

**ECE 120 Second Midterm Exam
Spring 2017**

Tuesday, March 14, 2017

Name:

SOLUTIONS

NetID: _____

Discussion Section and TA Name:

9:00 AM	<input type="checkbox"/>	AB1 Rui	
10:00 AM	<input type="checkbox"/>	AB2 Rui	
11:00 AM	<input type="checkbox"/>	AB3 Matt	
12:00 PM	<input type="checkbox"/>	AB4 Pawel	
1:00 PM	<input type="checkbox"/>	AB5 Pawel	
2:00 PM	<input type="checkbox"/>	AB6 Gowthami	<input type="checkbox"/> ABA Hui ren
3:00 PM	<input type="checkbox"/>	AB7 Gowthami	<input type="checkbox"/> ABB Hui ren
4:00 PM	<input type="checkbox"/>	AB8 Yu-Hsuan	<input type="checkbox"/> ABC Sifan
5:00 PM	<input type="checkbox"/>	AB9 Yu-Hsuan	<input type="checkbox"/> ABD Surya

- Be sure that your exam booklet has 10 pages.
- Write your name and netid and check your discussion section on this page.
- Do not tear the exam booklet apart, except for the last two pages.
- 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 (both sides).
- Absolutely no interaction between students is allowed.
- Clearly indicate any assumptions that you make.
- The questions are not weighted equally. Budget your time accordingly.
- Show your work.

Problem 1	18 points	_____
Problem 2	16 points	_____
Problem 3	16 points	_____
Problem 4	16 points	_____
Problem 5	18 points	_____
Problem 6	16 points	_____

Total	100 points	_____
-------	------------	-------

Problem 1 (18 points): K-maps and Don't Cares

Consider the 4-variable function $F(A, B, C, D)$, with the following K-map (drawn twice). **Note:** there are extra copies of this K-map on the last page of the exam. Use them for scratch work, but they will NOT be graded. Make sure that you mark the two K-maps below correctly.

Minimal SOP

$F(A, B, C, D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0

Minimal POS

$F(A, B, C, D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0

1. (4 points) Give a **minimal SOP** expression for $F(A, B, C, D)$ and show the corresponding loops on the **left map**.

Minimal SOP: $F(A, B, C, D) = \underline{\overline{B}D + \overline{B}D}$

2. (5 points) Is the solution to part 1 unique? If it is not, write a different minimal SOP solution.

Circle one: **UNIQUE**

NOT UNIQUE

If not unique, write a different minimal SOP solution (but do not mark the loops):

Minimal SOP: $F(A, B, C, D) = \underline{\overline{B}D + CD}$

3. (4 points) Give a **minimal POS** expression for $F(A, B, C, D)$ and show the corresponding loops on the **right map**.

Minimal POS: $F(A, B, C, D) = \underline{(B+D)(\overline{B}+\overline{D})}$

4. (5 points) Is the solution to part 3 unique? If it is not, write a different minimal POS solution.

Circle one: **UNIQUE**

NOT UNIQUE

If not unique, write a different minimal POS solution (but do not mark the loops):

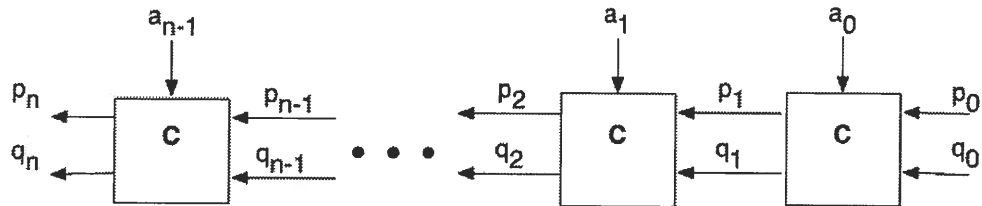
Minimal ~~SOP~~ POS: $F(A, B, C, D) = \underline{(B+D)(C+\overline{D})}$

Problem 2 (16 points): Bit-Sliced Design

Let $A = a_{n-1} \dots a_2 a_1 a_0$ be an n -bit input. In this problem you will design a circuit which outputs $p_n = 1$ if A contains two consecutive 1s.

