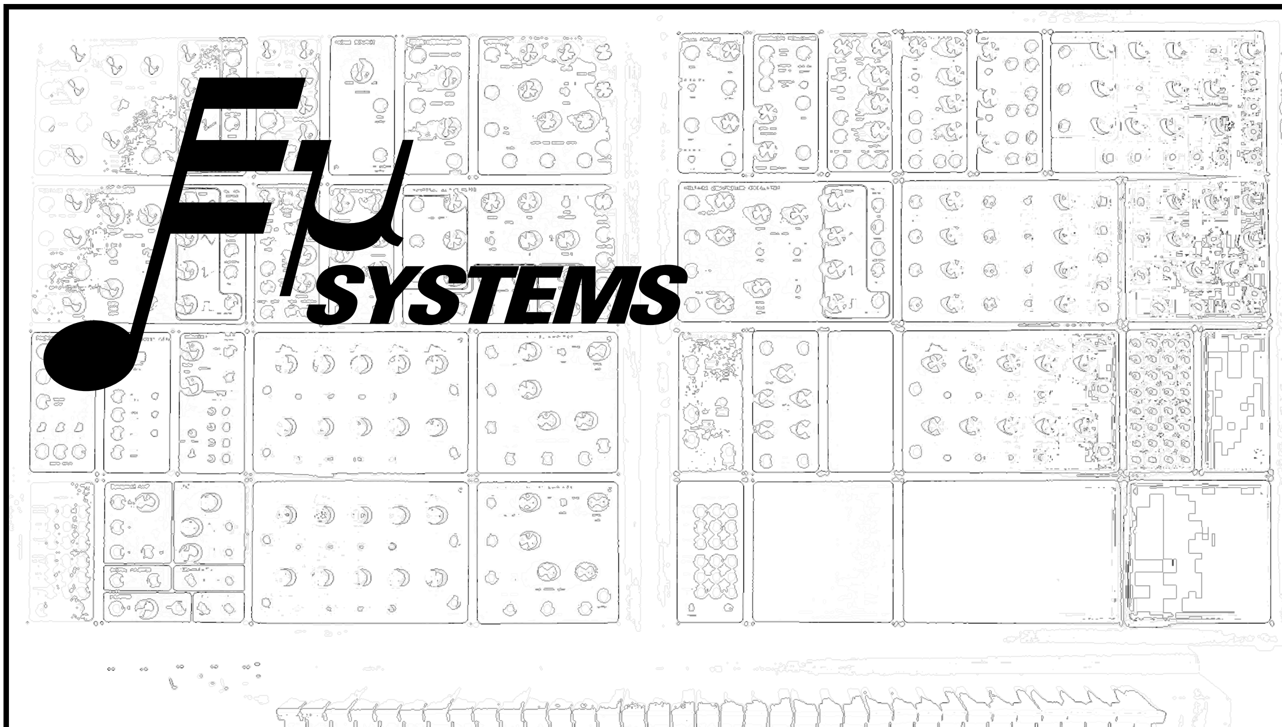
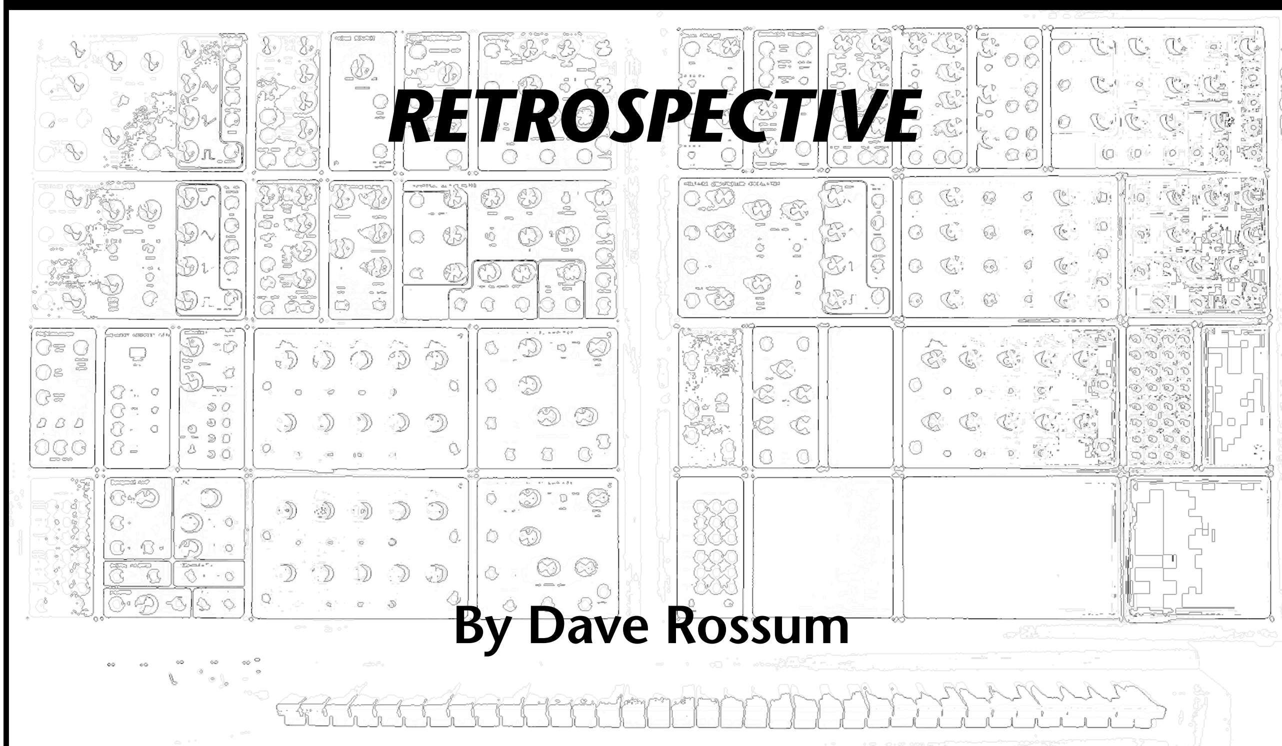


E-mu SYSTEMS

MODULAR SYNTHESIZER

OPERATION MANUAL



RETROSPECTIVE

By Dave Rossum

The Eμ Systems Modular Synthesizer Owners Manual *RETROSPECTIVE*

By Dave Rossum

Edited by Riley Smith

A Retropective…

The operation manual for the Eμ modular system was never officially published by Eμ Systems for reasons unknown. This retrospective, compiled almost 30 years after it was originally written, is far more than just a time capsule to the golden age of analog synthesizers. It is an advanced course in the art of sound synthesis.

This manual should be of interest to anyone interested in building or playing analog modular synthesizers, or those wishing to expand their knowledge of synthesizer programming in general. Although electronic music machines have changed dramatically over the years, many of the underlying concepts of the modular system remain.

This retrospective manual was compiled in the same spirit that created the Eμ modular system— it was a “labor of love”. We sincerely hope you love reading about it.

