Lab Notes:

Shepard Functions

By: John S. Simonton, Jr.

The first audio illusion I ever heard was the Shepard Tone. Maybe you know it by the more descriptive term "barber pole" tone. It got that name because, like the stripes on a barberpole, it seems to defy the old saw "what goes up must come down". The effect is that of a continuously rising (or falling) tone which never resolves.

How the Shepard Tone works. There is nothing very mysterious about the Shepard Tone, as disconcerting as it can be at first, and if you've worked with synthesizers for a while you can figure out pretty quickly what's going on. The spectrum of the tone consists of a large number of octavely related components, all stepping up-scale together. The harmonics at the high and low ends of the spectrum have relatively low amplitudes, while harmonics in the middle of the tone are at maximum amplitude.

Imagine for a moment that you are following the lowest harmonic that makes up the Tone. At first the amplitude of this component is so low that it is, for all practical purposes, inaudible. But as it steps up-scale its level increases, peaking when its frequency corresponds to a point midway between the high and low limits. After peaking, the amplitude decreases as the frequency continues to step higher until finally, at the upper frequency limit, the harmonic is again inaudible.

When the harmonic reaches the high frequency limit it disappears, only to be replaced by a new harmonic at the lower limit.

Since the eight or ten harmonics which make up the Tone are all rising in a "staggered" progression, each in turn starting over again as it reaches its upper frequency limit, the overall effect is that of a tone which is constantly increasing in pitch while not actually getting any "higher" (or lower if the tones are all falling). It's an interesting illusion.

Shepard's original work used a computer program written by Max Matthews, but the same type of effect can be accomplished using analog synthesis equipment controlled by a gadget which, for lack of a better name, we may as well call a "Shepard Function Generator".

The Shepard Function Generator. Thinking about what happens with the frequency and amplitude of each harmonic of a Shepard Tone makes it easier to understand the composite sound. The frequency increases constantly and linearly from low to high, until the higher limit is reached. At this point, it begins again at the lower limit. This is a ramp function. The amplitude of any harmonic increases from the lower limit until it reaches the middle frequency, and then decreases as it approaches the upper limit. This can be a triangle function.

So, we need a gizmo which will produce a bunch of ramp waveforms (eight is a convenient number), and an equal number of triangle waves. The ramps and triangles both must have fairly precise phase relationships to one another, as summarized in the diagram in figure 1.

In the interest of conserving drawing time and space, I have shown only four of the eight functions pairs that our device will generate. I'm sure that you can see the pattern, and that the missing odd functions fit between the even functions that are shown. Notice that each function pair in the complete series is 45 degrees (pi/4 radians) out of phase with each of its neighbors. The even pairs shown are 90 degrees out of phase with one another.

Now, there are almost certainly lots of possible analog ways to generate these function

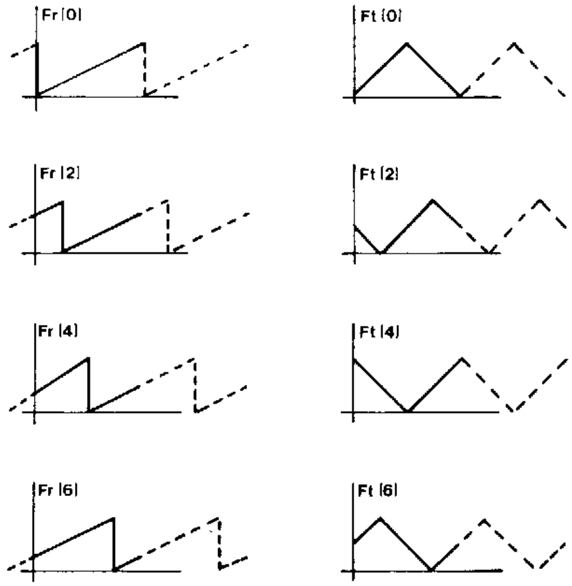


figure 1

pairs. But the simplest circuits I can think of to do this are too simple (for example, they wouldn't be able to generate the functions over a very wide frequency range), and what complicated approaches come to mind are very complicated. But I know of some workable digital approaches and I'd like to show you one, a discrete logic machine.

The day will soon be here when we wouldn't even discuss a logic machine approach to this problem. We would just truck ourselves down to our local electronics store and pick up a blister packed Single Chip Data Processor to be programmed on our trusty Home Data System.

