Lab 10: Logic Gates

Object: To acquaint yourself with the properties of And, Nand, Nor, and Exclusive-Or gates, the application of Boolean algebra to digital circuitry, and transistor-transistor logic (TTL).

Concepts: This experiment is designed to reinforce the student's exposure to the following: logic states, gates, truth tables, logic functions, gating networks, DeMorgan's theorem, Boolean algebra, and logical expressions, digital integrated circuits, 7400 TTL logic, and the Digi-Designer.

Apparatus: Digi-Designer, TDS2004B Digital Oscilloscope, 7400 quad 2-input Nand chip

Introduction: The function of any digital system can be understood in terms of four operations: the *And, Or, Negation (Not),* and *Memory* operations. This experiment deals with the first three and the experiment to follow this one deals with the fourth. Actually, the focus of this experiment is on the Nand rather than the And operation due to our choice of 7400-series transistor-transistor-logic (TTL) for this work (see the attached circuit diagram from Simpson) and by the fact that the Digi-Designer manifold upon which all experimental work will be performed is designed for TTL.

Orientation to Digi-Designer: The Digi-Designer provides a collection of logic elements (pulsers, lamp indicators, logic switches, and a clock) on a convenient manifold that contains a +5.0V power supply and a universal strip socket. This array of hardware simplifies the design and testing of simple digital circuits based on ICs belonging to the TTL family. The following discussion is intended to introduce the novice to TTL logic in general and the Digi-Designer in particular.

TTL logic is binary in character — it admits only two possible states called Hi (or "1") and Lo (or "0"). The range of voltage that corresponds to a Hi or Lo differs somewhat depending upon whether one is dealing with TTL inputs or outputs (see Fig. 1). Permanent Hi's are available on the Digi-Designer at the +5V pins above the strip socket; permanent Lo's are available at the ground pins below the strip socket. The LOGIC SWITCHES can be used to provide temporary Hi's and Lo's. A LOGIC SWITCH in the up position produces a Hi (+5.0V) at its output pin while a switch in the down position produces a Lo (0.0V). When the position of the logic switch is reversed, contact bounce in the switch produces a short but random burst of pulses before the final state is attained. Such a burst of pulses is unacceptable in many applications and hence the user is provided with a pair of PULSERS which make clean transitions from one state to another. Each PULSER is dual (it has two outputs); one of these outputs (designated "1") makes a temporary transition to Lo as long as the button is depressed; the "0" output on the other hand, makes a temporary transition to Hi. Pulses of any time duration can be produced. The PULSER Hi is 3.8V and its Lo is 0.0V.

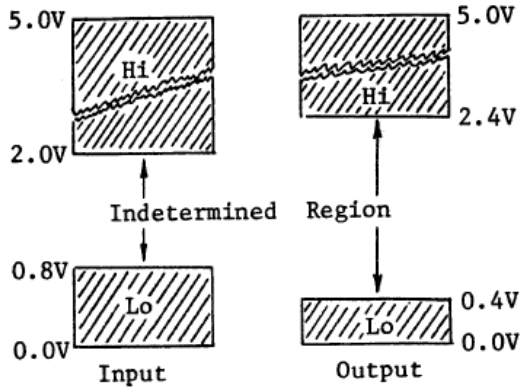


Fig. 1. Voltage ranges for TTL.

If one needs a continuous train of pulses (actually a squarewave alternating between Hi and Lo), one can use the CLOCK. The CLOCK resembles the PULSERS in the sense that it has a pair of outputs that are conjugate to one one other. Six clock rates are available ranging in frequency from about 1Hz to 100KHz. Output Hi's are 5.0V and Lo's are 0.0V. The fourth

type of logic element on the manifold is the LAMP MONITOR. These devices provide convenient indicators of logic states (a lamp "on" indicates a Hi whereas a lamp "off" indicates a Lo). The lamps are buffered by transistors so that they appear to be very light loads.

Wiring on the Digi-Designer: All wiring on the Digi-Designer should be done with #22 solid wire. Wires of appropriate length should be stripped approximately 1/4" at both ends; then the wires are merely inserted into the small holes in the golden binding posts. Wires should never be wrapped around binding posts. Four pairs of binding posts are connected internally to the banana posts to facilitate input and output from the manifold. Posts attached to BNC connectors are available for the same reason.

The strip socket on the Digi-Designer is intended primarily for dual-inline IC packages (DIPs); the socket can hold up to eight 14-pin packages ("chips"). Chips should be inserted very carefully such that they straddle the central groove of the strip socket . When this is done, connections to the leads of the ICs can be made by inserting #22 solid wire (stripped to 1/4") in any of the four remaining holes vertically in line with the appropriate IC pin.

The horizontal rows of holes near the top and bottom of the strip socket are busses. Each of the four rows is electrically independent from the others, but the 25 pins on the left half of any row are connected internally, as are the 25 pins on the right half. There are no internal connections between the strip socket and the Digi-Designer itself. Only newly stripped #22 wire or new components with #22 (or lighter) leads should be inserted into this socket. Wires that have been bent or wires that have rough surfaces due to solder *must not be used*. These sockets are expensive, and the contacts must not be abused.

Measurements. Part (A): As an initial exercise to familiarize yourself with the Digi-Designer as well as a 7400 chip, verify the truth table for a 7400 quad 2-input Nand gate as follows: insert the chip so that the identifying mark (an indentation, small circular well, or short slot) is located on the left as you address the chip and Digi-Designer. Careful adherence to this orientation should minimize confusion regarding pin layout and insure the applicability of Fig. 2. Check the first gate (pins 1, 2 and 3) by using LOGIC SWITCHES on both inputs and a LAMP MONITOR on the output. Check the second gate by using one LOGIC SWITCH and a PULSER for the inputs. Check the third gate (pins 8, 9 and 10) using PULSERS on both inputs, and check the fourth gate by driving it with one LOGIC SWITCH and the output from the CLOCK. In the last case, use a second LAMP MONITOR to observe the state of the CLOCK, and run the CLOCK at 1Hz.

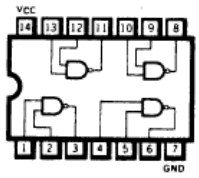


Fig. 2. Pin-out for 7400 chip (top view).

Part (B): Return to the set-up where the LOGIC SWITCHES are connected to the inputs of the first gate. After repeating the original test, disconnect one of the switches and infer the status of a floating input to a TTL gate.

Part (C): Connect one of the inputs of the second gate to +5.0V in order to make this gate into an inverter (make sure you understand why this is true). Then connect the inverter to the output of the first gate so that the composite is an And gate. Verify the operation of this And gate using LOGIC SWITCHES; construct and complete a truth table.

Part (D): In the previous section, you showed that an And gate could be constructed from two Nand gates. In this section, you will construct an Or gate from several Nand gates. The fact of

the matter is that all four of the basic functions — And, Or, Negation and Memory — can be constructed exclusively from Nand gates.

DeMorgan's theorem of Boolean algebra states that

$$\overline{A \cdot B} = \overline{A} + \overline{B} \quad , \quad \text{or equally well,} \quad \overline{A+B} = \overline{A} \cdot \overline{B} \ ,$$

where $\cdot$ is understood to represent the And operation, + represents the Or operation, and the bar represents Negation. In the first of these expressions, $\overline{A \cdot B}$ stands for the Negation of $A \cdot B$, and hence it constitutes a symbolic representation of the Nand operation. The expression $\overline{A}+\overline{B}$, on the other hand, is the result of Or-ing $\overline{A}$ with $\overline{B}$; it is therefore a symbolic representation of the Negated-input Or operation (see Fig. 3). This means that the Nand gate and the Negated-input Or gate should be functionally equivalent and should have identical truth tables. In a similar way, the second part of DeMorgan's theorem

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

suggests that the Negated-input And gate should be functionally equivalent to the Negated-output Or (Nor) gate (see Fig. 3).

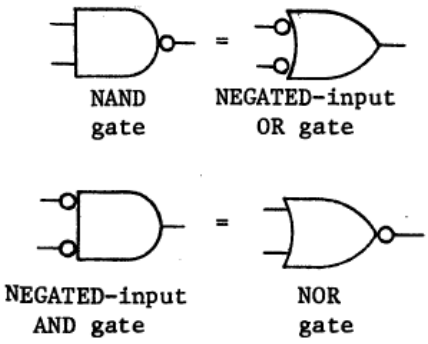


Fig. 3. Functional equalities of pairs of gates.

Finally we take up the Or gate. The Boolean expression

$$A+B = \overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}} \ ,$$

which uses DeMorgan's theorem in the second equality, suggests the equivalence of the Or gate to the Negated-input Nand gate (see Fig. 4). Construct a composite Or gate using three Nand gates. Verify the Or operation by using a pair of LOGIC SWITCHES and one LAMP MONITOR. Construct a truth table and fill it in on the basis of your experimental results. Before disassembling your Or gate, add a fourth gate to the composite to make a Nor gate. Construct a truth table and write down a Boolean expression that describes your Nor configuration of Nand gates.

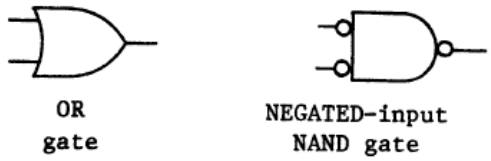


Fig. 4. Another gate equivalence.

Part (E): This exercise is intended to illustrate how a set of logical inputs and associated outputs can be represented by electronic digital logic. You are to assume that some truth table pertinent to a particular task exists, and you want to develop a logic network that will provide the correct output given any particular combination of inputs. The first step in this process is to write down a Boolean expression that represents the desired logic. Then after manipulating this expression, one implements the various steps in the expression with whatever chips one has at his or her disposal. In our case, we assume that we have only Nand gates.

Consider the Exclusive-Or operation represented by $A \oplus B$. The Exclusive-Or is identical to the regular Or operation A+B except when A=1 and B=1, in which case $A \oplus B = 0$ (instead of

1). The truth table for both Ors is shown in Fig. 5. We now proceed to devise a gating circuit that will accept inputs A and B and provide an output $A \oplus B$.

A little thought confirms that the Exclusive-Or operation can be written as

$$A \oplus B = (A+B) \cdot \overline{A \cdot B}$$

which, by distributivity, can be expanded as

$$A \oplus B = A \cdot \overline{A \cdot B} + B \cdot \overline{A \cdot B} \quad .$$

Negating the right-hand side of this equation twice, one gets

$$A \oplus B = \overline{\overline{A \cdot \overline{A \cdot B}} + \overline{B \cdot \overline{A \cdot B}}} \quad .$$

Using DeMorgan's theorem, we have

$$A \oplus B = \overline{\overline{A \cdot \overline{A \cdot B}} \cdot \overline{B \cdot \overline{A \cdot B}}} \quad .$$

Using this expression, construct an Exclusive-Or gate from four Nand gates. Use a pair of LOGIC SWITCHES and a LAMP MONITOR to verify its operation.

A	B	A+B	A ⊕ B
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

Fig. 5. Truth table for regular Or and Exclusive-Or operations. The figure also shows the Exclusive-Or gate symbol.

Part (F): Wire the CLOCK and a LOGIC SWITCH to the inputs of the first gate, but make provision for wiring the output of this gate to the BNC connector, a LAMP MONITOR, and/or all six inputs to the other *three* gates. With the clock set at 1KHz, use your scope to investigate the absolute and comparative loading effects of the gates and the LAMP. Note the input and output voltage levels (especially the voltage of the Lo) and compare to Fig. 1.

Part (G): Measuring the Speed of TTL Gate Response

Method I: A single TTL gate responds too fast to be measured directly with the equipment available to you. But, if you connect eight TTL NAND gates (on two 7400 chips) in series (seven of them as inverters), you should be able to measure the accumulated delay for the eight consecutive gates to switch output states. Wire the circuit shown below on the Digi-Designer using one of the BNC in/out connections to connect the "TTL out" signal from the function generator (also 'T' the input to the oscilloscope channel 1), and the other to connect the output of your circuit to the oscilloscope.

- Before testing the circuit, analyze the circuit function and predict its behavior 1) if the switch is in the LOW position, and 2) if the switch is in the HIGH position.
- Turn on the function generator and test your prediction.
- Set the function generator to its highest frequency and attempt to measure the response time of a single TTL NAND gate.

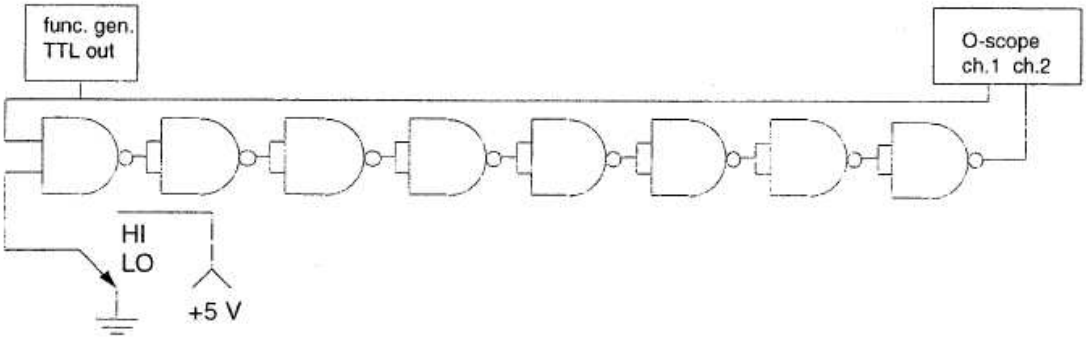


Fig. 6: Circuit for measuring TTL gate response by method I.

Method II: Remove the function generator connection and one of the NAND gates. Connect the output of the seventh gate back to one of the inputs on the first gate as shown below. Also connect the output of the seventh gate to the oscilloscope.

- Before testing the circuit, analyze the circuit function and predict its behavior 1) if the switch is in the LOW position, and 2) if the switch is in the HIGH position. Why is it necessary to remove one of the NAND inverters to make this measurement?
- In one of the two switch positions, you should conclude that the output oscillates with a period equal to the total response time of 14 TTL NAND gates. Verify this prediction experimentally and use it to measure the response time of a single TTL NAND gate. Compare the response time to the value obtained by Method I. Which measurement has a smaller uncertainty?

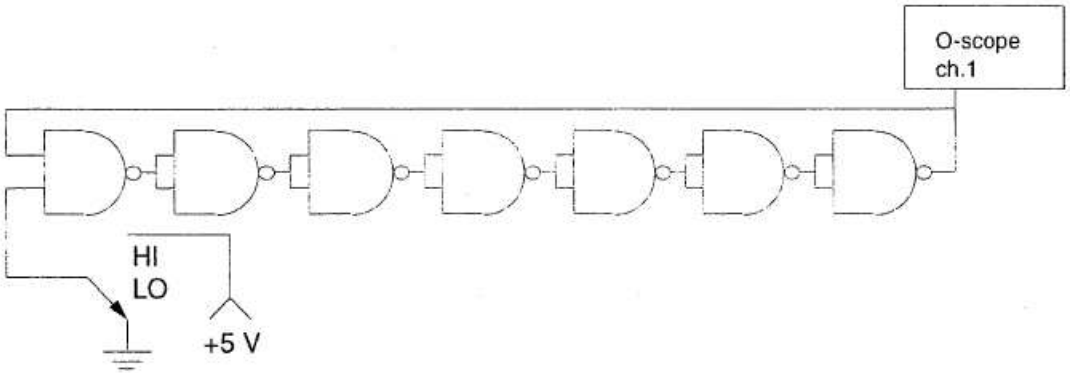
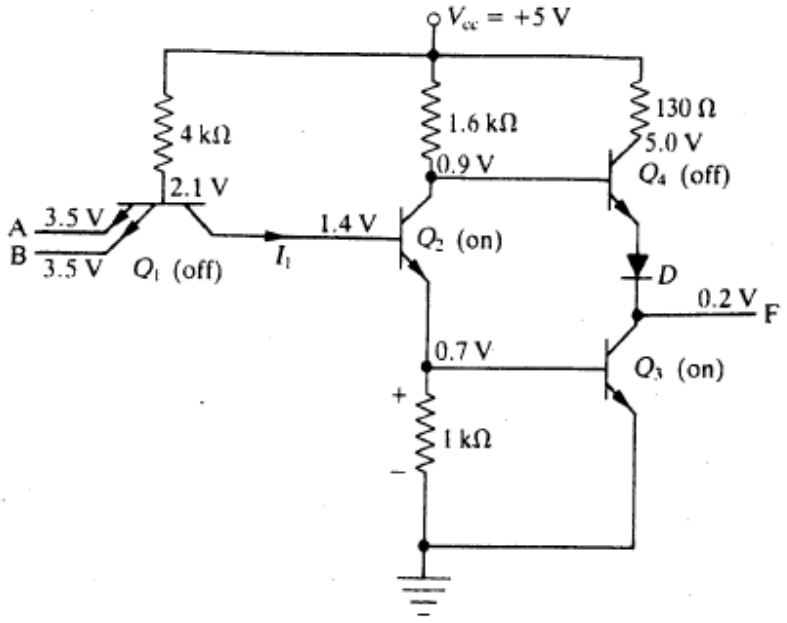
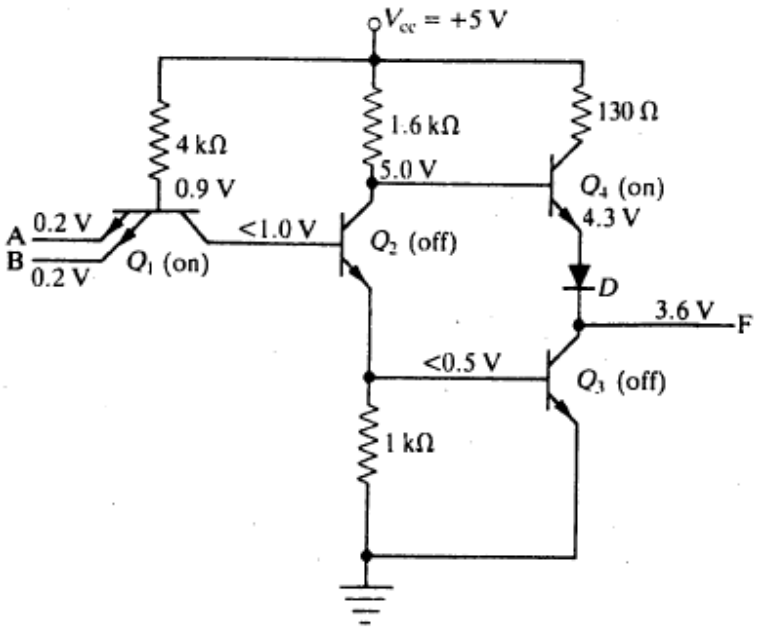


Fig. 7: Circuit for measuring TTL gate response by method II.



(a) A = 1, B = 1; F = 0



(b) A = 0, B = 0; F = 1

FIGURE 12.26 Two-input 7400 TTL NAND gate.

Figure borrowed from Introductory Electronics for Scientists and Engineers, 2nd Ed., by Robert E. Simpson, Allyn and Bacon (1987).