Assignment#2 CS207 Fall 2023

Ben Chen(12212231)

November 8, 2023

PROBLEM 1. Obtain the simplified boolean expressions for the output F and G in terms of the input variables in the given circuit.

SOLUTION. **a)** Considering the dependency of each function, we obtain T2 and T3 first, then T1 and finally F and G.

$$
\begin{aligned}
T2(A,B,C,D) &= (A'D)' \\
T3(A,B,C,D) &= A' + BC \\
T1(A,B,C,D) &= (A' \cdot T2)' = (A'(A'D)')' \\
F(A,B,C,D) &= T1 \cdot T3 = (A' + BC)((A'(A'D)')') \\
G(A,B,C,D) &= (T2 \cdot T3)' = ((A' + BC)(A'D)')'
\end{aligned}
$$

b) Simplify the outputs of F and G

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

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

c) The truth table of F and G

A	B	C	D	$F(A,B,C,D)$	$G(A,B,C,D)$
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

PROBLEM 2. A circuit with four inputs $A_3A_2A_1A_0$ and two output P and D, where the former one is TRUE when the number is prime number and the lattar one is TRUE if the number is a multiple of 3.

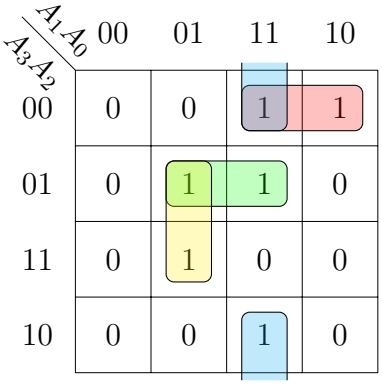
SOLUTION. **a)** It's obvious that

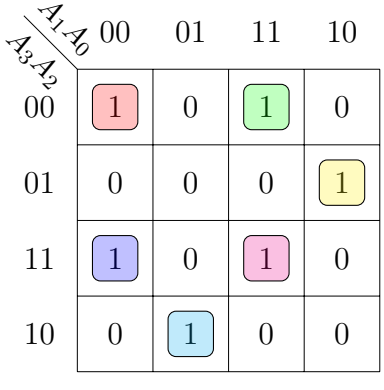
$$P = \Sigma(2, 3, 5, 7, 11, 13)$$
$$D = \Sigma(0, 3, 6, 9, 12, 15)$$

so the truth table of the circuit is

A_3	A_2	A_1	A_0	P	D
0	0	0	0	0	0
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) The K-map of P and D are



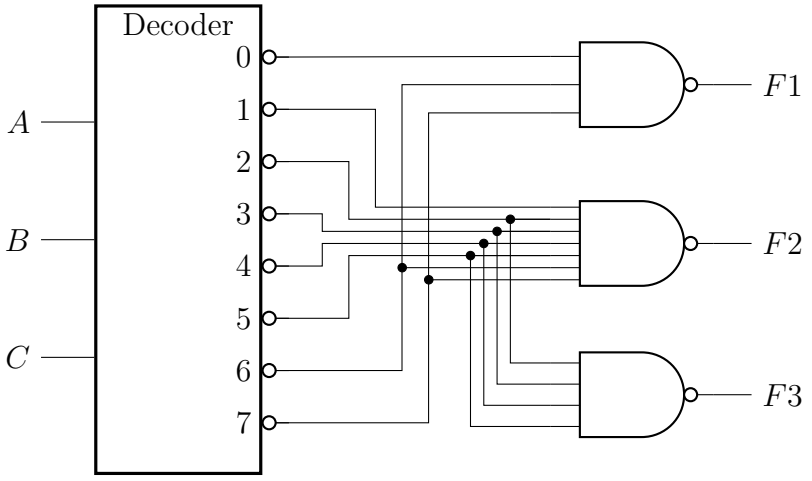


PROBLEM 3. Design a circuit to implement the three functions with decoder and NAND gates, and draw the diagram.

