CS M51a / EE M16
Fall 2018
Midterm

Name: Yu-Hsuan Liu

UID: 705 182 685

Discussion Section (circle): (1A) 1B 1C 1D 1E

Problem 1: 20 /20

Problem 2: 20 /20

Problem 3: 16 /20

Problem 4: 19 /20

Problem 5: 20 /20

Total: 95 /100

Read each question carefully before you answer. You have 1 hour and 50 minutes for the exam, which is on average 22 minutes per problem. If you find yourself spending more than 22 minutes on any problem, move on to the next problem and come back to it later. Don't spend the majority of your time on any one problem if there are still other problems to solve, or you may risk running out of time before finishing the exam.

Make sure to show all your work in the space provided on the exam. Points will be taken off for answers without any work. Partial credit will be given where substantial work has been done.

Problem 1 (20 points)

Suppose we want to design a digital clock that displays the current time in hours and minutes as $H_1H_2 : M_1M_2$, such as, for example 12:30, which would set $H_1=1$, $H_2=2$, $M_1=3$, and $M_2=0$.

We will use the standard 12-hour format, so 04:00 can either represent 4am or 4pm (as opposed to the 24-hour military format such as 1600 for 4:00pm).

a) Determine for each digit (H_1, H_2, M_1, M_2) of the display:
   i. The range of each digit (i.e. what is the smallest value and the largest value, in decimal)
   ii. The minimum number of bits necessary to represent each digit in a standard binary format (not BCD)

i) $H_1 : 0 \sim 1$
$H_2 : 0 \sim 9$
$M_1 : 0 \sim 5$
$M_2 : 0 \sim 9$

ii) $H_1 : \lceil \log_2 1 \rceil = 1$
$H_2 : \lceil \log_2 9 \rceil = 4$
$M_1 : \lceil \log_2 5 \rceil = 3$
$M_2 : \lceil \log_2 9 \rceil = 4$

$1 + 4 + 3 + 4 = 12$ (bits)

b) Is the <u>total</u> number of bits for representation the minimum, or is there a more efficient (less number of bits) way to represent the time as a single component (which may need some digit conversion to display the time properly as $H_1H_2 : M_1M_2$)

If there is **not** a more efficient way to represent the time, explain why not.

If there **is** a more efficient way, explain why, and what is the minimum **total** number of bits to represent the time in hours and minutes.

$\underset{\text{minutes in an hour}}{60} \times \underset{12\ hrs}{12} = 720$

$\lceil \log_2 720 \rceil = 10$ bits

Yes.
If we represent the time as a single component, we just need 10 bits. However, this representati[on] is not straight forward. We need more work to convert the code in order to display the time properly. For example, $0001000000_{(2)}$ is $64_{(10)}$
$64 \div 60 = 1 \ldots 4$, so it is 01:04.

(20)

Problem 2 (20 points)

Prove or disprove the following equalities, by either constructing the corresponding tables, or using the laws of Boolean Algebra (you do not have to state the name of the law in each step). Note: ⊕ represents the XOR function, and ⊙ represents the XNOR function, + represents the OR function, and • represent the AND function.

a)

$$((w \oplus x)\cdot(x \oplus y)\cdot(y \oplus z))' = (w \odot x) + (x \odot y) + (y \odot z)$$

w	x	y	z	a: w⊕x	b: x⊕y	c: y⊕z	abc	LHS: abc'	d: w⊙x	e: x⊙y	y⊙z	RHS: d+e+f
0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	1	0	0	1	0	1	1	1	0	1
0	0	1	0	0	1	1	0	1	1	0	0	1
0	0	1	1	0	1	0	0	1	1	0	1	1
0	1	0	0	1	1	0	0	1	0	0	1	1
0	1	0	1	1	1	1	1	0	0	0	0	0
0	1	1	0	1	0	1	0	1	0	1	0	1
0	1	1	1	1	0	0	0	1	0	1	1	1
1	0	0	0	1	0	0	0	1	0	1	1	1
1	0	0	1	1	0	1	0	1	0	1	0	1
1	0	1	0	1	1	1	1	0	0	0	0	0
1	0	1	1	1	1	0	0	1	0	0	1	1
1	1	0	0	0	1	0	0	1	1	0	1	1
1	1	0	1	0	1	1	0	1	1	0	0	1
1	1	1	0	0	0	1	0	1	1	1	0	1
1	1	1	1	0	0	0	0	1	1	1	1	1

↑ have the same output ↑

$\Rightarrow ((w\oplus x)\cdot(x\oplus y)\cdot(y\oplus z))' = (w\odot x)+(x\odot y)+(y\odot z)$

b)

$$(((w'+x+y)' + (w+x'+y)' + (w+x+y')' + (w'+x'+y')')')' = w' \oplus x \oplus y'$$

