EEC 180A Final Exam

Fall 2004

December 17, 2004

This examination is closed book and closed notes, and no calculators are allowed. The only permitted help is a single page of notes.

For all problems, state any assumptions you make, show all work, and clearly mark your answers. Correct but unclear or ambiguous answers will not receive full credit.

Excerpts from the UC Davis Code of Academic Conduct state:

1. Each student should act with personal honesty at all times.
2. Each student should act with fairness to others in the class. This means, for example, that when taking an examination, students should not seek an unfair advantage over classmates through cheating or other dishonest behavior.
3. Students should take group as well as individual responsibility for honorable behavior. This includes notifying the instructor or TA if you observe cheating.

I understand the honor code and agree to be bound by it.

Signature _____

Name (printed) _____

Lab Section Number _____

Are you taking ECS 154A? Circle one: 1) This quarter, 2) past quarter, 3) in the future

Total points: 100

1. **[6 points]** Binary Arithmetic and Number Systems. Show all work.

a) [1 point] Write -17_{10} as a 6-bit 2's complement number.

$17_{10} = 010001_2$

$-17_{10} = 101111_2$

b) [1 point] Is it possible to write -17_{10} as a 5 bit 2's complement number?

No. The most negative 5 bit 2's complement number is $-16_{10} = 10000_2$.

c) [2 points] Multiply 0101_2 and 0110_2 and write the product as a hexadecimal number.

$0101_2 \times 0110_2 = 11110_2 = 1E_{16}$

d) [2 points] Subtract 7_{10} from -3_{10} using addition of 5 bit 2's complement binary numbers.

$7_{10} = 00111_2$ $\quad -7_{10} = 11001_2$ $\quad 3_{10} = 00011_2$ $\quad -3_{10} = 11101_2$

$-3_{10} + (-7_{10}) = 11001_2 + 11101_2 = 110110_2 = -10_{10}$

2. [**9 points**] Boolean Algebra. Show your work. Do NOT use Karnaugh maps.

a) [3 points] Write $X \oplus Y \oplus Z$ as a sum-of-products expression with a minimal number of literals.

b) [2 points] Write the simplest sum-of-products expression for:

$(X + Y)(Y + Z)(X' + Z)$

c) [2 points] Write the following expression as a sum of two product terms:

$XWY + YZ + (X' + W')Z$

d) [2 points] Simplify (write the simplest expression for) $X + Y + (X + Y)Z$:

3. [**25 points**] Logic Design. Suppose f is a Boolean function of four variables a, b, c, and d: $f(a,b,c,d) = \Sigma\, m(0, 2, 4, 6, 7, 8, 10, 12)$.

a) [5 points] Using a Karnaugh map, find the minimum sum-of-products expression for f.

b) [5 points] Implement f using a 2-level NAND-NAND circuit. Assume the true and inverse versions of a, b, c, and d are available as inputs.

c) [4 points] Identify any static hazards present in the minimum sum-of-products implementation for f and name the type of hazard.

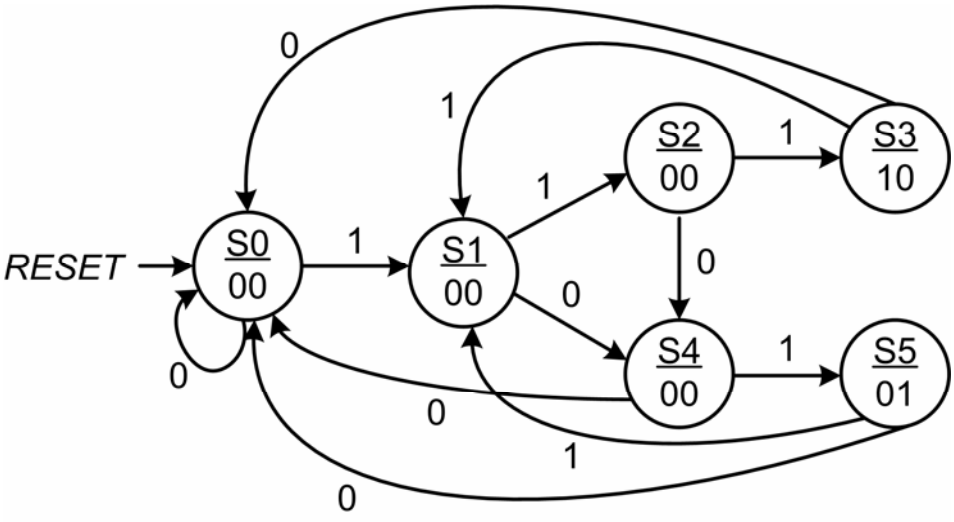
d) [6 points] Draw a timing diagram showing the inputs which cause the static hazard(s) in part c) to occur, and the output.

e) [4 points] Write a new sum-of-products expression for *f* which eliminates any static hazards.

f) [1 point] Suppose *f* represents a next-state equation for a D flip-flop in a Moore FSM. Which expression for f, when implemented as a logic circuit, is **more** likely to cause a setup time violation? Explain your answer.