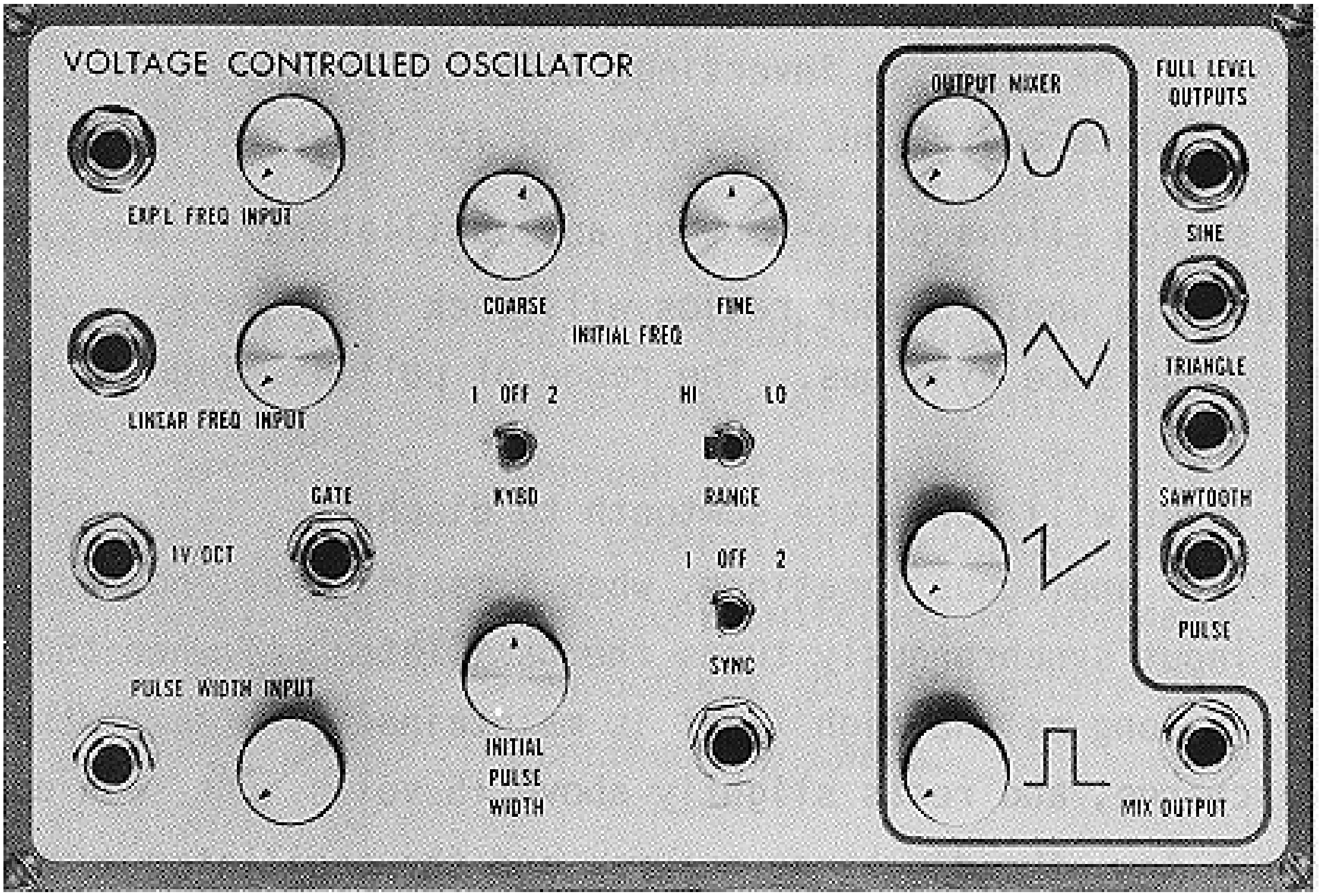
Thanks Dave, for permission to finally release it.

R. S.

Table of Contents

Voltage Controlled Oscillator 4
Sawtooth/Pulse Voltage Controlled Oscillator 9
Voltage Controlled Amplifier 10
Quad Voltage Controlled Amplifier 14
Voltage Controlled Lowpass Filter 16
Voltage Controlled Highpass Filter 20
Universal Active Filter 24
Resonant Filter 28
Lag Processor 29
Dual Transient Generator 31
Voltage Controlled Transient Generator Input Unit 36
Noise Generator 38
Sample and Hold 40
Envelope Follower 43
Ring Modulator 46
Quad Inverter 48
Dual Preamp 50
Dual Reverb 51
Voltage Controlled Clock 52
Hex Digital Inverter 54
Triple Or Gate 55
Triple Latch 56
Dual One-Shot 58
Analog Switch 59
Eight Position Address Generator 60
Voltage Source Output Unit 62
Digital Sequencer 65
  Memory Address Generator 65
  Programmer 66
  Memory Module 68
  Tape Interface 74
4000 Monophonic Keyboard 76
4050 Polyphonic Keyboard 80
Applications - Speech Synthesis 85

Voltage Controlled Oscillator



Function

The VCO is, of course, the vital organ of a synthesizer, being the signal source from which most tones are derived. A few words on voltage control of pitch would be helpful for starters. To say an oscillator is voltage controlled is to say that its frequency is somehow determined by a control voltage—it doesn't say what this function is.

Some synthesizer VCO's have linear frequency control, which means if the control voltage increases by 1 volt, the oscillator frequency increases by a certain number of Hz, say, 1V/1000 Hz. In other words, the pitch increases linearly with voltage. Most synthesizers, including Eμ's, have exponential frequency control, which means that a 1 volt increase in control voltage multiplies the original frequency by some factor. If this multiplicative factor happens to be 2, the exponential control is said to be 1V/octave (since a doubling of frequency is equivalent to raising the pitch one octave). The distinction between linear and exponential voltage control is critical. Consider two VCO's harmonizing an octave apart at 1000 and 2000 Hz.

With linear VCO's (at 1V/1000 Hz) the addition of 1V to each control input increases the frequencies to 2000 and 3000 Hz. These are no longer an octave apart; linear oscillators won't "track" with each other.

On the other hand, two exponential VCO's (1V/octave) at 1000 and 2000 Hz respond to a 1V increase by shifting to 2000 and 4000 Hz, keeping the harmony the same.

The Eμ 1200 VCO submodule, when properly trimmed by the Volts/octave trimmer on the module board, responds at precisely 1V/octave to its incoming control voltage. The control voltage which the submodule sees is the sum of several voltages coming from different places on the front panel. The range switch and the coarse and fine initial frequency pots contribute to the “summing node”, as it is called, and so do the three frequency control inputs at the upper left. The lowest of these three has no attenuator, so any voltage applied to that input has a precise 1V/octave effect on the frequency. The upper two inputs are both attenuable; 1 volt applied at one of these inputs raises the frequency roughly one octave if the attenuator is completely clockwise. When the attenuator is turned down somewhat, the same 1V input may raise the pitch by only a fifth. It’s control function would then be 1V per 7/12 octave, or about 1.71 V/octave. In other words, the sensitivity of pitch to control voltage can be one octave or less per volt. Frequency control voltages can come from an infinitude of different sources. One such voltage source is a keyboard (q.v. 4000 Keyboard), which gives a different voltage for each key depressed.

Our keyboard, and most other synthesizer keyboards, puts our zero volts for the lowest key, 1/12 V for the second key, 2/12V for the third, up to 1V for the first octave. An octave above that is 2V the next octave is 3V, up to, 5V for the highest key. Patched into the 1V/octave input on a VCO, the pitch of the VCO thus follows the keyboard in tempered tuning.

This is such a commonly used patch that it is preset on our VCO’s through the KYBD switch in “KYBD 1”. The VCO is pre-patched through the KYBD 1 bus in the ribbon cable to the KYBD 1 voltage of a keyboard, if present. Oscillators on “KYBD 2” are similarly pre-patched to the KYBD 2 bus, which may also have an associated keyboard. Assuming a keyboard is present on the KYBD 1 bus, its control voltage also appears on the jacks labeled “Control Voltage 1” on the power supply front panel. It is important to understand that these jacks, as well as the gate and trigger jacks, merely tap into the busses running throughout the machine and can serve either as outputs from or inputs to these busses. Thus, if a keyboard is in use on the KYBD 1 bus, a control voltage from another source can be sent to VCO’s not needed for the keyboard by using the KYBD 2 bus; this is accomplished by patching the DTG into the power supply “control voltage 2” jack, and switching the desired VCO’s to the KYBD 2 position. If the DTG were patched into the “Control Voltage 1” jacks, it would be competing with the keyboard for control of the bus. Since the keyboard has a very low output impedance compared to the 1 Ohm output impedance of the DTG, the keyboard would completely overpower the DTG and is the only voltage source which would appear on the bus.

Since the effective control voltage to a VCO submodule is the sum of initial frequency settings, keyboard control voltage, and the three F.M. inputs, several controls can be happening simultaneously. For example, you could play an arpeggio on the keyboard, have a low frequency sine wave on one attenuable F.M. input to give a shallow vibrato, and a clocked VSOU on another F.M. input causing the arpeggio to be transposed in any desired melody and rhythm. This still leaves one F.M. input unused, which, if you patch to the slow random noise output, turns your masterpiece into total garbage. (*Don’t feel too bad if you like it.)*

The 2200 VCO simultaneously outputs sine, triangle, sawtooth and pulse waveforms. These are available from the full-level outputs at 10V peak-to-peak for sine, triangle and pulse, and zero to 5V for sawtooth. A mix of the four waveforms is available at the mix output, to provide greater selection of timbres. The mix output is inverted, making it possible to obtain one of the basic waveforms and its inverse from the same VCO.

This allows two VCA’s to be used as a voltage-controlled panner, or a VCA control to go up while a VCF control goes down, or anything else you can think of.