For example, for $n = 5$: if $A = 10010$, then $p_5 = 0$; if $A = 01101$ or $A = 01111$, then $p_5 = 1$.

Your implementation should use the array shown below, which consists of n identical copies of a network C . C has inputs a_i and carry-in bits p_i and q_i . C has outputs p_{i+1} and q_{i+1} .



- The carry bits are used to convey information from one stage to the next. We need 3 such "messages" and will use $p_{i+1}q_{i+1} = 00, 01$, and $1x$ (where x denotes "don't care"). Explain in words the meaning of each $p_{i+1}q_{i+1}$ carry ($p_{i+1}q_{i+1} = 01$ has been provided).

p_{i+1}	q_{i+1}	Meaning
0	0	11 has not been detected & $a_i = 0$
0	1	11 has not been detected and $a_i = 1$
1	x	11 has been detected

- Specify the p_0 and q_0 values: $p_0 = 0$ and $q_0 = 0$

- Fill in the K-maps below for p_{i+1} and q_{i+1} .

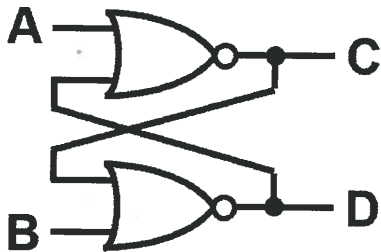
		$p_i q_i$			
		00	01	11	10
a_i	0	0	0	1	1
	1	0	1	1	1
		p_{i+1}			

		$p_i q_i$			
		00	01	11	10
a_i	0	0	0	x	x
	1	1	x	x	x
		q_{i+1}			

Problem 3 (16 points): Sequential Feedback Circuits and Multiplexers

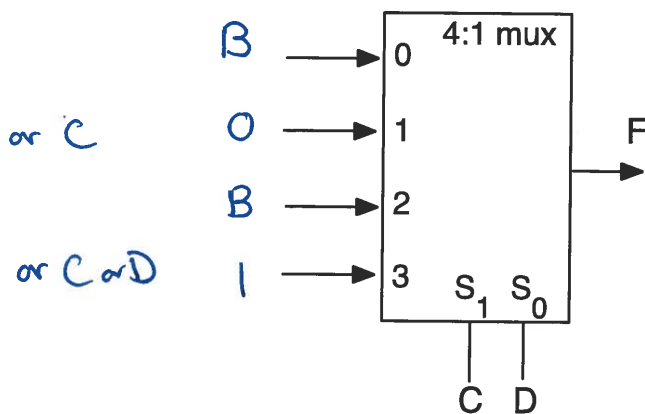
1. (8 points) For the circuit below, fill in the table with all stable configurations of the outputs C and D for each possible combination of inputs A and B. **Use only as many rows as necessary.**

Write only 0s and 1s in the boxes.



A	B	C	D
0	0	0	1
0	0	1	0
0	1	1	0
1	0	0	1
1	1	0	0

2. (8 points) Consider function $F(A,B,C,D)$ as defined in the following K-map. Implement F using **only** the multiplexer given below. **You may not add any components, nor any additional gates.** Complemented inputs are **not** available.



