ECE 198JL Third Midterm Exam
Spring 2014

Tuesday, November 11th, 2014

Name: ____________________________ NetID: ________________________

Discussion Section:

<table>
<thead>
<tr>
<th>Time</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 AM</td>
<td>[ ] JDA</td>
</tr>
<tr>
<td>10:00 AM</td>
<td>[ ] JD1 [ ] JD3</td>
</tr>
<tr>
<td>11:00 AM</td>
<td>[ ] JD2</td>
</tr>
<tr>
<td>12:00 PM</td>
<td>[ ] JD7</td>
</tr>
<tr>
<td>1:00 PM</td>
<td>[ ] JD9 [ ] JD4</td>
</tr>
<tr>
<td>2:00 PM</td>
<td>[ ] JDB [ ] JD5</td>
</tr>
<tr>
<td>3:00 PM</td>
<td>[ ] JDC [ ] JD6</td>
</tr>
<tr>
<td>4:00 PM</td>
<td>[ ] JD8</td>
</tr>
</tbody>
</table>

- Be sure your exam booklet has 10 pages.
- Write your name and lab section on the first page.
- Do not tear the exam booklet apart; you may only detach the last page.
- We have provided a scratch sheet as the last page. If you need more, you may use the backs of pages.
- 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.
- Clearly indicate any assumptions that you make.
- The questions are not weighted equally. Budget your time accordingly.
- Don’t panic, and good luck!

Problem 1 15 points: __________
Problem 2 10 points: __________
Problem 3 20 points: __________
Problem 4 30 points: __________
Problem 5 15 points: __________
Problem 6 10 points: __________

Total 100 points: __________
**Problem 1 (15 points): Synchronous Counter**

In this problem, you must design a 3-bit synchronous counter that counts in the sequence shown below.

\[ \rightarrow 011 \rightarrow 111 \rightarrow 110 \rightarrow 010 \]

1. **(8 points)** The current state of the counter is denoted \( S_2S_1S_0 \). Fill in the K-maps for \( S_2^+ \), \( S_1^+ \), and \( S_0^+ \), using don’t cares when possible.

\[
\begin{array}{c|c|c|c|c}
S_2^+ & S_2S_1 & S_1^+ & S_2S_1 \\
00 & 01 & 11 & 10 \\
0 & & & \\
1 & & & \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
S_0 & S_0 & & & \\
0 & & & & \\
1 & & & & \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
S_0^+ & S_2S_1 & & & \\
00 & 01 & 11 & 10 \\
0 & & & & \\
1 & & & & \\
\end{array}
\]

2. **(3 points)** Write minimal SOP Boolean expressions for \( S_2^+ \), \( S_1^+ \), and \( S_0^+ \). Use the area heuristic discussed in class (number of literals + number of operations) to minimize your expressions.

\[
S_2^+ = \_\_\_\_\_\_\_\_\_ ; \quad S_1^+ = \_\_\_\_\_\_\_\_\_ ; \quad S_0^+ = \_\_\_\_\_\_\_\_\_ \\
\]

3. **(4 points)** Does the counter always converge to the desired sequence, regardless of its initial state? Explain - and be specific to *your solution*.
Problem 2 (10 points): FSM Next-State Analysis

Shown below is an implementation of an FSM with input T and output Z. Complete the next-state table below.

<table>
<thead>
<tr>
<th>S_1</th>
<th>S_0</th>
<th>T</th>
<th>S_1^+</th>
<th>S_0^+</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 3 (20 points): Memory

Your lab partner wants to build a 64×16-bit RAM using 32×8-bit RAM chips (shown to the right). As indicated in the figure, the output Q of the 32×8 RAM chips is gated with tri-state buffers, so more than one chip’s output can be connected together safely provided that only one of the connected chips has CS=1 (chip select).

You have the four 32×8-bit chips that you need, but you only have a 2-to-4 decoder with enable. Your 64×16-bit RAM must have external signals Address, DataIn, DataOut, ChipSelect, and WriteEnable.

1. (5 points) How many bits wide should each external signal for 64×16-bit RAM be?

   Address ___________  DataIn _________  DataOut _______
   ChipSelect _________  WriteEnable _________

2. (15 points) Indicate which external signals or decoder outputs should be connected to each of the component signals to implement your 64×16-bit RAM. Decoder outputs are named D0, D1, D2, and D3. Be sure to indicate which bits of a signal must be connected (for example, Address[3:0] or DataIn[12:5]) if the signal has more than one bit.

