# COMP 273
# **Assignment #2**

Due: Sunday February 20, 2017 at 23:30 on myCourses

## Basic Von Neumann Designs

Please answer these questions on your own and submit your own unique solution to myCourses.
Before starting to solve this assignment, to save your answers' files, make an empty folder and name it *answer-folder*.

## *QUESTION 1: Full 4-Bit Adder-Subtractor (12 Points)*

In class we saw a circuit diagram for a 1-bit full adder. In this assignment you will build a simple 4-bit adder-subtractor that implements two different functions (addition and subtraction). You must build your circuit in the **Logisim-Evolution**[1] using only the basic gates provided in the built-in library, specifically, AND, OR, NOT, XOR, and XNOR. You may set the properties on these gates such as changing the negation of inputs, the number of inputs, or the number of data bits. We provided a starter project for you to help you organize your solutions. You must implement your solution in that starter project by following rules that are mentioned in follow.
You will also need to use wiring such as splitters to organize your implementation. To complete the objectives of this assignment, you must organize your solution into sub-circuits using the names and labels specified below (leave the main circuit empty).

1) You will need to also edit the appearances of some sub-circuits to better organize your solution. But, be careful not to change the sub-circuits' names, and input/output labels in the starter project file!
2) You are free to create additional sub-circuits with custom appearances as you see it and then use them in the starter sub-circuits. The sub-circuits for the main objectives, and in some cases the inputs and outputs, are already set up for you in a provided starter logisim-evolution project file called "Four_Bits_Add_Sub.circ." Make sure that you are filling the starter project and its corresponding sub-circuits with correct circuitries. Do not leave them empty and design other sub-circuits to answer this question!

**(A) Warm up (2 points):**
Implement a one bit full adder in the "Add_1Bit" sub-circuit that takes A, B and Cin as single-bit inputs and produces the Sum and carry-out (Cout) functions.
Note that the sub-circuit appearance was created for you in the starter code. It is also important to note that the text labels in the appearance are short form from the official input and output names defined in the circuit layout! Specifically, Sum is labeled with S for short.

**(B) Build a 4-bit adder/subtractor (10 points):**
To do so, you require to fill out the starter sub-circuit ("Add_Sub_4Bits") with proper circuitries as it is mentioned below:

- You already have implement a 1-bit adder in sub-circuits "Add_1Bit". Implement a 4-bit adder-subtractor in "Add_Sub_4Bits" by using four instances of "Add_1Bit" and complementary circuitries. (6 points)
  Note that there is a control input signal, "Add_Sub", that whenever it is asserted to '1', this circuit ("Add_Sub_4Bits") should perform 4-bit addition; otherwise 4-bit subtraction by using

[1] **DO NOT USE "LOGISIM" VERSION FOR SOLVING THIS ASSIGNMENT! We provided the software (Logisim-evolution) in the assignment's folder. First update your *java*, then just run it.**

2's complement methodology. Then, your circuit should show the correct result on the output "R".
Note that the inputs are in binary and are not represented in 2's complement. Hence, in your circuit, you should take of 2's complement conversion if it is necessary.

- Also, your implementation must be able to handle overflows[2] and zero[3]. (4 points)

***Mandatory;* to organize your solution, for this question, do this:**

1) Save the project file that you have completed, "Four_Bits_Add_Sub.circ", in the *answer-folder*. Note DO NOT change the project file's name ("Four_Bits_Add_Sub.circ")!

## *QUESTION 2: CPU Computations (8 Points)*

Answer the following questions (each is worth 1 point):

(A) The CPU execution time on a benchmark is exactly 11 seconds. What is the working frequency of the CPU if the number of instructions in that benchmark is one million and it takes five clock cycles for this CPU to execute an instruction on average?

(B) Assume that we have a 64-bit bus that works at 300 MHz (it can transmit 64 bits per clock cycle). What is the bus bandwidth in terms of Mbytes per second?

The 5 stages of the processor have the following latencies:

| Fetch | Decode | Execute | Memory | Write-back |
|---|---|---|---|---|
| 300ps | 400ps | 300ps | 550ps | 100ps |

(C) Based on the information that is mentioned in the table above, what is the smallest cycle time for this pipelined processor?

(D) What is the smallest cycle time for this processor if it was not pipelined?

(E) If a CPU runs at 2.5 GHz then how many clock ticks does it make in a second?

(F) If a CPU "cycle" is 2 nanoseconds (i.e., $2 * 10^{-9}$seconds) then how many cycles can occur each second?

(G) We can measure the "speed" of a CPU by calculating the number of instructions it can execute per second. This measurement can be carried out in either of two ways: (a) figuring the average number of clock ticks it takes to execute a single instruction, or (b) by figuring out the average amount of time, in seconds, it takes to

[2] When overflow happens in the operation (add/sub), "Overflow" output signal, in your circuit, should be asserted to '1'; otherwise '0'.

[3] When the result ("R") is all zeros, "0000", "Zero" output signal should be asserted to '1'; otherwise '0'.

execute a single instruction (this is called the "cycle"). Answer the following: (1) Given a 2 GHz CPU and an average instruction execution time of 10 ticks, how many instructions can be executed in a second?
(H) In many of today's microprocessors a pipeline is used to speed up execution of instructions in stages. If we have a 4-stage pipeline and we assume it takes 1 cycle of 2 nanoseconds to pass through a single stage, how many instructions can our four stage CPU execute in 1 second?

***Mandatory;* to organize your answers, for this question, do this:**

1) Write out your answers clearly on paper. You are free to either write your answers by hand, or to typeset them using a software package as long as they are written clearly and legibly. For typeset answers, don't use a font size less than 10pt?
   Note that you should demonstrate for the graders the way that you solve the questions for this section.
2) If you have handwritten answers, scan them, and save the corresponding files into your *answer-folder*. Otherwise, for the typeset answers, save the corresponding files in PDF or MS-Word formats into the *answer-folder*.
3) **Enter your numerical final answers in the answer-sheet that is provided for the Question 2 ("question_2_answer_sheet.txt"), and save it into your *answer-folder*. You can find the corresponding file in the assignment2 folder.**
4) After completing the "question_2_answer_sheet.txt", make sure that your *answer-folder* contains a) "question_2_answer_sheet.txt" file that just contains your final answers, and b) detailed answers (could be scanned papers, or typeset files) that show how you solved different parts of this question.

# WHAT TO HAND IN FOR THIS ASSIGNMENT

Everything should be handed in electronically on myCourses. Each student is to submit his or her own unique solution to these questions on myCourses. **Do not forget that you should submit a single zip file!**

1) Zip your *answer-folder*, rename it with your student ID number. For example, 260763964.zip
2) Submit this single compressed file on myCourse in the box called Assignment#2.
3) Make sure that you submit a single file (the zipped file), not many files.

# HOW IT WILL BE GRADED

- This assignment is worth a total of 20 points
- The breakdown is stated on each question
- Each question is graded proportionally. In other words, if your question is 50% correct you get 50% of the marks.
- For the first two days after the due date, we deduct 10% per day. After that we will dedicate zero to the late assignments.