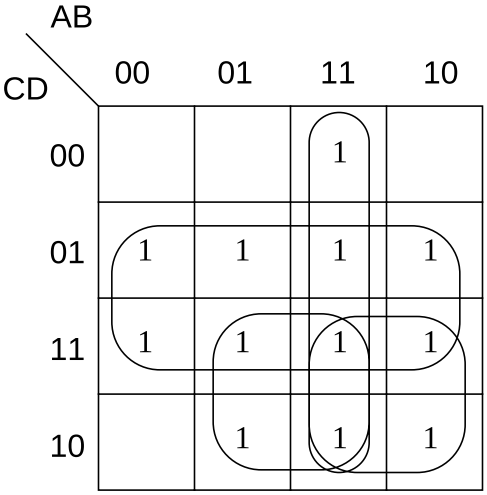
Problem No. 1:

The locking mechanism to the entrance of your company building uses a 4-bit code to open. To unlock the door, you must enter 4 bits ABCD such that the corresponding decimal number is odd or a multiple of some odd number greater than 1. In other words, the door unlocks if the input is not 0, 2, 4 or 8.

a) Draw the truth table and the corresponding Karnaugh map to represent the network implementing this lock. An output of 1 indicates that the door is unlocked.

Solution:

ABCD	F
0000	0
0001	1
0010	0
0011	1
0100	0
0101	1
0110	1
0111	1
1000	0
1001	1
1010	1
1011	1
1100	1
1101	1
1110	1
1111	1



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

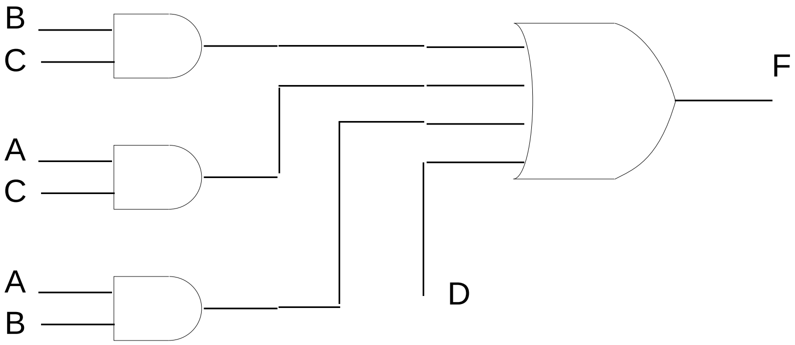
b) Realize a minimum network for this locking mechanism using two-level logic and only NAND gates. Write the number of gate inputs required.

Solution:

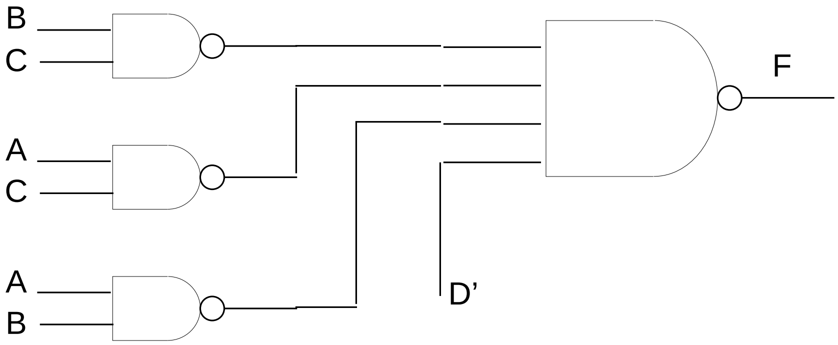
From the K-map in part **a**, we get the minimum two-level sum-of-products expression for F as

$$F = D + BC + AC + AB$$

Therefore the corresponding AND-OR network is as follows —



Now we convert this to a NAND-only network by changing all gates to NAND, and complementing the variables connected directly to the output gate. The resulting network is



This network requires 4 NAND gates and 10 gate inputs.

