Name: Hayano Hajime SID: 604460151

Last First

UCLA COMPUTER SCIENCE DEPARTMENT

MIDTERM EXAMINATION

CS M51A/EE M16 Summer 2016 Section 1
Logic Design of Digital Systems

July 24, 2016

Dr. Yutao He

Rules:

This is a closed-textbook, closed-note, and independent exam (110 minutes). You may use two-page 8.5"x11" single-sided note. No scratch paper or calculator is allowed. Points are assigned to the problems based on estimates of how long they should take. PACE YOURSELF ACCORDINGLY. The order may not reflect the degree of difficulty. BROWSE THROUGH THE ENTIRE SET first to decide the order you want to follow. READ THE PROBLEM DESCRIPTION CAREFULLY. Be sure to include all final answers at indicated locations. Write down your Student ID at the top of each page. Use provided space for all work. Have fun and good luck!

Honor Code:

I attest that I have not given or received any aid or discussion in relationship to this exam.

Your Signature

Your Score:

No.	Your Score	Maximal Score
#1	8	10
#2	15	15
#3	10	10
#4	20	20
#5	10	10
#6	13	15
#7	19	20
Total	95	100

47

Problem No. 1 (10 points)

Part (a) An electric garage door opener *iLock* uses a small radio transmitter that sends a 5-bit sequence to a receiver inside the garage to open the door, and each receiver is supposed to respond to a different sequence. Suppose you live on a street with 40 houses, each using the same model of the opener.

×1 **(a.1)** How many houses must share a sequence with some other houses on the street?

Answer: 16 houses.

Show your work below for full credit: 9 houses $\leq$ X $\leq$ 16 houses

8 houses ×2

16 8 4 2 1

max $2^5 = 32$ unique sequence $8 \times 2 = 16$.

2 × **(a.2)** Suppose a burglar Digi Hak wants to break into one specific garage using this type of opener. If he has a transmitter which can send different 5-bit sequences, at most how many times he needs to try before he breaks into any particular garage?

Answer: 32

Show your work below for full credit:

$2^5 = 32$

Part (c) A electronic cylinder *iCylinder* measures liquid volume by one 3-digit mixed-radix number system (*gallon*, *quart*, *pint*). Recall their relationships are: 1 *gallon* = 4 *quarts*, 1 *quart* = 2 *pints*. Assume that radix for digit *gallon* is fourteen. 14

×5 **(c.1)** Using binary code, at least how many bits are needed to encode the largest number of *pints* that *iRuler* can measure?

Your Answer: 7 bits.

$14 \times 4 \times 2 =$ $\begin{array}{r} 14 \\ \times\ 8 \\ \hline 112 \end{array}$ $112 - 1 = 111$

$2^7 = 128 - 1 = 127 > 111$

(c.2) How many *pints* are represented by a reading of $X = (12, 2, 1)$?

Your Answer: 101 *pints*.

Show all your work below for full credit:

$12 \times 4 = 48$
$+2$
$50 \times 2 = 100$
$+1$
101

Problem No. 2 (15 points) 15

Part (a) Name the following 2-input switching functions:

(a.1) The output of f_1 is 1 **if and only if** both inputs are 1. It is and function.

(a.2) The output of f_2 is 1 if the inputs **are different**. It is xor function.

(a.3) The output of f_3 is 1 **if no more than** one input is 1. It is nand function.

(a.4) The output of f_4 is 1 **only when** both inputs are 0. It is nor function.

(a.5) The output of f_5 is 1 **if at least** one input is 1. It is or function.

(a.6) The output of f_6 is 1 if number of 1's in inputs **are even**. It is xnor function. ✓

Part (b) The logic designer Logik Luv plans to design one single two-input (x,y) "multi-function" logic module with two-output (g_1, g_2) to implement these switching functions in Part (a). To do so, he introduces two additional inputs (a, b) that decide the output functions g_1 and g_2 as follows:

a	b	g_1	g_2
0	0	x'	y'
0	1	f_1	f_2
1	0	f_3	f_4
1	1	f_5	f_6

(b.1) Filling in the following 2-D truth table for the functions g_1 and g_2. Place values of g_1 and g_2 in each cell **in order of (g_1, g_2)**.

		(x,y)			
		00	01	10	11
(a,b)	00	(1,1)	(1,0)	(0,1)	(0,0)
	01	(0,0)	(0,1)	(0,1)	(1,0)
	10	(1,1)	(1,0)	(1,0)	(0,0)
	11	(0,1)	(1,0)	(1,0)	(1,1)

✓

(Continued on the next page)

(Problem No. 2 - Continue)

(b.2) Write the *minterm* expression for $g_1(a,b,x,y)$ in compact form:

$g_1(a,b,x,y) = \sum m$ { 0, 1, 7, 8, 9, 10, 13, 14, 15 }

(b.3) The *switching* expression for m_0 is $a'b'x'y'$.

