CS 211 Computer Architecture: Logic Design Problem Set
Instructor: Prof. Santosh Nagarakatte

Problem 1:
Simplify the following equations using Boolean theorems. Check the correctness using a truth table and a K-Map. (3 points each for the Boolean simplifications, 1 point for the truth table, 3 points for the K-map, 3 point for the resultant circuit)

1. $Y = AC + \bar{A}\bar{B}C$
2. $Y = \bar{A}\bar{B} + \bar{A}B\bar{C} + \overline{(A + \bar{C})}$

Problem 2:
Write the sum of products canonical representation (minterms) for each of the following truth tables. Y is the output of the circuits.

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Problem 3:
Minimize each of the circuits in Problem 2 and implement them using a simple combinational circuit using two-input AND and OR gates, and/or a NOT gate.

Problem 4:
Design the simplest sum of products circuit that implements: $f(x_1, x_2, x_3) = \sum M(0,1,2,3,4,6,7)$. Use Boolean algebra to simplify the function and confirm your solution using K-Maps.

Problem 5:
Consider the following truth table:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0

0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Show its sum of products (SoP) and product of sums (PoS). Also, use Boolean algebra to find the minimum-cost SoP form.

Consider the following truth table:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	×
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	×
1	1	0	0
1	1	0	0

Use K-Maps to design the simplest sum of products circuit.

Problem 6:

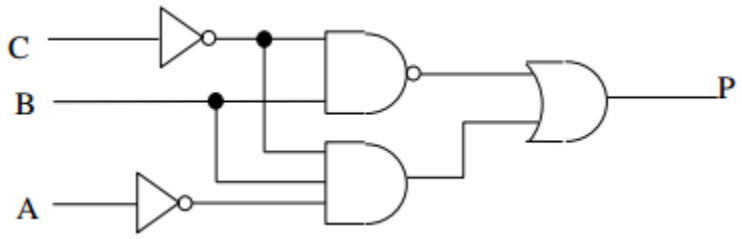
Find a minimal Boolean Equation for the truth table given below. Remember to take advantage of the don't care entries. Draw the resultant circuit.

A	B	C	D	Y
0	0	0	0	×
0	0	0	1	×
0	0	1	0	×
0	0	1	1	0
0	1	0	0	0
0	1	0	1	×
0	1	1	0	0
0	1	1	1	×
1	0	0	0	1
1	0	0	1	0
1	0	1	0	×
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1

1	1	1	0	×
1	1	1	1	1

Problem 7:

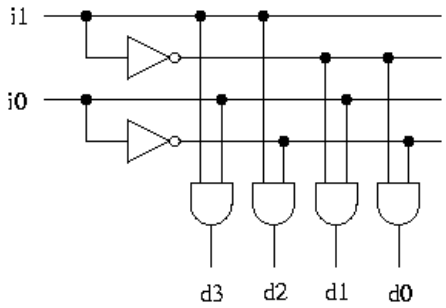
Consider the following circuit. What is its truth table?



What is the minimal-cost expression for this true table? Find the simplest SoP and confirm your solution using K-Maps.

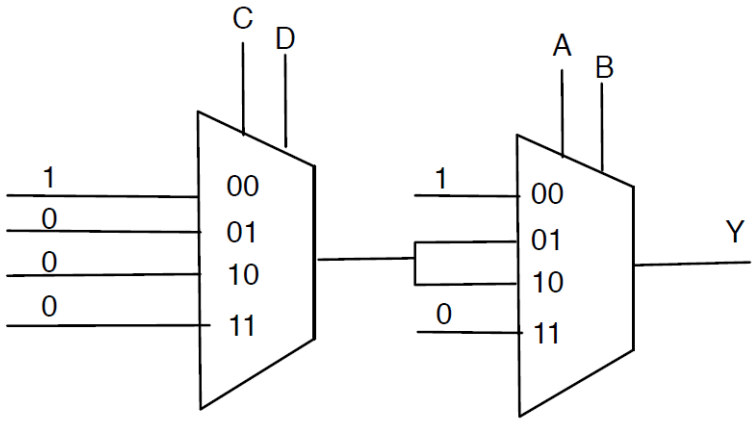
Problem 8:

The following circuit (with ***NOT*** and 2-input ***AND*** gates) implements the 2-4 decoder.