2-to-4 Decoder

   Select1 _________________
   Select0 _________________
   Enable _________________

32×8 RAM Chip 1

   A _________________
   D _________________
   CS _________________
   WE _________________
   Q _________________

32×8 RAM Chip 2

   A _________________
   D _________________
   CS _________________
   WE _________________
   Q _________________

32×8 RAM Chip 3

   A _________________
   D _________________
   CS _________________
   WE _________________
   Q _________________

32×8 RAM Chip 4

   A _________________
   D _________________
   CS _________________
   WE _________________
   Q _________________
Problem 4 (30 points): FSM Analysis, Implementation, and Extension

The FSM described here controls two traffic lights at the intersection of roads A and B. FSM inputs are supplied by two traffic sensors, $T_A$ and $T_B$, which are high (1) when traffic is detected on the corresponding road. FSM outputs are light signals $L_A$ and $L_B$ that control the intersection light according to the table shown to the right. The next-state table is shown below.

<table>
<thead>
<tr>
<th>Current State</th>
<th>Input signals</th>
<th>Next state</th>
<th>Outputs for the current state</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>$T_A$</td>
<td>$T_B$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$x$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$x$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$x$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

1. **(9 points)** Draw a state transition diagram for the FSM, labeling states and arcs as shown.

2. **(8 points)** Write minimal SOP Boolean expressions for next-state and output logic. Use the area heuristic discussed in class (number of literals + number of operations) to minimize your expressions.

   $S_1^+ =$ ____________________________

   $S_0^+ =$ ____________________________

   $L_A[1] =$ ____________________________
   $L_A[0] =$ ____________________________

   $L_B[1] =$ ____________________________
   $L_B[0] =$ ____________________________
Problem 4, continued:

3. (6 points) Citizens complain that the traffic lights take too long to change during times of heavy traffic. Your boss instructs you to add a counter and make the green lights time out on both roads, even if traffic continues to arrive.

After being reset and counting to the appropriate timeout value, the counter that you must use produces a signal C=1. Redraw your state transition diagram from Part 1 below and modify it to include this signal.

4. (4 points) Write updated expressions for the next-state logic.

\[ S_1^+ = \]  
\[ S_0^+ = \]

5. (3 points) Write a Boolean expression for the signal that should be used to reset the counter.
Problem 5 (15 points): FSM Design with Components

You are charged with designing an FSM as part of a soda vending machine. The vending machine can hold up to three cans of soda. There are two inputs to the FSM: R (refill), which is asserted when the machine is refilled with a new can of soda (one can at a time), and D (dispense), which is asserted when a can of soda is dispensed by the machine (one can at a time as well). The machine can be refilled in the same cycle in which it dispenses. Your FSM must keep track of the number of cans of soda in the machine and must produce an output L (to control a light) that signals when the vending machine is out of soda (0 cans). Design an FSM to control the light.

The machine cannot hold -1 cans, so your FSM should stay in the same state if an input combination indicates -1 cans. Treat an input combination indicating that the machine contains 4 cans analogously.

1. (8 points) Draw the FSM diagram below using the notation shown below for each state. Choose meaningful state names.
Problem 5, continued:  (problem statement repeated for your convenience)

You are charged with designing an FSM as part of a soda vending machine. The vending machine can hold up to three cans of soda. There are two inputs to the FSM: R (refill), which is asserted when the machine is refilled with a new can of soda (one can at a time), and D (dispense), which is asserted when a can of soda is dispensed by the machine (one can at a time as well). The machine can be refilled in the same cycle in which it dispenses. Your FSM must keep track of the number of cans of soda in the machine and must produce an output L (to control a light) that signals when the vending machine is out of soda (0 cans). Design an FSM to control the light.

The machine cannot hold -1 cans, so your FSM should stay in the same state if an input combination indicates -1 cans. Treat an input combination indicating that the machine contains 4 cans analogously.

2. (7 points) Write the Boolean expressions needed to implement an FSM to control the light L using a 2-bit up-down counter. The counter operates as follows. When H=0, the counter holds its current value Q_1Q_0. When H=1, the counter counts either up by 1 (when U=1) or down by 1 (when U=0).

![2-bit up/down counter with Hold]

Write the three expressions below using **minimum POS form**. *Expressions in other forms will not receive credit.* Use the area heuristic discussed in class (number of literals + number of operations) to minimize your expressions.

\[
U = \underline{\underline{\underline{\underline{}}} }
\]

\[
H = \underline{\underline{\underline{\underline{}}} }
\]

\[
L = \underline{\underline{\underline{\underline{}}} }
\]
Problem 6 (10 points): von Neumann Model

For each question, circle exactly one answer.

Which of the following manages instruction processing in the von Neumann model?

- control unit
- ALU
- MDR
- PC
- IR
- MAR

Which of the following contains the instruction currently being executed by the computer?

- control unit
- ALU
- MDR
- PC
- IR
- MAR

In the von Neumann model, the control unit does not include

- states
- bits
- FSM
- PC
- IR
- MAR

In the von Neumann model, the memory interface includes which of the following?

- IR
- MDR
- PC
- ALU
- FSM

How are instructions represented in a computer based on the von Neumann model?

- opcodes
- bits
- I/O
- control unit
- hexadecimal