POLYPHONY

April/May 1978

MODIFY YOUR MINI-MOOG

POLYPHONIC SOFTWARE

GUITAR PROCESSING

build a digital transposer

construction
modifications
patches

TIGER .01

Introduced three years ago, our "Tiger .01" is still one of the finest amplifiers available in its power class. This amplifier introduced our 100% complementary circuit which has become a standard feature in many of the better amplifiers. This combined with an output triple produces a circuit that can honestly be rated as having less than .01% IM distortion at any level up to 60 Watts. Relatively low open loop gain and a conservative amount of negative feedback results in clean overload characteristics and good TIM characteristics.

Other features are volt-amp output limiting, plus three fuses and an overheat thermostat. Despite the "budget" price an output meter is standard equipment. Each channel measures 4¼ x 5 x 14. Four will mount in a standard width relay rack for four channel systems.

SPECIFICATIONS

60 Watts—4.0 or 8.0 Ohm load Minimum RMS from 20 Hz to 20 KHz with less than .05% Total Harmonic Distortion.

IM Distortion .... less than .01%
Damping Factor .... 50 or greater 20 Hz to 20,000 Hz.
Hum and Noise .... −90 dB

#207/B Amplifier (single channel) .... **$110.00 PPd**
207/B Amplifier — Kit .... **$ 77.50 PPd**

Southwest Technical Products Corp.
219 W. Rhapsody, Dept. FM
San Antonio, Texas 78216

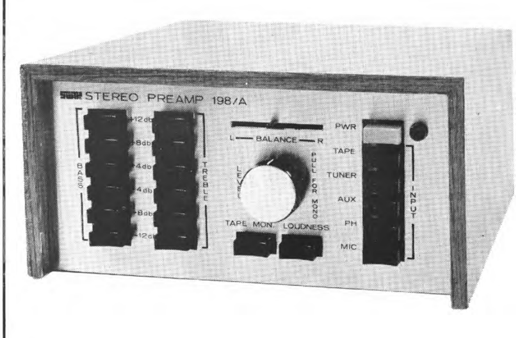
SWTPC Stereo Preamp

Would you believe a high quality preamp and control center for less than $75.00?? Well it's true. With a few hours work you can assemble our #198 preamp kit and have a unit equal to, or superior to products costing three, or four times this amount. Our kit features push-button tone controls, and extremely low distortion gain modules. Descrete transistors are used for lowest noise and distortion—no cheap IC's in this unit. Self contained 110 Volt power supply and wood grain finished cabinet.

SPECIFICATIONS

Sensitivity .... 2.0 Mv. for 1.0 Volt Output
Maximum Output................ 5.0 Volts RMS
Freq. Response .... ± 1 Db 10 Hz to 100K Hz.
Distortion.......... Less than .05% Harmonic or IM
Noise and Hum .... More than 65 Db down

 # 198 Preamp Kit .... $74.50 PPd



Southwest Technical Products Corp.
219 W. Rhapsody, Dept. FM
San Antonio, Texas 78216

STAFF

EDITOR:
Marvin Jones

ASSISTANT EDITOR:
Linda Kay Brumfield

CONTRIBUTING EDITOR:
John S. Simonton, Jr.

PRODUCTION:
Debbie Collins
Shirley McConnell

PRINTING:
Reggie Terry

Published Quarterly by:
PAIA Electronics, Inc.
1020 W. Wilshire Blvd.
Oklahoma City, OK
73116

Subscriptions

Subscriptions are accepted on a one year basis only - 4 issues.

Subscription rates:

Mailed in the USA	$4.00/yr.
Canada & Mexico	$5.00/yr.
International	$6.00/yr.

Address corrections requested.

For change of address: Be sure to send in your former address and zip code along with your new address. Also include the numbers at the top of your mailing label to enable us to make your address change as efficiently as possible.

POLYPHONY
PO Box 20503
Oklahoma City, OK 73156

CONTENTS

Volume 3, No. 4 1978

ON THE COVER: The wolf wails in a tone on tone illustration by David Lee Renfro of Lima, Ohio.

LETTERS:

"DEFENDING DIGITAL"

Dear Editors,

Your magazine is really getting slick! I thought I was getting Contempory Keyboard from the feel of the front cover. Your printing of pictures has really improved.

I'm afraid that I must take issue with the "back to music" letters in the Feb. 1978 issue. Such thoughts crop up many times in discussions I have with friends, and technology vs. music vs. musician is a hot subject.

I DO NOT feel that the magazine OR the company has left the idea. While you seem to have gone strongly digital, and the analogue seems to have slowed down, the goal is still the same - - MAKE MUSIC. Whether I do it by fingers or microprocessor is irrelevant. The beauty of your system is that I can do it EITHER WAY. Or both at the same time.

Audio is strongly going digital. Who cares? Anybody who likes music should. I call your attention to the "reworked" Carruso record, which in effect amounts to a remix job without a master tape. How about the digital tape and discs showing up at the trade shows? These show promise for previously unheard of levels of distortion and noise (namely NONE) and dynamic range almost exceeding natural events. How about the 'environmental' experiments (by JVC?). With proper processing and speaker placement, et al, walls seem to 'disappear'.

The point is, that anybody involved with the technical side of music (or sound), be he (she) synthesist, recording engineer, performing musician, or composer, etc., MUST keep up with the advances in his area of interest. If they deal with your company, they must be into synthesis, and synthesis is strongly digital, by nature.

But why should we have to learn digital electronics? I just want to PLAY! To answer a question with a question (A questionable idea at best): Why should the brass player know the harmonic response of a brass tube, and the results of changing its length? Why should the string player understand the vibration pattern of a taut string or

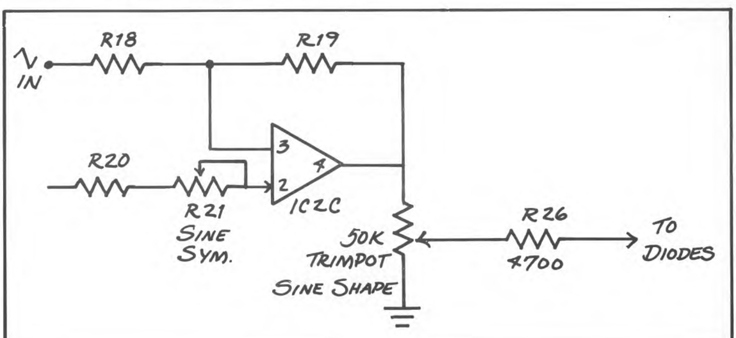
continued on page 37.......

CORRECTION

for "An Ultra VCO from a 4720" by Gary Bannister, Polyphony February 1978

The drawing of the conversion circuit for the Sine output shown in figure 6 should have shown the new 50K trimpot replacing both R24 and R25. The 4700 ohm R26 should then be connected to the wiper of the trimmer. This control allows an extra fine adjustment for minimum sine distortion. To set the control, you should first adjust R21 for proper symmetry. Then use the new 50K trimmer to adjust for minimum distortion, or the smoothest curve as viewed on a 'scope.

The 2 mfd bypass capacitors shown in figure 4 were non-polarized in the original prototype, however polarized electrolytics could be used if you first note the voltage which has been selected at the wiper of the offset trimpot and then install the electrolytic with the appropriate polarity.



MODIFY YOUR "PHLANGER" for lower noise

By Craig Anderton

I have become rather enamored of the flanging effect, and use the PAIA Phlanger not just for flanging but for chorusing, pseudo-doubling, and vibrato. However, one aspect of the Phlanger where there is room for improvement is the noise level. Bucket brigade devices like the SAD 1024 are inherently noisy, so there's not much you can do about that part of the circuit; but there is also some noise contributed by the input stage, IC1c, and the output mixer, IC1d. By replacing these stages with special, low noise op amps, it's possible to obtain a dramatic decrease in the Phlanger's noise level.

This is not a particularly messy modification, but you will have to be patient and careful ... remember that the goal of this exercise is to improve the Phlanger, not destroy it! So, follow the instructions carefully as we go along.

Here are the items required for the modification:

1) 4739 dual low noise op amp. Don't confuse this with the regular 739, which requires external compensation and is unsuitable for our application. The 4739 is manufactured by Raytheon (second sourced by Exar) and may be purchased from local stocking distributors, or is available by mail from Bill Godbout Electronics, PO Box 2355, Oakland Airport, CA 94614 for $1.65 (orders under $10 add 50¢ handling). See figure 1 for a pinout of the 4739.

2) A 14 pin IC socket for the 4739. It's best to use a mechanically strong circuit for this application... low profile types are not recommended; use 2 level wire wrap types or some other sturdy socket that can hold up to having its leads soldered to and generally messed with.

3) A .1 to .22 uF ceramic or mylar capacitor; any working voltage over 10V is fine.

We might as well start the modification by doing the roughest part first. We need to disable and remove pins 8 through 13 of IC1 on the Phlanger board from the component side. IC1 is either a 3900 or a 3401, and is located towards the right hand side of the Phlanger. I luckily had the forethought to mount this IC in a socket when I built the unit, so disabling the pins simply involved unplugging the IC, cutting off pins 8-13, and re-inserting the IC back into its socket. If your IC1 is soldered to the board, you can either use a very fine point diagonal cutters to clip off leads 8-13 very close to the board, then bend the remaining part of the pin up horizontally (see figure 2); or, if you have the patience, unsolder the IC (or at least the side with pins 8-13) from the board, cut off the appropriate pins, and reinsert into the board. Be careful when unsoldering the IC that you don't overheat it (which can cause failures) or overheat the board (which can cause the traces to lift up). Also, make sure during this operation that when it's all over pin 14 is still connected to the board, since that's the pin that carries power to IC1.

OK, now we have pins 8-13 disconnected from the circuit. Next step is to remove R16, the 470K resistor, and replace it with a .1 uF capacitor. This modification adds extra decoupling on the bias line, and keeps noise carried on the power supply lines out of the input stage after the 4739 is wired in.

Next step is to take your 14 pin socket and solder eight 1-1/4" - 1-1/2" lengths of flexible, thin, stranded wire to pins 1, 5, 6, 7, 8, 9, 13, and 14, as shown in figure 3.

Figure 1: 4739 pinout

Output, amp 1	1	14	(+) supply
No connection	2	13	Output, amp 2
No connection	3	12	No connection
No connection	4	11	No connection
+ input, amp 1	5	10	No connection
− input, amp 1	6	9	+ input, amp 2
(−) supply	7	8	− input, amp 2

Table 1: Wiring the 4739

Connect the wire from	pin 1 of the 4739 to pad 9 of IC1 on the foil side of the board.
Connect ...	pin 5 of the 4739 to pad 12 & 13 (see figure 4) of IC1
Connect ...	pin 6 of the 4739 to pad 8 of IC1
Connect ...	pin 7 of the 4739 to pad 7 of IC1
Connect ...	pin 8 of the 4739 to pad 11 of IC1
Connect ...	pin 9 of the 4739 to pad 12 & 13 (see figure 4) of IC1
Connect ...	pin 13 of the 4739 to pad 10 of IC1
Connect ...	pin 14 of the 4739 to pad 14 of IC1

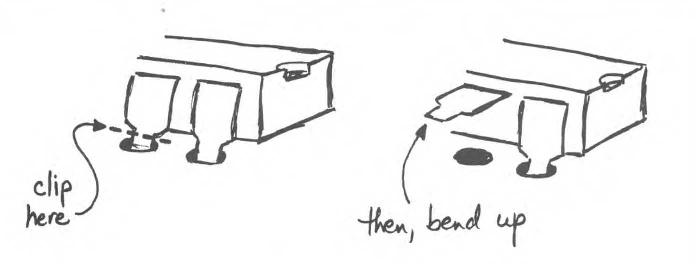


Figure 2

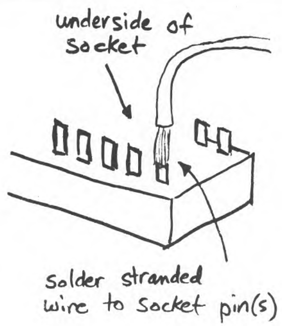


Figure 3

Now, from the foil side of the board, hook up these wires to the pads formerly associated with pins 8-13 of IC1, in accordance with table 1. Then hook up the power and ground lines, pins 14 and 7. The wires coming from pins 5 and 9 of the socket should have their ends twisted together; put these combined leads inbetween pads 12 and 13 of IC1, and let solder flow over both pads and both pieces of wire (see figure 4) so that pin 12 and pin 13 are connected to each other as well as the two wires. After checking that all the wires from the 4739 connect to the right pads associated with IC1, plug the 4739 into its socket.

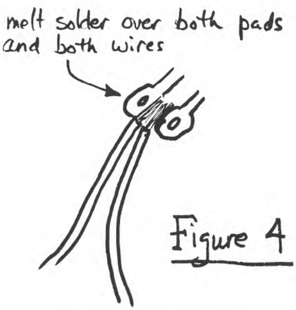


Figure 4

You are going to have to re-calibrate the Phlanger to complete the modification. I recommend using a sine wave generator or PAIA VCO for calibration. Plug in the Phlanger, listen to the output on an amp, and feed the generator into the input. Obviously, the case has to be open to do this, so make sure you don't come into contact with any "live" parts of the circuit.

Set the controls so that speed, accent, center, and span are fully counterclockwise, with the mix control set full clockwise. Turn up the sine wave generator to a low level; don't be surprised if you don't hear any sound yet, as the bias trimpot, R5 must be properly set for everything to work. Adjust R5 until you obtain a clean, undistorted sound. You will note that turning the trimpot to the left or right of this position will give distortion, and eventually the sound will poop out altogether at either extreme. Now, turn up the signal generator a bit more, and re-adjust the trimpot for minimum distortion. Eventually, you will reach an input level where there is no undistroted setting of the trimpot. Leave the trimpot where it is -- you now have minimum distortion consistent with maximum level. If you don't have a signal generator or VCO (although the odds are admittedly remote that you don't if you're reading this!), plug an instrument like electric guitar into the Phlanger's input and simply adjust R5 for minimum distortion consistent with maximum level. While you're at it, you may want to readjust R6, the balance control, since R5 and R6 are somewhat interactive. Referring to page 10 of the Phlanger instructions, adjust trimpot R27 and turn up the span control (with the speed control full clockwise) until you obtain a "wheep-wheep-wheep-wheep" sound; then adjust R6 for a minimal amount of "wheep" level.

Well, your Phlanger is now modified for low noise operation. Before putting the case back together, sandwich a piece of foam around the 4739 and its associated socket so that it's held firmly in place between the foil side of the circuit board and the case bottom (see figure 5).

The difference in noise level is most dramatic, and shows up the most under studio conditions where every decibel counts. Now instead of having one of the neatest flangers on the block, you can have one of the quietest ones, too.

P.S.: You may have noticed that grounding the cancel control line gives a bit of a thump at the output. You can remedy this by locating the end of R10 that connects to the R11-C5-R12 junction, and inserting a .22 uF capacitor in between this end of R10 and the junction (see figure 6). Also add a 1M resistor from point C to ground. Now your cancel line is as quiet as your front end and output mixer. ■

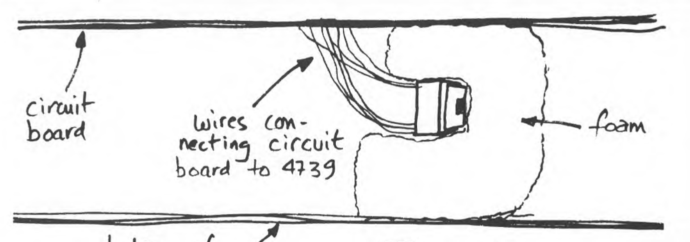


Figure 5

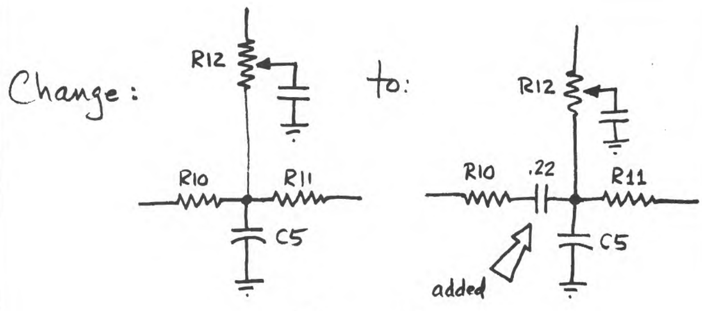


Figure 6

MUSIC MODULES FOR NON-KEYBOARDISTS

By Marvin Jones

Walk into a room full of musicians and mention the word synthesizer. Chances are that you will meet with about as many opinions as there are people in the room. Obviously, the keyboard artists are going to be pretty much in favor of the device. And why not?! Most of the synthesis equipment rolling off the assembly lines is aimed directly at the keyboardist. With the widespread use of synthesizers in bands, many of the other musicians have developed an interest in the possibility of using synthesizer equipment with their own instrument. However, in many cases the musician gets a bit confused when confronted with all the new terminology. But even worse, he doesn't know where to turn when he walks into his local music store, heads to the synthesizer department, and only sees row after row of keyboards staring at him. Well, let's take a look at some of the effects that are available to non-keyboard musicians, and how to use them.

The body of musicians which would most easily be initiated to the world of synthesis are probably the guitarists. Actually, guitarists were probably the first modern synthesists due to the barage of effects boxes which were created for them back in the '60s. Fuzz boxes, wa-was, treble boosters, octave boxes, and talk boxes were all available before the turn of the decade for the adventurous experimental guitarists. And in recent years, even more effects units have become available. Therefore, it should be relatively easy to introduce the guitarist to a modular, patchable effects console which would eliminate most of the stand-alone effects gadgets he has lying at his feet. Using synthesizer modules will offer the guitarist increased flexability, future expansion capabilities, and possibilities for increased complexity of effects.

A closely related group of instruments includes the instruments which can have contact mikes or pickups installed to convert them to electric instruments. These include violin, saxophone, banjo, autoharp, acoustic guitars, trumpet and many more. The module system array for these instruments would be the same as for guitar, and would operate in the same manner, but the musicians would probably require a bit of introduction to the techniques of signal processing.

A third group of potential synthesists is drummers. Drum sensor systems are actually rather easy to build, and can consist of custom built "drum-type" sensor pads or they could actually be converted acoustic drums with some type of pickup mounted on them. The module array for a drum package would be slightly different from a guitar-type system, and it would be operated in a different manner also.

Let's take a look at the types of modules which are particularly suited to use as processing elements for external signals. First of all, the audio processor modules such as VCFs, VCAs, reverb, and balanced modulators are immediately applicable to external signals. All that is involved is patching your external signal into the module rather than the typical synthesizer signal source such as VCOs, or noise sources. The VCFs can be used to do wa-wa effects, pseudo-phasing, treble or bass boosting, and much more. A VCA acts like a voltage controlled volume control, and lends istelf to effects such as volume pedals, attack delay units, generation of new percussive or soft envelope timing structures, noise gating, and tremelo or amplitude modulation effects. Initially, you may feel that a reverb unit in an effects system may be a redundancy if you have a reverb unit built into your amp. Not so! With a separate reverb unit which can be patched into various points throughout the effects unit, a myriad of effects can be obtained which are absolutely impossible with a standard guitar amp reverb. Balanced Modulators can do a number of jobs from giving your instrument a "metallic" sound, to making the strings of your guitar sound like rubber bands, and so on. One of the biggest problems with balanced modulators is that they so drastically alter the input signal that a great deal of time and patience must be spent on trying to use the effect artistically - and not overdoing it.

