ECE 198JL Second Midterm Exam
Spring 2013

Tuesday, March 5th, 2013

Name: ______________________________ NetID: ______________

Discussion Section:

10:00 AM	[ ] JD1	
11:00 AM	[ ] JD2	
2:00 PM	[ ] JD3	
4:00 PM	[ ] JD4	

- **Be sure your exam booklet has 10 pages.**
- **Be sure to write your name and lab section on the first page.**
- **Do not tear the exam booklet apart; you can only detach the last page.**
- **We have provided Boolean properties at the back.**
- **Use backs of pages for scratch work if needed.**
- **This is a closed book exam. You may not use a calculator.**
- **You are allowed one handwritten 8.5 x 11" sheet of notes.**
- **Absolutely no interaction between students is allowed.**
- **Be sure to clearly indicate any assumptions that you make.**
- **The questions are not weighted equally. Budget your time accordingly.**
- **Don't panic, and good luck!**

Problem 1	11 points:	________
Problem 2	14 points:	________
Problem 3	15 points:	________
Problem 4	10 points:	________
Problem 5	22 points:	________
Problem 6	18 points:	________
Problem 7	10 points:	________
Total	100 points:	________

Problem 1 (11 pts): Boolean algebra

1. Simplify expression y'(x'z+y'z)'. Write each step separately in the space provided. Name the property used for each step. First step is already written for you. (Refer to Boolean algebra properties on the last page of the exam booklet.)

			Property
y′(x′z+y′z)′	=	y′(x′z)′(y′z)′	DeMorgan
	=	y'(x+z')(y+z')	DeMorgan
	=	y'(xy+z')	Distributivity
	=	y'xy + y'z'	Distributivity
	=	xyy' + y'z'	Commutativity
	=	x.0 + y'z'	Complementarity
	=	0 + y'z'	Null
	=	y'z'	Identity

2. Prove by **perfect induction** consensus property: xy + yz + x'z = xy + x'z

x	y	z	xy+yz+x'z	xy+x'z
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

Since the truth tables for both functions are the same, then they must be equivalent

3. Write dual for x+y′zx′+0•x. **Do not simplify.**

Answer: x(y'+z+x')(1+x)

4. Let f(w,x,y,z)=m_9. Find its dual and write it in M_i notation.

Answer: M_6

$m_9 = wx'y'z \longrightarrow$ Dual: $w + x' + y' + z = M_6$

Problem 2 (14 pts): Canonical forms

A committee has members A, B, and C. Variables a, b, c have value 1 iff A, B, C respectively vote in favor of a proposal. Design a combinational circuit whose output g is 1 iff there is a majority in favor.

1. Fill in truth table.

a	b	c	g
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2. Using above table, write **canonical SOP** representation using **literals**.

Answer: $g = a'bc + ab'c + abc' + abc$

3. Using above table, write **canonical SOP** representation using **minterm** notation m_i.

Answer: $g = m_3 + m_5 + m_6 + m_7$

4. Using above table, write **canonical POS** representation using **literals**.

Answer: $g = (a+b+c)(a+b+c')(a+b'+c)(a'+b+c)$

5. Using above table, write **canonical POS** representation using **maxterm** notation M_i.

Answer: $g = M_0 M_1 M_2 M_4$

6. For function f(a,b,c,d)=a′bc+a′cd′, write corresponding **canonical SOP**.

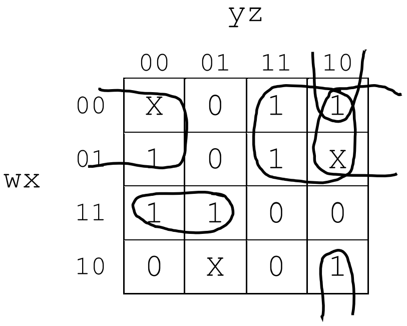
Answer: $f = a'bcd' + a'bcd + a'b'cd'$

7. For function g(w,x,y,z)=(w+x′)(w+x′+y+z′), write corresponding **canonical POS**.

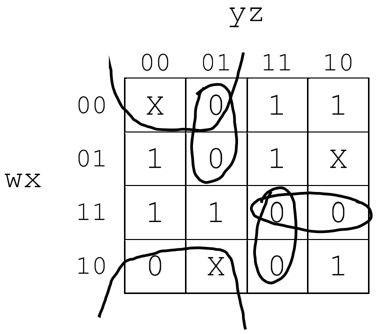
Answer: $g = (w+x'+y'+z')(w+x'+y'+z)(w+x'+y+z')(w+x'+y+z)$