4. [30 points] Consider the design of a Moore-style FSM with the following description: The FSM has 2 inputs, *X* and *RESET*, and two outputs *Z1*, and *Z0*. After the *RESET* input is asserted the FSM accepts a stream of bits applied to the input *X*. When the input *X* matches the pattern "1-1", the outputs *Z1,Z0* must equal 10 for one cycle if the "-" is a 1, and *Z1,Z0* must equal 01 for one cycle if the "-" is a 0. In all other cases, the outputs must be *Z1,Z0*=00. The *RESET* signal can come at any time and initializes the FSM to again look for bit patterns. Recognizable bit patterns do not overlap. For example, the bit stream (with time moving left to right through this bit string) 0**101**10010**111**0**101** contains only three recognized bit patterns, 101, 111, and 101.

a) [15 points] Draw the state diagram for the Moore FSM. Hint: you need less than eight states.



b) [7 points] Create the symbolic state table and output table below. ("symbolic" means that you do not need to encode the states – just make up state names).

Current State	Next State		Ouputs
	X = 0	X = 1	Z1Z0
S0	S0	S1	00
S1	S4	S2	00
S2	S4	S3	00
S3	S0	S1	10
S4	S0	S5	00
S5	S0	S1	01

Reset always places you in S0. This should be expressed somehow (either in the state table, or as a comment). If they don't handle reset, -2points

Can either have it all in one table, or have two separate tables… The output table would just have state and then output since it is a Moore machine.

c) [8 points] Write the equations for next states and outputs. DO NOT USE ONE-HOT ENCODING FOR STATE ASSIGNMENTS.

Current State	Next State		Ouputs
Q2Q1Q0	X = 0	X = 1	Z1Z0
000	000	001	00
001	100	010	00
010	100	011	00
011	000	001	10
100	000	101	00
101	000	001	01

STATE ASSIGNMENT IS UP TO THEM, so you'll have to check…

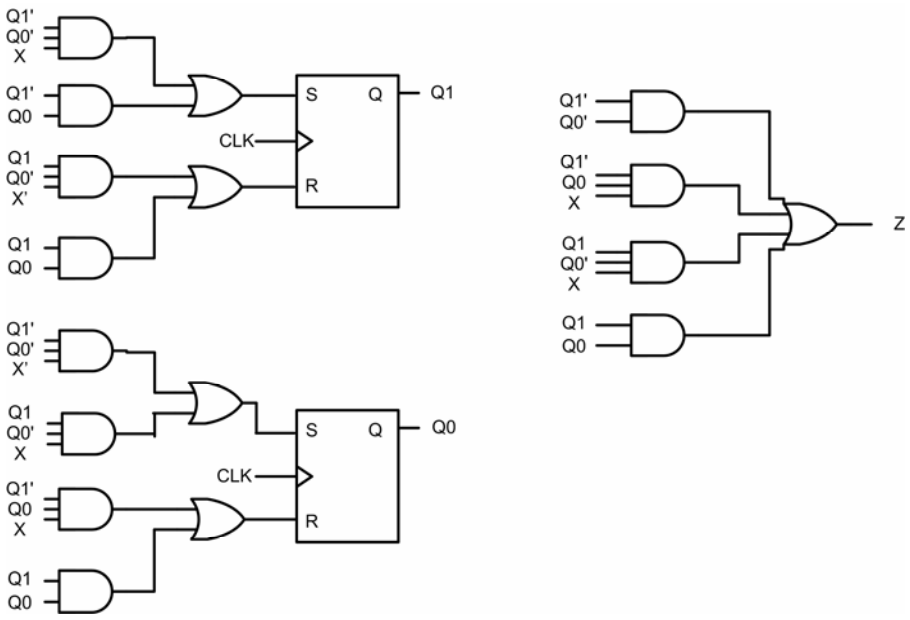
They don't need to have the table, however it is difficult if they don't.

