CS/ECE 252 Introduction to Computer Engineering

Fall 2018
Instructor: Adil Ibrahim

Homework 7 (47 points)
Due On: November 28, 2018

Primary contact for this homework: Zetong Qi (zqi38@wisc.edu)

For every question below, you need to show the complete working to receive full points. Please utilize the space provided under each question. Upload a PDF version on canvas.

Problem 1: **(6 points)**

Answer the questions below for the following program:

```
      .ORIG x3000
            LD  R2, LOW_A
            NOT R2, R2
            ADD R2, R2, #1
            LEA R0, STRG
      ; Comment 1
      L1    LDR R1, R0, #0
            BRz FINISHED
      ADD R3, R1, R2
      BRnp SKIP

            LD  R1, UPP_A
            STR R1, R0, #0
      SKIP ADD R0, R0, #1
      BRnzp L1
FINISHED LEA R0, STRG
            PUTS
      HALT
LOW_A .FILL x70 ; ASCII Character 'p'
STRG .STRINGZ "Salt and Pepper"
UPP_A .FILL x50 ; ASCII Character 'P'

            .END
```

a. Fill out the following symbol table:

SYMBOL	ADDRESS
L1	x3004
SKIP	x300A
FINISHED	x300C
LOW_A	x300F
STRG	x3010
UPP_A	x3020

b. Explain what the program does and also what is the output of this program.

This program capitalizes all lowercase ‘p’s in a string.

Output: “Salt and PePPer”

Problem 2 (5 points)

Identify 5 assembly errors in the following program:

```
.START x3004                ; Should be .ORIG
       AND R5, R5, Some
       LD R5, STR           ; Should be STRZ
Forward ADD R5, R5, #1
       BRz Forward
       LDR R4, R5, #0
       MUL R4, R4, #1        ; MUL is not a valid command
       ST R4, STRING
Forward HALT                 ; Forward label is used twice (initialization must be unique)

Some    .FILL #0
STRZ  .STRINGZ "HI!!!"
.STOP                        ; Should be .END
```

Problem 3: **(6 points)**

You are given two pieces of program. They have been written by different programmers to store x0018 into memory location x6000.

Module A

```
.ORIG  x5000
AND R2,R2,#0
ADD R2,R2,#15
ADD R2,R2,#9
STI R2,PTR
<rest of the program…..>
HALT
PTR .FILL x6000
.END
```

Module B

```
.ORIG x6000
.FILL x0018
.END
```

a) Why there are two 'ADD' instructions in module A?

Imm5 has a limitation of a decimal value of 15. x0018 is equivalent to 24, which exceeds this limitation. As such, two 'ADD' instructions are needed to add 24 (15 + 9).

b) Explain the fundamental differences in their approaches.

Since Module A uses instructions, the value is stored during runtime. Module B, on the other hand, simply uses pseudo code, which causes the value to be stored during assembly time.

c) Give one example where approach of module A is preferred.

Since Module A's approach involves storing the value within a register (R2), it allows for more flexibility within the code, such as if the value was needed for a purpose other than being stored in memory location x6000.

Problem 4 (15 points)

In this problem, you will write an assembly code that will implement a circular left shift by 6 bits. If you have a binary string 00000011 then the result after circular left shifting by 6 bits will be 11000000. The program we want you to design should take in a 16-bit value at memory location Loc1. Implement the circular left shifting operation by 6 bits and store the 16-bit result in Loc2. You can declare the memory locations Loc1 and Loc2 using .BLKW directive. For example, if [Loc1]=0xF000 then after execution of the program [Loc2] = 0x003C. **Your code should start at memory location 0x3000.**

Your code will be graded through automated scripts. **You should name your file as hw7_q4.asm**. Files with names other than q4.asm will not be recognized by the automated script. To check the functionality of your code, you may run the given sample script for this question using the command mentioned below. However, graders may check the working of code using values different then used in the sample script.

script q4_hw7.lcs

If your test passes successfully, you have to see the message "TRUE(check M3 xC03F)" in the last line of command output window. Any other message that includes "FALSE(check M3 xC03F)", indicates your test is failed.

Problem 5 (15 points)

In this problem, you will write a LC-3 assembly code that removes blank spaces from a string. Assume that the string starts at memory location 0x5000, and is terminated by a '\0' character (ASCII value = 0). Your program should store the modified string in the memory location starting at 0x5100. You do not need to modify the original string stored at 0x5000. You can assume that the original string at 0x5000 will always be less than 100 characters in length, and it will always start with a letter (A-Z, lowercase or uppercase possible).

Note: If the modified string has 7 characters, and the original string has 15 characters, the last 8 characters of your modified string should be all '%' (ASCII value = 0x25).

For example: If the original string at 0x5000 was "aa 12 d e f", the modified string at 0x5100 after your program completes execution should be "aa12def%%%%".
Note that the original string has 4 blank space characters, and the modified string has 4 extra "%" in the end. **Your code should start at memory location 0x3000.**

Your code will be graded through automated scripts. **You should name your file as hw7_q5.asm.** Files with names other than q5.asm will not be recognized by the automated script. To check the functionality of your code, you may run the given sample script for this question using the command mentioned below. However, graders may check the working of code using values different then used in the sample script.

script q5_hw7.lcs