COMP-273 Assignment 2

School of Computer Science
McCall University
Available On: Monday, February 4th, 2019.
Due Date: Monday, February 18th, 2019 by 11:59 pm.

Hand in your solutions using *mycourses*, following the instructions at the end of this assignment, on page 3. Before starting create a folder on your computer system and call it *answer-folder*. For Questions 1 and 2, use the provided templates, *.circ and asm* (text) files as a starting point. You must be sure that the solution you provide for Q1 runs in logism-evolution. As well, you must make sure that the solutions you provide for Q2 and Q3 run in MARS. For Q1 we also ask that you explain your circuit design in words, in a separate electronic file. For Q2 and Q3 your explanations should be in the form of comments in your .asm files. **Late policy: 10% off per day late, for up to 2 days. If submitted 48 hrs or more after the deadline, your assignment will not be accepted and will be scored 0!.**

1 An 8-bit Shift Register (30 marks)

Design an 8-bit shift register with output bits $A_7A_6A_5A_4A_3A_2A_1A_0$. Each bit A_i is represented by the Q output of a *rising edge* triggered D Flip-flop, which is available in logisim-evolution. The circuit has a clock and it also has the following two selector inputs: S_0, S_1. The roles of the selector inputs are described in the table below. The timing is such that a hold, shift right, shift left or circular shift right, is controlled by the clock input.

S_1	S_0	Function
0	0	No shift - hold
0	1	Shift right
1	0	Shift left
1	1	Circular shift right

The bit that is shifted out, i.e., when you do a shift right or a shift left, should be stored in an additional output called B_{out}. Correspondingly, an input bit, called B_{in} should be brought in to the space created by the open position, when you carry out a shift operation. A template file has been provided for you called *shift.circ*, which you can start with and then modify. Explain your design in words in a separate text file, which you will also upload, and then submit the circuit (*shift.circ*) implementation in logisim-evolution. Note that the TA's will download and test your circuit design.

2 String Capitalization (30 marks)

You are to write a MIPS program that takes as input a string of text and then does the following.

- Prints the input string
- Capitalizes and prints the capitalized string.

You can assume that the string input is comprised of words consisting of keyboard alphabetic characters, with each word separated by a single “space” character, and with each word beginning with an alphabetical character. The program should be able to handle strings up to 127 characters in length. No characters other than lowercase letters should be modified. In other words, the input string may have some of its letters already capitalized. Use properties of the ASCII code, and in particular the manner in which characters are represented, to solve this problem!

As a working example, given the input string "this is A StrinG", the printed capitalized output should be "THIS IS A STRING". A template file *capitalize.asm* has been provided for you. In this file, you will note that there is an input string that is hard-coded and there is place for an output string. The TA’s might change the contents of the input string while testing your code. You should use the space allocated for the output string to write in the content of your capitalized output. Then, it should be a simple matter to print it.

3 Quadratic Congruence (40 marks)

One of the problems in the set of NP-Complete problems is that of quadratic congruences: given positive integers a, b and c, does there exist an integer $x \leq c$ such that $x^2 \equiv a \bmod b$? For small values of a, b, and c, we can decide by brute force if the congruence can be solved. Write a program in assembly to get values for a, b and c from the user and then outputs each value of x, for which congruence holds, or reports back that there is no solution. You can use a single `main` program for this question; i.e., you do not have to use subroutines. However, if you use subroutines make sure that they are commented and your code is well structured. Note that you can use the `rem` pseudo-instruction to get the modulus of two numbers. Please name your solution file `quad.asm`.

An example of four test cases is given below. We will test your program with these and some other (different) test cases. Note, you can print your solutions for x in any way you like - the table below just shows the full solution set for certain choices of a, b, c.

a	b	c	Solution (x)
4	7	5	Yes (2,5)
1	5	6	Yes (1,4,6)
3	10	4	No solution
5	12	7	No solution

ASSIGNMENT SUBMISSION INSTRUCTIONS

Everything should be handed in electronically on *mycourses*. Each student is to submit his or her own unique solution to these questions.

1. The Logisim circuit (shift.circ) must run under logisim-evolution, to be graded.
2. Comment your code in detail. If there are no comments, and the code is incorrect, you will get ZERO marks. If you make any special assumptions in your programs, or if you feel there are ideas that need explanation, describe them in your comments.
3. For Q1 provide a separate electronic file, explaining your design.
4. Proper use of the argument registers, return value registers, and the stack to save variables or to allocate memory when necessary is required if you use subroutines. Note that for this assignment it is not strictly required to use subroutines.
5. Zip your *answer-folder*, rename it with your student ID number. For example, 260763964.zip
6. Turn in a single zipped file containing one .circ file, two .asm files and the electronic file for Q1, in your assignment folder on *mycourses* under Assignment 2.
7. Hints, suggestions and clarifications may be posted on the discussion board on *mycourses* as questions arise. Even if you don't have any questions, it is a good idea to check the discussion board.
8. Make sure that you submit a single file (the zipped file), not many files.
9. Make sure that the file is in your assignment folder following your intended upload. In other words, make sure what is present in your assignment folder it was what you intended us to grade. Unfortunately, if it is not there or it is corrupted, you cannot submit a corrected one after the deadline.