Digital Electronics Using Multisim

In this lab you will be learning about the digital electronics features in Multisim, and how they can help you to analyze and learn about digital circuits.

Prelab Exercises

1. You are having lunch at Chartwells and really want some soft-serve ice cream. However, you have been disappointed so many times because you find out at the last minute that the machine is out of order. In order to make sure you always know if you can get ice cream, you decide to install several sensors connected to a digital logic circuit. You have a sensor (A) that outputs a HIGH if the machine is running. You also have two sensors that output HIGHs to warn you if either the chocolate (B) or vanilla (C) is empty. (You don't care which flavor you get, just that at least one of the flavors is not empty.) The final sensor tests the temperature of the ice cream and outputs a HIGH to warn you that the ice cream is too warm. Even when the machine is off, if the ice cream is still cold enough, you will still be able to have ice cream. Your final circuit should output a HIGH which turns on a green light telling you if you <u>can</u> get ice cream. If you can't get ice cream, your circuit should output a LOW.

    a. Write down and label which variables are important. (B = chocolate is empty)

    b. Write out a truth table for this problem. Give all the possible combinations of inputs, and mark which ones will give you a HIGH output or a LOW output. Write out the resulting logic expression. Make sure to use the same input definitions as you defined in part (a)!

    c. Simplify your expression using a K-map, and write the resulting equation.

    d. Draw the circuit resulting from this expression using logic gates. Draw this neatly!

Lab Exercises

In this lab you will learn how to use the digital logic components in Multisim, as well as how to use the tools provided in Multisim for designing and analyzing digital logic circuits.

Part 1

Logic Gates:

1. In this part of the lab you will test the truth tables for 5 basic logic gates.

    - In Digital Logic, the inputs are either a "HIGH" (1) or a "LOW" (0). So, we will be using digital sources for this lab rather than regular voltage sources. Find and place two digital sources from the part group *SOURCES*. (Hint: if you find a source that you can change "interactively," this part will go somewhat faster)

- There are two ways to do this part of the lab efficiently. One way is to place one component at a time. If you choose to do this, you will find the "Replace Component" command very helpful. You can find this command by right-clicking on a component. This keeps you from having to re-wire everything each time you change gates.

- The other option is to create bus lines as shown in Figure 1, and attach and observe the outputs of all the gates at once. Note, these are only two-input gates, so you will only need bus lines A and B for this part.

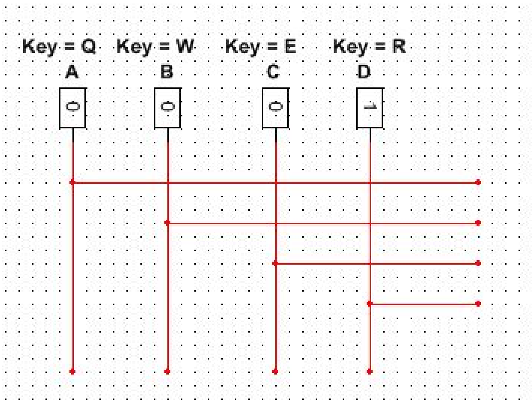

Figure 1

- Now find and place each of the following logic components from the part group *TTL*.

  – NOT (7404N)

  – AND (7408N)

  – OR (7432N)

  – NAND (7400N)

  – NOR (7402N)

- A menu will pop up when you place these components as shown in Figure 2.

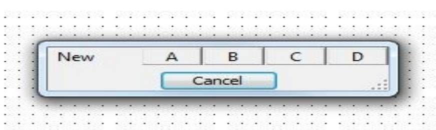

Figure 2

It does not matter what you choose to name this component, so just pick any button and then you will be able to place the component in the work space.

