Midterm Exam Spring 2013
(105 minutes)

Last name:       First name:

Notes:
- If you perceive any ambiguity in any of the questions, state your assumptions clearly
- Questions vary in difficulty; it is strongly recommended that you do not spend too much time on any one question.
- This exam is open book/notes but no electronic devices.

1. [5 points] Circle the correct answer among the choices given. If you circle more than one answer, you will lose the grade of the corresponding question.

(A) Which language of the following is hardware independent?
   a. High Level language
   b. Assembly language
   c. Machine language
   d. They are all hardware independent

(B) If we want to design a computer system, what is the correct order?
   a. develop the ISA first, then the hardware
   b. develop the hardware, then the ISA
   c. develop the high level language first, then the ISA, then the hardware
   d. the order is not important

(C) For the data path given in question 4, which of the following statements is not a valid microinstruction?
   a. MDR->MAR
   b. PC->MDR
   c. MDR->M[MAR]
   d. MAR->ALU
   e. M[MAR]->MDR

(D) If we want to add an extra instruction to the instruction set of a processor, which of the following MUST be updated?
   a. control unit
   b. datapath
   c. the memory system
   d. all of them

(E) The main difference between MIPS ISA and x86 ISA is:
   a. MIPS ISA is more complicated
   b. MIPS ISA is less complicated
   c. x86 uses way more registers
   d. x86 has less instructions
2. [3 points] Suppose you wish to express -64 as 2’s complement number.
   a. What is the minimum number of bits we will need?

   -64 → 1000000 so 7 bits (try to implement it with fewer bits and see if you can!).

   b. With this number of bits you mentioned above, what is the largest positive number you can represent, assuming signed numbers of course? (Please give answer both in decimal and binary)

   with 7 bits and signed number, the most significant bit must be 0 (to designate positive). So the largest we can get is:
   0111111 → +63

   c. With that same number of bits you used in the above two questions, what is the largest unsigned number you can represent? (Please give answer both in decimal and binary).

   In case of unsigned, the largest is: 1111111 → 127

3. [4 points] What is the IEEE 754 presentation of the decimal number: -35?

   35 = 100011 in binary
   100011 = 1.00011 * 2^5
   sign: 1 because number is negative
   exponent – 127 = 5 so exponent = 132 = 10000100

   we have a hidden 1

   so the solution is
   1 1000100 000110000000000000000000
4. [4 points] For the following datapath:

a. Write the microinstructions needed to do fetch phase (optimize as much as you can)

PC → MAR
M[MAR] → MDR  PC +4 → PC
MDR → MAR
MAR → IR

b. Write the microinstructions required to implement: $sw\ Rd, 5(Rt)$ (That is, store the content of register Rd in memory location $[5 + \text{content of Rt}]$).

SE(imm(IR)) → AIR
R[Rt] + AIR → AOR
AOR → MAR
R[Rd] → MDR
MDR → M[MAR]
5. Suppose we want to design a logic circuit that accepts three binary inputs a, b, and c. There is one output for that circuit. The output, f, is set to 1 if the number of 1s in the input is even (example: if a = 1, b=1, and c = 0 then f is 1), otherwise it is set to 0. Assume f = 1 when all inputs are 0.

a. [1 point] Draw the truth table for that circuit.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</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>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

b. [1 point] Draw the logic circuit using gates.

From the table above we can see that: \( f = a'b'c' + a'bc + ab'c + abc' \)

---

c. [2 points] Try to design the same circuit using one decoder and any number of OR gates.

Simple use 3x8 decoder where a, b, and c are the input; and connect outputs 0, 3, 5, and 6 to an OR gate.