CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING
UNIVERSITY OF WISCONSIN—MADISON

Prof Mark D. Hill and Prof. Gurindar Sohi
TAs: Mona Jalal, Preeti Agarwal, Pradip Vallathol, Rebecca Lam

Midterm Examination 4
In Class (50 minutes)
Wednesday, May 8, 2013
Weight: 17.5%

NO: BOOK(S), NOTE(S), OR CALCULATORS OF ANY SORT.
The exam has 12 pages. Circle your final answers. Plan your time carefully since some problems are longer than others. You must turn in the pages 1-9. The LC-3 instruction set is provided to you on the last page.

LAST NAME: ___________________________________________________________
FIRST NAME: ___________________________________________________________
ID# _________________________________
<table>
<thead>
<tr>
<th>Problem</th>
<th>Maximum Points</th>
<th>Points Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1: Multiple Choice Questions (4 Points)
For the following questions, select the best answer. Choose only one answer per question.

i. The TRAP instructions in LC-3 are similar to which of the following instructions in terms of the number of memory accesses that are made to the fetch and execute the instruction?
   a. LD
   b. LEA
   c. LDI
   d. LDR

ii. Which of the following is not true about polling?
   a. The CPU keeps monitoring status register.
   b. CPU cannot perform other tasks during polling.
   c. Polling wastes a lot of CPU time.
   d. **Polling requires changes to the Fetch and Decode logic of the CPU.**

iii. Which of the following is not true about comments in an LC-3 program?
   a. **It is used by the assembler to understand the program.**
   b. Can be used to separate pieces of the program.
   c. Anything after the semicolon is a comment.
   d. They can be used multiple times in a program.

iv. **JSRR R5** is equivalent to
   a. LEA R5, #1
      JMP R7
   b. LEA R7, #1
      JMP R5
   c. LEA R5, #1
      JMP R5
   d. LEA R7, #1
      JMP R7
   e. All of the above are equivalent
Problem 2: Assembly Process  

(5 Points)

Answer the questions below for the following program:

```
.ORIG x4000
LD R2, LOW_P
NOT R2, R2
ADD R2, R2, #1
LEA R0, STRG
; Comment 1
L1  LDR R1, R0, #0
    BRz DONE
    ADD R3, R1, R2
    BRnp SKIP
LD  R1, UPP_P
STR R1, R0, #0
SKIP ADD R0, R0, #1
    BRnzp L1
DONE LEA R0, STRG
    POTS ; Display the string at the address in R0
HALT
LOW_P .FILL x70 ; ASCII Character 'p'
STRG .STRINGZ "Salt and Pepper"
UPP_P .FILL x50 ; ASCII Character 'P'
.END
```

a. Fill out the following symbol table: 

(3 Points)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>x4004</td>
</tr>
<tr>
<td>SKIP</td>
<td>x400A</td>
</tr>
<tr>
<td>DONE</td>
<td>x400C</td>
</tr>
<tr>
<td>LOW_P</td>
<td>x400F</td>
</tr>
<tr>
<td>STRG</td>
<td>x4010</td>
</tr>
<tr>
<td>UPP_P</td>
<td>x4020</td>
</tr>
</tbody>
</table>

b. What is the output of this program? 

(2 Points)

Salt and PePPer
Problem 3: Assembly Errors

Identify the assembly errors in the following assembly program.

```assembly
.ORIG x3000

ADD R4, R4, R4
; OR R2, R3, R4

LOOP ADD R1, R2, #21
    AND R3, R3, #0
    ADD R3, R3, #-1
    BRzp JUMP

STRG .STRINGZ "Error"

HALT STR R4, R4, #16
    TRAP x25

.END
```

(a) 21 cannot be represented in 5 bits

(b) Label JUMP is undefined

(c) HALT cannot be used as a label
Problem 4: TRAPS

Suppose the following LC-3 subroutine implements a new service routine called GETS. The subroutine will store the input string starting at the address in R0 and then return to normal execution. It performs this operation by repeatedly taking input characters from the keyboard and storing it in the location specified by R0 until it sees the '\n' character.

