

ECE212 and ECE212s
Fundamentals of Logic Design
Fall 1996
Tuesday, December 10, 1996
Final Examination

Name (Please Print)

Social Security Number

If you desire to have your grades posted, then you must sign the following statement:

By my signature below, I give Dr. Clay Gloster permission to post my grades for ECE 212 or ECE 212s using my social security number for identification.

Signature

This is a 180-minute open-book test. Please put away all papers except the exam itself and your text. You can use the back of the pages for work space. However, make sure that your answers are placed in the space provided for each problem. Clearly indicate your answers. If you need additional paper, please raise your hand and additional paper will be provided. Be sure to check the point value of each question before beginning. For partial credit, show your work, especially where the answer is just a number. Correct answers with no work shown will receive **NO** credit. (a)

Prob. 1	Prob. 2	Prob. 3	Prob. 4	Prob. 5	Prob. 6	Prob. 7	Prob. 8	Total Points

Name:

1

1. (*Data Shifter - 10 points.*) Draw the schematic for the 4-bit barrel shifter whose function is described in the following truth table. A_3 to A_0 are the primary data inputs and C_1 , C_0 are the control inputs. (C_1 and A_3 are the MSB for C and A respectively.) The four outputs are F_3 to F_0 .

C_1	C_0	F_3	F_2	F_1	F_0	Description
0	0	A_3	A_2	A_1	A_0	No Shift
0	1	A_2	A_1	A_0	A_3	Circular Shift Left One
1	0	A_1	A_0	A_3	A_2	Circular Shift Left Two
1	1	A_0	A_3	A_2	A_1	Circular Shift Left Three

Name:

2

2. (*Laws and Theorems of Boolean Algebra - 10 points.*) Prove that the following Boolean expression is true using the laws and theorems of Boolean algebra found in the text on page 51. Make sure that each step uses only one law and write the law used beside each step. You may use the back of the previous page for scratch paper.

$$A \overline{B} + \overline{A} C + BC = A \overline{B} + C$$

3. (*Combinational Logic - Arithmetic Unit*) - 10 points. Using full adders, multiplexors, and logic gates, design a 4-bit arithmetic unit that functions as shown in the following truth table. The circuit has 8 data inputs, A_3 , A_2 , A_1 , A_0 , and B_3 , B_2 , B_1 , and B_0 . (A_3 and B_3 are the MSBs of the 4-bit numbers A and B.) There are two control inputs that determine the function of the arithmetic unit, C_1 and C_0 . There is a 4-bit output Z_3 , Z_2 , Z_1 , and Z_0 (Z_3 is the MSB.) and an output for carry out, C_{out} . All operations are performed using two's complement numbers.

C_1	C_0	Description
0	0	$Z = A + B$
0	1	$Z = A - B$
1	0	$Z = B - A$
1	1	$Z = -A$

Name:

4

4. (*Multiplexors/Decoders - 10 points.*) Draw schematics for the following truth table mapped into multiplexer/decoder circuits.

A	B	C	D	F_1	F_2
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	0	0

- a. (**5 - points**) Implement the 2-output function using 2, 4:1 multiplexors. You may assume that all literals and their complements are available as inputs.

Name:

5

- b. (5 - points)** Implement the 2-output function using one 4:16 decoder and OR gates.

Name:

6

5. (*PLA Implementation - 10 points.*) Show how to implement the following function using a programmable logic array. Use only groups of two when minimizing outputs to facilitate sharing of product terms. Use the alternate notation for programming found on page 173 of the text.

A	B	C	F_1	F_2	F_3	F_4
0	0	0	0	0	1	1
0	0	1	0	1	1	1
0	1	0	1	0	0	1
0	1	1	1	1	0	1
1	0	0	1	1	0	0
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

6. (*Word Problem - Combinational Circuit - 20 points.*) The list of operations for a combinational circuit are given in the following table, where N_1 and N_2 are two-bit numbers, C_1 , C_0 are control inputs, and N_3 is a 4-bit number. Assume that we assign input variables AB to N_1 , CD to N_2 and ABCD to N_3 . Thus if $N_1=2$, then A=1 and B=0, etc. Also note that when the output should be negative, it is set to zero. The circuit contains one 4-bit output Z . (Z_3 is the MSB.)