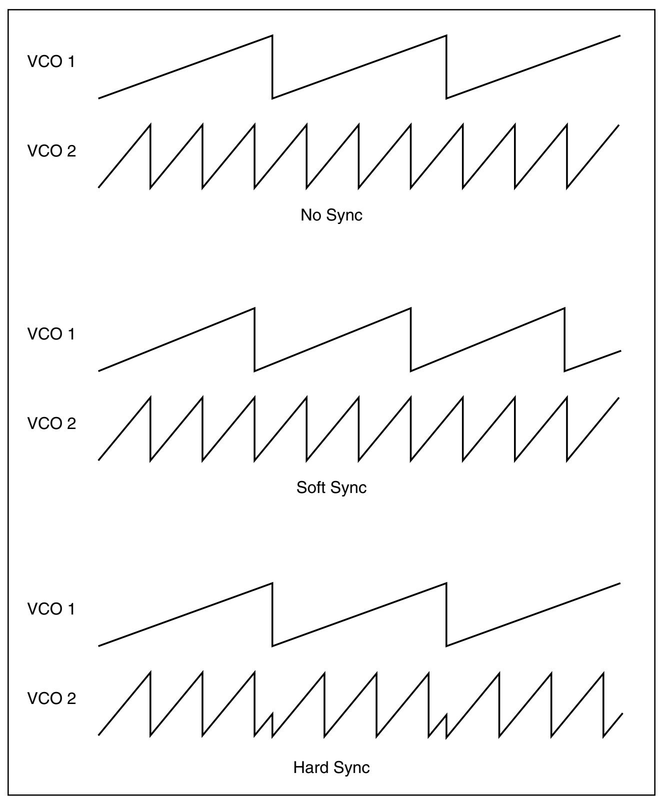
As can be seen from the panel drawings, the phase relation among the waveforms is such that the sine trough, the triangle peak, the sawtooth fall and the pulse rising edge are simultaneous. The pulse falling edge can occur anywhere from 0% to 100% of the way through the cycle. This is determined by another summing node which includes the initial pulse width pot and the attenuable pulse width input. With the attenuator fully clockwise, the input increases the pulse width

10% per volt. If the effective pulse width control voltage (pot plus attenuated input) is sufficiently negative, the pulse width goes to zero turning the pulse wave into a -5V DC level. Similarly, a large positive voltage gives a +5V level. These DC levels are useful for voltage sources, and since they are not audible, they provide a good way to shut off the slight capacitive pulse-wave buzz which leaks through the output mixer even when all the waveforms are supposedly off. (Simply turn the initial pulse width pot full left or right). If you find the pulse output of a VCO dead, or cutting out intermittently, check the pulse width modulation to determine whether the pulse width is pinning at one of these extremes. Pulse width modulation is a very useful tool for achieving smooth and interesting timbral changes. Try it with a transient generator, or as a source of vibrato. Notice that P.W.M. has no effect on the other waveforms. Actually, the VCO submodule generates sawtooth first, then the wave converter submodule inverts half the cycle to give triangle and distorts the triangle to give sine (as such, sine is not completely pure, but can be trimmed to be nearly so and filtered to be perfectly so. Pulse is obtained in the WC submodule by watching the sawtooth output, and giving a +5V level when the ramp is below a critical voltage, and -5V when above. The pulse width summing node simply determines the value of the critical voltage.

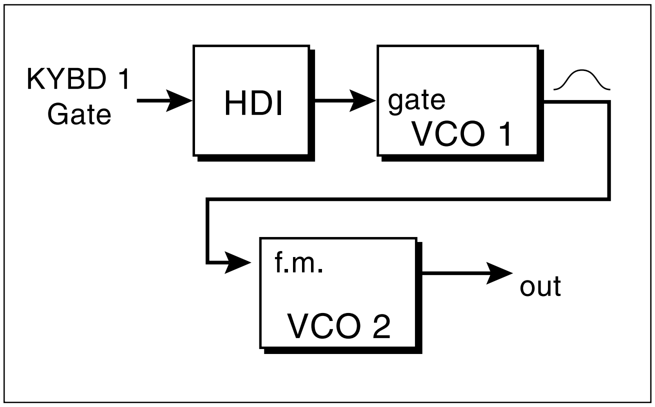
The sync switch allows the VCO to be connected to one of two sync busses running throughout the machine. Oscillators which are connected to the same sync bus and all somewhat less than a semitone apart in pitch will all be pulled, or "synced" to precisely the same frequency (equal to the highest frequency among them). In addition, an oscillator within a semitone of any harmonic of another oscillator will be pulled up if its frequency is lower than the harmonic, or will pull up the other if higher than the harmonic. This is essential for keeping oscillators in tune with one another or in constant phase relation. If the so called "chorale effect" is desired, the sync can be turned off.

Syncing is accomplished by the use of sync pulses; when the sawtooth of an oscillator falls, a sync pulse is issued on any engaged sync bus. Any other oscillator an that sync bus already close to discharge will discharge its sawtooth when a sync pulse is seen on the bus. Thus, with all the oscillators both issuing and observing each others' sync pulses, all are pulled to the rate of the fastest. This is known as "soft" sync, since the slower oscillator will only discharge its sawtooth if it was already close to doing so anyway. "Hard" sync causes the sawtooth of a slave oscillator to discharge whenever the master oscillator does. This is equivalent to soft sync when the slave is lower in frequency than the master, but results in deep harmonic sidebands and consequent-bizarre timbral changes when the slave is higher than the master.

In any case, hard sync is incapable of syncing to harmonics, as can be seen in the drawing below, which describes the operation of both types of sync:



The gate input is a digital input (q.v.) sensitive to two states: high (>2.5V) and low (<2.5V). When the gate is low, the oscillator runs normally. When high, the sawtooth discharge is prevented, so sine and pulse stay at -5V, while sawtooth and triangle stay at +5V At the transition from low to high, the last cycle begun is allowed to finish, and when the gate falls, oscillation immediately commences at the beginning of a cycle. These transition details are easiest to visualize when the gated oscillator is running sub-audio. Consider the following patch:



VCO 1, the gated VCO, is normally off, since the hex digital inverter (q.v.) outputs a digital "one" to its gate input. When any key is depressed, the output of the HDI goes to a digital zero, and VCO 1 synchronously begins a slow sine wave output at the sine jack. This is used to modulate the frequency of VCO 2, giving a precisely reproducible trill at VCO 2's output. If the key is just tapped, exactly one cycle of the trill will occur, then VCO 2 will return to its original frequency.

If the low octave gate is also used to gate a transient generator which controls a VCA in the signal path (or if the LOG controls the VCA directly), and suitable warbling F.M. is applied to VCO 2's other F.M. inputs, a patch is obtained which gives a peculiar trilling bird chirp at the mere tap of a key!

Although more interesting uses of gate will be discussed in the next section, the synchronous initiation of one cycle of a sine wave is not obviously possible any other way. (Correction— the 2500 VCC can also be gated synchronously, and its pulse output can be filtered to give a sine. There's always another way, given a big enough system!)

Through the use of various types of filters on the mix output of the VCO, a very wide range of timbres can be obtained. Most of the voices we patch here for traditional music are just that simple.

However, there are other less obvious patches which give even more complex timbres. This section describes two of these patches which involve only VCO's, audio-rate F.M. and audio-rate synchronous gating. Subsequent filtering of these additional waveforms is, of course, always possible.

At sub-audio rates, shallow F.M. is called "vibrato". When a vibrato patch is set up through an attenuable F.M. input, the attenuator is controlling the "modulation index" of the F.M. Turning the attenuator clockwise causes the extremes of the modulated VCO's frequency to get further apart (the modulation index increases). The frequency of the VCO doing the modulating is called, oddly enough, the "modulation frequency", and that of the VCO being modulated (when the modulation is off) is the "carrier" frequency. Notice in the vibrato patch that as the modulation frequency increases into the audio range, the vibrato effect dwindles, and a different effect, easy to hear but hard to describe, takes its place. Turn up the modulation index to give a greater effect.
This is asynchronous audio-rate F.M. It is somewhat similar in sound to the effect of a ring modulator (which produces audio-rate amplitude modulation since both types of modulation give non-harmonic sidebands.

See what happens to the sound as the modulation frequency, carrier frequency, and modulation index are varied separately. Switch both VCO's to the same sync bus and notice how a steady tone replaces the sideband sound.

With the syncing in effect, modulator and carrier are locked in some rational phase relationship (for example, carrier - 5/2 x modulator), making the output a periodic waveform and therefore a steady tone.

Another way to say this is that all non-harmonic sidebands are eliminated, leaving only those sidebands which are harmonically related. With the syncing on, sweep the modulation and carrier frequencies independently as before. Notice that their respective effects on the tone are now much more clearly defined; the apparent pitch (the fundamental) of the F.M. output is actually the modulation frequency, since a full cycle of modulation is required before the F.M. waveform repeats itself.