Controller modules are required to "tell" the processing modules what you want them to do. There are a wide variety of controller modules available, and when used in conjunction with each other, some incredibly complex control functions can be programmed. One of the simplest controller devices is a bias supply such as in the Paia 2720-7 Power Supply. A bias supply is used to set a fixed voltage output which you won't need to change. For example, you could use a bias supply to control a VCF. The result would be like a tone control - set the bias control to get the desired amount of "sharpness" in your tone, and then leave it. Another type of controller which should be familiar and easily implemented by the guitarist is a Foot Pedal control. These are most commonly seen as volume pedals and wa-was. Minimal parts and labor are required to convert a basic volume pedal into a multi-purpose pedal which can process audio or control voltages, or can be used as a variable source of control voltage with the pedal supplying it's own voltage source. For more information on this easy modification see issue #2-76 of Polyphony, page 9. It is highly recommended that the prospective "guitar synthesist" invest in one of these goodies as well as the following item. These two devices are going to be the most important in the entire modular system. An Envelope Follower is the one module which is directly responsible for allowing the guitar or other external signal to trigger and control the various parameters of the effect you have programmed. Let's initially take a few lines to define our terms. An Envelope Follower is a circuit which senses the "volume" of an input signal and puts out a varying DC voltage which is proportional to the "volume" changes at the input. Most envelope followers will also put out a trigger gate or pulse which can trigger external equipment whenever a signal is present at the input. When the signal drops below a certain level, the trigger output will then disappear. NOTE that the envelope output is a <u>variable</u> fluctuating voltage from 0 to about positive 5 or 6 volts, while the trigger output is always a fixed voltage jumping from 0 volts to positive 5 volts or more. One point of frequent confusion is the commercial availability of certain types of sound modifier products which are also called envelope followers. These are filter devices in which the filter (or wa effect) is variably swept as the input signal is changed in amplitude. In actuality, these devices are envelope followers which are driving VCFs. They are NOT just the basic envelope follower, and in most cases they don't make the variable voltage and trigger available at the rear for use with external voltage controlled equipment. Another point of importance is the frequent confusion between envelope followers and pitch (or frequency) followers. Recent (so-called) guitar synthesizers frequently feature pitch followers and circuitry which allows the machine to determine which note is being played by the guitarist, converting it to an appropriate control voltage, and using the voltage to drive a VCO to a pitch identical to what the guitarist originally played. For the purposes of this article, we need not get into how this works — or whether it even works at all. This would take an entire article in itself. The important point is that the Envelope Follower does not sense pitch changes at all, only changes in amplitude.

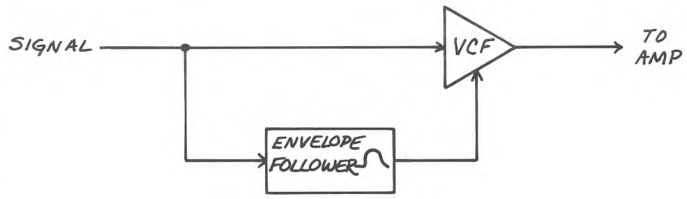


FIG. 1: ENVELOPE DRIVEN FILTER

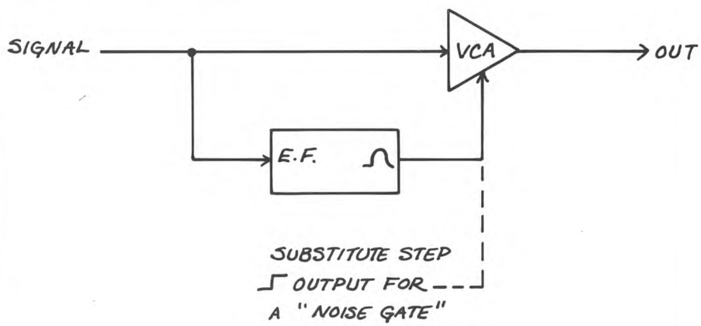


FIG. 2: EXPANDOR

Using only the envelope follower variable output in conjunction with the previously mentioned audio processing modules, several interesting effects are available. With the envelope output feeding a control input on a VCF, the envelope controlled filter effects which are currently so popular can be easily obtained. Every time a guitar string is played, the envelope follower will detect the sudden increase and gradual decay of the guitar output and will output a voltage which will cause the filter to suddenly jump to a "sharp" high frequency setting and gradually fall to a mellow, bassy filtered tone. See figure 1. This is basically the effect which the commercial "envelope follower" devices create, however, we can already begin to appreciate the added power of a modular effects system by realizing that not only the standard bandpass filter type of wa-wa is available, but also low pass and high pass outputs if you are using a multimode filter. Additionally, most filters will have adjustments available for filter "Q" or resonance. This is something you won't find on most of the inexpensive envelope-filter units! And the versatility has only begun. Note that most modules have multiple control voltage inputs so you can sum together several independent control sources. With this in mind, you can use your control voltage pedal to operate the VCF as a wide-range, multimode wa-wa pedal ... perhaps even in conjunction with the envelope follower. Another variation could be used to get pseudo-phasing effects. Through the use of an external mixing circuit, the lowpass and highpass outputs of a multimode filter can be combined to create a NOTCH filter which will sound very much like a simple phaser. Using your foot pedal, you can achieve manually controlled phaser effects rather than the fixed periodic sweep of most commercial phasers. Or, how about using the envelope follower to get phasing sweeps which are proportional to how loud or how hard you are picking. I always think it makes an effect seem much more like an integral part of the music when it is directly related to the dynamics of the music such as in this set-up. Give it a try. I think you'll find it a bit more appealing than the repetitive sweeping of a phaser over its entire range.

When the VCA is substituted for the VCF above, the two primary modes of operation are as an expandor and a volume pedal. When the foot pedal is used to provide the control voltage required to vary the VCA level, it looks like a waste of equipment since the pedal itself can be used as a volume control. The primary advantage would be that rather than feeding an audio signal across the stage (perhaps alongside power lines or lighting equipment lines, which are notorious for inducing noise in audio lines), you are feeding only a slowly varying DC voltage. Your audio signal remains inside your little box of modules, safe and sound. An expandor is generated by using the envelope output to vary the VCA level (See figure 2) in proportion to the input signal, causing the original dynamic range to be increased by a factor of two. This gives the signal added punch and definition. Using the envelope follower step trigger output to control a properly calibrated VCA will act as a simple noise gate

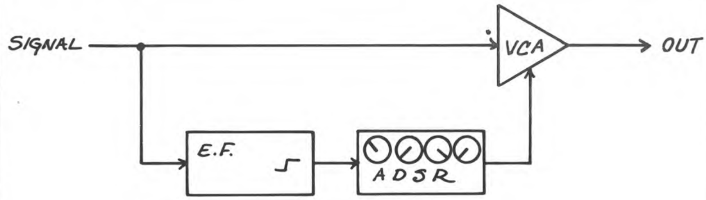


FIG. 3 : ATTACK DELAY OR "BOWED" ENVELOPE SHAPING

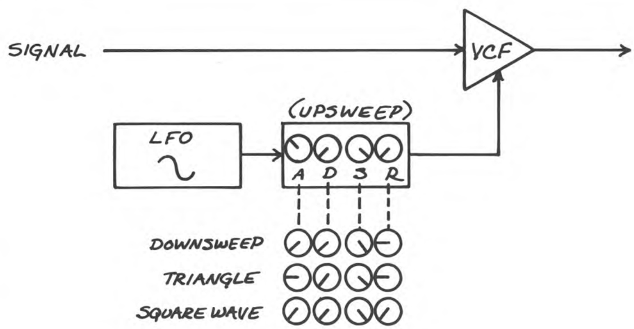


FIG. 4 : MULTI-WAVEFORM FILTER SWEEPS

device. NOTE that the VCA must be properly calibrated, otherwise the fast switching of the step trigger being fed into the VCA control input may cause a pop. If you are using a number of signal processing devices which are adding to your noise level, a noise gate could be the answer to your quest for silence between songs!

From this basic starting point, additional controller modules can be added to make your special effects unit a bit more 'super'. Adding an ADSR Envelope Generator will allow you to preset a timed sequence of sweeping voltage which can be used to control any of the modules which are voltage controlled. When an ADSR is triggered, the output begins rising from 0 volts towards 5 volts at a rate that is set with the Attack control. When the Attack peak is reached, the voltage Decays at the same rate you selected until the present Sustain level is reached. At this point, the output holds until the trigger voltage is removed from the input. The output voltage then falls through the Release period until the output is again 0 volts. Using these characteristics, the step or pulse trigger output of the envelope follower can be used to trigger the Envelope Generator, which will in turn sweep a filter, vary the level of a VCA, and so on. The effects achieved with these types of patches are similar to the use of the 'envelope' output from the Envelope Follower, except you no longer have the "randomness" related to the varying input amplitude. Each generated envelope will have exactly the same timings, dynamics, and characteristics. Using the Envelope Generator to control a VCA can produce some good "bowed" string effects or reverse tape simulation if you set the Attack for a moderate to long timing. See figure 3. This patch also tends to work best if the Release is set to minimum. This way, the Envelope Generator will reset immediately after the input signal has been damped or removed, and the system will be ready for the next note. The ADSR/VCA combination will also generate a noise gate which is much less prone to popping as mentioned earlier. This would be the recommended noise gate configuration.

Adding a Low Frequency Oscillator to the system will allow any of the voltage controlled parameters to be periodically swept at various rates and with various waveforms. We could go back to the NOTCH filter which we discussed earlier, and use a LFO to continuously sweep the effect, just like the commercial phasors. Alternatively, the LFO could be used to automatically sweep the VCF or control the VCA for tremelo effects. By itself, the LFO as a controller tends to be a bit (dare I say it) boring. BUT, when summed with or processed by additional modules, this can be one of the most useful circuits. To get an unending variety of low frequency waveforms, the LFO output can be used to repeatedly trigger an Envelope Generator. With minimum Attack and medium to long Release, a falling ramp wave can be generated. Conversely, an upward ramp is easily generated. These techniques can be used to also generate triangle-type waves and square waves for control purposes. And I haven't even mentioned repeating ADSR shape waveforms yet! There's a lot of fun to be had with this patch! See figure 4.

Adding the Inverter will greatly increase the possibilities from any given set of modules, as you will now have not only the original audio and control waveforms available, but also their inversions. Instead of upward deflections of the filter with an Envelope Follower or Envelope Generator, you will now be able to achieve downward sweeps. Using any of the controller devices, you can feed the original controller output to one effect module (let's say a VCA), and feed the inverted control signal to an identical effects module which is connected to a second output amplifier. In the case of VCAs, this will cause a panning effect as the controller output changes. See figure 5. If the controller is an Envelope Follower or Envelope Generator, you will get one "pan" for each note. With a LFO, you will get continuous panning, and so on. This same patch could be used to route the signal between two different effects or to manually select the effect "depth" with a foot pedal control voltage.

Another interesting mode of operation is to use the input signal (guitar, mike, or whatever) to control some other signal. As an example, we can use the white noise source which is built in to the Paia LFO to augment the basic input signal. If the input is detected by an envelope follower, the trigger output (pulse trigger this time) can be used to trigger an ADSR which is set for minimum attack, minimum decay, maximum sustain, and 25% to 50% release. The resulting envelope can be used to control a VCA which has the noise source as an input. The VCA output is then fed to the external amp along with the input signal. The result? Every time a note is played on the instrument, it will sound as if a cymbal was played simultaneously. There will be a short decaying burst of "hissing". To use this technique in a more interesting patch, substitute the noise source for a reverb unit. Set the reverb output mix so you get ALL reverb and no normal signal. Feed the input of the reverb with the input signal which is feeding the Envelope Follower. The input signal should also be fed to the final amp as usual. Now, every time a note is played, there will be a sudden burst of reverb which will shortly die away. See figure 6. This patch can give a kind of spooky effect, because it sounds as if the size of the room is changing as you are playing. One more variation of this patch would be to delete the ADSR, and use the variable envelope output to directly control the gain of the VCA. This gives the same basic effect as before, except the effect depth is now responsive to your playing technique, breath inflections, and dynamic variations. Don't forget the use of the inverter for applications such as these, also. The inverter could be used to invert the envelope control signal (whether it is from the envelope follower output, or from the envelope generator makes no difference) such that there would be NO reverb when an input signal was present and the reverb would then fade in as the input signal got softer or was removed. Bizarre, to say the least!

All of these effects could be achieved with what I would consider to be a basic processing package. For starting your experiments in signal processing synthesis you should get a 4761 wing cabinet, a 2720-7 power supply (so you'll have the bias supplies for fixed parameters), a 2720-11 envelope follower, a 2720-12 inverter, a 2720-1 VCA, a 4712 reverb, a 2720-5 LFO/noise source, a 4740 ADSR, and a 4730 multi-mode filter. Also remember that a foot pedal should be included as an important part of the system even though it is not part of the module cabinet. With this system, you should be able to blow away most of the people who are attempting processing synthesis using an array of foot pedal effects devices.

Another common question from people interested in signal processing concerns an easy way to use footswitches to turn an effect on and off. The best way involves using the footswitch to actually supply a control voltage or logic signal rather than actually switching the audio signal. The reasons for this are the same as when we were discussing use of a pedal and VCA for a volume pedal. Rather than get into all the details of analog switches and how to build a footswitch system, I refer you to Craig Anderton's book, "Electronic Projects for Musicians" ($6.95 from Guiter Player Productions, PO Box 615, Saratoga, Cal. 95070) in which Craig describes an electronic footswitch system. This would perfectly suit the experimenters needs, and a kit is even available for the project.

Drummers interested in expanding the capabilities of their "skins" have recently taken to drum triggered signal synthesis systems. For this type of application, the package described previously is still valid with the exception of the requirement of a VCO. For drum type sounds, the VCO is required to produce the drum "strike tone". Further, the two waveforms most likely to be used to approximate traditional drum sounds would be sine and triangle waves. These waves are nearly pure (meaning that they have little or no harmonic content) and should need no filtering, so we can exchange a 4720 VCO for the 4730 VCF mentioned in the package outlined previously.

The most requested electronic drum patch is the "bending" conga drum effect. To achieve this effect, a drum sensor is used to trigger the ADSR which has been preset for minimum attack, minimum decay, maximum sustain, and moderate release times. The variable ADSR output should be summed into the VCO control inputs along with a fixed bias voltage from the power supply. See figure 7. The bias supply is used to initially tune the VCO, while the ADSR variable output sets the "bend" range. The sine wave output of the VCO is fed to a VCA which is being controlled by the full-range output of the ADSR. The VCA output goes to the external amp.

To backtrack a few lines, we need some type of trigger input from the

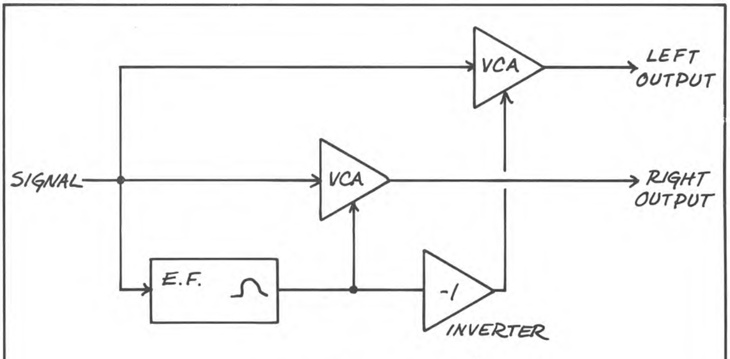


FIG. 5: AUTO-PANNING

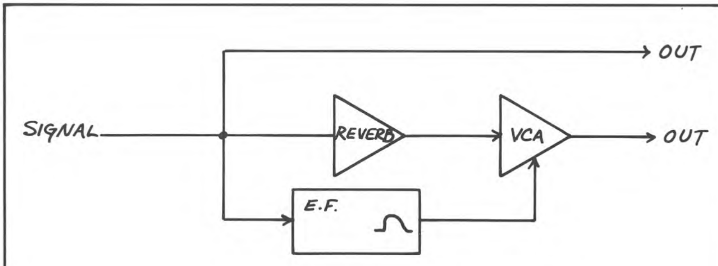


FIG. 6: ENVELOPE DRIVEN REVERB

drum sensors. There are several ways to get "solid-state" drum sensors which don't generate any sound of their own. However, for the initial experimentation of a drummer who already has a trap set, probably the best route to take would be to detect triggers from your existing drums. For our purposes, you could use an inexpensive contact mike, or a small mike like you can get to use with portable cassette recorders, etc. This can just be taped to make contact with the shell or bottom head of the drum which you wish to make a sensor. The mike can then be run through an envelope follower to generate one output proportional to the drum dynamics, and another straight trigger signal. In most cases, you will need to work with the sensitivity control on the envelope follower to make sure that the vibration caused by playing nearby drums will not falsely trigger the electronics array. If you have a particularly sensitive microphone, you may even wish to wrap it in a rag or place a thin piece of foam rubber between the mike and drum to make it less sensitive. With the sensor problem out of the way, you have a clear road to the latest in drum effects units.

On the commercial Syndrums, a switch is provided to cause the drum to bend up or down. In our system, we can use the inverter module to change the ADSR output from a falling VCO bend to a rising bend. The LFO can also be summed into the VCO to give your drum vibrato. The noise source can be processed through an envelope controlled VCA to add a "snare" effect to <u>any</u> of your drums. You could perhaps use the pulse waveform output of the VCO, with the drum triggered ADSR causing a sweep of the pulse-width modulation. This would give a raspy, phased drum effect. And the tricks with the reverb unit mentioned earlier are equally as useful on the drum system as on the processing system. There is a lot of power to be discovered in a modular drum synthesizer. As usual, if one drum sensor and module package is neat, an array of two, three, or more could be . . . . hazardous to your sanity? . . . . lethal? . . . . Well, at least interesting, wouldn't you say?! ■

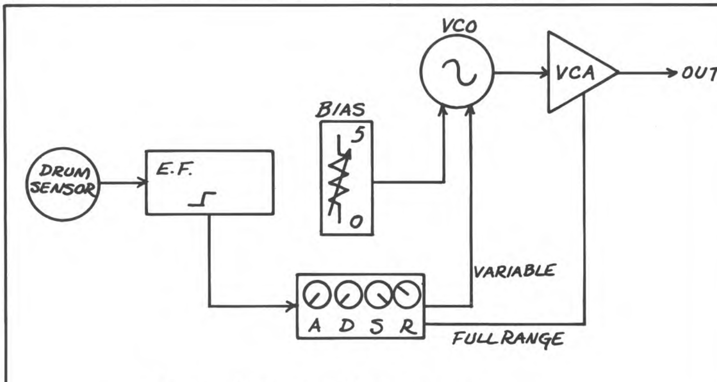


FIG. 7: "BENDING" CONGA DRUM

TO PHASE OR TO FLANGE — OR TO EACH HIS OWN

By
Gary
Bannister

The Age of IC's has been here a long time, and with each new chip a previous impossibility has become reality.

And so it is with the SAD 1024. This unit is of the Charge Coupled Device class, which also include solid state image sensors (a TV camera with no tubes - Remember Yule Brenner's eyes in "WestWorld"?). However, the exact workings of such devices is not important here. For those interested, a block diagram is included of the PAIA #1500 Phlanger. A more complete explanation may be found in the #1500 User's Manual.

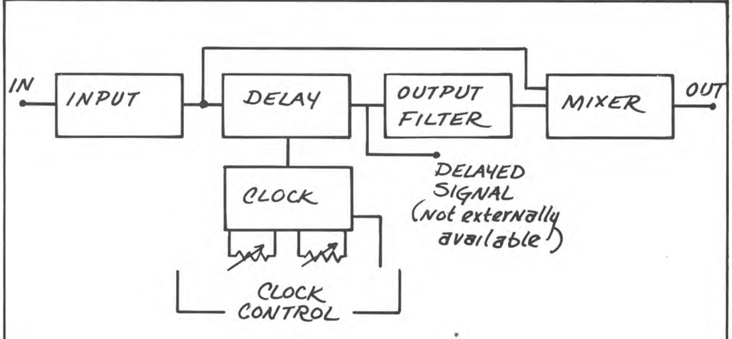


FIG. 1

What is important is the question, "Is this enough?" If you purchase a Phlanger (I prefer Flanger for spelling. Phlanger will be used to indicate the PAIA unit. - GB) You have everything you need to become another Isao Tomita right? Well, maybe. Let's see just what you get.

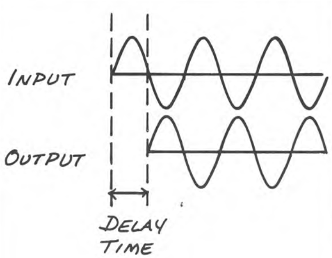


FIG. 2

The purpose of any flanger is to produce a true TIME delay of a signal (phase delay will be covered later). Ideally there should be no degrading of the waveform, although this is a necessity as PAIA so carefully explains in the Phlanger manual.

If the above rules are adhered to, we would have the following available. (See figure 2)

Due to the characteristics of the Phlanger these two signals are subtracted from each other. Depending on the amount of delay this produces various results. (See figure 3)

Note that with the proper delay a sine wave can be made to totally cancel or to boost to a higher output. These boost and cut actions occur at different frequencies over the audio range.

The amount of delay necessary to boost or cut a frequency is a function of the PERIOD of the frequency. The period is the amount of time necessary for the waveform to complete ONE CYCLE. This is found by dividing one second by the number of cycles per second. (See figure 4)

For any given time delay, the boost and cut action will produce a series of "peaks and valleys" which look like figure 5.

