The University of Michigan
Department of Electrical Engineering and Computer Science

EECS 270 Fall 2003

Exam #1 Solutions

Name:____ UM ID:_______________

For all questions, show all work that leads to your answer.

Problem #	Possible Points	Points Earned
1	20	
2	13	
3	30	
4	12	
5	17	
6	8	
Total	100	

I have neither given nor received any unauthorized aid on this exam.

Signed: _________________________________

1. (20 Points Total)

(a: 5 pts) Prove the following theorem of Boolean algebra using existing theorems and axioms. Do not use K-maps or perfect induction. Indicate whenever you use one of the following theorems: covering, consensus, combining.

Theorem: $X + X'Y = X + Y$

$$
\begin{aligned}
& X + X'Y \\
&= X \cdot 1 + X'Y \\
&= X \cdot (Y + Y') + X'Y \\
&= XY + XY' + X'Y \\
&= (XY + XY') + (XY + X'Y) \\
&= X + Y \quad \textit{(Combining)}
\end{aligned}
$$

(b: 6 pts) Using the theorem you have just proven, reduce the following expression to two product terms. Do not use K-maps. Indicate whenever you use one of the following theorems: covering, consensus, combining.

$$ABC + C'D'E' + ABD + ABE$$

$$
\begin{gathered}
ABC + C'D'E' + ABD + ABE \\
AB(C + D + E) + C'D'E' \\
AB(C'D'E')' + C'D'E' \\
AB + C'D'E' \quad \textit{(by above theorem)}
\end{gathered}
$$

(c: 6 pts) Given the following function:

$$F = \sum_{X,Y,Z}(i_1, i_2, i_3) \ ; \ \ 0 \le i_1, i_2, i_3 \le 7 \ ; \ i_1 \ne i_2, i_2 \ne i_3, i_1 \ne i_3$$

What is the dual of F? *(Hint: pick three numbers for i1, i2, and i3 and work through the problem.)*

Recall that taking the dual of any function F results in a function G where the truth table of G is equal to that of F complemented, and then flipped upside-down.

Complement F:

$$F' = \prod_{X,Y,Z}(i_1, i_2, i_3)$$

Flip the truth table of F' upside-down:

$$F^D = G = \prod_{X,Y,Z}(7 - i_1, 7 - i_2, 7 - i_3)$$

Example:

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$F = \sum_{X,Y,Z}(0, 2, 7)$

X	Y	Z	Fd
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

F^D

X	Y	Z	Fd
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$F^D = \prod_{X,Y,Z}(7–0, 7–2, 7-7)$
$= \prod_{X,Y,Z}(7, 5, 0)$

(d: 3 pts) Given the function:

$F = \sum_{X,Y,Z} (i_1, \ldots, i_n) \ ; \ 0 \le n \le 8$, where $i_1, \ldots, i_n$ are not equal to each other.

For F to be self-dual, what must be the value of *n*? Explain your answer.

Recall that if a function is self-dual, the truth table of F' flipped upside-down must be equal to the truth table of F. This implies that, for F to be self-dual, it is necessary that F and F' have the same number of 0's and 1's. The only way for this to happen is if the number of 1's (and the number of 0's) of F is equal to 4. Therefore, n = 4.

2. (13 Points Total) Perform the following number problems:

(a: 3 pts) $BF.AC12_{16} = ?_8$

$$
\begin{array}{l}
\quad B \quad F \quad . \quad A \quad C \quad 1 \quad 2_{16} \\
= 1011\ 1111\ .\ 1010\ 1100\ 0001\ 0010_2 \\
\\
= 010\ 111\ 111\ .\ 101\ 011\ 000\ 001\ 001_2 \\
= 2 \quad 7 \quad 7\ .\ 5 \quad 3 \quad 0 \quad 1 \quad 1
\end{array}
$$

$BF.AC12_{16} = 277.53011_8$

(b: 3 pts) $233.61_{10} = ?_2$ Use 6 digits of precision to the right of the binary point.

233			.61
116 R 1	LSB	MSB	1 . 22
58 R 0			0 . 44
29 R 0			0 . 88
14 R 1			1 . 76
7 R 0			1 . 52
3 R 1		LSB	1 . 04
1 R 1			
0 R 1	MSB		

$233.61_{10} = 11101001.100111_2$

(c: 3 pts) Write the following number using signed-magnitude, one's complement, and two's complement representation using 6-bit strings.