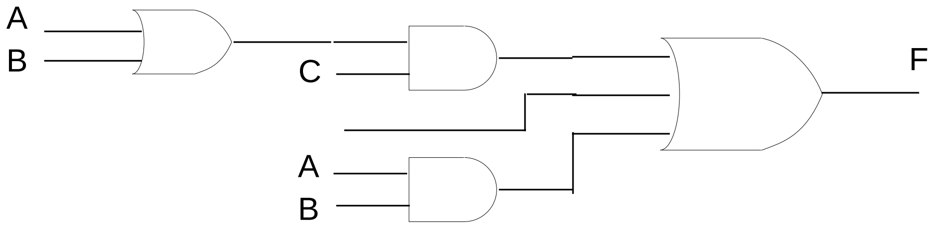
c) Realize a minimum network for this locking mechanism using three-level logic and AND-OR gates. Write the number of gate inputs required.

Solution:

We have

$$\begin{aligned} F &= D + BC + AC + AB \\ &= D + C(A+B) + AB \end{aligned}$$

This yields the following three-level minimum network —



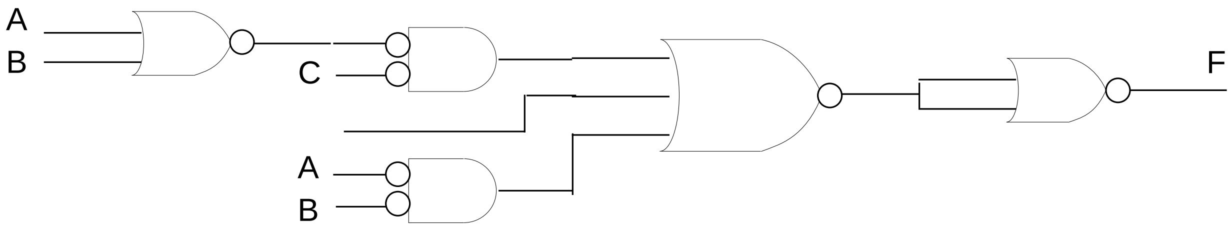
This network requires 4 gates and 9 gate inputs.

d) Convert the three-level network of part **c** into a network with only NOR gates.

Solution:

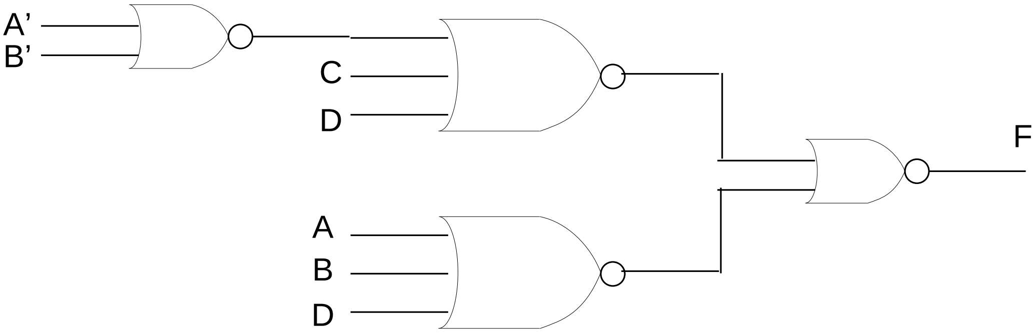
This was slightly tricky, because to directly apply the conversion rules for NOR gate networks, the original network should have an AND gate at the output. Therefore, one can follow two methods —

Method 1: Use the bubbles method, but complement the output to balance the extra bubble at the output.



Method 2: Start with a product-of-sums form expression from the K-map and convert it to a three-level NOR gate network.

$$
\begin{aligned}
F &= (A + C + D)(B + C + D)(A + B + D) \\
&= (AB + C + D)(A + B + D) \\
&= [(AB + C + D)' + (A + B + D)']' \\
&= [((A' + B')' + C + D)' + (A + B + D)']'
\end{aligned}
$$



Problem No. 2:

You have been recently hired by an electronics company. Your predecessor was fired because he did not know how to design networks with more than 4 inputs. You are now assigned to his unfinished project to design the following 5-input network —

$$F = \sum m(0, 1, 2, 6, 8, 9, 10, 11, 20, 23, 28, 31) + \sum d(3, 17, 21, 25, 30)$$

a) Your predecessor has left the following unfinished Quine-McCluskey table. Complete it to obtain all the prime implicants.