(b.4) Write the *maxterm* expression for $g_2(a,b,x,y)$ in compact form:

$g_2(a,b,x,y) = \prod M$ { 1, 3, 4, 7, 9, 10, 11, 13, 14 }

(b.5) The *switching* expression for M_4 is $a+b'+x+y$.

✓

g_1

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

g_2

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

(End of Problem No. 2)

Problem No. 3 (10 points)

Your high school buddy BB Frank is interviewed for an internship position at a startup *Cookle Electronics*. One of his interview questions is the tabular minimization using the Quine-McCluskey algorithm for the following 4-input switching function:

$$f(a,b,c,d) = \sum m(0,1,2,5,6,7,8,9,10,14)$$

BB Frank has completed the first step and the Prime Implicant Chart is shown below. You have to help him identify the essential prime implicants and then write the minimal AND-OR (sum-of-product) expression for $f(a,b,c,d)$.

Prime Implicant Chart

	Minterms									
Prime Implicants	0	1	2	5	6	7	8	9	10	14
a b c d $-00-$	X	X					X	X		
$-0-0$	X		X				X		X	
$--10$			X		X				X	X
$0-01$		X		X						
$01-1$				X		X				
$011-$					X	X				

+5

Part (a) The essential prime-implicants in switching expressions are:

$b'c'$, cd'

+5

Part (b) The minimal switching expression in AND-OR form is:

$f(a,b,c,d) =$ $b'c' + cd' + a'bd$

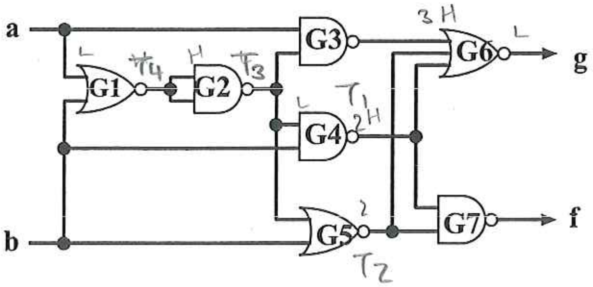
Show all your work on the *prime implicant chart* above for full credit.

$x + y$ L

Problem No. 4 (20 points)

20

Given the gate network below, answer the following questions:



Part (a) Assuming that negated variables are available and that NAND and NOR gates have the same delays, identify the *critical path* of the network by listing its gates along the path, starting at the inputs:

G1 → G2 → G4 → G6

Part (b) Assuming that load factors of all gates equal to 1 and that both outputs f and g have the output load value L, list the output load value of every gate in the *critical path* (e.g., G3: 1):

G1 : 2, G2 : 3, G4 : 2, G6 : L

Part (c) Write the expression of the longest network propagation delay T_{pHL} in terms of delays of each gate (You do not need to compute the final result but the transition direction at each gate has to be indicated):

$T_{pHL} =$ $t_{pLH}(2NOR) + t_{pHL}(2NAND) + t_{pLH}(2NAND) + t_{pHL}(3NOR)$

Part (d) Assuming that negated variables are available, find the minimal switching expression of the output f in two-level AND-OR (sum-of-product) form. Show your work below for full credit.

$f(a, b) =$ $a+b$.

✓

Your work for Part (d):

(Extra space available on the next page)

$(T_1T_2)'$

$((T_3b)'(T_3+b)')'$

$(T_3b)+(T_3+b)$

T_3b+T_3+b

T_3+b

$(T_4T_4)'+b$

$T_4'+b$

$(a+b)''+b$

$a+b+b$

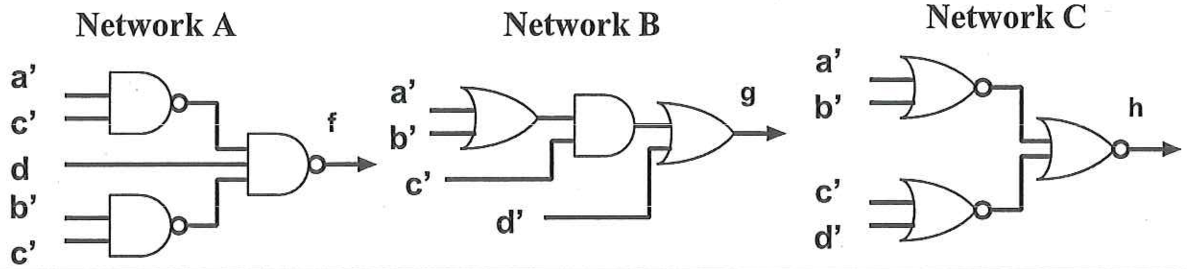
$a+b$

(Extra space for Problem No. 4)

(End of Problem No. 4)

Problem No. 5 (10 points)

Three gate networks A, B and C are given below. Tests have shown that two of them are *equivalent*, that is, they implement the same switching function. You are asked to identify the network that **is not** equivalent.



