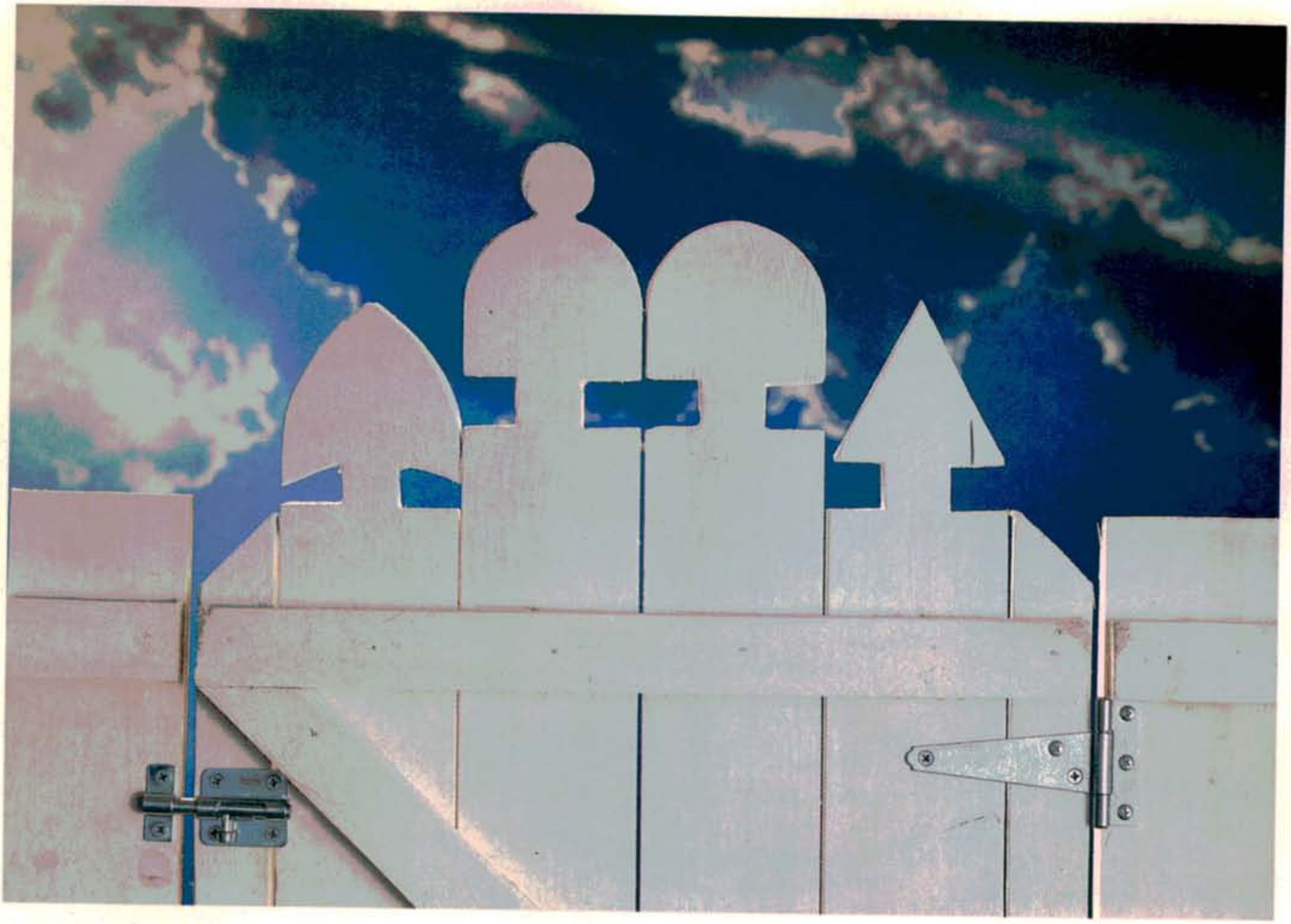


Logic Gates

The decorative patterns scrolled at the top of our gate are some of the lesser known symbols found in schematic drawings, and are used to indicate the Logic Gates found in computers. Logically enough, these symbols are showing up in more and more automotive schematics. Maybe you've seen them already, and wondered what they were. Understanding what these symbols stand for is an aid to understanding computer controls

Over the past few months we've been discussing things like analog inputs to computers, and the importance of computer switch outputs. But there's still a very important piece of the computer puzzle missing—the Logic Gate.

