Lab 7 Specification – Exploring the Logical Circuit Design.
Due (via your git repo) no later than 2 p.m., Wednesday, 11th Nov 2020.
50 points

Lab Goals

- Solidify our understanding of Logical operators and practice circuit design.

Learning Assignment

If you have not done so already, please read all of the relevant "GitHub Guides", available at https://guides.github.com/, which explains how to use many of the features that GitHub provides. In particular, please make sure that you have read guides such as "Mastering Markdown" and "Documenting Your Projects on GitHub"; each of them will help you to understand how to use both GitHub and GitHub Classroom. To do well on this assignment, you should also read

- Principles of Computer Hardware by **Alan Clements**:
  Chapter 02 - 2.1 to 2.3, 2.6;
- Computer Organization and Design by Patterson and Hennesssy - Appendices Section B - B.3;
- Class discussion notes, slides, and in-class coding files.

Assignment Details

Now that we have discussed some fundamental principles behind low-end operators such as the left and right shift, multiply and divide, and got a lead towards the development of logical operators, it is now time to implement some challenging requirements from an operational perspective. In this process, we will also solve multiply, and divide operations using the computing algorithms discussed in class, and develop a logical truth table, and circuit diagram. Additionally, we will watch and reflect on a video to learn some intricacies of the logical operators, to retain the knowledge from the class discussions so far.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share code files and/or other implementation details. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.
2. Submitting a copy of the other's program(s) is strictly not allowed. Please note that all work done during laboratory sessions will be an opportunity for students to learn, practice, and master the materials taught in this course. By doing the work individually, students maximize the learning and increase the chances to do well in other assessments such as lab assignments, skill tests, projects, etc.

At any duration during and/or after the lab session, students are recommended to team up with the Professor and/or the Technical Leader(s) to clarify if there is any confusion related to the items in the lab sheet and/or class materials.

Section 1: Reflection on logical operators.

This section is worth 10 points. The points breakdown is provided below:

- Task 1 = 10 points, a maximum of 2 points awarded for each question.

Details related to logical circuits such as adders are discussed in the video. To complete this part, it is required to do the following:

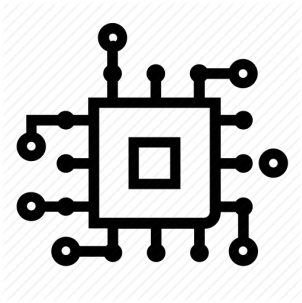
- **Task 1:** Watch this short video, by using the link below:

  `https://www.youtube.com/watch?v=VPw9vPN-3ac`

  After watching the video, create a markdown file and name it as `video-reflection`. In this file, provide a detailed description of the questions presented below:

  1. What is the not equivalent (neq) operation, that the speaker discussed in the video?
  2. What is the limitations of implementing a 3 bit adder by using just an exclusive or?
  3. How do you generate a carry out bit in a half adder?
  4. What did the speaker say about 1 1 1 input for a full adder?
  5. How to develop a Full adder using two half adders?

Section 2: Logical Operators



This section is worth 20 points. The points breakdown is provided below:

- Task 2 = 10 points
- Task 3 = 10 points

We discussed in-class examples to understand the logical expression, circuit, and truth table implementation related to logical operators. In this section, we will further solidify our understanding by practicing the implementation details of these operators.

1. **Task 2:** Review the logical operators discussed in lesson-5 (part-1 and part-2) slides and refer to your class notes. Develop a logical circuit diagram for the expression provided below using the Logicly web interface. **NOTE:** You are only allowed to use nor, nand, xor, and not gate to implement the circuit. You may any combination of the above said gates. For example, you are not allowed to use other gates such as or, and gates. The web interface may be accessed using the link below:

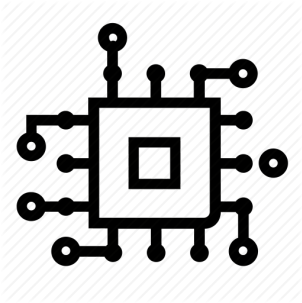
   `https://logic.ly/demo`

   $z = (\neg(\neg a \vee b)) \oplus (\neg(\neg p \wedge q))$

   Once the circuit design is completed and the correctness verified, capture a clear screenshot of the circuit and name the file as `section2-circuit`. Add this file for submission.

