THE DRAWING BOARD

More on the 4018

ROBERT GROSSBLATT

ANYONE WHO SPENDS A LOT OF TIME hacking around with hardware soon finds out that there are certain kinds of circuit requirements that pop up over and over again. Forget the old axiom that there is nothing new under the sun—it's only half right. There may only be a few new questions, but there are always lots of new answers. One of the words to keep your eyes peeled for when you're browsing through data books is "programmable." Whenever you see that word, pay special attention to what follows because there's a good chance that the information there can save you all sorts of trouble.

The 4018 is billed as a "programmable" counter, meaning that it can be preset to perform division by any number up to ten. And, like the 4017, it can be cascaded to increase the range of division; that is, two IC's will divide by 100, three by 1000, and so on. Now, those of you out there who have been following along for the last few months on our little trip through the "suburbs of counterland" will probably be wondering why the 4017 was called a "decade counter" while the 4018 enjoys the added adjective of "programmable." Well, the answer is really simple.

When we used the 4017 for frequency division, there were lots of problems we had to overcome. Some of them, like fixing the reset, could be handled by adding a bunch of extra parts to the circuit. Other problems, like the duty cycle of the output, weren't quite that simple to handle. Squaring up the output of the 4017 for any kind of division would have required the kind of hardware design that went out the window with 200-watt soldering irons and 12-gauge wire. What we mean is that the 4017 wasn't really designed to handle the job of frequency division. Sure, if all you care about is knowing "how many" over a period of time, the IC will do the job. But if your application is finicky about the output waveform, you'll have to turn to the 4018. The 4018 is a real "divide-by-n counter" while the 4017 is simply called a "counter."

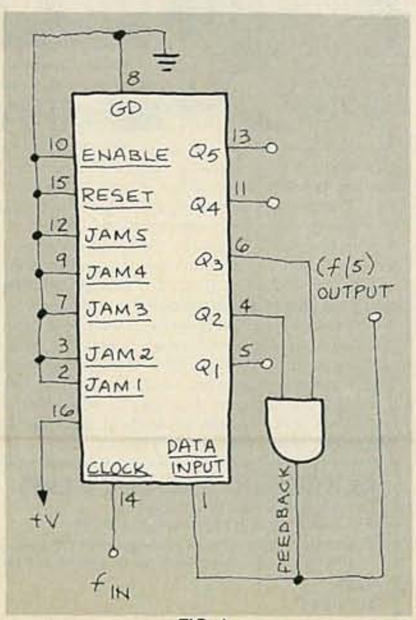


FIG. 1

In order to explore that a bit further, let's make the 4018 do something and see how it differs from the 4017.

First of all, there are two ways we can use the 4018—let's call them the "fixed" and "preset" modes; let's talk about the fixed mode first. In that mode, the IC can do pretty much what the 4017 did—divide by any number from 2 to 10. The device requires a feedback loop to operate and the output is picked off the feedback path at pin 1, the data input.

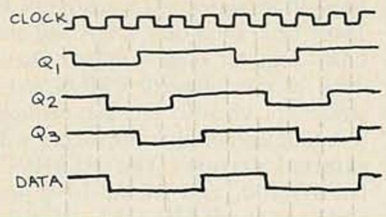


FIG. 2

Figure 1 shows a typical circuit using 4018. In that figure, the IC is set up in the fixed mode to perform frequency division by five. (In our last column we gave you a table showing which pins to use to divide by a particular number.) Since we're not using the preset features of the IC, we have to ground the JAM inputs as well as

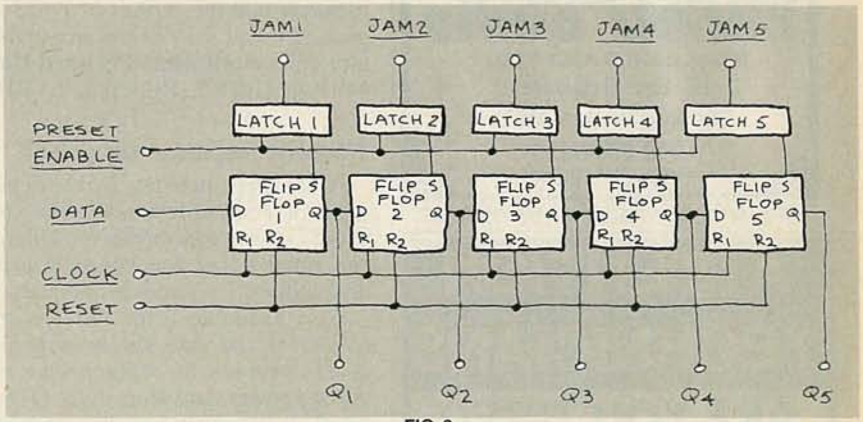


FIG. 3

the RESET and ENABLE pins. That is standard practice for all unused CMOS inputs. Although the ENABLE pin really controls the preset functions of the IC you can think of it as somewhat similar to the ENABLE pin of the 4017.

