UCLA
Electrical Engineering Department
EEM16, Fall 2013, Prof. Cabric
Midterm
Tuesday, October 29, 2013

1. Exam is closed book. You are allowed one $8\frac{1}{2} \times 11''$ double-sided cheat sheet.
2. Calculators are allowed
3. Show the intermediate steps leading to your final solution for each problem.
   There will be NO partial credit for work done correctly using a wrong answer from a previous part of a question. For example, if part a) is wrong and part b) depends on part a), then part b) will be wrong. Therefore, be very careful and double check your work!
4. You can use both sides of the sheets to answer questions.

Problem	Maximum Score	Your Score	Comments
1(a)	6	12	
1(b)	3		
1(c)	3		
2(a)	8	16	
2(b)	8		
3	10	10	
4(a)	3		
4(b)	3	17	
4(c)	3		
4(d)	3		
4(e)	3		
4(f)	3		
5	12	12	
6	12	12	
7	20	5	
Total:	100	84	

[redacted]

Problem 1 (12 pts): Number conversion

(a) (6 pts) $X = 85$ in decimal, convert it to Radix-2, Radix-8, Radix-16.

(b) (3 pts) $X = -47$ in decimal, convert it into 2's complement with 8-bit length.

(c) (3 pts) $X = 100110$ is the 2's complement representation of a number, convert this number to a decimal signed number (radix-10).

Answer:

a) $X = (85)_{10}$

$\frac{85}{2} = 42$ R 1

$\frac{42}{2} = 21$ R 0

$\frac{21}{2} = 10$ R 1

$\frac{10}{2} = 5$ R 0

$\frac{5}{2} = 2$ R 1

$\frac{2}{2} = 1$ R 0

$\frac{1}{2} = 0$ R 1

$(1010101)_2$

001 010 101
1 2 5

$(125)_8$

0101 0101
5 5

$= (55)_{16}$

b) $X = (-47)_{10}$

$2^8 = 256$

$256 - 47 = 209$

$\frac{47}{2} = 23$ R 1

$\frac{23}{2} = 11$ R 1

$\frac{11}{2} = 5$ R 1

$\frac{5}{2} = 2$ R 1

$\frac{2}{2} = 1$ R 0

$\frac{1}{2} = 0$ R 1

$47 = 00101111$

$(-47) = 11010001$

c) $X = 100110 < 0$ $X_R = 2 + 4 + 32 = 38$

$2^6 - |x| = X_R$

$x = 38 - 64 = -26$

16

Problem 2 (16 pts) Boolean Algebra:

(a) (8 pts) An operation * is defined for binay variables a and b as follows:

$$a * b = ab + a'b'$$

Let $c = a * b$. Determine which of the following identities are valid:

a1. $a = b * c$

a2. $a * bc = 1$

(b) (8 pts) Reduce the following Boolean expression to a minimum number of literals:

$$xyz + x'y + xyz'$$

Answer:

a) 1) $a = b * c$?

$bc + b'c'$

$b(a*b) + b'(a*b)'$

$b(ab + a'b') + b'(ab + a'b')'$

$b(ab + a'b') + b'((ab)'(a'b')')$

$b(ab + a'b') + b'(a'+b')(a+b)$

$ab + 0 + b'(0 + a'b + b'a + 0)$

$ab + b'(a'b + ab')$

$ab + ab'$

$a(b+b') = \boxed{a}$ $\boxed{\text{Valid}}$

2) $a * bc = 1$

~~$(a*b)c = abc + a'b'c$~~

$a*(bc) = abc + a'(bc)' = abc + a'(b'+c')$

$= abc + a'b' + a'c'$

note that:

abc	abc + a'b' + a'c'
100	0

(partial truth table to provide counter example)

Therfore $a * bc \neq 1$

$\boxed{\text{Invalid}}$

see back

b) $xyz + x'y + xyz'$

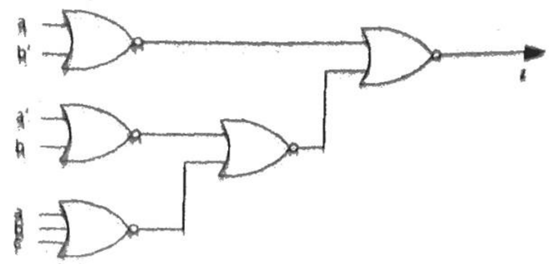
$xy(z+z') + x'y$

$y(x+x') = \boxed{y}$

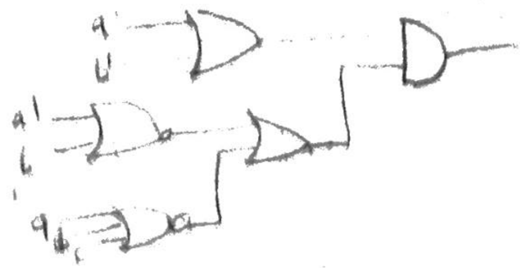
10

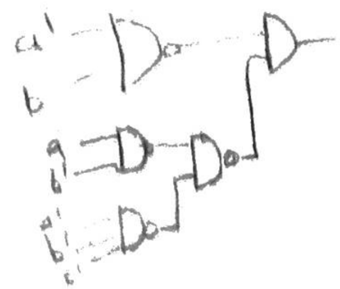
Problem 3 (10 pts):

Obtain a minimal two-level NAND-NAND network for the function implemented by the network shown below.



