ECE 190 Exam I Fall 2005

Tuesday, September 27th, 2005

Name:

- **Be sure your exam booklet has 12 pages.**
- **Write your name at the top of each page.**
- **This is a closed book exam.**
- **You may not use a calculator.**
- **You are allowed one handwritten 8.5 x 11” sheet of notes.**
- **Absolutely no interaction between students is allowed.**
- **Show all of your work.**
- **Be sure to clearly indicate any assumptions that you make.**
- **More challenging questions are marked with a ***.**
- **Don’t panic, and good luck!**

“A professor is one who talks in someone else’s sleep.” – W. H. Auden

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

Part A (5 points): The IR (Instruction Register) holds the instruction that is to be executed. Given that the instruction bits can be held in the MDR, why is an IR necessary?

Part B (5 points): Consider the following LC-3 instruction (x3500 is the address at which the instruction is located):

x3500 LD R5, _____ ; we want to put the value x2BFF in register R5

Given the above instruction, what is the range of memory addresses at which the value x2BFF can be stored such that the above instruction can be executed successfully?

Part C (5 points): A certain memory chip has a total of 2^{32} bits and is 8-bit addressable. How many address bits must be specified when reading or writing a location on this chip?

Part D (5 points): What fraction of the range of numbers that can be represented with an N-bit 2's complement data type can also be represented with an (N+1)-bit unsigned data type? (Justify your answer.)

 Name: ____

Problem 2 (20 Points): Representations

Part A (7 points): Your friend complains to you that the number 1,073,741,825 (in hexadecimal, x40000001) cannot be represented using an IEEE single-precision floating point representation (1-bit sign, 8-bit exponent, 23-bit mantissa). Is your friend right?

If so, why would one ever use floating point, given that 32-bit 2's complement can represent the given number? If not, give the floating point representation (the bits for each field).

Part B (6 points): Suppose we were to build a finite state machine that takes in a string of bits and determines whether the number of 1s in the string is odd or even. The length of string is arbitrary. What is the least number of states required to build this state machine? How many bits are required to represent the states?

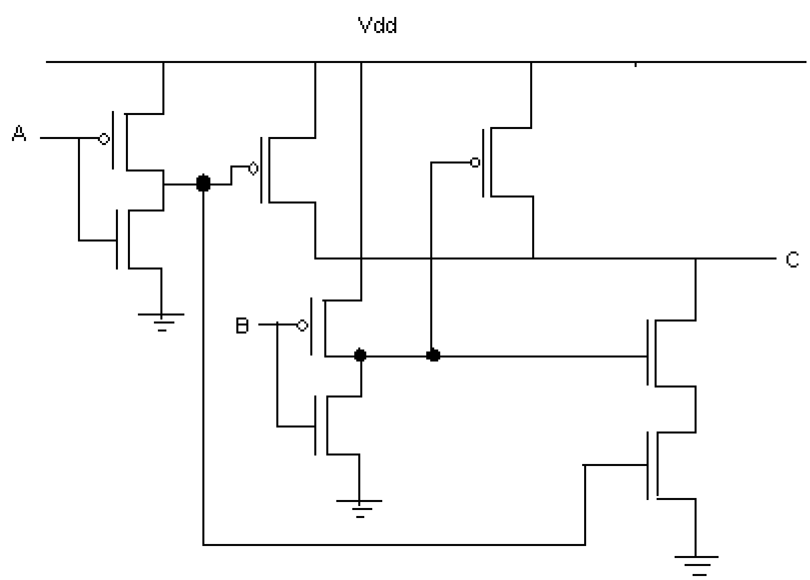
Part C (7 points): UIUC has 35,000 students and offers N classes in a given semester. To represent a student's class list with a bit vector, we need N bits.

Describe a more efficient representation to record a student's class list, stating any necessary assumptions. Write an expression for the number of bits necessary using your representation (in terms of N), then write an inequality (also in terms of N) specifying when your representation requires fewer bits than a bit vector.

 Name: ____

Problem 3 (20 points): Combinational Logic

Part A (5 points): Write an expression for the function implemented by the CMOS circuit shown below.

