THINK TANK

By John J. Yacono

Opening-up Your Gates

By way of introduction, my name is John and happily I've been given the responsibility for *Think Tank*. I've been an avid electronics tinkerer since I was a kid. To me there's nothing as much fun as being creative, except sharing creativity. So I'm really looking forward to reading and presenting your letters, and throwing in a few ideas of my own—so let's get to it.

Before we jump into the mailbag, I'd like to share a little off-the-beaten-path knowledge with you regarding old TTL logic gates. Because of their internal circuitry, you can sometimes reduce the number of TTL gates needed to do a certain job in an unusual yet simple way, as I'll show you.

I'll use the common NOR gate as an example. As you might know, you can wire-up NOR gates in a variety of ways to make them simulate other gates. The standard symbol for a NOR gate is shown in Fig. 1A and its truth table is shown in Fig. 1B. In negative-logic terms, it is like a negative-input AND gate (see Fig. 1C).

The simplest gate a NOR can imitate is an inverter, as shown in Fig. 2A. You just apply the input signal to both gate inputs, and its logical complement (inverse) will appear at the output (work out the truth table to convince yourself). Of course, you can combine such an inverter with a NOR gate to get an OR gate, as shown in Fig. 2B.

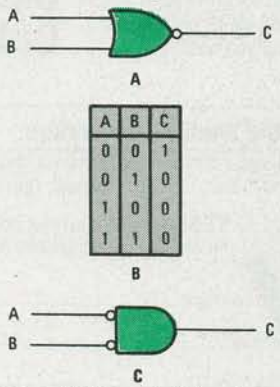


Fig. 1. A NOR gate (A) only goes high when both inputs are low, as shown in B. That is just what a negative-logic AND gate (C) does.

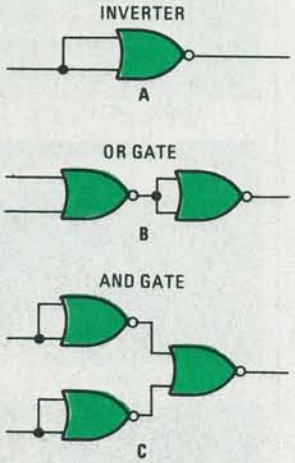


Fig. 2. You can easily make a NOR gate act like an inverter by connecting its inputs to the same signal (A). Coupling an inverting NOR gate to a plain NOR gate will cause it to act as an OR gate (B). By simply inverting its inputs (as shown in C), we can transform a NOR gate into an AND gate.

As I mentioned before, a NOR is just an AND with inverted inputs. So if you invert the inputs to a NOR gate, you'll get a regular AND gate. The resulting circuit using only NOR gates is shown in Fig. 2C. We can take the process a step further by adding an inverter to the output to get a NAND gate (see Fig. 3).

Building XOR and XNOR gates is a little trickier, so we'll just present the two circuits in Fig. 4. Note that the XOR circuit in Fig. 4A was implemented using five NOR gates. That's an unfortunate number of gates; most gates are available in IC packages that contain four gates each. That means you would need two IC's, but only use five out of the eight gates they provide.

However, there is a way to use only four gates provided that you have some really old TTL gates around. Some good candidates are the MC1010, MC1012, MC1210, and MC1212. Such gates use a negative power-supply voltage. They represent a logic high by a voltage a little less than zero volts, and a logic low with around −2 volts. However, if need be, such chips can be interfaced with modern circuits by connecting their ground terminal to 5-volts, their supply-voltage pin to the circuit ground, and using pull-down resistors at the outputs.

Those chips have a particularly interesting property: you can OR two gate outputs by just tying them together! In practical terms, that means you could replace the circuit in Fig. 4A with the circuit in Fig. 5. So you can implement an XOR gate with only one old quad NOR-gate chip and no waste.

No presentation of NOR-gate equivalents would be complete without mentioning R-S latches. The simple R-S data latch is shown in Fig. 6A, along with its truth table in Fig. 6B. A clocked version of the R-S latch is shown in Fig. 7A, and its truth table is presented in 7B. Now let's get to that mail.

