CSE370 Winter 2010 – Exam 1 (27 January 2010)

Please read through the entire examination first! This exam was designed to be completed in 50 minutes and, hopefully, this estimate will be reasonable.

There are 4 problems for a total of 90 points. The point value of each problem is indicated in the table below. Each problem and sub-problem is on a separate sheet of paper. Write your answer neatly in the space provided. If you need more space (you shouldn't), you can write on the back of the sheet where the question is posed, but please make sure that you indicate clearly the problem to which the comments apply. Do NOT use any other paper to hand in your answers. If you have difficulty with part of a problem, move on to the next one. They are mostly independent of each other.

The exam is CLOSED book and CLOSED notes. Please do not ask or provide anything to anyone else in the class during the exam. Make sure to ask clarification questions early so that both you and the others may benefit as much as possible from the answers.

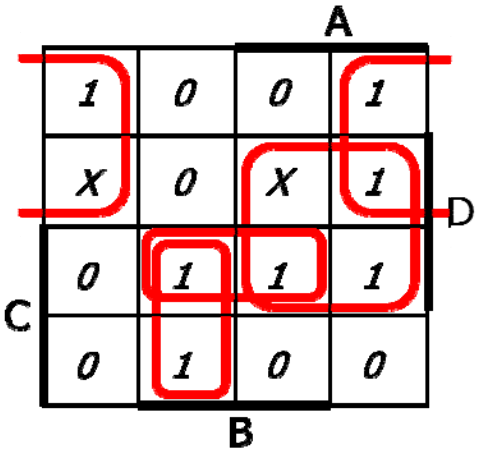
Name:

ID#:

Problem	Max Score	Score
1	20	***20***
2	30	***30***
3	25	***25***
4	15	***15***
TOTAL	**90**	***90***

1. Boolean Simplification (20 points)

(a – 10 pts) Minimize the function F = Σ m(0, 6, 7, 8, 9, 11, 15) + d(1,13) using the K-map below. List all prime implicants, essential prime implicants, and the minimal solution in SOP form in the spaces provided.



Prime implicants:

B'C' AD BCD A'BC

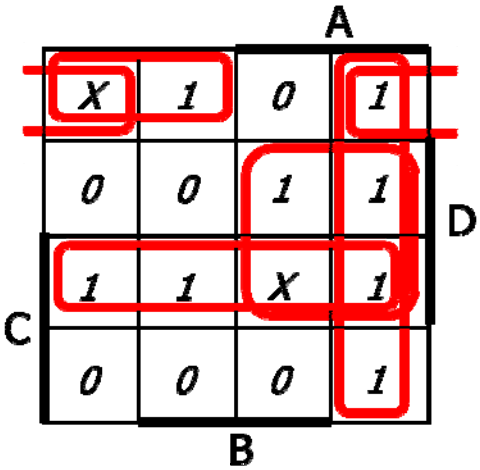
Essential prime implicants:

B'C' AD A'BC

Minimal cover:

F = B'C' + AD + A'BC

(b – 10 pts) Minimize the function G = Π M(1, 2, 5, 6, 12, 14) + D(0, 15) using the K-map below. List all prime implicants, essential prime implicants, and the minimal solution in SOP form in the spaces provided.



Prime implicants:

AB' CD AD B'C'D' A'C'D'

Essential prime implicants:

AB' CD AD A'C'D'