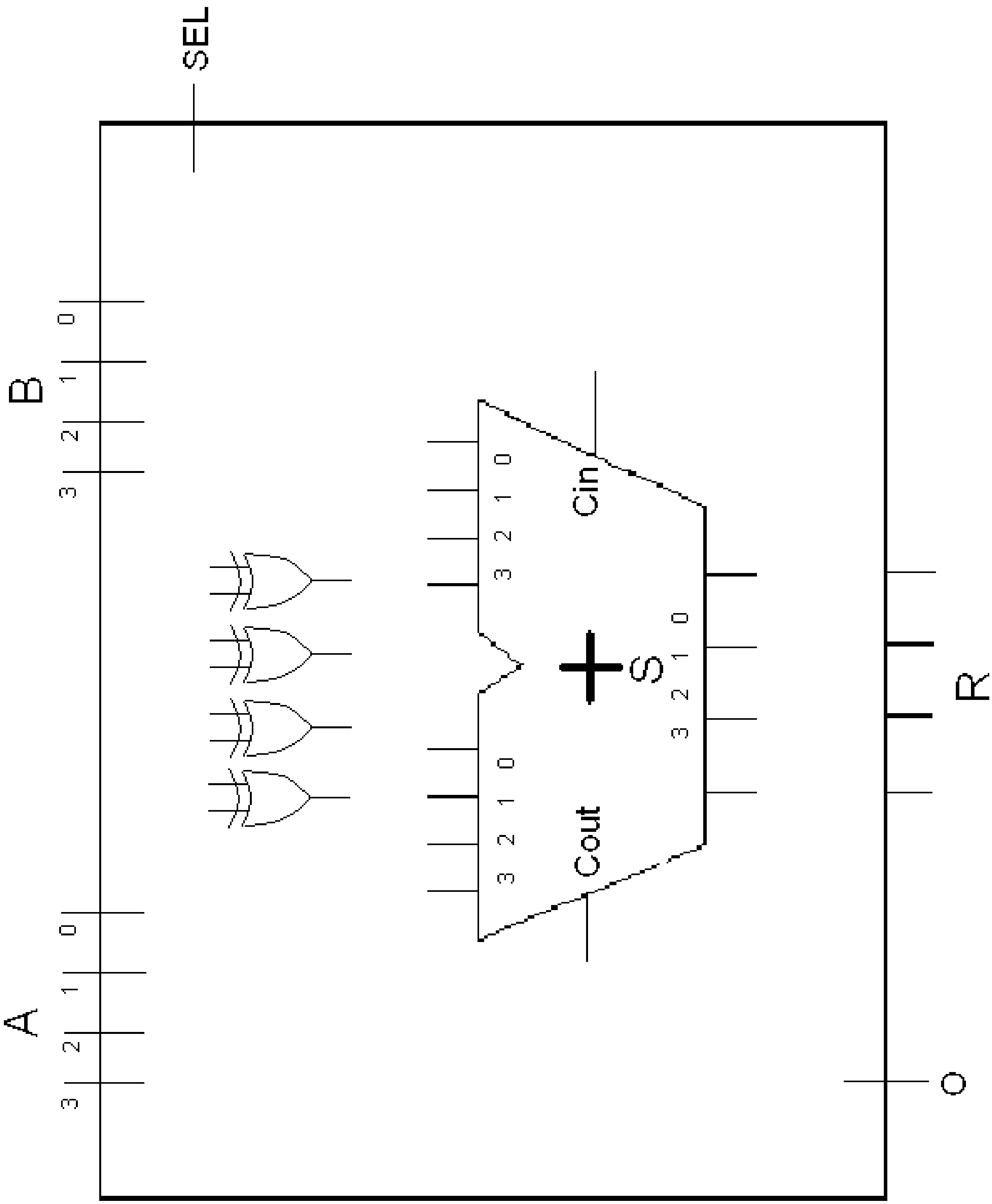


Parts B and C refer to the circuit on the next page. The values A, B and R are in unsigned binary representation.

Part B (12 points): Connect the wires in the diagram in such a way that the four-bit value R is equal to A + B when SEL=0, and A – B when SEL=1. You may not add any additional gates. Ignore the output O for this part.