No doubt, but let's look at a way to do essentially the same thing with counters, DACs, MUXes, and such...things we can get today. Figure 2 shows some components which we'll use in the Shepard Function Generator (as you probably realize, a counter connected directly to a DAC generates ramps). If the counter is counting up, upward sloping ramps come out. Having the counter count down, or inverting the counter output before it gets to the DAC, produces downward ramps (see figure 3).

Consider this: A triangle may be thought of as a ramp which changes its mind halfway up. If we replace the inverters in the figure above with Exclusive-OR gates, we can produce a single logic input that when high, causes the DAC to produce an upward ramp and when low, causes a downward ramp. By using the most significant bit of the counter as well as the control signal to the EX-ORs, the digital input to the DAC will count up until the MSB goes high, then it will count down -- in other words, a triangle function (see figure 4).

If we're interested in generating only a single function pair, it's a simple matter to pick up a new Least Significant Bit on the counter and use it to effectively switch back and forth between the circuitry of figures 3 and 4, producing first a small section of the ramp, and then a small section of the triangle. This new LSB also switches between two sample-and-hold circuits to de-multiplex the composite output of the DAC. Figure 5 shows how a little more logic gives Fr(0) and Ft(0).

I'm sure that we're together so far, and to make sure that we stay together I should mention a useful way to think of the eight Most Significant Bits of the counters. Think of them as phase, summarized in Table 1 below.

TABLE 1

Counter Output binary	hex	Equivalent Phase
00000000	$00	0 degrees
00100000	$20	45 degrees
01000000	$40	90 degrees
01100000	$60	135 degrees
10000000	$80	180 degrees
10100000	$A0	225 degrees
11000000	$C0	270 degrees
11100000	$E0	315 degrees

