Lab 2: NAND and NOR Gates

Introduction

In this experiment, the student is introduced to the operations of multiple input NAND and NOR gates. In addition the student will use these gates to implement logic functions and will demonstrate the value of Boolean algebra in reducing logic circuits to their minimum configuration.

See the previous lab on circuit wiring hints for TTL technology. This information should be documented in the student design notebook. Experimental results form this and all labs should be recorded in the design notebook.

In the early 1960s, **integrated circuits** (ICs) were invented. Transistors, resistors and diodes could be manufactured together on silicon "chips." This discovery gave rise to SSI (small scale integration) ICs. An **SSI IC** typically consists of a 3-mm-square chip of silicon on which perhaps 20 transistors and various other components have been etched. A typical chip might contain four or six individual gates. These chips shrank the size of computers by a factor of about 100 and made them much easier to build.

As chip manufacturing techniques improved, more and more transistors could be etched onto a single chip. This led to MSI (medium scale integration) chips containing simple components, such as full adders, made up of multiple gates. Then LSI (large scale integration) allowed designers to fit all of the components of a simple microprocessor onto a single chip. The **8080 processor**,

released by Intel in 1974, was the first commercially successful single-chip microprocessor. It was an LSI chip that contained 4,800 transistors. VLSI (very large scale integration) has steadily increased the number of transistors ever since. The first Pentium processor was released in 1993 with 3.2 million transistors, and current chips can contain up to 20 million transistors.

In order to experiment with gates, we are going to go back in time a bit and use SSI ICs. These chips are still widely available and are extremely reliable and inexpensive. You can build anything you want with them, one gate at a time. The specific ICs we will use are of a family called **TTL** (Transistor Transistor Logic, named for the specific wiring of gates on the

IC). The chips we will use are from the most common TTL series, called the **7400 series**. There are perhaps 100 different SSI and MSI chips in the series, ranging from simple AND gates up to complete ALUs (arithmetic logic units).

The 7400-series chips are housed in DIPs (dual inline packages). As pictured on the right, a **DIP** is a small plastic package with 14, 16, 20 or 24 little metal leads protruding from it to provide connections to the gates inside. The easiest way to construct something from these gates is to place the chips on a solderless breadboard. The breadboard lets you wire things together simply by plugging pieces of wire into connection holes on the board.

All electronic gates need a source of electrical power. TTL gates use **5 volts** for operation. The chips are fairly particular about this voltage, so we will want to use a clean, regulated 5-volt power supply whenever working with TTL chips. Certain other chip families, such as the 4000 series of CMOS chips, are far less particular about the voltages they use. CMOS chips have the additional advantage that they use much less power. However, they are very sensitive to static electricity, and that makes them less reliable unless you have a static-free environment to work in. Therefore, we will stick with TTL here.

Equipment needed

- Microprocessor power supply
- Digital voltmeter
- Logic probe
- Solderless breadboard
- TTL Integrated circuits (IC), (7400, 7402, 7420)

Procedure

The pin outs for the IC's listed above are shown below (remember – in TTL 14 DIP IC pin 7 is ground and pin 14 is +5V). Make a truth table for each of these gate types.

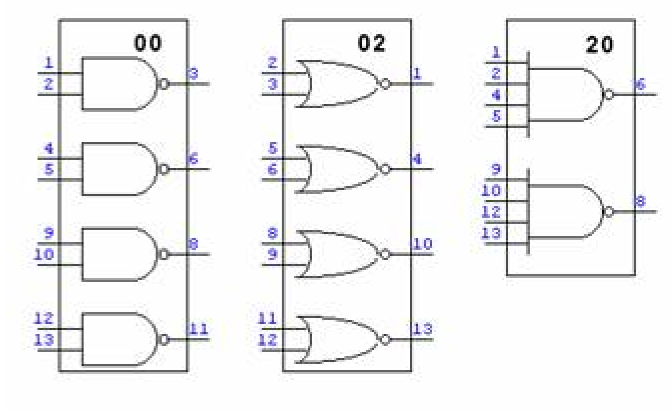
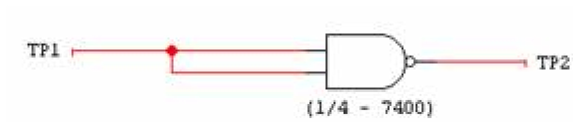


Fig. 1

Part A)

1. Mount the 7400 TTL IC on the breadboard and apply power to the chip according to the pin out diagram (make sure to connect Vcc and GND).

2. Connect one of the four NAND gates in the 7400 as shown in circuit 1.

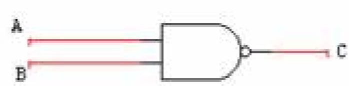


Circuit 1

3. Using the voltmeter record the measured voltages at TP1 and TP2 for all input combinations. Using the Logic Probe, record the measured states at TP1 and TP2 for all input combinations.

4. Draw the function table and truth table for circuit 1. What logic is being performed?

5. Using the same IC (7400) wire circuit 2.



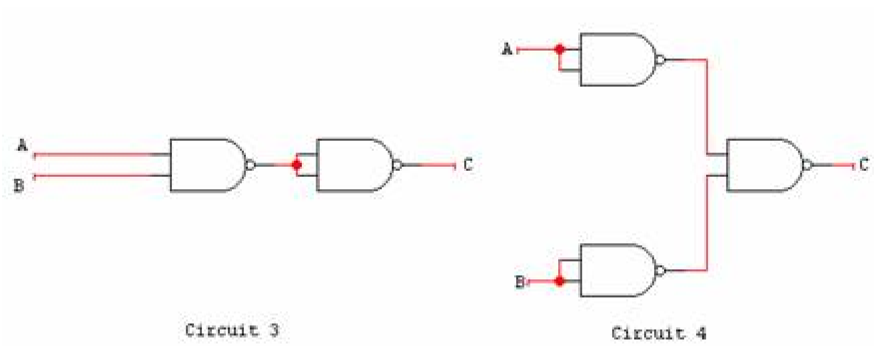
Circuit 2

6. Apply all possible input combinations to circuit 2 and measure and record the output voltage for each set of inputs.

7. Using positive logic convert your electrical truth table into a Boolean truth table and a function table. What function is being performed?

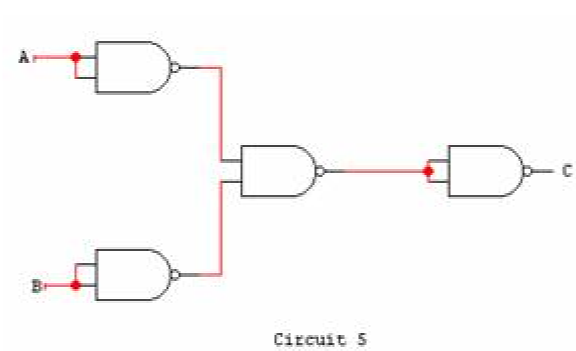
8. Convert the data in the above truth tables to negative logic. What logic function is being performed?

9. Wire circuit 3. Write the Boolean truth table for all input combinations. What logic function is being preformed?



10. Repeat step 9. above for circuit 4. What is the function being performed.

11. Add the fourth NAND gate on the 7400 to the output of circuit 4 as shown in circuit 5. Repeat step 9. for this circuit. What logic function is being performed?

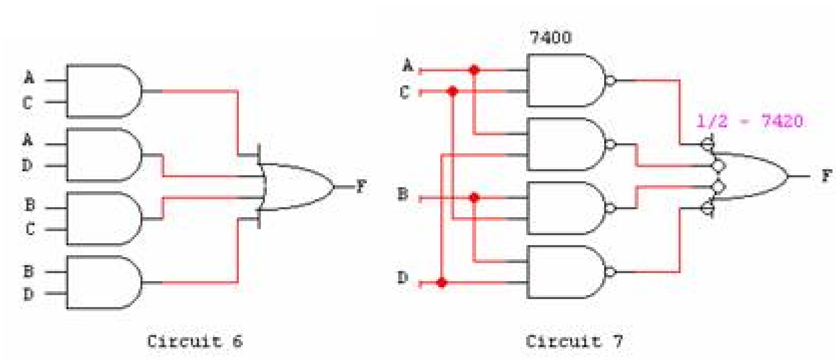


12. Remove the 7400 IC and replace it with the 7402 IC. Repeat steps 1-11 using the 7402 IC. Remember to enter all data into your design notebook (try to summerize). It is this information that will be graded by the instructor.

!<Highly depends if material for this portion can be covered. It will be announced in lab if this section of the experiment is required>!

Part B)

1. Write the output expression of circuit 6 (**do not** build this circuit).

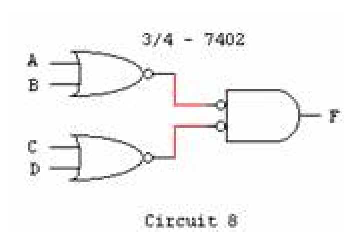


2. Circuit 7 shows the NAND gate implementation of the circuit 6. Construct circuit 7 using the 7400 and 7420 IC's. The pin-outs for these devices are shown in figure 1. Don't forget to provide power to the chip.

3. Analyze circuit 7 by creating a corresponding truth table for the circuit.

4. Using Boolean algebra, reduce the output equation you found in step 1. and verify that circuits 6 and 7 are the same. What is the minimized expression?

5. Construct circuit 8 and write the output equation to describe this circuit. Compare this equation to the minimized expression in Part B step 4.



6. Analyze circuit 8 as you did in step B3 and compare their truth tables. What do you conclude about circuits 7 and 8?

7. Draw the AND and OR gate logic diagram of the expression

$$X = \overline{L}(\overline{K}(K+L)+M) \qquad (1)$$

8. Redraw this circuit using positive logic and only NOR gates.

9. Construct this circuit using the 7402 IC.

10. Analyze your new circuit by creating a truth table for it.

11. Reduce the expression above using Boolean algebra and draw its circuit equivalent of the new expression.

12. Construct this circuit using only NOR gates and compare the output its analysis with the output in step 10.

Discuss the circuits you have constructed in this experiment and try drawing the circuits using the MultiSim program. Make sure all the information gets into your lab report.