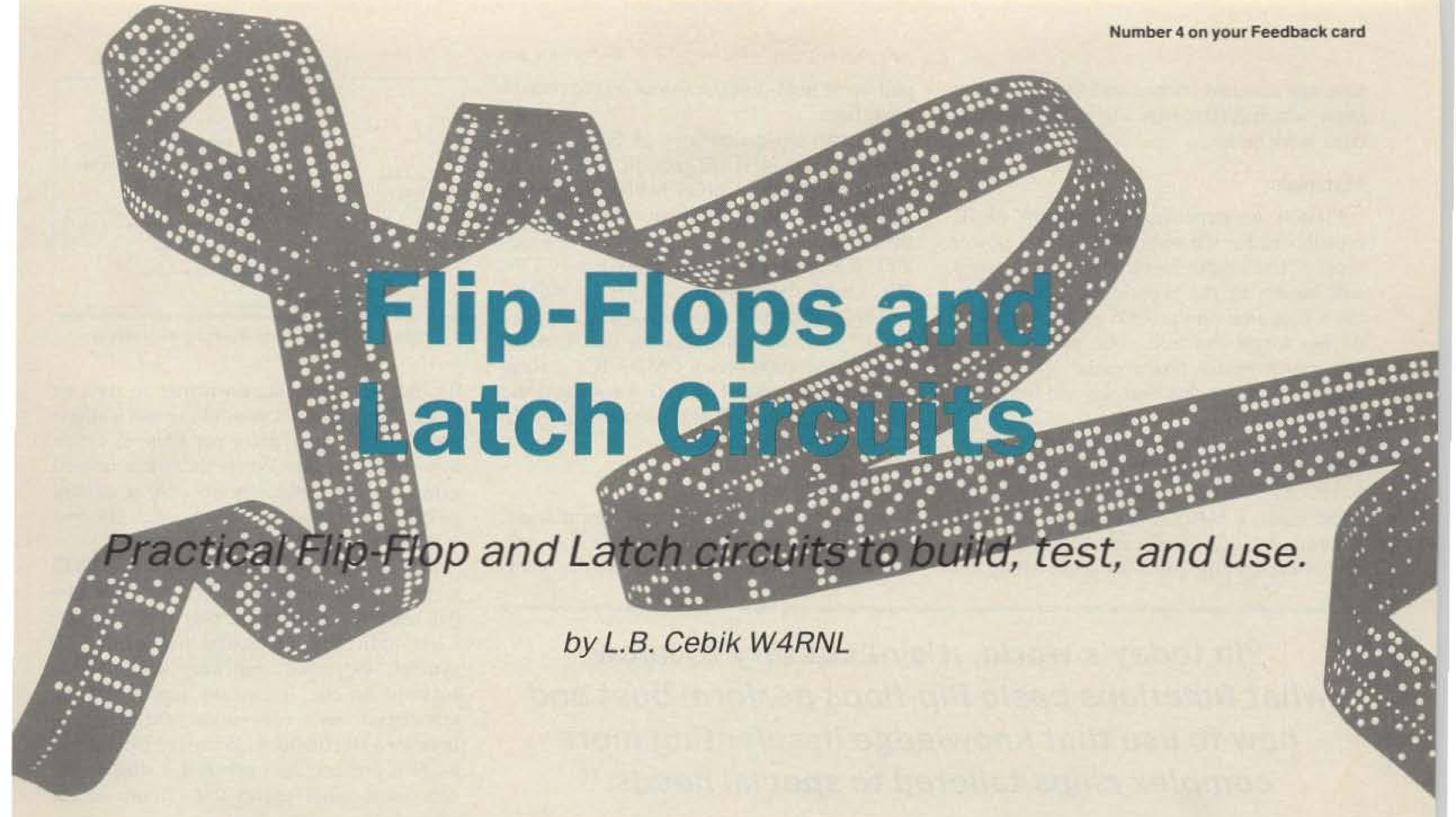

Flip-Flops and Latch Circuits

Practical Flip-Flop and Latch circuits to build, test, and use.

by L.B. Cebik W4RNL

Why Know Flip-Flops and Latches?

The flip-flop is one of the most versatile chips in the entire digital IC family. "Flip-flop" is actually the name of a circuit configured in many different ways; for example, with transistors or gate chips. A more formal name for the circuit is the bi-stable multivibrator. The more informal name reflects the fact that modern ICs add special features to the basic circuit to make it even more flexible. It's possible to clock in data, preset the outputs, clear the outputs, and feed the output back to the input. These features allow the creation of many useful circuits.