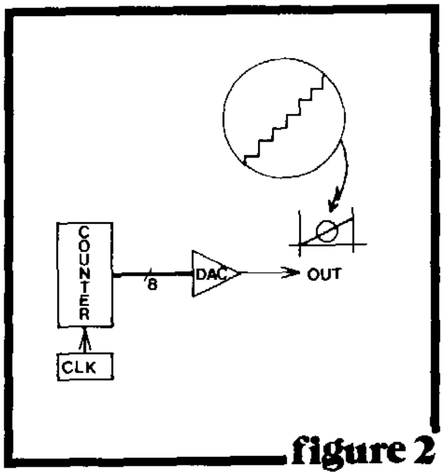


figure 2

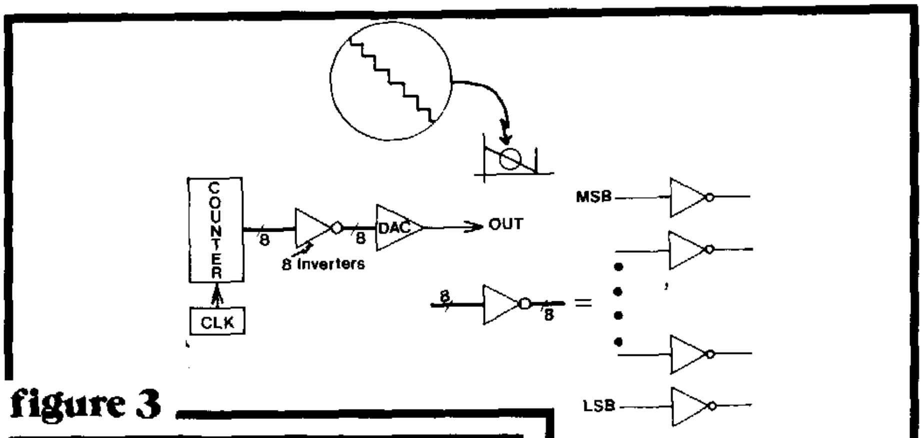


figure 3

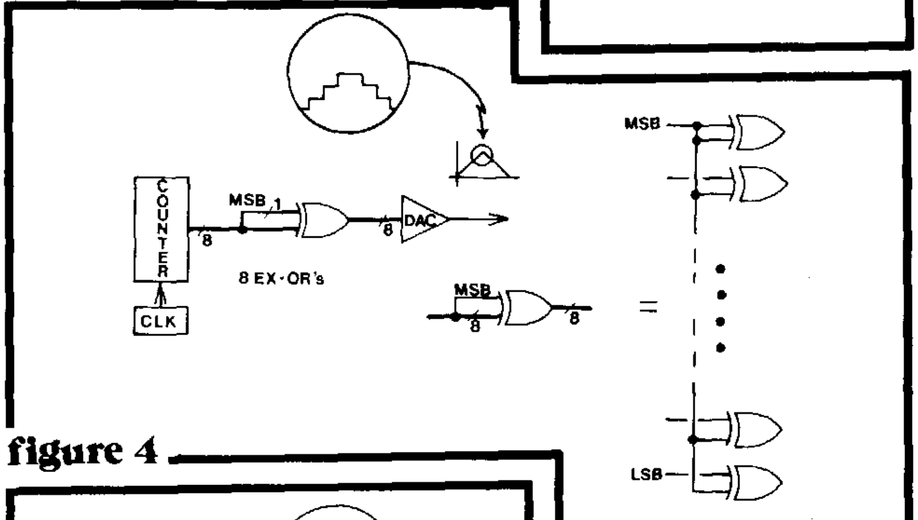


figure 4

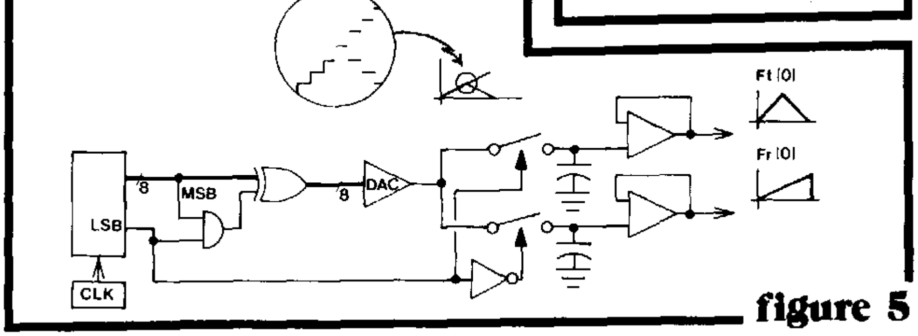


figure 5

If you're more comfortable with a graphic representation, see figure 6. The benefit of thinking of the counter data in this way is that phase shifts are produced by simple additions. For example, to shift the phase of the waveforms produced by the counter and DAC by 45 degrees, simply add $20 to the output of the counter. This is a pretty handy thing to know, particularly when it just happens that we are looking for a way to generate eight sets of functions which are 45 degrees apart.

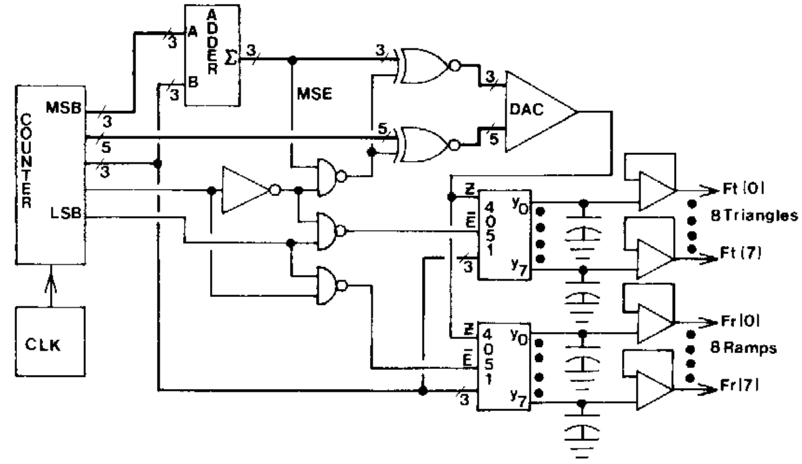
Figure 7 shows a block diagram of the complete Eight Phase Shepard Function Generator which results when we include an added IC to calculate digital phase offsets and de-multiplex the output with 8/1 analog switch ICs; figure 8 shows the schematic for the complete Shepard Function Generator. In the same way that the circuit of figure 5 alternately generated pieces of Fr(0) and Ft(0), the Shepard Function Generator sequentially puts out pieces of Fr(0), Ft(0), Fr(1), Ft(1), Fr(2).......Ft(6), Fr(7), Ft(7).

Details: Starting from the Most Significant end of the counter, the first eight bits of the counter serve the same functions that they did in the warm-ups. And we've decided to think of that function as phase. Unlike the previous sketches, these phase bits are broken down into two groups: the three Most Significant and the next five. If you don't see the significance of this grouping, review the binary representation in Table 1. To produce 45 degree phase shifts, the three Most Significant Bits are the only ones which change.

Below the eight phase bits, you'll see another grouping of three bits. Think of these as "offset" bits and notice that they are what's added to the three Most Significant Bits by the adder. And note that the offset bits also serve as address bits for the De-Muxes so that any given phase offset always gets strobed into the same sample-and-hold.