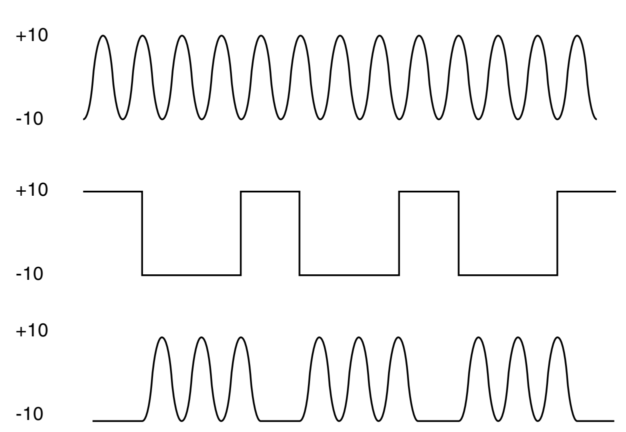
The carrier frequency, on the other hand, has a profound effect on the timbre of the output, introducing more high harmonics as its frequency goes up. If both VCO's are switched so as to track a keyboard, the F.M. output will keep the same waveform regardless of pitch. Some readjustment of frequencies may be necessary if the sync is operating close to the limit of its pulling range. This is evidenced by fluttering on certain keys, and sections of the keyboard being out of tune with respect to other sections. It is always correctable with the fine tuning on the VCO's. If you need to duplicate the same F.M. timbre in two voices, the tuning can be difficult. Tune the modulation VCO's together, then tune the carrier VCO's together (with the modulation off, of course), then turn up the modulation indices until the desired timbre is achieved in each voice. Notice that you only have two sync busses, so you'll have to use sync 1 on one voice and sync 2 on the other. More than two voices are not possible with syncing unless special sync busses are installed.

An interesting special case of F.M. is something that could be called "auto-F.M.", or "feed-back F.M." Patch the full-level sine output of a VCO back into an attenuable F.M. input of the same VCO. Listen to the mix output (any waveform will demonstrate the effect, though sine may show more extreme timbral changes). As the attenuator is turned up, the fundamental goes down and the timbre gets more razzy. It is difficult to explain these changes precisely without plenty of math, except to say that the output at the full-level sine jack is the one and only waveform which will give itself at that jack as a result of the functions performed on it by the VCO and WC submodules.

The output waveforms which appear at the other jacks will be other functions of the same input waveform. In this example, where the sine jack (no longer a sine wave) does the auto-F.M., it may be fairly obvious that within one cycle of the resultant waveform, the VCO will sometimes be running faster than the original frequency and sometimes slower. It will, in fact, spend more time running slower than it will running faster since the output-voltage feeds back to control the oscillator frequency. The result is a lowering of pitch with increased modulation index. (This can be compensated by adjusting the initial frequency setting). Note that when sawtooth is the modulator, the pitch goes up with modulation index— this is because sawtooth is always positive. As for the timbral effects of auto-F.M., they can probably be best understood by seeing the output waveforms on a scope, if available. Since any of the four waveforms can be used as modulator, any can be used in the output mix, and the modulation index is variable, an infinite number of new timbres is made available.

An effect somewhat related to audio-rate F.M. is audio-rate synchronous gating. This has already been discussed for sub-audio rates; the gate input performs the same function at audio rate. The figure below shows the result of gating an audio rate VCO with the pulse wave from another VCO running at a considerably lower audio rate. Though any waveform with portions over 2.5V will operate the gate, pulse is used here for simplicity.

Synchronous Gating



Since the gating is synchronous, the same number of complete cycles is always present in each burst of the higher frequency, and they always begin and end synchronously. Completely asynchronous gating could be achieved by turning a VCA in the signal path on and off with a pulse wave, but since there would be no phase-locking between the two frequencies, the output would contain non-harmonic sidebands much like asynchronous F.M. The gating and gated VCO's can be made to track each other with the keyboard, thus giving constant timbre, as in the case of F.M.

Since gated oscillators maintain the phase relation automatically, syncing the two oscillators is not necessary, and any number of voices can be patched with identical timbres. (Recall that synced F.M. needs a separate sync bus for each voice, so only two voices are possible).

Though the similarities between F.M. and gating are many, you can hear that each gives unique timbres unavailable with the other. Check out the effects of gating frequency, gated frequency and pulse width on the output pitch and timbre.

Sawtooth/Pulse VCO

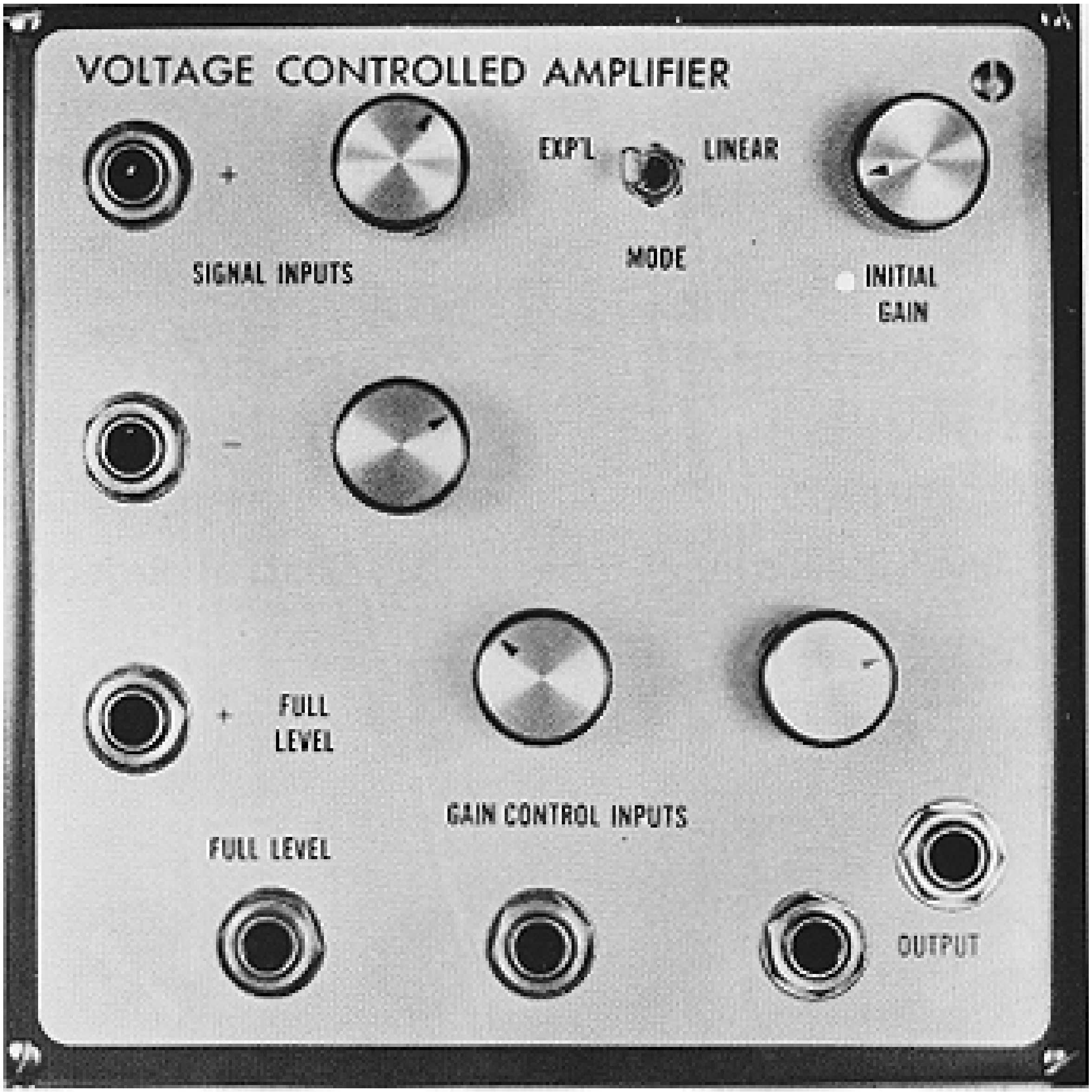
The sawtooth/pulse VCO is a simplified VCO containing a 1200 VCO submodule but no wave converter. It thus outputs only full-level sawtooth and pulse waveforms. It differs in other respects from the 2200 VCO only in the following ways: it has only one attenuable F.M. input, no syncing switch, no hard sync input and no gate. For some purposes, it is fully as useful as a 2200 VCO, particularly since the full-level waveforms can be filtered to give reasonable approximations of the other waveforms, and attenuated at most inputs to other modules.

