CSE 140 Homework Two

January 30, 2019

*Only **Problem Set Part B** will be graded. Turn in only **Problem Set Part B** which will be due on February 12, 2019 (Tuesday) at 3:00pm.*

*Submit homework via gradescope <gradescope.com> as a **PDF.** Other formats may or may not be viewed and evaluated accurately.*

1 Problem Set Part A

All questions in this part are from Roth&Kinney, 7th Edition.

- 4.5, 4.6, 4.8, 4.10, 4.13, 4.15, 4.17, 4.26, 4.29, 4.32
- 5.5, 5.8, 5.10, 5.11, 5.12, 5.13, 5.20, 5.28, 5.30, 5.32, 5.33, 5.34, 5.42
- 6.6, 6.9, 6.11, 6.16, 6.17, 6.24

2 Problem Set Part B

I. (K-map Mystery)

A **minimum sum-of-products** expression for a function is defined as a **sum-of-products (SoP)** expression which *(a)* has a **minimum number of terms** and *(b)* of all those expressions which have the same minimum number of terms, has a **minimum number of literals**. A **minimum product-of-sums (PoS)** expression is defined similarly. As a simple example, the minimum SoP expression of the following function contains only the prime implicant ***y*** rather than ***x'z'*** since ***y*** leads to the minimum number of literals.

z \ xy	00	01	11	10
0	X	1	X	0
1	0	X	X	0

A Karnaugh map is a systematic method for simplifying boolean functions. When you are practicing function simplification using K-maps, you realized that you misplaced the paper containing the original K-maps after determining minimum expressions of a few functions. You recalled that all the K-maps contained **Don't Cares**, so the minimum expressions that you still have may fail to recover the K-maps that you worked on. As you are suffering from anxiety, you realize that you can still trace a lot of information from the expressions, and hopefully recover the original K-Map with a bit of jolting of your memory.

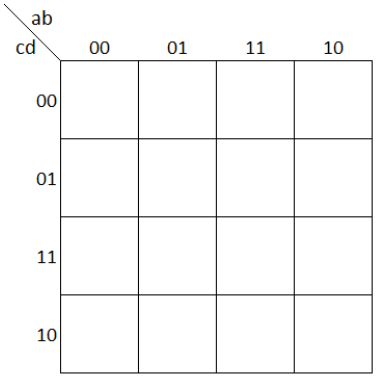
(Part A) The following minimum sum-of-products expression contains **4 essential prime implicants**. **Circle the implicants on the map and determine the minterms that make each implicant essential**.

$$f(a,\ b,\ c,\ d) = a'c' + b'd + acd + ab'c$$

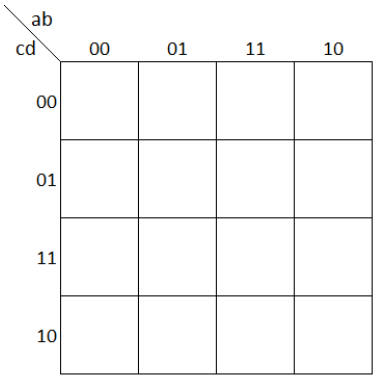
cd \ ab	00	01	11	10
00				
01				
11				
10				

(Part B) The following minimum product-of-sums expression contains **4 prime implicates**. **Circle the implicates on the map and identify all the implicates that are *necessarily* essential**.

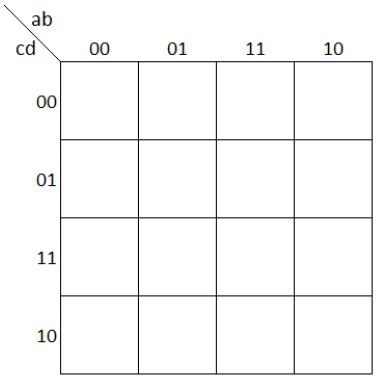
$$g(a, b, c, d) = (a' + c)(b' + d)(a + b' + c')(a + c' + d)$$



(Part C) Fill in the map with the **maximum possible number of Don't Cares** so that the expression ***g*** in (Part B) is one of its minimum PoS expressions.



(Part D) At this point, you should have noticed that we cannot recover the exact form of the original K-map from a minimum expression due to the ***Don't Cares***. It is interesting to ponder what may happen if you manage to jolt your memory to recollect a bit more information about the K-map. You suddenly remember that the **minimum expressions in (Part A) and (Part B) have been derived from the same K-map**, and that there are **exactly 2 Don't Cares** in the original K-map. **Please recover the exact form of this K-map**.