Problem 3 (15 pts): Function simplification

Consider a 4-variable Boolean function f(w,x,y,z) given by its K-map (drawn twice):



wx \ yz	00	01	11	10
00	X	0	1	1
01	1	0	1	X
11	1	1	0	0
10	0	X	0	1



wx \ yz	00	01	11	10
00	X	0	1	1
01	1	0	1	X
11	1	1	0	0
10	0	X	0	1

1. List the **essential prime implicants**.

Answer: $w'y$ and $x'yz'$

2. Give a **minimal SOP** expression for f(w,x,y,z) and **show the corresponding loops** on the left map.

Answer: $f = w'y + x'yz' + w'z' + wxy'$

3. Give a **minimal POS** expression for f(w,x,y,z) and **show the corresponding loops** on the right map.

Answer: $f = (x+y)(w+y+z')(w'+y'+z')(w'+x'+y')$

(The solution is not unique, other minimal POS exist)

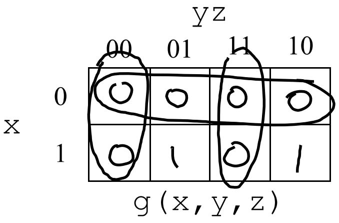
4. Do your answers to **Part 2** and **Part 3** represent the same Boolean function? Justify your answer.

No, because their values for the "don't cares" may differ. For example:

Part 2: $f(0,0,0,0)=1$ Part 3: $f(0,0,0,0)=0$

5. Let g(x,y,z)=x(y⊕z). Fill in its K-map on the right and write **minimal POS** below. Show the corresponding loops.

Answer: $g = x(y+z)(y'+z')$

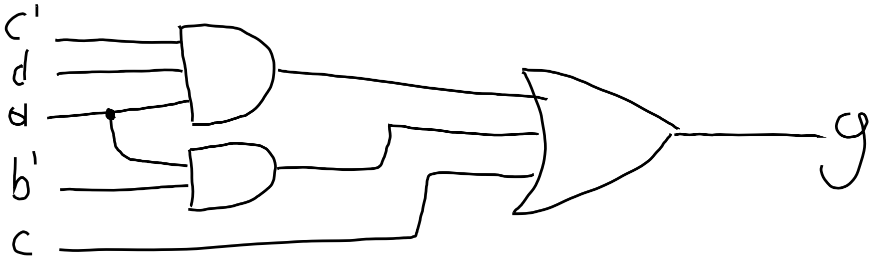


x \ yz	00	01	11	10
0	0	0	0	0
1	0	1	0	1

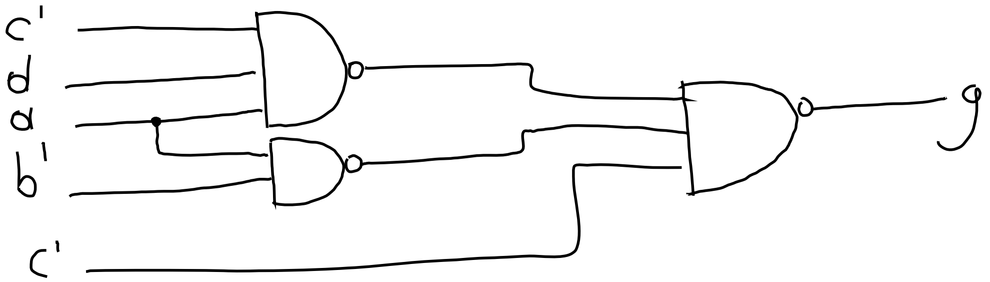
g(x,y,z)

Problem 4 (10 pts): 2-level circuits

1. Implement Boolean function `g(a,b,c,d)=ac'd+ab'+c` as a two-level network using **AND** and **OR** gates **only**. Assume that inverted inputs are available. Draw the circuit.

