Reminder: You must perform parts P1 and P2 before coming to the lab.

EE101 Laboratory 11 (FL03) Name ______________________________

Date ______________________ Partner's name ______________________________

Instructional Objectives (at the end of this lab you should be able to:)

- Research the internet for IC datasheets and circuit diagrams to obtain required information.
- Understand the fundamentals of Boolean algebra as applied to digital logic.
- Construct combinational logic circuits and test to determine the output signal as a function of all possible combinations of three input signals.
- Create a truth table representation of the combinational logic circuit.

Description and Background

Digital logic *gates* are specialized electronic circuits that implement Boolean algebra expressions. Boolean algebra is the language of computer electronics and consists of logical '1' (sometimes called "true", "high", or "on") and logical '0' (sometimes called "false", "low", or "off"). In the electronic circuits in this experiment, we will be using ~5V to represent a logical '1' and ~0 V to represent a logical '0'. This is a common standard; however, other voltage representations for logical '1' and '0' are sometimes implemented.

From the two Boolean elements '1' and '0', all *binary* numbers are determined. In contrast, our common decimal numbering system has 10 elements: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and from these, all our numbers are determined.

In Boolean algebra, three common logical operations are "AND", "OR", and "NOT". Logical AND has similarities to multiplying, and a "dot" symbol such as • is used to indicate AND. The output of an AND circuit is 1 if and only if all the input signals to the AND are 1. For example, a two input AND function gives: 0 • 0 = 0, 0 • 1 = 0, 1 • 0 = 0, and 1 • 1 = 1. Logical OR is represented by the "+" symbol. The output of an OR circuit is 1 when any of the input signals are 1. Thus, for a two input OR function: 0 + 0 = 0, 0 + 1 =1, 1 + 0 = 1, and 1 + 1 = 1. Logical NOT is represented with a prime, " ′ ", or an over bar " ‾ ". NOT operation is the logical inverse of the expression, for instance, 1′ = 0, and 0′ = 1. For many types of physical logic it is convenient to fabricate NAND and NOR circuits rather than AND and OR circuits. NAND means "NOT AND" and NOR means "NOT OR".

A simple Boolean expression might be X = (K • L) + M. A *truth table* can be constructed, as shown using an arbitrary example below, indicating all possible combinations of inputs and the corresponding output.

K	L	M	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

X = (K • L) + M

Equipment

Your own lab kit which includes electronic chips and other components, alligator clips, etc., lab power supply and DMM, plus cables furnished in the lab for connecting to the DMM and power supply.

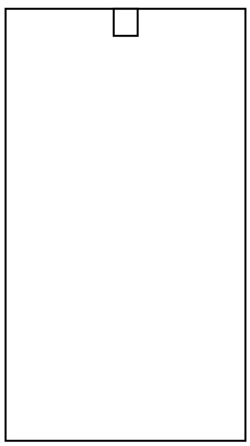
Procedures

<u>You are to complete Procedures P1 and P2 PRIOR to coming to your lab session.</u>

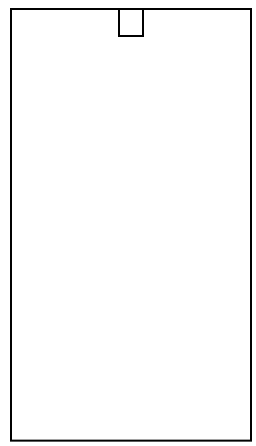
P1. Perform an internet search for manufacturers' datasheets on the two Texas Instruments IC's SN74LS00 and SN74LS02. Log on to http://www.ti.com and perform a product search. Download a copy of the datasheets to obtain pin diagrams, power requirements, etc.

P2. → Using the outline below, draw the DIP package pin diagram <u>including internal gate connections and pin labels</u> for the quad package 2-input NAND gates. → Do the same for the quad package 2-input NOR gates.

NAND



NOR



Make sure you find the portion of the data sheet dealing with the 74**<u>LS</u>** chips.

→ What is the recommended power supply voltage (V_{CC}) that must be provided to the chips before a signal can be applied to the inputs of the NAND or NOR ICs?

V_{CC}: ______________________

→ What are the *typical* and *maximum* propagation delays (including proper units!) for <u>output</u> transitions from LOW-to-HIGH (t_{PLH}) and HIGH-to-LOW (t_{PHL}) for both 74**LS** ICs?

Typical: t_{PLH}: ____________________ t_{PHL}: ____________________

Maximum: t_{PLH}: ____________________ t_{PHL}: ____________________

→ Using the *maximum* output LOW-to-HIGH and HIGH-to-LOW propagation delay times from the datasheets, how long will it take a change in logic of input signal B of Figure1 to propagate through the circuit and cause a change in the output logic signal X?

$t_{P(B\ to\ X)}$: ________________

P3. NAND gate is the logic equivalent to “NOT AND”. The output of a NAND gate is 0 if and only if all inputs to the gate are 1. Similarly, a NOR gate is the logic equivalent to “NOT OR” with the output 0 if any input is 1.

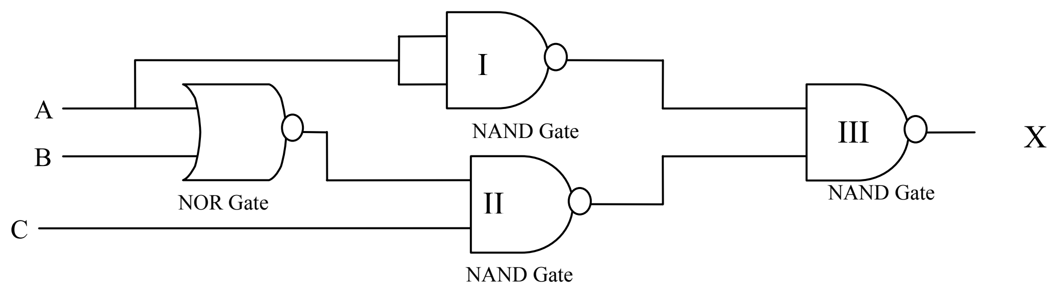
→ Using the two digital logic IC chips in your lab components, the 74LS00N (quad package, 2-input NAND gates) and the 74LS02N (quad package, 2-input NOR gates), construct the circuit shown in Fig. 1. Not all the gates inside the chips will be used.

P4. Use the +5V terminals of the power supply for the needed 5V supply (power and ground) for each chip.

→ Measure and record the output X for all Boolean combinations at the inputs A, B, C, where the inputs are either +5V (for logical 1) or 0V (for logical 0). How many input combinations (A, B, C) will there be? Show your results in the truth table below. The output X can be measured easily with respect to ground using the multimeter set to volts.

→ Now determine the expected (theoretical) output X for each A, B, and C combination using the NOR and NAND logic functions. Show your results in the truth table. Verify that your results match the expected results for this logic combination, and re-check your circuit if there are any discrepancies.

Figure 1:



A (logic 0 or 1)	B (logic 0 or 1)	C (logic 0 or 1)	Measured X (volts)	Boolean X (logic 0 or 1)	Predicted Boolean X