$-14_{10} = ?_2$ $+ 14_{10} = 001110_2$

Signed Magnitude: 101110

One's Complement: 110001

Two's Complement: 110010

(d: 4 pts) Perform the following two's complement subtraction using 6-bit strings in the same way that a computer would. Briefly explain why an overflow did or did not occur.

(14) subtract (-20)

$14_{10} = 001110_2$
$-20_{10} = 101100_2$

```
 0111111  <-- carry-in = 1
  001110
+ 010011
  ------
  100010
```

The carry in to the most-significant column is different from the carry out of the most-significant column → overflow

Or, equivalently:

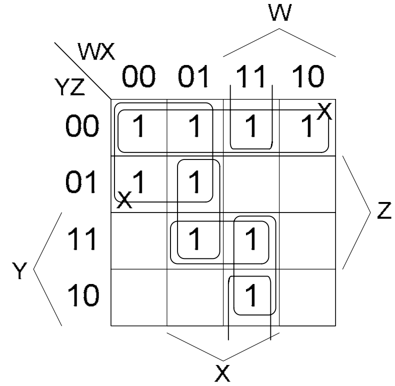
The MSB of the two addends is the same and different from the MSB of the result → overflow

3. (30 Points Total)

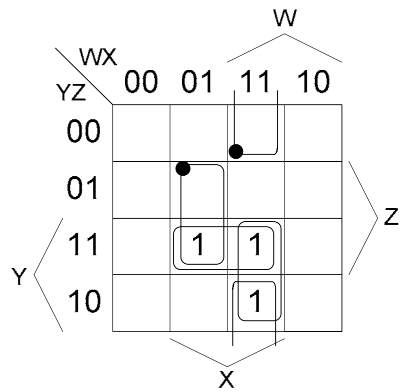
(a: 6 pts) Given the function:

$$F = \sum_{W,X,Y,Z}(0, 1, 4, 5, 7, 8, 12, 14, 15)$$

Represent F on the K-map below and identify all prime implicants. Mark all essential prime implicants with an "**X**".



Redraw the K-map below after the essential prime implicants have been removed. Of the remaining prime implicants, mark any that can be eclipsed with a "•".



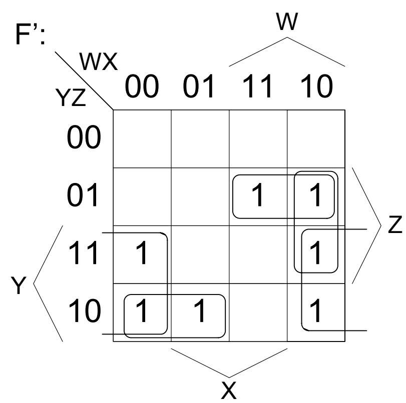
Derive the minimal SOP for F:

F = W'Y' + Y'Z' + XYZ + WXY

(b: 4 pts) Does your minimal SOP implementation contain any potential hazards? If so, identify the input transition(s) that could potentially cause a hazard.

W	**X**	**Y**	**Z**
0	1	↕	1
1	1	↕	0

(c: 6 pts) Derive the minimal POS for F.



Minimal POS:

F' = X'Y + W'YZ' + WY'Z

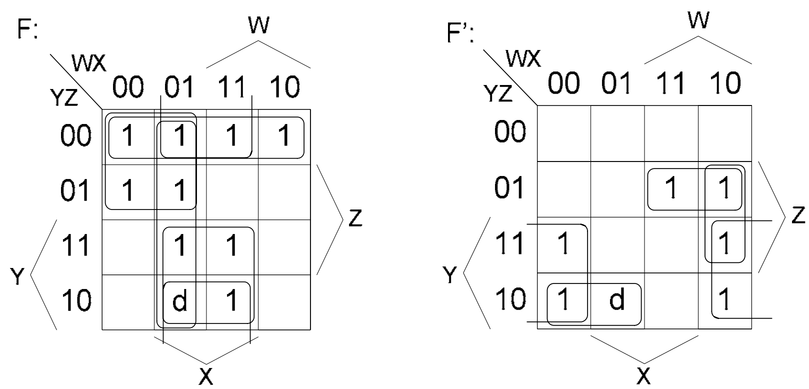
F = (X + Y') · (W + Y' + Z) · (W' + Y + Z')

Which is smaller in terms of number of literals, the SOP or the POS?

SOP – 10 literals
POS – 8 literals
The POS is small than the SOP, in terms of number of literals.