The actual frequencies at which

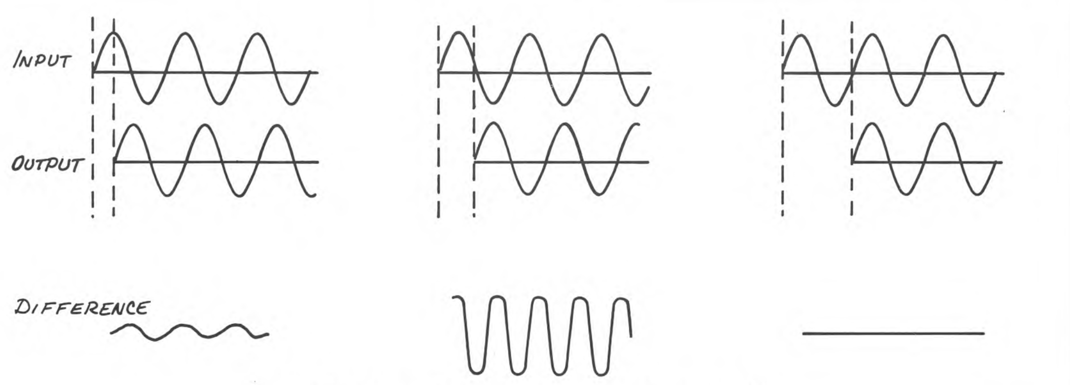


FIG. 3 DON'T FORGET, THESE WAVES SUBTRACT !!

IE: 1KHz PERIOD = $\frac{1 \text{ SEC.}}{1000 \text{ CPS}}$ = .001 SEC. = 1 MILLISECOND = 1 MSEC.

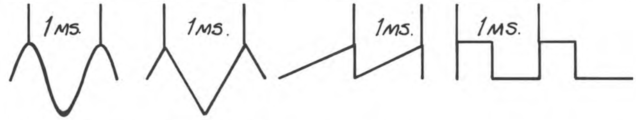


FIG. 4 ALL OF THESE ARE 1KHZ SIGNALS.

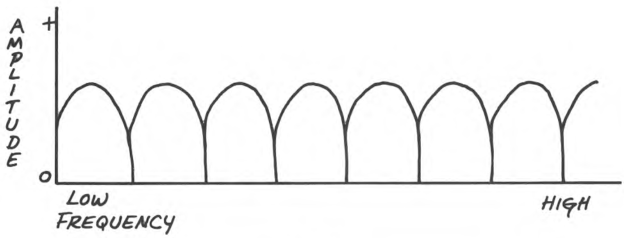


FIG. 5

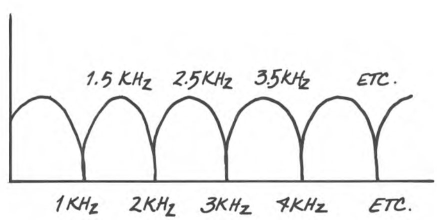


FIG. 6

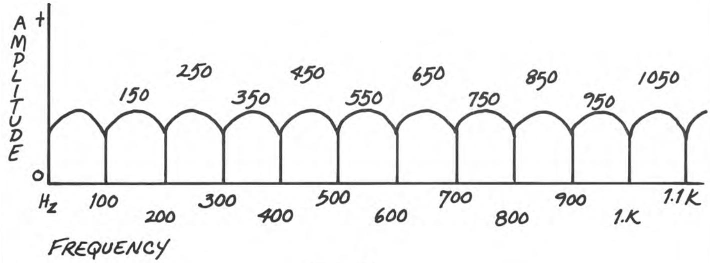


FIG. 7

these peaks and valleys occur is a function of the delay time. The delay time is, of course, dependent on the number of stages in the delay chip (512 per side of the SAD 1024) and the clock frequency (usually from 30KHz to 500Hz).

For specifics, let's introduce a hypothetical example:

> The SUPERDUPER QMOP 1000 delay chip (500 stages per side)
> A clock of 500KHz
> This will give us a delay of .001 sec or 1 msec.

(In practice, the exact formula for finding delay time with any of the popular delay line chips is: Td=N/(2Fc). In this equation, N represents the number of stages in the delay line, and Fc is the frequency of the clock. The halving factor in the denominator is present due to the fact that the 'charge packets' in the delay line are actually transferred on each clock transition, thus two stages are passed per clock cycle. In the example, the delay time would be .5 msec., making the notches shown below at 2KHz intervals. However, the overall frequency response pattern would be as shown. Marvin)

A delay time of 1 msec. corresponds to a frequency of 1KHz. This means that any frequency that is a multiple of 1KHz (1KHz, 2KHz, 3KHz, 4KHz, etc.) will receive a cut and frequencies between them (500Hz, 1.5Hz, 2.5KHz, etc.) will be boosted. (See figure 6) Note that these are all evenly spaced.

Now, if the delay time is increased by a factor of 10 to 10 msec. the response becomes: (figure 7). Note that there are now MANY more humps and dips, and that they are closer together.

From the above examples it can be seen that as the delay time is increased the number of peaks and valleys increases, and vice versa. The change in response is what gives the flangers their characteristic 'smooth' sound.

Now let us consider the true phaser. The difference is more than just semantics. While the flangers depend on TIME delay, the phaser depends on PHASE delay.

The actual effect of phase delay is quite similar to time delay. However, we must look at each in a different light. With a flanger we can make any given time delay that will affect ALL frequencies that pass through. With a phaser we can affect only one frequency with a given response. Restated, a flanger affects ALL frequencies

EQUALLY, while a phaser produces a given effect at only ONE frequency.

Figure 8 shows one simple phase-control circuit. This uses FETs and a variable potentiometer. The same thing can be done with modern op-amps. (See figure 9) This uses less parts to do the same as figure 8, but neither one is quite sufficient for our purposes. These two examples only produce 90° of phase shift (at only ONE frequency - remember), and for a true cancellation we need 180° of shift. So let's connect two such units in series. (Figure 10) This circuit gives us the following response. (See figure 11) The exact frequency at which the notch occurs is a function of the values of the fixed resistors and the capacitors, and the setting of the two pots.

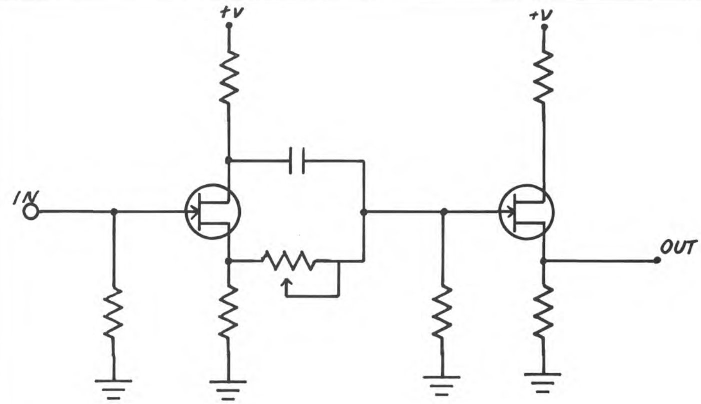
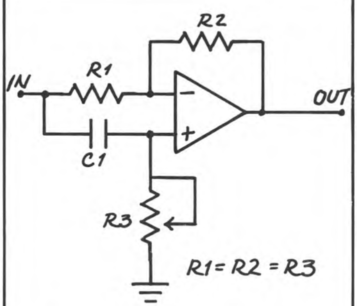
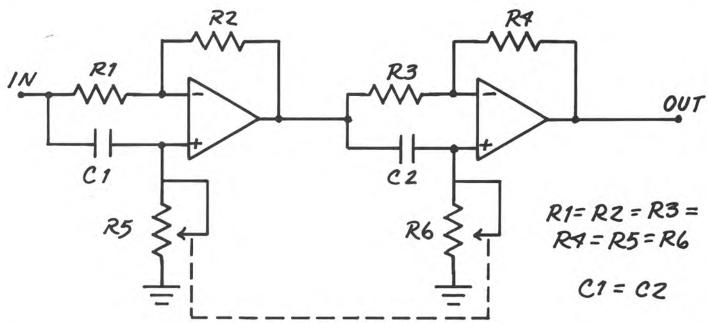


Fig. 8



POWER SUPPLY OMITTED FOR CLARITY

FIG. 9



THE DOTTED LINE MEANS THAT BOTH POTS ARE CONNECTED TO THE SAME KNOB.

FIG. 10

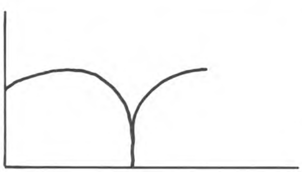


Fig. 11

It would be rather difficult to turn these pots by hand fast enough all the time (as it is with all synthesizer modules). Also, if we use more than two of these phase shift stages, multiple pots are not practical at all. So let's use FETs as voltage controlled resistors. (See figure 12)

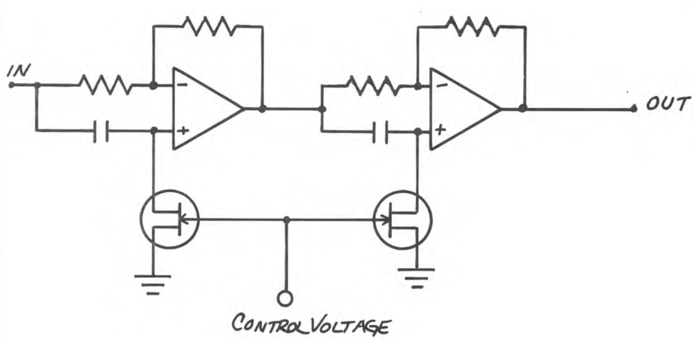


FIG. 12

Actually, even two of these stages are not enough. The minimum to produce a one-hump-two-notch response is four. Indeed, this is what is in

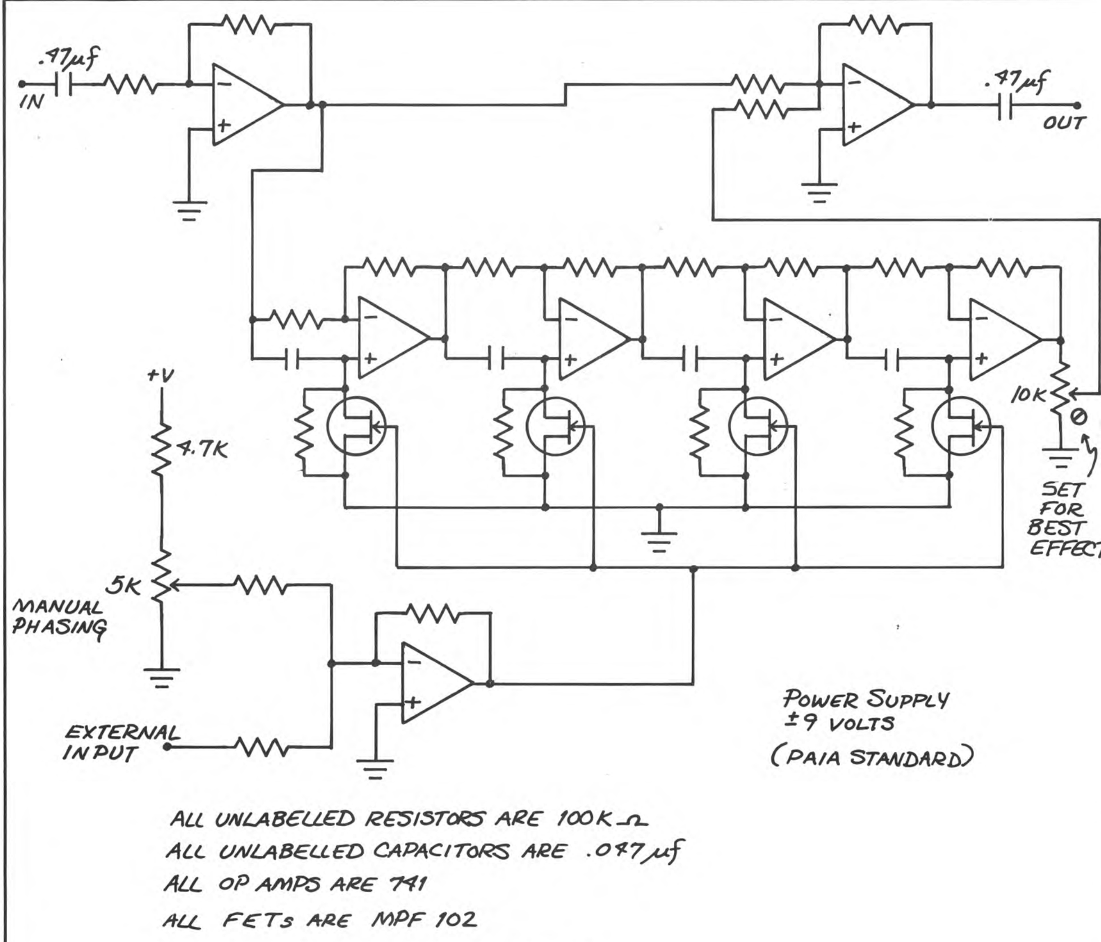


FIG. 13

many of the popular 'pedal phasers'. Many larger units use 6, 8, even up to 12 such stages. In general, two stages produce a notch, and three are necessary for a hump.

Taking a quantum jump from theory to design, figure 13 is a workable schematic of a 4-stage phase shifter. This works as is, but lacks an internal sweep, resonance controls, and variable mix. It may also need some part value optimization for best operation. In addition, this basic unit may be easily expanded to 6, 8, or even 12 stages.

Simple, huh? There is nothing really critical about the operation. Assembly is easy and can be done on perfboard, wire wrap, or PC. Some sources claim that the FETs must be matched, but actual operation proves this to be necessary in only the most critical applications. Well regulated supplies are not necessary, either, but some extra filtering on the board won't hurt.

Before we look close, let's look at the response that is produced by this phaser. (Figure 14) Figure 14A is the response with no control voltage input. Note that there is one hump and two notches. For comparison we will assume that the first notch occurs at 1KHz, although it may be anywhere due to slightly different component values. In figure 14B we have applied 5 volts of control. Note that the response is now shifted DOWN, but otherwise remains the same.

continued on page 29...

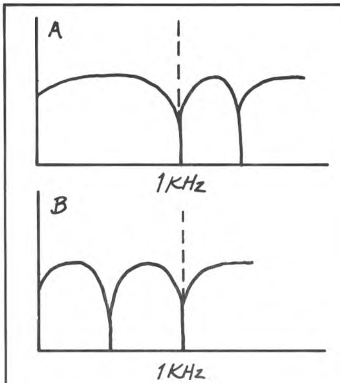


Fig. 14

augmenting the 3750's memory

by Steve Wood

When it comes to electronic drum machines, the PAIA 3750 is pretty unique. The reason is obvious. Aside from the almost irresistable touch pad controls, it's programmable! This one feature gives the user a definite advantage over working with conventional type drum machines in that you can make up, and enter into the machine's memory, your own rhythmic patterns using whatever combination of the available drum sounds you like. Further, when it's time to go to the bridge of the song, you have the option of letting the main score continue to play, or causing the 3750 to "bridge" so that it will play what you have previously programmed there. Both main score and bridge can be set up to 64 events long, and there are two of each. That's a total of 256 events, which translates to 256 bytes of memory in the machine. When in use, some of these will contain "rest" commands (no drum sound). For most practical applications, that's plenty, but there are always those of us who can't leave well enough alone. Besides, there are some very worthwhile advantages to be gained for very little expense and a couple hours work.

Suppose there were 512 events, room for 4 scores and 4 bridges. And suppose that two scores, or two bridges could be played back at the same time. Now you could not only get twice as much music in the machine, but you could also do multi-tracking! But the only way to have more than the original 256 events available in the 3750 is to add more memory, right? Sure!

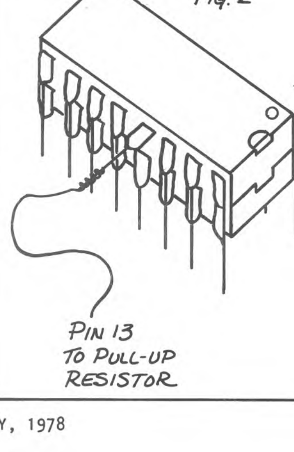
Ordinarily this might be a bit of a problem, but we're in luck here, because we're dealing with tri-state chips. Standard logic RAM would be capable of assuming one of two conditions, - "read" where data is taken from the memory, or "write" where new data is loaded into the memory. In either case, the data appearing on the lines connected to the memory will affect, or be affected, by the memory. Tri-state devices can assume a third condition in which the data lines "float". They do not specify data, and they may be manipulated without affecting the contents of the memory. This makes it possible for more than one IC chip to occupy the same location. With some very careful soldering techniques, you can cascade the 2112 type memories so that two ICs occupy each memory chip location (ICs 14 & 15). Then the chip enable line can be routed to either of the two pair of ICs via a SPDT switch. The schematic of the completed modification is shown in figure 1.

CASCADING THE MOS CHIPS

No doubt there are many ideas on safely handling MOS devices, and some will prefer their own time tested techniques, but allow me to briefly outline how I went about it.

A piece of aluminum foil folded once or twice makes an effective pin shorting element and an excellent heatsink at the same time. Remove IC 14 from its socket on the 3750 PC board. Stab the pins carefully through the sheets of foil. Then position one of the two new chips (2112) on top of IC 14, piggy-back style (See figure 2). Check that the notches on both packages are at the same end, and the pins of the new chip, (A) slide snugly over those of IC 14. Pin 13 WILL NOT BE SOLDERED! All other pins will be soldered; pin 1 of IC A to pin 1 of IC 14, — pin 2 of IC A to pin 2 of IC 14, and so on, with the exception of pin 13. This is the enable pin, or in this case "not chip enable" ($\overline{ce}$). This pin on IC A should be bent outward slightly so as not to make contact with pin 13 of IC 14. Remove the foil shorting material. At this point IC 14, with the new 2112 on

FIG. 2



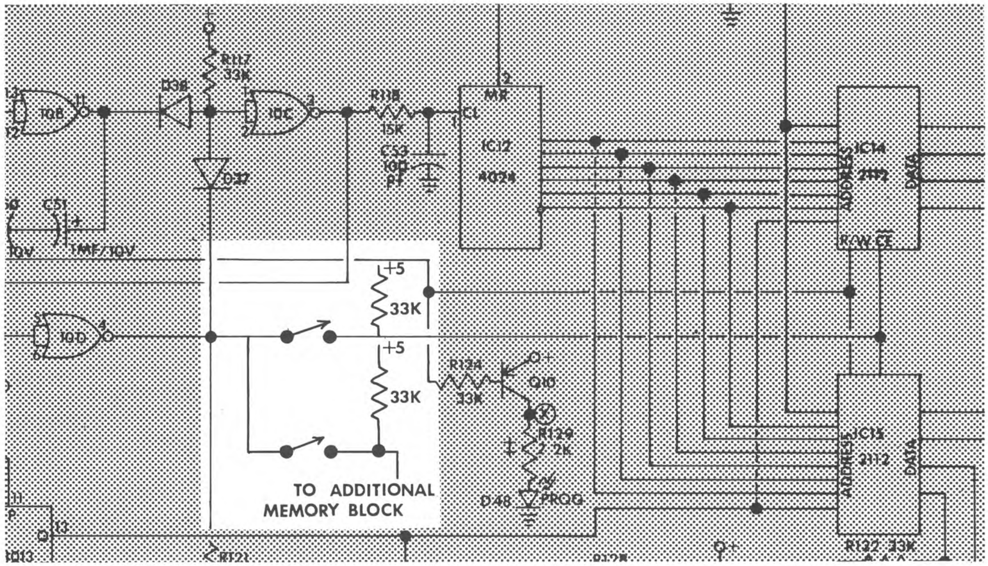


FIGURE 1

its back, can be reinstalled in its socket, and the piggy-backing process repeated with IC 15 and the other new chip.

PULL-UPS AND MEMORY SELECT SWITCH

As we said, pin 13 is the "not chip enable" pin, so the memory can function normally as long as this pin is held at ground. If pin 13 is held high, the memory cannot be written into or read from at any address, and the data lines float. To insure that the $\overline{ce}$ lines of either block of memory stay high during the period in which that block is not being used, we will install 33K pull-up resistors between the positive 6 volt supply and pin 13 of ICs 14 & 15, and between supply and pin 13 of the new memory chips. Positive supply can be tapped at TS 1 lug 2. One end of each 33K resistor should be soldered to this lug, leaving the other end of each resistor suspended free. A 5" length of light guage wire is connected to the free end of one of the resistors. The other end of this wire connects to the solder pad or foil trace that leads to pin 13 of IC 14 & IC 15.