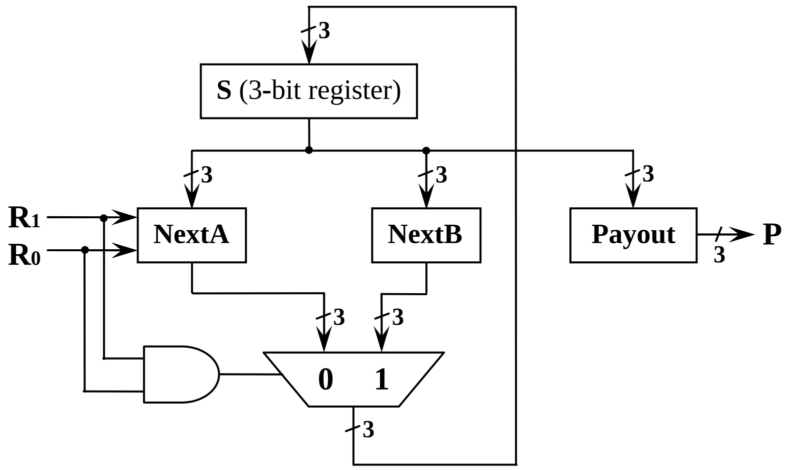
***__Part C__** (3 points). Extend your answer to part B to generate output O=1 if an *unsigned addition* overflows or an *unsigned subtraction* underflows, and O=0 otherwise. You may use a single additional gate with either one or two inputs.

 Name: ____



Problem 4 (20 points): Finite State Machines

As part of your job with the IRS, you are responsible for validating the accuracy of a certain casino's claims about the amount paid out by their new slot machine model. Each time a gambler inserts a coin, the machine randomly chooses one of four fruits—orange, peach, cherry, or lemon. A 3-bit FSM makes a transition based on the fruit chosen (represented with 2 bits), and the new FSM state specifies a payout in coins (a 3-bit unsigned value). A high-level diagram of the FSM appears below.

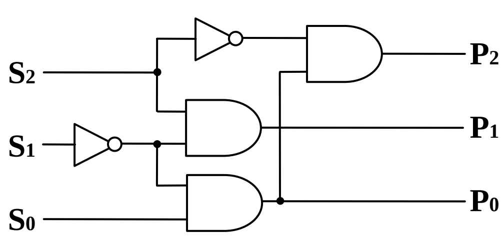


The inputs R_1 and R_0 encode the fruit selected as follows:
orange: R_1R_0=00 peach: R_1R_0=01 cherry: R_1R_0=10 lemon: R_1R_0=11

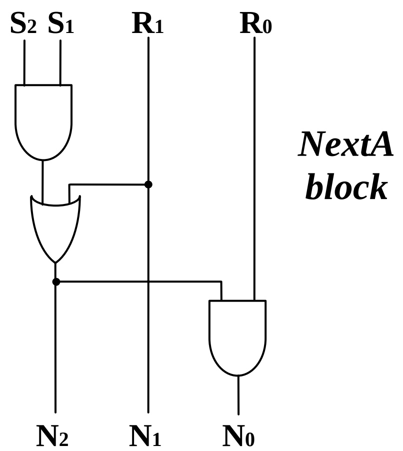
Part A (2 points): Which fruits use the NextA block to determine the FSM's next state?

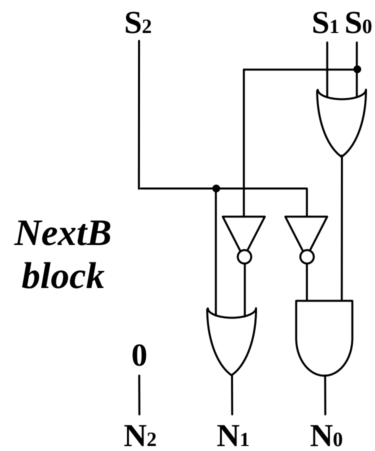
Part B (4 points): Based on the payout component shown below, fill in the table with the number of coins $P_2P_1P_0$ paid for each FSM state.

S_2	S_1	S_0	P_2	P_1	P_0
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



 Name: ______________________________________





The NextA and NextB blocks are shown above. For both blocks, the S input is the current FSM state, the R input is the current fruit, and the N output is the next FSM state.

Part D (1 point): Find the next FSM state $N_2N_1N_0$ when a cherry is seen (R_1R_0=10).

Part E (2 points): The next FSM state takes one of two values when an orange is seen (R_1R_0=00). Fill in the table below specifying the next states as a function of the current state, using X to represent irrelevant inputs.

S_2 S_1 S_0	N_2 N_1 N_0
else	

Part F (2 points): The next FSM state takes one of two values when a peach is seen (R_1R_0=01). Fill in the table below specifying the next states as a function of the current state, using X to represent irrelevant inputs.