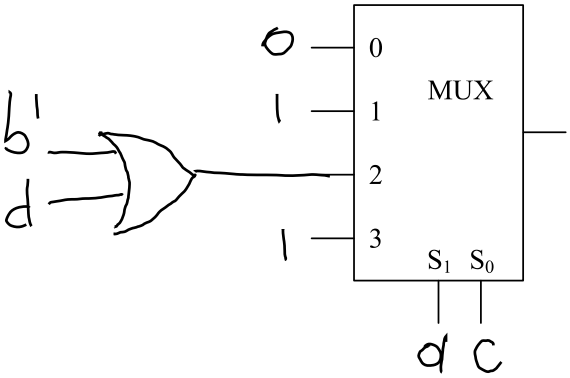


2. Re-implement the same function using **NAND** gates **only**. Do not use more than 6 NAND gates. Assume that inverted inputs are available. Draw the circuit.



3. Re-implement the same function using a **4:1 MUX** and no more than **one extra gate**. Assume that inverted inputs are available. Draw the circuit. Show your work. *Hint:* Draw K-map.

ab \ cd	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	1	1	1
10	1	1	1	1



$a=0, c=0 \Rightarrow g=0$

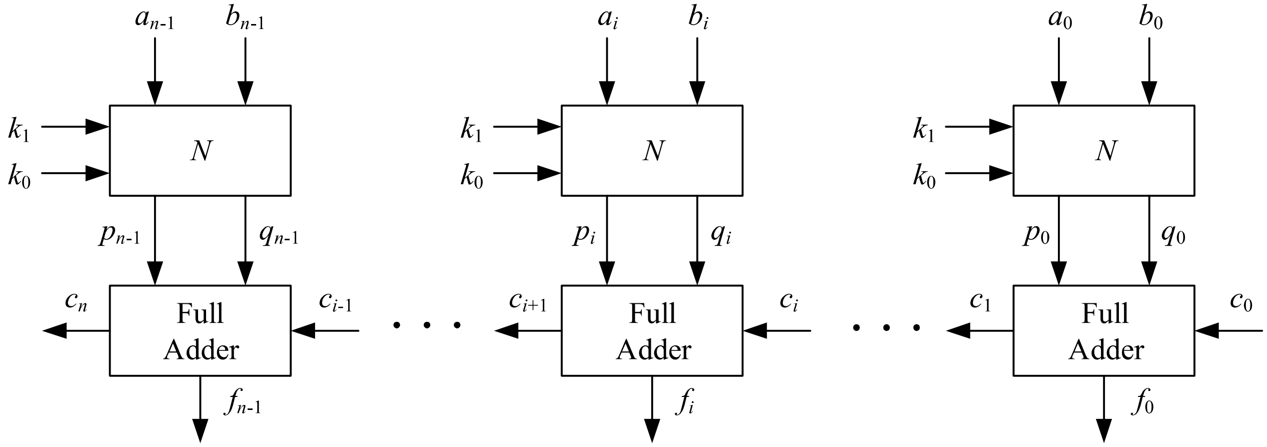
$a=0, c=1 \Rightarrow g=1$

$a=1, c=0 \Rightarrow g=b'+d$

$a=1, c=1 \Rightarrow g=1$

Problem 5 (22 pts): Combinational logic design

Part A. An n-bit arithmetic unit takes inputs $A=a_{n-1} \dots a_0$ and $B=b_{n-1} \dots b_0$, interpreted as the n-bit two's-complement representations of numbers.



The control signals are k_1, k_0, and c_0 (the carry-in to stage 0). At each stage i, the inputs to the full adder are

$$p_i = a_i k_1' k_0 + b_i k_1$$
$$q_i = a_i k_1' k_0' + b_i k_1 + b_i' k_0$$

K_1	K_0	p_i	q_i
0	0	0	a_i
0	1	a_i	b_i'
1	0	b_i	b_i
1	1	b_i	1

1. Determine the function of A and B produced by each of the following combinations of control signals:

k_1 k_0 c_0	Function
0 1 1	A − B
1 0 0	2B

2. Determine the values for the control signals to produce each of the following functions:

Function	k_1	k_0	c_0
A plus 1	0	0	1
B minus 1	1	1	0

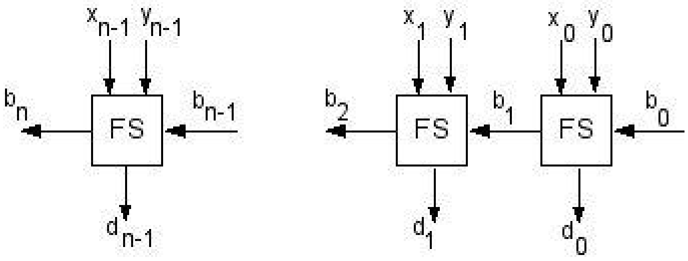
Hint: write out truth tables for p_i and q_i as functions of k_1 and k_0.

