Getting Inside an NCO

TOM NAPIER

Poptronics Magazine, October 2000 Gernsback Publishing

One of the things that makes electronics fun is discovering the odd chips that have been developed to solve specific problems. I first heard of numerically controlled oscillator (NCO) chips some ten years ago. If I hadn't some of my designs, such as the VME bit synchronizer that I described in 'Sifting Signals from Noise' **(Electronics Now,** April 1999), would not have been possible. Here, I'm going to explain how NCOs work and how you can learn about them by building one from a handful of TTL parts.

What Do NCOs Do? NCOs are very handy to have in your electronic 'bag of tricks' once you understand their capabilities. For example, you can generate very pure and very accurate sinewaves anywhere from sub-audio frequencies to around 30 MHz with only a crystal oscillator, an NCO chip, a fast digital-to-analog converter (DAC), and a low-pass filter. The output can readily be phase or frequency modulated.

The frequency synthesizer in signal generators and in virtually every TV set and FM radio uses a voltage-controlled oscillator (VCO). Dividing its output down and then comparing it with a low-frequency reference sets its frequency. Any phase error between the divided frequency and the reference changes the voltage, which drives the oscillator. You change the output frequency by changing the division ratio. Unfortunately, the extent of the change is usually limited to about two-to- one by the varactor diode that tunes the LC oscillator. Since the rate at which the output frequency can change is limited by the bandwidth of the loop filter, a VCO can take tens of milliseconds to settle to a new frequency. Although that doesn't sound like much time, it does have an effect.

The NCO, as its name implies, uses a wide binary number to specify its output frequency. Consequently, an NCO's frequency can be set very accurately over a very wide range. Because the NCO is an open loop device, its output frequency changes almost instantaneously. This makes the NCO ideal for generating FM- and frequency-.shift-key (FSK)-modulated signals. Without NCO's, spread-spectrum transmitters and receivers would not be possible. As I'll explain, it's also easy to phase modulate the NCO's output.

The NCO is particularly easy to tune to a specific frequency since its output frequency is a linear function of the tuning number that you apply. Some NCO chips even accept a binary-coded-decimal (BCD) input and can be driven directly from front-panel thumb-wheel switches. The accuracy and stability of the output frequency is as good as that of the crystal oscillator that drives the NCO and the output sinewave is free of harmonics. (It does contain traces of non-harmonically related, spurious frequencies, which can be a nuisance in some applications.)

How Does An NCO Work? The NCO is based on a simple principle. If you keep adding the same number to a fixed-length binary register, the rate at which the register wraps around is a linear function of the number you add. Of course, it is also a function of how many times a second you add it.

Suppose you had a 16-bit register and an adder. Since any carries into the 17^{th} bit are lost, any sum higher than 65,535 takes you back to the start. Because it takes 65,536 counts to loop around once, let's add a number to the register 65,536 times a second. If that number is 1, then every second you will have added it 65,536 times and the register will have wrapped around once. If you could connect a 16-bit digital-to-analog converter to the register, you would see an output voltage that would make 65,536 steps from zero to full-scale and then back to zero. It would repeat that ramp once per second,

Now add 2 every time. The ramp will go up in steps of two and repeat twice per second. If you added 1000 every time, the ramp would make 65 or 66 big steps. The wrap-around rate would now be 1 kHz even though the register wouldn't pass through the some steps each time. As you use bigger numbers, the shape of the ramp becomes more irregular, but its repetition rate is still exactly proportional to the number you add, if that number is less than about half the register size,

A ragged ramp might not look like a very useful output, but that's because we are not finished with it yet. Most NCO chips contain a sine conversion section. This treats binary numbers as the phase of a sinewave and converts them into numbers that correspond to samples of a sinewave. When you put in a linear ramp, you get out numbers that, when converted to analog form, make a perfect sinewave. A complete NCO system consists of the digital NCO chip and a 12-bit DAC. The DAC generates a sinusoidal analog output from the numbers supplied by the NCO,

Of course, the sine output still has little steps in it. Since these occur at the NCO clock rate, they contain frequencies that are usually much higher than the desired output frequency, Filtering them out leaves a pure sinewave at exactly the specified frequency.