The second 33K is installed between positive supply and pin 13 of ICs 14A and 15A. These two chips should have their #13 pin sticking out to the side of the package making no connection. A 6 inch length of light guage wire can be used to connect pin 13 of IC 14A to pin 13 of IC 15A, and then to the free end of the second 33K resistor. The line that normally drives the $\overline{ce}$ pins of the original IC 14 and IC 15 originates at pin 4 of IC 10. This line connects to the memory ICs by way of a wire jumper which is directly above R124 on the PC board. Remove this jumper and connect one 12" length of light guage wire to each of the two solder pads where the jumper had been connected before. Now, locate the free end of the wire which you connected to the pad that connects by copper trace to pin 4 of IC 10. This will attach to the common lug (wiper) of a SPDT switch. Each of the remaining two switch lugs should now be wired to the junction of one of the pull-up resistors and the wires leading to the "$\overline{ce}$" lines of the two memory banks.

Operation of the unit will be the same as before, with one simple difference. When you've programmed 4 pages of memory (both scores and both bridges), just change the position of the new switch and PRESTO! - another 4 pages to fill! This should prove particularly useful to you "one man bands", because you can go longer before reprogramming.

If you would like to take this concept a step further, you can do multi-tracking. If the SPDT switch discussed above were replaced with two SPST switches, each memory bank could be selected individually OR in unison. Now, with the score switch in the "1" position, and the new switches selecting "Bank 1" of memory, one score or bridge could be programmed. With the score switch still in the "1" position, and the new switches altered to select "Bank B", the first score in the second block of memory could be programmed to jive with the first score entered in "Bank 1". Now, because the first score in either bank of memory occupy the same address, (the same is true of the second score and the bridges), we can play both scores back at the same time by placing the new switches in the Simultaneous Enable positions and touching play.

While it's not impossible to have more than one drum sound occupy a single event with a stock 3750, it is at very best a tedious task, and not very practical in terms of real world musical applications. This modification makes it easy.

There may be one little problem here, caused by this memory expansion modification. The extra memory will increase the load on the power supply by a substantial amount. Some units may not run on batteries after this modification. In any case, the battery life will be considerably shorter. A simple regulated power supply, as shown in figure 3 will quickly solve the battery problem once and for all.

If the transformer used here has a sufficient current rating, say 1 amp or so, I see no reason why one could not elaborate on the memory expansion farther. Perhaps even to the point of adding a whole bunch of outboard memory (provided he is willing to sacrifice some of the portability of the 3750, and willing to do the point to point wiring that would be involved). You wouldn't have to stack the chips in a piggyback, but you would need some way to decode the addressing. A possible decoding scheme might look something like figure 4. The address lines would connect to the outboard memory ICs in the same way they connect to IC 14 and IC 15. In fact, all the pins of the expansion chips will connect in parallel to the corresponding pins of IC 14 and IC 15, with the exception, once again, of pin 13. We can set up 7 of these blocks external to the 3750, 14 memory chips in all. In each block, the #13 pin of each of the two chips of that block will

continued on page 23...

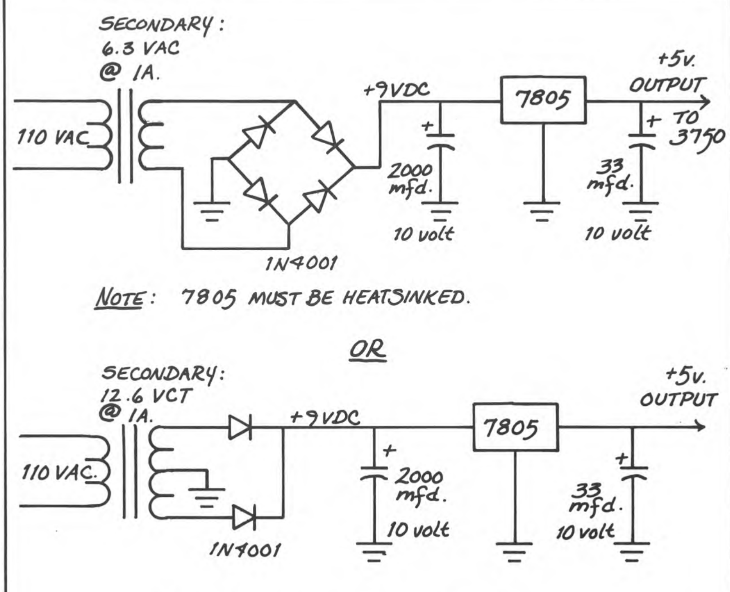
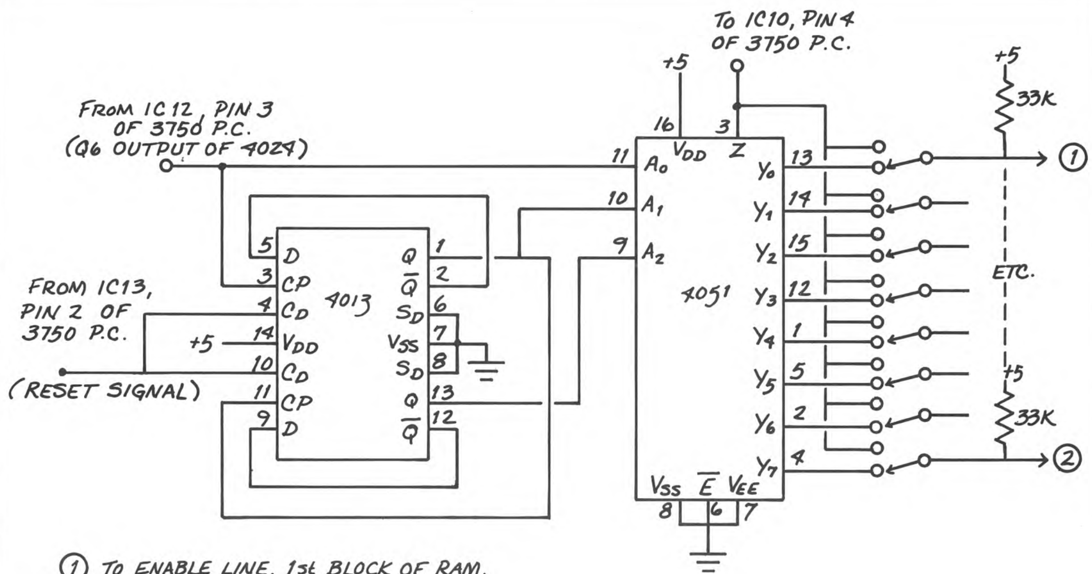


FIG. 3: AC POWER SUPPLY



① TO ENABLE LINE, 1st BLOCK OF RAM, PIN 13 OF IC14 + IC15, 3750 P.C.

② TO ENABLE LINE OF LAST BLOCK OF RAM.

FIG. 4: OUTBOARD MEMORY DECODING

EXPANDING THE PATCHABILITY OF THE MINI-MOOG

By Marvin Jones

Anyone who works with electronic music in any way can't help but respect the Mini-Moog. Robert Moog and his concept of modular voltage controlled sound generation elements really threw the musical world for a loop. When acceptance of these "far out" gadgets came to the popular music market, it was primarily because of the development of the Mini-Moog. Finally, the progressive musician could have one of these goodies without having to worry about patch cords or large boxes of separate modules. No wonder it was so popular. But, as with public utilization of any form of new technology, it was hard to envision what consumers would want to do with a unit like this in the distant future. Well, here we are in the future, a great number of musicians bought Mini-Moogs during their initial appearance on the market, and now those musicians are well acquainted with the Mini and want to expand. That's where some of the problems begin to occur.

The rear of the Mini has jacks to provide for external control voltage for the VCO bank, VCF, and for the VCA. When you get right down to business, these are the only three expansion jacks which are really compatible with most other gear. Also available on the rear panel is a Moog-style "S-trigger", which requires that one contact be shorted to the other (either via a transistor switch, or mechanical switch or relay). This turned out to be a rather non-standard feature, as most synthesizers went with the step-type gate trigger which features a voltage output to indicate a trigger signal. Additionally, the S-trigger is accessible only through a two prong Cinch-Jones type plug. This definitely precludes use of most standard audio patch cords. The remaining jacks on the rear panel allow external signal input, Mini-Moog output, and two power jacks to provide power for outboard devices. This is actually a rather limited array of expansion jacks by today's standards. And, in fact, this is one of the most common questions from Mini-Moog owners who wish to expand their systems; "How can I hook it up to do this?"

There are a great number of patch points in the circuitry where jacks could be added to facilitate future expansion or "re-patching" of the Mini's normalization scheme. For this article, we will discuss two of the most needed and most useful expansion jacks: Keyboard Control Voltage Output, and Standard Gate Trigger Input/Output. We'll discuss the other modifications in later issues.

After adding these two jacks, you will be able to use the Mini to trigger and control additional external banks of modules such as more VCOs, or triggered effects such as single-sweep flanging or single-pass sequencing. Also, you will be able to interface a sequencer with the Mini to operate as a free running sequencer system. Because of the use of the "S-trigger", the best you could do before was to have the sequencer control voltage output feed the VCO and trigger the effect by playing a note on the keyboard. Now you will be able to have the step trigger output of the sequencer trigger the Mini by itself! The only time you need to touch the keyboard is to change keys (if desired). It's always handy to have a machine that will run itself! The Gate input jack will also be helpful in interfacing external instruments or signals. As was mentioned, the Mini already has an external signal input. Now you can use an envelope follower which has a trigger output to trigger the Mini from the guitar, OZ organ, Bionic Trumpet, or whatever.

For you Mini-Moog owners who have never attempted any of your own

Figure 1 Figure 2

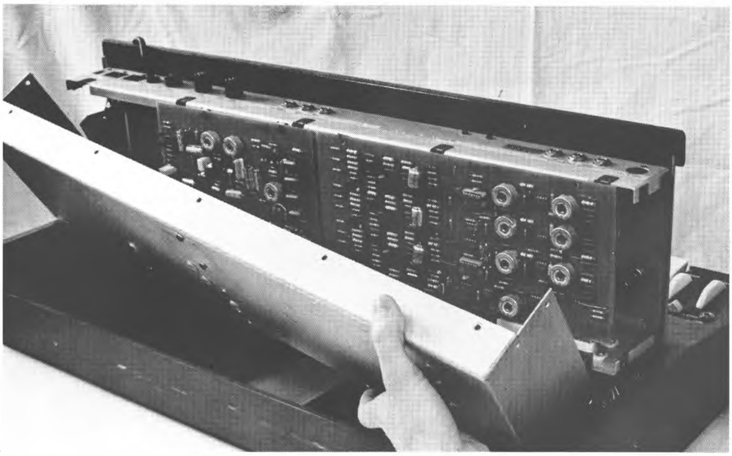
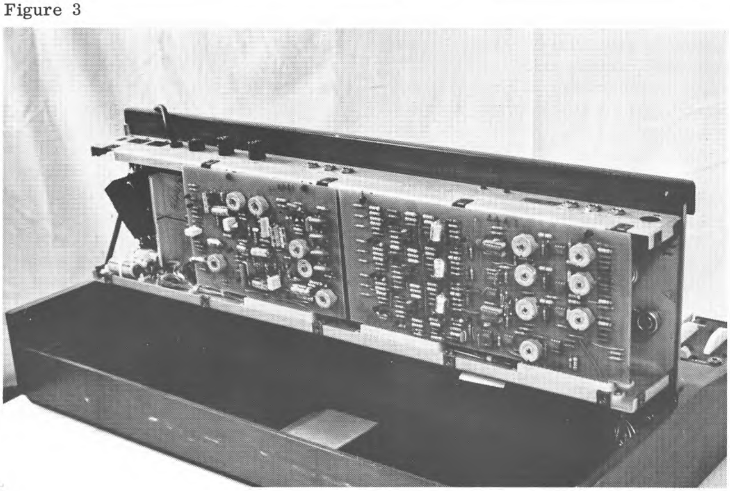


Figure 3



repairs or modifications, we have attempted to make the modification process as painless as possible with lots of pictures and step-by-step instructions. NOTE that the original manufacturers warranty will probably be voided, unless you have a lenient or understanding local service center.

With all the warnings out of the way - - Here we go!

1) Remove the rear panel. There are 18 Phillips head screws holding the rear panel in place: 5 along the top, 5 along the bottom, and 4 on each end. See figures 1 and 2. Once the cover is removed, you will not be able to use the Mini's device which props up the electronics box, so it would probably be helpful at this point to look around your workbench for something to set under the chassis to prop it up in as vertical a position as possible.

2) With the cover removed, note that there are two circuit boards visible at the rear plane of the chassis. The circuit board at the far right (when looking at the rear of the unit) is board #1 and contains the oscillator bank. There are two screws which mount the top edge of this board to the top panel of the chassis. Remove these screws, making sure you don't lose the fiber washers which act as standoffs between the circuit board and chassis. See figure 3. Once the screws are removed, use a wide blade screwdriver to VERY CAREFULLY pry the circuit board from its two sockets. The sockets are very tight, so you will need the screwdriver to wedge between the socket and the edge of the circuit board. Twist the screwdriver, and move to the other end of the board repeating the procedure until the board is far enough out of the socket to pull it free. See figure 4.

3) With the oscillator board removed, you will see board #2 which contains the envelope generators and keyboard output circuitry. Like the VCO board, board #2 has two screws holding the top edge in place. Remove the two screws, and use the prying screwdriver technique to remove this circuit board. See figure 5. With this board removed, half of the front panel and controls should be visible from behind.

4) On the top panel of the chassis, mark the two points where holes will be drilled to mount the expansion jacks. Both holes will be on a line which runs through the center of all the other jacks. One of the holes should be located 11 inches (27.94 cm) from the

right edge of the case. (The right edge should be the one nearest the S-trigger jack) The second hole should be at a point 12 inches (30.48 cm) from the right end of the chassis. When drilling these holes, it is advisable to use a small drill to start a pilot hole (about 1/8 inch), followed by an intermediate drill (1/4"), and finally the 3/8" drill for the final hole. ALSO, in an attempt to keep the drill from hitting any of the internal circuitry, be sure to support the body of the drill as it finally cuts through the panel. Metal shavings from the drilling are difficult to remove from the circuit board sockets, so it would also be advisable to cover as much of the circuitry as possible with a rag. After drilling, inspect the sockets and circuitry for any shavings which may cause shorts or problems. See figures 6 and 7.

5) Install the two 1/4" open circuit phone jacks such that the connection lugs of the jacks are facing the rear opening of the chassis. See figure 8. Note that the rightmost jack should be turned slightly to avoid any interaction with the nearby potentiometer. After mounting, the jacks will need to be labelled on the outside of the chassis in some manner. I used dry transfer lettering covered with a layer of clear vinyl tape, such as Scotch Magic Transparent tape. The leftmost jack should be labelled "Gate In/Out", or designated with a gate symbol, _⌐. The right jack should be labelled "KBD CV Out" or similar.

6) Using a 2 inch (5 cm) length of insulated wire, connect the ground lug of each jack. Solder at the right jack only. Using a 2 inch (5 cm) length of wire, connect the ground lug of the left jack to the bare wire ground buss which runs along the three adjacent output/input jacks. Solder both ends of this wire.

7) Select the #2 circuit board which was previously removed. We will now attach two wires to the foil side of this board which will later connect to the jacks we just installed. Lay the circuit board on your workspace such that you are looking at the foil side of the board, and the "fingers" of the edge connector are facing you. The left set of contacts is designated Section A, and should consist of 14 contacts. Look closely at the etched copper designations near the "fingers", and you will see that the left end is contact #1A, while the right end is designated #14A. The larger set of contacts which is near the center of

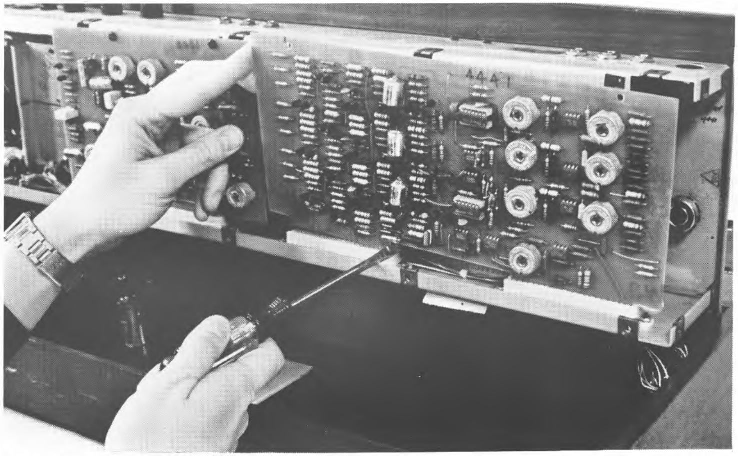


Figure 4

Figure 5

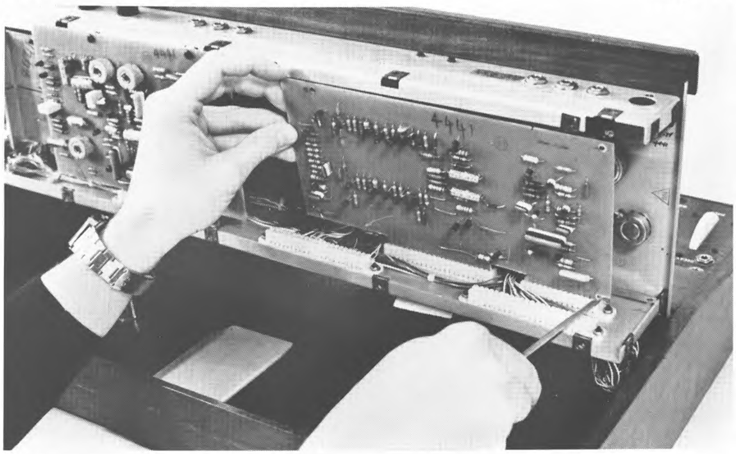


Figure 6

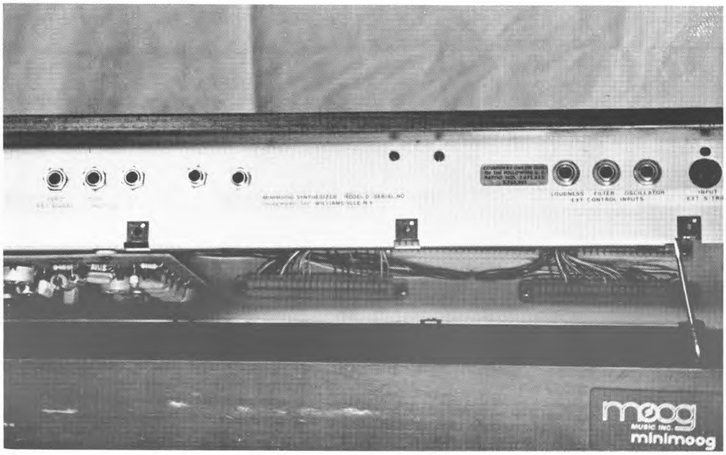
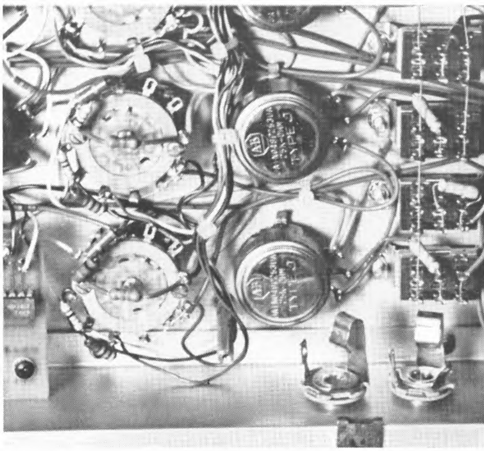


Figure 7

Figure 8



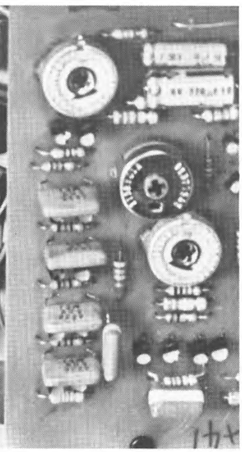
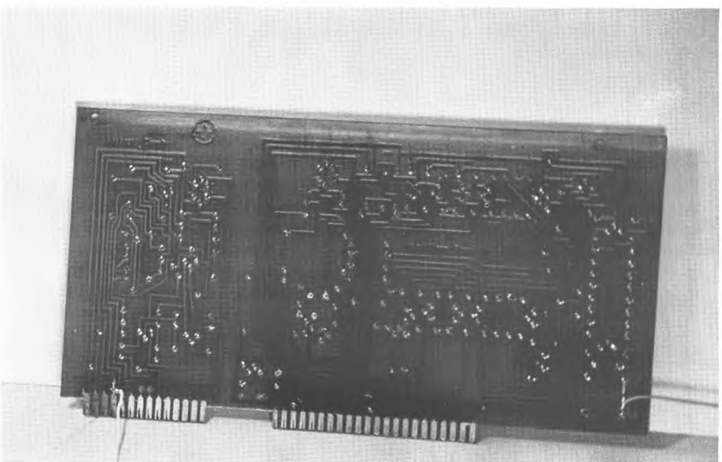


Figure 9



the board is the B socket section. This section consists of 22 contacts, once again designated on the circuit board from left to right as #1B through #22B. See figure 9.