3. Write c_0 as a function of k_1 and k_0.

Answer: $c_0 = K_1'$

K_1	K_0	C_0
0	0	1
0	1	1
1	0	0
1	1	0

Part B. You are designing a Full Subtractor (FS) circuit. The FS has inputs x_i, y_i, and a borrow input b_i. There are two outputs: difference d_i and borrow-out b_{i+1}.



The FS cell should be designed so that the n-bit subtractor network shown above correctly computes D=X−Y, where $X = x_{n-1} \ldots x_1x_0$ and $Y = y_{n-1} \ldots y_1y_0$ are nonnegative n-bit binary numbers. Assume X >= Y.

Examples for n=3:

```
  X     x2x1x0    101     100
- Y  -  y2y1y0  - 011   - 001
  D     d2d1d0    010     011
```

1. Draw K-maps for d_i and for b_{i+1}. *Hint:* Remember that b_{i+1} and d_i are functions of only the 3 inputs: x_i, y_i, b_i. Try some examples, and start with the rightmost FS cell.

x_iy_i

b_i	00	01	11	10
0	0	1	0	1
1	1	0	1	0

d_i

x_iy_i

b_i	00	01	11	10
0	0	1	0	0
1	1	1	1	0

b_{i+1}

$d_i = (b_i + x_i + y_i)(b_i' + x_i + y_i')(b_i + x_i' + y_i')(b_i' + x_i' + y_i)$ (POS)

2. Give minimal POS expressions for d_i and for b_{i+1}.

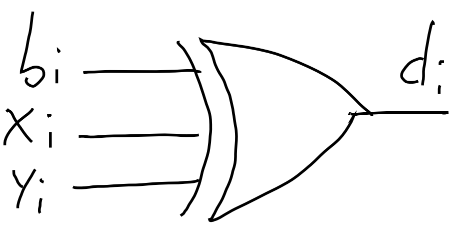
$d_i = b_i x_i' y_i' + b_i' x_i' y_i + b_i x_i y_i + b_i' x_i y_i'$ (SOP)

$b_{i+1} = (b_i + y_i)(b_i + x_i')(x_i' + y_i)$

Using SOP:

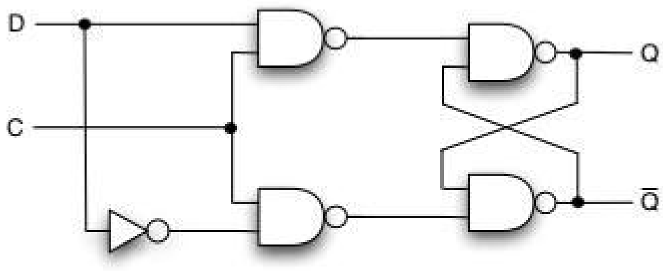
3. Implement circuit for computing d_i value using XOR gate(s) only.

$$d_i = b_i'(x_i \oplus y_i) + b_i(x_i' y_i' + x_i y_i)$$
$$= b_i'(x_i \oplus y_i) + b_i(\overline{x_i \oplus y_i})$$
$$= b_i \oplus x_i \oplus y_i$$



Problem 6 (18 pts): Sequential logic components

Part A. Shown below is the logic diagram of a gated D latch. It consists of 4 NAND gates and an inverter. It has 2 inputs: D and C.