C_1	C_0	Function	Comment
0	0	$Z = N_1$ minus N_2	$Z = 0000$ if $N_2 > N_1$
0	1	$Z = N_1$ plus N_2	
1	0	$Z = \overline{N_3}$ plus 1	The Two's Complement of N_3
1	1	$Z = \overline{N_3}$	The One's Complement of N_3

- a. (10 - points) Complete the following truth table.
- b. (10 - points) Write expressions for Z_3 , Z_2 , Z_1 , and Z_0 using little m sum of product notation.

$$Z_3 = \sum m$$

$$Z_2 = \sum m$$

$$Z_1 = \sum m$$

$$Z_0 = \sum m$$

Name:

8

m#	C_1	C_0	A	B	C	D	Z_3	Z_2	Z_1	Z_0
0	0	0	0	0	0	0				
1	0	0	0	0	0	1				
2	0	0	0	0	1	0				
3	0	0	0	0	1	1				
4	0	0	0	1	0	0				
5	0	0	0	1	0	1				
6	0	0	0	1	1	0				
7	0	0	0	1	1	1				
8	0	0	1	0	0	0				
9	0	0	1	0	0	1				
10	0	0	1	0	1	0				
11	0	0	1	0	1	1				
12	0	0	1	1	0	0				
13	0	0	1	1	0	1				
14	0	0	1	1	1	0				
15	0	0	1	1	1	1				
16	0	1	0	0	0	0				
17	0	1	0	0	0	1				
18	0	1	0	0	1	0				
19	0	1	0	0	1	1				
20	0	1	0	1	0	0				
21	0	1	0	1	0	1				
22	0	1	0	1	1	0				
23	0	1	0	1	1	1				
24	0	1	1	0	0	0				
25	0	1	1	0	0	1				
26	0	1	1	0	1	0				
27	0	1	1	0	1	1				
28	0	1	1	1	0	0				
29	0	1	1	1	0	1				
30	0	1	1	1	1	0				
31	0	1	1	1	1	1				

Name:

9

Continued										
m#	C_1	C_0	A	B	C	D	Z_3	Z_2	Z_1	Z_0
32	1	0	0	0	0	0				
33	1	0	0	0	0	1				
34	1	0	0	0	1	0				
35	1	0	0	0	1	1				
36	1	0	0	1	0	0				
37	1	0	0	1	0	1				
38	1	0	0	1	1	0				
39	1	0	0	1	1	1				
40	1	0	1	0	0	0				
41	1	0	1	0	0	1				
42	1	0	1	0	1	0				
43	1	0	1	0	1	1				
44	1	0	1	1	0	0				
45	1	0	1	1	0	1				
46	1	0	1	1	1	0				
47	1	0	1	1	1	1				
48	1	1	0	0	0	0				
49	1	1	0	0	0	1				
50	1	1	0	0	1	0				
51	1	1	0	0	1	1				
52	1	1	0	1	0	0				
53	1	1	0	1	0	1				
54	1	1	0	1	1	0				
55	1	1	0	1	1	1				
56	1	1	1	0	0	0				
57	1	1	1	0	0	1				
58	1	1	1	0	1	0				
59	1	1	1	0	1	1				
60	1	1	1	1	0	0				
61	1	1	1	1	0	1				
62	1	1	1	1	1	0				
63	1	1	1	1	1	1				

7. (*Vending Machine Design Example*) - 20 points. You are to design a vending machine. The vending machine accepts nickels and dimes only and only contains one product gum that costs \$0.15. Since the machine takes only nickels and dimes, it has two inputs (N and D). The machine can either dispense nothing, dispense gum, or dispense gum and \$0.05 change. Design the vending machine using J-K flip-flops. The state table presented has already been minimized. Use the given state assignment to facilitate grading. Rather than drawing the schematic of the circuit, please show each output in little m notation along with the minimized Boolean equation found using KMAPS.

The block diagram of the vending machine is shown in Figure 1. The resulting state table after state minimization is shown in Table 1. You must use the state assignment provided. Fill in the encoded state transition table shown in Table 3. Use the excitation tables for J-K flip-flops to complete the excitation table for the vending machine shown in Table 4.

Block Diagram:

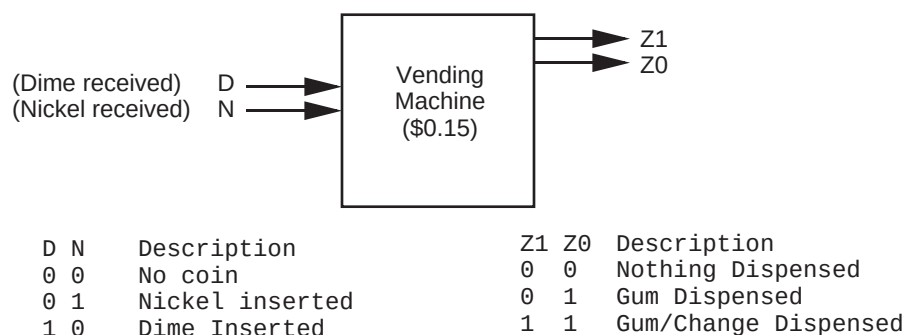


Figure 1: Block Diagram for the Simple Vending Machine.