   Task 3: Review the logical operators discussed in lesson-5 (part-1 and part-2) slides and refer to your class notes. Develop a logical truth table for the logical expression provided in the previous part. There are four inputs, and hence there are a total of 16 combinations for the total number of inputs. Edit the truth-table markdown file to provide the correct details related to the logical expression. In the markdown file, t1 represents the first term and t2 represents the second term respectively. That is, $t1 = (\neg(\neg a \vee b))$ and $t2 = (\neg(\neg p \wedge q))$. Note: Make sure to verify the correctness of this truth table by testing out the combinations using the circuit diagram developed in the previous part. Once work is completed, add the completed truth-table markdown file for submission.

Section 3: Logical Circuit Design



This section is worth 20 points. The points breakdown is provided below:

- Task 4 = 10 points
- Task 5 = 10 points

We discussed in-class examples to understand the logical circuit design for half and full adder circuit, and truth table implementation related to the circuit design. In this section, we will further solidify our understanding by practicing the implementation details of these functionalities.

1. **Task 4:** Review the logical full adder discussed in lesson-5 (part-2) slides and refer to your class notes. Develop a logical circuit diagram for a 4 bit full adder using the Logicly web interface. **NOTE:** You are required to use 4 full adders to implement the circuit. There should be two sets of 4 bit input switches. The final output should be displayed using both a series of bulbs and the digit output device. We did this as a class activity and this part is required to be completed to reiterate what was learned in class. The web interface may be accessed using the link below:

   `https://logic.ly/demo`

   Once the circuit design is completed and the correctness verified, capture a clear screenshot of the circuit and name the file as `section3-circuit1`. Add this file for submission.

   Task 5: Implement a circuit that performs a 4 bit full subtractor. You may assume that the circuit works only for inputs, where Minuend is greater than Subtrahend. For example, in (8-3), the Minuend is 8 and Subtrahend is 3. Detailed thinking, and applying proper revisions to the circuit is Task 4, should help completing this part. A hint is that we need to apply 2's complement to the subtrahend. 2's complement is simply inverting the input and adding 1 to it. A constant 1 may be used for designing this part of the circuit. Note: Make sure to verify the correctness of this truth table by testing out the combinations using different inputs.

   Once the circuit design is completed and the correctness verified, capture a clear screenshot of the circuit and name the file as `section3-circuit2`. Add this file for submission.

Submission Details

For this assignment, please submit the following to your GitHub lab repository.

1. **video-reflection** markdown file
2. an upload of the **section2-circuit** image file.
3. an upload of the **section3-circuit1** image file.

4. an upload of the **section3-circuit2** image file.
5. an updated version of **truth-table.md** file.
6. It is highly important, for you to meet the honor code standards provided by the college and to ensure that the submission is made before the deadline. The honor code policy can be accessed through the course syllabus. Make sure to add the statement ”This work is mine unless otherwise cited.” in all your deliverables such as source code and PDF files.
7. It is recommended to upload a summary file, with the details that you would like the Professor to know while grading the work. For example, it may be a reflection of your experience in the lab by highlighting some of the challenges that you had faced and a brief mention of how you addressed those challenges while implementing this lab. The summary file may also include a brief mention of any details that one should know about executing your program and what to expect during the execution.

Grading Rubric

1. Details including the points breakdown are provided in the individual sections above.
2. If a student needs any clarification on their lab credits, it is strongly recommended to talk to the Professor. The lab credits may be changed if deemed appropriate.