D latch logic diagram

1. Complete the next-state table for this latch circuit.

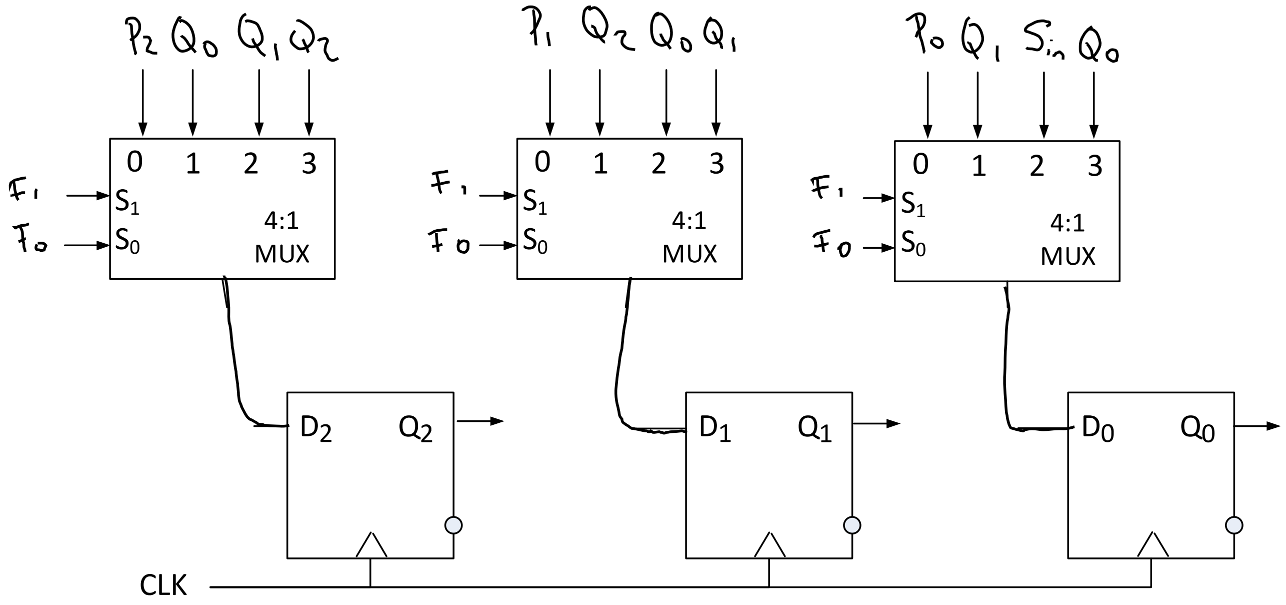
C	D	Q+
0	0	Q
0	1	Q
1	0	0
1	1	1

2. Express next state Q+ as a function of C, D, and Q.

Answer: $Q^+ = QC' + DC$

Part B. Complete the design of a 3-bit register that performs the operations listed in the table to the right. Parallel load inputs are labeled and indexed as `Pi`. Serial input is labeled as S_{in}. You may use inputs without drawing the wires, just write the appropriate labels at the ~~MAX~~ MUX inputs.

F_1	F_0	Operation
0	0	Parallel load
0	1	Circular shift right
1	0	Logical shift left
1	1	No change



Problem 7 (10 pts): Program analysis

Consider the following C program:

```
#include <stdio.h>

int main()
{
   unsigned int a,b,c;
   int function;
   int notfirst=0;

   for ( a = 0; a <= 1; a = a + 1 )
   {
      for ( b = 0; b <= 1; b = b + 1 )
      {
         for ( c = 0; c <= 1; c = c + 1 )
         {
            function = a & (b | ~c);
            if (function)
            {
                if (notfirst) printf("+");
                if (a) printf("a"); else printf("a'");
                if (b) printf("b"); else printf("b'");
                if (c) printf("c"); else printf("c'");
                notfirst = 1;
            }
         }
      }
   }

   printf("\n");

   return 0;
}
```

$f = a(b+c')$

Prints minterms of function

1. Write down EXACTLY the formatted text that will be printed on the terminal screen by the program.

$ab'c' + abc' + abc$

2. Explain in one sentence the function of the program; that is, what does it print?

It prints the cononical SOP of the boolean function $f = a(b+c')$

Boolean algebra properties

Commutativity	$x \cdot y = y \cdot x$	$x + y = y + x$
Associativity	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	$(x + y) + z = x + (y + z)$
Distributivity	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + y \cdot z = (x + y) \cdot (x + z)$
Idempotence	$x \cdot x = x$	$x + x = x$
Identity	$x \cdot 1 = x$	$x + 0 = x$
Null	$x \cdot 0 = 0$	$x + 1 = 1$
Complementarity	$x \cdot x' = 0$	$x + x' = 1$
Involution	$(x')' = x$	
DeMorgan’s	$(x \cdot y)' = x' + y'$	$(x + y)' = x' \cdot y'$
Absorption	$x \cdot (x + y) = x$	$x + x \cdot y = x$
No-Name	$x \cdot (x' + y) = x \cdot y$	$x + x' \cdot y = x + y$
Consensus	$(x+y) \cdot (y+z) \cdot (x'+z) =$ $(x+y) \cdot (x'+z)$	$x \cdot y + y \cdot z + x' \cdot z =$ $x \cdot y + x' \cdot z$