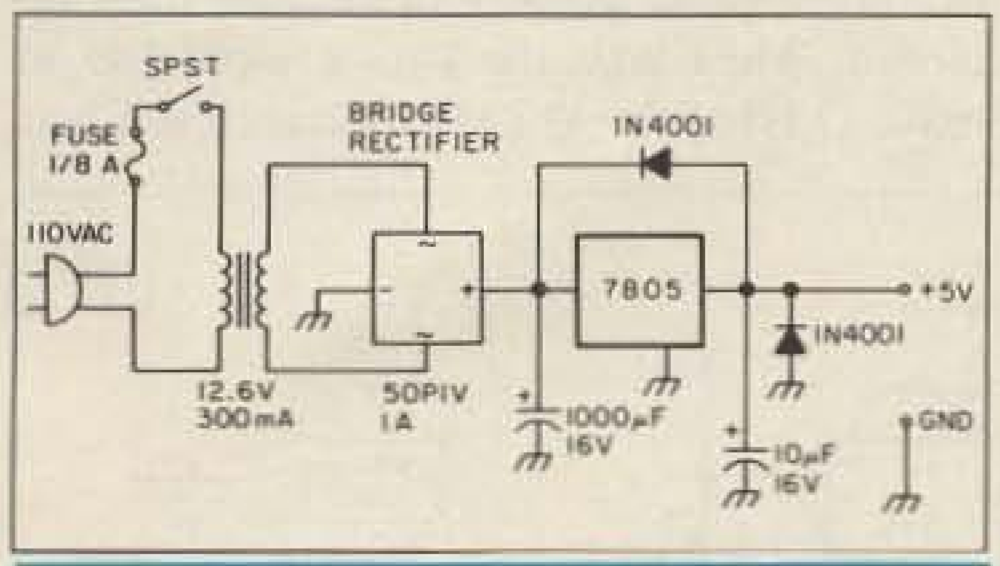

Figure 1. 5-volt test power supply.

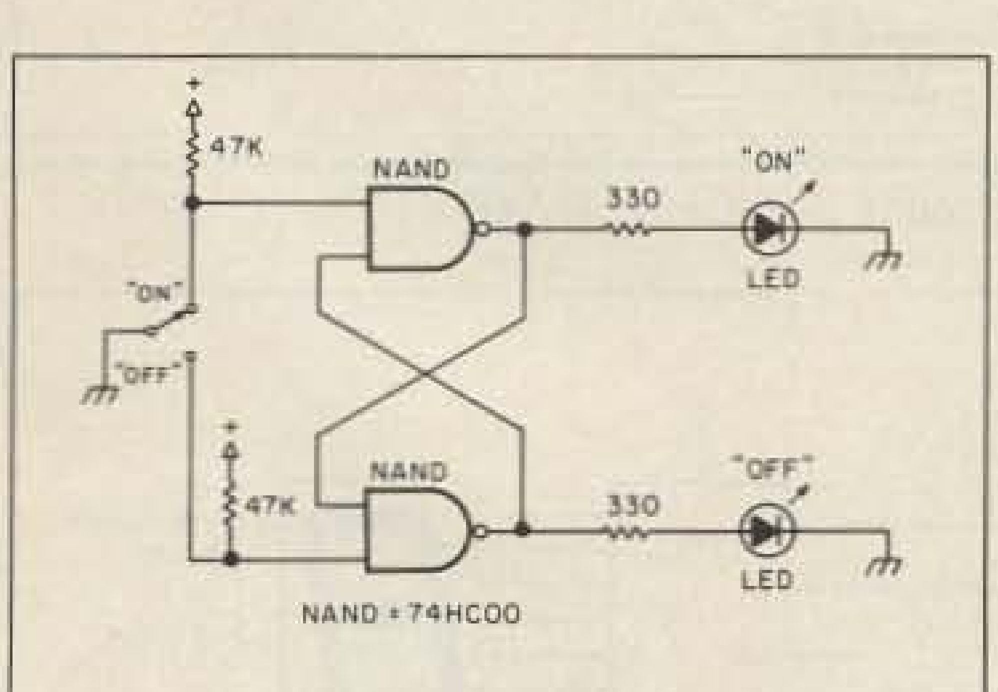

Figure 2. NAND and NOR flip-flop switches.

There are many good sources of information on the basic theory of the flip-flop in its many forms. Lancaster's *Cookbooks* devote entire chapters to fundamental flip-flop operation. Times have changed since the publication of this information. Digital families now have hundreds of members, many of which perform special functions that, at one time, had to be configured with several gate and flip-flop chips. In today's world, it's necessary to know what functions basic flip-flops perform best and how to use that knowledge in selecting more complex chips tailored to special needs. Therefore, there will be little theory here.