What's Inside An NCO Chip? A typical NCO chip is shown in Fig. 1. The register and adder are generally at

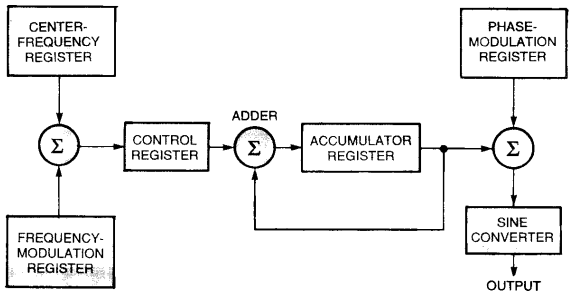

Fig. 1. A typical NCO chip contains circuits for phase- or frequency-modulation in addition to the basic phase accumulator and sine converter.

least 32 bits wide: the registers of some chips go as high as 48 bits. The longer the register, the finer the frequency resolution. Only the top 14 or so register bits are connected to the sine look-up circuit. Additionally, not more than 12 bits are available to be connected to the DAC, mainly because fast DACs don't have more than 12 input bits.

The basic parts of the NCO are the frequency-control register, the adder register (often called the phase accumulator) and the sine converter. However, practical chips often contain other functions. For example, they might have two multiplexed frequency control registers, allowing rapid FSK modulation by switching between them. Alternately, two frequency- control registers can be added together before being used, That allows one number to set the center frequency and a second number to apply frequency modulation. There is no discontinuity in the output sinewave when the frequency of an NCO changes. The new frequency starts at the phase and amplitude where the old one left off.

A number can also be added to the ramp from the phase accumulator before it is sine converted. This number changes the phase of the output signal, making it easy to apply phase modulation to the output, In an important special case, only the two most significant bits of the ramp are changed. This applies quadratic phase-shift modulation (QPW to the signal. QPSK is widely used for transmitting digital data, since two bits can be sent with each change of the carrier phase. Placing a multiplier chip between the NCO and the DAC allows the output to be amplitude modulated, Combining amplitude and phase modulation generates the QAM signals used in modems to transmit data down telephone lines.

What sets the output frequency? The output frequency of the NCO depends on three things:

- the accumulator width
- the update rate
- the size of the added number

The NCO's designer fixes the first, the clock that drives the NCO sets the second, and the user controls the third. A useful relationship between them is the frequency generated by one unit in the control number, This frequency is the resolution with which the output frequency can be set, it may be a tiny fraction of a Hz. The unit frequency is given by:

clock frequency/$2^{\text{register width}}$

If the clock frequency is 20 MHz and the register has 32 bits, the unit frequency is 20,000,000/4,294,967,296, or 0.0046566 Hz. With those constants, you can set any frequency you want from less than I Hz to over 5 MHz with an accuracy of better than 0.005 Hz.

If you wanted an output of 12,345 Hz, you would divide 12,345 by 0.0046566 and get a frequency setting number of 2,651,068. In this case, the error in setting the output frequency is under 0.5 parts per million, probably much less than the error in the crystal frequency.

The practical upper frequency limit of an NCO is about 25% of the clock frequency. With careful filter design, that can be pushed to about 40%. What limits the clock frequency is how fast the chip can add wide numbers and how fast a DAC you can afford. Commercial NCO chips use tricks such as pipelining the addition to achieve clock rates in the 80 MHz region. With a fast enough DAC, they can generate sinewave outputs up to 30 MHz. Very expensive Gallium Arsenide NCO chips run about ten times faster.

Why Filter? The filter on the output of the DAC is important because, by virtue of the steps generated by the DAC from each sample, the DAC output contains 'image frequencies.' The most bothersome of these is equal to the clock frequency minus the desired output frequency. The output also contains further images at these two frequencies summed with multiples of the clock frequency, but they are less important.