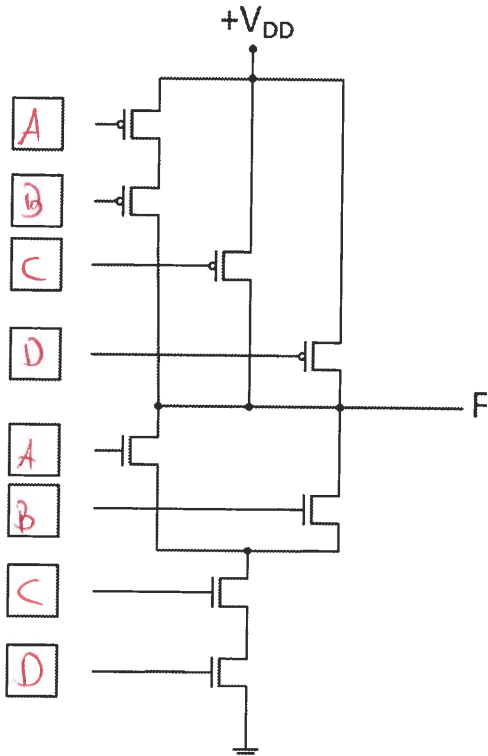
		CD			
		00	01	11	10
AB	00	0	x	1	x
	01	1	0	x	1
	11	1	0	x	1
	10	x	x	1	0
$F(A,B,C,D)$					

Problem 4 (16 points): CMOS Gates, Boolean Expressions and 2-Level Design

1. (8 points) The transistor-level circuit below implements the logic equation:

$$F = \overline{(A + B)CD}$$

Label the inputs to all transistors by filling in the boxes.



2. (4 points) Write the truth table for a function $Z(X)$ that takes a 3-bit unsigned integer input $X = x_2x_1x_0$ and outputs a 2-bit unsigned integer $Z = z_1z_0$. The output Z is the remainder after dividing X by 3. In other words:

$$Z = X \% 3$$

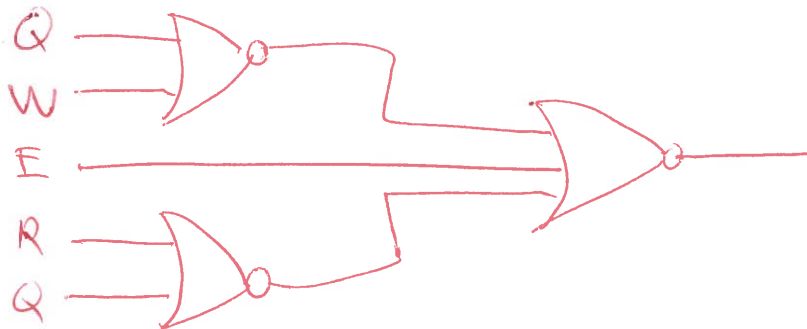
For example, if $X=5$, then $Z=2$. Describe the behavior of your system as a truth table.

x_2	x_1	x_0	z_1	z_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

3. (4 points) Implement the function

$$G = (Q + W)E'(R + Q)$$

as a 2-level NOR-NOR circuit using the minimum number of gates. Complemented inputs are available.



Problem 5 (18 points): Basic Registers

1. (6 points) Suppose a 3-bit register initially stores $Q_2Q_1Q_0 = 101$ and in the next clock cycle, the register has the values $Q_2Q_1Q_0 = 110$. Which of the following operations could the register be implementing? Circle **ALL** possible answers.

