INTRODUCTION

In this lab, you will explore the operation of the most popular logic gates and build some useful circuits with them.

We are now beginning the topic of digital circuits in earnest. Most of the labs from now on will use many digital ICs.

Some of the more common pinout diagrams are included in this writeup. Others are available in digital logic data books in the 116 lab.

1. LOGIC GATE TRUTH TABLES

Begin with the basics. Experimentally construct truth tables for the 2 input NAND, AND, NOR, OR, and XOR gates. To do this, find the part numbers and pinouts for each of these gates in a TTL data book. Get one of each IC from the parts bins. Doing one IC at a time, connect the power supplies and test one gate on the IC. Use two of the logic switches for the two inputs. Use one of the LED monitors as the output. Set the switches in all 4 of their possible configurations and record the inputs and output in a truth table. Note that you do not need to record exact voltages; a "1" or a "0" is enough information for a digital system.

For your lab report, give your 5 truth tables along with the logic symbol for each gate. For each truth table, note whether or not it agrees with the expected truth table given in the text or in lecture. (I would expect these to agree.)

2. LOGIC COMPARATOR

The function XNOR = XOR is often called the equality or comparator function. It is similar to the voltage comparator but it compares two digital numbers and tests for equality. For this circuit, let us compare two 2-bit numbers, A and B. The 2 bits of A will be called A1 and A0 for the most significant and least significant bits, respectively, and similarly for B. The circuit shown in figure 1 will compare these two numbers. The Q output will be high when the two numbers are equal.

3. MULTIPLEXER

A multiplexer is a digital circuit that selects one of several inputs. We will make a 4-input 1-bit multiplexer. This means we have 4 data lines, A, B, C, and D, and four control lines, A_c, B_c, C_c, and D_c, one for each of the four data lines. The function we must implement is Q = A_A + B_B + C_C + D_D. Draw a circuit that will implement this function. Wire that circuit using any available ICs and verify its operation. The AND-OR-Invert, or AOI, circuit may be a good choice for implementing this function. For your report, include your circuit diagram and a description of how you tested your circuit.
The multiplexer described in Sec. 3 of Lab 12 has four separate control lines \( A_c, B_c, C_c, D_c \) to select which of the 4 inputs \( I_0 \equiv A, I_1 \equiv B, I_2 \equiv C \) or \( I_4 \equiv D \) is connected to the output (traditionally called \( Y \) rather than \( Q \)). In the new notation,

\[
Y = I_0 A_c + I_1 B_c + I_2 C_c + I_3 D_c
\]

One and only one of the control lines is allowed to be true at any given time.

More commonly, we use two selection lines which represent the binary number of the input to be selected. Let’s call them \( S_1 \) and \( S_0 \) where \( S_0 \) is the least significant bit. Then \( A_c = \bar{S}_1 \bar{S}_0 \), \( B_c = \bar{S}_1 S_0 \), \( C_c = S_1 \bar{S}_0 \) and \( D_c = S_1 S_0 \). For example, to select input \( I_2 \), \( S_1 \) is set high and \( S_0 \) is set low.

Here is a circuit symbol for the resulting 4-to-1-line multiplexer (MUX):

![Circuit Symbol]

Do the following for your lab report.

1. Find a logical expression for \( Y \) as an OR ("sum") of four terms, each of which is an AND ("product") of three variables. (Hint: just substitute the expression in terms of \( S_0 \) and \( S_1 \) for each control variable.)

2. Make a logic diagram for the resulting circuit. You should require 2 inverters, 4 3-input ANDs and a 4-input OR.

3. Show how to implement the circuit using 4 3-input NANDs, a 4-input NAND and 2 inverters.

4. rather than build the circuit on your breadboard, simulate the circuit and verify correct multiplexer operation using the diglog program (see the information on the Physics 116 web site). We will demonstrate the use of this program at the end of the lab period. The program is available on the computers in Room 106 Physics. You can also download a copy

   (a) Include a printout of the circuit diagram from diglog.