Dordt College

Engineering Department

EGR 204 DeMorgan's Theorems and Logic Simplification Spring 2005, Lab 2

OBJECTIVES:

- To learn more about how to verify a design by simulating the logic circuit
- To learn more about how to build logic circuits using TTL gates
- To demonstrate the use of Boolean Algebra to simplify a logic circuit.

EQUIPMENT NEEDED

- Computer with Quartus II installed (Web Edition or similar. This handout was developed with version 4.1)
- 1 FOX kit microprocessor trainer with matching numbered power supply and a logic probe
- The following TTL gates,
  3 type '00, Quad 2-input NAND
  1 type '02, Quad 2 input NOR
  1 type '10, Triple 3 input NAND

RECOMMENDED REFERENCE

The handout for Lab 1 contains a tutorial on using Quartus II for logic circuit simulation.

DEMORGAN'S THEOREMS AND THE SUFFICIENCY OF NAND GATES

Simulate the logic circuit shown in Figure 1 below. A suggested name for this project is "LAB2_DEMORGAN." Use the models suggested above in the equipment list for the gate symbols. (Recall that they are found in the "others | maxplus2" library.)

Notice that the gate on the right-hand side of Figure 1 has only one input. The smallest available NAND gate (type '00) has two inputs. It is conventional when such a situation arises to connect the unused input to a logic level which will not force the output of the logic gate to either 1 or 0. For the NAND gate, connect unused inputs to logic-1 (also known as "+5 V" or "V_{CC}."). This way the remaining input and output function as an inverter. (For a NOR gate, you would connect unused inputs to logic-0 or ground, or "V_{EE}." Again, the remaining input and output function as an inverter.) In schematics found in books and magazines, and in this handout, the wiring for unused inputs in not always explicity shown. In such a case you must figure that out for yourself when you wire or simulate the circuit.

Constant values of logic-1 and logic-0 are available from primitive symbols in the simulator. In Quartus II you can find them under "<filepath>/primitives/other" where they are labeled "vcc" for logic-1 and "gnd" for logic-0. Be sure to add a "vcc" symbol to your schematic and connect it to the unused input on the right-most gate in the schematic drawing file that you create.

In the waveform editor, make S1 and S0 cycle through all four possible combinations in the order of a truth table. Make sure L0 and L1 are shown in the waveform editor. Choose a reasonable simulation end time (a few microseconds) and an appropriate grid size.

When you do the simulation, be sure you do a functional (not a timing) simulation. Here is a new feature that was not covered in the previous lab: You can save the simulation output in the waveform file. To do this, click on "Assignments | settings" and in the "Categories" window click on "Simulator." Then put a check mark in "Overwrite simulation input file with simulation results." (recommended for this lab.) If this is done, then any time after the simulator is run, you can open the waveform file and view the simulation results without re-running the simulation.

After the simulation runs, evaluate the output and see if it makes sense according to your understanding of how the gates should work. To do this evaluation, use a truth table to figure out what the logic circuit's output values should be. Compare your truth table to the simulation's output.

After simulating the logic circuit, build it in actual hardware. To do this, first place one type '02 integrated circuit (IC) and one type '00 integrated circuit on your breadboard. Connect power and ground pins appropriately and then apply power momentarily to check for shorts. (If "ready" shows in the FOX kit's display, then there probably are no shorts.) After the power and ground connections are known to be good, connect the inputs and outputs of the gates as shown in the figure. The labels S0, S1, L0, and L1 refer to Port 2 LED's and switches. (Be sure there are no connections to the Port 2 enable inputs, P2XEN, P2YEN, P2XCL and P2YCL.)

After the wiring is complete, apply power and manipulate the switches while observing the LED's to follow the signals shown in the simulation waveform file. The behavior of the real circuit should match that of the simulated circuit. (The signal connected to L1 should produce exactly the same result as the signal connected to L0 does. Can you use DeMorgan's theorem to justify this algebraically?)

