CS2100 Computer Organisation

Lab 08: Simple Logic Circuit (Week 26th October)

Instruction

Short and clean
We have separated the lab information into i) **instruction** and ii) **report**. Whenever there is a question in the instruction (easily identified as they have **[X pts]** tagged to the end), write / type your answer in the corresponding location in the **report** document. Please take note of the submission specification at the end of this document.

Objective

We will start the journey on **digital logic circuit** for the next 3 labs. In this lab, you will

1. Learn to use **Digital**, a software simulator to simulate physical logic circuit.
2. Use **Digital** to analyse simple digital circuit.

Due to Covid (sigh), we have to move the digital circuit lab online by using software simulator. For your curiosity and learning purpose, we will try to do a live stream / recording of the physical counterpart for this lab.

Reference:

Refer to Lecture 16 and 17 for the relevant topics.

Preparation (before the lab):

1. Download and setup the **Digital** software. Go to https://github.com/hneemann/Digital , browse to the section "Download and Installation". Follow the instruction for your platform. **Digital** supports Windows, Mac OS and Linux.

2. You may need to install Java Runtime Environment (JRE). Proceed to install if you are prompted.

3. Create the simple XOR circuit by following the tutorial (you can also view a walkthrough video recording by Uncle Soo, found in Luminus→Multimedia→Supplementary).

Task: Prime Day! [12 marks]

1. Suppose we want to build a circuit to **check for prime numbers** (e.g. 2, 3, 5, ….). Give the truth table for function **F(a, b, c)**, where **F** output a **1** if **abc** as interpreted as a **3-bit unsigned integer** is a prime number; Output is **0** for other numbers. [Note: Since this truth table drives the **entire lab**, make sure you get it correct. Discussed with your friends, lab TA if you are not sure.] **[4 pts]**

2. Give the Σ*m* notation for function **F**. **[1 pt]**

3. Draw the logic diagram for function **F** as an 2-level AND-OR circuit. Note: i) complemented inputs (*a'*, *b'*, *c'*) are not available; ii) Use a **fan-in of 2** only. (i.e. AND / OR gate takes only 2 inputs).
   Label the final output and all intermediate outputs. Use product/sum term, e.g. **a.b.c** instead of minterm/maxterm notation, e.g. **m7**.

   (Marks will be deducted if the labelling is not complete or correct, or if the diagram is not neat. Refer to "**CS2100 Logic Circuit Guideline**" provided.)

   Note that in writing a product term, write the dot (•) operator. For example, write *a*·*b*·*c* instead of *abc*. **[4 pts]**

4. Simplify the function **F** to the **simplified SOP expression** using Boolean Algebra (showing each step clearly and naming theorems used) **[3 pts]**

   F = (A'.B)+(A.C)

5. Using the **Digital** software simulator, implement function F using the **simplified SOP** (i.e. from steps 4 above). Make sure you use **and label** the components "input" and "output" correctly. Demonstrate to your lab TA. **[4 pts]**

You need to submit into Luminus Folder no longer than **2359 on the same day** you have the online lab. Please rename the **report document** with your student id: **[student_id]_lab08.docx** or **[student_id]_lab08.pdf**, e.g. **A1234567X_lab08.pdf**. Submit into the correct folder on Luminus before 2359 of your lab day.
Marking Scheme: Report – 12 marks; Demonstration – 4 marks; Total: 16 marks.