

ECE 199 Exam 1 Spring 2003

Tuesday, February 25th, 2003

Name:

- **Be sure your exam booklet has 7 pages.**
- **Write your name at the top of each page.**
- **This is a closed book exam.**
- **You are allowed one 8.5 x 11 sheet of notes.**
- **Absolutely no interaction between students is allowed.**
- **Show all of your work.**
- **Challenge questions are marked with a *****
- **Don't panic, and good luck!**

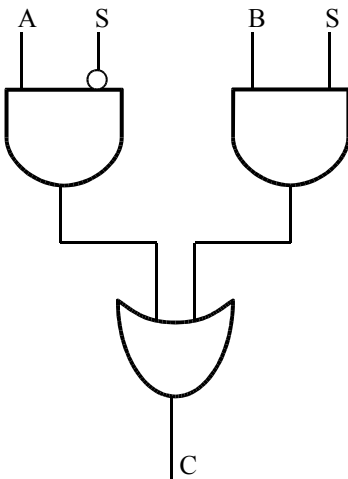
Problem 1	20 points	_____
Problem 2	20 points	_____
Problem 3	20 points	_____
Problem 4	20 points	_____
Problem 5	20 points	_____
Total	100 points	_____

Name: _____

Problem 1 (20 points): Short Answer

Answer the following questions with a **very brief and precise** statement. A long statement is a good indication that your response will be marked incorrect.

Part A (8 points): We have mentioned that any logic function can be implemented using only NAND gates. Given this logic diagram of a 2-to-1 multiplexer below, give an equivalent circuit using **only** NAND gates. You do not have any other types of gates at your disposal, not even inverters. (Hint: use DeMorgan's Theorem!)



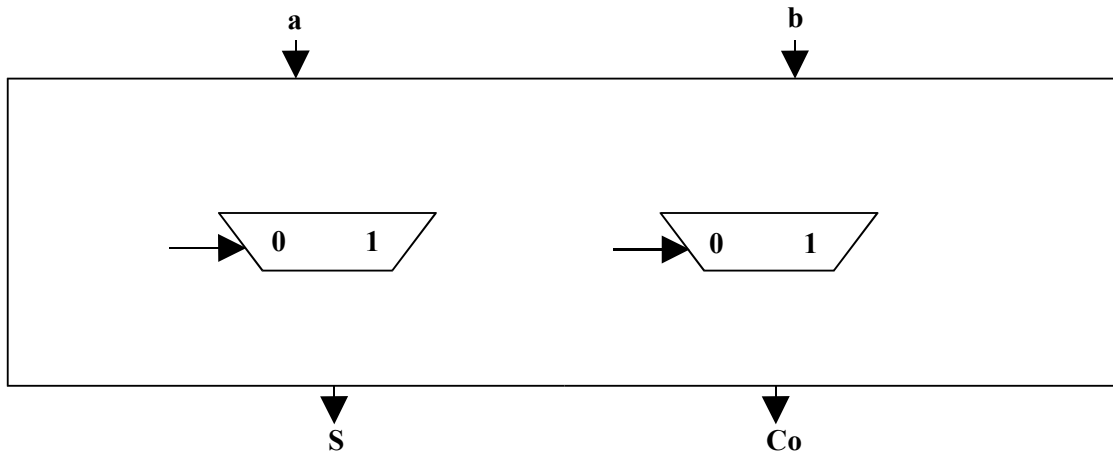
*****Part B** (5 points): The LC-4 is a new version of the LC-2 where the NOT instruction is replaced by an MINUS instruction. The MINUS instruction takes one input register operand and writes its additive inverse into the destination register. For example, MINUS R3, R2 will put $-R2$ into R3. Demonstrate that the LC-4 is computationally complete. That is, show that it can compute anything that the LC-2 can.

Part C (7 points): We represent signed integer numbers using 2's complement representation. We could instead represent negative numbers using 1's complement: a negative number is formed by taking the bitwise NOT of the positive representation. For example, 0101 is equal to 5 and 1010 is equal to -5 in 4-bit 1's complement. What range of values can we represent with 4 bits in 1's complement?

