Instructions:

- **Turn off all cell phones, beepers and other noise making devices.**
- **Show all work on the front of the test papers. Box each answer.** If you need more room, make a clearly indicated note on the front of the page, "MORE ON BACK", and use the back. The back of the page will **not** be graded without an indication on the front.
- You may **not** use any notes, HW, labs, other books, or calculators.
- This exam counts for 24% of your total grade.
- Read each question **carefully and follow the instructions.**
- You must pledge and sign this page in order for a grade to be assigned.
- Put your name at the top of this test page and be sure your exam consists of **10** distinct pages. Sign your name and add the date below.
- Boolean expression answers must be in **lexical order**, i.e., /A before A, A before B, & D₁ before D₂.
- Label the inputs and outputs of each circuit with activation-levels.
- The point values for problems may be changed at prof’s discretion.
- Notation reminder: \( A(H) \) is the same as \( A.H \).
- Always optimize your answer. The best answer gets the most points.

PLEDGE: On my honor as a University of Florida student, I certify that I have neither given nor received any aid on this examination, nor have I seen anyone else do so.

---

SIGN YOUR NAME

DATE (12 Nov 2008)

---

<table>
<thead>
<tr>
<th>Page</th>
<th>Available</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100</strong></td>
<td></td>
</tr>
</tbody>
</table>
1. A new flip-flop, the GS-FF has the following abbreviated truth (logic) table. (a) Find the full truth table for it. (b) Find the excitation table for this GS-FF. The inputs are active-high and both activation-levels of the output are available.

<table>
<thead>
<tr>
<th>G</th>
<th>S</th>
<th>Q^*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Q</td>
</tr>
</tbody>
</table>

(b) Full truth table

<table>
<thead>
<tr>
<th>G</th>
<th>S</th>
<th>Q</th>
<th>Q^*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) Excitation table

<table>
<thead>
<tr>
<th>G</th>
<th>Q</th>
<th>Q^*</th>
<th>G</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

8%] 2. Design a counter to count the following sequence: 010, 111, 011, 110, 010, 111, etc. When the count is 011, the active-low output G should be true.

Draw the block diagram for the counter (including the flip-flops) and a next state truth table (showing all signals including Z and inputs to the flip-flops). For this design the most significant bit should use a T-flip-flop, the next bit should use a JK-flip-flop. You can use anything you want for any other bits. You may assume that all flip-flops used in this problem have active-low asynchronous pre-set and pre-clear inputs. Also, use a "Start" input that will start the counter at 010. For maximum credit, use the minimum number of flip-flops.
10%] 3. Given the timing diagram below, reconstruct the ASM chart that corresponds to it.
4. Complete the below timing diagram for this ASM chart. Show small propagation delays. Assume that each of the flip-flops used are rising-edge triggered.
10%] 5. Answer the following for the given next-state truth table for an ASM. Y and Z are outputs and W is the only input.

2%) a) What type of output is Y; and what type is Z? Be specific.
   Y: unconditional (Moore)  
   Z: conditional (Mealy)asynchronous

8%) Next, design a circuit to implement this truth table. Use only an EEPROM and D-FF’s. Use no other SSI, MSI or LSI elements. The signals W and Y are active-high and Z is active-low.

(b) Show connections to all EEPROM signals shown below (Please use labels).

(c) List the EEPROM contents in the table below. List them in order starting at address 0. Use as little of the EEPROM as possible. Assume all X’s are programmed to ‘0’.

(c) EEPROM contents

<table>
<thead>
<tr>
<th>Address (Hex)</th>
<th>Data (binary)</th>
<th>Data (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$00 01</td>
<td>0111 1000</td>
<td>$78</td>
</tr>
<tr>
<td>0100 1000</td>
<td>0000 0000</td>
<td>00</td>
</tr>
<tr>
<td>0000 0000</td>
<td>0000 0000</td>
<td>00</td>
</tr>
<tr>
<td>0111 1000</td>
<td>0111 1000</td>
<td>78</td>
</tr>
<tr>
<td>0100 1000</td>
<td>0100 1000</td>
<td>78</td>
</tr>
<tr>
<td>0000 0000</td>
<td>0000 0000</td>
<td>00</td>
</tr>
<tr>
<td>0111 1000</td>
<td>0111 1000</td>
<td>78</td>
</tr>
<tr>
<td>0100 1000</td>
<td>0100 1000</td>
<td>78</td>
</tr>
<tr>
<td>0000 0000</td>
<td>0000 0000</td>
<td>00</td>
</tr>
<tr>
<td>0111 1000</td>
<td>0111 1000</td>
<td>78</td>
</tr>
<tr>
<td>0100 1000</td>
<td>0100 1000</td>
<td>78</td>
</tr>
<tr>
<td>0000 0000</td>
<td>0000 0000</td>
<td>00</td>
</tr>
</tbody>
</table>
6. Design a circuit to implement the following ASM chart. Use only SSI gates (ANDs, NANDs, ORs, etc.) and D-FF’s. Use no MSI elements (Decoders, MUXes, etc.) or LSI elements (ROMs, PALs, etc.). Show all work and label each step of your design. You can abbreviate your signal names by using the bold/underlined letters; otherwise use signal names shown.