+5 Part (a) Describe in one sentence your approach.

Answer: Derive the switching functions and compare them.

+5 Part (b) The non-equivalent network is C.

Show all your work below for credit:

$A = ((a'c')'d(b'c')')'$ $B = ((a'+b')c')+d'$ $C = ((a'+b')'+(c'+d')')'$

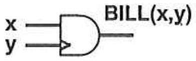
$= a'c'+d'+b'c'$ $=$ $a'c'+b'c'+d'$

$(a'+b')(c'+d')$

$a'c'+b'c'+d'a'+b'd'$

Problem No. 6 (15 points) 13

A former CSM51A class student Minnie has easily landed a job as a digital system designer in *YourTubix, Inc.* after graduation. She is motivated to revolutionarize the logic design and has invented a new type of logic gate, called *BILL* gates. Its symbol and truth table are given below.



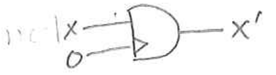
x	y	BILL(x,y)
0	0	1
0	1	1
1	0	0
1	1	1

Part(a) Is the *BILL* gate a universal set?

Your answer: yes, since we are able to make not, and, and or gates

Show all your work below for full credit.

not

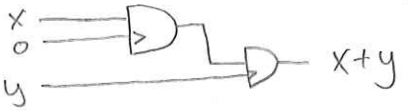


x 0	bill (x,0)
0 0	1
1 0	0

not y=0 x

Bill(x,0) = x'

or



x y	bill(x',y)
0 0	0
0 1	1
1 0	1
1 1	1

or

Bill(x',y)

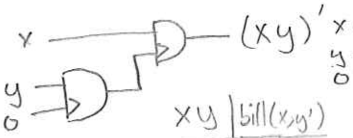
x' y	
1 0	0
1 1	1
0 0	1
0 1	1

and

bill(bill(x,y'),0)

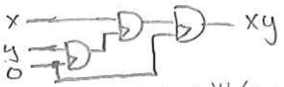
0 1 1 0
0 0 1 0
1 1 1 0
1 0 0 1

n and



x y	bill(x,y')
0 0	1
0 1	1
1 0	1
1 1	0

and



x y	bill(x,y')'
0 0	0
0 1	0
1 0	0
1 1	1

✓

Part(b) Implement a three-input XOR gate using one of the following approaches:
(1) use *BILL* gates only if the answer in Part (a) is yes, otherwise
(2) use at least one *BILL* gate and any other types of gates as needed.

Show all your work below for full credit.

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

(Extra space available on the next page)

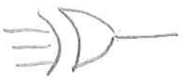
1 y = y

0 y = 1

x 1

1 0 1 0 1 0
a n 0 0 0 0
a r 1 1 1 0
1 0 1 0 0 1

(Extra space for Problem No. 6)

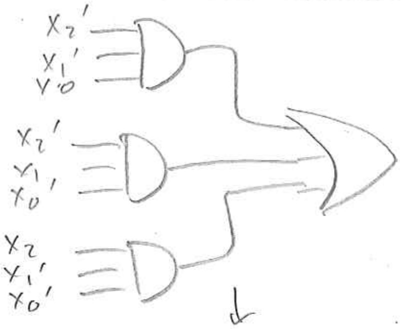


	x_2	x_1	x_0	z
	0	0	0	0
1	0	0	1	1
2	0	1	0	1
	0	1	1	0
4	1	0	0	1
	1	0	1	0
	1	1	0	0
	1	1	1	0