The term Logic Gate can be a little awe inspiring at first. But logic gates are really quite—well, logical. Basically, they are nothing more than switches which act like gates to control traffic inside the computer. Each type of logic gate does a different job, and each has its own "combination," used to open or close it, similar to the tumblers on a combination lock.

To open or close a logic gate, you need to know the combination of signals used by that gate. But before we get into that, let's look at digital signals.

Highs and Lows

Let's get more specific about logic level terminology. You have two levels, or logic states called "1" and "0" (say zero). What's a "0"? It's nothing, no voltage, a LOW, ground voltage, zippo, zilch, zero, to give it a few names.

A "1" is the opposite of "0." It is a voltage at a level designated by the manufacturer. Simply stated, we have a HIGH (voltage) or a LOW (no voltage).

From Vocabulary to Language

There are other forms of communication which use simple combinations of HIGHs and LOWs, similar to the HIGHs and LOWs used by computers. We use different combinations of raised dots on paper to allow a blind person to read using a system called Braille. In a similar way, we use different combinations of dots and dashes to communicate in Morse code.

It's the combination and the sequence of HIGHs and LOWs, not the HIGHs and LOWs themselves, which allow the computer to "think."

Managing Traffic

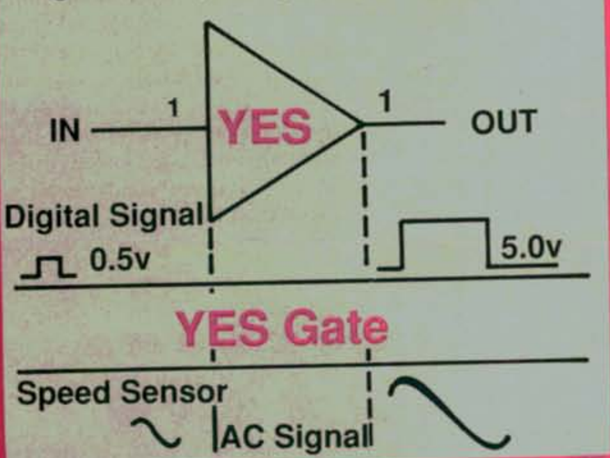
So how does the computer keep track of all this HIGHing and LOWing? Logic Gates. Logic gates are the traffic cops opening gates, or closing them to control computer signal traffic. These traffic cops respond to inputs from sensors, decide if the inputs are the right combination, and then they either open or close the gate.

These logic gates teach us something about computers that we should never forget. We'll tell you what that is before we're done. In the meantime, let's look at some examples of logic gates, and ways to recognize each gate and the job it does. While there are other logic gates to consider, we'll concentrate on the more common gates you're likely to see.

The YES Gate

Our first symbol is a simple triangle which we'll call the YES gate. One of the jobs this gate can do is to act as an amplifier. In this case, the logic gate works like the amplifier on your stereo, converting a weaker incoming signal to a stronger outgoing signal. This is the most basic of all logic gates.

Figure 1 Op-Amp Amplifies Output



Logic gates are made up of at least one transistor and resistors to make a complete amplifier circuit. Using this triangle eliminates the need to draw each of those internal components separately. When you see the triangle, you know what it does in the circuit (at least you will in a minute).

In **Figure 1**, what goes in looks like what comes out. The only difference is that as the signal passes through the logic gate, it is amplified (made stronger) in terms of voltage and current. The YES gate increases the power level. This is necessary because weak signals are easily lost or distorted by noise or interference. Increasing the power level of the signal makes it more stable and effective in the circuit.

One good example of this function is the way the YES gate takes a low level signal from a magnetic speed sensor, and amplifies, or strengthens that signal before sending it on to the computer.

If the incoming signal is an AC signal from a magnetic speed sensor, the YES gate will amplify the AC signal.