- In order to easily see the output, attach a probe from the part group INDICATORS to the output of each gate. This probe is like an LED, but is easier to use in order to just test the output. Note: a colored probe will more easily show the state of the output.
- Test and record the truth table for each gate (you must simulate the circuit by pushing the green arrow in order for the circuit to work correctly). You may combine the truth tables all into one large truth table with output columns for each gate if you would like.
- A NAND gate is the same as a NOT-AND. This means that the truth table for a NAND gate should be exactly opposite as the truth table for an AND gate. Is this true?
- Verify that you see the same relationship between the truth tables for the NOR and OR gates.

2. Now you will test the circuit you designed in the prelab.
   - Implement the circuit diagram you designed in the prelab (make sure to SAVE!) and test it using the digital sources and a probe at the output. It will be easiest to do this by creating bus lines such as those suggested above (see Figure 1).

Part 2

Logic Converter:

3. Calculating K-maps by hand for 4 inputs or less is fairly simple. However, with 5 or more inputs, a 3-dimensional K-map is needed. This is very difficult and time-consuming to do by hand! Fortunately, we can make Multisim do this with very little work on our part. All we need is the Logic Converter.
   - In order to learn how the Logic Converter works, we will first "redo" the circuit from the prelab.
   - Find the Logic Converter icon on the right side of the screen, click on it, and place the tool in the workspace. Double click on it. This tool gives you the ability to create your own Truth Table. Implement your truth table from the prelab by turning on the desired number of inputs and marking the output for each one as either a 1 or 0.
   - Click the button to the right of the table that gives you the resulting expression (shown as numbers to letters). The expression will show up at the bottom of the Logic Converter window. Does this agree with what you had in the prelab?
   - Find and click the button that simplifies this expression (similar button as before, except also says "SIMP"). Does this agree with your prelab answer after the K-map?
   - Implement this expression in gates using the button in the logic converter. Click on the workspace to place the set of gates. (SAVE!) Verify this logic circuit as you did in Part 1 of the

lab, by attaching digital sources to the inputs and a probe to the output, making sure each combination gives you what you expect.

4. Now we will take advantage of the fact that Multisim can handle 5- or more input logic circuits very easily:

   - Pick two random 5-bit numbers (ex: 10010, 11001). We will build a circuit that lights up a probe when either of these two sets of inputs is detected.
   - Using the logic converter, make the truth table, simplified expression, and gate structure for this 5-bit detector circuit as you did earlier. (Make sure to SAVE!)
   - This will most likely be a very large circuit! In order to NOT have to draw it in your lab book, write down the two numbers you want to be detected, and have a TA verify your circuit performs as expected. **You must have a signature here in order to be signed off for the lab!**

Part 3

Logic Analyzer:

5. A logic analyzer is a very useful tool in analyzing digital signals. (Hint: don't draw any of these signals until the end of this section – this section builds on itself)

   - Find the logic analyzer in the same area you found the Logic Converter, and place it in the workspace.
   - First, to see how it works, find and place a digital clock source, and attach it to the first pin of the Logic Analyzer. Simulate the circuit and click on the logic analyzer. The signal you see is a clock pulse, similar to the square wave you used in the analog lab.
   - Now place another digital clock, and change the frequency to one-half the value of the first clock you placed. Attach this clock to the logic analyzer below the first clock. Simulate and view the output as before.
   - Place an AND gate and an OR gate on the workspace. Connect each input to one of the clock sources, and attach the output of each gate to the logic analyzer. (Hint: to make the logic analyzer easier to read, it may help you to know that the signals come up the same color as the net. You can change this color for every net if you want.) (Another Hint: create bus lines for each of the clocks as you did earlier so it is easier to connect the different gates.)
   - SAVE, and then run the simulation and view the output as before. Draw the four output signals (at least 1.5-2 periods) in your lab book, lined up as shown on the logic analyzer. Explain what these signals are telling you. (Hint: look at the input clock signals in four places: both are high, both are low, one is high and the other is low, and vice-versa. What are the AND and OR outputs for each? Does this match the truth tables for each?)