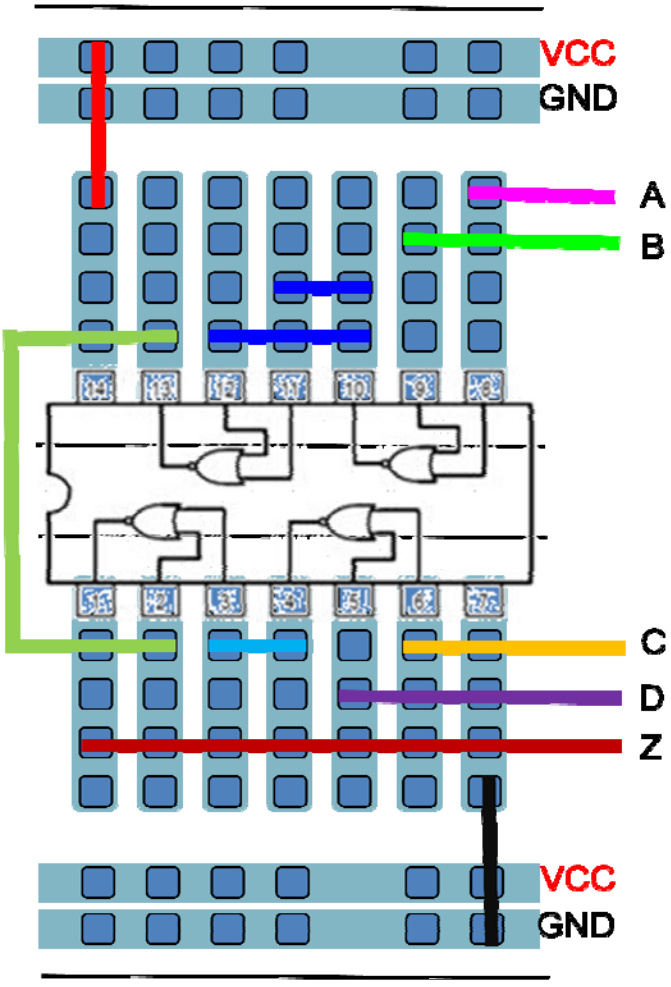
Minimal cover:

G = AB' + CD + AD + A'C'D'

2. Gates to Functions back into Gates (30 points)

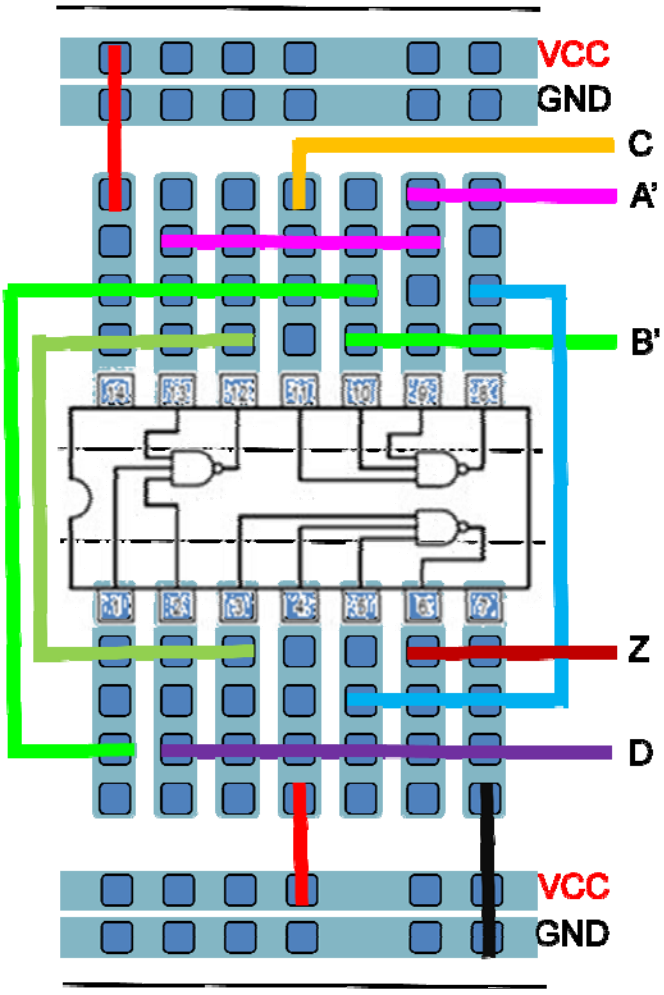
(a – 15pts) Given the following breadboard wiring diagram, derive a SOP equation for the single output, Z, through the use of Boolean algebra The package in the center is a set of 4 NOR gates (the ’02 part in your lab kit).

HINT: deMorgan’s Law is (x + y)’ = x’y’ and its dual (xy)’ = x’ + y’.



Z = [[(A + B)’]’ + (C + D)’]’
Z = (A + B)’(C +D) = A’B’(C + D)
Z = A’B’C + A’B’D

(b – 10pts) Remap the function you derived for Z in part (a) to the package of 3 3-input NAND gates shown below (the ’10 part in your kit). You can assume that A’, B’, C’, and D’ are available as inputs if you should need them. Have the inputs and outputs on the right as in the diagram of part (a).



(c – 5pts) Assuming that ’02 2-input NOR gates have a delay of 7ns and ’10 3-input NAND gates have a delay of 9ns, what is the delay to Z of a change in an input (or its complement) for the circuit in part (a) and the circuit you wired up in part (b).