In the left "A" connector, locate fingers #4 and #5. These two should be connected by a trace directly above the fingers. Using a screwdriver or knife, carefully scrape away the green anti-oxidant coating on this trace. Tin the trace by flowing a small amount of solder on the now exposed trace. Cut a 15 inch (38 cm) length of insulated hookup wire, and prepare by stripping 1/4" (7mm) of insulation from each end. "Tin" the exposed ends of the wire by flowing a small amount of solder into the strands of the wire. Lay one of the prepared ends of this wire on the pretinned trace above fingers #4 and 5. Heat this connection until solder flows around the wire, remove the iron and let the connection cool. See figure 9.

In a similar manner, cut and prepare a 7 inch (18 cm) length of insulated wire. Locate finger #20 of the Section B connector and trace the copper foil pattern which leads from this finger. The first component (and the only one) to which this path leads is the anode of a 1N34A signal diode. At the other end of this diode (the banded cathode) there should be a short trace which leads to a 100K resistor (color coded brown-black-yellow). On this short trace, once again scrape away a section of the green coating and 'tin' the exposed section of the copper. Lay one of the prepared ends of the 7 inch (18 cm) length of wire against this area and heat the joint to allow the solder to flow smoothly. Remove the iron and allow to cool. See figure 9.

8) The wire connected to finger 4/5A is the keyboard control voltage output. Connect the prepared free end of this wire to "hot" lug of the rightmost jack on the top panel of the Mini. This jack should be the one labelled "Kbd CV Out" or similar. Solder the connection at the jack lug.

9) The remaining wire originating at the diode/resistor junction is to be connected to the remaining (Gate Input/Output) jack which you installed. Solder this connection.

10) This completes the installation of the modification. Reassemble the Mini-Moog by following the disassembly steps in reverse. Replace board #2. Replace board #1. DON'T FORGET the screws and fiber washers when reinstalling these boards. Finally, replace the rear chassis cover, how-

Figure 10. Sequencing is but one of the possibilities for the expanded Mini-Moog.

ever, you may want to play with the expansion jacks for a while to make sure all is well before closing the case.

As mentioned earlier, this simple inexpensive modification can open up quite a few possibilities for the Mini owner. Any of the many standard controllers available will now easily interface with the Mini. Envelope followers can be used to trigger the Moog from guitars, mikes, prerecorded signals, click tracks, or any other audio signal. Joysticks could be used to feed control voltage to two of the three control inputs available, and (if you should perhaps choose to use the Paia joystick) the joystick trigger output could also be used via the new trigger jack. Using an instrument like the OZ mini-organ through the Mini-Moog external input can produce a very powerful polytonic instrument. Previously, you would have had no easy way to interface the trigger from the OZ. But now, ... it's just a patchcord away! Sequencers are always fun, and with a Mini-Moog to control you can get some dynamite effects. Using the added trigger jack, you can now have the sequencer trigger the Mini for each note of the sequence ... the way the sequencer was designed to be used. Alternatively, you could use the sequencer as a single pass effect - perhaps for high speed arpeggiation. To get this effect, the sequencer is set to produce a non-repeating pattern with the new Mini gate jack used as an output from the keyboard to trigger the initiation of the sequencer pattern. Now, whenever a note is played on the keyboard, the sequencer will cycle through the pattern once and return to the note being played on the keyboard. And remember that with the sequencer being used to control the exponential VCOs in the Mini-Moog, the sequencer pattern will transpose from key to key, as different notes are played on the keyboard. Using the trigger jack as an output, you can also trigger external envelope generators to be used to sweep an external flanger, filter, or even perhaps to be fed back into the Mini-Moog VCF control input to generate a more complex envelope pattern n could be done with the original SR built into the instrument.

The keyboard output jack which has been added is a source of 1 volt per octave control voltage governed by the keyboard. This voltage can be used to drive external banks of exponential type VCOs and VCFs. For the advanced experimenter, this output could easily be quantized, stored, and processed digitally for memory sequencing, control voltage delay or cascading, and many other special effects. This jack will turn out to be very handy as you will see in future articles.

Go Moogsters .... play with your newest toy.

For more information on the stock Mini-Moog, contact:

Norlin Music
7373 N. Cicero Ave.
Lincolnwood, IL 60646

THE AUGMENTED 3750 CONTINUED

(from page 18)

be tied together, pulled up, and fed to the automatic "Bank select" circuitry. This is where the decoding comes in.

Let's back up and take a look at the counter in the 3750. Only 6 bits from the counter are used to address the memory. That leaves us with the most significant bit unused, until now. We'll use this bit to tell us when the counter has cycled once through one complete 64 event score. This 7th bit will become address line A0 to a 4051 multiplexer chip, and also the input to a 4013 D type flip-flop whose output will serve as line A1 of the 4051. This line also connects to the input of a second flip-flop whose output is A2 of the 4051. Each of the eight output lines (Y0 through Y7) of the multiplexer chip will connect to the Z pin, one after the other as the 4024 counter cycles over and over again. This is, of course, assuming that there are no repeat instructions programmed into the first score. The Z pin of the 4051 is connected to the enable driver IC 10, pin 4 on the 3750 PC board. The enable line of each block of outboard RAM will connect to one of the pins Y1 through Y7 of the 4051, or to the enable driver directly as determined by a SPDT switch in each line. The #13 pins of IC 14 and IC 15, on the 3750 PC board are connected to a SPDT switch which can select between the enable driver and Y0 of the 4051. This way, we can play each score in each block sequentially, or several scores at once for complex multi-track percussion backgrounds.

Pretty neat, huh? That's 2K bytes of memory if you do it this way - 2,048 events and brother, if that ain't enough, it's computer time!

A Digitally Addressed Transposer for your Analog Keyboard or Sequencer

By Larry Pryor

Have you ever been playing your keyboard, or using the sequencer and wished you could change keys at will, any time you wanted, and do it knowing it will be exactly where you want it? From the key of 'C' to the key of 'G' just like that. Well, I have had need of such a black box. About a year ago, I started building up some circuits to do this but did not know if anyone else had need of it so I quit beating my brains out and put it on the shelf.

About a year later Paia comes up with the digital keyboard, and my breadboard comes off the shelf. I now have a black box that not only transposes in any one of thirteen keys, but has a 64 note memory that can be programmed and called for at will.

There are going to be a lot of people out there who will buy digital keyboards and later put it under computer control. Transposing and other goodies will be easy to do. But there are many people out there who are happy with their present keyboards, or may not be able to afford a computer. All I wanted to do is transpose and I didn't want a computer. So I thought I would pass this information along to anyone who would be interested.

I will show in the following paragraphs how this circuit was developed. Figure #1 shows a basic block I started with.

About eight IC's and all I had was a circuit that, when you pushed a button, held it in the latch and addressed a 4 to 16 line IC, which in turn selected a pot through one switch of the 4016's.

The thirteen push button keyboard is encoded to a BCD code. It then goes to a 4 bit latch. The BCD code is presented to a BCD addressable 4 to 16 line decoder which addresses one of the 4016 switches.

To reduce the IC count, I replaced the 4-4016's with a single CMOS analog multiplexer CD4067. But, go out and try to find or buy one of those CD4607s. You can't find them. Well, if you can't find a 4067, and you don't want to use 4 4016s, then what? Well, how about those 4051s? Gee, where have I seen that chip before? A 4051 is an 8 channel multiplexer. For its operation details, read April, '76 Polyphony. That will work real well, and you don't need a 4 to 16 line decoder because the 4051 is BCD addressable. So now the circuit looks like Fig. 2.

Now we are down to four ICs but still we push one button, get one note change. That's OK, but can we go one step more? Yes. What if we could find an IC that would store more than one note at a time? Well, there is one I ran across, it's called a FIFO! A what? A FIFO. That is short for First In First Out. Again I say, what is that? A 3341 is a 64 word x 4 bit memory that works in a First In First Out mode. The input and output are completely independent or asynchronous. You can put in 1 word, 10 words, 40 words or all 64 words and walk away and clock them out tomorrow if you want at any speed you want. That's really neat. Well now, that changes our block diagram to look like Fig 3.

However, you have a problem in that with 64 words to move around, you have housekeeping problems. Well, it's really not a problem on the 3341. You have data input, shift in, data output, and shift out commands. So what does that mean? Any time you push a button on the data keyboard, you enter data on the input pins and at the same time you enter a "shift in" command. You did the same with the latch as in fig 1 and 2, but how do you get it out? Very easy. Clock it out with a clock on the "shift out" pin. For every clock pulse in, out comes the data, one at a time, and they come out in the order you put them in - FIFO. Really neat.

That means you enter a key change, and clock it out. "Clock it out with what?" you say. Well, how about a sequencer? Look at the patch in Fig 4 and think how it might sound if I punch in thirteen keys on the data keyboard and clock them out. (Note: I have not mentioned that each resistor on the outputs of the 4051 are in series with your Control Voltage that goes with the VCO. Remember, when you put a pot in a series with the control voltage line, it lowers the frequency of the VCO. So, this device switches pre-set pots tuned to half-steps in and out of the circuit.)

Step 1. Set the sequencer for a chromatic progression from C to C ... CDEFGABC.

2. Enter data on transposer by pushing buttons 1-14. This will instruct the transposer to sequentially step up by one half-step until the passage has shifted up one octave.

3. Connect trigger input of transposer to the stage #1 gate on the sequencer.

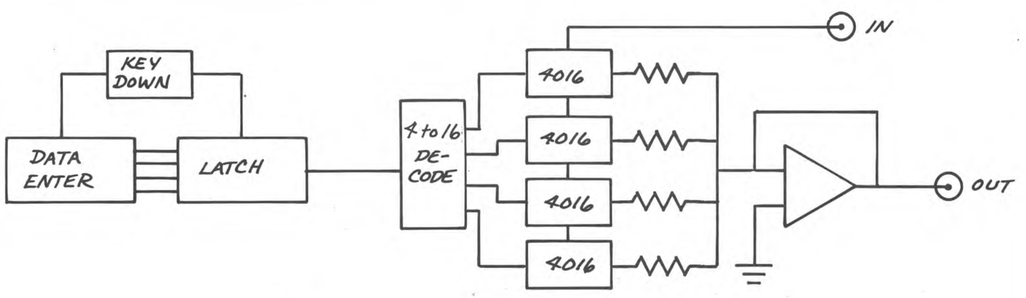


FIG. 1

4. CV out of sequencer to input of transposer, out of transposer to VCO.
5. Hit run on sequencer.
6. Sequencer should go from 1-8, then trigger transposer, repeat 1-8, and so on, until the sequencer output raises one octave and shuts off.

Figured out how it would sound? Well, if it was patched correctly, the sequencer would produce a progression of notes that would play a whole scale - starting on C, D, E, F, up to C, then get a trigger pulse. This shifts out the next data change BCD code - which changes resistors, which changes keys to the next half-step. The sequencer repeats 1-8 again and so on. So your 8 step seq. is now 8 x 13 or 104 steps long! WOW.

Besides the thirteen keys on the data enter keyboard, I have put on one more - #14. When this is entered, it outputs a slight negative voltage. This can be labeled "No output", "Pause", or whatever. This is necessary because after the FIFO has emptied its data, the last bit remains on the output pins. So, if the sequencer is still running, it will output the same thing over and over again. So, at the end of your data entered, if you want no output from the VCOs, just push #14 and when the sequencer is done playing your tune, it will flip a negative voltage to the VCOs and cut them off. I find it works well, with no pop or anything. There is also a reset button, so if you enter data from the keyboard and make a mistake, push reset and start over.

When all 64 words are filled, there is an input ready pin that goes high. This is used to turn on an LED. as a visual indication that all 64 words are filled.

There is one more feature I put on. This transposer can of course be plugged into a keyboard, and you can transpose while you are playing. Yes, I know you have a pitch control on the keyboard and VCO, but what if you had two tracking VCOs and wanted to change one while you were playing. Easy. Plug one VCO directly to the keyboard CV and the other VCO into the transposer. Push key #13 which is straight through, and tune both VCOs to the same pitch. Now, one problem. How do you shift out data? You can enter it in from the data keyboard, but how do you get it out? Well, built on the board is a fast read out clock - just a simple 555. With this running, it will shift out data faster than you

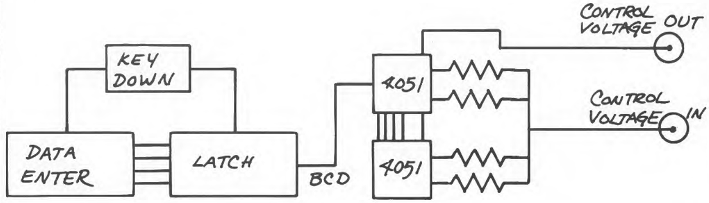


FIG. 2

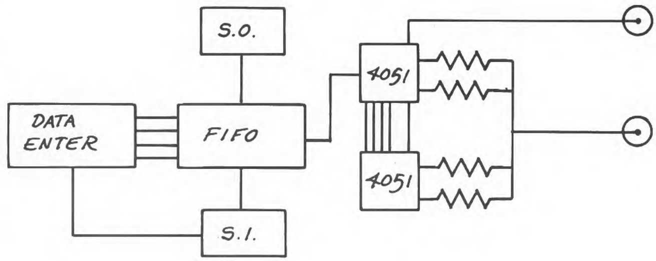


FIG. 3

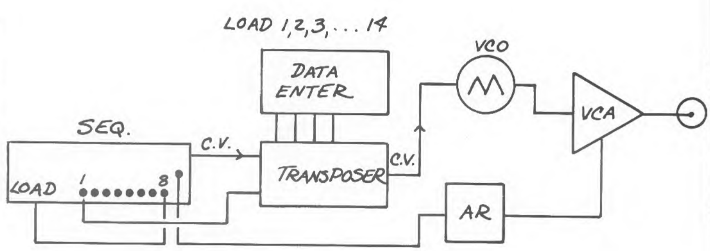


FIG. 4

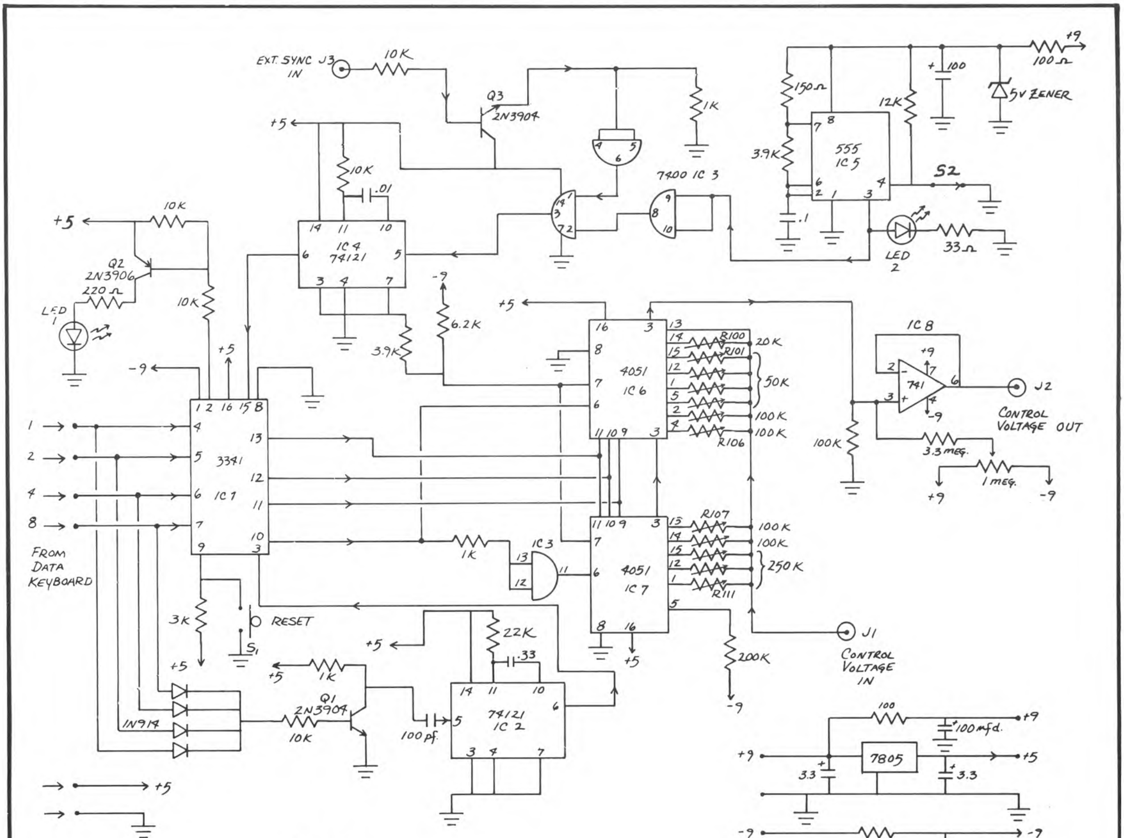


FIG. 5A

could ever push data in. So as you are playing, you can hit a new key and transpose while you play. Fig. 5 shows the complete schematic.

CIRCUIT DESCRIPTION:

Diodes 1-25 are used to encode the data keyboard into a BCD code for the 3341. R1-R4 are pulldown resistors for the 3341's TTL input. R5 and C1 are used to put a pulse on the desired input pins of the 3341 whenever a key is pushed down. But remember, we have to have a "shift in" command. By "OR"ing the BCD pins through four more diodes, and conditioning them through Q1 and IC2, we get a shift in pulse of about 8 milliseconds when any key is pushed down. (The keyboard and diodes can be built as a separate module and a 6 conductor wire brought out from the keyboard to the transposer.) After 64 words have been stored, LED1 lights up. Q2 serves the purpose of turning on LED1 when pin 2 of the FIFO goes high. It also blinks on for every "shift in" command to verify that data has been entered. Now we have the chip full of words. We have to get them out. We can do this one of two ways. First you can clock it out from an external sequencer or other clock. With S2 in the "external" position, apply a positive-going pulse to the input of J3. For every clock pulse in, you get a shift out of the FIFO. For sequence use, this external trigger comes from one of the stage gate jacks of the sequencer. It can be made to trigger anywhere you want. I think normally it would be #1. The transposer shifts on the leading edge, so when your sequencer has finished the last stage and returns to #1, the rising gate output from stage #1 will trigger the transposer and change keys. This happens so fast that you don't hear any delay in VCOs, VCA, transposer, or anything.

Next, if you want to use the transposer with your keyboard, remove the signal from J3, and place S2 in "internal" position and the 555 will run and do the shifting out for you. LED2 should come on, indicating the clock is running.

After a BCD code has been outputed from the FIFO, it must control the on/off switches of the 4051, and that is easy to do. Since the 4051s have 8 switches and we need 14, you must use two. But, how do you turn one off and one on, and visa versa? Well, these 4051s are also BCD addressable with 1-2-4 BCD input. It has a Z output, and an Enable line (Pin 6) that turns all switches off when E= Logical 1, or 5 volts. Let's look at IC #6 and count - 1, 2, 3, 4, ---7. The BCD at the output of the FIFO looks like Fig #6.

Note: 8 'C', the inversion of this is a '1' so let's apply this to the 'E' input of IC7. Thus, IC7 is off and IC6 is up and running. Now let's put in an eight. So what happens, 8 goes high.