No change

Shift right

Shift left

Parallel load

Circular shift right

Circular shift left

Clear

Logical shift right

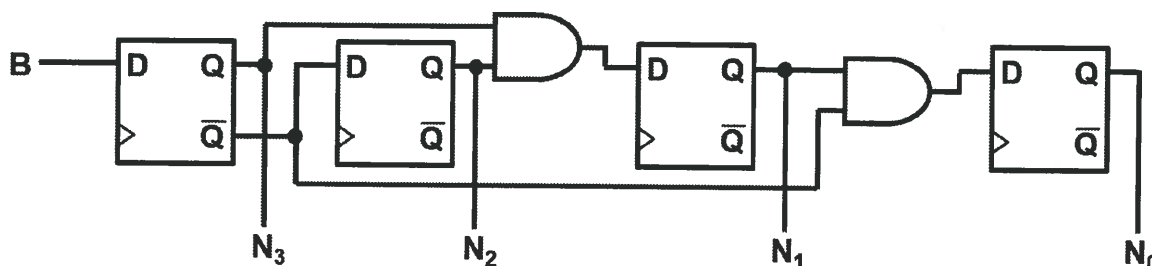
Logical shift left

Flip all bits

Arithmetic shift right

Arithmetic shift left

2. (12 points) Consider the register below, which is constructed from four positive-edge-triggered D flip-flops and two gates. A serial input B is fed into the leftmost flip-flop, and the values of all flip-flops can be read in parallel as the output $N = N_3N_2N_1N_0$. **All four flip-flops share a common clock signal (not shown in figure to reduce clutter).**



Assume that $N_3N_2N_1N_0$ are unknown at the start of operation (that is, in clock cycle 0) and that input B is set to the sequence 0, 1, 0, 1 (from clock cycle 0 to 3). Fill in the values of $N_3N_2N_1N_0$ for clock cycles 2 to 4 in the table below. If a particular bit's value is not possible to know, write "unknown."

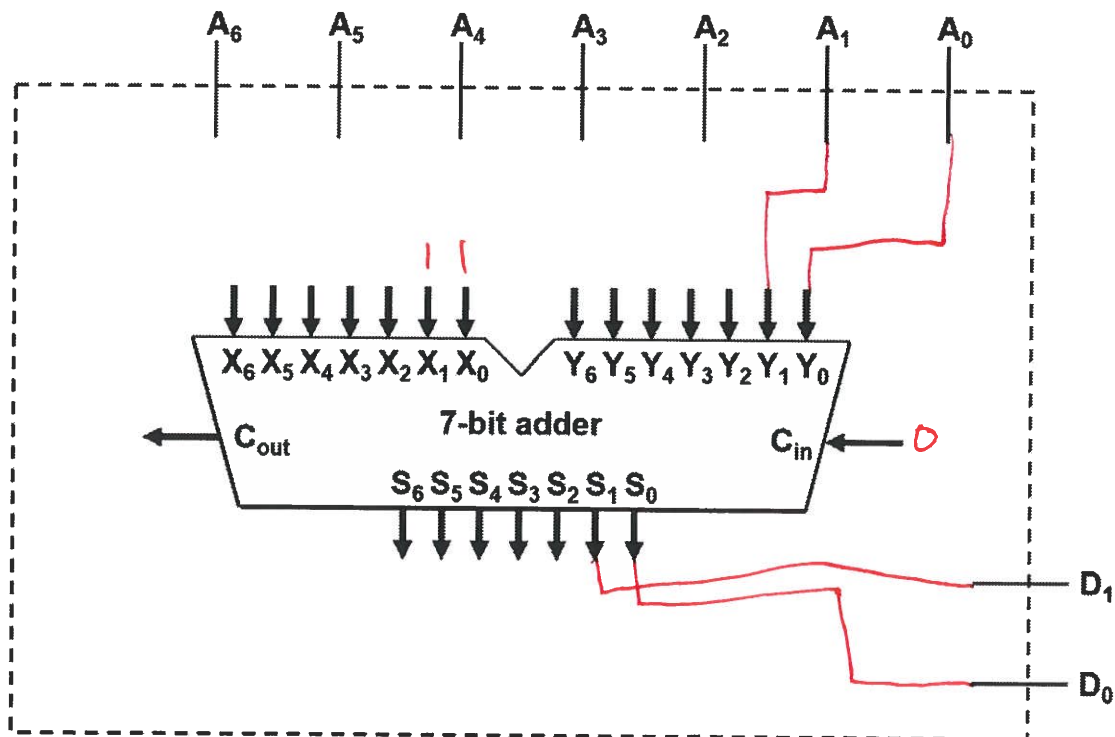
	B	N_3	N_2	N_1	N_0
cycle 0	0	unknown	unknown	unknown	unknown
cycle 1	1	0	unknown	unknown	unknown
cycle 2	0	1	1	0	unknown
cycle 3	1	0	0	1	0
cycle 4	unknown	1	1	0	1

Problem 6 (16 points): Building with Components and Abstraction

Prof. Lumetta needs your help. The lines at Daily Byte café have grown too long, and he needs your assistance in developing appropriate technological advances to aid the baristas. A barista is a person who makes caffeinated beverages, arguably the second-most important job in the world (after engineer).