Answer:





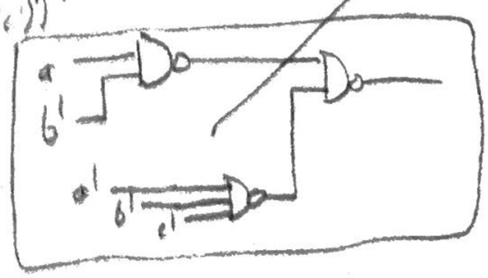
$((a+b')' + ((a'+b)' + (a+b+c)')')' = z$

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

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

$ab' + a'b'c'$

$((ab')' \cdot (a'b'c')')'$



Problem 4 (18 pts):

Design an efficient multiple-of-3 circuit that takes in a decimal digit 0-9 represented as a 4-bit binary number a[3:0], i.e. a circuit whose output is logic 1 when the input a[3:0] represents number 3, 6, or 9.

(a) (3 pts) Write a truth table for the function.

(b) (3 pts) Draw a Karnaugh map of the function.

(c) (3 pts) Identify the prime implicants of the function.

(d) (3 pts) Identify which of the prime implicants (if any) are essential.

(e) (3 pts) Find a switching expression of the function with minimal gate inputs and minimal number of gates.

(f) (3 pts) Draw a logic gate diagram of the function **using NAND gates only**.

Answer:

(a) Write your truth table in the table below:

a[3]	a[2]	a[1]	a[0]	output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	-
1	0	1	1	-
1	1	0	0	-
1	1	0	1	-
1	1	1	0	-
1	1	1	1	-

3

(b) Draw a Karnaugh map of the function.

Fill the following Karnaugh map template:

			a[1]	a[1]	
	0	0	1	0	
	0	0	0	1	a[2]
a[3]	-	-	-	-	a[2]
a[3]	0	1	-	-	
		a[0]	a[0]		

3

17

(c) Identify the prime implicants of the function.
List the prime implicants:

$a_2'a_1a_0$, $a_2a_1a_0'$, a_3a_0 3

(d) Identify which of the prime implicatns (if any) are essential.
List the essential prime implicants:

$a_2'a_1a_0$, $a_2a_1a_0'$, a_3a_0 (All are essential) 3

(e) Find a switching expression of the function with minimal gate inputs and minimal number of gates.
Show the work on the Karnaugh map on the previous page and then list the implicants you would use and the corresponding Boolean logic equation here.

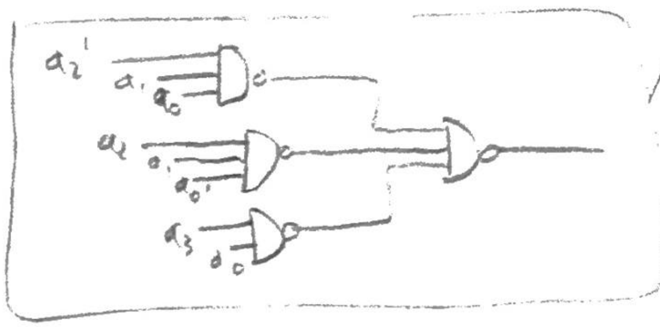
implicants: $a_2'a_1a_0$, $a_2a_1a_0'$, a_3a_0

$a_2'a_1a_0 + a_2a_1a_0' + a_3a_0$ 2

(f) Draw a logic gate diagram of the function (remember, only NAND gates)

$(a_2'a_1a_0 + a_2a_1a_0' + a_3a_0)$

$((a_2'a_1a_0)' \cdot (a_2a_1a_0')' \cdot (a_3a_0)')'$ 3



12 ✓

Problem 5 (12 pts):

Among the following three gates, which ones are universal and prove them. (you may use constants 0 and 1 as inputs wherever necessary)

X	Y	X*Y	XΔY	XΦY
0	0	0	1	0
0	1	0	1	0
1	0	1	0	1
1	1	0	1	1

Answer:

{NOT, AND} is a universal set

$*$ NOT: $(1*a) = a'$

AND: $((a*0)*(a*1))*(1*((b*0)*(b*1)))$ $a \cdot b$

$*$ is universal ✓

Δ NOT: $a\Delta 0 = a'$

AND: $(((a\Delta 0)\Delta 0)\Delta(b\Delta 0))\Delta 0 = ab$

Δ is universal ✓

Φ $X\Phi Y = X$ (no dependence on Y, so not universal) ✓

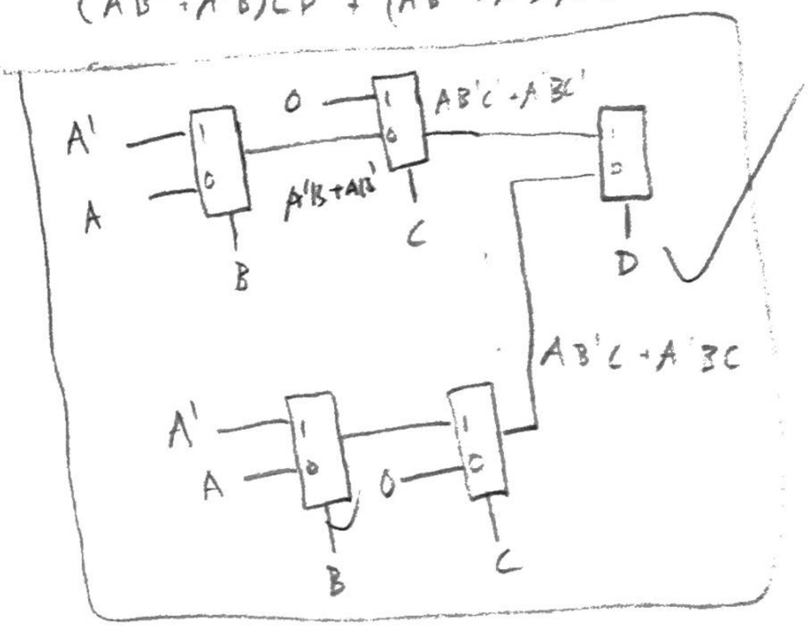
12 ✓

Problem 6 (12 pts):
Implement the following Boolean expression with MUX 2:1 gates:

$$F = AB'CD' + A'BCD' + AB'C'D + A'BC'D$$

Answer:

$F = F(A,B,C,0)D' + F(A,B,C,1)D$
$= (AB'C + A'BC)D' + (AB'C' + A'BC')D$
$(AB' + A'B)CD' + (AB' + A'B)C'D$

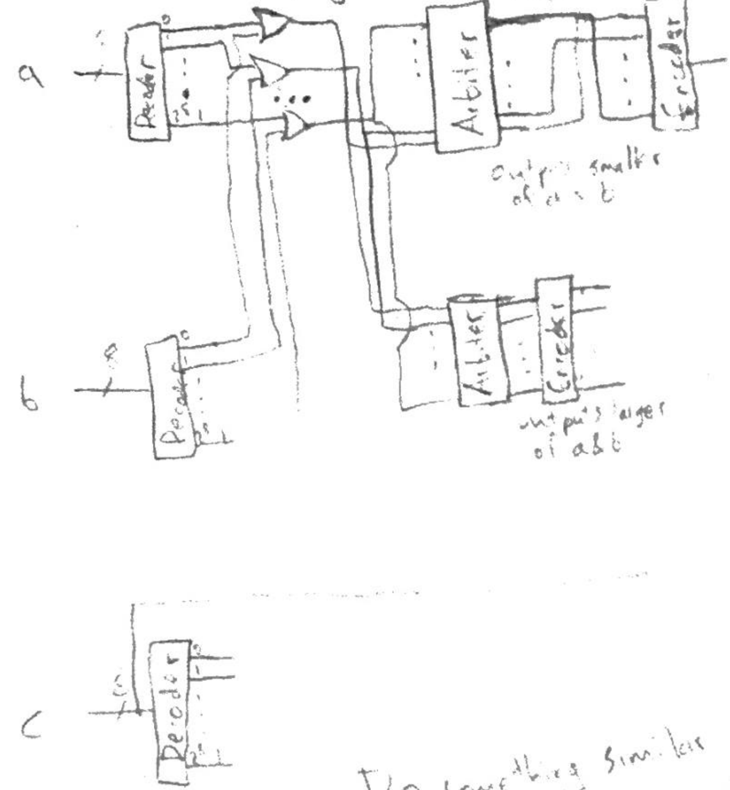


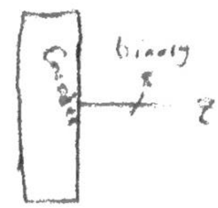
Problem 7 (20 pts):

Design a combinational circuit with three 8-bit inputs a[7:0], b[7:0], and c[7:0] representing binary numbers that outputs the median value of the three input number (i.e. the middle value if the input numbers a, b, and c are ordered by their value). You may use any of the primitive gates (NOT, AND, OR, NAND, NOR, XOR, XNOR) or any of the building blocks discussed in the lecture (decoder, multiplexer, encoder, arbiter, priority encoder, comparator, adder).

Answer:

Use OR gates on One hot & use Arbiter to get a one hot representation of largest or smallest





5

Do something similar to compare a & c as well as a & b. Then use a multiplexer to select a final comparison to be greater than or less than and send this to the Encoder at the end