✓	0,1	0 0 0 0 -
✓	0,2	0 0 0 - 0
✓	0,8	0 - 0 0 0
✓	1,3	0 0 0 - 1
✓	1,9	0 - 0 0 1
✓	1,17	- 0 0 0 1
✓	2,3	0 0 0 1 -
	2,6	0 0 - 1 0
✓	2,10	0 - 0 1 0
✓	8,9	0 1 0 0 -
✓	8,10	0 1 0 - 0
✓	3,11	0 - 0 1 1
✓	9,11	0 1 0 - 1
✓	9,25	- 1 0 0 1
✓	10,11	0 1 0 1 -
	17,21	1 0 - 0 1
✓	17,25	1 - 0 0 1
	20,21	1 0 1 0 -
	20,28	1 - 1 0 0
	21,23	1 0 1 - 1
	28,30	1 1 1 - 0
	23,31	1-111
	30,31	1111-

Solution:

✓	0,1,2,3	0 0 0 - -
✓	0,1,8,9	0 - 0 0 -
	~~0,2,1,3~~	~~0 0 0 - -~~
✓	0,2,8,10	0 - 0 - 0
	~~0,8,1,9~~	~~0 - 0 0 -~~
	~~0,8,2,10~~	~~0 - 0 - 0~~
✓	1,3,9,11	0 - 0 - 1
	~~1,9,3,11~~	~~0 - 0 - 1~~
	1,9,17,25	- 0 - 0 1
	~~1,17,9,25~~	~~- 0 1 0 1~~
✓	2,3,10,11	0 - 0 1 -
	~~2,10,3,11~~	~~0 - 0 1 -~~
✓	8,9,10,11	0 1 0 - -
	~~8,10,9,11~~	~~0 1 0 - -~~

0,1,2,3, 8,9,10,11	0 - 0 - -
~~0,1,8,9,~~ ~~2,3,10,11~~	~~0 - 0 - -~~
~~0,8,2,10,~~ ~~1,3,9,11~~	~~0 - 0 - -~~

The corresponding prime implicants are —

(0,1,2,3,8,9,10,11)
(2,6)
(20,21)
(21,23)
(23,31)
(1,9,17,25)
(17,21)
(20,28)
(28,30)
(30,31)

Of these, (17,21) is not required as it consists of only don't-care minterms.

b) Draw the prime implicant chart from the completed table in part **a** and find all minimum sum-of-product expressions for F.

	0	1	2	6	8	9	10	11	20	23	28	31
❊ (0,1,2,3,8,9,10,11)	×	×	×		×	×	×	×				
(1,9,17,25)		×				×						
❊ (2,6)			×	×								
(20,21)									×			
✓ (20,28)									×		×	
(21,23)										×		
(28,30)											×	
✓ (23,31)										×		×
(30,31)												×

The minimum expression for F is, therefore

$$F = A'C' + A'B'DE' + ACD'E' + ACDE$$

Problem No. 3:

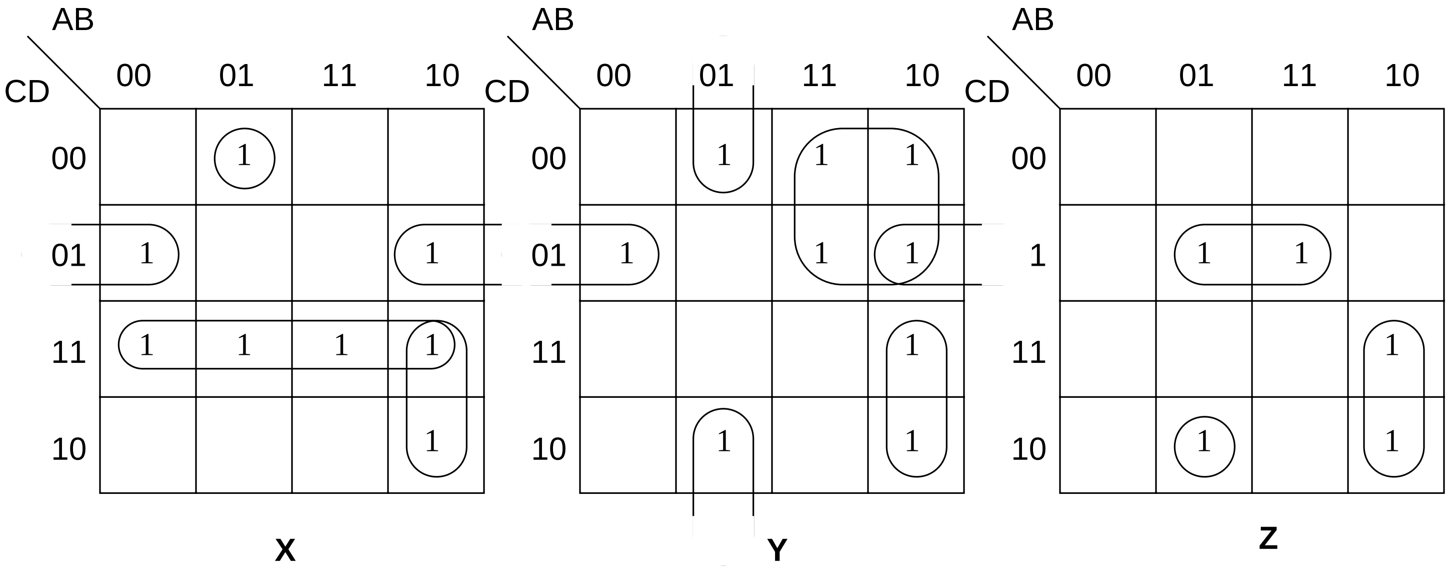
A leading medical electronics company is designing a logic network that will perform preliminary diagnosis on patients based on the presence or absence of four symptoms ABCD. The diseases to be diagnosed are XYZ, and their relationship with the symptoms is as follows —

ABCD	XYZ
0000	000
0001	110
0010	000
0011	100
0100	110
0101	001
0110	011
0111	100

ABCD	XYZ
1000	010
1001	110
1010	111
1011	111
1100	010
1101	011
1110	000
1111	100

a) Draw the K-maps for X, Y and Z to implement this network as a 4-input 3-output system.

Solution:



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

X

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

Y

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

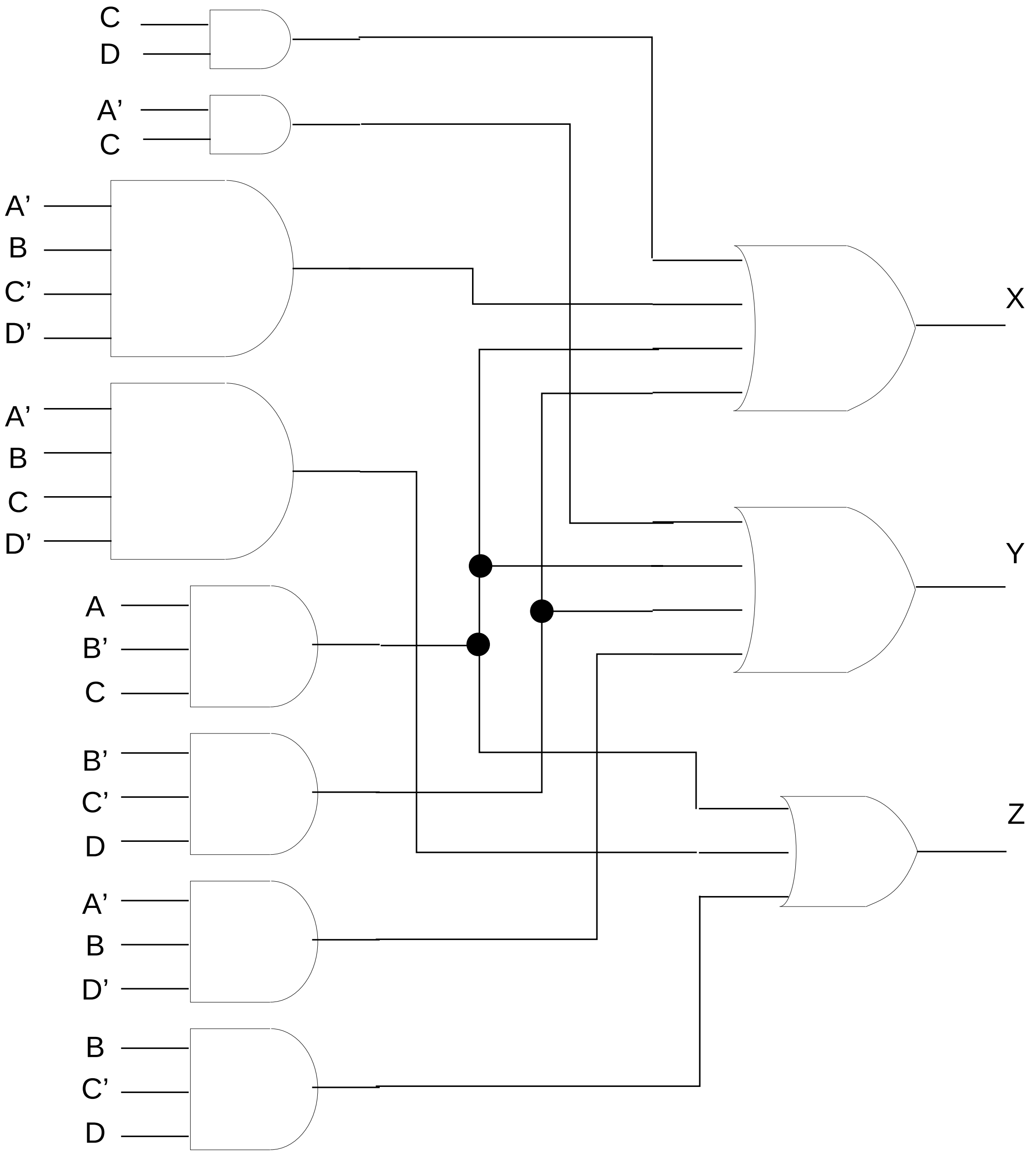
Z

b) Find a minimum 2-level AND-OR network to implement this system. Draw the corresponding network diagram.

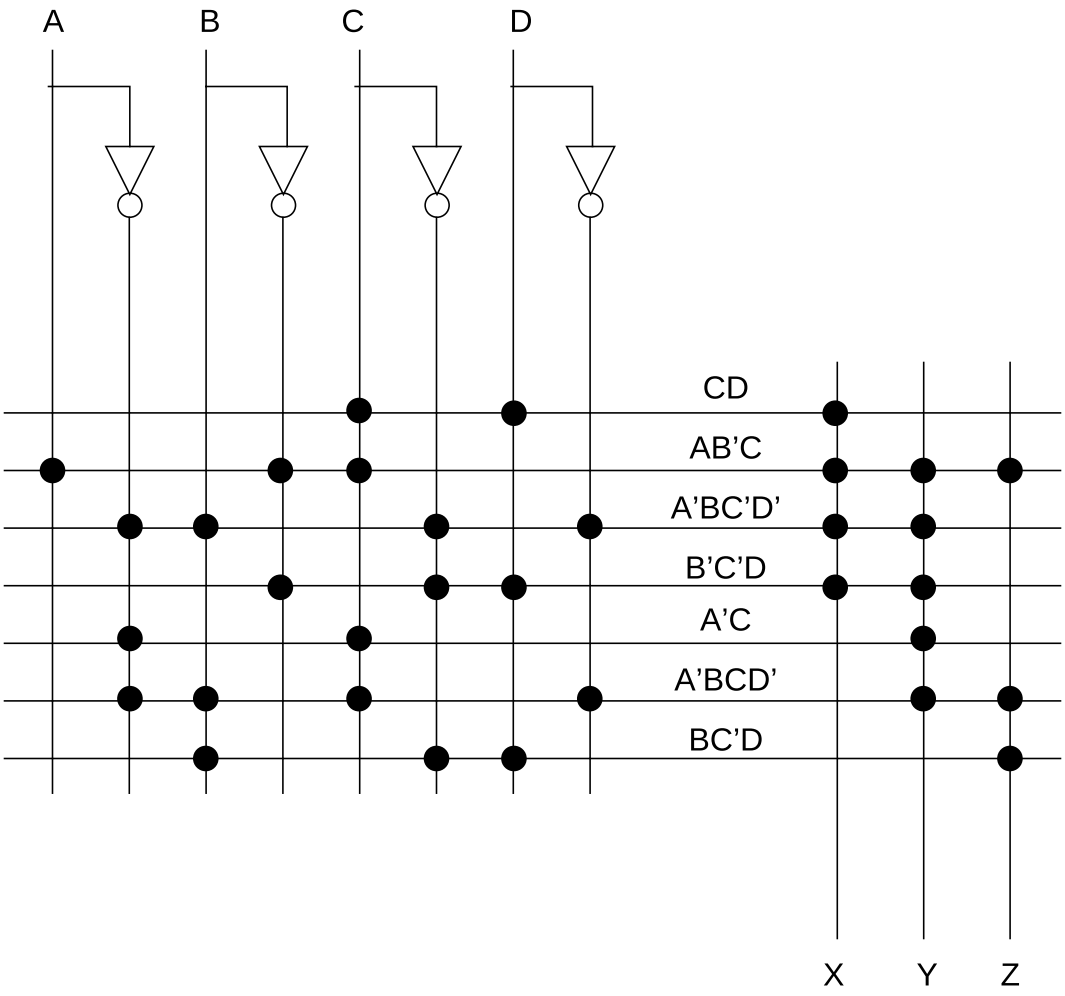
Solution:

From the K-maps in part **a**, we can see that the minimum 2-level network is —

$$X = CD + AB'C + A'BC'D' + B'C'D$$
$$Y = A'C + AB'C + A'BD' + B'C'D$$
$$Z = AB'C + A'BCD' + BC'D$$



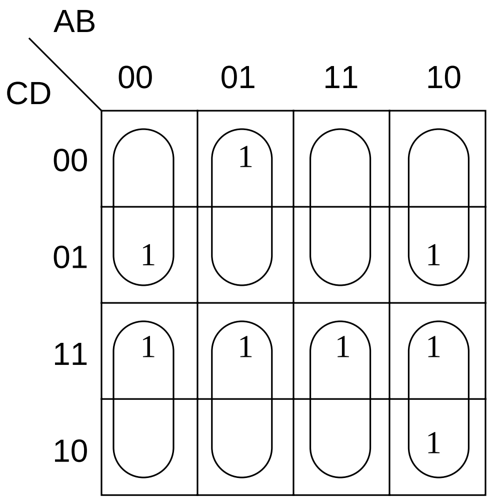
c) Implement the network in part **b** on the PLA shown below. Mark the connections to create the required AND and OR terms.



Note that we split the A'BD' term for Y and used the A'BC'D' and A'BCD' terms used for X and Z instead to generate it.

d) Implement the function for X using an 8-to-1 multiplexer with A, B and C as the control inputs.

Solution:



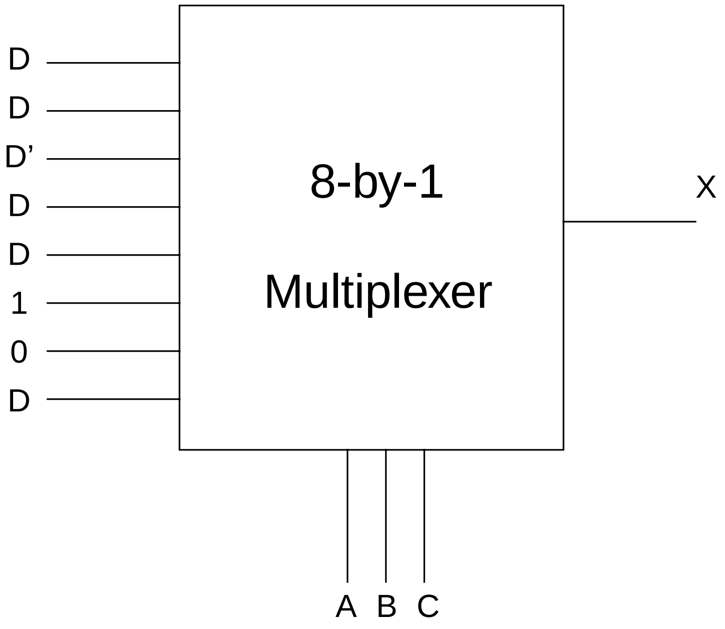
CD \ AB	00	01	11	10
00		1		
01	1			1
11	1	1	1	1
10				1

X

From the adjacent K-map we can see that

ABC	**X**
000	D
001	D
010	D’
011	D
100	D
101	1
110	0
111	D

Therefore, the MUX design is as follows —


D
D
D’
D
D
1
0
D
8-by-1
Multiplexer
X
A B C