If you want an output of 10 MHz and your clock is 50 MHz, then there is an image frequency in the output at (50-10) or 40 MHz. The output filter must reduce the amplitude of that unwanted frequency to an insignificant level. This is not too hard, but as you increase the output frequency, the image frequency gets lower. At an output of 20 MHz, the image is at 30 MHz. You would need a rather sharp filter to get rid of it and still pass 20 MHz.

People often ask me, “Doesn't the filter need to be tuned to the output frequency?” The answer is “In theory, no. Its job is to filter out frequencies above half the clock rate, and that is fixed.” In practice, if the clock rate were much higher than the highest output frequency you will ever use, it would pay to use a filter that cuts off just above your highest output frequency. This will give you a cleaner output with a simpler filter.

Choosing the filter is the one area of NCO design where you run into compromises. In general, sharp cut-off filters ring when you feed steps into them. This may not matter unless you phase modulate. Binary- phase modulation can cause full- scale steps in the sinewave output. A sharp cut-off filter will generate horrible spikes in the output waveform. It may be better to use a filter with a gentler roll-off and a better step response, although this limits the output frequency to a smaller fraction of the clock frequency.

Making Tunable Squarewaves. Oddly enough, a major use of NCOs is to generate accurate tunable square- waves. The filtered sinewave is fed to a zero-biased comparator chip to make a jitter-free squarewave whose frequency can

