

COMPSCI 210 Assignment 2

Due date: 21:00 16th September 2011

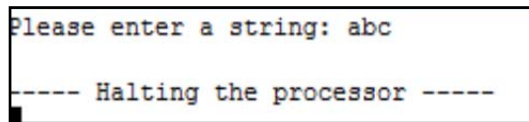
Total marks: 100

The objective of this assignment is to gain good understanding of the execution of the instructions through writing LC-3 assembly language programs.

Note: Most people should be able to do part 1 to 4 without any difficulty. Part 5 needs a lot more efforts than other parts.

Part 1 (25 marks)

Write a LC-3 assembly language program to echo each character typed in by the user. For example, if the user typed in “abc”, the LC-3 console should show “abc”. The user’s input is ended with a LF character (the ASCII value of LF is x0A). The LF character corresponds to the “enter” key on a PC running Window Vista. When the program starts, it should prompt the user for input by displaying a message “Please enter a string: ”.The example below shows the screenshot of the execution of the program.



```
Please enter a string: abc
----- Halting the processor -----
```

Name this program as p1.asm.

Part 2 (25 marks)

Expand the program in part 1 to write a LC-3 assembly language program to (a) read in one string, (b) check whether the string is a palindrome, and, (c) display the result of the checking. The detailed requirements are as below:

1. A string is a palindrome if it can be read the same way in either direction. For example, “abcba” and “abba” are palindromes while “abca” and “abc” are not palindromes.
2. Echo each character typed in by the user.
3. The input is ended with a LF character (the ASCII value of LF is x0A).
4. After a user presses the “enter” key, the program should check whether the string is a palindrome and output the result of the checking. As an example of the execution of your program, a couple of screenshots of the LC-3 console are shown below. In each

screenshot, the first line shows the characters typed in by the user, and, the second line is the output from the program after the user typed the “enter” key.

```
Please enter a string: abcba
The string is a palindrome.
```

```
Please enter a string: abc
The string is not a palindrome.
```

5. Name this program as p2.asm.

Part 3 (30 marks)

In this part of the assignment, you are required to write an assembly program to display the machine code that corresponds to a given LC-3 assembly language instruction. The detailed requirements are as below:

1. Your program should translate LC-3’s AND and ADD assembly language instructions into machine code. In this part, you only need to display the 4-bit opcode of the converted instruction.
2. Your program should accept an assembly language instruction entered by the user. It should be assumed that (a) the instruction entered by the user is terminated by a LF character, (b) the instruction is a valid AND or ADD instruction, (c) there is exactly one space separating the opcode and the operands of the instruction, and, (d) the operands are separated by exactly one comma “,”.
3. Your program should display the instruction typed in by the user.
4. After the user ends the input with a LF character, your program should convert the opcode of the instruction into 4-bit machine code and display the result of the conversion.
5. Name this program as p3.asm.
6. Some examples are given below:

```
Please enter an instruction: ADD R1,R2,R3
0001
```

```
Please enter an instruction: AND R2,R3,R4
0101
```

Part 4 (10 marks)

Expand your program in Part 3 to convert operands as well as the opcode of the instruction. For this part, it should be assumed that the operands only use the “register” addressing mode. That

is, the values of all the operands are stored in registers. Name this program as p4.asm. Some examples are shown below:

```
Please enter an instruction: ADD R1,R2,R3
0001001010000011
```

```
Please enter an instruction: AND R7,R6,R5
0101111110000101
```

Part 5 (10 marks)

Expand your program in Part 4 to allow the operand use the “immediate” addressing mode. That is, the value of an operand is stored in the instruction. Some examples are shown below:

```
Please enter an instruction: ADD R1,R2,2
0001001010100010
```

```
Please enter an instruction: AND R4,R5,-11
0101100101110101
```

Name this program as p5.asm.

Submission

1. You **MUST** thoroughly test your program in the lab before submission. Programs that cannot be assembled or run on a lab machine will **NOT** get any mark.
2. Submit p1.asm, p2.asm, p3.asm, p4.asm and, p5.asm using the assignment drop box.
3. **NO** email submission will be accepted.