$LHS = ((wx'y' + w'xy' + w'x'y + wxy)')'$

$= wx'y' + w'xy' + w'x'y + wxy$

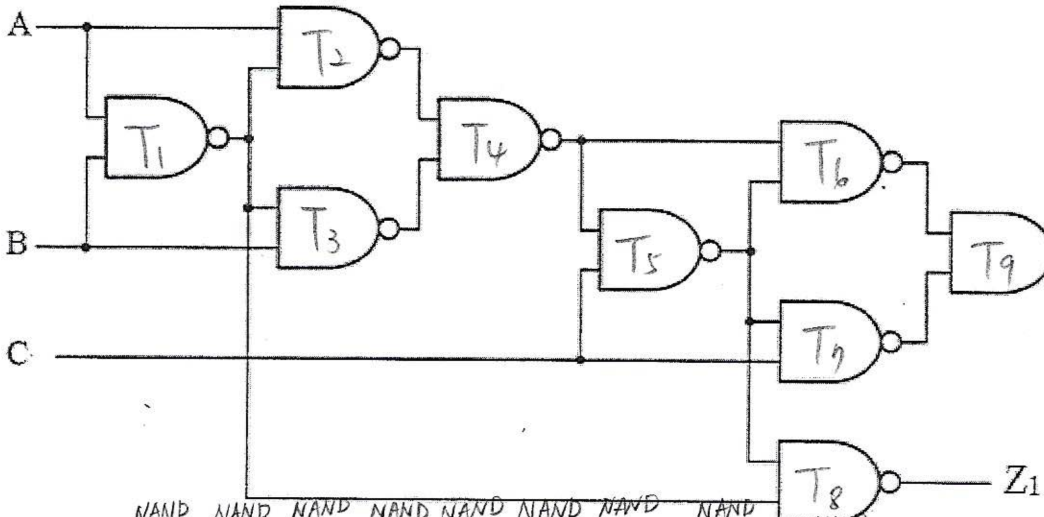
w	x	y	w'	x'	y'	a: wx'y'	b: w'xy'	c: w'x'y	d: wxy	a+b+c+d	e: w'⊕x	e⊕y'
0	0	0	1	1	1	0	0	0	0	0	1	0
0	0	1	1	1	0	0	0	1	0	1	1	1
0	1	0	1	0	1	0	1	0	0	1	0	1
0	1	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	1	0	1
1	0	1	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	1	1	1	1

↑ have the same output ↑

$\Rightarrow (((w'+x+y)' + (w+x'+y)' + (w+x+y')' + (w'+x'+y')')')' = w' \oplus x \oplus y'$

Problem 3 (20 points)

a) Analyze the following network by determining the expressions for the outputs (Z_1, Z_0) based on the inputs (A, B, C).
{Hint: label each gate as a sub-function, T_i, and using a table, determine the result of each sub-function based on its inputs, until you reach the values for Z_1 and Z_0}
DO NOT CALCULATE THE CRITICAL DELAY OF THE NETWORK.



	ABC	T_1 NAND (A,B)	T_2 NAND (A,T1)	T_3 NAND (T1,B)	T_4 NAND (T2,T3)	T_5 NAND (T4,C)	T_6 NAND (T4,T5)	T_7 NAND (T5,C)	$T_8(Z_1)$ NAND (T5,T1)	$T_9(Z_0)$ NAND (T6,T7)	
0	000	1	1	1	0	1	1	1	0	0	0
1	001	1	1	1	0	1	1	0	0	1	
2	010	1	1	0	1	1	0	1	0	1	1
3	011	1	1	0	1	0	1	1	1	0	2
4	100	1	0	1	1	1	0	1	0	1	1
5	101	1	0	1	1	0	1	1	1	0	2
6	110	0	1	1	0	1	1	1	1	0	2
7	111	0	1	1	0	1	1	0	1	1	3

$z_1 = \Sigma m(3,5,6,7)$
$= A'BC + AB'C + ABC' + ABC$

$z_0 = \Sigma m(1,2,4,7)$
$= A'B'C + A'BC' + AB'C' + AB$

✓

b) From a high-level description, what <u>mathematical</u> function is the network performing? Do <u>not</u> state your answer simply as when the output equals 1 such as: "*z_1 =1 when A=... and B=... and C=... or when A=... and B=... and C=...*", but give a clear descriptive answer.

Z = { 0, when A, B, C all do not have value of 1
1, when one out of A, B, C has value of 1
2, when two out of A, B, C has value of 1
3, when A, B, C all have value of 1 }

Z will show how many inputs have the value of 1.