-2

x_2 \ x_1x_0	00	01	11	10
0	0	1	0	1
1	1	0	0	0

$x_2'x_1'x_0 + x_2'x_1x_0' + x_2x_1'x_0'$



not

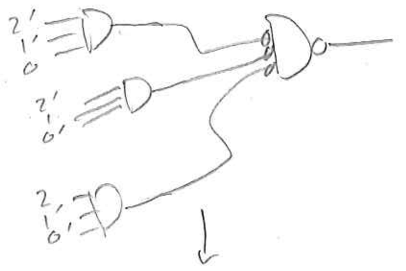
Bill(x,0) = x'

or

Bill(x',y) = Bill(Bill(x,0),y) = or

nand

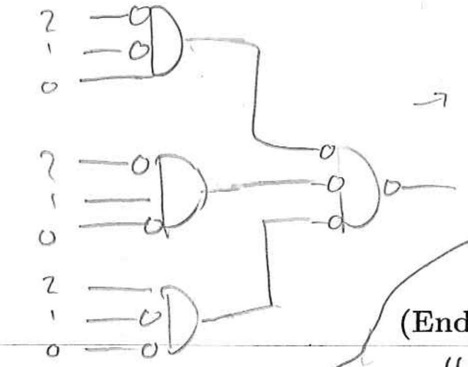
Bill(x,y') = Bill(x, Bill(y,0)) = nand

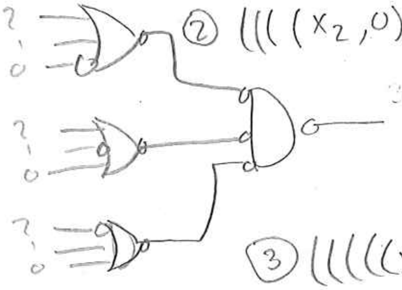


$(\overset{①}{(}x_2+x_1+x_0')\overset{②}{(}x_2+x_1'+x_0)\overset{③}{(}x_2'+x_1+x_0))'$

① Bill(Bill(Bill(Bill(x₂,0),x₁),0), Bill(x₀,0))

② ((((x₂,0),(x₁,0)),0), x₀)





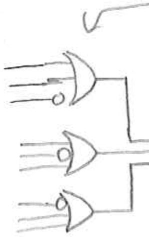
③ (((((x₂,0),0),x₁),0), x₀)

(End of Problem No. 6)

((((x₂,0),x₁),0),(x₀,0)),(((((x₂,0),(x₁,0)),0),x₀),0)

((((x₂,0),x₁),0),(x₀,0)),(((((x₂,0),(x₁,0)),0),x₀),0),

→ ((((((x₂,0),0),x₁),0),x₀),0))



* implement this using the or/not/nand gates we can make using only bill gates

((((x₂,0),x₁),0),(x₀,0)),(((((x₂,0),(x,0)),0),x₀),0,((((((x₂,0),0),x₁),0),x₀),0))

Problem No. 7 (20 points)

Fill in blanks in the following table by performing conversions and arithmetic operations in specified **binary number systems**. For one's complement and two's complement in parts (c) and (d), use only complementation and addition, and **indicate overflow if it occurs**. For part (e), use only shifting, complementation, and addition as needed, and no subtraction, multiplication, or division is allowed. Use provided space for all your work. Please **clearly LABEL** your steps and the final answer.

	Sign/Magnitude		1's Complement		2's Complement	
Operations	Bit Vector	Signed Integer	Bit Vector	Signed Integer	Bit Vector	Signed Integer
Part (a): x	11000	−8	11000	−7	11000	N/A
Part (b): y	01011	11	01011	11	01011	11
Part (c): $s = x + y$	N/A	N/A	00100	4	N/A	N/A
Part (d): $d = x - y$	overflow	−19	N/A	N/A	overflow	−19
Part (e): $z = \frac{1}{2}x + 2y$	N/A	N/A	N/A	N/A	010010	18

Show all your work below for full credit.

Part (a): x

SM) negative

1 1000

−8

comple form

1's) 11000

00111 true

−7

4

Part (b): y

pos

0 1011

8+2+1 = 11

1's true

01011

11

2's true

0 1011

11

4

(Continue on the next page)

(Continue from Problem No. 7)

Part (c): $s = x + y$

x= 11000

y= 01011

1's

```
  11000
  01011
 ------
 100011  =  00011
           ------
            00100  = 4
```

4

+3

Part (d): $d = x - y$

x= 11000

y= 01011

SM

x-y= x+(-y)

-y= 11011

```
   x      11000
 + -y   + 11011
         ------
         110011  ← overflow
```

Sign bits can't be added.

2's

x + -y

-y= 10100 +1

```
   10101
   11000
 + 10101
  ------
  101101
```

the carry-in does not equal carry out, overflow

(Continue on the next page)

(Continue from Problem No. 7) 8

0011
01000 0100

Part (e): $z = \frac{1}{2}x + 2y$

2's

x = 11000

y = 01011

$\frac{1}{2}$x = 1100

16+4+2
22

2y = 01011. * extend 001011 * shift = 010110

x extend

111
111100
+ 010110
1|010010

1100
+ 010110 => 111100 + 010110 = 1|010010 => <u>010010</u>

cut off

16 + 2 = 18

+4

—

ok

(End of Midterm)