1. (8 points) Prof. Lumetta has some 7-bit adders sitting around from an old class. He wants you to incorporate one into your solution for this problem. Given an ASCII character $A=A_6A_5A_4A_3A_2A_1A_0$ representing a digit from '1' (x31) to '4' (x34), calculate a 2-bit drink code $D=D_1D_0$ according to the table below. You may use the bits of character A as well as 0s and 1s to produce D . **You may not add any components, nor any additional gates.**

A	D_1D_0
'1'	00
'2'	01
'3'	10
'4'	11



(This problem continues on the next page.)

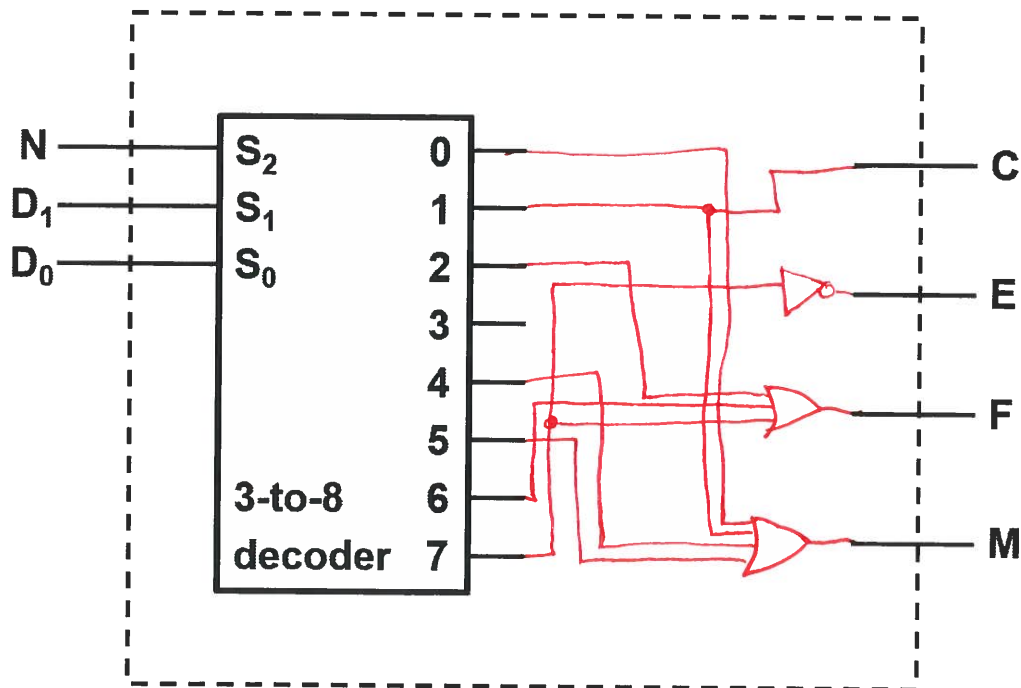
Problem 6, continued:

2. (8 points) Prof. Lumetta needs some logic to manage the various liquid dispensers. There are four liquids to be dispensed, each controlled by one output bit. E is for espresso, M is for steamed milk, F is for foamed milk, and C is for chocolate. A drink is made by dispensing combinations of these four liquids for two cycles, as shown by the table below.

	latte D ₁ D ₀ =00	mocha D ₁ D ₀ =01	cappuccino D ₁ D ₀ =10	macchiato D ₁ D ₀ =11
cycle N=0	E, M	E, M, C	E, F	E
cycle N=1	E, M	E, M	E, F	F

Each column in the table shows the liquids that must be dispensed for the two cycles needed to create a drink. For example, for a macchiato, the drink code is D₁D₀=11. In this case, in cycle N=0, espresso alone must be dispensed (so E=1 while M=0, F=0, and C=0), and in cycle N=1, foam alone must be dispensed (so F=1 while E=0, M=0, and C=0).