Design a 3-8 decoder circuit with ***NOT*** and 2-input ***AND*** gates. How many gates (***NOT*** and 2-input ***AND*** gates) are needed to build the n-2^n decoder?

Problem 9:
Give the Boolean expression for the function performed by the following circuit:

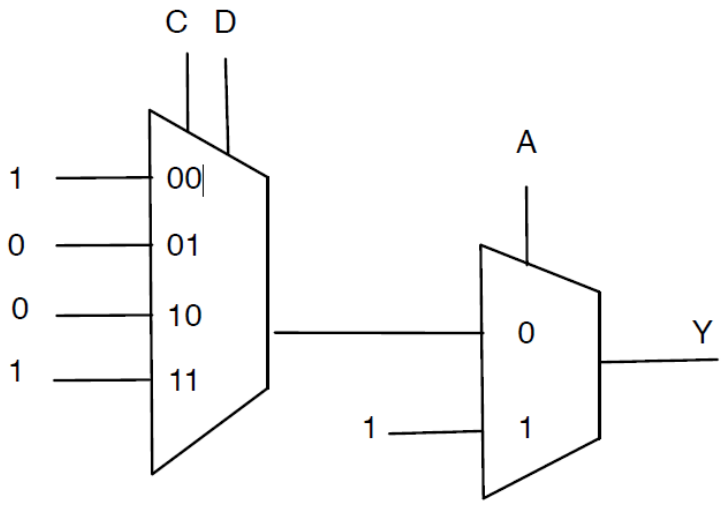


Problem 10:
Implement the following truth table as described below:

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

1. Using a 3:8 decoder and one other logic gate
2. Using a 8:1 multiplexer
3. Using a 4:1 multiplexer and one inverter
4. A 2:1 multiplexer and two other logic gates

Problem 11:
Identify the Boolean equation performed by the circuit below and minimize the circuit and implement it using simple gates.



Problem 12:
Design a 4-to-16 decoder using five 2-to-4 decoder

Problem 13:
Design a 32-to1 multiplexer (MUX) using
1. 8-to-1 MUX and 2-to-4 decoders.
2. 4-to-1 MUX and 2-to-4 decoders.

Problem 14:
Given a 3-input Boolean expression $f(x_1, x_2, x_3) = \sum M(0,2,4,6,7)$.
1. Implement this expression using only 2-4 decoders and ***OR*** gates.
2. Implement this expression using only 4-1 MUX.

Problem 15:
Implement the function $f(x_1, x_2, x_3, x_4, x_5) = \bar{x}_1\bar{x}_2\bar{x}_4\bar{x}_5 + x_1x_2 + x_1x_3 + x_1x_4 + x_3x_4x_5$ by using a 4-to-1 multiplexer and as few other gates as possible.

Problem 16:

Consider the Boolean function $f(x_1, x_2, x_3) = x_1\bar{x}_2 + \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3$.

1. Use a 3-to- 8 decoder plus logic gates to implement this function.
2. Use an 8-input multiplexer to implement this function.

Problem 17:

Gray codes have a useful property in that consecutive numbers differ in only a single bit position. The 3-bit gray code is given below.

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Design a 3-bit modulo 8 Gray code counter FSM that has no inputs but produces three outputs. When reset the output should be 000. On each clock edge the output should advance to the next Gray code. After reaching 100, it should repeat with 000. Implement the circuit using combinational logic and D-flip flops.

Problem 18:

Design an FSM that detects a stream of two consecutive 1's in a stream of 0's and 1's. Implement it using a combination of sequential and combinational logic.

INPUT: 0 1 1 0 1 1 1 0 1 0 ....

OUTPUT: 0 0 1 0 0 1 1 0 0 0 ....