DRAWING BOARD

Counting with the 4089

ROBERT GROSSBLATT

ANYBODY WHO HAS BEEN FOLLOWING this column on a regular basis knows that I'm a real freak when it comes to paperwork, data sheets, and generally anything that makes life on the bench a little easier. That's because it's always a good idea to collect as much information as possible before you start "getting your hands dirty."

However, there are limits to how much information you can absorb. Sooner or later you'll find it impossible to learn any more from paperwork alone. That means that you must actually work with a device in order to truly understand its operation. So, in line with that sage advice, let's get our hands dirty, and see what we must do with rate multipliers to make them perform some useful task.

Using rate multipliers

Once again, we'll be using the 4089 for our discussion because, in the first place, it's a CMOS IC—I believe in using CMOS whenever possible. Second, I just happen to have some of them lying around the house—always an important consideration when trying to cut costs!

The chart in last month's column should've given you a good idea of what the 4089 can do. Although the numbers may seem confusing at first, you'll find that actually powering up one of those IC's will make its use a lot clearer. As with most other special-purpose IC's, the majority of the control pins are held either high or low during normal operation. That greatly simplifies understanding how the IC works.

The majority of uses for the rate multiplier will revolve around some sort of arithmetic—usually multiplication or division. Other kinds of operations are possible as well (because more-complex arithmetic, such as square roots or exponential functions, can usually be reduced to repetitive basic arithmetic). But, let's start off by seeing what we have to do to multiply two numbers together.

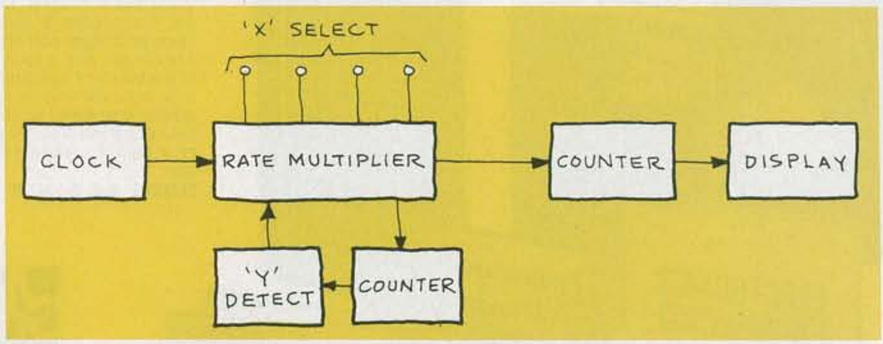


FIG. 1

Multiplication circuit

Figure 1 is a block diagram of the circuit we want to put together to multiply "X" times "Y." Although there are a number of boxes there, things aren't as complicated as they may seem to be. The basic operation of the circuit is simple and the clock can be any type of arrangement that you want to use, as long as it's noise-free and the waveforms look something like a squarewave.

A basic 555 oscillator or some other type of clocking arrangement works fine. You may be wondering about the frequency needed from the clock. Well, the answer to that question will surprise you: It doesn't matter what the frequency is! How's that for simplicity? But right now, let's clean off our hands and see why that apparently screwball statement is true.

First consider how the rate multiplier works: It takes the input-clock frequency, does some internal division and gives us two kinds of outputs.

Now recall that the "base-rate" is equal to one sixteenth the input clock, and that the "multiplied-rate" is the base rate multiplied by whatever number is presented at the binary inputs. If we were to write that statement as a formula, it would look something like what appears below:

Base rate = Input Clock/16
Multipled Rate = (X)(Input Clock/16)

Where "X" is one of the numbers we're multiplying.