SOLUTION. Convert the equations into sum of minterms

$$F1 = \Sigma(0, 6, 7)$$
$$F2 = \Sigma(1, 2, 3, 4, 5, 6, 7)$$
$$F3 = \Sigma(2, 3, 4, 5)$$

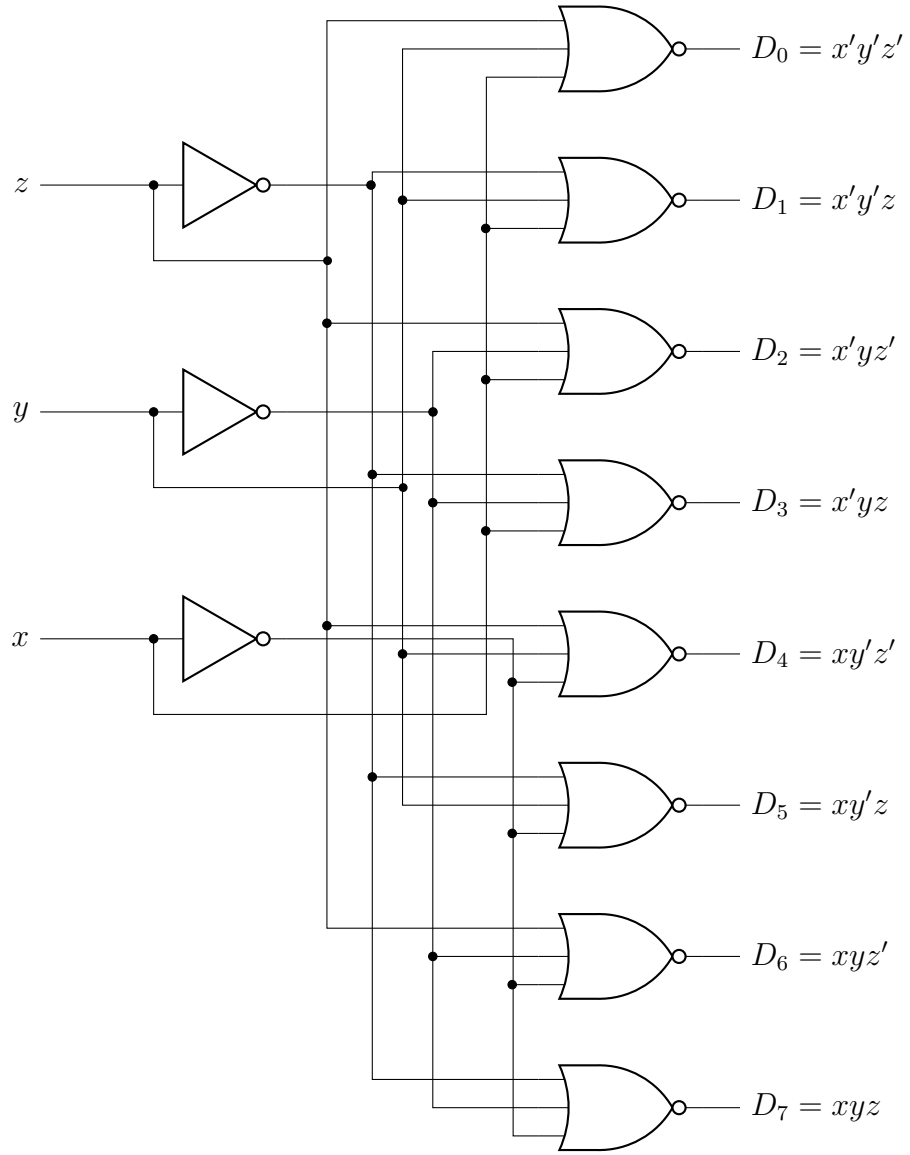
and then design the circuit



PROBLEM 4. Develop a 3-to-8 line decoder using NOR gates only, and draw its logic diagram.