1 2 4 8
-0 -1 -2 -3
5 4 3 2

Problem 4 (20 points)

a) Using the 5-input K-map below, find all the prime implicants of the function:
$f(x_4, x_3, x_2, x_1, x_0) = \Sigma m(7,11,13,14,15,19,21,22,23,25,26,27,28,29,30,31)$

$x_4 = 0$ (columns x_1x_0, with x_0 over the middle two columns and x_1 over the right two; rows x_3x_2, with x_2 beside the middle two rows and x_3 beside the bottom two)

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

$x_4 = 1$ (same layout)

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

List of Prime implicants:

$x_3x_2x_0,\ x_2x_1x_0,\ x_3x_2x_1,\ x_3x_1x_0,\ x_4x_3x_2,\ x_4x_1x_0,$
$x_4x_2x_0,\ x_4x_2x_1,\ x_4x_3x_0,\ x_4x_3x_1$

b) Which of the prime implicants from part a) are essential?

They are all essential.

c) Obtain a minimal sum of products for *f*.

$f(x_4, x_3, x_2, x_1, x_0) = x_4x_3x_2 + x_4x_3x_1 + x_4x_3x_0 + x_4x_2x_1 + x_4x_2x_0$
$x_4x_1x_0 + x_3x_2x_1 + x_3x_2x_0 + x_3x_1x_0 + x_2x_1x_0$

d) Describe at a high-level what the function *f* is determining, i.e. when is *f*=1? Do not merely restate the problem by saying *f*=1 when *x*=7 or *x*=11 or *x*=13,… but give a clear explanation. {Hint: It is a not a complex function, but a common function used when comparing bits}

When three out of five inputs have the value of 1, f=1.

majority

Problem 5 (20 points)

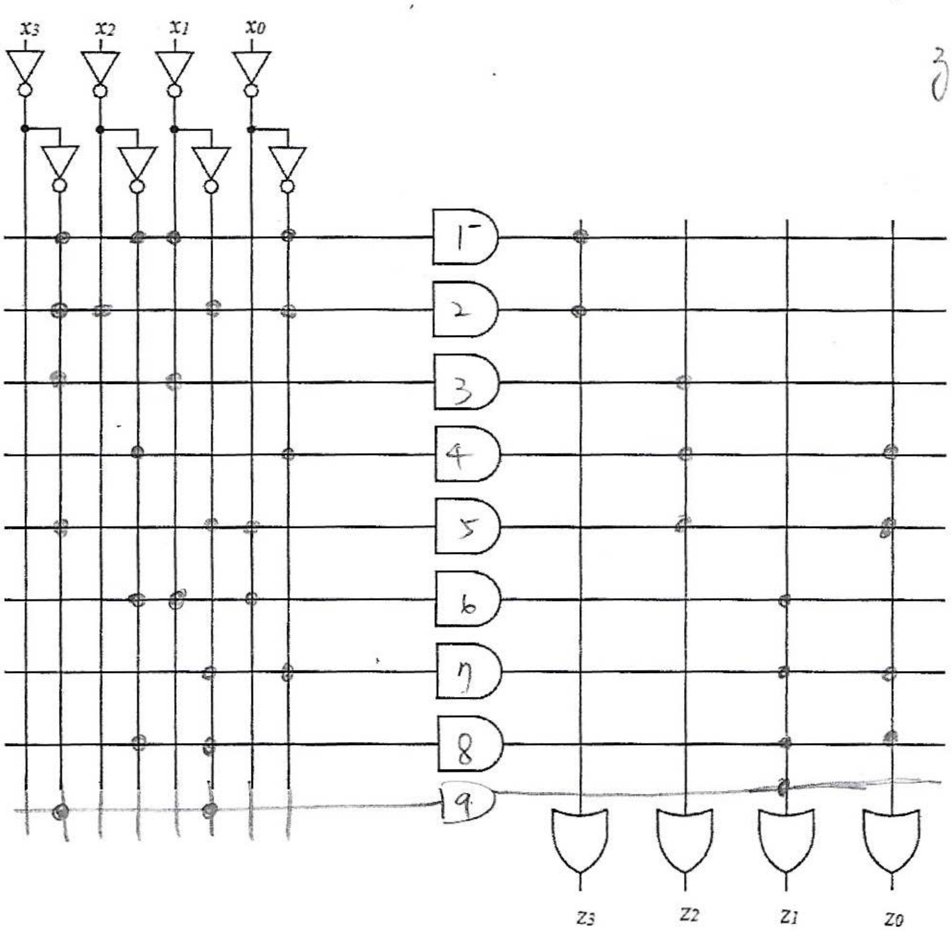
Using only the PLA below, without adding any rows of extra AND gates, implement a system that represents the following function $z = (z_3, z_2, z_1, z_0)$ given the input $x = (x_3, x_2, x_1, x_0)$, where $0 \leq x \leq 15$, and both x and z are represented in standard binary representation.

