Experimenting with Electronics

by DARREN YATES, B.Sc.

Exploring CMOS - 2: gate circuits

This month, we start looking at some practical CMOS circuits from a wide range of circuit design areas. Some are simple, others more complicated — but all worth keeping in your circuit scrapbook.

The fact that you're reading this suggests you made it through last month's EWE — congratulations. CMOS theory can be as exciting as sitting at the SCG in the middle of winter, but if I may borrow some from an infamous phrase, it was the EWE we had to have...

To really understand digital logic, and thus, most of digital electronics, you must have an understanding of the basics. If you skip over this, you'll end up having problems with most of the circuits you read or design. Time spent studying this *now* is an investment in problem-solving for the future.

Digital electronics didn't start with the advent of TTL or CMOS. In fact, it started right back with Morse code and telegraphy. The simple action of opening and closing a switch is in itself a form of digital communication. It's the forebear of the digital mobile phone we have today — greatly more advanced I grant you. But from humble beginnings...

With the first computer cranking into life in the late 1940s, digital electronics began to find its feet before the advent of the transistor. But until integrated circuits (ICs) hit the streets, transistor logic formed the backbone of digital circuits. Circuit fragments like the transistor inverter we've looked at in previous articles are an example of this.

CMOS logic created a simple, smaller and more efficient solution, allowing designers to work with building blocks in a way that isn't too dissimilar to Meccano or Lego.

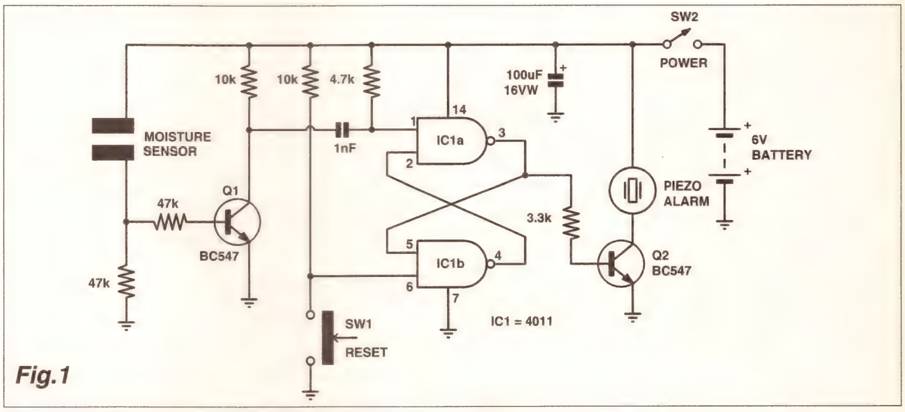


Fig.1

Moisture sensor

Right then. Our first circuit for this month is a moisture sensor and is shown in Fig.1. Now while it may not reach any great heights of excitement in terms of design, it contains a very simple building block which has formed the basis for everything we see in computers today. And remember — you have to crawl before you can walk.

Looking at the circuit, transistor Q1 acts as an inverter so that whenever moisture falls on the sensor and the resistance between the two pads falls below a preset value, Q1 turns on and the collector voltage goes low.

Now you'll notice two NAND gates that are cross-coupled. The input to IC1a at pin 1 is connected to the collector of Q1 via an RC time constant. This produces a short negative-going pulse at pin 1 when the collector of Q1 drops. These two gates form what's technically known as a *bistable multivibrator*, more commonly referred to as a 'flip-flop'.

Normally the output of IC1a is low and the output of IC1b high. When the trigger pulse arrives, the input at pin 1 drops low for a short time. Since this is a NAND gate, whenever any input goes low, the output goes high. This high output is cross coupled back to the input at pin 5 of IC1b. Since the reset switch is not pressed, this means that IC1b now has two high inputs and so the output of IC1b falls low. With the output at IC1a high, transistor Q1 turns on and sounds