That means that every time a pulse appears at the base-rate output, we'll get "X" number of pulses at the multiplied-rate output. To multiply "X" and "Y" all we need to do is count the base-rate pulses and stop after "Y" number of (multiplied-rate) pulses. Getting the right answer is really as simple as counting the total number of pulses at the multiplied-rate output, or expressed as a formula:

Base Rate = Input Clock / 16
Multiplied Rate = (X)(Base Rate)
= (X) (Input Clock/16)
X = (Multiplied Rate) / (Base Rate)
X = [(X) (Input Clock / 16) / (Input Clock / 16)]
X = X

As you can see, when we're doing multiplication with a rate multiplier, both the input clock and the internal base-number of the IC are completely unimportant—they cancel out. Getting the answer is only a matter of, as we said before, keeping track of the base-rate output pulses and counting up the multiplied-rate pulses.

The only part of Fig. 1 that could be at all tricky is the counter and other associated circuitry needed to detect when "Y" number of pulses have been generated at the base-rate output. There are two ways to do that. The method that you choose depends mostly on the type of counter you decide to use.

Since we want to count something "Y" times, we can either use an up counter starting at zero to detect "Y," or preload a down counter with "Y" to detect a zero. The choice again depends on the IC you want to use. Because up counters are a lot easier to come by, that's the way we'll go. Just remember that it's only a matter of personal choice.

One of the nicest things about CMOS counters is that there's a whole range of ripple counters that provide a one-IC solution to problems just like ours. They come in really handy when you want to count to some large number.

The 4020, 4040, and 4060 are all members of the ripple-counter family, but of those only the 4040 has outputs covering a continuous 12-stage count. That means that you can use it to detect any number from 0 to 4096.

Figure 2 shows the pinout of the 4040. It's used just the way you'd expect it to be; a clock is routed to pin 10, the reset pin is held low, and the IC will advance one count

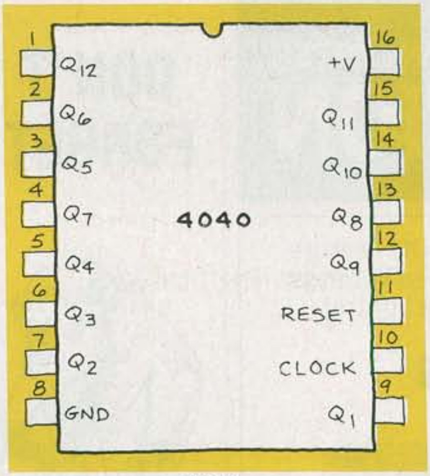


FIG. 2

on the negative-going edge of each incoming clock pulse.

Detecting "Y" involves a bit of gating. How you set things up, naturally, depends on the number that you're trying to detect. For instance, let's say that we want to multiply 14 by 67.

The only special thing about picking the numbers to be multiplied is to make sure that one of them is less than 16. That's because the 4089 has only four weighted inputs and the highest number that those inputs can represent is 15 (or 1111 in binary). However, the 4089 is easily cascaded for larger numbers. We'll examine that in more detail once we get through our example.

Figure 3 is a schematic of the circuit that we're designing; it shows everything except the counter and the display. We'll deal with them later.

The weighted inputs of the 4089 are set to load in a binary 14 (1110). We'll be using ⅓ of a 4073, 3-input AND gate to decode the Q1, Q2, and Q7 outputs of the 4040. Switch S1 is used to start the whole business going.

When S1 is pressed, IC3 resets and causes the output of the IC4-a to go low. That enables the rate multiplier, IC2, by bringing pin 11 low, causing it to start sending base-rate pulses to the clock input of IC3. For each base-rate pulse, 14 pulses are output at pin 6, the MULTIPLIED-RATE OUTPUT. You can see that the way we're doing our multiplication is to make IC2 count to 14 over and over until it's done it 67 times.