(d: 4 pts) You are now allowed to change any cell (whether it is currently 0 or 1) to a don't care. What is the optimal cell to place the don't care such that you obtain an expression for F (SOP or POS) with the least possible number of literals?

Adding a don't care to cell 6 leads to the following K-maps for F and F':



The new minimal POS is:

$$F = (X + Y') \cdot (W' + Y + Z'),$$

which is the expression with the least number of literals.

(e: 3 pts) Based on where you placed the don't care cell in part 3d, are the minimal SOP and POS for this new function equal? Briefly explain your answer.

The only place that the SOP and POS can potentially differ is in the value of the don't care.

In the new minimal SOP, the don't care was part of a prime implicant that was used in the minimal SOP expression. Therefore the don't care is assigned to a 1 in the minimal SOP.

In the new minimal POS, the don't care will not be part of any prime implicant that will be included in the minimal POS. Therefore, the don't care is assigned to a 1 in the minimal POS.

The don't care is assigned to a 1 in both the new minimal SOP and the new minimal POS, therefore the minimal SOP and POS for this new function are equal.

(f: 7 pts) Find a function in four variables,

$$F = \sum_{W,X,Y,Z}(?, ?, \ldots)$$

such that the following conditions on F are met:

- The minimal SOP for F is equal to the canonical sum for F.
- The minimal POS for F is equal to the canonical product for F.

The first condition implies that we will not be able to combine any terms from the canonical sum, i.e., every prime implicant in the minimal SOP must be a 1-cell prime implicant.

The second condition implies that we will not be able to combine any terms from the canonical product, i.e., every prime implicant in the minimal POS must be a 1-cell prime implicant.

Thus, once a single 1 or 0 is placed on the K-map, all adjacent cells must be given the complementary value to ensure that no combining can take place. Hence, the values of all other cells are determined. The K-map for the 2 functions satisfying these conditions is shown below:

YZ \ WX	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

YZ \ WX	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$F = \sum_{W,X,Y,Z}(0, 3, 5, 6, 9, 10, 12, 15)$$

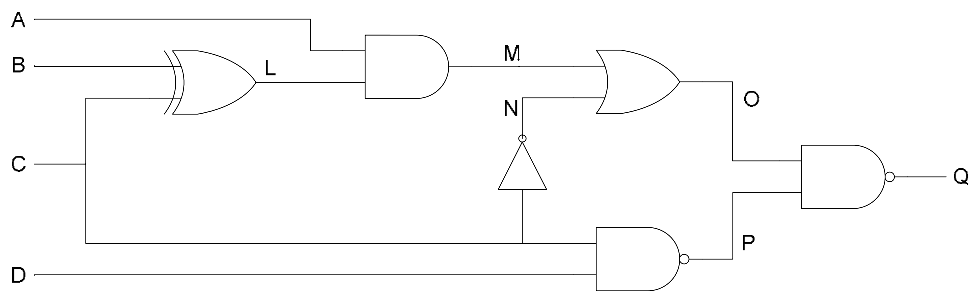
or

$$F = \sum_{W,X,Y,Z}(1, 2, 4, 7, 8, 11, 13, 14)$$

Given a function in *n* variables, how many functions that meet the above conditions exist?

No matter how many variables a particular function has, there will always be only 2 functions satisfying the above conditions.