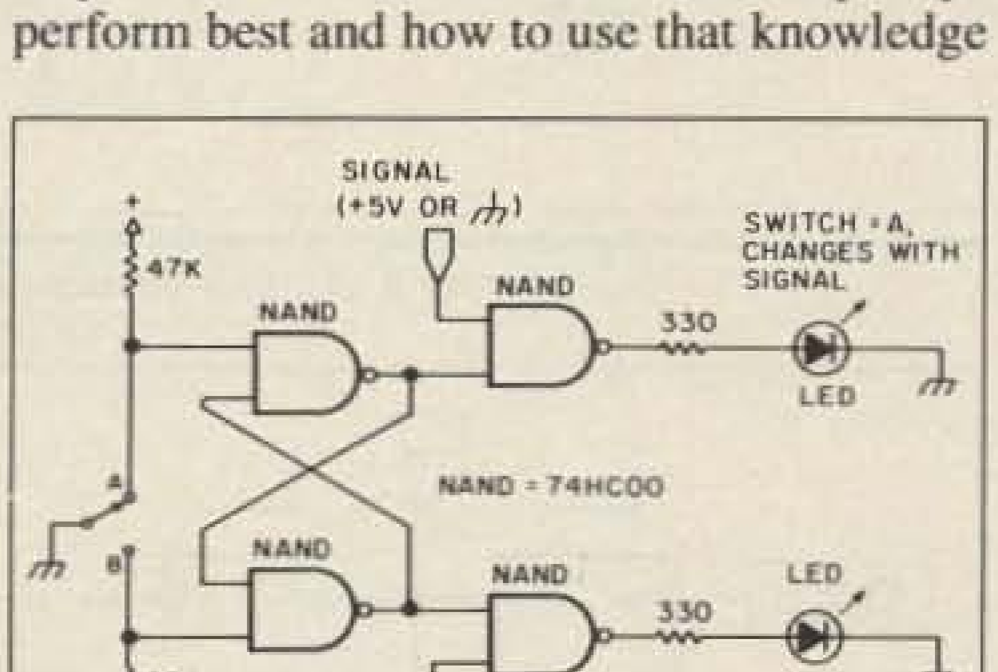

Figure 3. NAND flip-flop controlling pass/block gates.

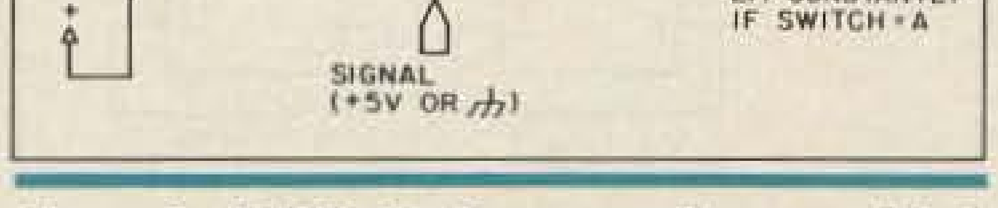

Figure 4. LED indicator symbol to simplify drawings.

This discussion will mainly look at practical circuits and see how hams can apply them to their designs. Some of the jobs basic flip-flops still do best are switching, dividing by two, making controllable shift registers, and controlling circuit functions. This is not all the work fit for flip-flops, but it provides an appreciation of their versatility.

Related to the flip-flop in most data-book listings is the latch. Although somewhat different in internal circuitry and function, the latch also holds a piece of data and controls circuit functions. This presentation will com-

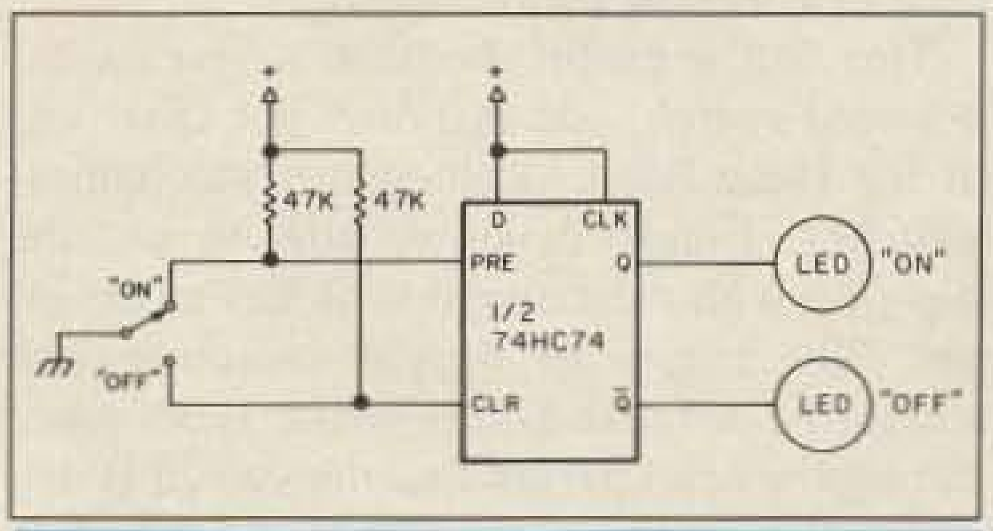

Figure 5. Flop-flop chip used as a debounced switch.

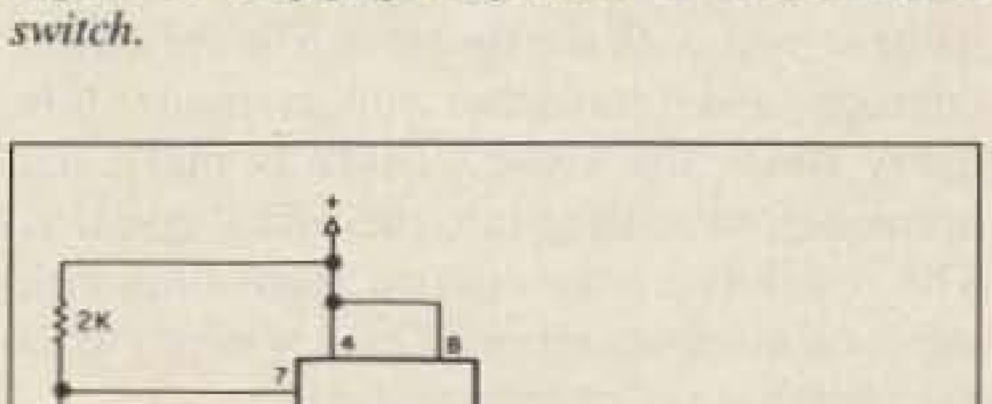

Figure 6. A handy source of slow pulses.

pare and contrast latches and flip-flops, and show which differences will make ham circuits work better.