The next "Less Significant" bit can be thought of as switching back and forth between ramps and triangles, as in figure 5. And since figure 7 is a block diagram of our working Shepard Function Generator, and not just theoretical like the previous figures, the Least Significant Bit of the counter serves as a strobe which allows time for the DAC to settle

figure 6



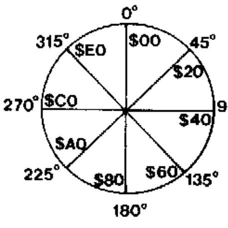
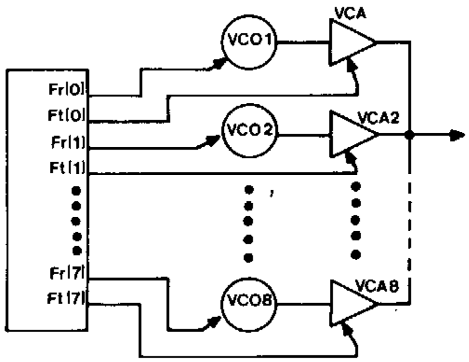
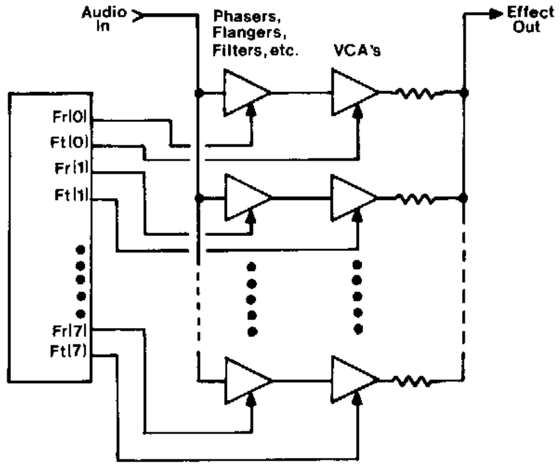


figure 7



VCO's and VCA's controlled by SFG produces shepard tones.

