CS 151
Quiz 4

Name : ____________________, ____________________
      (Last Name)  (First Name)

Student ID : _______________

Signature : _______________

Instructions:

1. Please verify that your paper contains 10 pages including this cover.
2. Write down your Student-Id on the top of each page of this quiz.
3. This exam is closed book. No notes or other materials are permitted.
4. Total credits of this quiz are 60 points.
5. To receive credit you must show your work clearly.
6. No re-grades will be entertained if you use a pencil.
7. Calculators are NOT allowed.
Q1: [ALU] [10 points]

We are going to design a very simple 4-bit Arithmetic Unit (AU) with the following functional table:

<table>
<thead>
<tr>
<th>M</th>
<th>Function Name</th>
<th>F(A,B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Subtract B from A</td>
<td>A-B</td>
</tr>
<tr>
<td>1</td>
<td>Subtract 1 from A</td>
<td>A-1</td>
</tr>
</tbody>
</table>

A and B are two, 4-bit binary numbers $a_3a_2a_1a_0$ and $b_3b_2b_1b_0$. M is the control input to this AU.

For doing this, the blocks labeled AE (Arithmetic Extender) and CE (Carry Extender) in the following block diagram should be designed:

![Block Diagram]

a. Fill the following table for $y_3, y_2, y_1, y_0$ and $c_0$ based on the inputs of the AU which are $a_3a_2a_1a_0$, $b_3b_2b_1b_0$, $M_1$ and $M_0$: [7 points]

<table>
<thead>
<tr>
<th>M</th>
<th>$y_3$</th>
<th>$y_2$</th>
<th>$y_1$</th>
<th>$y_0$</th>
<th>$c_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\overline{b_3}$</td>
<td>$\overline{b_2}$</td>
<td>$\overline{b_1}$</td>
<td>$\overline{b_0}$</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

CS 151 Digital Logic Design, Fall Quarter'2006, QUIZ 4
b. Using the table that you derived in part (a), derive the logic equations for $y_1$, $y_0$ and $c_0$.

$C_0 = 1$

$y_0 = \overline{b_0} \overline{M}$

$y_1 = M + \overline{M}b_1$
Q2: RTL design [35 points]

We want to design an 8-bit, 3-1 counter using RTL design method. The 8-bit, 3-1 counter has following block diagram and characteristics:

![Block diagram of 3-1 Counter]

a. "Cnt3" input: when Cnt3=1, it adds 3 to the current value, when Cnt3=0 it adds 1 to the current value

b. "LD" input: when LD=1, it loads from input I[7:0] regardless of the value of Cnt3

c. "CLR" input: when CLR=1, it clears the outputs (Q[7:0] = "00000000") regardless of the value of LD or Cnt3
1. Draw the Function Table. [5 points]

<table>
<thead>
<tr>
<th>Cnt3</th>
<th>LD</th>
<th>CLR</th>
<th>Current Value</th>
<th>Next Value</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Q</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>0</td>
<td>Q</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Q</td>
<td>Q+1</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Q</td>
<td>Q+3</td>
<td>x</td>
</tr>
</tbody>
</table>

2. Capture a high-level state machine [10 points]

From init we can go to every other state with appropriate combination of inputs, however I assumed that we always start either with clearing or loading!
3. Create the datapath (Hint: use a register, and an adder to do both increment) [10 points]
4. Connect the datapath to controller [5 points]
5. Derive the controller's FSM [5 points]

As for controller we have the same state machine. However the actions we should take and outputs we should generate are different. So I just draw the states and write the outputs for each state. This is the convention you can use in your quizzes, as long as you mention that everything is the same.

- **init**
  - \( \text{reg.clr}=1 \)

- **Clear**
  - \( \text{input.sel}=0 \)
  - \( \text{reg.id}=1 \)

- **Add3**
  - \( \text{Cnt3.sel}=1 \)
  - \( \text{reg.id}=1 \)
  - \( \text{input.sel}=1 \)

- **Add1**
  - \( \text{Cnt3.sel}=0 \)
  - \( \text{reg.id}=1 \)
  - \( \text{input.sel}=1 \)
Q2: [Shift Register] [15 points]

We want to design a 3-bit circular shift register by adding the circuit in the Black Box to a 3-bit register. By just using 4-to-1 Multiplexers design the Black Box in the figure.

M1M0 are control bits to this Circular Shift Register.
The value on load line is B = b2b1b0.

Function Table:

<table>
<thead>
<tr>
<th>M1</th>
<th>M0</th>
<th>Action</th>
<th>Register Current Value</th>
<th>Register Next Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Keep current value</td>
<td>a2a1a0</td>
<td>a2a1a0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Rotate Shift Right by 1 bit</td>
<td>a2a1a0</td>
<td>a0a2a1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Load</td>
<td>a2a1a0</td>
<td>b2b1b0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Rotate Shift Right by 2 bits</td>
<td>a2a1a0</td>
<td>a1a0a2</td>
</tr>
</tbody>
</table>