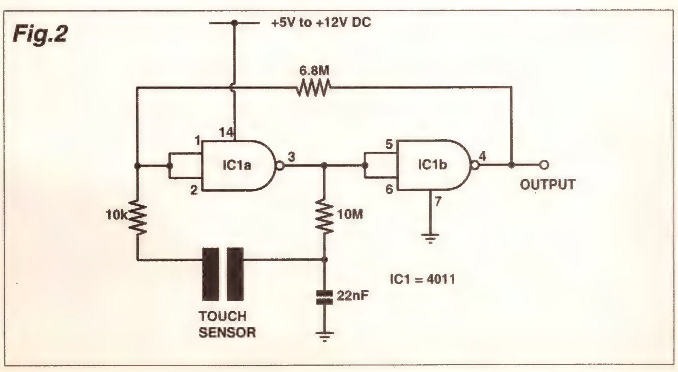


Fig.2

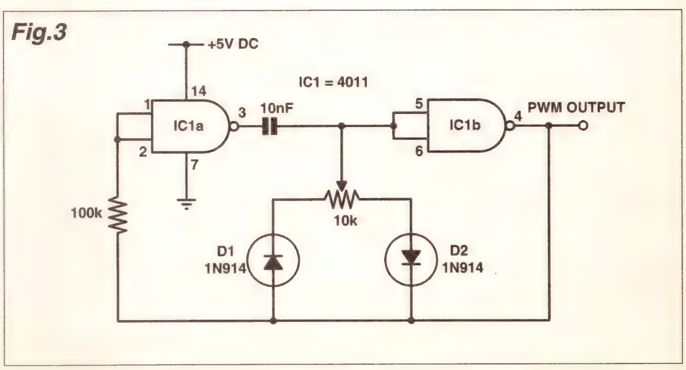


Fig.3

the piezo alarm.

Note that even though the trigger pulse last only a short time, the low from the output of IC1b then holds the IC1a high, regardless of what happens now to pin 1.

When the reset button is pressed, this pulls the input at pin 6 of IC1b low, which sends the output high. IC1a now has two high inputs and its output now falls low, silencing the piezo alarm.

You could use this circuit as a classic rain detector or a liquid spill detector. One way of making the detector is to use a length of Veroboard or stripboard. Take the board and connect up every second strip as one arm and then the other remaining strips as the other arm. This matrix system works very well as a moisture sensor; it's easy to make and quite cheap as well.

If you like, you can replace the transistor inverter with one of the spare NAND gates. Simply connect both inputs together of one of these gates, and connect the sensor between the positive supply rail and the gate inputs and a 10kΩ resistor between the gate inputs and ground. The output of the gate then connects to the 1nF capacitor in place of the transistor collector.

If you're in a real hurry, you could also remove the transistor circuitry, leave the 4.7kΩ resistor in place and simply connect the sensor between pin 1 of IC1a and ground. This is the rough-and-ready way and risks damage to the IC. The transistor interface adds an extra line of defence between the outside world and the IC.

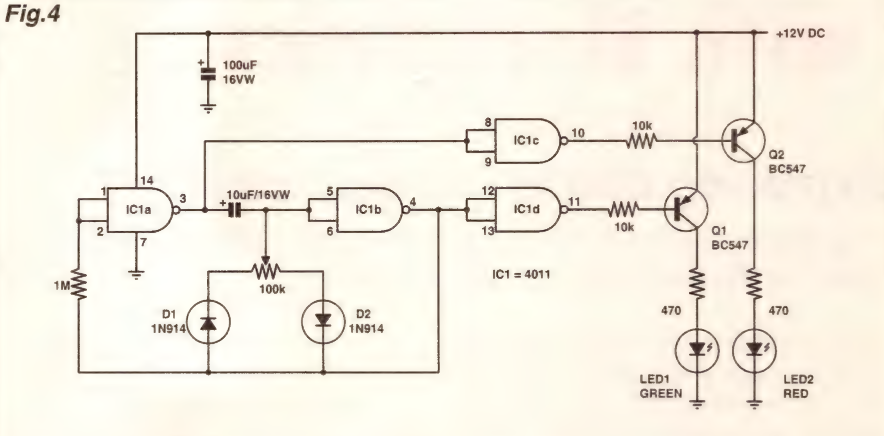
Alternatively if you're interested in just seeing how a flip-flop works, you could remove the front section altogether and connect another pushbutton switch between pin 1 of IC1a and ground, leaving the 4.7kΩ resistor to the supply rail from pin 1. You'll find that you can operate the flip-flop by pressing either switch.

Funnily enough, that's how the flip-flop gets its name. You can flip it in one direction with one switch and flop it back again with the other...

The point to make here is that the circuit remembers the last command you gave it via one of the switches. It won't change state until you tell it to, and that has made it one of the most useful circuit elements ever designed.

Obviously the flip-flop was around long before CMOS arrived on the scene, but the beauty of CMOS version

Fig.4



is that it draws only the smallest amount of current — around 30uA, making portable circuit designs so much simpler and smaller.

Touch switch

Every now and then you see ads on TV for those touch lamps. You know the ones — touch the base and the lamp switches either on or off depending on its current state. While the circuit in Fig.2 should never in its present form be used to control mains lights, it can form the basis of control for a wide range of battery-powered circuits.