**Draw the circuit diagram here:**

\[ V.H \]
\[ Q1.H \]
\[ Q0.H \]
\[ \overline{Q1.H} \]
\[ \overline{Q0.H} \]
BigROM

(a) What is the size of ROM-A in bits? \(1 \times 8\)
(b) What is the size of ROM-C in bits? \(2 \times 8\)
(c) What is the size of BigROM in bits? \(4 \times 8\)
(d) What is the 12-bit address of the first location in ROM-A (with respect to AD11-AD0)?
   \([0100_2,0000_2,0000_2]\) (in binary) \(400\) (in hex)
(e) What is the 12-bit address of the last location in ROM-A (with respect to AD11-AD0)?
   \([0111_2,1111_2,1111_2]\) (in binary) \(7FF\) (in hex)
(f) What is the 12-bit address of the first location in ROM-B (with respect to AD11-AD0)?
   \([0000_2,0000_2,0000_2]\) (in binary) \(000\) (in hex)
(g) What is the 12-bit address of the last location in ROM-B (with respect to AD11-AD0)?
   \([0011_2,1111_2,1111_2]\) (in binary) \(3FF\) (in hex)
(h) What is the 12-bit address of the first location in ROM-C (with respect to AD11-AD0)?
   \([1000_2,0000_2,0000_2]\) (in binary) \(800\) (in hex)
(i) What is the 12-bit address of the last location in ROM-C (with respect to AD11-AD0)?
   \([1111_2,1111_2,1111_2]\) (in binary) \(FFF\) (in hex)
8. A block diagram of a system from your Lab 6 is shown here. Complete the table below (using binary numbers only) to entirely calculate the following expression: $\text{NOT}(A\ \text{OR}\ B) \times 3$, where $A=1001$, $B=0010$. Note the answer should 1100. If a signal does not matter (i.e., any value would be ok), use a dash (−). Assume all registers are initialized to 0. Use ? if the output is unknown.

Use the minimum number of clock cycles necessary. Fill out only the number of rows necessary. If the numbers 1001 and 0010 are replaced, your algorithm should still work. Note that the + means “after the clock edge”

---

**Table from your Lab 6.**

<table>
<thead>
<tr>
<th>MSA</th>
<th>MSB</th>
<th>MSC</th>
<th>Input</th>
<th>Cin</th>
<th>RegA</th>
<th>RegB</th>
<th>Output</th>
<th>RegA+</th>
<th>RegB+</th>
<th>Output+</th>
<th>Cout+</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>010</td>
<td>---</td>
<td>1001</td>
<td>−</td>
<td>0000</td>
<td>0000</td>
<td>???</td>
<td>1001</td>
<td>0000</td>
<td>???</td>
<td>?</td>
</tr>
<tr>
<td>010</td>
<td>000</td>
<td>---</td>
<td>0010</td>
<td>−</td>
<td>1001</td>
<td>0000</td>
<td>???</td>
<td>1001</td>
<td>0010</td>
<td>???</td>
<td>?</td>
</tr>
<tr>
<td>111</td>
<td>011</td>
<td>011</td>
<td>−</td>
<td>−</td>
<td>1001</td>
<td>0101</td>
<td>1011</td>
<td>0010</td>
<td>1011</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>111</td>
<td>101</td>
<td>010</td>
<td>−</td>
<td>−</td>
<td>1111</td>
<td>0110</td>
<td>0100</td>
<td>0100</td>
<td>0100</td>
<td>1011</td>
<td>?</td>
</tr>
<tr>
<td>010</td>
<td>111</td>
<td>110</td>
<td>−</td>
<td>−</td>
<td>0101</td>
<td>0010</td>
<td>1000</td>
<td>0100</td>
<td>1000</td>
<td>0000</td>
<td>?</td>
</tr>
<tr>
<td>111</td>
<td>101</td>
<td>101</td>
<td>−</td>
<td>−</td>
<td>0100</td>
<td>1001</td>
<td>1100</td>
<td>1100</td>
<td>1000</td>
<td>0100</td>
<td>1</td>
</tr>
</tbody>
</table>
15%] 9. Given the following block diagram for a controller that controls the ALU on the previous page, complete the following ASM chart that will perform the following functions.

**When IR1, IR0 = 00:** Load REGA with data (from the INPUT bus). Then, left-shift the data in REGA and store it back into REGA.

**When IR1, IR0 = 01:** Sum whatever is in REGA with whatever is in REGB and put sum back into REGA.

**When IR1, IR0 = 10** (Exchange, operation) REGA gets contents of REGB and vice versa.

**When IR1, IR0 = 11** (Load, multiply, add)
Load REGA with data (from the INPUT Bus).
Load REGB with data (from the INPUT Bus).
(REGA times 2) plus REGB => REGA

After each function has been performed, go back to StateA.

For maximum credit, use **conditional outputs** when possible and the **minimum number of states**.
When not specified, you should hold REGA, Hold REGB, MSC = 000

For ease of grading, please use the notion:
- MSA = 01
- MSB = 00
- MSC = 101
10. Complete the following problems about “debouncing.”

2%) a) Why do we need to “debounce” clock inputs? What happens if we don’t?
    Multiple active clock transitions will occur for a single push on the switch.

4%) b) Complete the following timing diagram to illustrate how the following circuit produces a 
    debounced clock signal, DB.CLK.

Assume the switch is initially resting at terminal “B”. Complete the timing diagram to shown 
it leaving “B” (bouncing), then touching (and bouncing) at “A”, and finally resting at “A”.

(Switch is resting at “B” at this time)