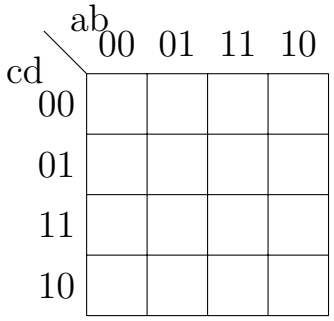
(Part E) Do you think that we can recover any K-map given **a minimum SoP expression, a minimum PoS expression, and the exact number of Don't Cares**? If yes, show an example in a 3-variable K-map. If not, explain the reason for the ambiguity and why we were able to recover the K-Map in (Part D) based on the interpretation of the *Don't Cares*.

II. (Quine McCluskey)

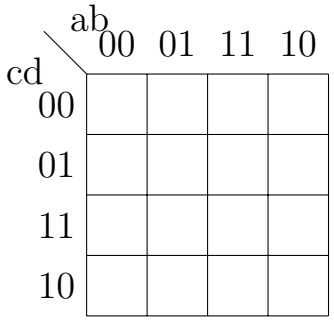
Determine whether the following prime implicant tables and Petrick's Method equations are valid in each of the scenarios given below. If so, provide a Karnaugh map. If not, explain. All valid solutions are representable in 4-dimension Karnaugh maps although you are free to use more dimensions in your solution.

(Part A) Consider the following prime implicant tables as having just been initialized without simplification. For this part, assume there are **no** *don't cares* in your boolean function.

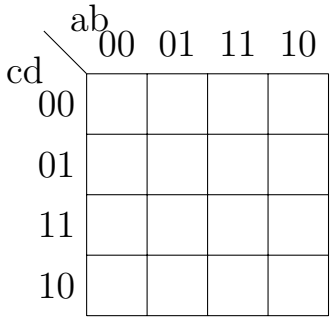
	m_a	m_b	m_c
PI_1	x		x
PI_2		x	x
PI_3	x		



	m_a	m_b	m_c	m_d
PI_1	x	x		
PI_2		x	x	
PI_3			x	x
PI_4	x			x

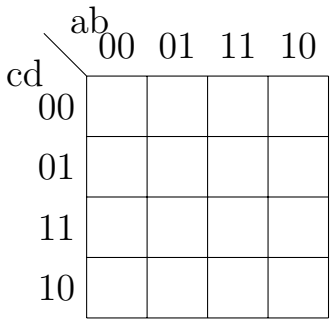


	m_a	m_b	m_c	m_d	m_e	m_f	m_g	m_h
PI_1	x	x	x	x				
PI_2			x	x	x	x		
PI_3					x	x	x	x

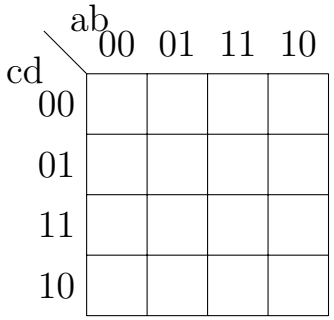


(Part B) Like the last problem, consider the following prime implicant tables as having just been initialized. This time, however, assume that *don't cares* **are possible** in your boolean function.

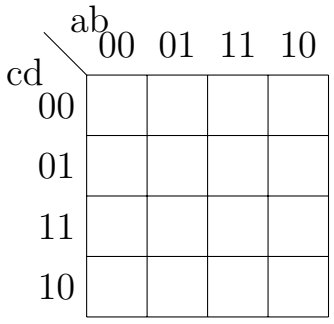
	m_a	m_b	m_c	m_d	m_e
PI_1	x		x		
PI_2		x	x	x	x
PI_3					



	m_a	m_b	m_c	m_d	m_e
PI_1	x	x	x	x	
PI_2		x	x	x	x

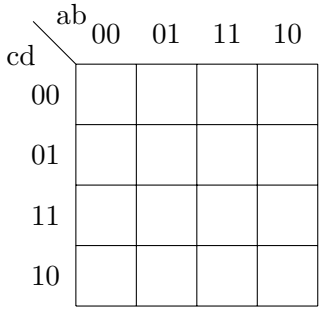


	m_a	m_b	m_c	m_d	m_e
PI_1	x	x			
PI_2		x	x		
PI_3			x	x	
PI_4				x	x
PI_5	x				x

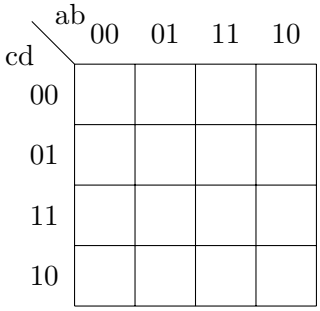


(Part C) Now we would like you to do the same thing with the following Petrick's Method equations. Consider for these equations that Quine-McCluskey was just run to prune out implicants and minterms such that only the remaining cyclic core was the input to Petrick's Method. Assume that the following Petrick's Method equations have already been simplified (by using distributivity, idempotency, absorption, and other Boolean theorems).