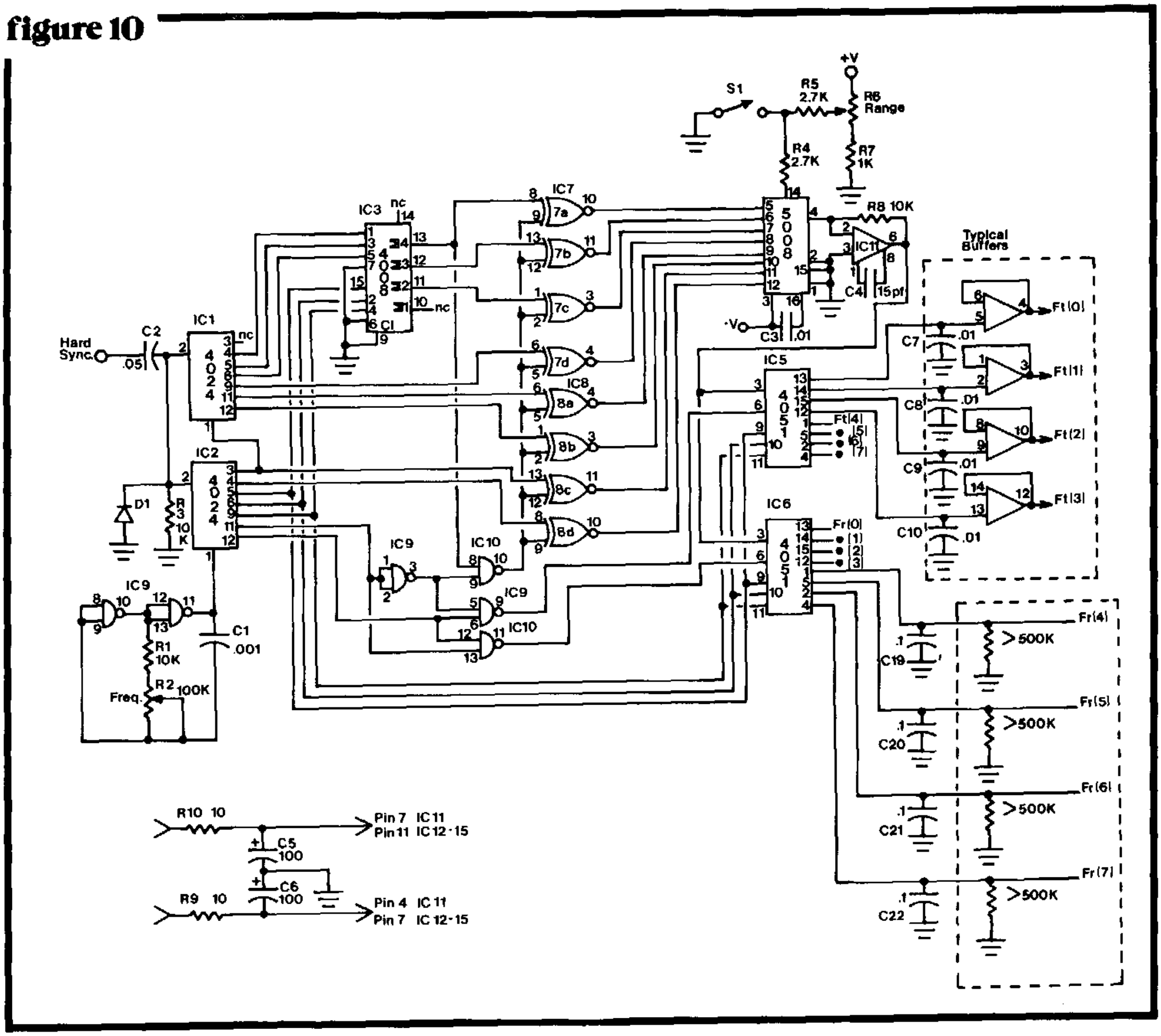
figure 8



Signal processing elements may be substituted for VCO's to produce interesting effects.

figure 9

figure 10



before selecting a sample-and-hold.

So, the Shepard Function Generator which we've developed isn't simple (though I would like to think it has a certain elegance) and when you consider that we'll also need eight VCOs and eight VCAs to produce the Tone (see figure 9), you might question if it's really worth the hassle.

But Shepard Tone generation is not the only application for this circuit; recently the same principles have been applied to other areas of electronic music. For example, the Barberpole Phaser invented by Harald Bode is a signal processing device which substitute phase change components applied to an external signal source for the frequency components of the Tone. The characteristic of multiple phase shifters are controlled by Shepard Functions so that the phasing effect doesn't simply swing back and forth, like we're used to hearing, but rather sweeps up and down eternally. It's really a most unusual effect and if it has occurred to you that the same principle might also work with other processing elements (such as filters, maybe) you're on the right track.

Figure 10 is the configuration such approaches would customarily take, with the ramp functions controlling the parameter being modified (phase, corner frequency, time delay, etc.) and the triangles controlling VCAs to fade the output of each modifier in and out.

You may have noticed that we're using gobs of equipment...lots of phasers or flangers or whatever. Chances are that you don't have eight flangers laying around. Even if you us the least expensive modifiers available (PAIA's EKx module series, for example) you will still have some bucks tied up in repetitive elements. For those who lament the fact that there don't seem to be any new effects, this one qualifies. It's unique all right, but worth the cost...?

Wait. We're being prejudiced by what we've seen so far (always a danger). We're thinking of the Shepard Function Generator only as a way to generate monophonic, noncyclic illusions by always using all 16 output functions to control 16 corresponding processing elements. But that's where we're getting of the track: You don't have to use all the outputs all the time, and the results don't have to be monophonic.

Now, there's no doubt that eight phase Shepard Functions are the absolute minimum number of components which will still preserve the "barberpole" illusion, but there are other times when sets of phase synchronized functions are useful. Is it obvious that any pair of triangles 180 degrees apart -- Ft(0) and Ft(4), for instance -- may be used with a pair of VCAs to give automatic stereo panning? Or that four triangles 90 degrees apart provide quad panning? With the arrangement shown in figure 11, the apparent "revolution" of the sound source is clockwise. To reverse the apparent direction, reverse either pair of corner sources.

Various combinations of triangles with unequal phase relationships may be used to produce effects which don't just swing round and round, but rush out of one of the "corners", swing around in front of (or behind) you to disappear into the other corner. When you start adding effects into this setup (such as phase shifters) under control of the ramps, as shown in figure 12, the sound really begins to move around you in some strange ways.

A nice thing about this is that the effects devices don't all have to be the same to produce interesting results. In fact, some of the most interesting results come from using completely different effects (such as phaser and echo) in opposite corners with only VCA processing on the other corners. While you might be hesitant to rush out and buy eight VCOs just to get a tone, you probably have enough modules or effects to get started. Voltage control is obviously preferable, but even effects which have only manual control are useful. Among other things, be sure to try synchronizing the frequency of the effects oscillator to the frequency of the Shepard Function Generator.