Delay of circuit of part (a) = ***3 gate delays at 7ns each or 21ns total.***

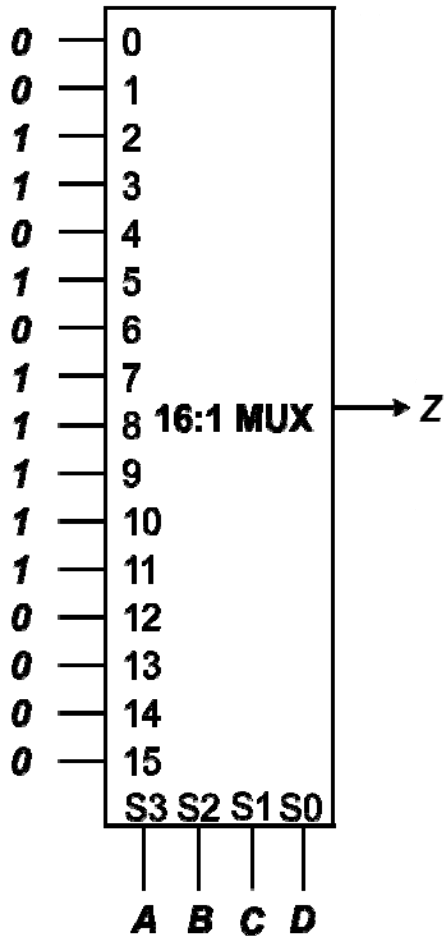
Delay of circuit of part (b) = ***2 gate delays at 9ns each or 18ns total.***

3. Multiplexer Logic (25 points)

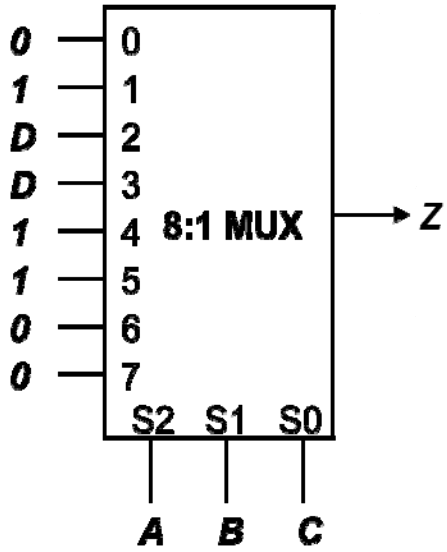
(a – 5pts) Given the function Z = B'C + A'BD + AB', fill in the truth-table below.

A	B	C	D	Z
0	0	0	0	***0***
0	0	0	1	***0***
0	0	1	0	***1***
0	0	1	1	***1***
0	1	0	0	***0***
0	1	0	1	***1***
0	1	1	0	***0***
0	1	1	1	***1***
1	0	0	0	***1***
1	0	0	1	***1***
1	0	1	0	***1***
1	0	1	1	***1***
1	1	0	0	***0***
1	1	0	1	***0***
1	1	1	0	***0***
1	1	1	1	***0***

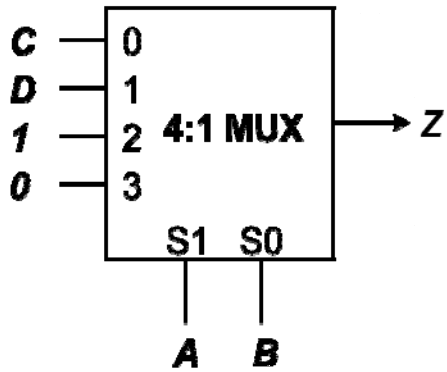
(b – 5 pts) Given the truth table of part (a), implement Z using a single 16:1 multiplexer shown below. Make sure to clearly label all inputs and outputs.



(c – 5 pts) Given the truth table of part (a), implement Z using a single 8:1 multiplexer shown below. Try to use as few other gates as possible. Make sure to clearly label all inputs and outputs.



(d – 10 pts) Given the truth table of part (a), implement Z using a single 4:1 multiplexer shown below. Try to use as few other gates as possible. Make sure to clearly label all inputs and outputs.



4. Decoder Logic (15 points)

(a – 5pts) Given the function W = (A xor C) + B, fill in the truth-table below.

A	B	C	W
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	*1*

(b – 10 pts) Given the truth table of part (a), implement W using a single 3:8 decoder shown below and no more than a single 2-input NOR gate. Make sure to clearly label all inputs and outputs.