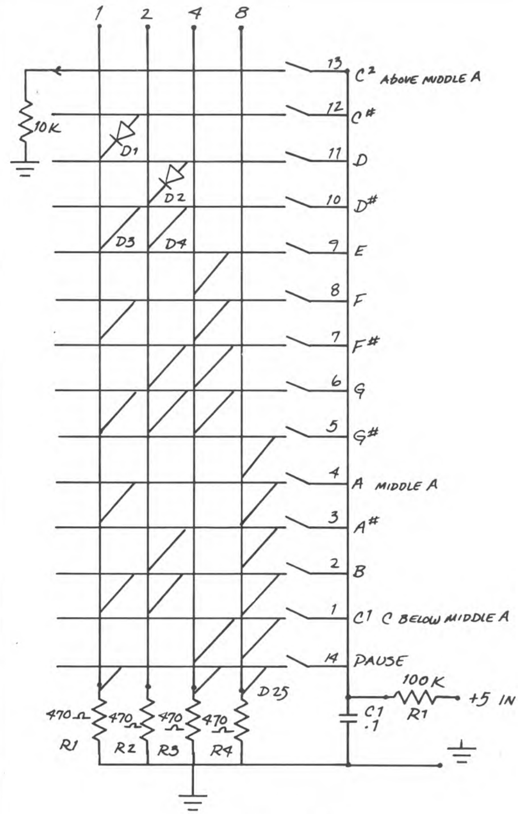


FIG. 5B

	1	2	4	8
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
↓				
7	1	1	1	0
8	0	0	0	1

FIG. 6

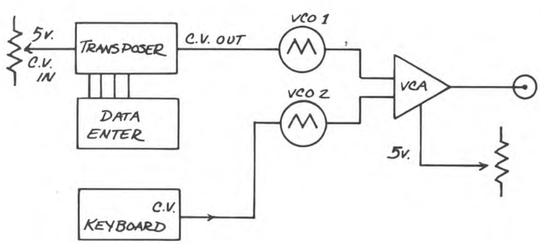


FIG. 7

The inversion of '1' is '0' so 'E' on IC7 goes low and IC7 is up and running and IC6 is off. The BCDs are paralleled on both 4051 and each receives a count from 0–7. We are almost done.

Inputs and outputs. The 'z' out of each 4051 are tied together and brought out to IC8 for buffering and is used as the new C.V. out. IC8 also has an offset adjust as is used in the present VCO current converters. The inputs of the 4051 have pots in series with them that set the half steps with the exception of position #1 and #14. #1 is a straight through and #14 is the blank, or 'No output', or 'Pause output', call it what you may. The other side of the pots are tied together and labeled V.C. in. And there you have it??!!! (Note: TTL chips were used for IC2, 3, and 4 because they were in my junk box, I suppose CMOS can be used just as well. Power requirements are + 9v, -9v, + 5v.)

Set up and calibration:

1. Check voltages for proper voltages of +9v + 5v -9v.
2. Place S2 on internal position, LED2 should light, indicating the 555 is running.
3. If you have a scope, check pin 6 of IC4, 74121, for shift out pulses.
4. Push key "C2" on data enter keyboard and use scope or VTVM and check for all '0' on data output pins of FIFO. This should address IC #6 and switch 1 should be closed. Plug up a fixed bias or keyboard control voltage to the VC input, there should be control voltage on the output.
5. Also check for '0' volts on 'E' input (pin 6) of IC6 and 5 volts on 'E' of IC7 (pin 6).
6. Now, with control voltage in, from keyboard, and a VCO hooked up to control voltage out, you should hear your note.
7. Press another key on data enter and this should change BCD code and resistors, and pitch should drop.
8. Calibration of pots: Use the patch shown in figure 7.
9. Set S2 on "internal". Push 'C' above middle 'A' on keyboard. Push C2 on data enter keyboard. Adjust the fixed bias going to transposer until you zero beat VCO #1 with VCO #2.
10. Push 'B' below 'C' on keyboard and push 'B1 or #1 on data keyboard and adjust R100 for zero beat.
11. Press A# on keyboard, and A# or #2 on data keyboard, and adjust R101.
12. Continue through all 12 pots.
13. There is one more adjustment, offset bias.
14. I guess the best way to do this is now connect the transposers CV into keyboard. (Fig. 8) Press C2 on data keyboard and adjust for zero beat between the two VCOs when high 'C' on

keyboard is pressed.

15. Press 'C' below middle 'A', and now press C1 on data keyboard, VCO1 should drop by one octave, if out of tune adjust offset bias.
16. Repeat steps 13-16.

There are a few options which can be used in some parts of the original circuit. The fast read out clock can be changed from a 555 to a 7413. The 555 has been used in many projects I have made, and they have all given good performance, but they all put noise on the B+ line. This is why I had to run it off of + 9volts, then drop it down with a zener diode. The 7413 does not do this, so it uses less parts.

Added notes on the operation of the 3341...if you have loaded the 3341, and are now clocking it out, you can also enter new data at the same time, however, you can not put in data when LED is on, this indicates a "full house". However, when the light goes out, you may enter new data..

Also added to this unit was a circuit that turned off the "sync" pulse that comes out of the sequencer. Some VCAs may pop if they are improperly calibrated, and if key #14 is depressed, feeding a negative voltage to the VCOs; this should kill the VCOs. But your Sequencer is still triggering the AR or ADSR, so the VCA may pop if not balanced. So ... if you route the sequencer trigger pulses to the transposer, play games with it, and bring it out again, it will function as before, but when #14 is depressed, the VCOs cut off AND no trigger pulses come out. Here is the circuit for that... See figure 10.

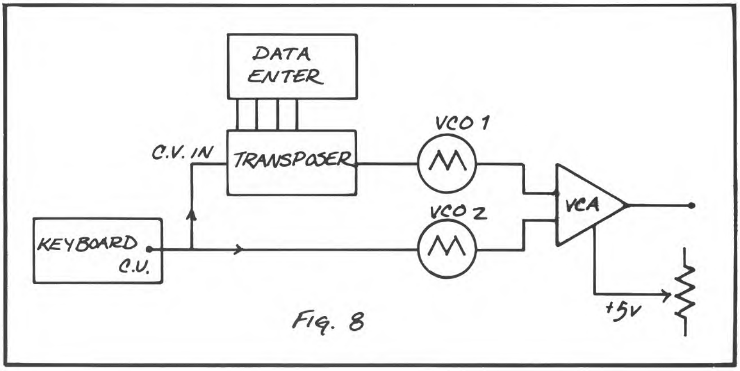


Fig. 8

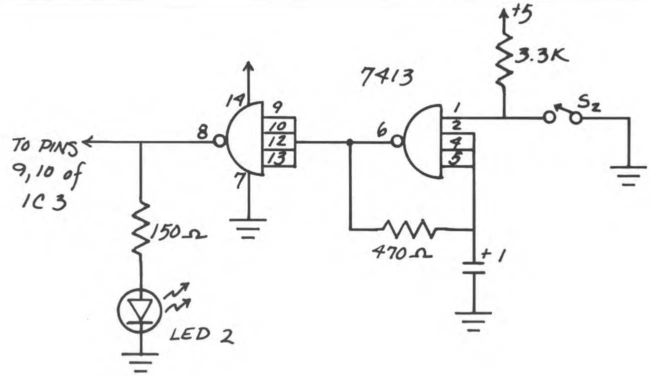


Fig. 9 : MODIFIED CLOCK

This does add to the cost of the unit, and is not really required, however, it can add a nice touch to the project.

All parts including FIFO and keyboard can be bought from "James Electronics" for about $25.00 to $30.00.

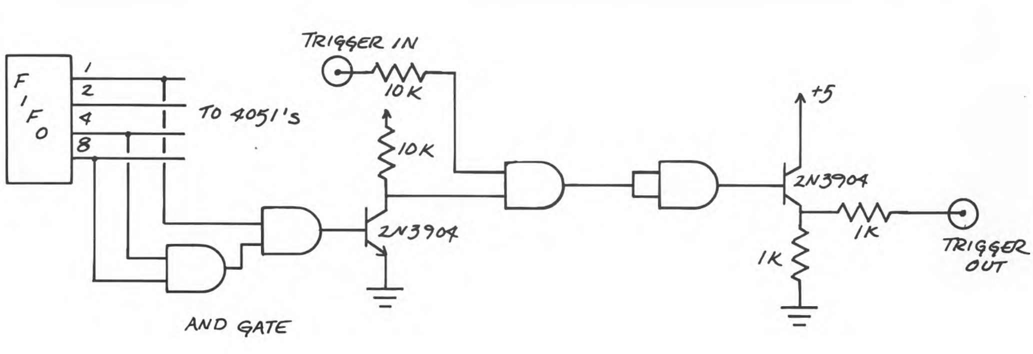


Fig. 10: TRIGGER CUT-OFF CIRCUITRY

...continued from page 15

Compare these two responses with those for the flanger in figures 6 and 7 and the true difference will become quickly apparent.

And now for a snappy summary that may serve to answer the original question, Is a Phlanger enough? Let's list some comparisons between the two units:

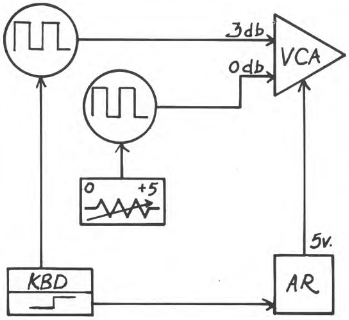
I) A flanger depends on time manipulation, while the phaser depends on PHASE manipulation.
II) The flanger affects ALL frequencies EQUALLY, while the phaser affects EACH frequency DIFFERENTLY.
III) The flanger is very smooth sounded, while the phaser has noticable bumps or resonances. The phaser may be made smoother by adding more stages.
IV) The flanger is presently more expensive than a phaser.
V) The flanger is more critical in its operation, than a phaser, and catastrophic failure of a flanger is more difficult and expensive to repair.
VI) Both units can be built with voltage control, variable resonance, internal clocks, and variable mix.

As an opiniated answer to the question, I don't think that either a flanger OR a phaser alone is the total way to go. As can be seen, each has an entirely different sound, although both operate in similar manners. If you're really into synthesis I don't think you can afford to own only a Phlanger, just as you can't pass up PAIA's deal to own a great piece of equipment at such a low price as the #1500 Phlanger.

References (and strongly suggested reading if you want the full story):

AUDIO HANDBOOK National Semiconductor Corp.; 2900 Semiconductor Dr.; Santa Clara, CA 95051.
FET PRINCIPLES, EXPERIMENTS, AND PROJECTS Edward M. Noll; Howard W. Sams & Co.; Indianapolis, Indiana 46268.
APPLICATIONS OF OPERATIONAL AMPLIFIERS Jerald Graeme; McGraw-Hill Book Co.; New York, New York.
ACTIVE FILTER COOKBOOK Don Lancaster; Howard W. Sams & Co.; Indianapolis, Indiana 46268.
SOUND RECORDING John Eargle; Van Nostrand Reinhold Company; New York, New York.

PATCHES



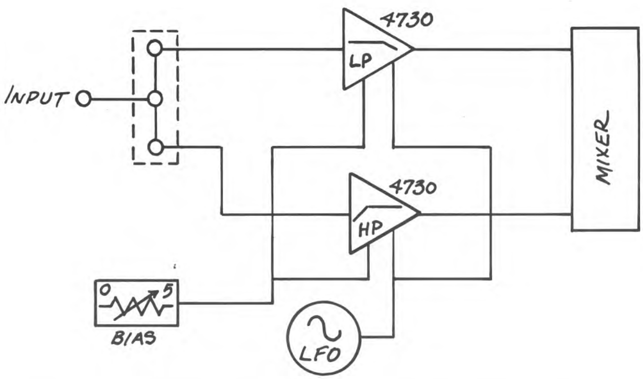
BAGPIPES

AR: Attack: Expand on, 0%
Release: 15%
Glide: medium amount
Oscillators: 20% pulse; tuned in mid range so that they are in unison when second "C" on keyboard is played.

The playing here should be very legato. Another "drone" oscillator would probably help, perhaps a fifth up.

Submitted by: Kenneth Keeler
Houston, TX

BI-PHASE IMITATION



This patch produces an imitation of the popular Mu-tron Bi-Phase.

The input is preferably "Thick" - multi-track brass or strings, although funky bass lines work well, too.

Settings:

Both filters - Q to suit.

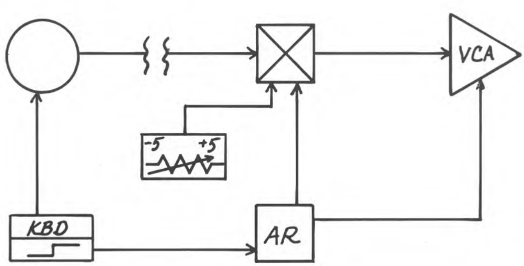
Initial frequency of Low Pass Must be slightly lower than that of high Pass but within one octave.

Set Bias to prevent Lock-up and/or Set sweep range.

LFO Frequency and Level to suit.

Submitted by: Gary Bannister
Indianapolis, IN

FAKE REVERB OR "BLIP"



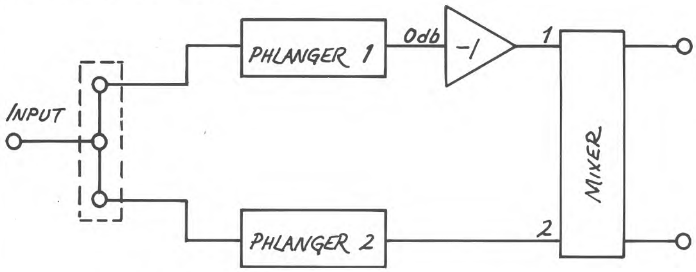
AR: both outputs 5v.
Bias: -2.5v.

Any patch into the balanced modulator can be used. AR settings are variable. A setting which produces this output: produces this envelope:

Submitted by: Kenneth Keeler
Houston, TX

PATCHES

THE "TOMITA FLANGE"

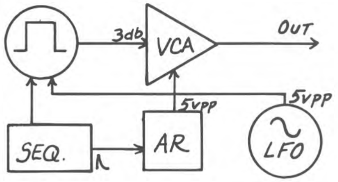


Control Settings:
Phlanger #1:
Speed: Minimum
Span: Minimum
Center: Maximum
Accent: To suit - minimum best
Mix: 100% Delayed only
Phlanger #2
Accent: To suit - minimum best
Mix: 100% Delayed only
Speed, Span, Center: Adjust the desired effect as though using a single phlanger
Inverter: use 0db input
Mixer: adjust input levels for best cancellation
Pan to suit, but Both inputs MUST be the same

This patch gives the distinctive "Eventide" cancellation used by Isao Tomita. It works best on "Thick" sounds - multi-track Brass, String Ensembles, or noise.

Submitted by: Gary Bannister
Indianapolis, IN

PSEUDO-RANDOM MELODY GENERATOR



SEQUENCER - SPEED - 60%
- 12 stage recirculation
- Alternate high and low stage settings, no glide
- 50% step trig. out

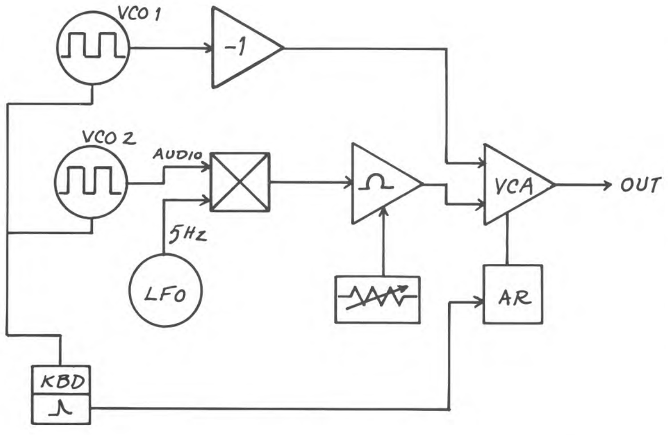
VCO - Pulse out 50% duty

AR - Attack 0%
- Decay 0%
- Expand off, 5vp-p

LFO - 1Hz, 5vp-p

Submitted by: David Setser
Morehead, KY

HONKY-TONK PIANO II



Keyboard:	High range - no glide
VCO 1 & 2:	Tuned almost in unison (about 30Hz beat)
LFO:	5Hz/VCF: Q maximum
AR:	Attack minimum/ Release 1/3, expand off

While somewhat more involved than Gary Bannister's patch in Polyphony 1/76, this patch has a nice metallic sound which I like and sounds good even down at the low end of the keyboard.

Submitted by: Ron Jones
Orange, CA

LAB NOTES: MUS 1 with the new miracle ingredient – stg

by John S. Simonton, jr.

With the exception of the bare-bones listing of POLY 1.0 that ran in the last issue, we haven't looked at any software- mainly because there was little to examine.

But MUS1 was just recently finalized, so that situation is beginning to change.

MUS1, for the benefit of those of you who haven't been waiting for it for the last six months, is what many would call "system firmware"- and since that has the sort of technical ring to it that tends to make things interesting, we'll call it that, too.

In almost any computer application there are some programs which, for one reason or another, are best handled as firmware- a name that these days means not software (which must be loaded from some storage media external to the computer) and not hardware (a permanently wired collection of gates, etc. which cause a specific, set sequence of actions to take place) but something betwixt and between; most usually, software that is contained in a PROM somewhere.

The most obvious firmware is a monitor program such as PIEBUG. Since this program is the thing that allows for the entry of data and instructions into the memory of the computer in the first place (as well as usually providing whatever de-bugging and editing features the designer thought were important and/or had room for), it is at least inconvenient to have to load it every time it is needed. Much better to have it in a dedicated PROM where it is always available for immediate use.

The firmware of MUS1 is roughly analogous. These are universally useful routines that, with rare exceptions, will be used with everything we do musically. It's a waste of time and resources to have to load them to RAM from tape (or worse yet, manually) every time they're needed. A PROM is their happiest home. In our 8700 Computer/Controller, MUS 1 is a 1702A PROM that occupies the address range $D00-$DFF (IC-17).

Examples? OK, the keyboard reading routine (LOOK). It isn't particularly long or complicated (a little over 30 bytes) but we're going to need it every time we turn on the system- even if it isn't used to read the keyboard, it's the thing that our protocols dictate will be the tempo-determining element in the system (based on the clock rate of the encoder). At some future date the occasion may arise when we can examine this in detail. Today, it's not the point.

The QuAsh drivers (called NOTE)- same thing- we're going to need them for almost everything we do. Why bother to load them?

In addition to these two routines, MUS1 also contains:

INIT: an initialization routine that takes care of setting various variables and buffer areas to a known, acceptable state (as opposed to the random numbers they will contain when power is first applied.)

POLY: essentially the polyphonic (I still prefer polytonic) allocation algorhythm from POLY1.0, except refined somewhat to take less memory space.

TRGN: The new miracle ingredient- Software Transient Generators (STG). A routine that will serve as a software substitute for ADSRs.

OPTN: A very simple option selecting program that allows the remaining firmware of MUS1 to be tied together into a 16 voice polyhonic synthesizer with or without software transient generators- without having to lead any additional software (though several parameters will need to be initiallized manually).

All of this is pretty straight-ahead code that should be understandable from the documented listing that appears at the end of this article- you may need to refer back to previous articles in this series for background information; "In Pursuit of the Wild QuAsh" (reference Polyphony, July '77) and "What the Computer Does" (reference Polyphony 4/76) would be particularly useful ones.

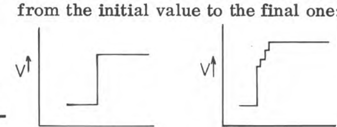
Two exceptions, NOTE and TRGN, need some additional explanation - they introduce some new ideas.

In an embryonic form, NOTE was a part of POLY1.0. It is the responsibility of this routine to take individual entries from the output buffer area (NTBL), add to it the corresponding entry from the Transpose buffer area (TTBL)and output the results to the QuAsh channels. Some aspects of the significance of the addition that takes place will be seen when we look at TRGN- for now, it will suffice to say that this will be an extraordinarily handy convention in a number of cases.

A more important function of NOTE is to make sure that what comes out of the QuAsh channels has no annoying glitches that may be artifacts of the D/A and multiplexing process. In an earlier story, we looked at one of the annoyances - the fact that our 8780 D/A, though quick, takes a finite amount of time to change from one value to the next and if appropriate settling time is not allowed between writes to the QuAsh channels we will be able to hear the changes as a slight "buzz" in each of the channels. The solution here is to output the data first to a "dummy channel" that is occupied solely by the D/A, with no corresponding QuAsh, followed immediately by a write of the identical information to an output which does correspond to a QuAsh channel. The first write allows the D/A to settle while the second strobes the settled output into the appropiate QuAsh channel.

And here we come face to face with the next problem; the QuAsh really need some settling time since they are at their heart nothing more than an RC circuit.