For control purposes, of course, it is often inadequate. Generally, if a system has several 2200 VCO's, it can have a number of SPVCO's to be used in simpler voices, and much money is saved.

An Apology

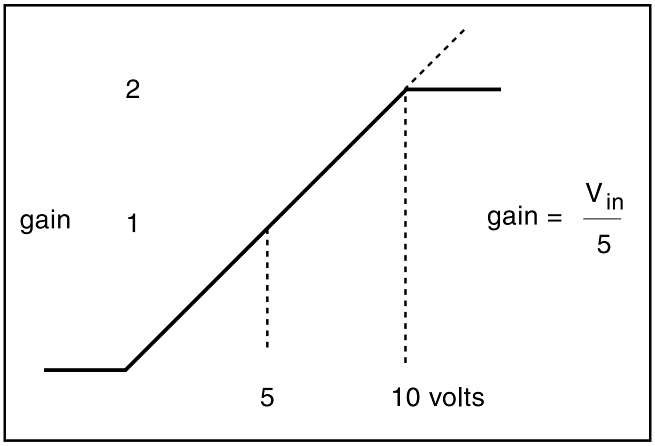
This description of the VCO has been pretty long-winded, mainly because it is such an important module which must be understood well before you can program a synthesizer. Almost every patch discussed in later chapters will include one or more VCO's, and you may be referring back to this chapter often for details of its operation. That's why we put it first.

Voltage Controlled Amplifier



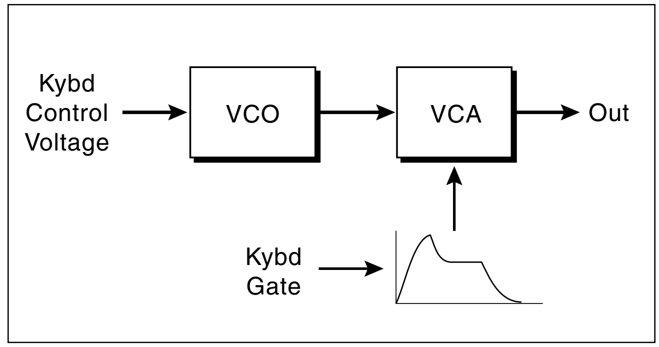
The 2000 VCA is a three-into-one mixer/ amplifier whose gain can be controlled, either linearly, exponentially, or both, by control voltage inputs. It has three signal inputs which are mixed in a summing node before amplification; one of these is full-level, one is attenuable, and one is attenuable and inverting. (The inverting feature will be discussed later in this chapter.) The mixed signal is then amplified by a factor determined by the mode switch, the initial gain pot and the control inputs, and the output appears on the output jack.

The function which determines the multiplicative factor, or gain (gain = Vout /Vin). is selected by the mode switch. The simplest mode to understand is linear. In linear mode, the three control inputs (two attenuable, one full-level) mix in a summing node with a voltage from the initial gain pot. This summing node enters a linear control input on the 1000 VCA submodule. The gain of the VCA increases linearly with respect to a voltage applied on this input, as shown in the graph:



Note that the gain will not exceed 2, or about 6 dB up, regardless of further voltage increase beyond 10V. Since the initial gain pot puts out -5V full left and +5V full right, zero volts is obtained approximately at midrange.
In this position, with no voltage on the control inputs, the VCA has zero gain; in other words, it is turned off. With the pot full right (+5 V), the gain is trimmed to be 1 (Vout=Vin) using the gain trimmer on the module circuit board. Study the linear mode by patching various control sources into the control inputs. A sub-audio sine or triangle wave applied to an attenuable input can give sub-audio amplitude modulation, or “tremolo”. When an audio-rate signal is applied to the control input, audio-rate A.M. is obtained, which will be discussed in detail later in this chapter. Notice with the signal input entirely off, the control input, if audio, can be heard leaking through to the output of the VCA to a small extent. The reason for this is that the circuitry in the submodule sees a slight offset in the signal input even though the voltage applied at the signal input is zero. The VCA tries to amplify this offset by a gain which is varying at the audio rate applied to the control inputs. The result is a signal with the same frequency appearing at the output of the VCA. The ability of a VCA to prevent this occurrence is called its control rejection; the control rejection of the 1000 VCA submodule can be trimmed on the module circuit board to about 40 dB.

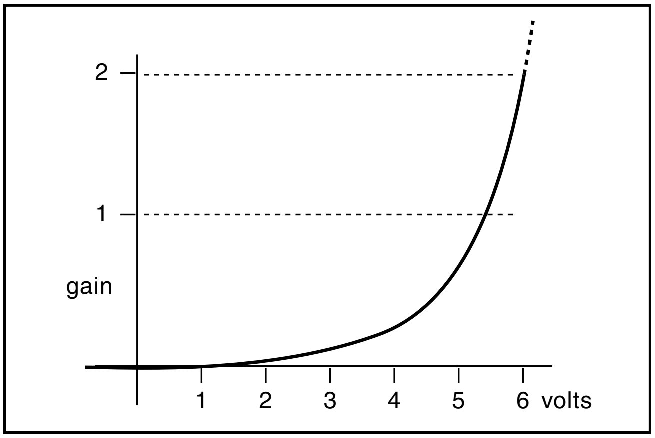
To better understand sub-audio gain control (actually a much more common application of the VCA), consider the following patch:



This is a simple “voice” (q.v.) When a key is depressed,, and its gate is used to initiate a transient from a DTG. This transient, patched into a control input of the VCA, controls the “envelope”, or loudness contour, of the signal from the VCO. For this reason, transient generators are sometimes called envelope generators. (We chose our name to indicate that the transients produced by a DTG can be used for many purposes other than controlling a loudness contour.) In linear mode, the initial gain pot can be adjusted to squelch the output. When the transient is then gated, the amplitude of the VCA output follows the graph of the transient proportionately. Notice the independent effects of the signal attenuator, gain control input attenuator, and initial gain control. When the initial gain control is turned further left than the zero gain point, more and more positive voltage must be applied to the control input to turn on the VCA. It is thereby possible to cut off the lower portions of the transient, which is a different effect from attenuating the transient. Attenuating the signal simply reduces the peak-to-peak voltage of Vin seen by the submodule, resulting in a lower amplitude at the output without affecting the gain.

Apply a sub-audio triangle wave to an attenuable control adjust its rate and the attenuator level to give a good tremolo. Now switch the mode switch to “exponential” and readjust the settings to give a similar tremolo. Notice the difference between the two tremolos. In linear mode, the VCA sounds as if it’s either on or off—your ear is not very able to hear the linear increase in gain once it hears any output at all. This is because the ear responds to volume exponentially, not linearly. In exponential mode, the amplitude increases faster as it goes up, and decreases slower as it comes down. The effect is a "bouncier" tremolo with more apparent variation in its amplitude, even though the actual peak-to-peak variation in the envelope of the output may be the same as in linear mode.

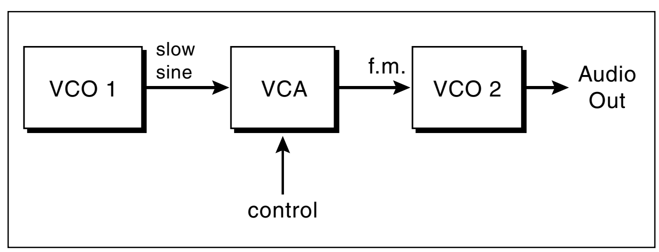
Electronically, exponential mode switches the control input summing node to an exponential control input on the submodule, giving a gain function like this:



The maximum gain available is still limited to 6dB (gain - 2). Notice that the gain can be trimmed ("exponential zero" trimmer) to be unity at 5V input; but, being exponential, it never goes to zero, even at negative control voltages. It can become essentially inaudible, however. Patch a transient generator into a control input in exponential mode. Since the transient already has exponential rises and falls, the gain responds in a doubly exponentiated way. This effect can be undesirable, particularly since the VCA won't go off when the transient falls to zero.

Clearly, then, there are advantages and problems with both linear and exponential amplification. It is possible to have the best of both in the middle switch position—exponential times linear, or "mixed mode". In this position, the full-level input, the left-most attenuable input, and the initial gain pot all sum into the linear input on the submodule-the right-most attenuable input goes into the exponential input. The gain function in this mode is proportional to the product of the functions in the linear and exponential modes. Specifically, gain = Vlin/5 x 10 v exp/2. Thus, when the voltage on the exponential input is zero, the response of the VCA to the pot and the other inputs is identical to the linear mode.

