

Theory Assignment 2

1.

a)

$$\begin{aligned}T_1 &= (A'T_2)' = (A'(A'D)')' \\T_2 &= (A'D)' \\T_3 &= A' + BC \\F &= T_1T_3 = (A'(A'D)')'(A' + BC) \\G &= (T_2T_3)' = ((A'D)'(A' + BC))'\end{aligned}$$

b)

$$\begin{aligned}F &= (A'(A'D)')'(A' + BC) \\&= (A + A'D)(A' + BC) \\&= AA' + ABC + A'D + A'BCD \\&= ABC + A'D + A'BCD \\&= ABC + A'D(1 + BC) \\&= \boxed{ABC + A'D}\end{aligned}$$

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

c)

We first write both functions in minterms:

$$\begin{aligned}F &= ABC + A'D \\&= ABC(D + D') + A'(B + B')(C + C')D \\&= ABCD + ABCD' + A'BCD + A'BC'D + A'B'CD + A'B'C'D \\&= \sum(1, 3, 5, 7, 14, 15)\end{aligned}$$

$$\begin{aligned}
G &= AB' + AC' + A'D \\
&= AB'(C + C')(D + D') + A(B + B')C'(D + D') + A'(B + B')(C + C')D \\
&= AB'CD + AB'C'D + AB'CD' + AB'C'D' + ABC'D + ABC'D' \\
&\quad + A'BCD + A'B'CD + A'BC'D + A'B'C'D \\
&= \sum(1, 3, 5, 7, 8, 9, 10, 11, 12, 13)
\end{aligned}$$

Then, we can write the truth table:

A	B	C	D	F	G
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	0

2.

a)

A_3	A_2	A_1	A_0	P	D
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	1	0	1

b)

$A_1A_0 \backslash A_3A_2$	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

P

$A_1A_0 \backslash A_3A_2$	00	01	11	10
00	1	0	1	0
01	0	0	0	1
11	1	0	1	0
10	0	1	0	0

D

$$P = A'_3A'_2A_1 + A'_3A_1A_0 + A_2A'_1A_0 + A'_2A_1A_0$$

$$D = A'_3A'_2A'_1A'_0 + A'_3A'_2A_1A_0 + A'_3A_2A_1A'_0 + A_3A'_2A'_1A_0 + A_3A_2A'_1A'_0 + A_3A_2A_1A_0$$

3.

By using active-low decoder and NAND gates, the circuit will perform as if it is an active-high decoder with OR gates. This is because:

$$\begin{aligned} \sum(a, b..c) &= m_a + m_b + \dots + m_c \\ &= ((m_a + m_b + \dots + m_c)')' \\ &= (m'_a \cdot m'_b \cdot \dots \cdot m'_c)' \end{aligned}$$

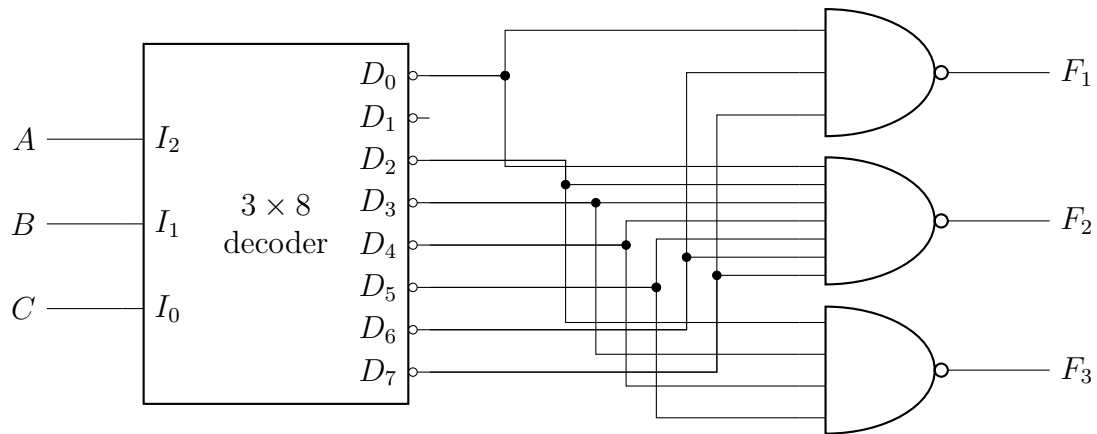
We just need to find sum-of-minterms expressions for each function:

$$F_1 = AB + A'B'C' = \sum(0, 6, 7)$$

$$F_2 = A + B + C' = \sum(0, 2, 3, 4, 5, 6, 7)$$

$$F_3 = A'B + AB' = \sum(2, 3, 4, 5)$$

Then, we can draw the block diagram:



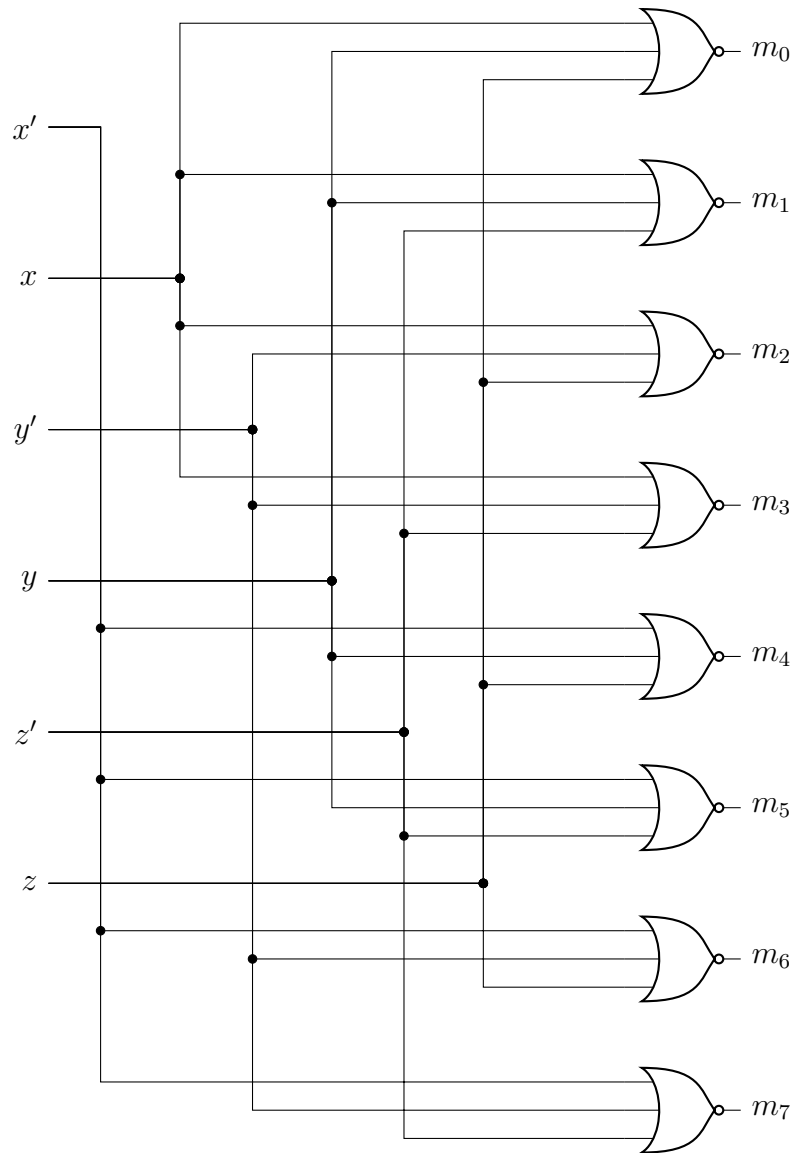
4.