Materials

First is an experimenter's board or IC breadboard for the tests. Second is a power supply. The simple 5-volt supply in Figure 1 will handle all the experiments. Be sure to use a heat sink on the 7805 regulator. Also, do not forget the fuse. The power supply and experimenter board make up a useful package both for this exercise and for future experiments.

Third, a small collection of parts. Four or five 74HC74 D-type flip-flops, a couple of 74HCOO NAND gates, a pair of 74C02 NOR gates, a 74HC75 latch, a few LEDs, a handful of 330Ω, 10kΩ, and 47kΩ resistors, plus some specific parts for some handy test modules will fill the bill. No soldering will go on here, since these parts will later go into real projects.

The Basic Flip-flop

Figure 2 shows a pair of basic flip-flop circuits made from gates. As the two diagrams show, either a pair of NANDs or a pair of NORs will work fine. Because the output of one gate feeds back to an input of the other, briefly changing the rest state of one of the control lines can switch and latch the output state. Not only does the output change when the input changes, it stays that way after releasing the switch or whatever pulse system used to affect the control inputs.

This fact is useful, because it gives a debounced switch, one that does not show any of the make-break problems of mechanical switches. Figure 2 shows how to use the flip-flop with mechanical switches at the input. The rotary or toggle switch switch should be a break-before-make type, since during the brief period that the switch is disconnected between positions, the flip-flop output will remain as it was left. It changes to its new state with the first touch of the switch contacts, and it stays that way, no matter how many times the switch contacts make and break before settling into their final position. The resistors on the control input lines hold the control inputs HI or LO (as needed) when the switch is not reversing them.

The Xs in the input lines show the points where it's possible to remove the mechanical switch and resistors in order to feed control signals directly to the inputs. This mode of operation requires signals always present at the inputs as either HIs or LOs. If a no-signal condition is desired (e.g., following a tri-state buffer) then one should keep the pull-up or hold-down resistors on the control input lines.

The two basic flip-flops in Figure 2 use 74HCOO and 74HCO2 gates. It's possible to substitute regular CMOS NAND and NOR gates (with changes of pin numbers) and use the same value resistors. If using 7400-series TTL ICs or 74LSOO-series low-power TTL ICs, check the manuals for proper pull-up and hold-down resistors. However, since the 74HCOO-series combines the best features of both the 4000-series CMOS ICs (except for the wide range of operating voltages) and the 74LSOO-series, I will use them throughout these experiments.

Versatility

Notice that the flip-flop has two outputs. They are conventionally called Q and −Q, since one is always the reverse of the other. When changing a control input, both outputs change together. This feature is very useful, because some complex circuits may require a HI in one place and a LO in another, with both conditions reversed for the off, inactive, or opposite condition. Figure 3 shows an example, using gates as passing/blocking elements for pulses. When gate 1 passes, gate 2 blocks, and vice versa. The output from the flip-flop tells the gates whether to pass or block. Note that it's possible to use a single gate IC with four gates per chip to create this entire circuit. When there are unused gates and the builder needs only a control switch, the simple flip-flop is often the best way to go.

"In today's world, it's necessary to know what functions basic flip-flops perform best and how to use that knowledge in selecting more complex chips tailored to special needs."

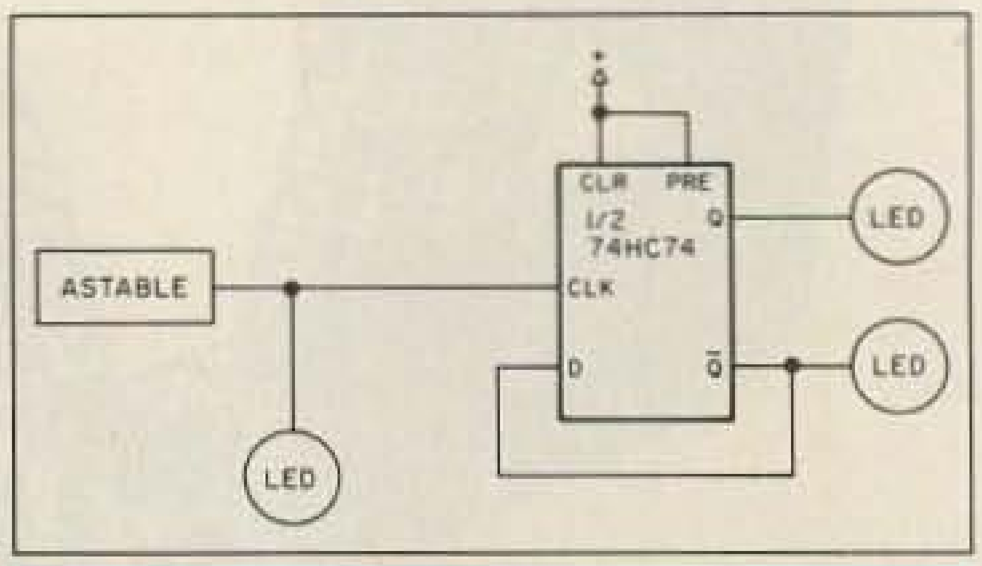

Figure 7. Basic divide-by-2 flip-flop circuit.