In any event, the proper feedback signal is provided by AND-ing the Q2 and Q3 outputs together and tying them back to the DATA input of the IC. Once we do that the incoming frequency is fed into the clock input and, as we said, we can pick off the divided output from the feedback path.

In order to appreciate that unique IC take a look at Fig. 2. Those are the waveforms you would see if you looked a various points in the circuit using an oscilloscope. Take a really good look a them because there's more here than i readily apparent—a little imagination will open up all sorts of wild possibilities.

Just as with the 4017, each of the outputs is phase-shifted from the previous one by exactly one (incoming) clock pulse. The difference lies in the fact tha the unused outputs of the 4017 only stayed high for one clock pulse, causing the out put waveforms to be really spike-filled and irregular. As you can see from the figure, the 4018 has an output frequency equal to the incoming clock frequency divided by whatever number we selected to divide by. The duty cycle is always "just about" fifty percent. We say "just about" because division by odd numbers is going to throw the output duty cycle "out of square" by exactly one period of the clock frequency. That is really only a minor annoyance and easy to live with—especially if you remember what the output waveforms of the 4017 looked like.

If we look at the output waveform in Fig. 2, we can see that things turned out as we could have predicted. Since we're AND-ing outputs Q2 and Q3 together, the output is high only when both Q2 and Q3 are high.

If you're dividing by ten, you can get the same output symmetry from the 4017 by taking the output from pin 12, the CARRY OUT pin. What's so special about the 4018 is that division by any number from two to ten will produce the same symmetry at the output. All that you have to do is feed the required Q outputs back to the DATA input. At most, the whole thing is going to cost you one AND gate, and that's a pretty cheap price to pay. If you don't have a spare gate on the board you can always accomplish the same things with a pair of diodes and a resistor, or some other similar arrangement.

Preset mode

Now let's see what happens in the preset mode—so we can use the programmable features of the 4018. The JAM and ENABLE pins allow us to preset the 4018 to divide by any number we want. What's happening inside the IC is really very straightforward. Remember that what we are dealing with is nothing more than a series of interconnected flip-flops. The 4017 is a "serial-input-only" type of shift register while the 4018 has both serial and parallel inputs. When we use the 4018's JAM inputs, what we're really doing is presenting the appropriate information to the SET inputs of the internal flip-flops and then strobing that information into internal latches by taking pin 10, the ENABLE pin, briefly high.

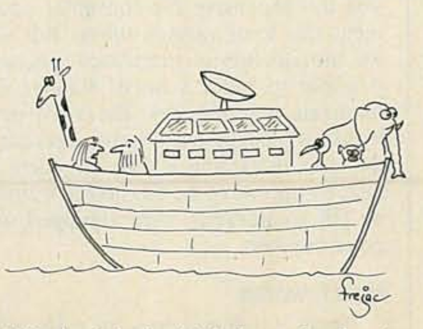
If you remember the design of the keyboard encoder we did some time ago, you'll realize that we used the same sort of strobing technique to latch the selected keyboard entry onto the data bus. What's happening here with the 4018 is exactly the same sort of thing. The designer of the IC was kind enough to put the latches on the substrate for us, so we don't have to go through the brain damage of hardwiring it ourselves. The code that we have to use to preset a number in the 4018 is, however, not a standard sort of code. That makes sense when we look at Fig. 3, a block diagram of the 4018's guts.

As you can see, what we have is a series of five flip-flops daisy-chained together. The incoming frequency of the clock line controls the speed at which the data is going to be routed through the flip-flops. The actual data is just various combinations of the Q outputs of the flip-flops that are fed back to the start of the daisy chain. If you think of that whole arrangement as a shift register, which is what it really is, you should have no trouble understanding exactly what's going on. Parallel loading with the JAM inputs is exactly the same as loading a shift register with parallel inputs. As a matter of fact, you should be able to see that the code that has to be used to load a number into the IC is the same code that the Q outputs present for any particular number.

When you reset the IC using the JAM inputs, all you're really doing is forcing the IC to start its count as if that number had already been reached by means of feeding the serial input. In other words, parallel loading a number is going to make the IC start its count at that number. Frequency division will have to be rethought a bit, since the IC is going to start out at a particular number and then reset to that same number when the count in the IC reaches 10. If you want to divide by four for instance, you'll have to load a six into the IC. That way the IC will reset after 10 − 6 = 4 counts. That may sound confusing but five minutes of actually playing with the IC will make it clear.

Parallel loading should make you think about what you have to do to make the operation switch-selectable. A simple rotary switch (if you're lazy) or a keyboard select (if you're ambitious) should allow you to divide by any number you choose, and the IC will provide glitch-free, highly symmetrical outputs. The added advantage of using that IC over the 4017 is that the output will be square, (or nearly square), regardless of the shape of the incoming wave. Think about that for a while.

Next month we'll discuss trying out that IC in a "real-world" application. We'll see how it can serve as the bridge for us to cross from the somewhat restrictive digital world to the occasionally flaky and always unpredictable world of analog circuitry. **R-E**



"Noah, I don't think that solar heating was a good choice for the Ark."