Note: The most significant bit of the KBSR is 1 if keyboard has received a new character.

a. Fill in the blanks. There should be only one instruction per line. (4 Points)

```assembly
.ORIG x0540
ST R0, R0_TMP
ST R1, R1_TMP
ST R2, R2_TMP
L1 LDI R1, KBSR
   (a) BRzp L1 ; Check KBSR
   (b) LDI R2, KBDR ; Load value in the KBDR into R2
   LD R1, NEGCHAR
   ADD R1, R1, R2
   BRz DONE ; Check for '\n'
   STR R2, R0, #0
   ADD R0, R0, #1
   BRnzp L1
DONE (c) AND R2, R2, #0
   STR R2, R0, #0 ; Store NULL CHAR
   LD R2, R2_TMP
   LD R1, R1_TMP
   LD R0, R0_TMP
   (d) RET
```

KBSR .FILL xFE00 ; Address of KBSR
KBDR .FILL xFE02 ; Address of KBDR
NEGCHAR .FILL xFF6 ; Negative value of character '\n'
R0_TMP .FILL 0
R1_TMP .FILL 0
R2_TMP .FILL 0
.END

b. Assume the above assembly code is a service routine that can be called using TRAP x33. What is the address of the corresponding System Control Block entry and what are its contents? Give your answer in hex. (1 Point)

<table>
<thead>
<tr>
<th>Address of trap vector table entry</th>
<th>Contents at this memory location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0033</td>
<td>0x0540</td>
</tr>
</tbody>
</table>
Problem 5: Subroutines (5 Points)

a. There is a problem with the below assembly code segment for a subroutine called `PUTCH`. What is it, and how can you fix the error? (2 Points)

```assembly
.ORIG x5010

PUTCH

ST R0, TMP_R0
ADD R0, R4, 0
OUT ; TRAP x21 which displays the
; character in R0
LD R0, TMP_R0
RET

TMP_R0 .FILL 0
(END)
```

R7 has to be saved before OUT is called.

b. Is the above subroutine `PUTCH` a callee-save or caller-save subroutine? Explain. (1 Point)

Callee-save, it restores the register that it modifies (R0) to its original value.

c. Given the following initial values of registers, what are the values of the registers after the execution of an instruction at address x5050: `JSR PUTCH`; and before the execution of the first instruction of the subroutine. (2 Points)

<table>
<thead>
<tr>
<th>Register</th>
<th>Initial</th>
<th>Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>0x5010</td>
<td>0x5010</td>
</tr>
<tr>
<td>R4</td>
<td>0x5050</td>
<td>0x5050</td>
</tr>
<tr>
<td>R7</td>
<td>0x5010</td>
<td>0x5051</td>
</tr>
<tr>
<td>PC</td>
<td>0x5050</td>
<td>0x5010</td>
</tr>
</tbody>
</table>
Problem 6: I/O  
(3 Points)
Let us monitor the contents of the KBSR (Keyboard Status Register), KBDR (Keyboard Data Register), DSR (Display Status Register) and DDR (Display Data Register) during the execution of TRAP x23 (IN) in LC-3. The leftmost bit of the block is the MSB and the rightmost bit is the LSB of the registers. Note: TRAP x23 (IN) prints prompt to console, read and echo a character from the keyboard.

Below fill in the contents of the different registers at the different steps b, c, and d during the execution of the trap handler for TRAP x23.

a. Initial State:

**KBDR**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**KBSR**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

b. The user types in character “\(\text{V}\)” on the keyboard, but the character is not read.

**KBDR**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

**KBSR**

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

c. The character “\(\text{V}\)” is read from the keyboard and no new character is typed.

**KBSR**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

d. The display is ready but the character is not yet written to the Display Data Register.

**DSR**

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
Problem 7: General Questions (5 Points)

Answer the following short answer questions using 1-2 sentences.

a. What do labels represent in an LC-3 assembly program? (1 Point)

Labels are symbolic names for addresses.

b. What is the difference between Memory Mapped I/O and Special I/O instructions? (2 Points)

Memory mapped I/O has reserved locations in memory that store the addresses of the I/O devices and corresponding registers whereas special I/O instructions use special opcodes for I/O.

c. Why are two passes required during the assembly process? (1 Point)

The first pass is used to find the addresses of all labels used in the program, and the all the offsets used in the machine instructions are calculated in the second pass.

d. What is the difference between a subroutine call and a branch instruction? (1 Point)

A subroutine call is like a function call and saves the return address in R7 in addition to changing the PC while a branch instruction only changes the PC.