In Figures 2 and 3, LEDs with series 330Ω resistors are indicators. Figure 4 shows how the remaining diagrams will represent the LED indicator. By using the simplified symbol, the reader can keep his attention focused on the important aspects of the test circuit. As a convention, the LED will indicate a HI condition. When the LED is off, a LO is present. As I proceed, I shall represent some other useful test circuits in the same way.

One can do the same basic switching job of Figure 4 by using one of the designated flip-flop ICs, such as the 74HC74 D-type flip-flop shown in Figure 5. For the moment, ignore the Data and Clock inputs by tying them to the positive supply voltage line. For this chip, the Preset and Clear inputs are the control lines that correspond to those in the gate flip-flop circuit. Then take the Preset input LO to make Q HI and −Q LO, and take the Clear

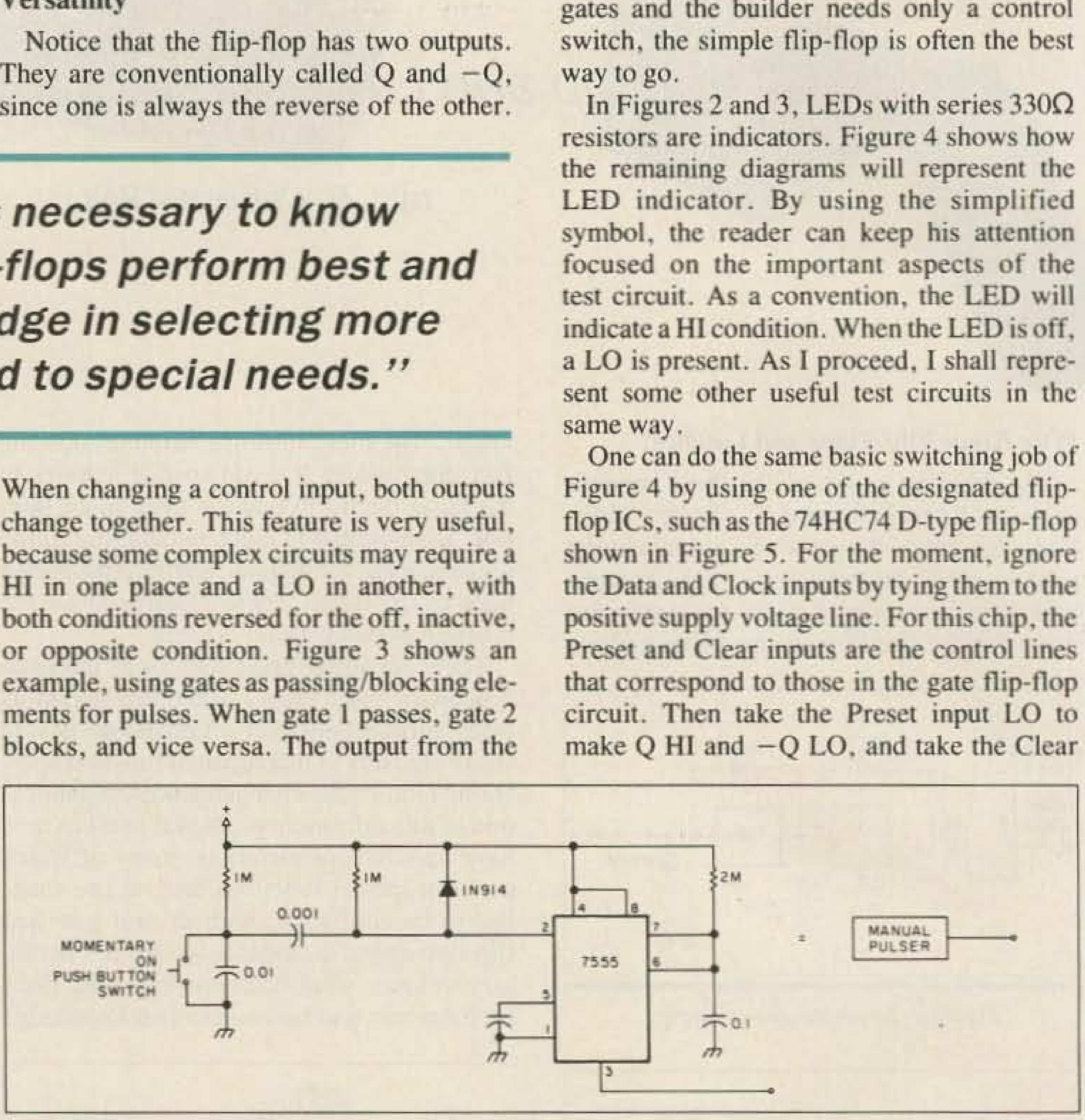

Figure 8. A handy manual pulser (pulse duration ⅓ second).

