Department of Computing
COMP2340 Computer Systems Organization
Semester II 2024-25

Lab 01: Digital Logic

Exercise 1: Introducing Logisim [10 marks]

Logisim is a computer program that can be used to simulate digital logic circuits. It makes learning about logic gates and circuits easier than it would be if you had to connect wires on a physical circuit.

1.1 Preparation

1. Download Logisim and install the program on your computer.
2. Download the files **Lab01.circ** and save it to a location from which you can open later.

1.2 Using Logisim

1. Start the Logisim program and open the file **Lab01.circ.** You should see several circuits in the **Ex1.2** workspace. Each circuit has a logic gate with wires that are connected to squares and round shapes.
2. The squares are switches that you can toggle between the value 0 or 1 when they are clicked with the hand tool. For this lab we will consider 1 to mean high voltage and 0 to mean low voltage. Remember, a bit is represented as a voltage on a wire.
3. The little round shapes represent LED's (light emitting diodes). An LED glows whenever the wire to which it is connected has the value 1. In Logisim, a glow is present if the LED is coloured light green. Otherwise, the LED is coloured dark green. In the simulator, any wire carrying a 1 will glow.
4. The icon that is shaped like a hand is the manipulation tool. Click on this icon to make sure that you can set the values in the input. The mouse pointer should now be shaped like a hand. Test it out by clicking on the input to the NOT gate. The value at the input should toggle from 0 to 1 or vice versa. Click on the input again to turn it off.
5. Observe the behaviour of the logic gates for the AND, OR, NAND and NOR gates, and complete the truth table (on the **Lab 01 Exercise Sheet**) based on the behaviour.
6. Add an XOR gate to the circuit. Add two switches to the inputs, and an LED at the output, using the wiring tool to make the connection. Complete the truth table based on information observed in the simulator.
7. Add an XNOR gate to the circuit, and repeat the steps taken in part 6 above.

Exercise 2. Combinational Logic Circuits [40 marks]

Digital logic gates can be wired together to create circuits that perform more involved operations than just basic Boolean operations. It may be a good idea to review the ***beginner's tutorial*** in the Logisim documentation prior to this exercise.

2.1 Working with more Complex Circuits (15 marks)

1. Open the **Ex2.1** workspace and complete the exercise.

2.2 Converting Boolean Expressions to Digital Logic Circuits (25 marks)

1. For this exercise, you will have five circuits (**Ex2.2 A – Ex2.2 E**) to design based on the given Boolean Expressions:

    a. $\boldsymbol{ABC + \overline{A}\,\overline{B}\,\overline{C}}$

    b. $\boldsymbol{ABC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C}}$

    c. $\boldsymbol{A(B\overline{C} + \overline{B}C)}$

    d. $\boldsymbol{ABC + A(\overline{B + C})}$

    e. $\boldsymbol{(A + B)(A + C)(\overline{A} + \overline{B})}$

2. Ensure that the circuits you design have labels for all **Inputs**, **Intermediates**, **and the Final Output**. Use the Text tool where necessary.

3. **NOTE:** You are expected to use only the following gates to design your circuits: **NOT, AND, OR, NAND, NOR, XOR, XNOR**.

Exercise 3: Working with Adders [40 marks]

Digital logic gates can be wired together to create circuits that perform more involved operations than just basic Boolean operations. For this Exercise, you will be creating logic circuits to perform binary addition.

3.1 Creating a Half-Adder (10 marks)

1. Create a half-adder component:
    a. Open the **Ex3.1-halfadd** workspace from your Lab 01 circuit library.
    b. Open the Combinational Analysis tool (under the Window menu.)
    c. Use the Inputs tab to create two inputs labelled ***a*** and ***b***.
    d. Use the Outputs tab to create two outputs labelled ***sum*** and ***carry***.
    e. Use the Table tab to specify the truth table for

    $$\boldsymbol{sum = a\ XOR\ b}$$
    $$\boldsymbol{carry = a\ AND\ b}$$

    f. Build, test your circuit, and construct your truth table.

3.2 Create a Full-Adder Using the Half-Adder Component (15 marks)

Most addition problems will involve adding numbers with more than one bit. For example, to add the value 7 to the value 5 you need to add the binary values 0111 and 0101. Addition of such values can be accomplished by simply adding the bits in corresponding positions of the two values from right to left just as we do with decimal values. A half-adder cannot do this properly because it does not have an input that allows it to add a carry (if any) from the previous position after adding the two bits in a position. To even contemplate anything more than simplistic addition, we need a device that can add two bits in the proper manner, accommodating addition of a carry where necessary. The circuit that does this is called a ***full-adder***. You will now use two half-adders to compose a full-adder.

1. Open the **Ex3.2-fulladd** workspace from your Lab 01 circuit library.
2. By clicking on the **Ex3.1-halfadd** component, drag two half-adders onto the workspace. Add three (3) one-bit inputs labelled **A**, **B** and **Cin** respectively and two one-bit outputs labelled **Sum** and **Cout**.
3. Connect the components to form a ***full-adder*** using the strategy discussed in the lecture

slides.

4. Construct your truth table in the **Lab 01 Exercise Sheet.**

3.3 Creating an 8-bit Adder (15 marks)

Review the “Wire bundle” section of the Logisim documentation and complete the following:

1. Open the **Ex3.3-8bitadd** workspace from your Lab 01 circuit library.
2. Load **add4.circ** from your circuit library. (This will allow you to insert 4-adder circuits). Add two 4-adders to the workspace.
3. Create two 8-bit wide input pins named **Arg1** and **Arg2**.
4. Create a 1-bit wide output pin named **Cout**.
5. Create an 8-bit wide output pin named **Sum**.
6. Attach East-facing splitters to **Arg1** and **Arg2**. Before adding these splitters, make sure the bit-width is 8 and the fan-out is 2. Make sure the fan wires are not blue.
7. Attach a West-facing splitter to **Sum**. Its wires will be blue.
8. Implement the connections between the adders that will allow the two 4-adders to work together to perform the function of an 8-bit adder.
9. Connect the input of each adder (or half adder) to the corresponding wire of the splitter connected to **Arg1**.
10. Repeat for **Arg2**.
11. Test your circuit. A truth table is not required. Why do you think so? Bonus marks for those who can provide a reasonable explanation.

Submission

You are required to submit the **Lab01.circ** and completed **Lab 01 Exercise Sheet** on VLE using the submission link by 11:45 pm Friday, February 28th.