By applying bubble pushing, we can find all input should be inverted when using NOR gates. For example:

$$\begin{aligned} m_0 &= x'y'z' \\ &= (x + y + z)' \end{aligned}$$

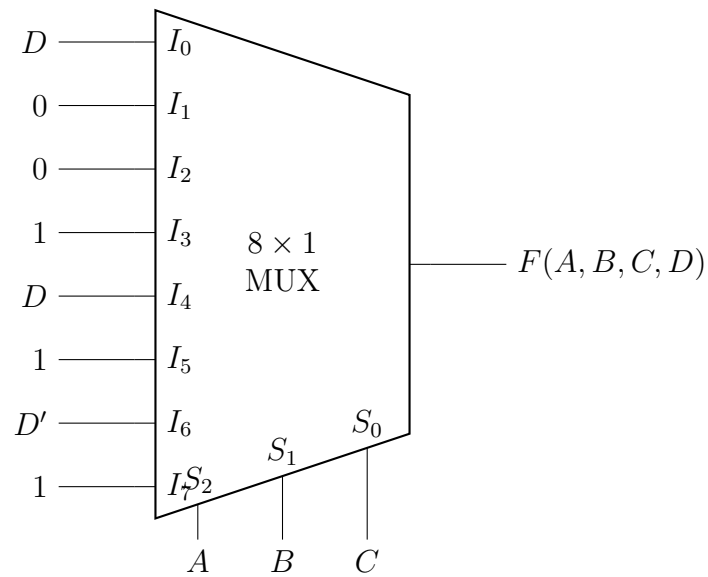
The m_0 minterm used to be conjunction of x' , y' and z' , but now it is disjunction of x , y and z followed by a NOT gate.

Therefore, we can draw the block diagram:



5.

a)



The truth tables are:

$S_2(A)$	$S_1(B)$	$S_0(C)$	$F(A, B, C, D)$
0	0	0	D
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	D
1	0	1	1
1	1	0	D'
1	1	1	1

MUX truth table

A	B	C	D	$F(A, B, C, D)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

function truth table

b)

$\begin{array}{c} C/D \\ \backslash AB \end{array}$	00	01	11	10
00	0	1	0	0
01	0	0	1	1
11	1	0	1	1
10	0	1	1	1

The simplified function is:

$$F(A, B, C, D) = AC + ABD' + BC + B'C'D$$

6.

a)

A	B	C	D	$F(A, B, C, D)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	x
0	1	0	1	x
0	1	1	0	0
0	1	1	1	0
1	0	0	0	x
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

b)

By using K-map, we can find the simplified function in SOP and POS form:

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

SOP form

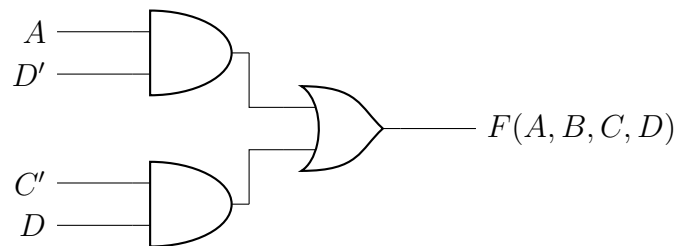
AB \ CD	00	01	11	10
00	0	1	0	0
01	x	x	0	0
11	1	1	0	1
10	x	1	0	1

POS form

Therefore, the simplified function is:

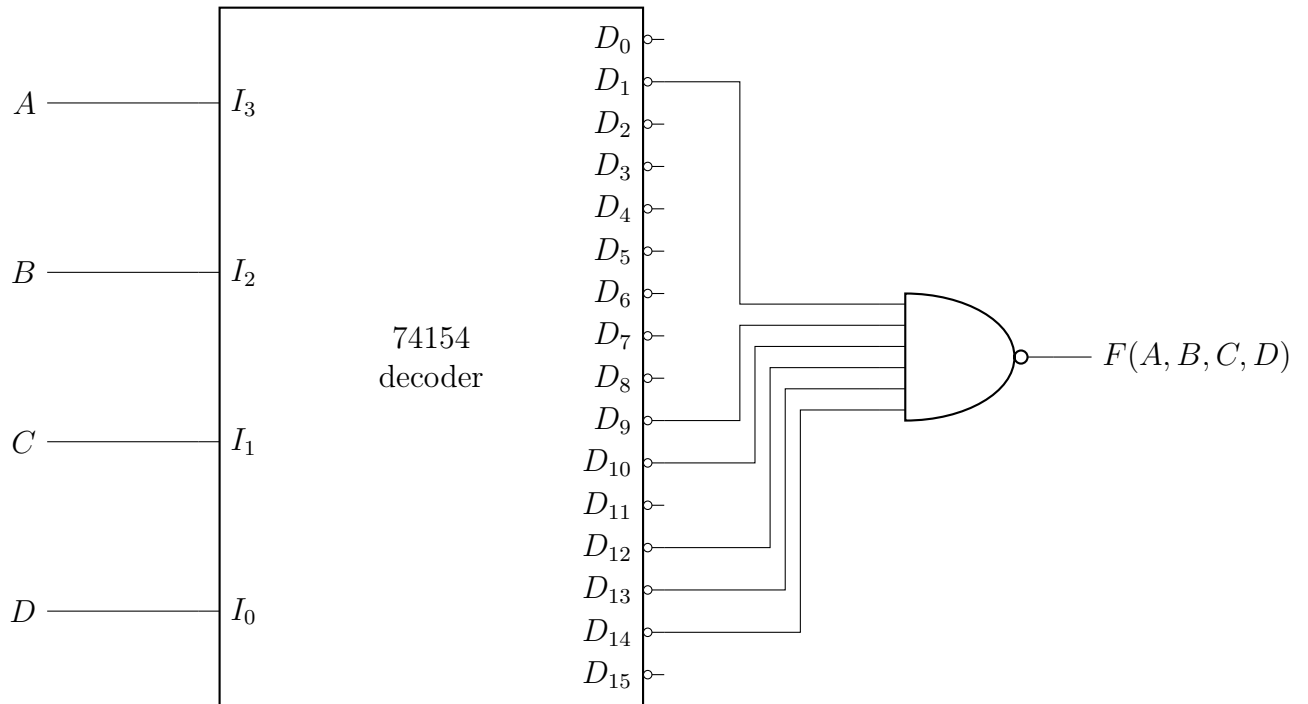
$$\begin{aligned}
 F(A, B, C, D) &= AD' + C'D \\
 &= (A + D)(C' + D')
 \end{aligned}$$

In this case, any implementation of the function will have at least 3 gates. The logic diagram for SOP form is:



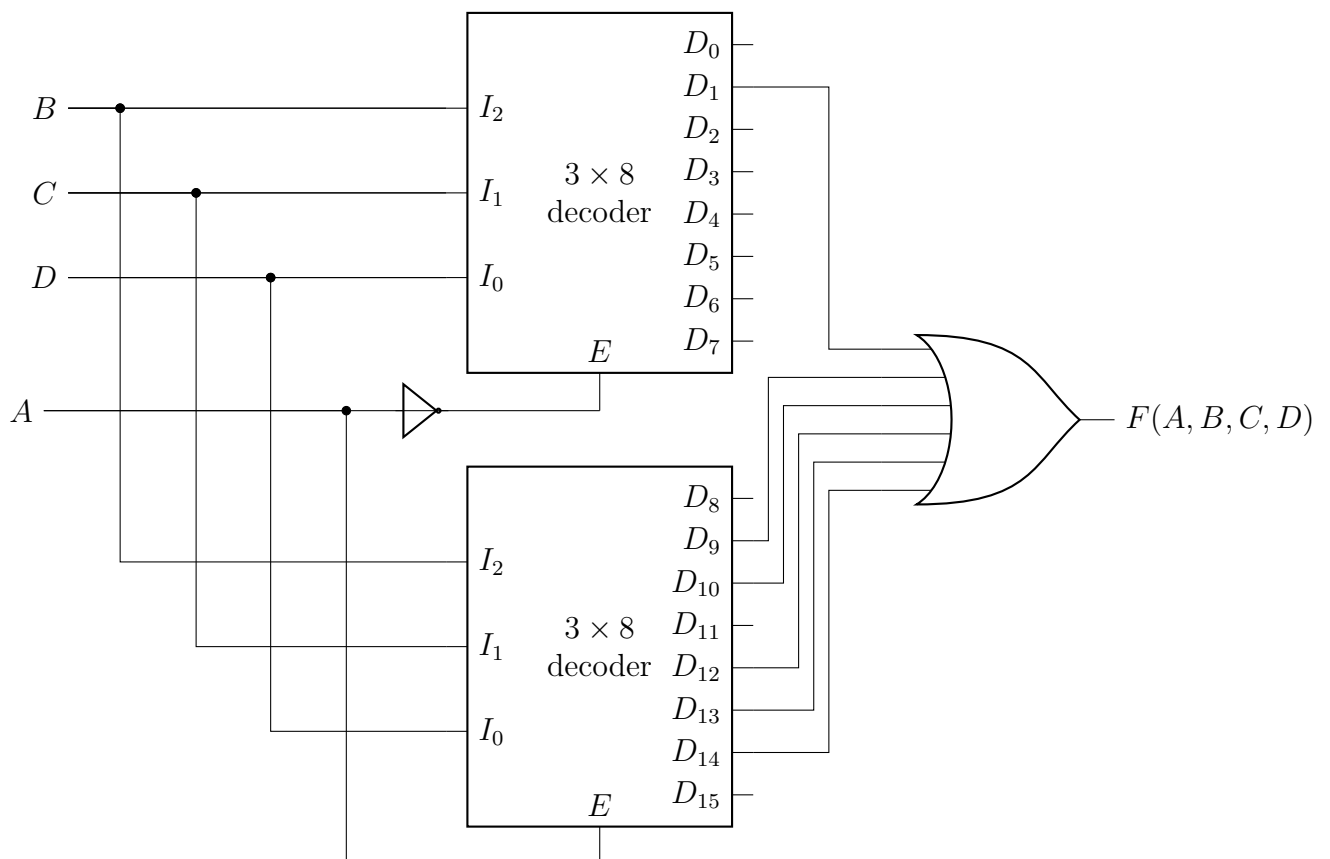
c)

