NAME: ________________________________________ STUDENT #: ______________________

**EECE 259: Introduction to Microcomputers** **QUIZ 1 – 201** **Feb 2, 2009**

1. Fill in the rest of the table.

| Decimal (unsigned) | Binary (%) | Hexadecimal ($) |
|---|---|---|
| 42 | % __ __ __ __ __ __ __ __ | $__ __ |
| __ __ __ | % 1 0 1 1 0 1 0 1 | $__ __ |
| __ __ __ | % __ __ __ __ __ __ __ __ | $ F A |

2. Perform each of the operations in the table below. Express the answer in binary, decimal, and hexadecimal, and compute the flags Z, N, Vsub, Vadd, B and C. The grayed-out cells **will not be marked**.

| Operation on 8-bit values | Results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Binary | Decimal | Hex | Z | N | $V_{sub}$ | $V_{add}$ | B | C |
| a. $FA + $08 | % __ __ __ __ __ __ __ __ | __ __ __ | $__ __ | __ | __ | __ | __ | __ | __ |
| b. $69 + $3C | % __ __ __ __ __ __ __ __ | __ __ __ | $__ __ | __ | __ | __ | __ | __ | __ |
| c. $89 – $3D | % __ __ __ __ __ __ __ __ | __ __ __ | $__ __ | __ | __ | __ | __ | __ | __ |
| d. $22 – $23 | % __ __ __ __ __ __ __ __ | __ __ __ | $__ __ | __ | __ | __ | __ | __ | __ |

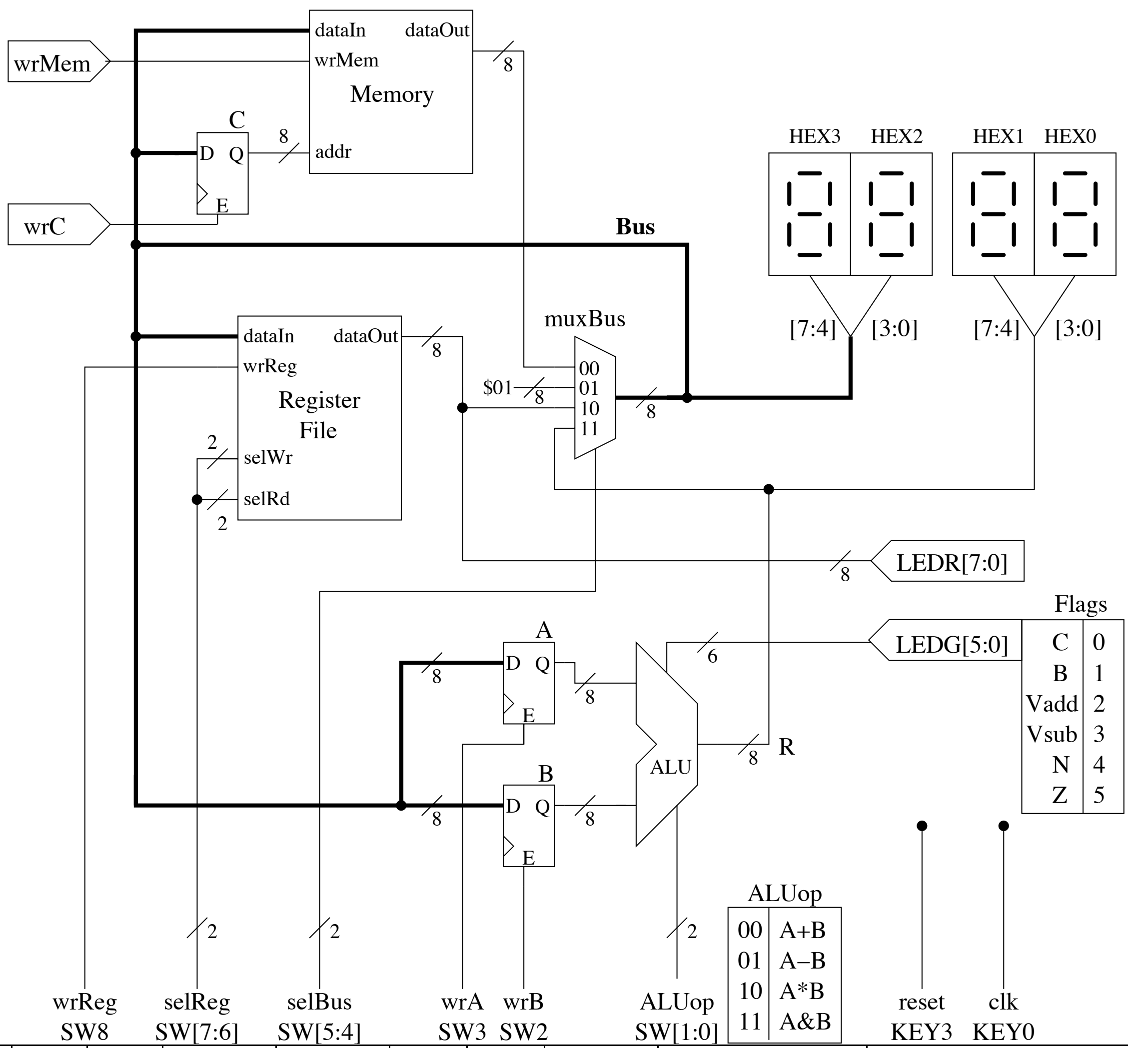
3. Suppose you are using 8-bit signed values to compute something important, and –128 is subtracted from 0. Does this produce an overflow to be concerned about? Explain in plain English, not logic equations.

4. With 4-bit arithmetic, the logic equations for C and B are C= $a_3 b_3 + a_3 \overline{r_3} + b_3 \overline{r_3}$ and B= $\overline{a_3} b_3 + b_3 r_3 + \overline{a_3} r_3$. Give an example where adding two 4-bit values results in both C=0 and B=1. What is the correct way to interpret the B=1 result (keep in mind that your example uses addition)?

5. Use the **Computational Datapath** worksheet on the back to perform each of the operations below. For each operation, write out the sequence of μ-ops required in RTN. Treat each operation as if it is a stand-alone operation that does not depend upon the previous one. Do not rely upon the registers having any particular initial values. Change only the register or memory location specified. There may be multiple solutions.

   a. Mem[R1] ← R3

   b. R2 ← R1 + Mem[2]




| Cycle # | wr Mem | wrC | wr Reg | selReg [1:0] | selBus [1:0] | wrA | wrB | ALU op[1:0] | Micro-operation | Overall Operation |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | |
| 2 | | | | | | | | | | |
| 3 | | | | | | | | | | |
| 4 | | | | | | | | | | |
| 5 | | | | | | | | | | |
| 6 | | | | | | | | | | |
| 7 | | | | | | | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |
| 10 | | | | | | | | | | |
| 11 | | | | | | | | | | |
| 12 | | | | | | | | | | |
| 13 | | | | | | | | | | |
| 14 | | | | | | | | | | |
| 15 | | | | | | | | | | |
| 16 | | | | | | | | | | |
| 17 | | | | | | | | | | |
| 18 | | | | | | | | | | |
| 19 | | | | | | | | | | |
| 20 | | | | | | | | | | |