Table 1: Reduced State Table for the Simple Vending Machine.

PS	NS, $Z_1 Z_0$		
	DN=01	DN = 10	DN = 00
S0	S1,00	S2, 00	S0, 00
S1	S2,00	S0, 01	S1, 00
S2	S0,01	S0, 11	S2, 00

Table 2: State Assignment for the Simple Vending Machine.

State	AB
S0	00
S1	10
S2	11

Table 3: Encoded State Table for the Simple Vending Machine.

PS	NS, Z_1Z_0		
	DN=01	DN = 10	DN = 00
AB	A^+B^+, Z_1Z_0	A^+B^+, Z_1Z_0	A^+B^+, Z_1Z_0
0 0	1 0, 0 0	1 1, 0 0	0 0, 0 0
1 0			
1 1			

Table 4: Excitation Table for the Simple Vending Machine using J-K flip-flops. .

PS	NS, Z_1Z_0		
	DN=01	DN = 10	DN = 00
AB	$J_AK_AJ_BK_B, Z_1Z_0$	$J_AK_AJ_BK_B, Z_1Z_0$	$J_AK_AJ_BK_B, Z_1Z_0$
0 0		1 X 1 X, 00	0 X 0 X, 0 0
1 0			
1 1			

Name:

12

For each output, use little m notation and then on the next line, write the minimized equation found after using KMAPs for minimization. You do not have to draw the circuit.

$$\begin{aligned} J_A(D, N, A, B) &= \sum m \\ &= \\ K_A(D, N, A, B) &= \sum m \\ &= \\ J_B(D, N, A, B) &= \sum m \\ &= \\ K_B(D, N, A, B) &= \sum m \\ &= \\ Z_1(D, N, A, B) &= \sum m \\ &= \\ Z_0(D, N, A, B) &= \sum m \\ &= \end{aligned}$$

8. (*Moore Machine Design*) - 10 points. You are beginning to design a **Moore** machine that controls the tail lights of a 1965 Thunderbird. There are 3 inputs, (*Left*, *Right*, and *Hazard*), to denote a left turn, a right turn, or that all tail lights should flash (hazard condition). There are 3 tail lights on each side of the car that will be connected to the six outputs of our finite state machine. The left lights are called Lc, Lb, and La and the right lights are Ra, Rb, and Rc. The lights operate in a sequence as described below and in the subsequent figure. The state machine also has a clock input whose frequency is slow enough to achieve the desired flashing rate.

Light operation. The lights are all initially off. We assume positive logic here so a zero denotes that a light is off, and a one denotes that a light is on. If the input *Left* is asserted, then the left lights (Lc,Lb,La) will cycle through the sequence 000, 001, 011, 111 and repeat until *Left* is no longer asserted. If the input *Right* is asserted, then the right lights (Ra,Rb,Rc) will cycle through the sequence 000, 100, 110, 111 and repeat until *Right* is no longer asserted.

The inputs *Left* and *Right* are never asserted simultaneously. That is, the driver can only signal a left or a right turn. When the *Left* input is asserted and the left lights are sequencing, all right lights are off. Similarly, if the *Right* input is asserted and the right lights are sequencing, all left lights are off.

The *Hazard* input has precedence over both *Left* and *Right* turn signal inputs. Once the *Hazard* input is asserted, all lights will immediately begin flashing. When the lights are flashing, the machine cycles between a state where all lights are off and a state where all lights are on.

Complete the state diagram of a Moore machine that will be used to implement the tailight controller USING the MINIMUM NUMBER of ADDITIONAL ARCS. Do not add or delete states. States are labeled with outputs in the following order (Lc,Lb,La,Ra,Rb,Rc). Arcs MUST be labeled with the following order. (*Left*, *Right*, *Hazard*). **All changes of sequence must go through the IDLE state.** For example, if we are in the middle of a left turn sequence and we receive a one on *Hazard*, the machine goes directly to the idle state before going to the Flash state.