I think you get the idea: Play. Try different effects and different functions applied to different effects. Try controlling the VCAs with the ramps and the effects with the triangles -- try leaving out the VCAs altogether. Not all of the results will be particularly pleasant, but you will surely also find some that are unique beyond words. While many of these effects are somewhat less spectacular when done in stereo, they are still very effective.

This is getting long just when I could go on forever; but it has to end as soon as I draw your attention to the hard sync input. A positive pulse applied to this input resets the counter chain and causes all functions to start from the same known point. This feature will be particularly useful to us as Craig Anderton introduces us to Synchro-Sonic techniques in future issues of Polyphony.

No new effects indeed!

The following is available from PAIA ELECTRONICS, INC., P.O. Box 14359, Oklahoma City, OK 73116, (405) 843-9626:

Experimenter's kit of circuit board and electronic components (does not include hardware or jacks. specify No. EK-9 Shepard Function Generator Experimenter's Kit. $24.95 postpaid.

Etched, drilled and legended circuit board alone for the Shepard Function Generator Specify No. EK-9pc, $4.95 postpaid.

Visa & Master Card accepted ($10 minimum charge) or include check or money order with order.

Practical Circuitry

Build A Quadrature Function Generator

By: Tom Henry

What exactly is a quadrature function generator? To answer that, we must first understand the concept of a quadrature oscillator. An oscillator is said to have quadrature outputs if it produces simultaneous sine, cosine, negative sine, and negative cosine outputs. Such an oscillator can be created quite easily by setting a four pole lowpass filter into oscillation, and tapping the required outputs from consecutive stages[1]. If trigonometry alarms you, just consider the outputs to be four sine waves, each one ninety degrees out of phase with the previous output.

Despite the interesting patches possible with a quadrature oscillator, it does have one main drawback: Sine wave oscillators are fairly touchy, and you will often find that the amplitude will change as the frequency is swept over a wide range. In addition, under some conditions the oscillator may fail to oscillate; or, at the other extreme, hard clipping may bring about some undesirable distortion.

That's where the quadrature function generator comes in, since we will throw out the oscillator completely and replace it with a function generator. Oscillators are reactive; they depend on a resonating RC network. Function generators are non-reactive (in the sense that they don't resonate); their timing depends solely on the charging and discharging of a capacitor. With this circuit, the outputs are very stable in amplitude and purity over a very wide range. Finally, one more distinction between oscillators and function generators is that the former generates sinusoidal outputs, while the latter generates triangle waves (or sometimes ramp waves).