S_2 S_1 S_0	N_2 N_1 N_0
else	

Part G (4 points): Find the next FSM state when a lemon is seen (R_1R_0=11).

S_2 S_1 S_0	N_2 N_1 N_0
0 0 0	
0 0 1	
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

 Name: ____

Part H (5 points)**:** Identify all winning combinations for this state machine and the amount paid for each. A winning combination is a minimal sequence of input combinations that takes the FSM from an arbitrary state into a particular payout state. It is minimal in the sense that no suffix of the combination suffices to reach the same payout state. For example, if the combination “cherry lemon” were such a sequence, neither “lemon cherry lemon” nor “orange cherry lemon” could be, since “cherry lemon” is a suffix of both.

Hint: work backwards from each state with non-zero payout.

 Name: ____

Problem 5 (20 points): The LC-3 Instruction Set

Parts A and B refer to the LC-3 code below (execution starts at x3000 and ends when the HALT trap is decoded).

Address	**Contents**	**Instruction**
x3000	1110 000000000101	LEA R0, #5
x3001	0110 010000000001	LDR R2, R0, #1
x3002	0111 010000000000	STR R2, R0, #0
x3003	1010 011000000001	LDI R3, #1
x3004	1111 000000100101	TRAP x25 (HALT)
x3005	0011 000000001000	*not executed*
x3006	1011 101010101101	*not executed*
x3007	0110 000000001100	*not executed*
x3008	0011 010101101010	*not executed*

Part A (4 points): How are the contents of registers and the memory locations shown above modified by the code when it runs? Specify the final value stored at each memory location and register changed by the code.

Part B (4 points): You also need to figure out how long this program takes to execute. Knowing that the majority of the time spent will be on memory accesses, you decide to count the number of memory accesses to estimate the time.

How many times does LC-3 need to access memory to execute this snippet of code (until decode of the TRAP)? Justify your answer.

Part C (8 points): Translate the code below into assembly language or RTL. For any PC-relative addresses, perform the calculation and write the resulting address rather than writing "PC + …"

Address	Data	Instruction (Assembly or RTL)
x4013	0000 001111111001	
x4014	0001 101010100000	
x4015	0101 011010100011	
x4016	0001 100100111111	
x4017	0000 101111111000	
x4018	1111 000000110101	

***** Part D** (4 points): Explain what the code below does. A description that requires more than a few words is a good hint that you have the wrong idea. *Hint: R0 and R1 are positive inputs, and R5 is the output.*

Address	Data	Instruction
x3000	0101 010000100000	AND R2, R0, #0
x3001	0001 101010100001	ADD R5, R2, #1
x3002	0001 011000100000	ADD R3, R0, #0
x3003	0001 010010000101	ADD R2, R2, R5
x3004	0001 011011111111	ADD R3, R3, #-1
x3005	0000 001111111101	BRp #-3
x3006	0001 101010100000	ADD R5, R2, #0
x3007	0101 010010100000	AND R2, R2, #0
x3008	0001 001001111111	ADD R1, R1, #-1
x3009	0000 001111111000	BRp #-8
x300A	1111 000000100101	TRAP x25

 Name: ____

Use this page for scratchwork.

 Name: ____

	15 14 13 12	11	10	9	8 7 6	5	4 3	2 1 0
ADD$^+$	0001	DR			SR1	0	00	SR2
ADD$^+$	0001	DR			SR1	1	imm5	
AND$^+$	0101	DR			SR1	0	00	SR2
AND$^+$	0101	DR			SR1	1	imm5	
BR	0000	n	z	p	PCoffset9			
JMP	1100	000			BaseR	000000		
JSR	0100	1	PCoffset11					
JSRR	0100	0	00		BaseR	000000		
LD$^+$	0010	DR			PCoffset9			
LDI$^+$	1010	DR			PCoffset9			
LDR$^+$	0110	DR			BaseR	offset6		
LEA$^+$	1110	DR			PCoffset9			
NOT$^+$	1001	DR			SR	111111		
RET	1100	000			111	000000		
RTI	1000	000000000000						
ST	0011	SR			PCoffset9			
STI	1011	SR			PCoffset9			
STR	0111	SR			BaseR	offset6		
TRAP	1111	0000			trapvect8			
reserved	1101							

Figure A.2 Format of the entire LC-3 instruction set. Note: + indicates instructions that modify condition codes