When you are done with this experiment, remove the wires from the breadboard and remove the '02 IC but leave the '00 IC in preparation for the next experiment.

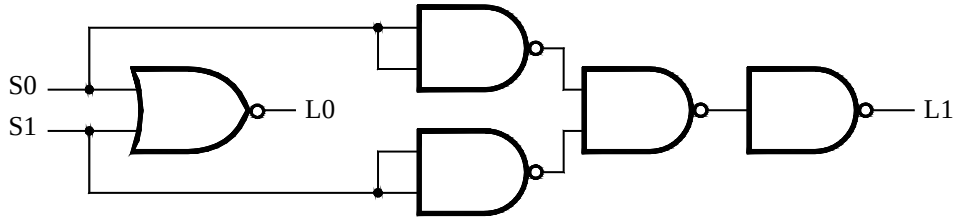

Figure 1. NAND gates can act as a NOR gate.

SIMPLIFICATION OF A LOGIC CIRCUIT VIA BOOLEAN ALGEBRA

As we continue to work with more complicated logic circuits the simulator becomes more of a time-saving tool to verify the correctness of a design. The simulator can be available on any computer, freeing you to test your design outside of a laboratory environment. This portion of the lab will allow you to design a simplified logic circuit (your first logic circuit design) and prove that it works as it should.

Simulate the logic circuit shown in Figure 2. A suggested name for this project is just "LAB2." In the waveform file, be sure the three inputs cycle through all the possible combinations (there are eight of them).

Build the logic circuit shown in Figure 2. You will need to add one '10 IC to the breadboard, but leave enough room for still one more integrated circuit chip to be added later. Connect power and ground appropriately to each chip and apply power for a moment to test for shorts. Then connect the logic wiring. After all the wiring is complete, apply power and obtain the truth table of this logic function by switching through all 8 possible combinations of the inputs. Compare the results to the output of the simulator. They should match.

Either from the logic diagram or from the truth table, obtain a Boolean expression to describe the algebraic function of this circuit. Simplify this expression to sum of products form, negate it twice, and then use DeMorgan's theorem to express the function in terms of NAND operations only. From the simplified equation, derive the schematic for a logic circuit that uses only NAND gates. This circuit ought to be simpler than the given circuit. (Hint: the simplified circuit can be built using only one type '00 chip--just 4 NAND gates.)

Go back to the simulator and add your simplified circuit to the circuit already drawn in the simulation's schematic. Connect the inputs to the same pins the original circuit was connected to, but connect the output of the new circuit to a new output pin, L1. Add L1 to the waveform file too. Run the simulation. The L0 and L1 outputs should be identical if you designed your simpler circuit properly.

Without disconnecting the original logic function, build your simplified circuit next to the original circuit. Connect the new circuit's output to L1. Connect the inputs of your simplified circuit to the same switches (S0, S1, and S2) that provide the inputs to the originally given circuit. For every possible combination of input values, compare the output of the simplified circuit to the given circuit. They should match.

CONCLUSION

The method of handling unused inputs has been illustrated. The simulator has been used to verify a design before building it. Your confidence in using the simulator and interpreting the results has increased.

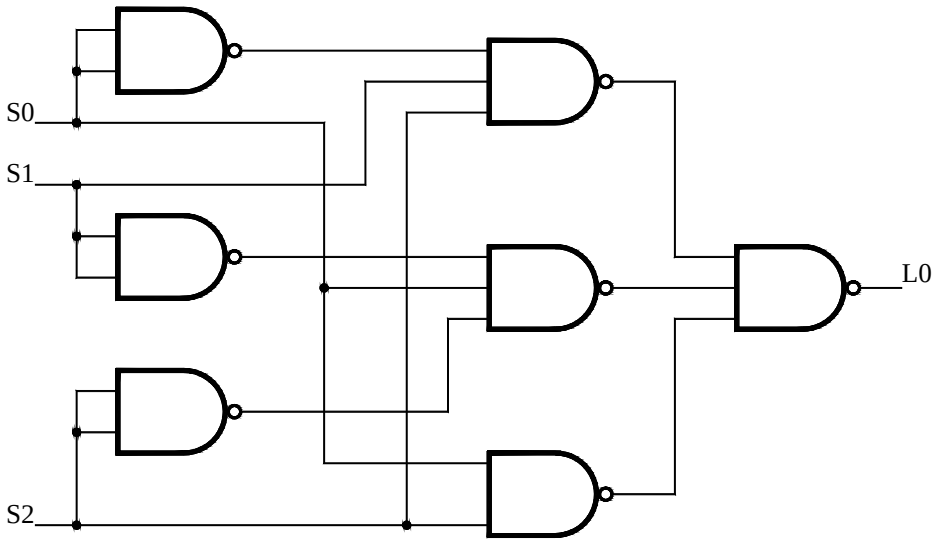

Figure 2. A circuit that can be simplified.

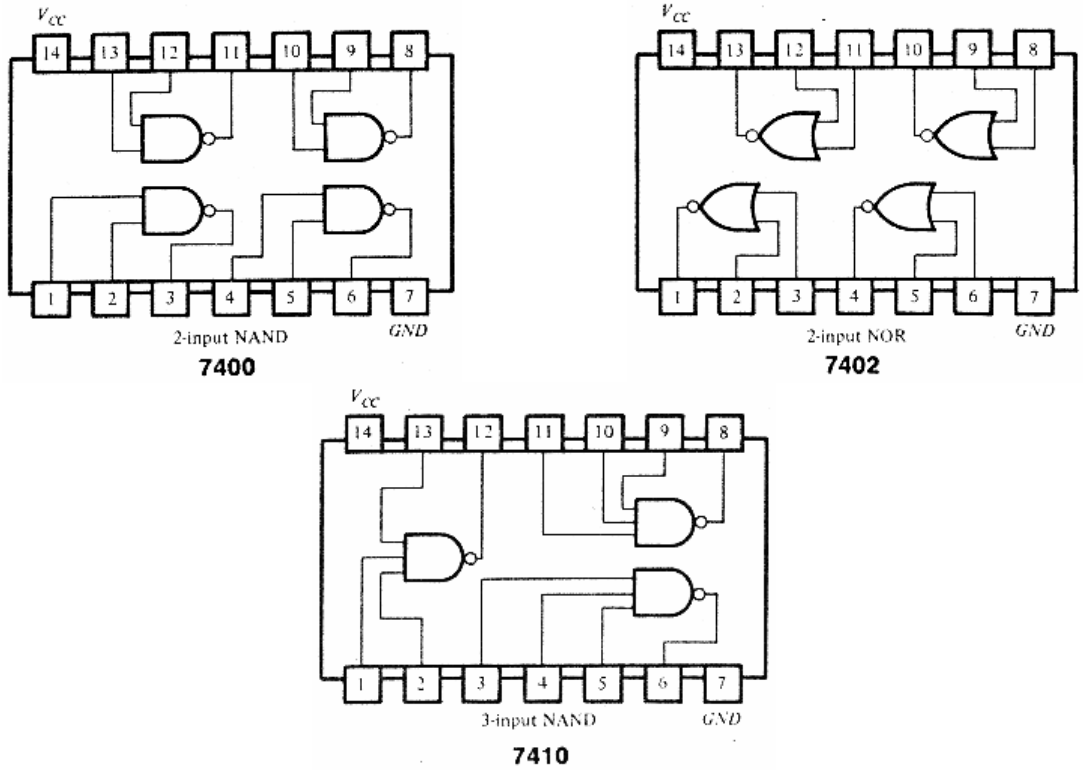

Figure 3. Pinouts of integrated circuits.