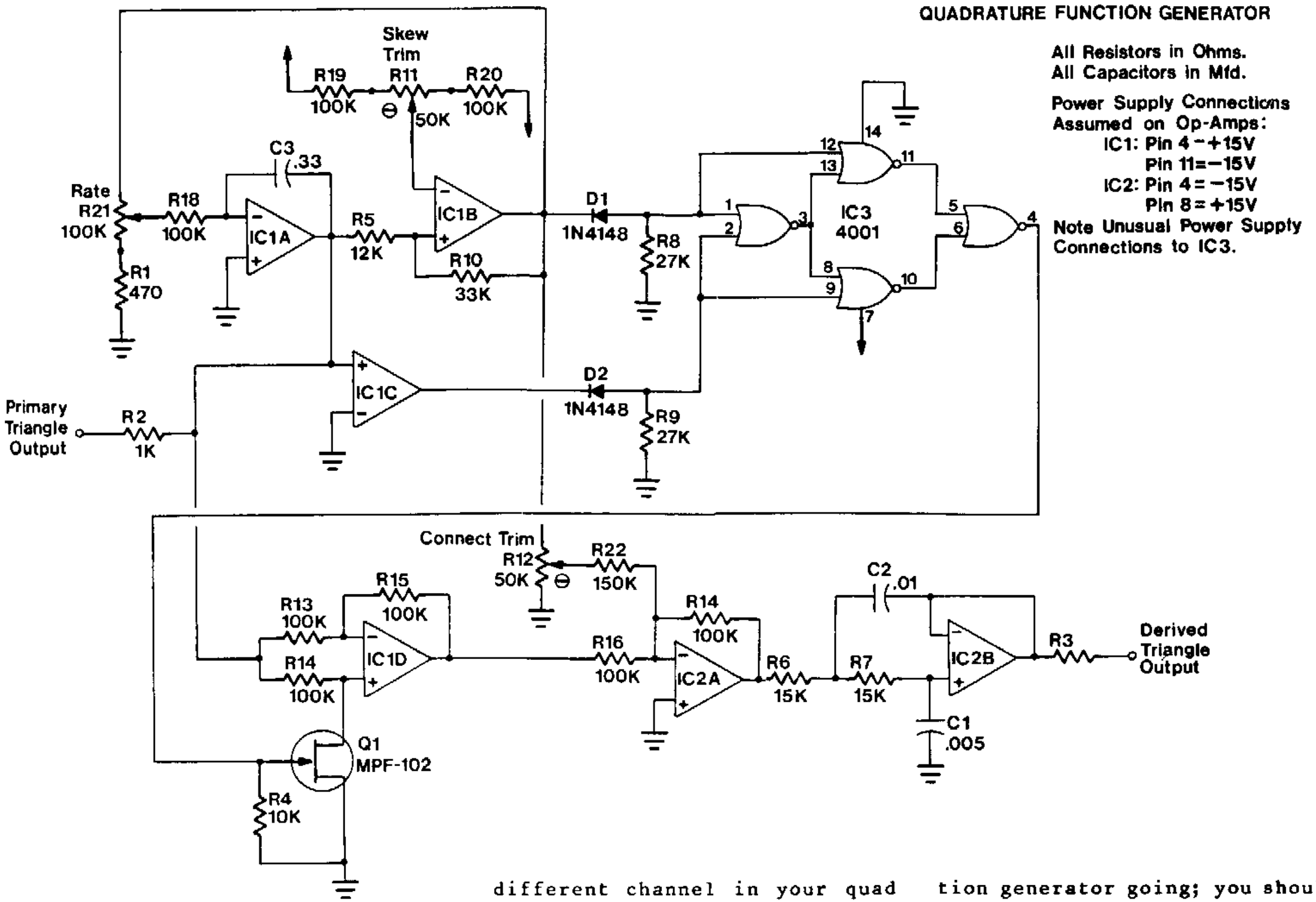
How it works. To fully understand the workings of a quadrature function generator, we must resort to some mathematics. There's hardly room to do that here, so if you're interested in the math behind the circuit, please refer to another article of mine which gives the complete analysis of a quadrature function generator[2]. (Actually, the circuit presented here is more compact and uses fewer parts than the earlier version, but the circuit action is very similar.) But even though we can't go through the mathematics here, we can still get an intuitive feel for how the circuit works. Referring to the schematic, op amps IC1A and IC1B form a Schmitt trigger/integrator function generator, an old friend from way back. C3 sets the basic frequency range, with R21 allowing for an adjustable rate. It is important to note that the output of IC1A is a triangle wave, and the output of IC2B is a square wave. The triangle wave goes directly to the "Primary Triangle Output" via R2. In addition, various line segments of the triangle wave are used in conjunction with Quadrature Function Generator's other circuitry to construct a new triangle wave ninety degrees out of phase with the first -- and that's where the square wave output from IC1B comes in. This output tells the circuit when to grab the various segments needed in the construction of the new triangle wave.

IC3 is configured as an EXCLUSIVE-OR gate; it seemed more cost-effective to use the dirt cheap 4001 quad gate for this, rather than using one gate out of a 4070 EX-OR gate. Note the unusual power supply hookup for this chip. Since this part of the circuit drives the gate of an N-channel FET, we need a swing from negative to ground.

IC1D, Q1, and their associated circuitry comprise a sign changer. This circuit will invert or not invert the input, depending upon the control voltage at the gate of Q1. Once again, see (2) for more details on the function of this sub-circuit.

Note that, so far, all of the op amps have each been 1/4 of a TL084 quad bi-fet op amp package. This particular chip must be used (instead of 741s, for example), since this circuit requires an extremely high slew rate -- we want all of the switching to be as clean as possible. But now we're going to reverse this philosophy and specify as slow an op amp as possible for IC2A and IC2B! IC2A sums together the various line segments to form a new triangle wave which is ninety degrees out of phase with the original. By specifying a low slew rate for this amp, any of the discontinuities in the derived triangle will be masked by the op amp's inability to slew fast enough. Pretty sneaky! In addition, IC2B is set up as a lowpass filter with a cutoff frequency of 1.5 kHz, which also helps smooth out the new triangle output.