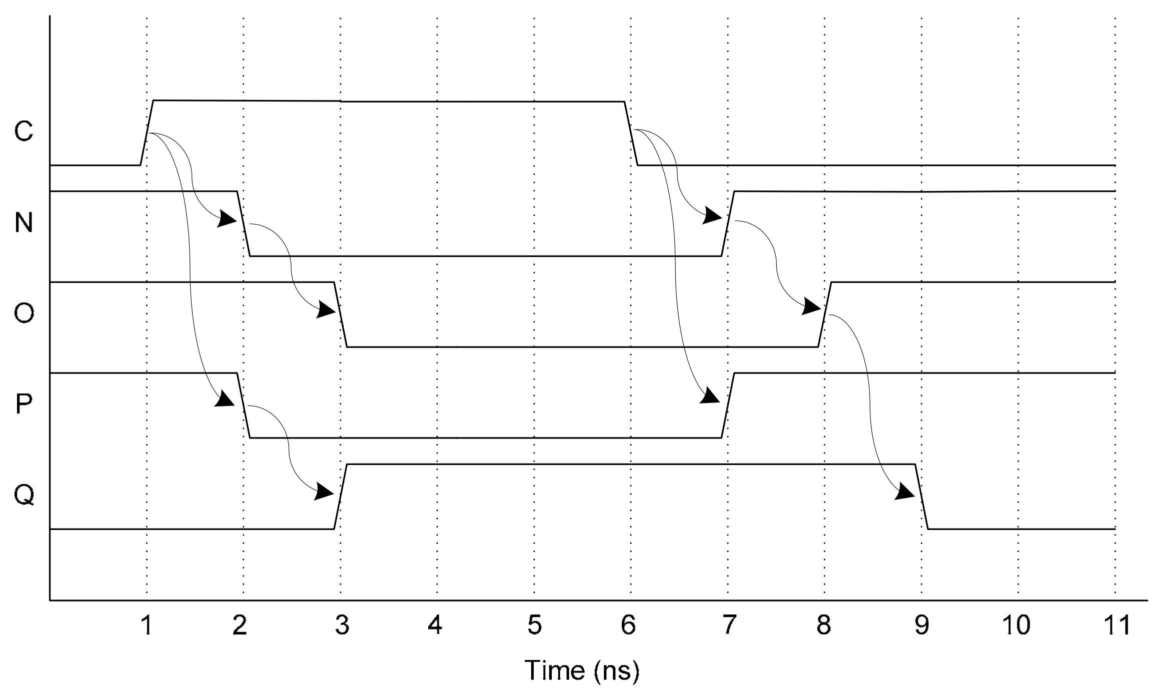
4. (12 Total Points) Consider the circuit below where each gate has a rising and falling delay of 1ns:



(a: 6 pts) Compute the delay $t_{pHL}^{C\rightarrow Q}$ and $t_{pLH}^{C\rightarrow Q}$ under the following input combination:

A = 0, B = 1, D = 1

Show your work on the timing diagram below, including all causality arrows.

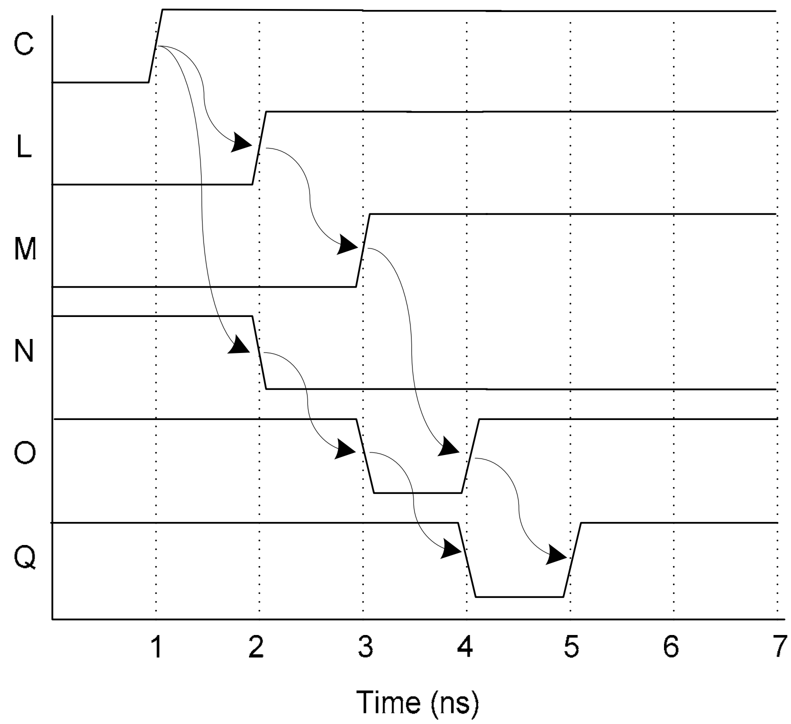


$t_{pLH}^{C\rightarrow Q} = 2\text{ns}$ $t_{pHL}^{C\rightarrow Q} = 3\text{ns}$

(b: 6 pts) Is this circuit hazard free? If yes, explain why it is hazard free. If no, indicate which input must be switched and in which direction (rising/falling) and what the fixed input state of the other inputs must be.

This circuit is not hazard-free. A particular input combination that will cause a hazard is:

A	B	C	D
1	0	↑	0



5. (17 Points Total) Given the three valid codes from a 15-bit error detection/correction code:

010100000101001
001011001000010
100000110010100

(a: 3 pts) What is the hamming distance of this code?

To get from one valid code word to another, 10 bits must be changed. Thus, the hamming distance of this code is 10.