We simply connect all minterms output to a OR gate.



d)

We implement a 4-to-16 decoder with two 3-to-8 decoders. X is don't care condition, it can be considered as 0 or 1, to minimize the logic gate usage, we consider the Xs to be 0s. Thus, the output can be expressed in sum of minterm form: $F = \sum(1, 9, 10, 12, 13, 14)$

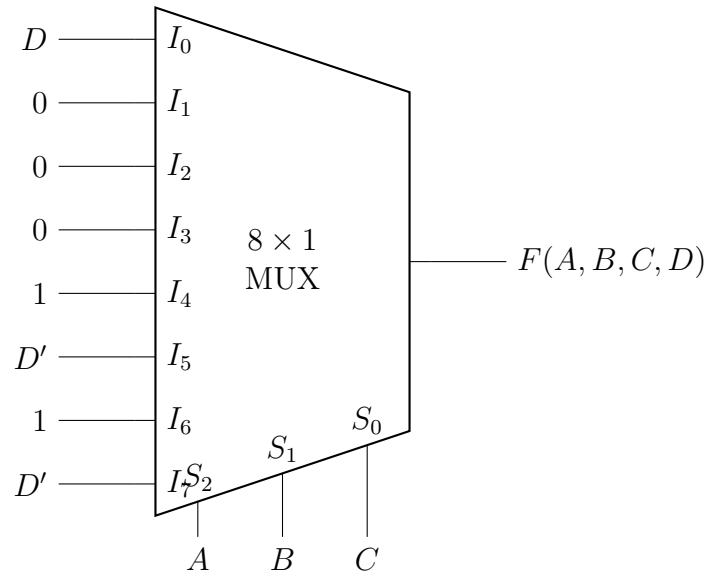


e)

Using the truth table in question 6a, we can modify the truth table for an 8-to-1 multiplexer:

$S_2(A)$	$S_1(B)$	$S_0(C)$	$F(A, B, C, D)$
0	0	0	D
0	0	1	0
0	1	0	0/1
0	1	1	0
1	0	0	$D/1$
1	0	1	D'
1	1	0	1
1	1	1	D'

Note: Different choices appear because of don't care conditions.
Therefore, the block diagram is:



f)

Using the truth table in question 6a, we can modify the truth table for a 4-to-1 multiplexer:

$S_1(B)$	$S_0(C)$	$F(A, B, C, D)$
0	0	D
0	1	AD'
1	0	$A/1$
1	1	AD'

Note: Different choices appear because of don't care conditions.