When a positive voltage is present from either the pot or the inputs at the linear input to the submodule, the gain responds exponentially to any changes on the right-most control input. In addition, when the effective linear control voltage is zero, the VCA is off. If a transient is patched into the linear attenuable input, and a tremolo voltage in the exponential input, the VCA will turn off when the transient goes to zero (the advantage of linear mode), and the response to the tremolo voltage will be exponential (the advantage of exponential mode). Notice that in linear mode, the same patch gives a very strange effect, in that the gain sweep of the tremolo is the same at the loud portions of the envelope as at the soft portions. Since the ear does not respond linearly to gain, the tremolo effect sounds shallower when the output is loud, and deeper as the envelope fades away. In mixed mode, the tremolo depth is scaled by the transient, giving a more even-sounding tremolo. In a sense, mixed mode gives a second VCA in the control path; without it, another VCA would be required to perform the multiplication of the linear and exponential inputs. VCA's can be used in control paths as well as signal paths, and the possibilities are impressive. A simple example of using one control input to control another is the following patch for voltage-controlled vibrato:



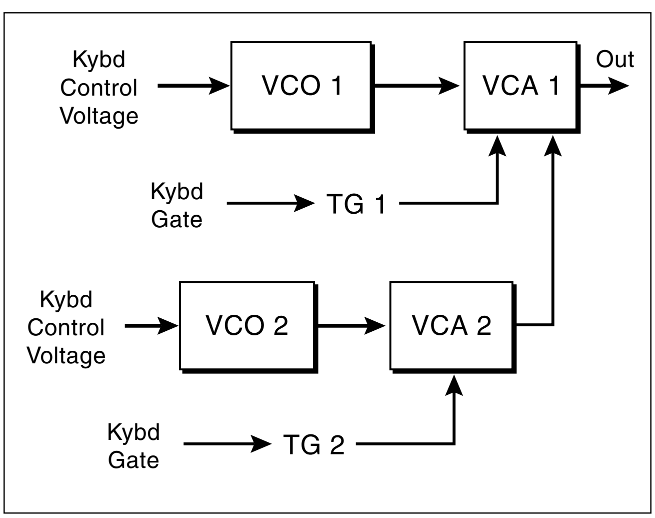
VCO 2 oscillates with a steady tone as long as the VCA is off. As the gain of the VCA increases, the depth of VCO 2's vibrato increases. The control voltage applied to the VCA could come from say, a foot pedal, a transient generator, another slow VCO, etc. It is when the VCA is used in a control path that the inverting input is useful. At audio rates there is no difference between a signal and its inverse, except that they will cancel each other if mixed 1:1 (try this with the two attenuable inputs of a VCA). At sub-audio rates, however, there is a big difference. Compare an ADSR transient and its inverse, or an upward glide and a downward glide.

(A basic note on our general philosophy is that if something is possibly useful and easy to do, we might as well include it. It is true that a quad inverter (q.v.) will perform the same function as the inverting input of a VCA, but you might be able to save the cost of one if you could use a VCA instead, and you'd have the option of voltage controlled gain in addition).

Audio-rate A.M.

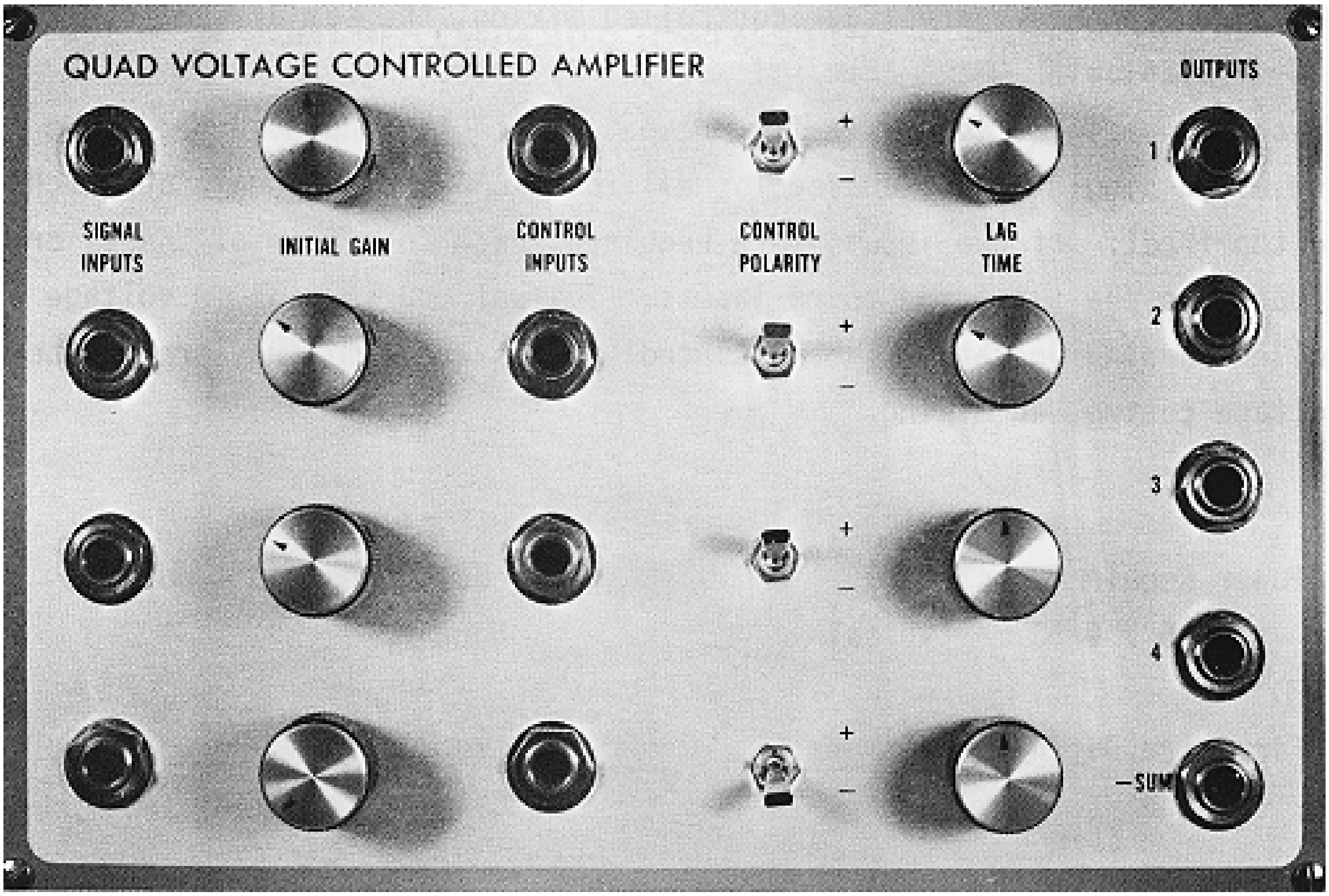
This effect, produced by applying audio-rate signals to both the signal and control inputs of a VCA, is still another way to achieve new timbres. It is closely related to ring modulation (q.v.), except that a ring modulator gives negative gain (gain with inversion) for negative control voltages, while a VCA only goes down to zero gain. Another way to say this is that a VCA is a "two quadrant" amplifier, while a ring modulator is a "four quadrant" amplifier. As in the case of all audio-rate modulations, the carrier and the modulator can be synched to eliminate non-harmonic sidebands. Unlike exponential F.M., however, the pitch of the output is not affected by the modulation index, so voltage control of modulation index, and, hence, of timbre, is a musically sensible patch.

Consider the following:



Since both VCO's track the keyboard, and both transient generators are gated by the keyboard, the evolution of the timbre of a note is the same for all keys. TG 1 is controlling the envelope of the output; TG 2 controls the gain of VCA 2, and hence, the modulation index. As the modulation index increases through the attack, the spectral bandwidth of the output broadens, changing the timbre considerably. As the modulation index falls, the bandwidth is constricted once again, and the output will settle into the unmodulated tone of VCO 1 (if TG 1 has not turned off yet). We just discovered this patch specifically for this chapter, so that may explain the excitement over it!

