The University of Michigan
Department of Electrical Engineering and Computer Science

EECS 270 Fall 2002

Exam #1 Solutions

Name:_____ UM ID:_____

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

Problem #	Possible Points	Points Earned
1	16	
2	16	
3	12	
4	20	
5	21	
6	15	
Total	100	

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

Signed: _____

1. (16 Points Total) Reduce the following expressions using any of the theorems or axioms of switching algebra. Note which theorems/axioms you are using.

(a: 8 pts) A·C·E + A·C·D′ + A·C·E′ + A·B·D′·E + B·C′·E

Combining (A·C·E + A·C·E′ = A·C)
A·C + A·C·D′ + A·B·D′·E + B·C′·E

Consensus (X·Y + X′·Z + Y·Z = X·Y + X′·Z;
where X = C′, Y = B·E, and Z = A·D′)
A·C + B·C′·E + A·C·D′

Covering (A·C + A·C·D′ = A·C)
A·C + B·C′·E

(b: 8 pts) A·B·C·D + A′·E·F + B′·E·F + C·D·E·F

Distributivity
A·B·C·D + (A′ + B′)·E·F + C·D·E·F

DeMorgan's
A·B·C·D + (A·B)' ·E·F + C·D·E·F

Consensus (X·Y + X′·Z + Y·Z = X·Y + X′·Z,
where X = A·B, Y = C·D, Z = E·F)
A·B·C·D + (A·B)′ ·E·F

DeMorgan's
A·B·C·D + (A′ + B′)·E·F

Distributivity
A·B·C·D + A′·E·F + B′·E·F

2. (16 Points Total) Perform the following number problems:

(a: 2 pts) $547_8 = ?_{16}$

$5 \quad 4 \quad 7_8$
$= 101\ 100\ 111_2 = 0001\ 0110\ 0111_2$
$= \quad 1 \quad 6 \quad 7_{16} = 167_{16}$

(b: 4 pts) 11001_2 (two's complement representation) $= ?_{10}$

Flip Bits: 00110
Add One: 00111 $=7_{10}$
So 11001_2 (two's complement rep.) $= -7_{10}$

(c: 4 pts) 10000_2 (one's complement representation) $= ?_{10}$

Flip Bits: 01111 $= 15_{10}$
So 10000 (one's complement rep.) $= -15_{10}$

(e: 3 pts) $113.33_{10} = ?_2$ (answer should be accurate to at least 6 binary places; mark any repeating digits)

Integer Part:

2 |113
2 |56 R 1
2 |28 R 0
2 |14 R 0
2 |7 R 0
2 |3 R 1
2 |1 R 1
0 R 1 (MSB)

Fraction Part:

.33 * 2
(MSB) 0 | .66 * 2
1 | .32 * 2
0 | .64 * 2
1 | .28 * 2
0 | .56 * 2
1 | .12 * 2
....

$113.33_{10} = 1110001.010101_2$ (approximate)

(f: 3 pts) In general, what is the minimum number of bits required to approximate a decimal number in binary with an error less than 0.01 ($= {}^1/_{100}$)

| Don't need

.X X X X X X X | X X
$^1/_2$ $^1/_4$ $^1/_8$ $^1/_{16}$ $^1/_{32}$ $^1/_{64}$ $^1/_{128}$ | $^1/_{256}$ $^1/_{512}$

7 bits are required after the binary point to approximate a decimal number with error less than .01

3. (12 Total Points) Perform the following two's complement arithmetic operations in the way a computer would perform them. Indicate if an overflow occurred and why:

(a: 6 pts) (5) add (-18)

```
 Carry      11  0
    5      000101
+(-18)   + 101110
  -13      110011
```

C_{in} and C_{out} of sign bit column are the same → no overflow

or

Sign bits of operands different → no overflow

(b: 6 pts) (-13) sub (+22)

```
 +22        010110
Carry      1   111
Flip(+22)   101001
 -13       +110011
  29        011101
```

C_{in} and C_{out} of sign bit column are different → overflow

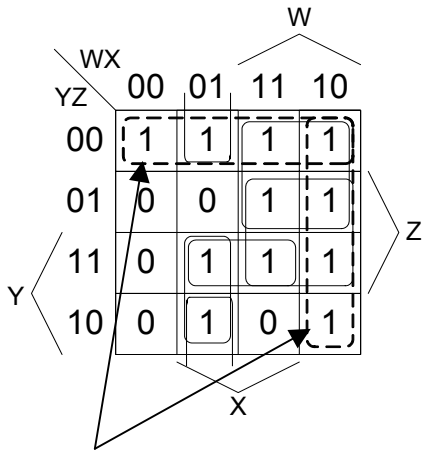
or

Sign bits of operands are same but different from sign bit of result → overflow

4. (20 Total Points) Given the following function:

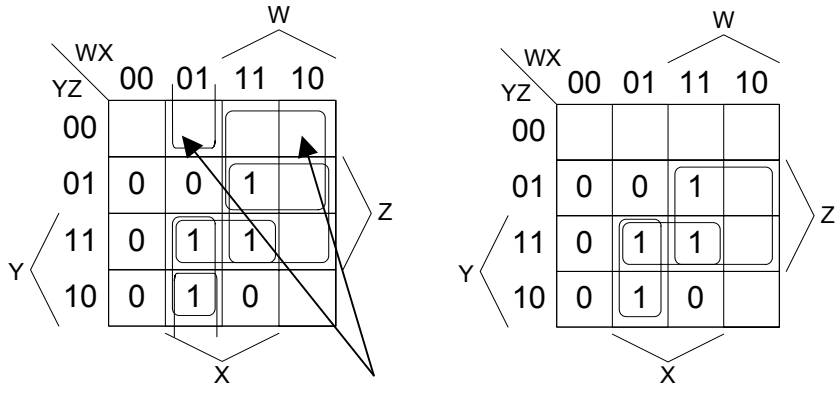
$$F = \sum_{W,X,Y,Z}(0, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15)$$

(a: 4 pts) Construct a K-map and circle all prime implicants.
(b: 3 pts) Mark all essential prime implicants in your K-Map.



Distinguished 1- Cells

(c: 3 pts) Mark which prime implicants can be eclipsed after all essential prime implicants have been removed.
(d: 4 pts) Construct the minimal S.O.P.



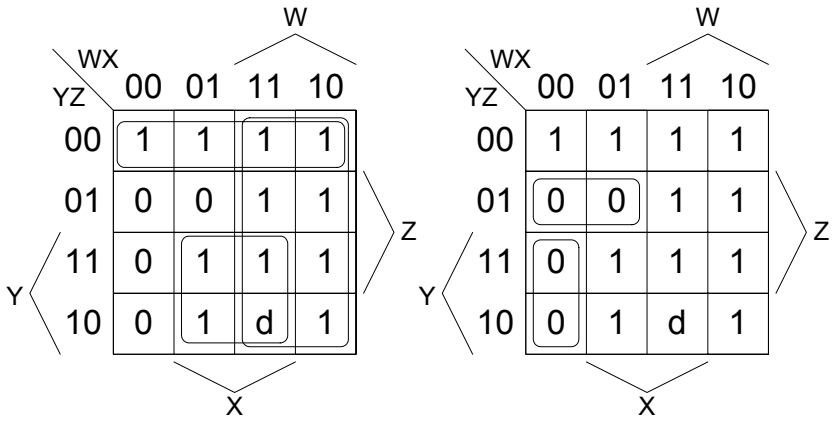
Eclipse

$$F = Y'Z' + W \cdot X' + W \cdot Z + W' \cdot X \cdot Y$$

(e: 6 pts) Add one “don’t care” to the above function, as follows:

$$F = \sum_{W,X,Y,Z}(0, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15) + d(14)$$

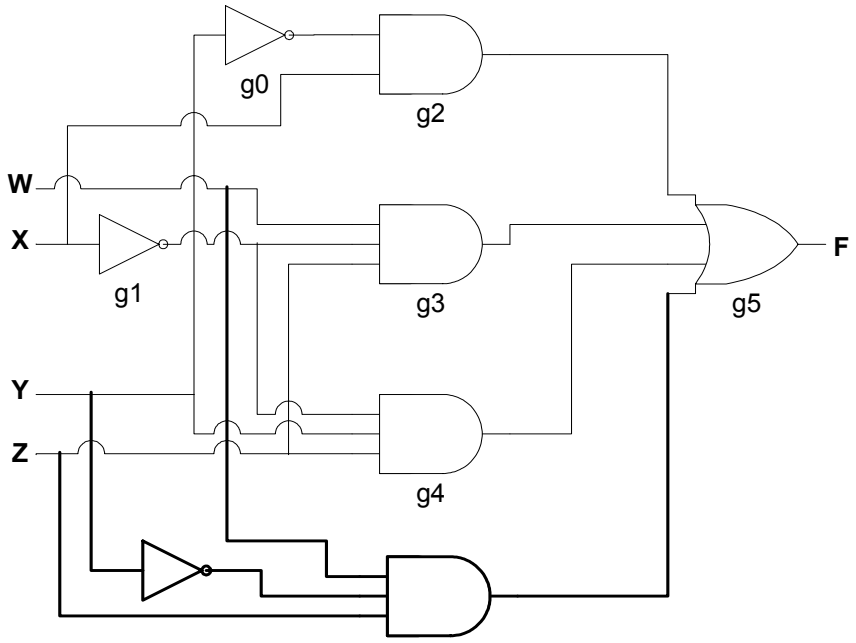
In this case, is the minimal S.O.P equal to the minimal P.O.S for F? Briefly state why.



