COMP-273, Winter 2020, Assignment 2

School of Computer Science
McGill University
Available On: Friday, January 31st, 2020.
Due Date: Friday, February 14th, 2020 by 11:59 pm.
Submit your solution in electronic form using mycourses.
Read the assignment submission instructions carefully!
Late policy: 10% off per day late, for up to 2 days.
No submissions accepted after then.

Question 1: Four-Input Register (40 marks)

There are two parts to this question. In the first part, you will design a register using 8 *rising edge* triggered D flip-flops. Use the template *reg.circ* file provided to design your register such that the input bits *in_1* are copied onto the output bits *out_1* when *clock_1* is enabled.

Name this circuit 'Sub_Register'. In the second part, you will design a register using the Sub_Register circuit built in the first part.

Begin by adding a new circuit to your project and name it 'Four_Input_Register'. A copy of your sub-register from the first part will be used as a basis for this circuit. The register will have three input sources:

- an external *data-in* input
- an 8-bit constant input initialized to zero, and
- the output of an Adder that increments the register by adding 1 to the current value in the register.

Add a single clock bit to control the operation of the clock. Finally, add a multiplexer with 8 data bits and 2 select bits. Add a 2-bit input to connect to the select input of the multiplexer. The roles of the select bits are described in the table below. The operation is controlled by the clock input.

S_1	S_0	Function
0	0	Load register from *data-in* input
0	1	Load register with constant 0
1	0	Increment the value in the register
1	1	Hold; Numbers in register are unchanged

A template file has been provided for you called *reg.circ*, which you can start with and then modify. You should not move the components given in the template. Explain your design in words in a separate (PDF) file, which you will upload, and then submit the circuit (*reg.circ*) implementation in logisim-evolution. Note that the TAs will download and test your circuit design.

Question 2: Minimum and maximum value (30 marks)

Write a MIPS subroutine, 'minmax', that has two input arguments: a pointer to a 1D array (i.e., the address of its first element), and the length of the array. Each element of the array is a 32 bit word representing a signed integer. The subroutine should traverse the array and should return the value of both the largest and the smallest numbers in the array.

Include a 'main' procedure to test this subroutine. For this, you can create an array of a fixed length, and initialize it as you wish, using hard-coded values. Print the minimum and maximum values using the appropriate syscalls. You are given a template *minmax.asm* file to work with.

Question 3: Testing for a Perfect Square (30 marks)

Write a MIPS subroutine, 'square', which takes as its input a single argument that is an integer. The subroutine determines if that integer is a perfect square and if so, it returns a value of 1. If it is not, it returns a value of 0.

Include a 'main' procedure to test this subroutine. The 'main' procedure should prompt the user to enter an integer, using an appropriate syscall. It should then read in that integer and call the subroutine, passing this integer as an argument. Finally, it should generate an appropriate print statement to indicate whether the input integer is a perfect square (or not). You are given a template *square.asm* file to work with.

Assignment Submission Instructions

By handing in this your assignment you acknowledge that the work you are submitting is your own, and that you have read the COMP 273 FAQs document under "Content" in *mycourses*.

1. For the MIPs questions, questions 2 and 3, use the provided templates, *.asm* (text) files as a starting point, and build on them, introducing whatever else you need. You must make sure that the solutions you provide compile and run in MARS. For these questions your explanations should be in the form of comments in your .asm files.

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 the MIPs questions proper use of the argument registers, return value registers, and register conventions are necessary.

4. Zip your *answer-folder*, rename it with your student ID number. For example, 260763964.zip. Ensure you use only the *.zip* format and no other compression software.

5. Turn in a single zipped file containing one .circ file, two .asm files and the electronic (PDF) file for Q1, in your assignment folder on *mycourses* under Assignment 2.

6. 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.

7. Make sure that you submit a single file (the zipped file), not many files.

8. Make sure that the file is in your Assignment 2 folder on *mycourses* following your intended upload. In other words, make sure what is present in your assignment folder is 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.