On the other hand, if the input is a digital input from a Hall Effect Switch, a "1" (HIGH voltage) is applied to the input which will also appear at the output side as a "1." Our weaker incoming signal is received as a "1" before being amplified into a stronger "1." We can measure a logic "1" with a logic probe, a DVOM (assuming the signal lasts long enough for the DVOM to capture it), or with a digital scope.

The triangle-shaped logic gate can also act as a BUFFER. In this use, it acts as a gate, and separates two circuits to keep one circuit from affecting the other. The logic gate may or may not act as an amplifier when it acts as a Buffer.

Auto manufacturers can use the logic gate to do both jobs. The gate can amplify a signal, or it can isolate two circuits acting as a Buffer. Then they call it a Buffer Amplifier or Buffer-Amp.

You don't necessarily need to dig into the computer's guts to test the gate. In fact, you may see these types of gates used in circuits outside the computer. You can check a Buffer Amp by checking its input and output. If the output is missing when the input is present, check to see if the Buffer Amp is getting both good operating voltage and good ground. If you have good voltage and ground, and the input is present (example: a magnetic sensor input) and still have no output, the Buffer-Amp is bad.

On the other hand, if the magnetic sensor input is bad, go back to the sensor itself and determine why it isn't sending a signal. The Buffer-Amp can't give you an output without a signal input. This is one place where a Logic Probe is a great tool for quick checks of signals in and out of the Buffer-Amp.

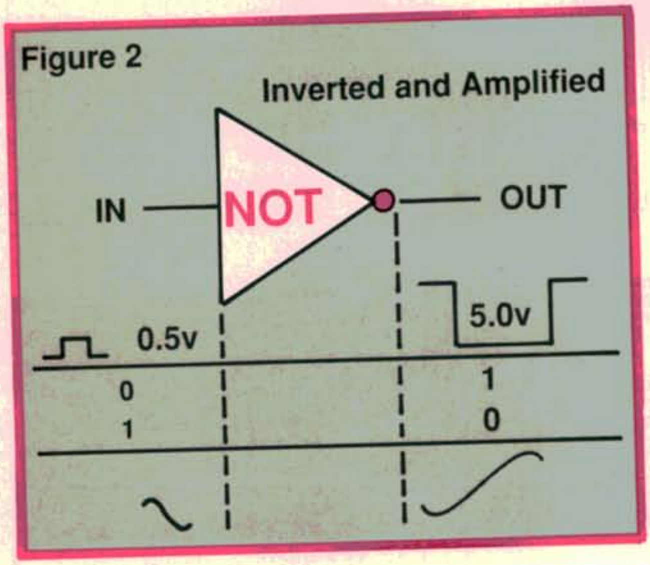
Logic Gates

The NOT Gate

Next on our list is the NOT gate. It's a variation on the basic logic gate we just discussed. It has one input. But you'll notice that our symbol now has a small circle at the output end. Think of this circle as an instruction to the gate to "do the opposite." In other words, the signal coming out of the NOT gate is NOT the signal coming in—it's just the opposite. The NOT gate inverts the input, and sends out the opposite signal to the output. If the input is a "1," then the output will be a "0." If the input is a "0," then the output will be a "1." This is just the opposite of the YES gate. Notice in **Figure 2**, that all inputs turn into the opposite signal at the output of the NOT gate.

Blame it on the little circle at the output.

Figure 2



In addition to inverting the output signal, the NOT gate may also amplify the output signal. Whether or not the signal is amplified depends on whether the NOT gate is an Amplifier, a Buffer, or both. Remember, with a NOT gate, the input logic level is inverted at the output (converted to the opposite signal).