They could also state that reset is attached to the reset of the Flip-flops, and therefore they don't have to include it in their equations.

$Q2^+ = (Q2'Q1'Q0X' + Q2'Q1Q0'X' + Q2Q1'Q0'X)*RESET'$
$Q1^+ = (Q2'Q1'Q0X + Q2'Q1Q0'X)*RESET'$
$Q0^+ = (Q2'Q1'Q0'X + Q2'Q1Q0'X + Q2'Q1Q0X + Q2Q1'Q0'X + Q2Q1'Q0X)*RESET'$
Reduces to
$Q0^+ = (Q2'Q1'Q0'X + Q2'Q1X + Q2Q1'X)*RESET'$

$Z1 = Q2'Q1Q0$
$Z0 = Q2Q1'Q0$

5. [20 points] Use the state machine shown in the figure to answer each part of this question. The state machine has one input X and one output Z.



a) [1 point] Is this machine Mealy or Moore? Mealy

b) [8 points] Write the general next state equations for Q0 and Q1, and the output equation for Z

$Q^+ = S + R'Q$ (just for reference…)

$Q1^+ = Q1'Q0'X + Q1'Q0 + Q1*(Q1Q0'X' + Q1Q0)'$

$Q0^+ = Q1'Q0'X' + Q1Q0'X + Q0*(Q1'Q0X + Q1Q0)'$

$Z = Q1'Q0' + Q1'Q0X + Q1Q0'X + Q1Q0$

c) [7 points] Create the state table for the machine.

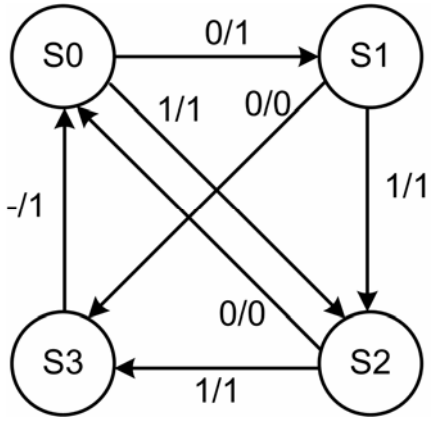
Could also do this symbolically…

Current State	Next State		Output Z	
Q1Q0	X = 0	X = 1	X = 0	X = 1
00	01	10	1	1
01	11	10	0	1
10	00	11	0	1
11	00	00	1	1

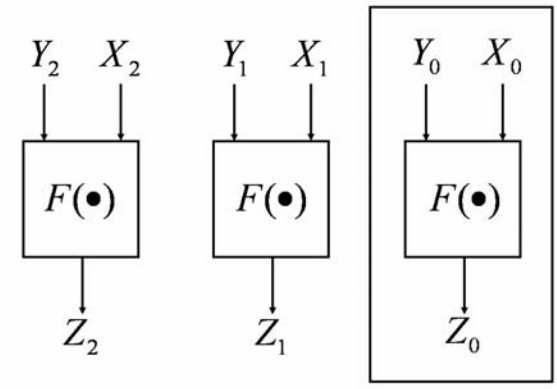
OR

Current State Q1Q0	Input X	Next State $Q1^+Q0^+$	Output Z
00	0	01	1
00	1	10	1
01	0	11	0
01	1	10	1
10	0	00	0
10	1	11	1
11	0	00	1
11	1	00	1

d) [4 points] Draw the state diagram for the machine.



6. [**10 points**] Datapath Design. Design a single bit slice (a circuit which computes a single bit of a multi-bit result) for the LSB for a combined adder/subtractor and divide-by-two (shift right 1 bit) circuit. The box in the figure below highlights an example of a bit slice for a function *F*, which operates on 3-bit inputs to produce a 3-bit output.



The circuit to be designed takes the *0*th bit of two inputs *X* and *Y*, and produces the *0*th bit of the output *Z*, depending on the settings of the two control bits S_0 and S_1. The operation is specified using the following truth table:

S1	S0	Z
0	0	X+Y
0	1	X-Y
1	0	(X+Y)/2
1	1	(X-Y)/2

Label the *0*th bit inputs X_0 and Y_0 and the *0*th bit output Z_0. Use a full adder, a two-to-one multiplexer, and a single two-input logic gate for your implementation. Draw a circuit schematic. Label all control inputs and each input to the full adder and multiplexer.