- $P_1P_3P_5 + P_2P_4P_6$

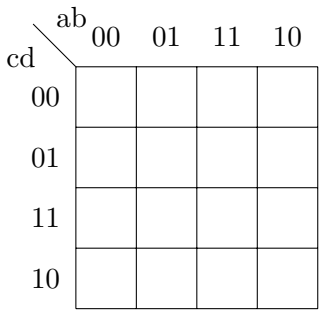


- $P_3P_2 + P_3P_1$

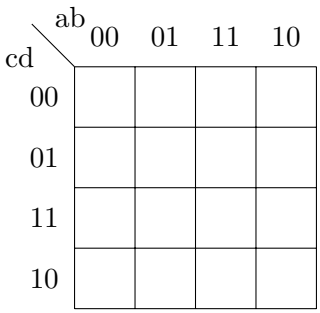


(Part D) Similar to what you did for (Part C), we would like you to again consider Petrick's Method equations but this time on a set of prime implicants that have **not** been **pruned** by Quine-McCluskey.

- $P_1P_2 + P_3P_4$



- $P_1P_2P_3 + P_1P_2P_4$



III. (Map Entered Variables)

In class you learned to turn a Karnaugh map with Map-Entered Variables (MEV) into a sequence of Karnaugh maps in order to find the minimum **Sum-of-Products (SoP) expression**. You were impressed by the representational power of the method, while still curious about how the process might change if you were to find the **minimum Product-of-Sums (PoS) expression**.

To help you solve this puzzle, we will walk you through a number of exercises. As you solve each part, keep in mind that you may **not** need to use all the empty K-maps provided. When you are filling in K-maps, make sure to fill in each and every entry with 0, 1, and X (Don't Care).

(Part A) Before diving right in, you choose to refresh your knowledge of finding the minimum SoP expression with 2 MEVs, E and F. **Please describe how you fill in the intermediate K-maps in the following cases.**

How to handle the case with no E or F:

How to handle the case when E = 1:

How to handle the case when F = 1:

Please apply your method to find the minimum SoP expression of the following function. Fill in the K-maps for certain variable assignments and write down the minimum SoP expression of **each** K-map.

wx\yz	00	01	11	10
00	**1**	0	0	0
01	**1**	**1**	**F**	**E**
11	**E**	**1**	**F**	**E**
10	**E**	0	0	0

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **SoP =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **SoP =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **SoP =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **SoP =**

Write the Boolean expression for the overall function in minimum SoP form:

(Part B) Now that you have reminded yourself how to find the minimum SoP expression, you think you are ready to tackle the minimum PoS expression. **Please define the cases you need to handle to find the minimum PoS expression with 2 MEVs, *E* and *F*. Also, briefly describe how you fill in the intermediate K-map for each case.**

Hint: think about how the overall PoS expression can be constructed with MEVs and intermediate PoS expressions.

Please apply your method to find the minimum PoS expression of the following function. Fill in the K-maps for certain variable assignments and write down the minimum PoS expression of **each** K-map.

wx\yz	00	01	11	10
00	**1**	0	0	0
01	**1**	**1**	**F**	**E**
11	**E**	**1**	**F**	**E**
10	**E**	0	0	0

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **PoS =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **PoS =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **PoS =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **F =** **PoS =**

Write the Boolean expression for the overall function in minimum PoS form:

____

(Part C) Inspired by your success in the last two parts, you decide to examine what happens when you include a variable's complement in the K-map. **Please define the cases you need to handle to find the minimum SoP expression with 2 MEVs, *E* and *E'*. Also, briefly describe how you fill in the intermediate K-map for each case.**

Hint: The order in which different cases are handled DOES matter.

Please apply your method to find the minimum SoP expression of the following function. Fill in the K-maps for certain variable assignments and write down the minimum SoP expression of **each** K-map.

wx\yz	00	01	11	10
00	**E**	0	0	0
01	**1**	**1**	**1**	**E′**
11	**1**	**1**	**1**	**E′**
10	**E**	0	0	0

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **SoP =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **SoP =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **SoP =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **SoP =**

Write the Boolean expression for the overall function in minimum SoP form:

(Part D) Now that you see how complemented variables fit in to MEVs, you realize that you can easily merge your discovery about complements with your findings about minimum PoS expressions. As such, you decide to find the minimum PoS expression with 2 MEVs, E and E'. **Please define the cases you need to handle to find the minimum PoS expression with complemented variables. Also, please describe how you fill in the intermediate K-map for each case.**

Please apply your method to find the minimum PoS expression of the following function. Fill in the K-maps for certain variable assignments and write down the minimum PoS expression of **each** K-map.

wx\yz	00	01	11	10
00	**E**	0	0	0
01	**1**	**1**	**1**	**E′**
11	**1**	**1**	**1**	**E′**
10	**E**	0	0	0

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **PoS =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **PoS =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **PoS =**

wx\yz	00	01	11	10
00				
01				
11				
10				

E = **E′ =** **PoS =**

Write the Boolean expression for the overall function in minimum PoS form:

____