Adjusting the trimpots is fairly easy. To simplify the process, temporarily replace C3 with a 0.05 uF capacitor. This will increase the frequency to a more easily observable range. Now monitor the "Derived Triangle Output" on an oscilloscope. While watching the waveform, go back and



QUADRATURE FUNCTION GENERATOR

All Resistors in Ohms.
All Capacitors in Mfd.

Power Supply Connections
Assumed on Op-Amps:
IC1: Pin 4=+15V
Pin 11=−15V
IC2: Pin 4=−15V
Pin 8=+15V
Note Unusual Power Supply Connections to IC3.

forth between trimmers R11 and R12 until the waveform "comes together" and connects to form a smooth triangle wave. This process is quite magical; I think the sight on the scope will really amaze you! If you have a dual channel scope, compare the two waveforms ("Primary" versus "Derived") and confirm that they are indeed ninety degrees out of phase with respect to each other.

To round out the circuit you will probably want to provide inverted versions of the "Primary" and "Derived" triangle outputs. This will give you a total of four outputs, each one ninety degrees out of phase with the previous.

Applications. Well, what shall we use it for? For a start, how about automatic quadraphonic panning: Gang the inputs of four VCAs together, and control each one by a different triangle wave. Send each output of the VCAs to a different channel in your quad system, and the result is circular location modulation. What?!? You don't have a quad system? Then you can still use the quadrature function generator for some neat stereo effects. For a really bizarre sound, try this patch: Gang the inputs of VCAs 1 and 3 and apply a dry signal to these VCAs. VCA 1 should feed the left channel and VCA 3 the right channel. Now gang the inputs of VCA 2 and VCA 4, and apply an echoed signal to these inputs. VCA 2 should mix into the left channel, while VCA 4 mixes into the right. Now really crank up the delay time and feedback and hit some staccato notes -- but don't get seasick!

What?! You say you don't have a stereo rig either? That makes it harder to think up patches, but here's a good one to try. Apply an audio signal to four different flangers, with each one set for a slightly different initial delay time. Then send each flanger output to a VCA, and finally sum the VCAs together into a monaural output. Set the function generator going; you should hear an incredibly dense and lush sound. This is especially good for full-bodied instruments, such as rhythm guitar.

You may think I'm getting ridiculous, but what do you do if you don't have any synthesizer at all? (I'm serious now!) Well, you can create some great Lissajous figures on an oscilloscope screen, or better yet, hook the unit up to your laser art show for a far out display. (What?! No laser? Well...) There's quite a lot this little black box can do.

I've had a real blast designing and building the quadrature function generator. The circuit has a real "that's neat!" aspect to it, and is lots of fun to play with. If you come up with some interesting applications be sure to jot me a line c/o Polyphony.

(Editor's note: Splitting a signal into four filters, whose outputs feed four VCAs controlled by the Quadrature Function Genera-

continued on page 30

Practical Circuitry

continued from page 27

tor, can make for fascinating timbral changes. Also, for those of you who are into synchro-sonic recording techniques, adding a CMOS switch in parallel with C3, and feeding the switch control terminal with an appropriate trigger pulse, will reset the oscillator at the rate of the trigger. All in all, for those experimenters who don't quite need the sophistication or versatility of John Simonton's "Shepard Function Generator" presented in the February 1982 issue of Polyphony, the Quadrature Function Generator provides a low-cost way to experiment with voltage controlled panning, cross-fading, channel-splitting, and the like.)

NOTES

(1) For example, see J. Patchell, "Build a Voltage-Controlled Quadrature Oscillator", Polyphony, Nov/Dec 1980, pp. 26-27.

(2) T. Henry, "A Function Generator With 'Quadrature' Triangle Wave Outputs", Electronotes #122, pp. 13-20.

PARTS LIST, QUADRATURE FUNCTION GENERATOR

Resistors (1/4 Watt, 5% tolerance preferred)

R1	470
R2,R3	1K
R4	10K
R5	12k
R6,R7	15K
R8,R9	27K
R10	33K
R11,R12	50K trim pot
R13-R20	100K
R21	100K potentiometer
R22	150K

Capacitors (15 or greater working Volts)

C1	0.005 uF, mylar preferred
C2	0.01 uF, mylar preferred
C3	0.33 uF, mylar preferred

Semiconductors

IC1	TL084 quad bi-fet op amp
IC2	1458 dual op amp
IC3	4001 CMOS quad NOR gate
Q1	MPF-102 N-channel FET
D1,D2	1N4148 or equivalent switching diode