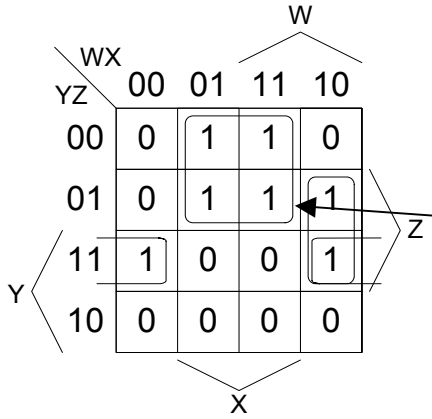
Yes, the functions are the same. The “don’t care” will be included in a PI in the minimal S.O.P., and therefore assigned to 1. It will not be included in a PI in the minimal P.O.S., and therefore also assigned to 1.

5. (21 Total Points) Consider the circuit below, which implements the following function:

$$F = \sum_{W,X,Y,Z}(3, 4, 5, 9, 11, 12, 13)$$



(a: 4 pts) Find all input combinations under which this circuit has a static hazard.



Hazard:

W	X	Y	Z
1	↕	0	1

(b: 3 pts) Is this a static-0 or a static-1 hazard? Static-1 hazard

(c: 3 pts) Draw the additional gate(s) necessary in the circuit above to make sure that the circuit will not glitch.

Add W·Y'·Z product term to above circuit – see above.

(d: 3 pts) Given that the rising and falling propagation delay of each gate g0 – g5 is 1ns, what is the transition direction (0→1 or 1→0) that causes the glitch?

$$1 \rightarrow 0$$

(e: 4 pts) You present your solution to your boss, but he does not want to spend the money for any additional gates in the circuit. Instead, he tells you to fix the circuit by changing the rising and falling delay of gates g1, g2, and/or g3. Write the inequalities in terms of t^{g1}_{pLH}, t^{g1}_{pHL}, t^{g2}_{pLH}, t^{g2}_{pHL}, t^{g3}_{pLH}, t^{g3}_{pHL} that must be satisfied to ensure that the circuit does not glitch under any input transition.

$$t^{g1}_{pLH} + t^{g3}_{pLH} \leq t^{g2}_{pHL}$$

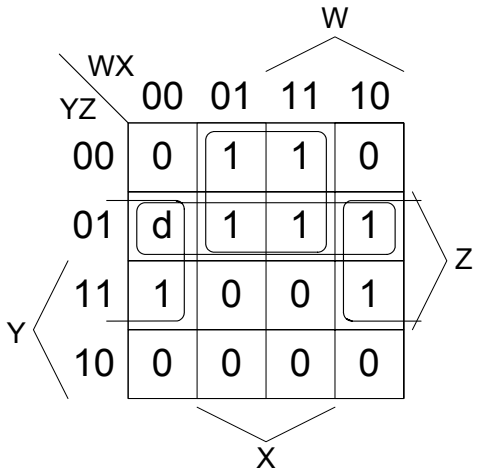
$$t^{g1}_{pHL} + t^{g3}_{pHL} \geq t^{g2}_{pLH}$$

(f: 4 pts) After spending a considerable amount of time on the above problem, your boss changes his mind and requests that you add one "don't care" to the function, such that its minimal S.O.P is hazard-free. In which cell of the K-map must you place the don't care; i.e:

$$F = \sum_{W,X,Y,Z}(3, 4, 5, 9, 11, 12, 13) + d(?)$$

(Show your K-map and circle the P.I.s used in the glitch-free implementation)

For minimal, hazard-free implementation, add d(1):



6. (15 Total Points) Consider a circuit that has three inputs i_2, i_1, i_0 and four outputs o_2, o_1, o_0 and ER. Let i_2, i_1, i_0 be the unsigned binary representation of the number X = {0-7}. For X in the range 0-4, the outputs o_2, o_1, o_0 should be the unsigned binary representation of X+3. For example, if i_2 i_1 i_0 = 011, then o_2 o_1 o_0 should be 110; if i_2 i_1 i_0 = 001, then o_2 o_1 o_0 should be 100, etc. For X equal to 5, 6, or 7, the output o_2 o_1 o_0 must be 000 and ER must be 1. In all other cases, ER must be 0.

Design this circuit using decoders, encoders, AND, OR, and NOT gates. Your implementation should only use three of the above components/gates.