$$z = \begin{cases} x & \text{if } x \text{ is a prime number} \\ \lfloor x/2 \rfloor & \text{otherwise} \end{cases}$$

Be sure to label every connection made in the PLA with a dot: • at the appropriate intersection of a horizontal and a vertical line. Do not label unused connections with ×

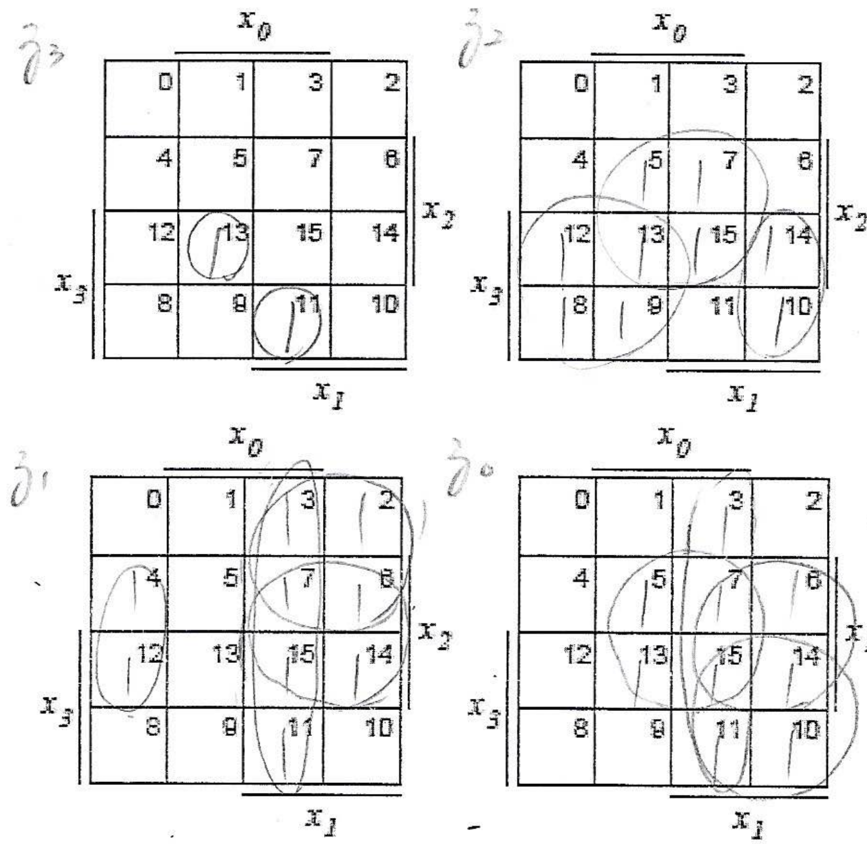
{Note: x=0 and x=1 are not prime numbers, and a prime number is only divisible by itself and 1. The notation $\lfloor x/2 \rfloor$ represents the "floor" function of taking the integer part of diving by 2).

{Hint: Create a minimal sum-of-products for each output bit of z using either K-maps or laws of Boolean Algebra, and reuse prime implicants for multiple output bits of z}



0 1 2
1 2 4

{Space for K-maps to show work for Problem 5, if necessary—the K-maps are optional and will not be graded—they are only for your convenience}



$z_3 = x_3 x_2 x_1' x_0 + x_3 x_2' x_1 x_0$

$z_2 = x_3 x_1' + x_0 x_2 + x_1 x_3 \ldots$

$z_1 = x_2 x_1' x_0' + x_1 x_0 + x_3' x_1 + x_2 x_1$

$z_0 = x_2 x_0 + x_1 x_0 + x_2 x_1 + x_3 x_1 x_0'$

{Space for table to show work for Problem 5, if necessary—the table is optional and will not be graded—it is only for your convenience}

	x_3	x_2	x_1	x_0	z_3 (8)	z_2 (4)	z_1 (2)	z_0 (1)	
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0
2	0	0	1	0	0	0	1	0	2
3	0	0	1	1	0	0	1	1	3
4	0	1	0	0	0	0	1	0	2
5	0	1	0	1	0	1	0	1	5
6	0	1	1	0	0	0	1	1	3
7	0	1	1	1	0	1	1	1	7
8	1	0	0	0	0	1	0	0	4
9	1	0	0	1	0	1	0	0	4
10	1	0	1	0	0	1	0	1	5
11	1	0	1	1	1	0	1	1	11
12	1	1	0	0	0	1	1	0	6
13	1	1	0	1	1	1	0	1	13
14	1	1	1	0	0	1	1	1	7
15	1	1	1	1	0	1	1	1	7