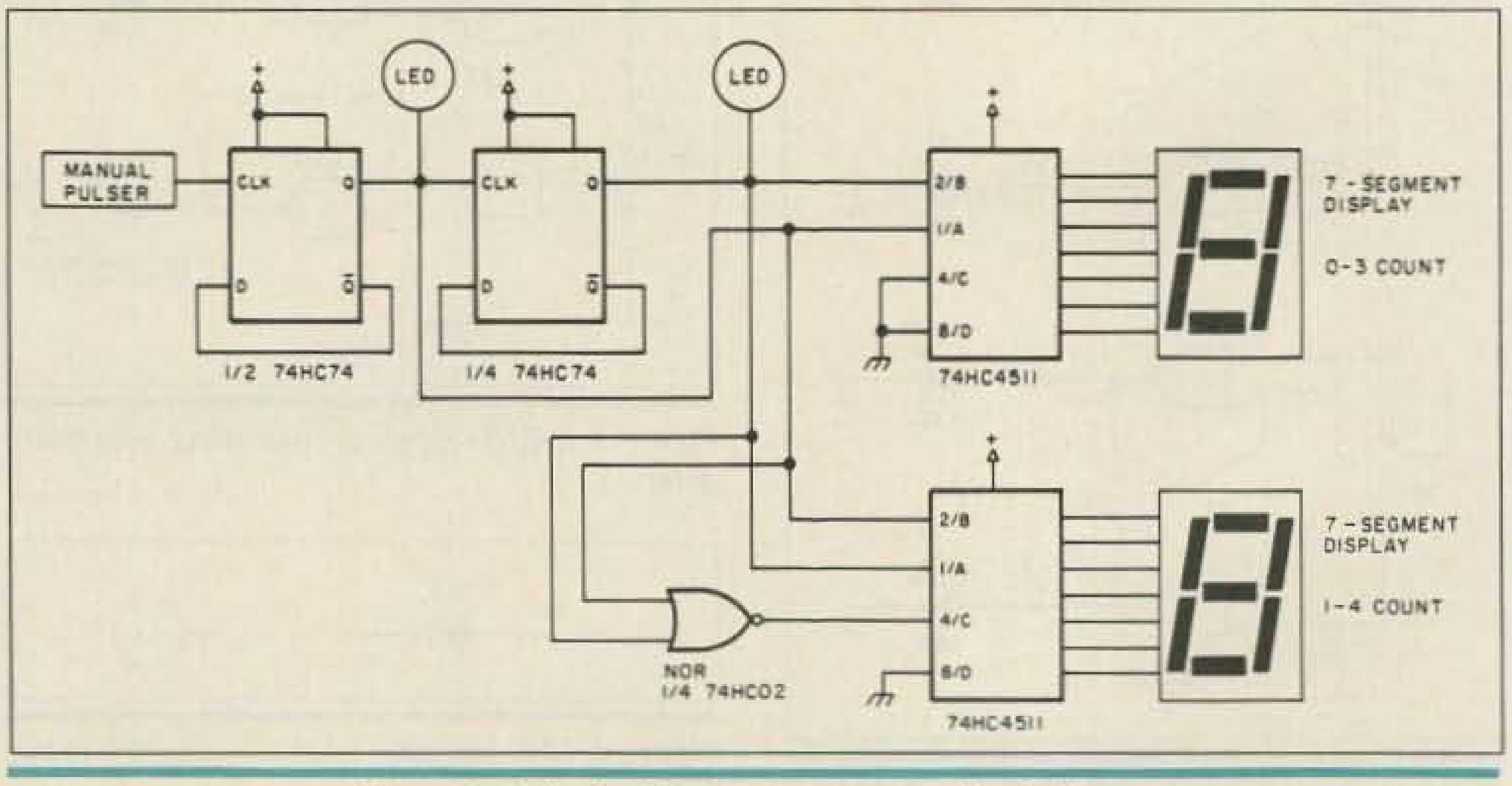

Figure 9. A flip-flop binary counter-controller-indicator.

line LO to make Q LO and −Q HI. Since the 74HC74 has two flip-flops per chip, only half a chip is used, just as the gate version of the switch.

There are other kinds of flip-flop chips, such as the J-K models, and there are multiple flip-flop chips with many individual circuits, but restricted control features. Rather than entering the confusion of trying to track many different kinds of ICs, I'll stick with the 74HC74 so the reader can try to master most of its capabilities. Then the reader can apply the lessons learned to almost any of the other variations.