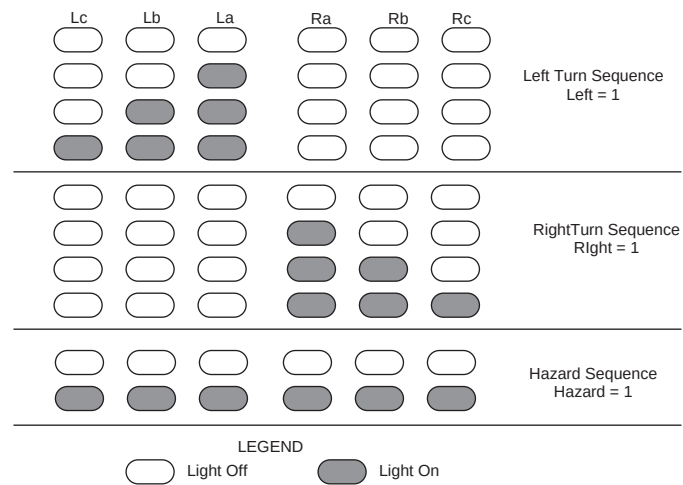


Figure 2: Sequencing for 1965 Thunderbird Tailights.

Inputs: Left, Right, Hazard
Outputs: Lc, Lb, La, Ra, Rb, Rc

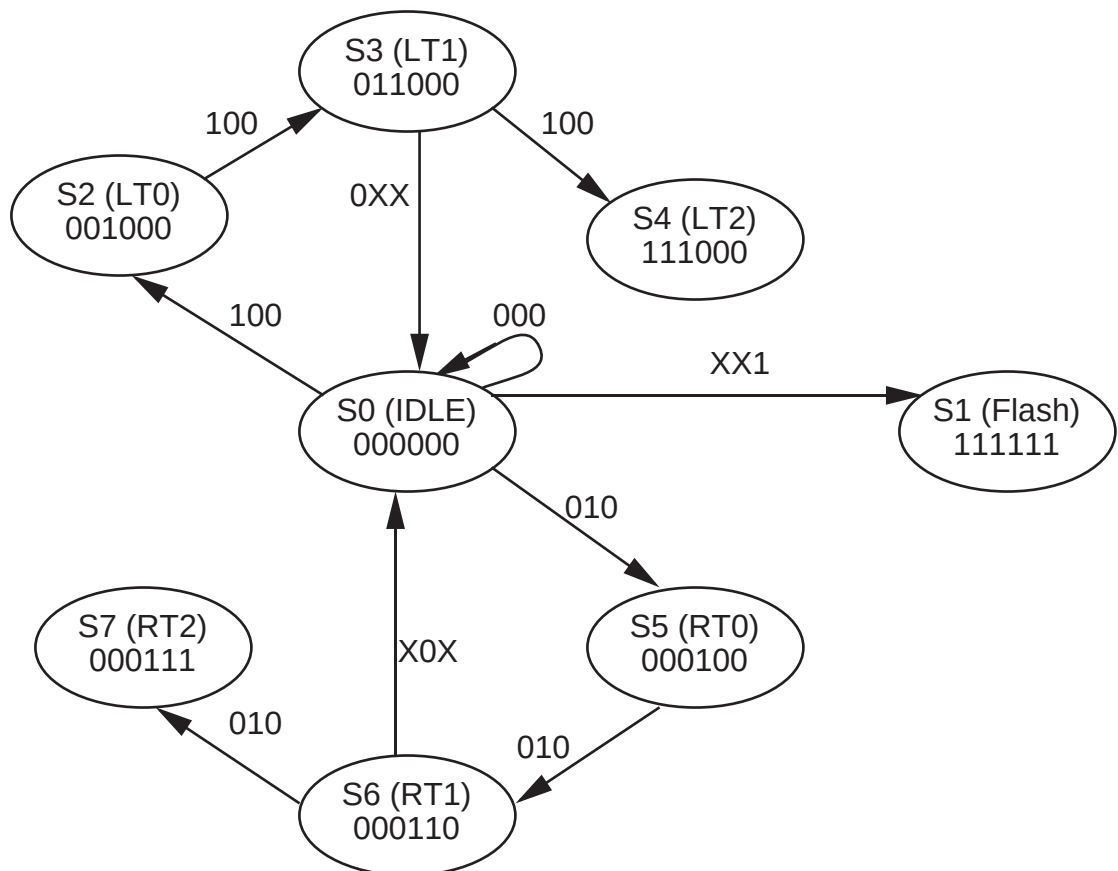


Figure 3: State Diagram for 1965 Thunderbird Tailight Controller.