SOLUTION. To implement a 3-to-8 decoder with nor gates only, we can come up with a method similar to the method using AND gates only.
For example,

$$(x' + y' + z')' = xyz$$
$$(x + y + z)' = x'y'z' = 0$$

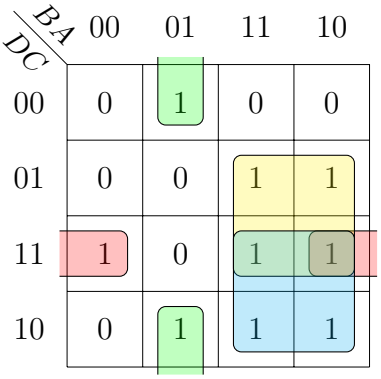


PROBLEM 5. An 8:1 multiplexer has inputs A, B, and C connected to the selection inputs S_2, S_1, and S_0, respectively. The data inputs I_0 through I_7 are as follows: $I_1 = I_2 = 0$; $I_3 = I_5 = I_7 = 1$; $I_0 = I_4 = D$; and $I_6 = D'$. Determine the Boolean function $F(A, B, C, D)$ that the multiplexer implements.

SOLUTION. **a)** The truth table of F is

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

b) And the K-map is



Thus, we got the sum of products from the K-map

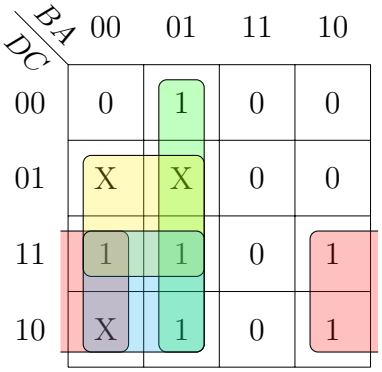
$$F(A, B, C, D) = BC + BD + AB'C' + A'CD$$

PROBLEM 6. Implement the boolean function $F(A, B, C, D) = \Sigma(1, 9, 10, 12, 13, 14) + d(4, 5, 8)$ using the following component.

SOLUTION. **a)** The truth table is shown below, which is obtained from the midterms and the don't care terms are denoted by X.

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

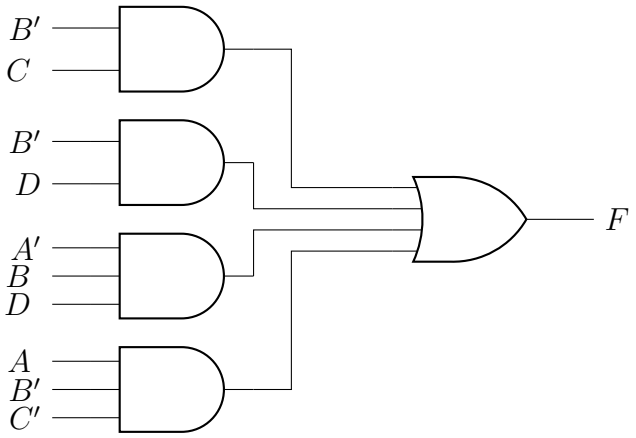
b) Simplify the boolean function using K-map first