Quad Voltage Controlled Amplifier

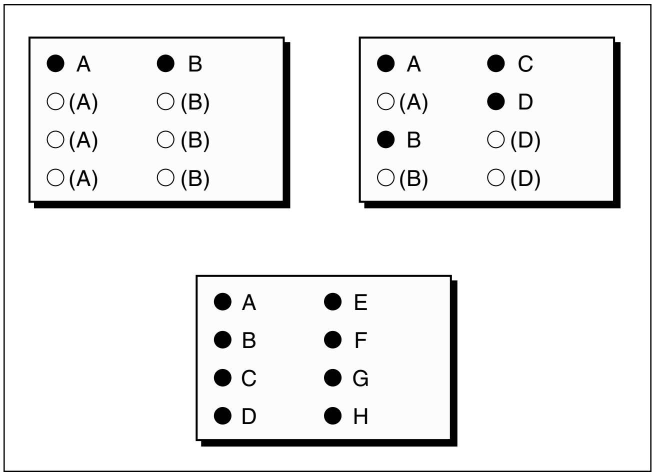


The 2010 Quad VCA contains four simplified VCA's; each channel contains only one signal input and one control input, both non-attenuable. Each channel has two features not found in the 2000 VCA: a "control polarity" switch, and a "lag time" pot. With the control polarity switch in the + position, a positive increase in control voltage on the input jack will produce a linear increase in the gain of that channel. In the - position, the gain would decrease linearly. (The response to the pot is always positive). The lag time pot allows the digital outputs (q.v.) of other modules to be used as control inputs without producing a "pop" in the QVCA output. The attack and cut-off do become somewhat mushed by the introduced lag time, so the pot is provided to adjust the lag time for the proper compromise between popping and mushing.

When continuous signals are used for control inputs, the lag time can be turned completely off (full left) if you want.

In addition to these special features on each channel, the four channels are interconnected in a way that makes the QVCA very useful for voltage controlled switching, mixing, and panning. The tip of each signal input jack is wired on the circuit board to the shunt of the jack below it—thus, a signal applied to channel 1 is automatically put through all the channels below it, unless one of them has a patch cord plugged into its signal input. In that case, the signal on that patch cord is fed to its channel and any open channels below it, etc. The same firm-wiring is done with the control inputs. (Notice that a patch cord plugged into a jack breaks the firm-wiring chain even if it is not used to input a new signal. A patch cord so used is called a "dummy" patch cord: not too useful in the case of the QVCA, but may be in other applications). In addition to the firm-wire feature, the four channels are also mixed; the inverted mix is available at the "-sum" jack.

A few diagrams of patches on the QVCA will clarify its uses considerably. In these diagrams, a filled-in jack indicates the presence of a patch cord, and its signal is identified with a letter. Pre-patched inputs to other channels are shown in parentheses:

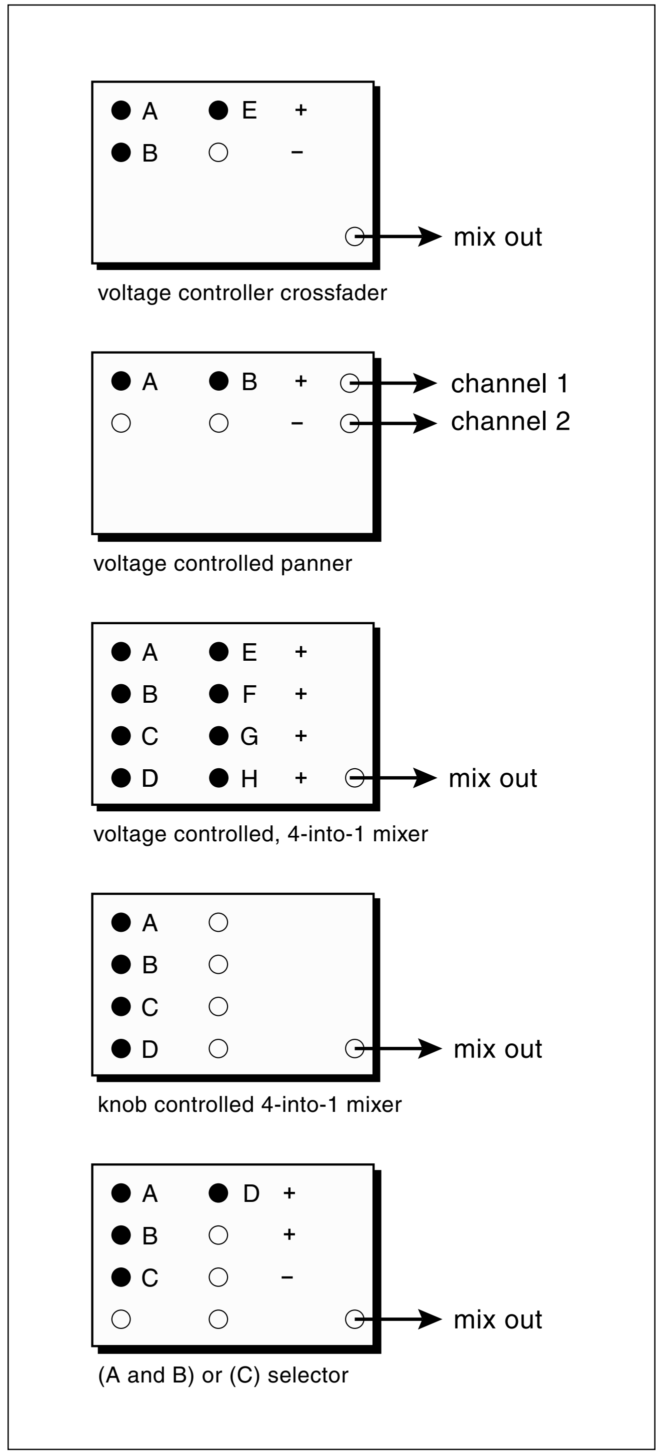


Since a positive voltage on each control input can turn that channel either up or down, depending on the polarity switch, you can imagine a myriad of uses for the QVCA. The drawings at right show a few possibilities:

That last one is a pretty obscure patch, but it illustrates a point. When D is at a positive voltage, the mix output contains A and B only. When D is zero, all channels are shut off (provided, of course, the initial gain pots are set appropriately). As D goes negative, A and B remain off, and C comes up in amplitude. Notice that the effect is different if channel 4's polarity switch is +; in that case. C would be heard at either positive or negative values of D. This could be corrected by patching a dummy patch cord into channel 4's signal or control jack, but, of course, switching its polarity to "-" is easier.

As you can see, the QVCA can do some complicated things; as such, it can be confusing. Watch your polarities and initial gains. Since there are no attenuators, even turning the initial gain pot full left won't shut off a channel if a sufficiently large (>5V) signal is applied to the control input. Switching polarity will often solve this, provided the control input then never goes below -5V.

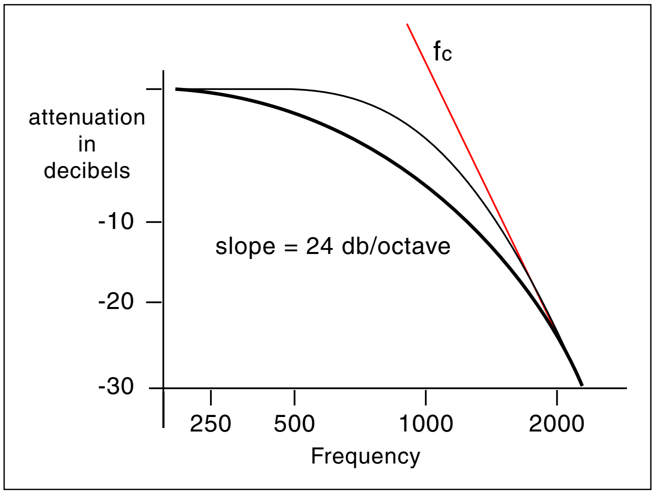
Remember it can also be used for exactly what it is—four separate VCA's.



Voltage Controlled Lowpass Filter

The 2100 VCLPF (VCF for short) is, a lowpass filter whose cut-off frequency is determined by an applied control voltage. Signals enter the 1100 VCLPF submodule via a summing node which includes a full-level input and two attenuable inputs. The total control voltage is another summing node comprised of the initial cut-off frequency pots, the KYBD switch, one full-level input (1V/octave) and two attenuable inputs. Any discussion about filters will necessarily refer to audio spectra, so you should review your "musical physics" if you don't already feel comfortable in frequency space. Speaking ideally, a lowpass filter is a black-box whose gain is unity for all inputs with a frequency less than some cut-off frequency f_c, and zero for all input frequencies greater than f_c. In actuality, the cut-off is not so sharp; the 2100 VCLPF has a cut-off slope of 24 dB per octave. That is, if the cut-off frequency is 1000 Hz, a 2000 Hz input would be attenuated 24 dB. Since the cut-off corner is not sharp, f_c itself experiences some attenuation, equal to 12 dB.