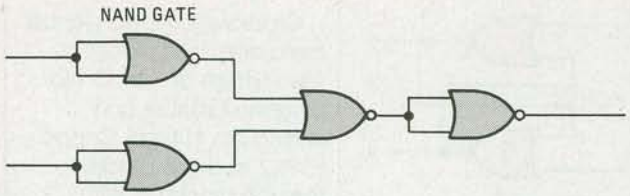


Fig. 3. By adding one more inverter, we can take or equivalent AND *gate a step farther to turn it into an* NAND *gate.*

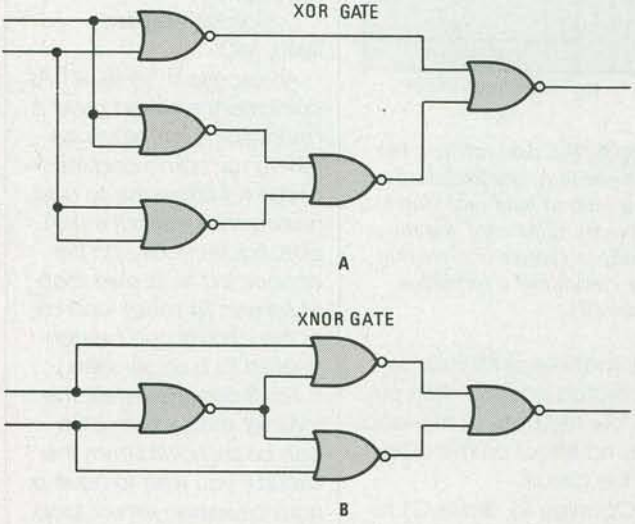


Fig. 4. The logic required to make an XOR *is not only a little complex, it requires five gates, as shown in A. The logic required to make an* XNOR *gate from* NOR *gates is a little complex, but we show such a circuit in B.*

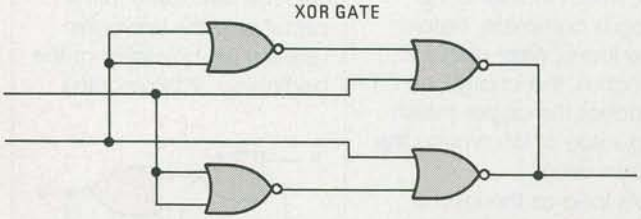
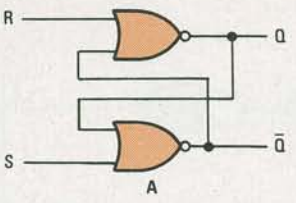


Fig. 5. If you use old TTL chips, you can fabricate an XOR *from only four* NOR *gates. Note that the result of the two tied outputs is the* OR *of their values.*



A

R	S	Q	$\bar{Q}$
0	0	Q_0	$\bar{Q}_0$
0	1	1	0
1	0	0	1
1	1	FORBIDDEN	

Q_0 – PREVIOUS VALUE

B

Fig. 6. R-S data latches, like the one in A, are useful when you want to hold onto (latch) a binary value until you are ready to change it. Note that the circuit has a forbidden state (B).

off. That keeps K1 in its deactivated position. With pin 13 low, the state of the loop has no effect on the output of the circuit.

Opening S1 allows C1 to charge. The charging time of that capacitor provides you with some time to exit the house. You should check to see that LED1 is on, which indicates the loop is complete, before you leave. After about 22 seconds, the charge on C1 reaches the upper threshold value of U1, arming the alarm circuit.

As long as the loop is closed, the bistable latch made of the cross-connected NAND gates keeps pin 11 high and C2 charged. If the loop is opened, U1-a will place a low on pin 8, flipping the state of the bistable latch. Even if the loop is closed again, the circuit would remain in the "flipped" state. With pin 11 now low, C2 discharges through R3, which takes about 22 seconds (which gives you a chance to disable the alarm when you get home). When the lower threshold of U1-d is reached, its output will go high turning on Q1, which activates the relay, sounding the alarm.

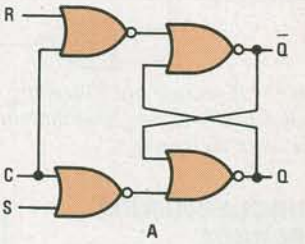
Close switch S1 to silence the alarm. That places a low voltage on pin 13, flipping the bistable and setting pin 11 high. Capacitor C2 charges quickly through diode D1. With C2 charged, pin 4 goes low, cutting off transistor Q1, and relay K1 drops out silencing the alarm.

—Gordon Reeder, Rolla, MO

An excellent "re-think." As coincidence would have it, I'm installing an alarm system in my home and this circuit will allow me to add more zones without much cost. Each circuit can be connected to its own loop of sensors (a zone), and all of the circuits can be connected to a single siren.

For those interested, the inverter made from U1-a can be removed from the circuit if you wish to have a normally-open sensor loop. However, note that LED1 will be off until there is a breach of the zone.

Note the similarity between the NAND inverters and the NAND latch in this circuit and the NOR-gate versions we provided at the beginning of this month's



A

R	S	C	Q	$\bar{Q}$
–	–	1	Q_0	$\bar{Q}_0$
0	0	0	Q_0	$\bar{Q}_0$
0	1	0	1	0
1	0	0	0	1
1	1	0	UNKNOWN	

B

Fig. 7. This latch (A) is similar to the simple latch in Fig. 6A, but, as you can see from the truth table (B), the circuit ignores any input until the C (clock) input is low.