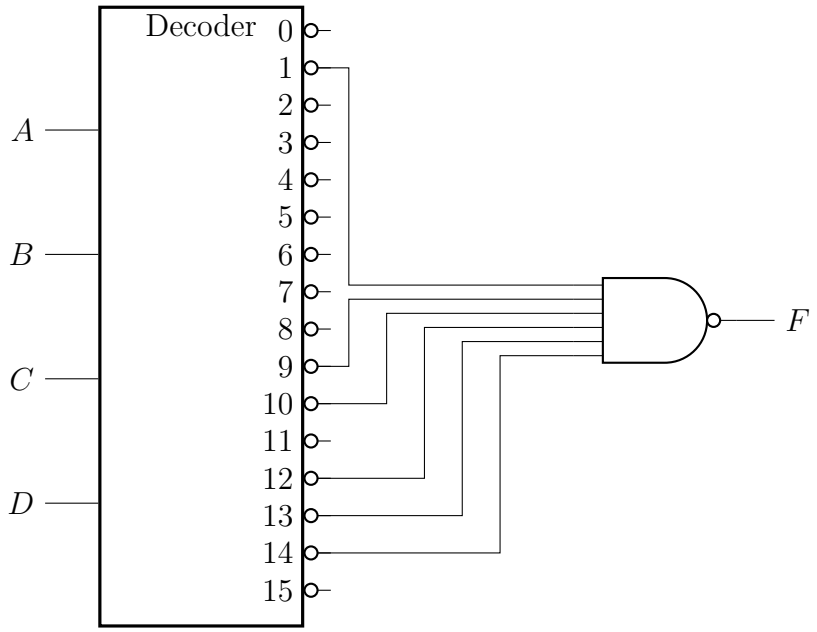
we got the terms

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

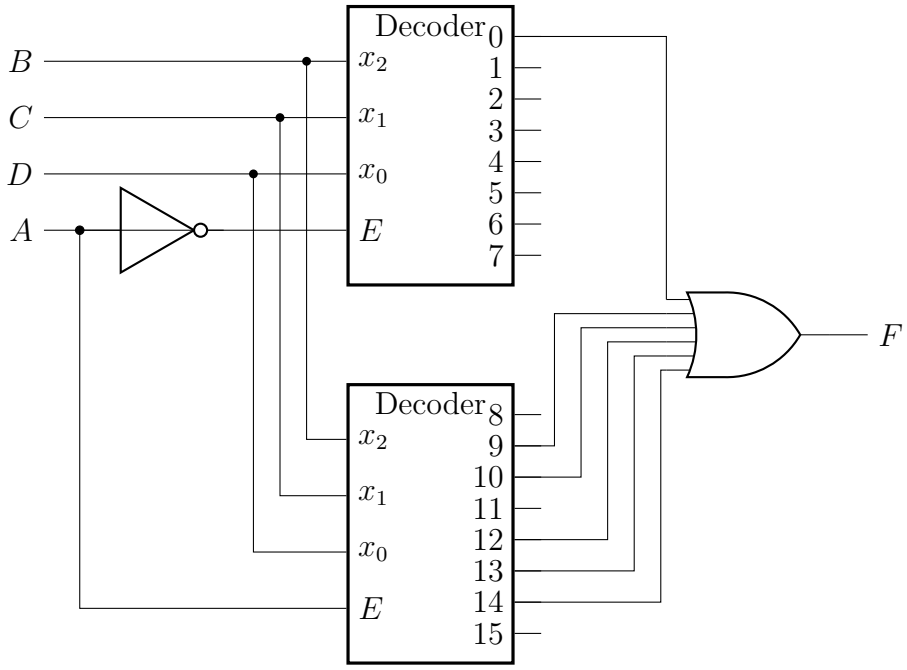
and the logic diagram using AND OR gates



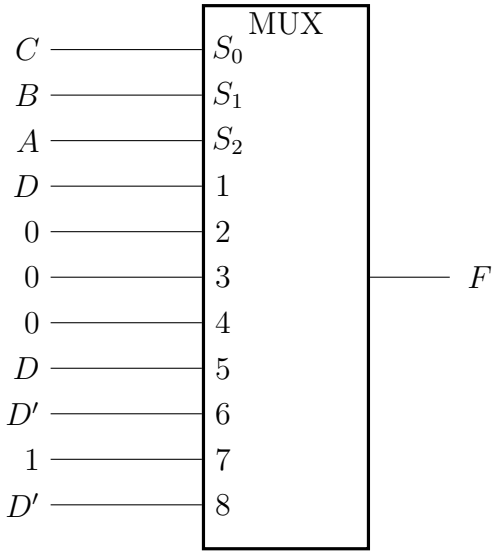
c) The implementation with 74157 decoder and NAND gate is



d) The diagram uses one decoder to decode $D_0 - D_7$ and the other to decode $D_8 - D_{15}$. And then connect the output of minterms with a OR gate.



e) The 8-to-1 multiplexer uses A, B, C as its selector and the input is determined by D or $0/1$ according to the truth table.



f) Design the 4-to-1 multiplexer with B, C with its selector from the (a) truth table. So we got when $BC = 00$ inputs $= A + D$, when $BC = 01$ inputs $= AD'$, when $BC = 10$ inputs $= 1$ and when $BC = 11$ inputs $= AD'$.