Given a 3-to-8 decoder with inputs wired to cycle number N (most significant bit), D₁, and D₀ (less significant bits), add **ONE INVERTER** and **TWO GATES OF YOUR CHOICE** (AND, OR, NOT, NAND, NOR, or XOR) to produce the signals C, E, F, and M that control the output of the four liquids.



M = $\overline{D_1}$ is ok, etc.

Table of ASCII Characters

Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex
(nul)	0	00	(sp)	32	20	@	64	40	`	96	60
(soh)	1	01	!	33	21	A	65	41	a	97	61
(stx)	2	02	"	34	22	B	66	42	b	98	62
(etx)	3	03	#	35	23	C	67	43	c	99	63
(eot)	4	04	\$	36	24	D	68	44	d	100	64
(enq)	5	05	%	37	25	E	69	45	e	101	65
(ack)	6	06	&	38	26	F	70	46	f	102	66
(bel)	7	07	'	39	27	G	71	47	g	103	67
(bs)	8	08	(40	28	H	72	48	h	104	68
(ht)	9	09)	41	29	I	73	49	i	105	69
(lf)	10	0a	*	42	2a	J	74	4a	j	106	6a
(vt)	11	0b	+	43	2b	K	75	4b	k	107	6b
(ff)	12	0c	,	44	2c	L	76	4c	l	108	6c
(cr)	13	0d	-	45	2d	M	77	4d	m	109	6d
(so)	14	0e	.	46	2e	N	78	4e	n	110	6e
(si)	15	0f	/	47	2f	O	79	4f	o	111	6f
(dle)	16	10	0	48	30	P	80	50	p	112	70
(dc1)	17	11	1	49	31	Q	81	51	q	113	71
(dc2)	18	12	2	50	32	R	82	52	r	114	72
(dc3)	19	13	3	51	33	S	83	53	s	115	73
(dc4)	20	14	4	52	34	T	84	54	t	116	74
(nak)	21	15	5	53	35	U	85	55	u	117	75
(syn)	22	16	6	54	36	V	86	56	v	118	76
(etb)	23	17	7	55	37	W	87	57	w	119	77
(can)	24	18	8	56	38	X	88	58	x	120	78
(em)	25	19	9	57	39	Y	89	59	y	121	79
(sub)	26	1a	:	58	3a	Z	90	5a	z	122	7a
(esc)	27	1b	;	59	3b	[91	5b	{	123	7b
(fs)	28	1c	<	60	3c	\	92	5c		124	7c
(gs)	29	1d	=	61	3d]	93	5d	}	125	7d
(rs)	30	1e	>	62	3e	^	94	5e	~	126	7e
(us)	31	1f	?	63	3f	_	95	5f	(del)	127	7f

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	$(x + y) \cdot (y + z) \cdot (x' + z) = (x + y) \cdot (x' + z)$	$x \cdot y + y \cdot z + x' \cdot z = x \cdot y + x' \cdot z$

Extra copies of K-map for problem 1 (use as scratch copies, we will NOT grade them)

$F(A,B,C,D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0

$F(A,B,C,D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0

$F(A,B,C,D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0

$F(A,B,C,D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0

$F(A,B,C,D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0

$F(A,B,C,D)$		CD			
		00	01	11	10
AB	00	0	X	1	X
	01	1	0	X	1
	11	1	0	X	1
	10	X	X	1	0