When that happens, 67 is decoded and the output of IC4-a goes high. That resets IC3 to zero

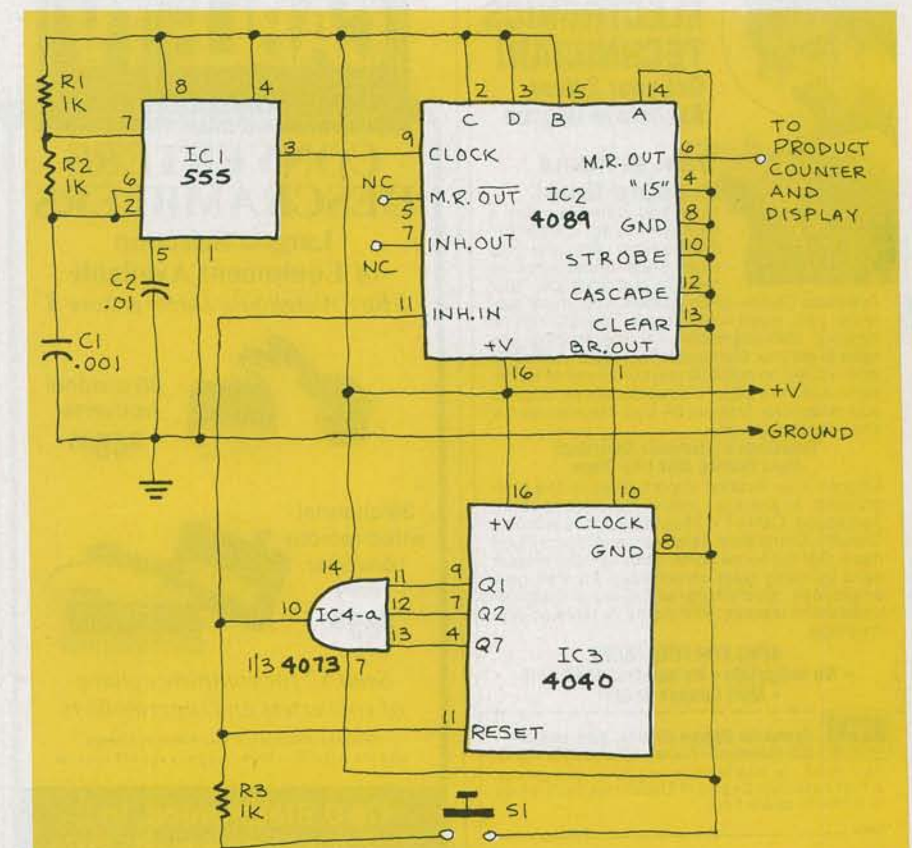


FIG. 3

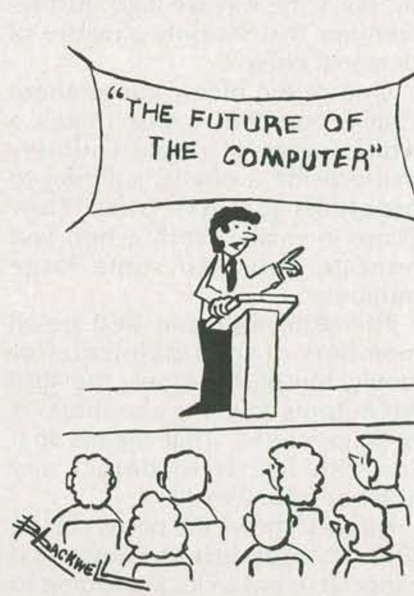
and brings the INHIBIT INPUT (pin 11) of IC2 high, preventing it from putting out any more pulses.

You could easily modify the circuit and use a gated oscillator so that the 4073 (IC4-a) would also stop the clock—there are many ways to accomplish that.

Since IC3 is reset to zero each time the circuit is used, any number that comes after 67 will never appear at the output. If you use a number other than 67, (which you probably will), you'll more than likely have to use a gate with more input legs. Not to worry though, because that can be taken care of by the two remaining gates in the 4073.

To add more inputs, just use a second gate as an input device and the third one to AND the first two together. It's not too difficult to figure out what gating you have to use and since that's not really what we're talking about, we won't go into it.

Next month we'll add a display to the circuit and see what kind of things must be done to cascade two or more rate multipliers together and allow us to multiply by virtually any number we want. We'll also see how to configure the circuit to do division. Believe it or not, that task is much easier than you think! And better yet, it gives us yet another opportunity to use—and become more familiar with—the 4089. R-E



"My computer tells me that there is no future in artificial intelligence."