1. For **lab #9**, you are given an **8K x 8 EEPROM** to hold your test program. Your design has the usual 4 bit program counter and **all unused address lines are tied high**. Show the memory contents you must program in the EEPROM to compute the following algorithm: **Multiply the contents of register A by 9 and store it in register B.**

Your code **should not exceed 9 nibbles** in memory and **your last instruction should be an infinite loop** so that no new instructions are executed beyond this point. (6 pt.)

<table>
<thead>
<tr>
<th>EEPROM Address</th>
<th>Required EEPROM Data (program all unused bits as zeros)</th>
</tr>
</thead>
<tbody>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
<tr>
<td>_______________ Hex</td>
<td>____________________________ Hex</td>
</tr>
</tbody>
</table>

2. In the above problem, what initial values of Reg. A will result in an over flow condition after the computation? (2 pt.) ____________________________

---

*Grade Review Information: (NOTE: deadline of request for grade review is the day the exam is returned.)*

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

---

**Point System:**

<table>
<thead>
<tr>
<th>Page 1 &amp; 2</th>
<th>20 points</th>
<th>Page 3</th>
<th>15 points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page 4</td>
<td>15 points</td>
<td>TOTAL</td>
<td>__________ out of 50</td>
</tr>
</tbody>
</table>

*Open book and open notes, *90-minute* examination to be done in pencil.*

*No electronic devices permitted.* All work and solutions are to be written on the exam where appropriate.
3. For the circuit below, derive the logic equation for Z.H. Do not simplify! (6 pt.)

\[ Z.H = \frac{(A \cdot C)}{(A + B)} (A \cdot B \cdot C) \]

4. Using only 2 input NAND gates, implement the following logic equation. Do not simplify! (6 pt.)

\[ Y = \frac{(A \cdot C)}{(A + B)} (A \cdot B \cdot C) \quad ; \text{Assume Y.L, A.L, B.L and C.L} \]
5. A vector containing **75 signed (2's complement) 8 bit numbers** is stored in memory. Write a G-CPU program to convert the 75 values to **unsigned values by adding 128** to the original signed number. i.e. signed numbers -128, 0, 1, and 127 convert to 0, 128, 129, and 255, respectively. Assume you have **1K x 8 ROM starting at $4000** and **1K x 8 RAM at $8000**. Also assume the following below:

   a. Starting address of the signed vector is **$4100**. Starting address of the unsigned vector should be **$8000**.
   b. The vector length is 75 and should be used as a counter value.
   c. Use the X register as a pointer to the signed vector. Use the Y reg. as a pointer to store the unsigned vectors.
   d. ROM $4000-$4FFF is designated as use for program. Only 75 values in RAM are used and the rest is open.

   **Write your answer on the leftmost column lines & if you need more room wrap around to the right column.**

   ORG $4000
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
   __________________________________________________     ______________________________________________
6. The following G-CPU program was put in EEPROM starting at address $4000; answer the questions below.

```
LDAA #$50
LDX #$4000
LDY #$8000
TOP  
    LDAA 0,X
    STAA $40,Y
    INX
    INY
    LDAB #$FF
    ADD_BA
    BNE TOP
```

6A. What is the effective address for the first LDAA #$50 instruction? __________________________ Hex (2 pt.)

6B. What is the effective address for the STAA $40,Y the last time the loop is executed? ______ Hex (2 pt.)

6C. What is the value of the BNE instruction operand? __________________________ Hex (2 pt.)

6D. If the clock is 1 MHz, how many seconds does it take to execute the STAA $40,Y instruction? (1 pt.)

6E. Show the expected values for each of the column variables below for all the cycles required to execute the LDAA 0,X and STAA $40,Y instructions the first time they are executed. All answers in Hex. (8 pt.)

Note: At reset all regs were initialized to zero. The above program will however alter many of them.

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Addr Bus</th>
<th>Data Bus</th>
<th>IR</th>
<th>PC</th>
<th>R/W</th>
<th>X Reg</th>
<th>Y Reg</th>
<th>Reg Driving Addr Bus</th>
<th>Device Driving Data Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>