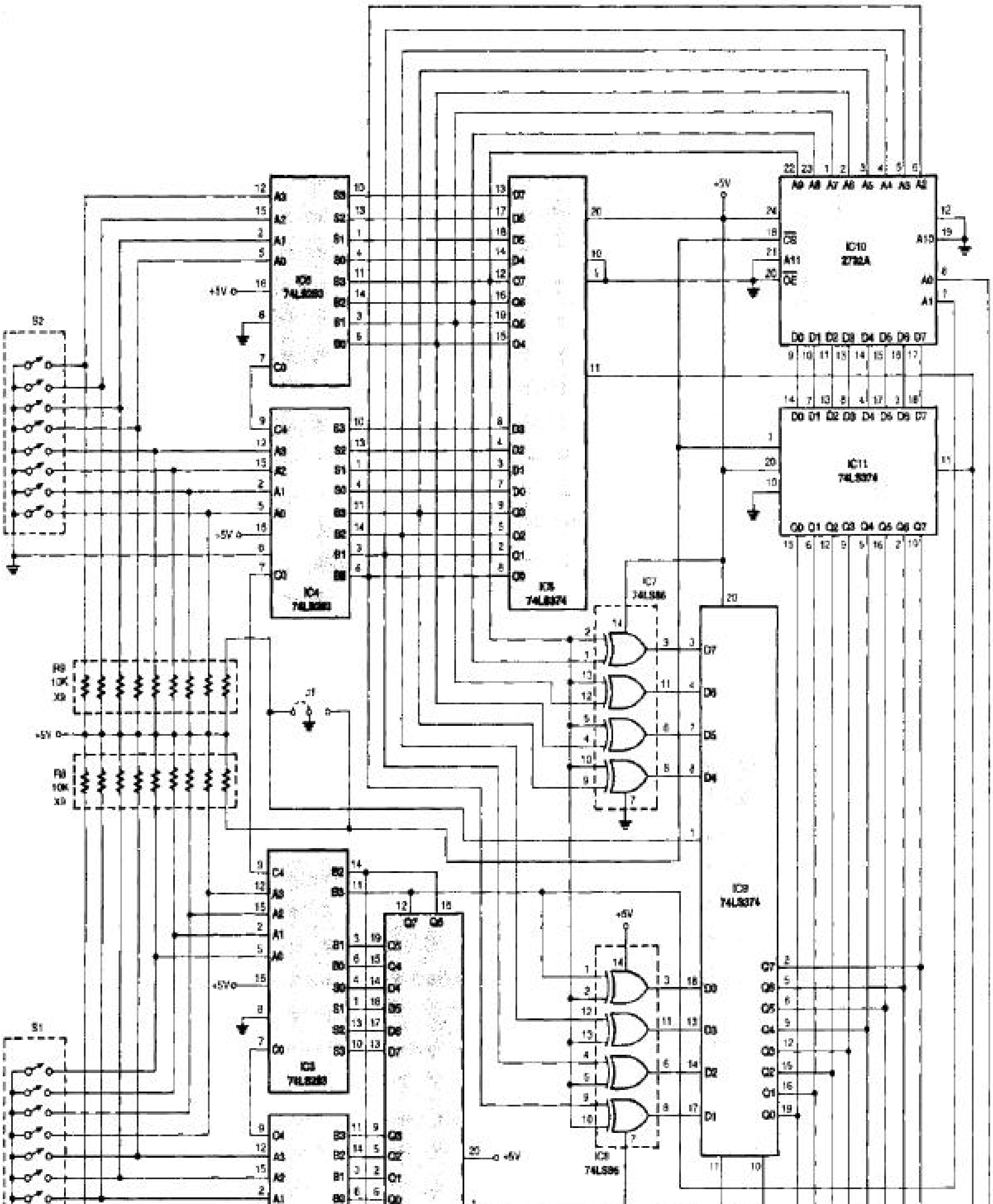

Fig. 2. The digital section of this NCO demonstrator uses registers, adders, and EXCLUSIVE-OR *chips. The EPROM is only needed if you want to experiment with a sinewave output.*

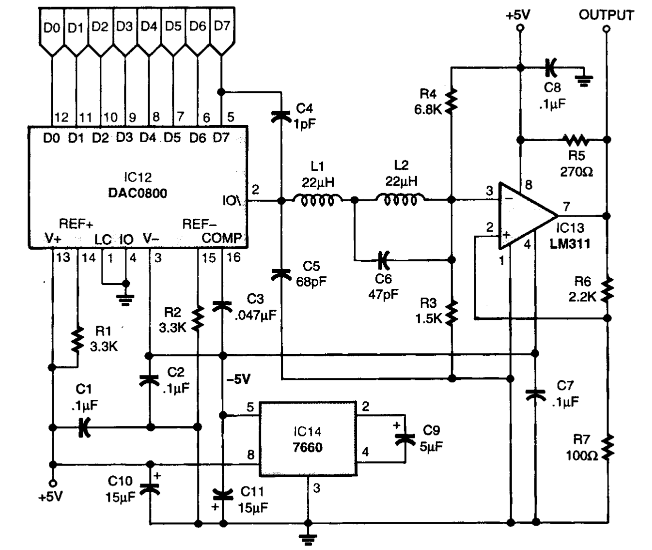

Fig. 3. This DAC, filter, and comparator circuit converts the ramp-up/ramp-down samples into a squarewave.

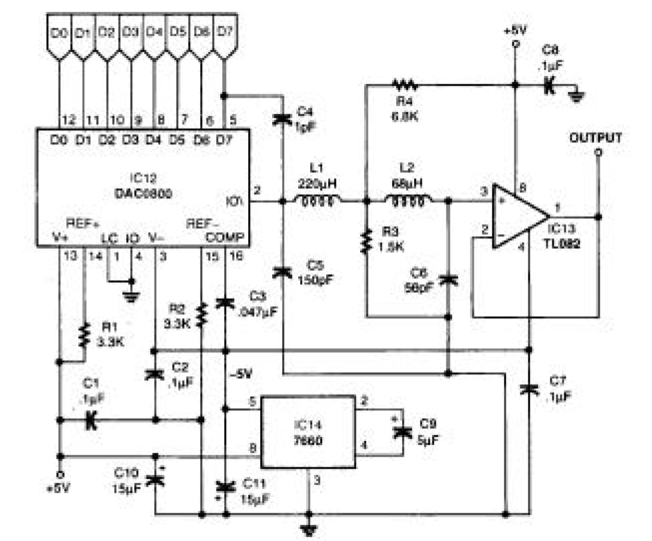

Fig. 4. Use this filter to get a sinewave output from the EPROM.

be set over many decades. In effect, the sine conversion and the filter interpolate between the fixed-time intervals set by the NCO clock. That makes a far more versatile and compact clock for a data-recovery system than the usual VCO and adjustable-divider circuits.

The problem with NCO chips is that not many companies make them, and you can't always get the combination of functions that you want. They also are not cheap- they range from about $15 to $150 each. Thus, users tend to look for alternative solutions. For example, it is quite easy to implement the phase accumulator part of an NCO in a programmable logic chip; the sine conversion part takes up space. That's what prompted me to find another way to interpolate the time intervals.

Doing Without Sine Conversion. Some years ago, I was using an NCO whose sine table was in an external PROM. This was a nuisance, so I looked for a way around it. The NCO's output was the raw ramp samples, but it could be switched to generate a ramp-up/ ramp-down waveform. Suppose I fed that NCO output to the DAC? Provided that the sample rate is greater than four times the output frequency, there will always be a line between two adjacent DAC output levels that crosses zero at the same point in each cycle.

As it happened, I was rather familiar with linear interpolation filters. Putting one of them on the DAC output generated a waveform whose zero crossings were exactly where the squarewave transitions should be. I tacked on a comparator chip and got squarewaves without using either a sine converter or a sharp cut-off filter.

This circuit gives perfect results up to 25% of the clock frequency. The squarewave doesn't become too jittery to be useful until you increase the output frequency beyond 35% of the clock, which is about where the conventional circuit also fails. The interpolation filter uses only two capacitors and two inductors, none of which is very critical. This circuit has another advantage: it can be easily built from TTL parts and standard inductors.

A Do-it-Yourself NCO. There are four ways to experiment with NCOS. You can buy an NCO chip and figure out how to drive it. If you have the tools, you can program an FPGA to emulate an NCO in hardware. You can even program a micro controller to emulate a slow NCO in software.

However, while no one would want to use so much board space in a commercial design, the cheapest and easiest way to get started is to build an NCO from TTL parts. Figure 2 shows the digital section of a basic 16-bit NCO made from registers and adders. The EPROM and its output register (IC IO and IC II) are optional.

The 74LS283 four-bit adder chips (ICI, IC3, and IC4) and the 74LS374 8-bit registers (IC2 and IC5) form the basic phase accumulator. This is one application where the pinout of the 74LS374 is more convenient than the straight-across pinout of the 74LS574. All chips have 0.1uF capacitors between their 5-volt lines and ground pins.

I used 8-bit DIP switches (SI and S2) with 10k pull-up resistors (R10 and R11) for the frequency input. You might want to use a more sophisticated input such as register chips linked to a computer's parallel port. Commercial NCO chips often use a computer-compatible byte-wide input to load the frequency setting register a byte at a time.

The ramp from the accumulator is converted to a ramp-up/ramp- down waveform by two 74LS86 EXCLUSIVE-OR chips (IC7 and IC8). I reclocked their output with a further 74LS374 (IC9) for two reasons. One is that the DAC output has smaller glitches when all of the input bits change synchronously. The other is that by disabling the outputs of the LS374, you can connect other sources, such as a sine generator, to the DAC. You don't need to use LS parts; HC or HCT series chips will work as well and might be easier to find these days.

A 4.9152-MHz crystal oscillator (XTALI) provids the clock. The adders can run at about three times that rate, but the DAC output gets messy if you go much faster. A 5 MHz speed is also about as fast as a sine-pattern look-up EPROM can run if you chose to add one.

This clock gives a unit frequency of 75 Hz so you can set any frequency which is a multiple of 75 Hz. The maximum useful output frequency is around 1.5 MHz. Using more adders and registers will improve the resolution by a factor of 16 per adder chip.

Figure 3 shows the DAC, filter, and the comparator circuit, which converts the ramp-up/ramp-down samples first into an analog ramp and then into a squarewave. Adjusting R4 can equalize the on-off ratio of the squarewave.

National Semiconductor's DAC0800 is a cheap ($2.20) 8-bit DAC chip that is just fast enough for this application. It does need a negative supply between -5 volts and -15 volts. I wanted to run this board off a single 5-volt supply, so I added an ICL7660 inverter chip (IC14) to supply a few milliamps at -5 volts. This rail also runs the LM31 I comparator (IC 1 3).

The filter shown here will interpolate linearly between samples that are 200 nS apart. If you change the clock frequency, you should change the inductance (L) and capacitance (C) values in proportion.

Since this circuit is only the start of an NCO-experimentation system, it pays to hook it up with wire-wrap sockets on a board with room for expansion. I wired the filter on top of a DIP socket so that I could unplug it and change things.

This was just as w 611 since I found two problems. One was that the comparator output chattered at low output frequencies. I had to add about 250 mV of hysteresis with R6 and R7; the other was that the 'major-carry' glitch of the DAC0800 comes just at the point where you want the comparator to fire. Glitches can't be filtered out, but they can be can- celled. Capacitor C4 in Fig. 3 is about I pF. It consists of two pieces of wire-wrap wire twisted together for about an inch. It couples enough of the digital input into the analog output to minimize the effect of the glitch.

Making Sinewaves. This circuit has a squarewave output. If you want a sinewave output; then, as shown in Fig. 2, you can wire in an EPROM that has been programmed to convert addresses to sinewave samples. The necessary code is available as an Intel hex file from the **Poptronics** FTP site ftp://ftp.gernsback.com/pub/pop/sinewave_pattern.hex

The sine table is only 1024 bytes long; which EPROM you use depends on what you can find and program, provided it has an access time of 200 nS or better. I used a 4-kB 2732A chip from an old computer board, but these are difficult to find now. You can use a 4-kB CMOS chip- the 27C32 (Digi-Key NM27 C32BQ200-ND) or the cheaper and faster 8k part, the 27C64 (DigkKey NM27C64Q- 150-ND). The latter part has 28 pins,, its unused address pins should be tied to ground.

Since the EPROM generates glitches every time the address inputs change, it needs an output latch (IC11) to get good results. Jumper Jl allows quick swapping from ramp to sine outputs. Wire pin 1 of IC9 to ground if you don't use a sine converter. For best results, you should change the filter to the one shown in Fig. 4.

Modulating The Output. Figure 5 shows how different types of modulation can be applied to the basic NCO. Adding frequency shift modulation is simple; just add a second frequency source (more DIP switches, for instance) and a bank of two-way multiplexer chips. Driving the multiplexer with a binary signal switches the out- put between the two frequencies you have set.

Quadratic-phase modulation is also easy. All it takes is two more EXCLUSIVE-OR gates between the adder and the existing EXCLUSIVE-OR bank. The EPROM can also be programmed to generate QPSK modulation from a signal applied to its address inputs.

If you want to experiment with continuous-frequency modulation, you need, to digitize the analog signal and add it to the DIP-switch input. You can generate narrow-band FM without adders by using the bottom bits of the frequency-control number as the modulation input and the top bits as the carrier-setting number. Putting adders between the ramp output and the EXCLUSIVE- OR gates implements continuous- phase modulation.

With a little experimentation, you will soon learn the value of these unique devices and will be ready to use an NCO chip in your own designs.

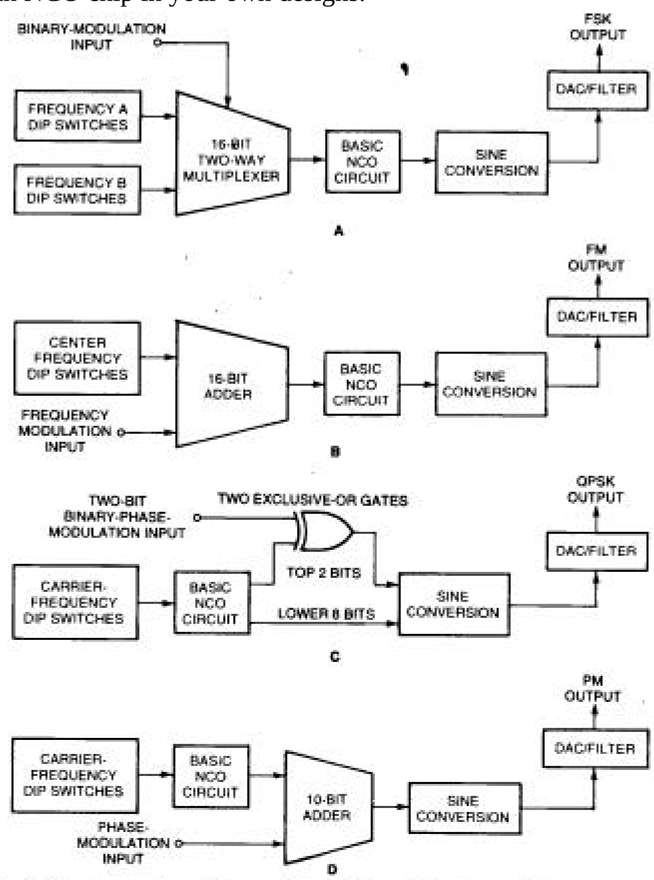

Fig. 5. These diagrams show additions to the basic NCO circuit for frequency-shift keying (A), frequency modulation (B), binary phase-shift keying (C), and phase modulation (D).

More Information About NCO Generators

"Digital Frequency Synthesis," *Circuit Cellar Ink,* October 1998 (www.circuitcellar.com). This article discusses how to generate accurate, modulated audio-frequency signals with a cheap microcontroller.

"Making Waves with NCOS," *Circuit Cellar* Ink, December 1997-January 1998. A signal-generator construction project using a Harris (now Intersil) NCO chip to generate both sinewave and squarewave output from I Hz to 10 MHz.

"Push Numerically-Controlled Oscillators Beyond Their Limits," EDN, September 12, *1997 (www.ednmag.com).* An introduction to NCOs and some ideas for extending their frequency range. This article includes more detail on generating a squarewave without a sine conversion.

Makers of NCO chips

Intel Corp.
350 E. Plumeria Dr. M/S XHP3-105
San Jose, CA 95134 408-545-9700
www.developer.inteLcomldesignldigitall

Intersil Corp.
2401 Palm Bay Rd. Palm Bay, FL 32905 888-468-3774 321-724-7000 www.intersil.com

Analog Devices, Inc. Box 9106
Norwood, MA 02062 781-329-4700 www.anaiog.com

PARTS LIST FOR THE NUMERICALLY-CONTROLLED OSCILLATOR

SEMICONDUCTORS

IC1, IC3, IC4, IC6—74LS283 4-bit adder, integrated circuit
IC2, IC5, IC9, IC11—74LS374 8-bit register, integrated circuit
IC7, IC8—74LS86 EXCLUSIVE-OR gate, integrated circuit
IC10—2732A 4-kB EPROM, integrated circuit (optional-see text)
IC12—DAC0800 8-bit digital-to-analog converter, integrated circuit
IC13—LM311 comparator or TL082 op-amp, integrated circuit (see text)
IC14—ICL7660 DC-DC converter, integrated circuit

RESISTORS

(All resistors are ¼-watt, 5% units unless otherwise noted.)
R1, R2—3300-ohm
R3—1500-ohm
R4—6800-ohm
R5—270-ohm
R6—2200-ohm
R7—100-ohm
R8, R9—10,000-ohm, 10-pin, single-inline package

CAPACITORS

C1, C2, C7, C8—0.1-µF, ceramic-disc
C3—0.047-µF, ceramic-disc
C4—1-pF (see text)
C5—68-pF or 150-pF, ceramic-disc (see text)
C6—56-pF or 47-pF, ceramic-disc (see text)
C9—5-µF, 16-WVDC, tantalum electrolytic
C10, C11—15-µF, 16-WVDC, tantalum electrolytic

ADDITIONAL PARTS AND MATERIALS

J1—Jumper block
S1, S2—Single-pole, single-throw, 8-device DIP-switch bank
OCC1—4.9152-MHz oscillator (Digi-Key CTX108-ND or similar)
L1—22-µH or 220-µH inductor (see text)
L2—68-µH or 22-µH inductor (see text)
IC sockets, wire-wrap board, wire-wrap wire, hardware, etc.