(b: 4 pts) If you are interested only in detecting errors in a particular channel, what is the maximum number of errors that can exist in the channel, when using this code?

$$HD = 2c + d' + 1; \quad d = c + d'$$

Possible solutions

c = 0, d' = 9, d = 9
c = 1, d' = 7, d = 8
c = 2, d' = 5, d = 7
c = 3, d' = 3, d = 6
c = 4, d' = 1, d = 5

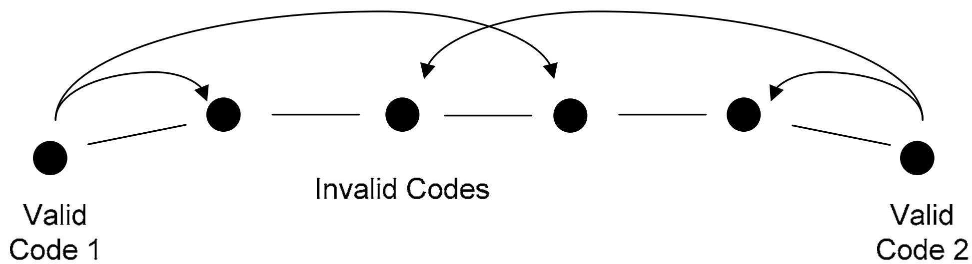
The maximum number of bit errors that can exist in a channel such that all bit errors can be detected with this code is d = 9.

If you are interested in correcting all errors that could occur in a particular channel, what is the maximum number of errors that can exist in the channel?

The maximum number of bit errors that can exist in a channel such that all bit errors can be corrected with this code is c = 4.

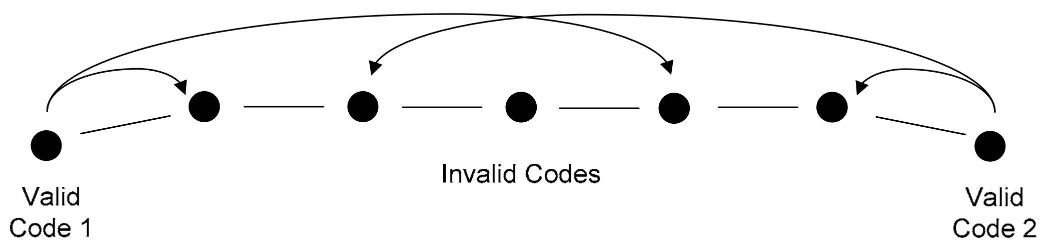
(c: 5 pts) Consider a very unusual channel in which either exactly 1 bit is flipped or exactly *n* bits are flipped (where $n \geq 3$). What is the minimum required hamming distance to both detect and correct the errors in this channel?

Case when $n = 3$:



A hamming distance of 5 will allow us to detect and correct all errors in this channel.

Case when $n = 4$:



A hamming distance of 6 will allow us to detect and correct all errors in this channel.

These examples generalize to larger values of *n*. Therefore, we will need a hamming distance of

$n + 2$

in order to correct and detect all errors in this channel.

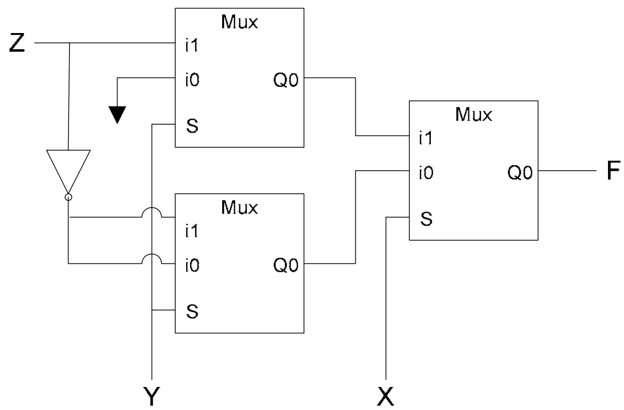
(d: 5 pts) If $n = 3$ for the above channel, what is the minimum hamming distance necessary if we <u>only</u> want to detect errors?

A simple parity code will detect 1 or 3 bit errors (it will detect any odd number of bit errors). Therefore the minimum hamming distance required for the above channel if we only want to detect errors is equal to the hamming distance of a parity code: 2

6. (8 Points Total) Implement the following function using only inverters and two input, single select muxes. For full credit, use as few muxes as possible. Draw your diagram very clearly.

$$F = \sum_{X,Y,Z}(0, 2, 7)$$

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



Reduce: