Logic Gate Basics

April 18, 2013

Logic gates are devices that take an input, interpret it as either high (1) or low (0), and then output a corresponding signal based on what kind of gate it is. Most gates are TTL (Transistor-Transistor Logic) compatible, meaning they work on a 0-5 volt scale. A general TTL low is between 0 and .8 volts and a general high is between 2 and 5 volts. This switching behavior is achieved using a network of transistors within the gate that are arranged to preform certain logical operations: AND, NAND, OR, NOR, NOT and other more complicated combinations. All logical operations can be created using a combination of AND, OR and NOT operations.

NOT gates

The first experiment to perform is to test the switching voltages of a 74AHCT04 Hex Inverter, a chip with six logic gates that perform the NOT operation. Find the datasheet with the pin connection diagram on the Web. After placing the the chip in your "breadboard", apply a 5 V_{PP} ramp wave as the input and measure the output on your oscilloscope in the usual way – you may also capture the output in XY mode. Use two different frequencies of the input signal, e.g., of 1 kHz and another of 250 kHz, to check how the gate behavior and the parameters of the output signal depend on the input frequency – in other words, to check "how fast" the gates are able to operate.

In particular, determine the input voltages for which the gate switches from the "logical zero" to "logical one" and from "logical one" back to "logical zero", and check whether the values you obtain are in agreement with the "manufacturer's list guaranteed values".

In the second experiment with the NOT gates, investigate the behavior of a "chain" of gates put together in series. First, put together two, three,..., etc. gates and check whether the output of the "last in chain" agrees with your expectations. Then, consider a situation when the output of the last gate is connected with the input of the first one, i.e., when the gates make a "closed ring". Begin your analysis by considering just one gate. Then add another one, and so on.

A closed "ring" of NOT gates may act either as an "oscillator", or as a static "memory cell" capable of storing a single logical bit. How does the circuit behavior depend on the number of gates in the "ring"? Try to predict the behavior, and then check your predictions experimentally.

One more interesting task will be to build a quartz-stabilized oscillator using NOT gates. However, it may take some time to persuade the circuit to start oscillating, so it may be perhaps a good idea to work on this task after finishing all experiments with NAND gates we want you to perform.

NAND gates

We want you to perform several experiments on a single NAND gate. Ciruits will be drawn on the blackboard, and details will be explained by the instructors.

Then, we would like you to combine several NOT and NAND gates to make a "detector of binary number 0110" – i.e., a four-input circuit that would output a 1 only when addressed by the four-bit number 0110, and output 0 for all other possible input combinations.

Finally, we would as you to construct and test a XOR gate using only NAND gates (as you perhaps remember, building such a gate "on paper" was one of the problems in the final exam of the Ph412 Course – now we want you to build and test the real thing).