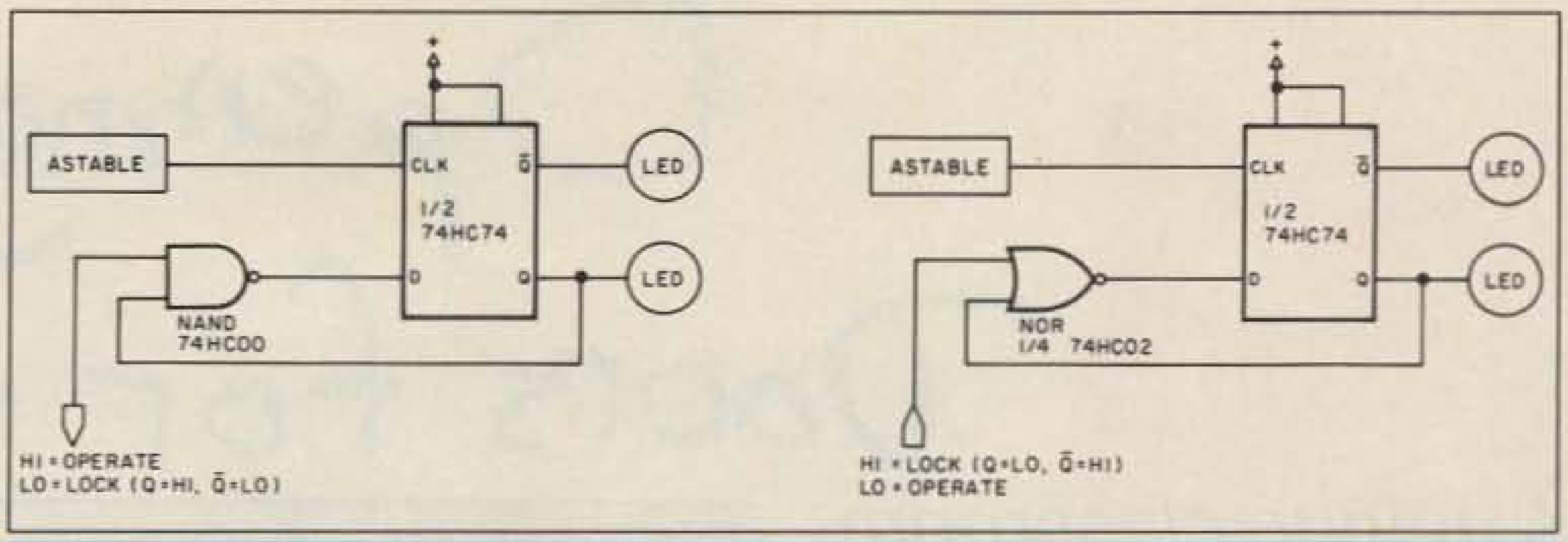

Figure 10. Controlled dividers using NAND and NOR control gates.

Flip-flops as Dividers

The D-type flip-flop provides additional circuitry for clocking data into the chip. A pulse source will help make use of this circuitry in these experiments. The astable circuit in Figure 6 uses a 7555 to produce pulses with about a 1 second on-time and a 1 second off-time. The 74HC00-series chips here can handle frequencies well above 10 MHz, but this slow pulse rate allows the reader to see the circuit operation more clearly.

To use the Data and Clock inputs of the 74HC74, hold the Preset and Clear lines HI. Next, we must place the data (a HI or a LO) at the Data input. Finally, place a LO-to-HI transition on the Clock input line. The first step is normally done when wiring the circuit, as in Figure 7. Then, whatever the state of the Data line, when the clock goes from LO to HI, the D-state will appear at the Q output and its opposite will appear at the −Q output.

However, Figure 7 shows the −Q output wired back to the D input. The circuit also shows the 7555 astable circuit feeding pulses to the Clock input. Take a close look.

Assume that Q is LO and −Q is HI. This means that D is also HI. When the Clock pulse goes from LO to HI, the HI on D appears at Q. Then −Q goes LO, which means that D also goes LO. But nothing will happen to D's LO until the next Clock pulse changes from LO to HI. With this second pulse, Q goes LO and −Q goes HI, feeding D a HI. This cycle continues. Note that the Q and −Q outputs make one complete pulse for every two pulses from the 7555. The circuit divides the frequency of the astable by 2.

The divide-by-two circuit is very useful. Sometimes an experimenter needs both the original frequency and another frequency half as fast. In many applications, the tester can start the astable at twice the desired frequency, and use the divider circuit to bring it to just the desired value. Remember that in the 7555 circuit, the HI and LO durations are not exactly equal. If equal on and off times are important (e.g., in coded systems), the divider circuit guarantees equality.

Simple Counter

A builder can make a simple binary counter and controller by stringing together divider circuits. To operate this circuit, let's make a debounced manual pulser. Figure 8 shows a good circuit, again using the 7555. The input resistors and capacitors to the trigger pin simply guarantee that there is one pulse per finger press on the switch. I use some extra keyboard switches pulled from a defunct project for this work. Keyboard switches are among the best press-release switches available. Incidentally, I keep two or three prewired perfboard versions of both the pulse circuit and the astable circuit close at hand and just plug connecting wires into the breadboard. In the long run, that saves a lot of time.