Gates With Two Inputs

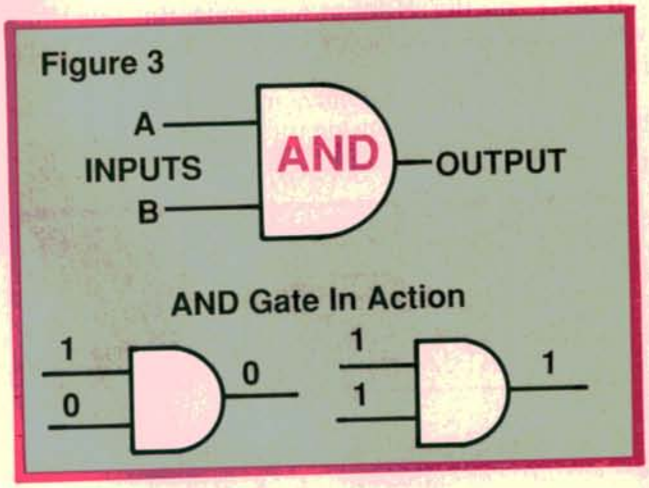
Now that we've looked at the first two basic types of logic gate, let's move on to slightly more complicated type of gate which needs two input signals. The AND gate is shown in **Figure 3**. An easy way to remember the symbol for the AND gate is to look at the symbol and notice how it resembles the "D" in the word AND.

The normal output from an AND gate is a "1." An AND gate takes two inputs to unlock it, and both inputs must be correct. In an AND gate, both A and B inputs must be in the HIGH or "1" condition to get a "1" at the output.

What if you have a "1" and a "0" at the inputs?
What if you have a "0" and a "1" at the inputs?
What if you have a "0" and a "0" at the inputs?
In any of these cases, there will be no "1" at the output which will stay LOW or "0." That's because none of these is the right combination for the AND gate.

Only a "1" and a "1" will open the AND gate. In this case "1" plus "1" does equal "1." The AND gate may have more than two inputs, but it will ignore them until the right combination of "1" and "1" open the gate.

Figure 3



Think About Relays

Some of you are suspicious about this whole computer thing. Can't blame you. But testing of logic gates doesn't necessarily mean digging into the printed circuits inside the ECU.

Do you disassemble an old electro-mechanical relay to test it? Nope. You find out which values are needed to energize the relay and you start by testing for those values at the relay plug. You may even use a couple of jumpers to see if the relay clicks on and off as it should.

If it clicks, you test across the input and output terminals for continuity or an open. And you also know the difference between a Normally Open and a Normally Closed Relay.

Maybe we should start thinking of logic gates as electronic relays so small that we could place more than a thousand on the period at the end of this sentence. Size has nothing to do with testing either electro-mechanical or transistorized "relays" since both are eventually connected to an input wire or an output wire. That's where you test.

If we know how the gate works, we can test inputs and outputs, compare the values to the truth table for that type of gate, and determine whether or not the gate is faulty without ever seeing inside the computer.

The NAND Gate

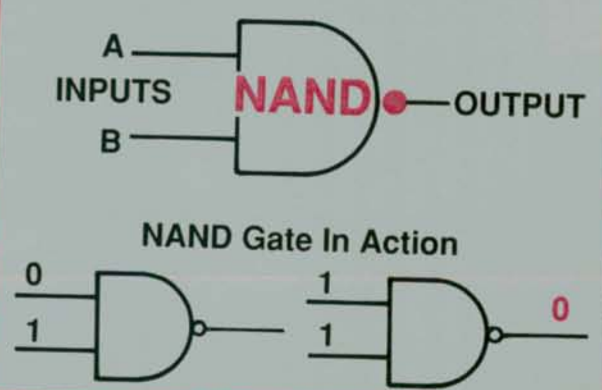
The symbol for the NAND gate is shown in **Figure 4**. An easy way to remember the NAND gate symbol is to look at the symbol and notice that it's an AND gate with that little circle on its nose. This time the circle means that the normal operation of the AND gate is inverted. This makes NAND gate output the opposite of AND gate output, even though the "1" and "1" input signals needed are the same as those used by the AND gate.

In this case "1" plus "1" equals "0."

Like the AND gate, the NAND gate needs an input of "1" on BOTH input terminals. But this time, when the gate switches, we get a LOW or a "0" at the output.

The NAND gate may have more than two inputs, but it will ignore the wrong inputs until "1" and "1" produce a "0" at the output.

Figure 4



A Simple Experiment

Over 20 years ago, Volkswagen used a control box in its D-Jetronic system. Although not as sophisticated as newer computers, it performed many similar input/output functions. Unfortunately the boxes were mounted in the quarter panel on Type III vehicles, and were exposed to a lot of moisture.