As long as we are thinking in terms of small systems(8 output channels or less) this is not a big problem since it can be dealt with simply by delaying after writing to one QuAsh but before setting up the next. If the delay is not long enough, we will hear changes from one value to the next not as an instantaneous change, but rather as a series of steps from the initial value to the final one:



We want this

But if the QuASH settling delay is not long enough, will get this.

figure a

In larger systems, this constant delay approach is not a practical solution because there is not enough time during alternate "dummy" scans of the keyboard (the time which our conventions allow for processing, output driving, allocations, etc.) to allow all of the output channels the luxury of a delay. The time comes for the keyboard to be read again (or other things to happen) and the processor is still busy waiting for all of those QuAsh to get to the right value.

The key to the solution of this problem is to notice that there is really only one set of circumstances under which the long QuAsh-settling delay is required, and that's when the output of one of these channels must change from one value to another (which happens only a small percentage of the time) and then, only when the glide of the channel is turned off. (if the glide is on, its integrating action will smooth out the steps; and, in fact, a short write time is preferable here since it will serve to increase the time required for the glide.)

The actual solution is what I feel we should call "DYNAMIC QuAsh DRIVERS" -a small block of programming, more or less in the middle of NOTE.

This part of the program first checks to see if the glide control bit (the most significant bit of the data just written to D/A and S/Hs) was turned on or not. If we are in "glide mode," no delay is required so the program immediately goes to see if there are any channels left to write; if there are, it services them.

If the glide is not on, we have a candidate for dynamic operation so the dynamic mode switch is checked (more later) and if this option is selected the current data is compared to the data that was previously written to this channel (requires a new table that we've generated called "LAST") and if they're different (a change), the program goes into the delay that allows the output of the QuAsh to instantaneously (apparently) step from its previous value to the next one. The new value is saved in LAST (for use next time) and if there are more channels to do- it does them.

SOFTWARE TRANSIENT GENERATORS

Here we begin, for the first time, to replace some of the elements that constitute traditional synthesizer hardware with software that performs the same function (hopefully as well, or better) with less costly hardware. STGs are a good place to start because they're not super difficult to implement.

Just like their hardware equivalents, STGs respond to a note which has just been triggered (pressed on the keyboard) by producing a voltage that rises at a controlled Attack rate. After reaching some peak value, the voltage then drops at a Decay rate until it reaches a pre-set Sustain level where it stays as long as the note remains triggered. When the key is released, the voltage drops to its lowest level at the Release rate.

Computing the number which represents the current value of the transient is only slightly more complicated than adding, subtracting and comparing.

Unlike an ADSR, an STG has no knobs to set, in their place you enter numbers setting Attack rate, etc. into the computer.

Perhaps the biggest problem having to do with STGs is deciding where they should come out. Oh, the QuAsh channels, obviously; but which ones? Of the numerous possibilities, we've selected the convention of having pitch setting voltages (those that correspond to notes) and transient voltages come from alternate QuAsh channels, primarily because this will work nicely with some stuff under development (or consideration, at least), without making obsolete all of the hardware that we've accumulated up to now.

This implies two distinct modes of operation; the first in which the STGs are not asserted and POLY assigns notes to sequential QuAsh channels; and, the second mode (STGs on), in which notes are assigned by POLY* to the odd number QuAsh channels (first, third, etc.) while transients are produced at the even number outputs (second, fourth, etc.).

The note produced at the first QuAsh output has a corresponding transient happening at the second output, and so on. Just as if the trigger from the first channel were patched to the input of an ADSR whose output was somehow tied to the output of the second QuAsh channel.

This would seem a good place to mention (in case it's not already obvious) that in this implementation all of the STGs produce the same kind of transient, and for the kinds of things that we're doing now, this is how it should be. It may also be worth mentioning that while the transients are all the same, they are totally independent where following the triggered and released states of their respective note channels is concerned.

There are also some internal details which muddy the STG waters. For instance, a key that is currently down may require a transient function that is either in the Attack cycle (increasing) or Decay/Sustain cycle (decreasing or holding) depending on its past history (had it already peaked?). Somewhere we need to save information on which cycle the transient is actually in.

Another, somewhat interrelated, problem concerns the smoothing of the transient waveform. Under most conditions, the glide of the QuAsh channels that are being used as transient outputs should be turned on so that a smoothly increasing or decreasing function is produced. But, the glide can't always be on because that would limit the maximum attack rate.

Without having the space to cover it entirely, I can only state that the solution to both of these difficulties lies in the use of the Transpose table and remembering that the data stored in TTBL entries is added to the output parameter in NTBL (where we're storing the actual current value of the transient) before the output operation takes place. Note also that while the data in NTBL is manipulated extensively by POLY and TRGN (as they calculate, allocate, - regurgitate?) TTBL is untouched by computer hands, and this makes it an ideal place to save control type functions. Not only transpositions, but a place that glide and trigger bits and such can be permanently set.

* Note that POLY checks to see if the STGs are turned on as it assigns notes to outputs.

These locations are so handy for this application that in TRGN they have been re-named CWRD (Control-Words... but do not be confused, this is still our old friend TTBL and has no relationship at all to the System Control Word-CTRL) and it is here that we keep track of the A/D/S state of each of the transient channels.

Also, to help me keep things straight in my own mind, the NTBL bytes that are used to store the current value of the transient have been re-named PARM (parameter); but, again, this is the same physical area as NTBL.

NOW, HOW DO WE USE ALL THIS?

Perhaps the best way to begin an essay on how to use MUS1 is to state one of the functions that it was devised to perform

As you are no doubt beginning to realize, we've carefully developed a system that will have applications far beyond what we've discussed to this point. It's complex; and while the complexity implies unmatched versatility, it undeniably has its intimidating aspects.

At one level of use, MUS1 should reduce this intimidation by giving the user an instrument with a specific (though within certain limits alterable) personality the instant that it's turned on, without having to hassle around with loading any additional programs (success) or variables (well...)

Also, these program modules should be written so that they easily interface with future expansions of the system, either hardware or software, so that, when needed, they can be accessed by programs offering distinctly different personalities (success here maybe- only time will really tell).

While we've reduced the intimidation, we've not eliminated it entirely because even when using MUS1 as a stand-alone personality there are some variables which must be initialized before you begin to play- some information that the system must have in order to operate properly. This data could be part of the PROM, but not without significantly compromising versatility.

For instance, we've mentioned in passing a couple of times the System Control Word-CTRL. This is a single word in the computer's RAM memory at location $0E8.

It is most helpful to visualize CTRL as a collection of eight "switches", each bit representing one switch. To MUS1, only two of these switches have any significance– D7, which turns the STGs on and off, and D6, which enables or disables the dynamic mode option. The rest are reserved.

Every time you power up the system, CTRL must be set so that the desired options are selected– there is no default setting that is part of MUS1. If you want dynamic mode (which you should, for now) then bit 7 should be turned on. If you want STGs, bit 8 must be set.

The 4 possible combinations of these 2 bits then have the following significance:

binary	hex	action
00000000	$00	STGs off; dynamic mode off
01000000	$40	STGs off; dynamic mode on
10000000	$80	STGs on; dynamic mode off
11000000	$C0	STGs on; dynamic mode on

CTRL is not the only variable which must be initialized manually. There's also:

```
EXTERNALLY INITIALIZED VARIABLES
================================
 LOC. LABEL          USE
--------------------------------
 0E8  CTRL   SYSTEM CONTROL WORD
             D7 SET TURNS ON
             TRANSIENT GENERATORS
             D6 SETS DYNAMIC MODE

 0E9  ODLY   SETS OUTPUT DELAY;
             IN DYNAMIC MODE $20
             RECOMMENDED

 0EA  OUTS   NUMBER OF HARDWARE
             SUPPORTED CONTROL
             CHANNELS AVAILABLE

 -- AND TRANSIENT PARAMETERS --

 0BA  ATCK   ATTACK RATE
 0BB  DCY    DECAY RATE
 0BC  SUST   SUSTAIN LEVEL
 0BD  RELS   RELEASE RATE
 0BE  PEAK   PEAK VALUE -SEE TEXT

             RATES: $01 (SLOW)
                    $3F (FAST)
             LEVEL: $01 (MINIMUM)
                    $3F (MAXIMUM)
--------------------------------
```

Most of these are easily understood or have been examined in the past, so we won't go into any great detail. A few points are worth mentioning, however.

ODLY– this is a number that represents the delay that the QuAsh drivers will use, when required. For normal use, a value in the range of $20–$30 is most appropriate.

OUTS– this variable tells the POLY subroutine how many output channels it has to work with, so that notes don't get lost; we talked about this last time. Now we need to notice that when the STGs are asserted we should think of the QuAsh channel that is producing the transient as simply an extension of the channel producing the note. In other words, the two QuAsh channels constitute a single "hardware supported" channel. A single QuAsh represents two such channels.

ATCK/DCY/SUST/RELS– When the transient generators are turned on, we also need to enter the attack, delay, sustain and release parameters that we want produced. These four entries should need little explanation other than the examples which follow shortly; their range is from $01–$3F, with $01 representing the lowest rate or level and $3F the highest.

PEAK– this fifth transient parameter needs a little extra attention. PEAK has only one use; it determines whether the transient produced is going to be percussive (quickest possible attack and full ADSR segments) or non-percussive. In the non-percussive mode, the glide is on for all segments of the transient and the Decay and Sustain states of the transient are eliminated entirely.

In fact, there is only one bit in the word PEAK that is changed to select one of these two options– the most significant bit. The remaining seven bits should (for now– until you have a real feel for what's happening) be set to $3F (00111111 in binary). If the most significant bit of this word is cleared, you're in percussive mode. If the bit is set (so that PEAK contains $BF – 10111111 in binary) you are in non-percussive mode.

The differences between the two are great. Assume for a moment that we have set the ADSR parameters at $3F/$04/$20 /$01 respectively (fastest attack/moderated decay/medium sustain/slowest release) and that we are only going to change the PEAK parameter. If PEAK contains $3F(percussive mode), a 'scope display of the transient will look something like this:

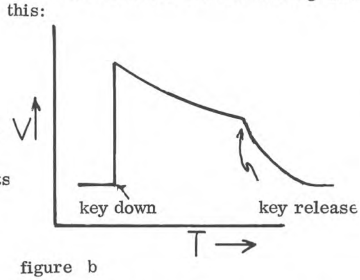


figure b

Setting PEAK to $BF (non-percussive) produces this result:

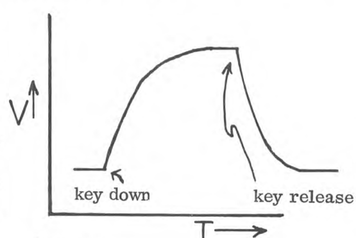


figure c

Because the glide is now on during the entire attack cycle and the Decay and Sustain portion of the transients are eliminated. Straightforward stuff, really.

We need to cover an example of system set-up before we wind up, but first must notice that the effect of having the PEAK parameter are far more far-reaching than we've been able to cover in detail. A quick example:

ADSR parameters set to $10/$04/$20 /$01 and PEAK containing $3F will produce this kind of transient:

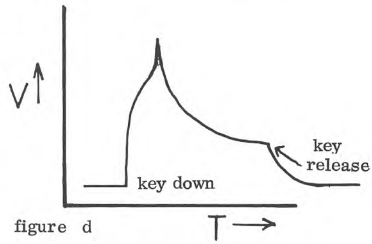


figure d

which, when heard, starts out with a non-percussive kind of "swell" with a percussive "pip" added at the last instant before the transition to the Decay and Sustain cycles. This would seem to be a unique and useful transient that isn't produced by traditional ADSRs.

Along the same lines, the TSGs can be considered to be "better" than our hardware ADSRs in that they need not finish the Attack cycle before transitioning to the Release state. If a key is released before its transient has gone all the way to PEAK, the transient immediately switches to the release state. This is frequently called "muting' and it offers the possibility of effective control of expression directly from the AGO keyboard.

A SUMMARY, OF SORTS

So, we've gotten our hands on a MUS1 PROM and are ready to start doing things. What has to be done first? Really very little.

First, the System Control Word, Output Delay and number of hardware channels available must be set. For example:

keystrokes	explanation
0-E-8-DISP	sets monitor pointer to $E8-CTRL
C-0-ENT	sets $E8-asserts STGs dynamic mode
3-0-ENT	sets ODLY value
0-2-ENT	sets output channels at 2

these entries define the personality of the instrument as a 2 voice polyphonic synthesizer (notes from channels A & C) with software transient generators (which appear at QuAsh channels B & D) .

Next, we must set the transient parameters to the desired values:

keystrokes	explanation
0-B-A-DISP	sets monitor pointer to $BA- ATCK
3-F-ENT	sets shortest attack
0-4-ENT	sets moderate decay
2-0-ENT	sets moderate sustain
0-1-ENT	sets slowest release
3-F-ENT	percussive mode

and you may recognize these parameters as being those that we examined in the illustration earlier.

Finally, we simply begin running the program:

keystrokes	explanation
D-0-0-DISP	sets monitor pointer to beginning of OPTN
RUN	presto- the program runs

A typical patching configuration that would be consistent with these entries would look something like this:

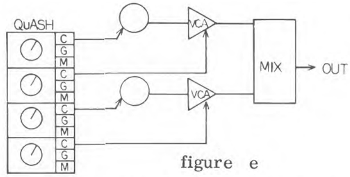


figure e

Oh, yes- I almost forgot. OPTN, like POLY 1.0, uses the 8700 keyboard to control two important functions. While OPTN is running, touching key 0 of the control keyboard will cause the entire system to be re-initialized. Not the entries that we made manually- those remain unchanged, but all the notes and transients go immediately to zero level.

Similarly, touching key #1 produces a tuning function that makes the synthesizer respond as if all the channels were seeing the second C on a three octave keyboard held down. The transients go, the notes play, etc. After tuning, be sure to re-initialize the system by touching control key #0.

I prefaced one of the earlier paragraphs with "at one level of use." In all of the preceding words, that's all that we've examined- one level of use (the simplest and most obvious level, at that.) . I've also referred in the past to "software modules" which can be strung together in different ways (just as can hardware modules) to produce different effects and personalities. MUS1 is the first set of these modules.

With more regret than you can imagine, I haven't the space here to go into all of the implications of this (even if I knew them all, which I'm sure I don't).