Figure 9 shows an example of a simple binary counter using two flip-flops in series. This starts with a debounced pulser. Note that there are four LEDs connected to some gates as a substitute for a controlled circuit. There are up to four pass/block gates or any other type of circuits among which to switch. When pressing the button, the next LED lights—or the next circuit becomes operative—with all the others off.

At the same time, a binary number between 0 and 3 is created, using the Q outputs of the two flip-flops. The experimenter can place these two outputs on the 1 and 2 inputs of a BCD driver chip that feeds a 7-segment display.

Now wire the 4 and 8 inputs low. The result is a display that reads 0-1-2-3 when pressing the button, indicating the number of the activated circuit. Figure 9 also shows some optional additional circuitry to allow the 7-segment display to read from 1 through 4. The NOR gate simply converts the flip-flop LO-LO = 0 into a HI to trigger the 4 digit.

This basic circuit is useful to about one more place for counting up to eight and controlling one of eight circuits manually. However, beyond eight, there are many more flexible binary and BCD counters, as well as programmable divide-by-N counters. There are decade counters (74HC160), dual decade counters (74HC390), 4-bit binary counters (74HC161), dual 4-bit binary counters (74HC393), divide-by-10 counters (74HC4017), up/down counters (74HC191), and 14-bit counters with a built in oscillator (74HC4060). They are all based on flip-flops, but that circuitry is buried inside the chip. One needs only to master the input and output connections and any special rules of operation to use them.

For example, some counters count on a LO-to-HI transition of the Clock input, others on the HI-to-LO transition. Most have Reset lines that instantly return the count to zero, but some need a HI to do the job, and others need a LO. When checking out these counters in data books, be sure to read all the rules.

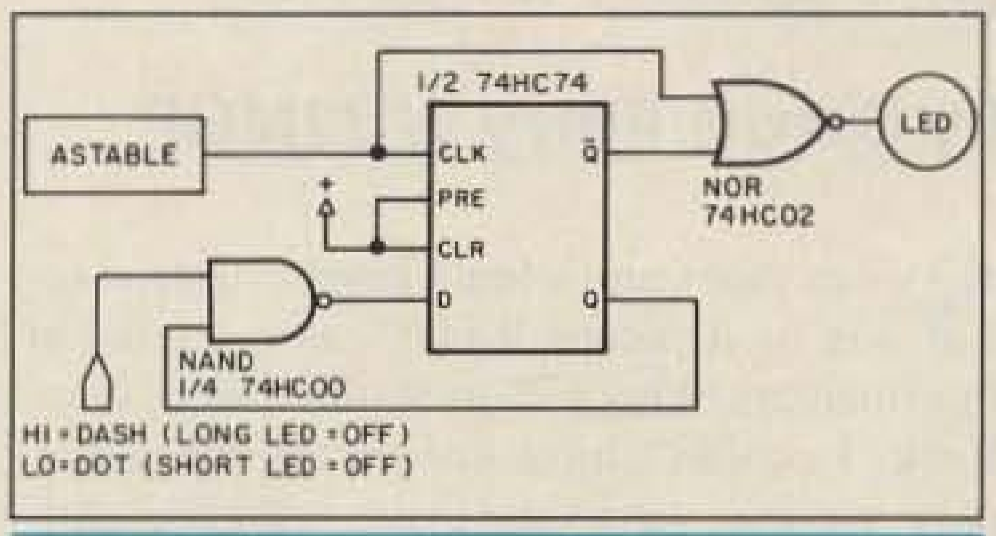

Figure 11. A controlled flip-flop divider for making Morse dashes from dots.

Controlled Dividers

So far, the divide function of D-type flip-flops have been used in an uncontrolled manner, dividing all the time. However, by adding a gate, it's possible to control just when the circuit divides. Figure 10 shows two controlled dividers, using NAND and NOR gates. Which one the experimenter uses depends upon the gates available in his circuit and whether he wants to control the process with a HI or a LO. Since the gate will reverse the level of the feedback signal, he feeds the Q output to the gate (instead of the −Q output).

Figure 11 shows an application of the controlled divider circuit to make Morse code dashes from dot pulses. When the NAND gate input is LO, its output to the D terminal is HI. The clock pulse will make Q HI and −Q LO. Since −Q goes to the NOR gate, the LO permits the gate to pass (inverted) whatever comes in the other input line. In this case, what passes are dot pulses.

When the NAND gate input is HI, its output to the D input will be the inverted Q output. If Q is HI, D is LO. The next clock pulse will make Q LO and −Q HI (with the D input seeing a HI). The −Q HI hits the NOR input and holds its output LO during both the HI and the LO part of the original clock pulse.

Now, with the start of the next clock pulse, the HI D input transfers to Q, with −Q going LO. Thus, one gate of the NOR is LO, allowing the dot pulse to pass as a dot. The result is a NOR output LO three times as long as a dot LO, while the space time (HI) is the same length as usual. By summing the dot pulse and the divider pulse, it makes perfect dashes.

Combine Figure 9 and Figure 10 elements to create a divider-controller-counter that operates only under certain conditions fed to a divider-controller gate. The possibilites are endless!

Next time: Shift register circuits.