Looking at Fig.2, it uses two NAND gates connected up as inverters. The good thing here is that you could use any inversion-type gate here — NOR, NAND or inverter — and the circuit will work exactly the same. You can also use a mixture of CMOS gates. This makes it easier to design into circuitry, because you can basically use whatever gates you have left over.

The two gates form yet another flip-flop. But notice that their connection is slightly different and the operation of the circuit relies on changes of resistance to change state. Let's take a look...

Since the output at IC1b can be either high or low at any one time, let's assume that it's low. This low is coupled back to IC1a via the 6.8MΩ resistor, Since there is no other DC path here, the inputs are considered low, and the output of IC1a is high.

Capacitor C1 is charged up via the 10MΩ resistor, but when the contacts are touched, the capacitor discharges into the low output of IC1b via the 6.8MΩ resistor, the 10kΩ resistor at the input of IC1a and the skin resistance of the finger.

By virtue of a voltage divider consisting of the 6.8MΩ, 10MΩ and 10kΩ resistors and the skin resistance, IC1a now recognises the input signal to be high. Its output therefore falls low, which forces the output of IC1b high.

The low output of IC1a ensures that the capacitor is discharged now, through the 10MΩ resistor. If the contacts are touched again, the high output from IC1b is connected via the 6.8MΩ and 10kΩ resistor, and the skin resistance to the discharged capacitor. Initially, this looks to the input of IC1a as a low input, so it sends its output high, which pulls the output of IC1b low again.

The only problem with this circuit is that if you keep your finger on the circuit, it will act like a very slow oscillator with a time period of around half a second, sending the output of IC1b high, low, high, low, etc. The trick here is to just *tap* the contact — the circuitry will take care of the rest in a flash.

Variable duty cycle oscillator

Over recent months, we've looked at a wide range of oscillator circuits; in particular, those that allow you to vary the pulse width and frequency. One circuit we looked at recently was the 555 timer audio amplifier circuit, which used the 555 timer as a pulse position modulator.

While that circuit worked well in terms of producing a modulated waveform, it had the disadvantage of also varying the frequency.

Most often, these circuits use fancy filter networks to remove remnants of the carrier frequency so that you're just left with the audio once the signal has been demodulated. The problem is that if the carrier frequency is changing as well, it's virtually impossible with simple electronics to remove this frequency.

This next circuit in Fig.3 is very basic, but it has the advantage of allowing us to change the pulse width without affecting the frequency. This type of circuit could be the basis of an excellent mini drill speed controller.

Looking at the circuit, it uses two NAND gates, although they could be any inverter-type gate, connected up in the standard oscillator configuration. The only difference is the diode/pot network between the inputs of IC1a and IC1b.

Normally here you would see just a single resistor, but this circuit uses a pot and two diodes connected in opposite directions. The diodes control the flow of current as the capacitor charges and discharges. When the capacitor charges up, current flows through one diode and when it discharges, it flows through the other.

The pot has two functions. First, the total resistance of the pot sets the overall frequency while the wiper allows us to vary the length of the pulse or duty cycle within that frequency.

Looking at the circuit, it may be hard to visualise what's going on here. So you might like to wire up the circuit in Fig.4, which should help you.

Fig.4 contains our basic PWM (pulse width modulation) oscillator plus the two spare gates from the same 4011 IC package. Each of these is connected up to the outputs of IC1a and IC1b. The outputs of IC1c and IC1d are themselves connected to two PNP transistors, which in turn operate a red and green LED.

So that you can see what's happening, we've made the oscillator run at a very low frequency. This will enable you to see each LED turn on and off. Looking at the circuit, when the output of IC1a is high, it sends the output of IC1c low which turns on transistor Q1 and the green LED. So the green LED indicates when the output of IC1a is high. Similarly, when the output of IC1b is high, the output of IC1d is low, which switches on transistor Q2 and the red LED.

The way this oscillator works is that as you vary the pot control, you'll see the lengths of time each LED is on will vary, in converse fashion. But the total length of time for both together in any one cycle will remain the same. This is the essence of what a PWM oscillator does.

Another benefit with this circuit is that you get two outputs which are out of phase with each other. Normally, with circuits such as a drill speed controller you only need one output; but say you wanted to connect this up to two motors and wanted one to slow down as the other sped up, this circuit would be ideal.

You simply connect one output via a driver transistor to one motor and the other output to the second motor. As you varied the pot, the PWM signals of both outputs will vary accordingly. As the duty cycle of one signal increases, that is, the positive pulse gets wider and wider over each cycle, the other signal's duty cycle decreases.

While this circuit is not unique to CMOS or digital electronics, it is much easier to build with CMOS gates and to operate over a wide duty cycle range because of their very high input impedance, which is of the order of 10^{12} ohms — or a million megohms.

Audio Signal Injector

Most people tend to not to think of digital circuits being useful in the audio domain, unless you start talking about analog-to-digital converters (ADCs) or digital audio. But this simple circuit in Fig.5 will enable you to locate problems within your audio circuits and make diagnosing faults that much easier.

It uses a single 4011 CMOS quad dual-input NAND gate IC and a handful of other components, most of which you'll already have lying around in your junkbox by now.

Looking at the circuit, IC1a and IC1b form another type of oscillator. It's a little different to what we've looked at in the past and relies on the AC signal being coupled back via cross-over capacitors from one gate's output to the other input.

Each gate is triggered when the output pulls low, say for example IC1a, allowing the capacitor connected to its output charge up via the 1MΩ resistor. Once it reaches a certain voltage, it then triggers the other gate to lower its output and the other capacitor now charges. Meanwhile the output of IC1a has been pulled high again, because of the triggering of IC1b. This cyclic process continues on indefinitely.

This first oscillator, using IC1a and b, is set to a frequency of 2Hz. From the output of IC1a, the signal is split two ways. Firstly, it's connected up to transistor Q1 which fires up the indicator LED. Since it's a test circuit, you need to know that it is working; so that's why the LED is there.

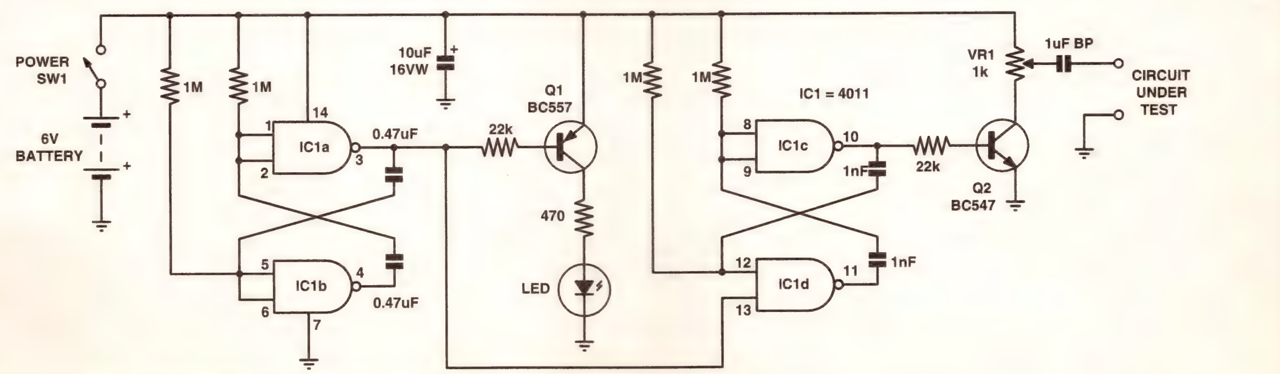
The second path is to one of the inputs of IC1c. Notice that IC1c and IC1d are connected up in a similar fashion as IC1a and IC1b. The only difference is the signal coming from the first oscillator and a change of component value for this second oscillator.

The component values shown should give you a 1kHz signal at the output of IC1d. Since one of the gates is also driven by the previous oscillator, this produces a *gated* tone signal. In simple terms, the output of the first oscillator controls when the second oscillator operates. The resulting output signal is a 250ms burst of 1kHz tone, followed by a 250ms length of silence.

The output of IC1d is then fed to a buffer transistor and connected to the test circuit via a 1kΩ pot and a 1uF bipolar capacitor. The 1kΩ pot allows you to vary the amount of signal you feed into the input of circuit under test. If you're testing a preamplifier for example, then you're not going to need to blast it with tonnes of signal!

Now you could argue that all you need

Fig.5



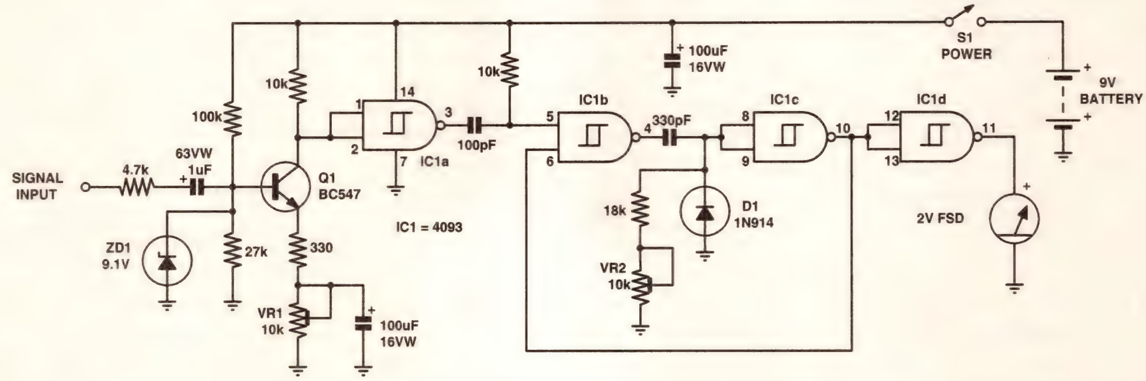


Fig.6

is a single tone oscillator to feed to your circuit under test. That may be so, but the signal from this circuit is much easier to hear, particularly if the circuit is noisy.

The way you use this circuit is just as important. If you have an audio circuit which you suspect has a problem somewhere, the best way to diagnose the problem is to start at the output, connect up the signal injector and listen to the output. That may sound a little like the bleedin' obvious, but you'll be surprised how many problems can be caused by dry solder joints even at the output.

Once you've found that the output is OK, you then start moving back through the circuit, one stage or even one component along the path at a time. Eventually you should get to a point where you find the test signal is being corrupted by the circuit under test. When this happens, you know that the area of fault is located between your current test point and the previous one. It's then a case of checking the various components along that section of the signal path, as well as checking various voltages to see where the problem may lie.

By necessity this is a very brief explanation and if by following this method you find the problem, then I would say that you've only had a fairly simple or obvious problem. Some problems require much more expensive equipment to find.

Audio frequency meter

While we're on the topic of digital circuits coming to the rescue of their analog cousins, this next circuit also falls nicely into that category. Looking at Fig.6, the circuit also introduces us to a new type of gate known as the Schmitt-trigger NAND gate. You'll only find these in a 4093 CMOS package. They're basic NAND operation is the same as the more familiar 4011 IC, except that each input has a Schmitt trigger tacked on the front end.

Some time ago, we looked at a Schmitt trigger circuit using discrete transistors, and the operation of the Schmitt triggers in each of these gates is similar. Basically, when an input rises about 2/3rds of the supply rail, Vcc, the input is then considered a high but it has to fall below 1/3rd of Vcc to be considered as a low input.

The major problem with the 4093 gate is that the upper and lower thresholds are somewhat arbitrary and depend upon the manufacturing process; they can even vary within the same IC. If you intend to use these gates in time-critical applications where you're relying on the threshold levels of the Schmitt trigger, I suggest that you try again unless the circuit is just going to be a one-off and you can build in some adjustment tools in case you need to replace the IC at some time in the future. If the latter is the case, then it should be OK.

Back to the circuit, the input audio signal is coupled to a single transistor amplifier formed from Q1 and its surrounding components. It has a gain of about 33dB (45 times) and its job is to provide raw gain. Zener diode ZD1 protects the input stage by clipping any signal which wants to head above 9V. We're not really interested in the output quality to the point that we actually want as much signal as possible.

The output is taken from the collector of Q1 and fed directly into the input of IC1a. This gate is set up as an inverter, but it's the Schmitt trigger function that we really want.

The signal at the output of this first gate should be a clean pulse waveform which has had the noise removed. The output from IC1a is then fed to a *monostable* ('one-shot') consisting of IC1b and IC1c, and their associated components. The 10kΩ trimpot, VR2, allows you to adjust the full scale output. For the 2V FSD meter, 20kHz equals 2V. You could also use your digital multimeter as well, if you like. Trimpot VR1 sets the sensitivity of the circuit.

The output from IC1c is then further inverted by IC1d and fed straight to a 2V voltmeter.

Now the way the circuit works is that the input signal is converted into a pulse waveform of the same frequency. On each falling edge, the monostable is triggered to produce a narrow negative-going pulse. If the frequency increases, the number of pulses also increases but the duty cycle also changes. Since the pulses are always the same length in time, if the number of them increases, then the duty cycle of the waveform changes accordingly.

The last inverter inverts the whole signal to produce a series of narrow *positive-going* pulses. The output signal going to the voltmeter is then a series of positive going pulses at the same frequency as the input. This change in duty cycle is picked up by the coil in the meter. Because the coil can't respond fast enough to instantaneous changes in voltage, it averages them to produce a constant voltage reading that is proportional to the duty cycle. The end result is that the needle rises in proportion to frequency — exactly what we want.

Obviously, by using a moving coil meter, we're limiting the accuracy. But as a guide to the input frequency, it should perform well.

OK, That should be enough to give you some ideas for projects of your own. We'll continue with our look at CMOS circuits next month. ❖