Providing you're more than just casually interested, you should spend some time trying to understand how MUS1 works internally (there are numerous different entry points to the routines that we haven't covered - for instance). I believe that the time investment will be wisely made.

```
                               ;*********************************
                               ;*                               *
                               ;*             MUS1              *
                               ;*                               *
                               ;*      BY JOHN SIMONTON         *
                               ;*(C) 1978 PAIA ELECTRONICS, INC*
                               ;*     ALL RIGHTS RESERVED       *
                               ;*                               *
                               ;*                               *
                               ;*  SYNTHESIZER SUBROUTINES      *
                               ;*         ++ AND ++             *
                               ;*  MULTIPLE OPTION POLYPHONIC   *
                               ;*   ALLOCATION PROGRAM WITH     *
                               ;*     SOFTWARE TRANSIENT        *
                               ;*         GENERATION            *
                               ;*                               *
                               ;*********************************
                               ;
                               ;           OPTN
                               ;POLYPHONIC SYNTHESIZER / OPTION
                               ;         SELECTION
                               ;*********************************
                               ;
                               ;
                               INIT . DL 0D21
                               POLY . DL 0D71
                               TRGN . DL 0DC3
                               NOTE . DL 0D2B
                               DECD . DL 0F00
                               FILL . DL 0D52
                               DISP . DL 0820
                               CLCK . DL 00BF
                               ;
                               ;OPTION TIES MUS1 FIRMWARE
                               ;TOGETHER INTO A POLYPHONIC SYNTH
                               ;WITH OR WITHOUT TRANSIENT GENER-
                               ;TION; W/WO DYNAMIC QUASH DRIVERS
                               ;
                               ;ALSO USES PIEBUG DECODE AND
                               ;ASSIGNS KEY #0 AS SYSTEM CLEAR
                               ;AND #1 AS TUNE - EQUIVALENT TO
                               ;ALL CHANNELS 2ND "C" ON KBD DOWN
                               ;
0D00-    20 21 0D  OPTN JSR INIT     ;ZERO ALL BUFFS
0D03-    20 71 0D  LOOP JSR POLY     ;ALLOCATE CHANS
0D06-    20 C3 0D       JSR TRGN     ;NEW TRANSIENTS
0D09-    20 2B 0D       JSR NOTE     ;OUTPUT-READ AGO
0D0C-    A5 BF          LDA *CLCK    ;GET CLOCK VALUE
0D0E-    8D 20 08       STA DISP     ;RAZZ-MA-TAZZ
0D11-    20 00 0F       JSR DECD     ;CHECK COMMANDS
0D14-    C9 01          CMP 01       ;0?  1?  >1?
0D16-    30 E8          BMI OPTN     ;0; CLEAR ALL
0D18-    D0 E9          BNE LOOP     ;>1; KEEP ON
0D1A-    A0 5C          LDY 5C       ;1; TUNE 2ND C
0D1C-    20 52 0D       JSR FILL     ;KEYS ALL DOWN
0D1F-    F0 E2          BEQ LOOP     ;BRANCH ALWAYS
                               ;
                               ;          INIT
                               ;  INITIALIZATION ROUTINE
                               ;*********************************
                               ;
                               CTRL . DL 00E8
                               TBEG . DL 00BF
                               ;
                               ;INIT CLEARS INPUT BUFFER (KTBL)
                               ;OUTPUT BUFFER (NTBL) AND TRANS-
                               ;POSE BUFFER/CONTROL WORDS (TTBL)
                               ;HEXADECIMAL MODE IS SELECTED
                               ;
                               ;ENTER AT INT0 TO FILL TABLES
                               ;WITH CHARACTER FROM ACCUMULATOR
                               ;
0D21-    A9 00     INIT LDA 00        ;PREPARE TO ZERO
0D23-    A2 28     INT0 LDX 28        ;SET POINT/COUNT
0D25-    D8             CLD           ;SET HEX MODE
0D26-    95 BF     INT1 STA *TBEG,X   ;ZERO BUFFER
0D28-    CA             DEX           ;POINT TO NEXT
0D29-    D0 FB          BNE INT1      ;SOME LEFT -LOOP
                               ;
                               ;        NOTEOUT/LOOK
                               ;16 CHANNEL QUASH DRIVERS AND AGO
                               ;   KEYBOARD READING ROUTINE
                               ;
                               ;*********************************
                               ;
                               CTRL . DL 00E8
                               ODLY . DL 00E9
                               KTBL . DL 00DF
                               NTBL . DL 00CF
                               TTBL . DL 00BF   ;ALSO CLCK
                               LAST . DL 00A9
                               S/H  . DL 09EF
                               D/A  . DL 0900
                               KBD  . DL 0810
                               ;
                               ;       *** NOTEOUT ***
                               ;     DYNAMIC QUASH DRIVERS
                               ;GETS NOTES TO BE PLAYED FROM THE
                               ;OUTPUT BUFFER (NTBL) AND ADDS
                               ;TRANSPOSING VALUE FROM TRANSPOSE
                               ;BUFFER (TTBL).   OUTPUTS RESULT
                               ;
0D2B-    A2 10     NOTE LDX 10        ;SET POINTER
0D2D-    B5 CF     NO0  LDA *NTBL,X   ;GET NOTE
0D2F-    18             CLC           ;PREPARE AND
0D30-    75 BF          ADC *TTBL,X   ;ADD TRANSPOSE
0D32-    8D 00 09       STA D/A       ;LET D/A SETTLE
0D35-    9D EF 09       STA S/H ,X    ;WRITE TO S/H
                               ;
                               ;NOW THE DYNAMIC PART; IF GLIDE
                               ;IS ON,DELAY IS SKIPPED.  IF NOTE
                               ;IS SAME AS LAST PLAYED (IGNORING
                               ;CONTROL BITS D6 & D7) DELAY IS
                               ;SKIPPED.   IF NOT IN DYNAMIC MODE
                               ;AND NO GLIDE, DELAY ALWAYS TAKEN
                               ;
0D38-    30 0F          BMI NO2       ;GLIDE? NO DELAY
0D3A-    09 80          ORA 80        ;IGNORE FLAGS
0D3C-    24 E8          BIT *CTRL     ;DYNAMIC MODE ?
0D3E-    50 04          BVC DLAY      ;NO,JMP TO DELAY
0D40-    D5 A9          CMP *LAST,X   ;COMPARE TO LAST
0D42-    F0 05          BEQ NO2       ;SAME:SKIP DELAY
0D44-    A4 E9     DLAY LDY *ODLY     ;GET DELAY VALUE
0D46-    88        NO1  DEY           ;DECREMENT DELAY
0D47-    D0 FD          BNE NO1       ;LOOP TIL DONE
0D49-    95 A9     NO2  STA *LAST,X   ;FOR NEXT TIME
```

```
0D4B-    CA              DEX          :POINT TO NEXT
0D4C-    D0 DF           BNE NO0      :SOME LEFT -LOOP
                    :
                    :LOOK WAITS FOR THE BEGINNING OF
                    :AN "ACTIVE" SCAN-BEGINS PUTTING
                    :THE NUMBERS OF KEYS DOWN IN SE-
                    :QUENTIAL IN-BUFF WORDS.  WHEN
                    :SCAN DONE REMAINING IN-BUFF IS
                    :ZERO'D.
                    :
0D4E-    E6 BF      LOOK INC *TTBL     :INCREMENT CLOCK
0D50-    A0 00           LDY 00       :PREPARE FOR CLR
0D52-    A2 08      FILL LDX 08       :SET UP POINTER
0D54-    AD 10 08   LK2  LDA KBD      :WAIT FOR
0D57-    30 FB           BMI LK2      :"ACTIVE" SCAN
0D59-    AD 10 08   LK3  LDA KBD      :GET KEY
0D5C-    30 0F           BMI DONE     :END SCAN? -CLR
0D5E-    2A              ROL          :STROBE TO D7
0D5F-    10 F8           BPL LK3      :D7=0, NO STROBE
0D61-    6A              ROR          :RESTORE DATA
0D62-    95 DF           STA *KTBL,X  :TO IN BUFFER
0D64-    CD 10 08   LK4  CMP KBD      :NOW WAIT FOR
0D67-    F0 FB           BEQ LK4      :NEXT KEY
0D69-    CA         LK0  DEX          :PNT TO NEXT BUF
0D6A-    D0 ED           BNE LK3      :SOME LEFT -LOOP
0D6C-    60         RTN  RTS          :LEAVE
0D6D-    94 DF      DONE STY *KTBL,X  :ZERO IN-BUFFER
0D6F-    30 F8           BMI LK0      :BRANCH ALWAYS
                    :
                    :            POLY
                    : A LIMITED RESOURCE ALLOCATION
                    :          ALGORHYTHM
                    :*********************************
                    :
                    OUTT . DL 00EB
                    OUTS . DL 00EA
                    CTRL . DL 00E8
                    KTBL . DL 00DF
                    NTBL . DL 00CF
                    :
                    :POLY-FIRST HALF OF ALGORHYTHM
                    :IN THIS BLOCK DE-ACTIVATED CHANS
                    :ARE REACTIVATED IF THE DATA THEY
                    :CONTAIN APPEARS IN THE IN BUFFER
                    :
                    :D7 IN CTRL SET - ALTERNATE MODE
                    :D7 "    "   CLR - SEQUENTIAL MODE
                    :
0D71-    A5 EA      POLY LDA *OUTS     :# OF OUT CHANS
0D73-    85 EB           STA *OUTT    :USE AS COUNTER
0D75-    A2 10           LDX 10       :PREPARE PNT/CNT
0D77-    B5 CF      POL0 LDA *NTBL,X  :GET NOTE
0D79-    F0 27           BEQ NWKY     :0-OLD KEYS DONE
0D7B-    29 7F           AND 7F       :CLEAR D7
0D7D-    09 40           ORA 40       :SET D6
0D7F-    A0 09           LDY 09       :PREPARE PNT/CNT
0D81-    88         LP0  DEY          :POINT NEXT KEY
0D82-    F0 12           BEQ NEXT     :DONE -NEXT NOTE
0D84-    D9 DF 00        CMP KTBL,Y   :SAME AS KEY?
0D87-    D0 F8           BNE LP0      :NO -NEXT KEY
0D89-    95 CF           STA *NTBL,X  :SAVE NOTE D6=1
0D8B-    C6 EB           DEC *OUTT    :ONE LESS OUTPUT
0D8D-    F0 33           BEQ OUT      :NONE LEFT-LEAVE
0D8F-    A9 00           LDA 00       :OR PREPARE AND
0D91-    99 DF 00        STA KTBL,Y   :ELIMINATE KEY
0D94-    F0 04           BEQ LP1      :& BRANCH ALWAYS
0D96-    29 BF      NEXT AND 0BF      :CLEAR TRIG (D6)
0D98-    95 CF           STA *NTBL,X  :& RESTORE NOTE
0D9A-    24 E8      LP1  BIT *CTRL    :ALTERNTE MODE?
0D9C-    10 01           BPL SKP1     :NO -DEC. ONCE
0D9E-    CA              DEX          :YES-DEC. TWICE
0D9F-    CA         SKP1 DEX          :POINT NEXT NOTE
0DA0-    D0 D5           BNE POL0     :SOME LEFT -LOOP
                    :
                    :NEWKEY - SECOND HALF. KEYS DOWN
                    :ARE ASSIGNED TO OUTPUT BUFFER
                    :LOCATIONS WHICH ARE STILL DE-
                    :ACTIVATED
                    :
0DA2-    A2 10      NWKY LDX 10       :NTABLE PNT/CNT
0DA4-    A0 09           LDY 09       :KTABLE PNT/CNT
0DA6-    A9 40      NK1  LDA 40       :PREPARE MASK
0DA8-    35 CF           AND *NTBL,X  :NOTE TRIGGERED?
0DAA-    D0 0E           BNE NK3      :YES -GO TO NEXT
0DAC-    88         NK2  DEY          :POINT NEXT KEY
0DAD-    F0 13           BEQ OUT      :NONE LEFT-LEAVE
0DAF-    B9 DF 00        LDA KTBL,Y   :KEY NEEDS HOME?
0DB2-    F0 F8           BEQ NK2      :NO -GET NEXT
0DB4-    95 CF           STA *NTBL,X  :YES-PUT IN NOTE
0DB6-    C6 EB           DEC *OUTT    :ONE LESS OUTPUT
0DB8-    F0 08           BEQ OUT      :NONE LEFT-LEAVE
0DBA-    24 E8      NK3  BIT *CTRL    :ALTERNATE MODE?
0DBC-    10 01           BPL SKP2     :NO -DEC ONCE
0DBE-    CA              DEX          :YES-DEC TWICE
0DBF-    CA         SKP2 DEX          :POINT NEXT NOTE
0DC0-    D0 E4           BNE NK1      :SOME LEFT -LOOP
0DC2-    60         OUT  RTS          :RETURN
```

```
                    :        TRGN
                    :TRANSIENT GENERATOR PROGRAM
                    :
                    :***************************
                    :
                    :
                    CTRL . DL 00E8
                    ATCK . DL 00BA
                    DCY  . DL 00BB
                    SUST . DL 00BC
                    RLS  . DL 00BD
                    PEAK . DL 00BE
                    NTBL . DL 00CF
                    PARM . DL 00CE
                    TTBL . DL 00BF
                    CWRD . DL 00BE
                    :
                    :       NTBL 00D0-00DF
                    :       TTBL 00C0-00CF
                    :
0DC3-    A5 E8      TRGN LDA *CTRL    :DO TRANSIENTS?
0DC5-    10 38           BPL RTN1     :NO  -RETURN
0DC7-    A2 10           LDX 10       :NTABLE PNT/CNT
                    :
                    :   A/D/S/R DETERMINATION
                    :ROUTINE PREPARES Y TO USE AS
                    :CONTROL WORD, GETS NOTE AND
                    :SHIFTS TRIG. TO CARRY, GETS
                    :CURRENT STATE (CS) PARAMETER.
                    :IF NOTE TRIG. NOT SET STATE IS
                    :RELEASE.  IF CS PARA IS POSI-
                    :TIVE STATE IS DECAY/SUSTAIN
                    :OTHERWISE, STATE IS ATTACK
                    :
0DC9-    A0 40      ADSR LDY 40       :PREPARE CWRD
0DCB-    B5 CF           LDA *NTBL,X  :GET NOTE AND
0DCD-    2A              ROL          :ROTATE TRIGGER
0DCE-    2A              ROL          :TO CARRY BIT
0DCF-    B5 CE           LDA *PARM,X  :GET CS PARA.
0DD1-    90 19           BCC RELS     :NO TRIG? -RLS
0DD3-    10 0B           BPL DS       :CS>0? -DECAY/S
                    :
                    :   ATTACK ROUTINE
                    :ADDS ATTACK PARAMETER TO CS PARA
                    :AND IF GREATER THAN PEAK
                    :SUBSTITUTES $3F AND SETS CONTROL
                    :WORD TO $40 (D6 SET - NO GLIDE).
                    :NOTE THAT CS PARA WILL BE >0
                    :WHEN NEXT CHECKED.
                    :
0DD5-    18         ATTK CLC          :PREPARE
0DD6-    65 BA           ADC *ATCK    :ADD ATTACK PARA
0DD8-    C9 BF           CMP 0BF      : >PEAK
0DDA-    90 1B           BCC NEXT     :NO -PLACE PARA.
0DDC-    A5 BE           LDA *PEAK    :YES-PEAK VALUE
0DDE-    D0 17           BNE NEXT     :BRANCH ALWAYS
                    :
                    :   DECAY AND SUSTAIN ROUTINE
                    :NOTE THAT CARRY IS SET.  DECAY
                    :PARAMETER IS SUBTRACTED FROM
                    :CURRENT STATE PARAMETER.  IF
                    :RESULT IS LESS THAN SUSTAIN
                    :PARAMETER THEN SUST. PARA.
                    :BECOMES CURRENT STATE PARA.
                    :D6 & D7 OF CONTROL WORD SET
                    :
0DE0-    A0 C0      DS   LDY 0C0      :PREPARE CWRD
0DE2-    E5 BB           SBC *DCY     :SUBTRACT DCY
0DE4-    C5 BC           CMP *SUST    :>SUSTAIN?
0DE6-    10 0F           BPL NEXT     :PLACE PARA
0DE8-    A5 BC           LDA *SUST    :CS PARA=SUST
0DEA-    10 0B           BPL NEXT     :PLACE PARA
                    :
                    :   RELEASE ROUTINE
                    :MAKES SURE THAT CURRENT STATE
                    :GLIDE BIT IS SET (NOTE-MAKES
                    :CS NEGATIVE).  SUBTRACTS RELEASE
                    :PARA. FROM CURRENT STATE.  IF
                    :RESULT >0, MAKES CS & CWRD =80
                    :
0DEC-    38         RELS SEC          :PREPARE
0DED-    09 80           ORA 80       :SET CS GLIDE
0DEF-    E5 BD           SBC *RLS     :SUBTRACT RLS
0DF1-    30 04           BMI NEXT     :CS<0 -PLACE CS
0DF3-    A0 00           LDY 00       :CS>0 -DONE MAKE
0DF5-    A9 80           LDA 80       :CS=80; CWRD=0
                    :
                    :   NEXT
                    :PLACES CS PARA AND CWRD IN
                    :PROPER CONTROL CHANNEL OUTPUTS
                    :DECREMENTS POINTER (TWICE) AND
                    :IF NOT YET DONE LOOPS FOR MORE
                    :
0DF7-    94 BE      NEXT STY *CWRD,X  :PLACE CONTROL
0DF9-    95 CE           STA *PARM,X  :PLACE CS PARA
0DFB-    CA              DEX          :DECREMENT POINT
0DFC-    CA              DEX          :AND AGAIN
0DFD-    D0 CA           BNE ADSR     :SOME LEFT -LOOP
0DFF-    60         RTN1 RTS          :RETURN
```

LETTERS:

......continued from page 4

format response? Why should the percussionist understand Chaldni patterns or vibration nodes? None of these directly affect the ability to play the instrument, or we would have had to have had three years of physics BEFORE we went into the Junior High Band. However, all of these affect the way in which the instrument is played, taught, and bought. Indeed, the superstars in ANY musical field have such knowledge, if only on a gut-reacting basis. (Be forwarned - having such knowledge does not automatically make you a superstar, either)

In summary, I would like to quote - but I can't remember who said it. Chick Corea, Joe Zawinul, Ken Perrin? It doesn't matter, the idea seems to be pretty universal: "Synthesis is a funny circle. The more I play, the more I want to know. The more I know the better I play. The better I play, the more I want to know.........."

Thanks,
Gary Bannister
Indianapolis

"LOOKING FOR MODIFICATIONS"

Dear Polyphony,

I've just read the February '78 issue and it's great. It's a definite 100% improvement over the Nov. '77 issue. In fact the only thing I learned from Nov. '77 was that an envelope follower could be very useful without external signals to process. But back to the present issue.

The article on the bionic trumpet interested me and usually I find such alternate controlling devices boring. (I'm a guitarist who learned KBDs to play synthesizers, so why can't horn players? Actually, from what I've seen, keyboard virtuosity is a handicap rather than a help with synthesizers, Keith Emerson excluded.)

I was really glad to see the Lab Notes article. The computer is finally going to open doors to PAIA users that are 10 times more useful than other manufacturers' equipment. I use a 5 octave KBD in a custom built case so I can't fit the 8700 in as it is designed to be. Oh well, I was thinking about getting a Commodore PET anyway. Keep up the software!

By far my favorite article was the

continued on following page ....

SPOTLIGHT

By Marvin Jones

SERVICING THE MIDWEST

Regardless of what brand of synthesis gear you use, it is important to have someone you can rely on when you need that gear serviced. Obviously, the best type of repair technician is one who is also actively involved with actually using, designing, and learning about synthesizers. It was for this reason that POLYPHONY began our search for a selection of outstanding regional technicians to help all of you get absolute maximum performance from your equipment. This is the second in our series of recommended regional service centers.

Gary Bannister has been a frequent contributor to Polyphony, so it should come as no surprise to his readers that he is a top notch technician as well. Gary is 25 years old, and lives in Indianapolis where he is putting the final touches on his BS in Music Education from Ball State University. Concurrently with his studies at Ball, Gary began studies with National Radio Institute of Washington, DC. He currently holds a degree in Television-Audio Servicing from NRI. In addition to Gary's synthesizer consultation and repair service, he is a private percussion instructor with the Paul-Mueller Percussion Studios, and performs regularly in the Paul Burton Orchestra -- a small dance combo. Gary has been performing professionally with local bands for over 10 years, so he really understands the musican's plight! Gary has an extensive array of service, synthesis, and recording equipment, as you can see from the picture. Gary has built ALL of his equipment, either from kit or from scratch, except for the equalizer shown and, of course, the tape deck. Gary expressed an interest in helping experimenters de-bug their homebrew projects or projects out of magazines or books. This should be an invaluable aid to those of you who are just getting started and don't know where to turn.

So, whether you are a beginner or a pro, POLYPHONY highly recommends the consultation and service work of Gary Bannister to the midwest audience. Gary's address and phone number will be listed in every issue for handy reference in the Local Happenings column. ■

LETTERS ..... continued

4720 modification layout. (I love modification articles. How about one for a 4730 24db per octave roll-off rate - a la Moog?) (Anyone else interested - Marvin) The night I recieved the issue, I'd made all internal modifications to one of my 4720's with good results. The author beat me to the draw with my own 4 waveform LFO design. I was going to use an EXAR 2066 function generator clip with all waveforms built in. C'est la guerre! His idea is a lot better.

I'd like to see in Polyphony a common problems and troubleshooting guide. Perhaps two modules per issue. Surely another reader has suggested this. Keep printing Lots of modification articles. If that 4730 mod. is not possible, how about an 8730 VCF kit? I, and I'm sure others, would willingly pay 3 times the 4730 price for that kind of filter.

'Til next issue,
David Mays

P.S. Are there any people interested in a user's group club deep in the heart of Texas? (If so, see Dave's address in LOCAL HAPPENINGS - Marvin)

MISSED SOMETHING?

BACK ISSUES AVAILABLE

Single issues $1.25 each postpaid.
Sets of Four issues $4.00 postpaid.

Address your Back Issue orders to:
POLYPHONY
BACK ISSUES
P. O. Box 20305
Oklahoma City, OK 73156

Write for Polyphony

...We are paying at the rate of $25 per printed page for feature articles including such things as circuit modifications, new circuits, narratives, how-to articles, etc. and $5 for each patch accepted for publication. All material submitted must be original and never before published. Payment is made upon publication. We cannot assume the responsibility for return of manuscripts unless specifically requested and supplied with a stamped, self-addressed envelope for its return. All accepted materials and associated copyrights become the sole property of Polyphony and may not be reproduced without written permission from the publisher.

INDEX - VOLUME III

FEATURES

CONSTRUCTION & MODIFICATIONS

DEPARTMENTS & COLUMNS

REVIEWS

PATCHES

Equipment Exchange

A place for our readers to offer for sale or trade equipment related to music and electronics. Keep listings as brief as possible and enclose $1. 00 for each listing. Persons responding to ads should write directly to the other party. DO NOT write to POLYPHONY. Polyphony is not responsible for any claims made in ads or results of transactions. We reserve the right to refuse or edit any ads submitted.

For Sale: 3750 Programmable Drum Set. $75 postpaid. Assembled, never used, factory checked. Jim Kowalski, 405 Aspen St. NW, Washington, D.C. 20012.

For Sale: 2720R, excellent condition. $190. Also, SWTPC Theremin in custom case. $15. Send SASE for more complete info. Craig Forman, 3045 Bavarian Dr. W., Apt. 413, Indianapolis, Ind. 46236.

For Sale: Assembled Pygmy amp, $35 postpaid. Works fine. Gurdy Leete, 37 Prince Royal Passage, Corte Madera, Cal 94925.

For Sale: Custom 4700-type system. Fully assembled and tested, less than 1 year old. Contains: 2720-5, -11, -12, 4710, 4711, 4740, 3- 4770, 2- 4780, 4730, 2- 4761, 8782, 8780. $800 or offer. David Smith, 2448 SE Caruthers St., Portland, OR 97214.

For Sale or Trade: Complete SC/MP II microprocessor chip set, sorry - no data, $20. SWTPC CT-1024 terminal in homemade case with UART board and data, needs some work, $160. Would like to trade for a working GNOME. Eric Barbour, CU Box 8303, Flagstaff, AZ 86011.

Must Sell - Need Money: Paia 4700S. Needs calibration and testing. Make offer. Ken Anderson, 12823 Luiseno St., Poway, CA 92064, (714) 485-1651.

Must Sell: 4700S, completely assembled. Excellent condition throughout. 2 years old. $550. In northern Cal., call Jeff at (415) 328-0395. In southern Cal., (714) 452-7618. Can deliver between San Diego and San Francisco.

PAIA Electronics has an opening for a Bench Technician. Contact: Marvin Jones - (405) 843-9626.

Local Happenings

Synthesizer Consultation and Repair Services are available from:

Synthetic Sound Labs
Doug Slocum
1 Gale Road
Bricktown, NJ 08723
(201) 477-3319

Gary Bannister
7208 New Augusta Rd.
Indianapolis, IN 46268
(317) 293-0606

The following people wish to contact users in their area ... etc.

Bob Makar
41 St. James
Buffalo, NY 14225
(716) 897-5003

David Mays
212 Elm St.
Hurst, TX 76053
(817) 268-5009

Ron Jones
780 N. Adele #D
Orange, CA 92667

PAiA 4700/G Modular Versatility Comes to Guitar Processing

The 4700/G- a guitar processor and more!

MORE...... because it's not just limited to guitar. Any instrument that can be recorded can be processed with the 4700/G.

MORE because modularity takes you that extra step beyond- to the versatility possible only with a patchable system.

MORE because the well-illustrated assembly and user's manual for each module plus the detailed system guide continue the PAIA tradition of making each kit both a low-cost alternative to traditional products an a valuable learning experience.

MORE because of the extensive synthesizer module complement: Envelope Follower/ Trigger, Voltage Controlled Multi-Modal State Variable Filter, Envelope Generator, Inverting Buffer, Control Oscillator/Noise Source, Reverb, Power Supply and Patch Bay, Patch Cords and PAIA/DeArmond foot pedal.

LESS......use the handy order coupon at right and you get an even better deal---- we'll pay the shipping!

PAiA ELECTRONICS, INC. • 1020 W. Wilshire Blvd. • Oklahoma City, OK 73116

I could use a little more....... and the little less is really appealing. Send my 4700/G to:

Name____________________

Address____________________

City__________ State______ Zip______

( ) $219.99 enclosed
( ) Please charge to my ( ) BAC/VISA ( ) Mastercharge
Card Number__________ Expiration Date______
( ) I can't buy today, but please send your latest catalog.