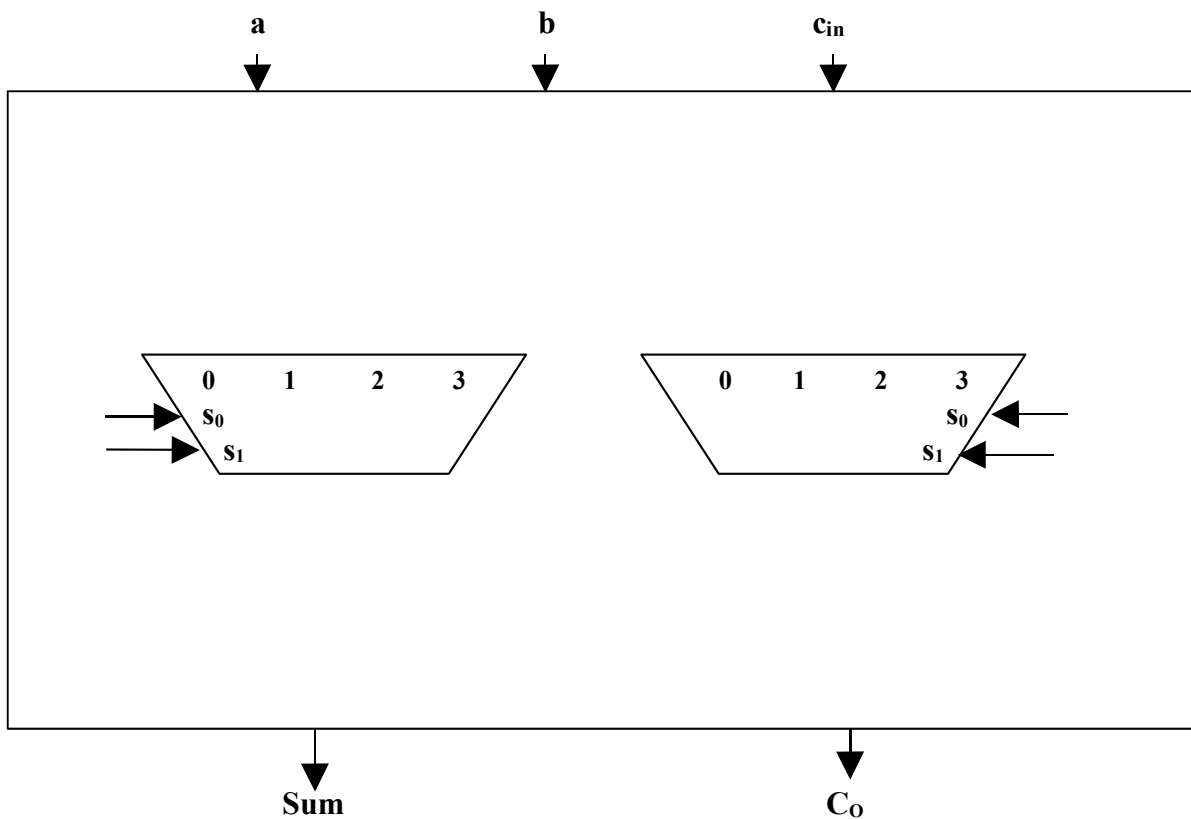
Name: _____

Problem 2 (20 points): Logic

Part A (10 points): Design a 1-bit half adder (i.e., no carry in) using only two 2-to-1 multiplexers and inverters as components. Inputs to the adder are **a** and **b**, both of which are one bit signals.



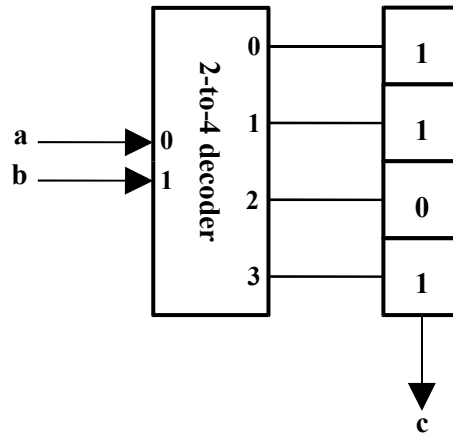
*****Part B** (5 points): Now design a 1-bit full adder using only inverters and the two 4-to-1 MUXs shown below. The inputs are **a**, **b**, and **c_{in}**, all of which are one bit signals.



Name: _____

Problem 2, continued:

Part C (5 points): The following logic diagram consists of a decoder and a 4-bit memory. The memory is read-only, and cannot be written. This logic circuit can be implemented in a much simpler fashion. Provide the simpler implementation



Name: _____

Problem 3 (20 points): LC-2 Programming

Part A (12 points): The code below is an incomplete program to multiply two positive integers in R1 and R2 respectively. The result is written into R0. This code is missing the very first instruction in x3000 and part of the instruction in x3004. Provide these missing parts.

x3000		
x3001	0001 0010 0111 1111	ADD R1 ← R1, #-1
x3002	0000 1000 0000 0101	BRn x005
x3003	0001 0000 1000 0000	ADD R0 ← R2, R0
x3004	0000 ____0 0000 0001	BR____ x001
x3005	1111 0000 0010 0101	HALT

Part B (8 points): Assuming the code above is completed to correctly calculate $R1 * R2$, what function would be calculated if the instruction in x3003 were replaced with the following. Phrase your answer in terms of R1 and R2.

a) $ADD R0 \leftarrow R2, R2$

b) $ADD R0 \leftarrow R0, \#1$

c) $ADD R0 \leftarrow R0 + R0$

Name: _____

Problem 4 (20 points): The von Neumann Model and the LC-2

For this question, you will need to provide all the values that the **MAR** and **MDR** transition through while processing the four instructions below. Assume the first instruction (**LEA**) has not yet been fetched, and that the initial values of the **MAR** and **MDR** are **x3050** and **x0E06**. The contents of memory locations x3050-x3054 are also provided.

```
...
x3006 - LEA R0, x51
x3007 - LDI R1, x51
x3008 - ADD R2, R1, R0
x3009 - STI R2, x52
...
...
x3050 - x0E06
x3051 - x3053
x3052 - x3054
x3053 - x0005
x3054 - x000A
...
```

MAR: x3050,

MDR: x0E06,

Name: _____

Problem 5 (20 points): Programming

The code below sums the elements of **10** consecutive memory locations starting at **x3051**, and stores the sum into memory at **x3050**.

```
x3000 - LEA R0, x51
x3001 - AND R7, R7, #0
x3002 - ADD R1, R7, #10
x3003 - LDR R6, R0, #0
x3004 - ADD R7, R7, R6
x3005 - ADD R0, R0, #1
x3006 - ADD R1, R1, #-1
x3007 - BRp x03
x3008 - ST R7, x50
x3009 - HALT
```

Part A (10 points): Assume that each instruction will take exactly 6 machine cycles to execute, due to the six phases of the von Neumann instruction cycle. Calculate the total number of cycles required to complete the program above.

Part B (10 points): Now assume that every memory access (read or write) requires 100 machine cycles (in addition to the 6 cycles that an instruction normally takes.) Calculate the number of machine cycles it takes to execute the program above.