At first, many of the problem cars were repaired by installing a new box which seemed to fix a majority of the cars. Then one day, a tech decided to try a simple experiment. He removed the box from a problem car and switched it with a car in the next bay which ran just fine. After completing the transplant operation, both cars ran as they should.

The tech suspected that the unplugging and replugging of the control box had cleaned the mating contacts between the box and the receptacle. He thoroughly cleaned all the contacts on the control box and mating plug, reinstalled the original box in the original car, and again it ran perfectly, thank you.

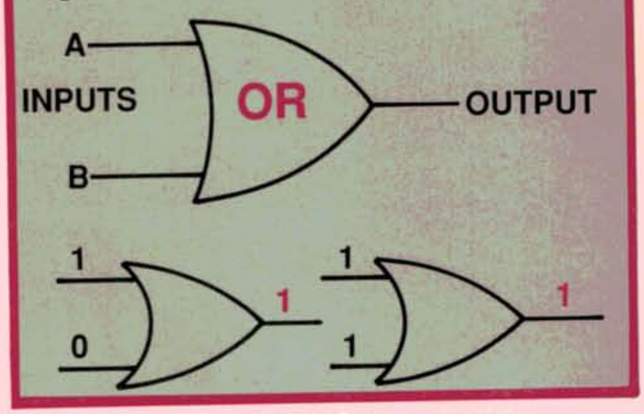
Months later, the parts department had stopped stocking control boxes, but saved the space on the shelf for a case of spray contact cleaner.

The OR Gate

The symbol for the OR gate is shown in **Figure 5**. The OR gate symbol is shaped like the head of a shovel and you use a shovel to dig for ORE. (Maybe it'll help you remember the OR gate symbol.)

If the A input or the B input is "1" (HIGH), or if both A and B inputs are "1" (HIGH), then the output will also be "1" (HIGH). This is the normal output from an OR gate.

Figure 5

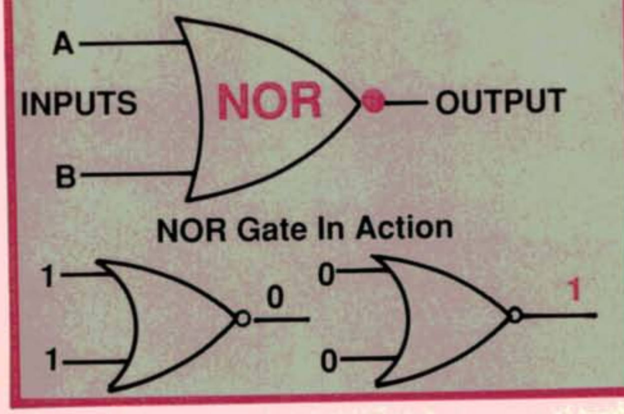


The NOR Gate

The symbol for the NOR gate is shown in **Figure 6**. As it did with the AND and NAND gates, adding the little circle to the output side of the OR gate makes it a NOR gate. And once again, while the inputs needed to switch the OR and NOR gates are the same, the normal output is no longer a "1," but a "0".

A NOR gate will produce a "0" output if either A or B inputs are a "1." It will also output a "0" if both A and B inputs are a "1." The only way you'll get a "1" output is if both inputs are "0." This is just the opposite of the OR gate.

Figure 6



What the Heck Does This All Mean?

We told you there was a valuable lesson to be learned about computerized systems. Here it is: Logic gates only give the correct output when they receive the correct inputs. If a computer doesn't get good inputs, it can't give you good outputs.

There is an old saying in the computer industry, "Garbage in—Garbage out." That means that if you send bad messages (garbage) to the computer inputs, you'll get garbage outputs. The same thing applies to the computer in the car. If you have garbage inputs from faulty sensors or bad connections, then you'll have garbage outputs no matter how good the computer is. Keep this in mind when you diagnose driveability problems which haven't set a code.

If the computer inputs are even slightly out of whack the poor computer can't possibly give you the right output signals. Your job is to correct those garbage inputs. A little practice with logic gates and truth tables will help you eliminate garbage inputs. You don't need to be a Vulcan with pointy ears to be logical.

—By Vince Fischelli