Thus, the attenuation curve for a VCF whose f_c is 1000 Hz is as follows:

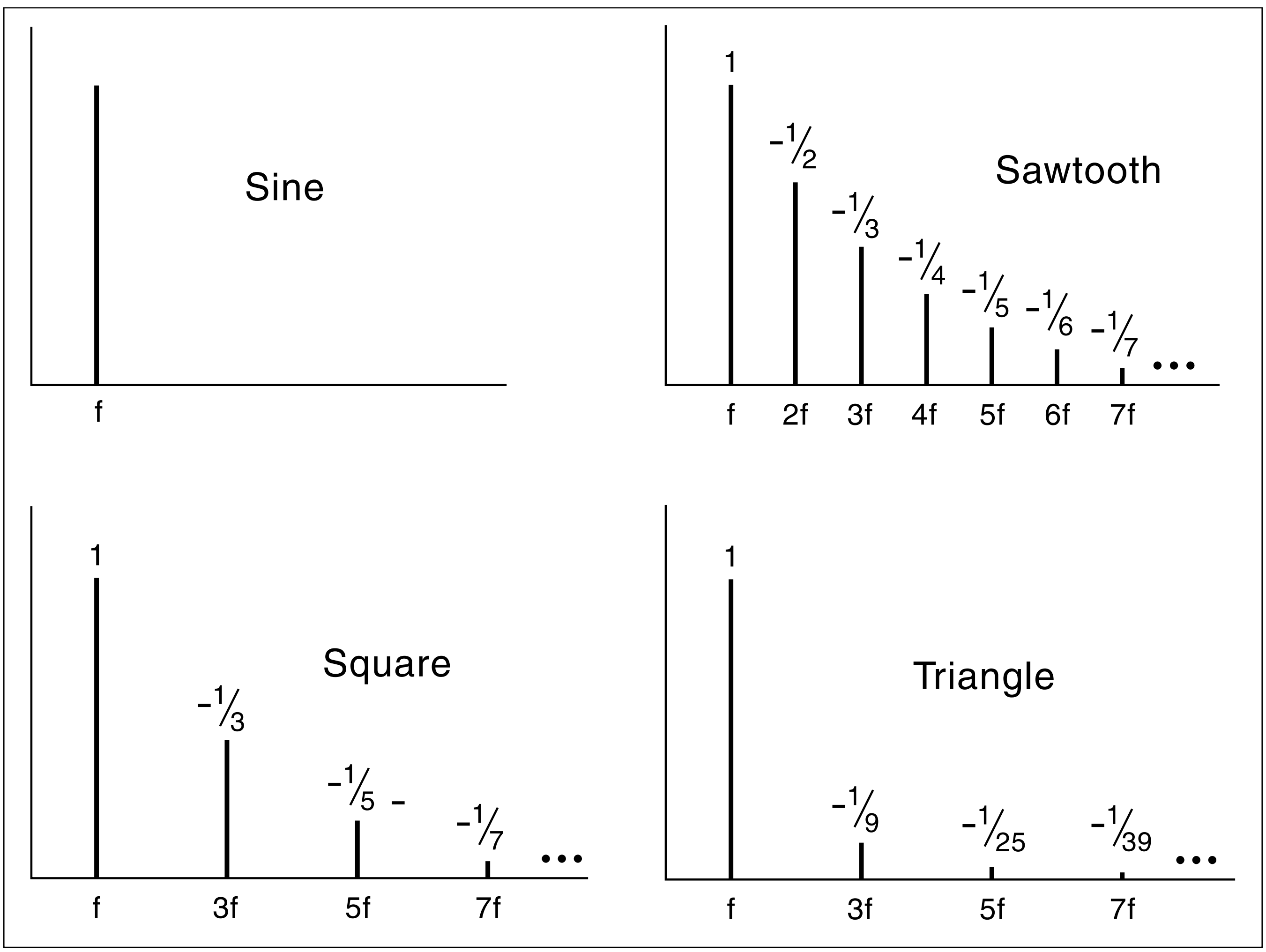


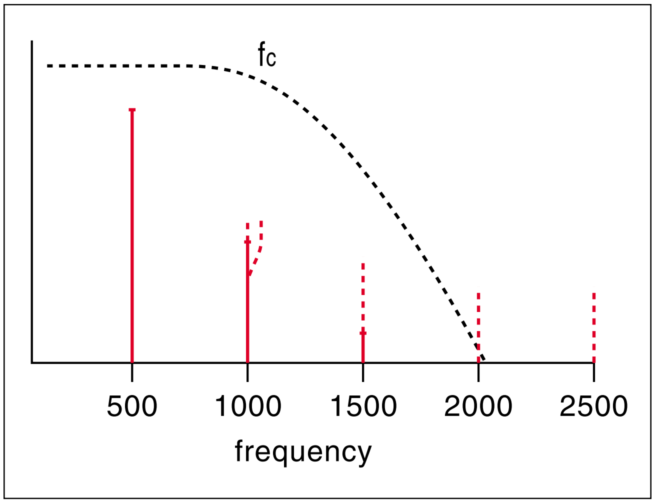
First off, what is this good for? For one thing, filtering of razzy tones will give mellower, more spectrally pure tones, and this can be done to any extent desired. Run a sawtooth wave through a VCF with the Q off (full left), and notice the effect of the initial f_c pots. As f_c is lowered (counter-clockwise), the timbre of the output gets less razzy, approaching and finally reaching the pure sine wave timbre.

At the same time, the overall volume goes down. A look at a few spectra will show why these effects happen. The diagrams below show spectra for several waveforms with fundamental frequency of 500 Hz:

Suppose the 500 Hz sawtooth is passed through a VCF whose f_c is 1000 Hz. The drawing on the following page shows what occurs; the attenuation curve appears in dotted black, the input sawtooth spectrum in dotted red, and the spectrum of the filtered output in solid red.

Notice that all harmonics above the 3rd are heavily attenuated, and the third is attenuated by about 15 dB. Such cutting-out of the high harmonics results in a mellowing of the timbre, and, since high harmonics contribute





a certain amount of power to the input signal, the overall volume goes down when they are filtered out.

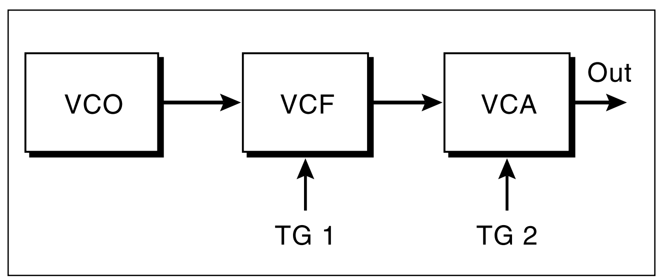
This kind of timbre generation is called "subtractive" synthesis, since an original signal with many harmonics is filtered to give a new signal with few harmonics. In the case of the filtered sawtooth above, the output is spectrally so simple that it could be duplicated almost exactly by an "additive" synthesis.

To do this, you would mix three sine waves from three synched oscillators at 500, 1000, and 1500 Hz in the appropriate ratio. Notice that if the cut-off frequency is brought low enough (f_c = 500 Hz will do), an almost pure sine wave results. This is how sine waves can be obtained using only sawtooth/pulse VCO's. In fact, pulse at 50% duty cycle can be filtered to give an even cleaner sine wave with less attenuation, since it has no 2nd harmonic.

Suppose we run a 250 Hz sawtooth through our VCF with its f_c = 1000 Hz. You can see that the output now will contain the first four harmonics in almost unchanged proportions, and significant levels of the 5th, 6th, and 7th harmonics as well. Clearly this output will have a razzier timbre than the output at 500 Hz. In order to maintain the same timbre over the whole range of desired pitches, we need some way that the value of f_c will track with the signal input frequency. If the filter is voltage controlled, this is easy. Notice the 1V/octave control input and the KYBD switch. These serve much the same function on the VCF as the analogous controls on the VCO; on the VCF they control the value of f_c. Thus, if the VCO and VCF in our example were both tracking a keyboard on one of the KYBD busses, we could use the initial f_c pots to tune the cut-off of the VCF to 1000 Hz when the oscillator is at 250 Hz (on the same key, of course), and they would track together up and down the keyboard, keeping the output timbre the same. (Just how f_c is tuned to a particular value will be discussed in the section on "Q".) If a keyboard is not used, the VCF can be made to track with the VCO by patching a control input to the 1V/octave inputs of both modules.

Notice that the effect of having a VCF track a VCO may also be desirable. Try filtering waveforms other than sawtooth; remember that for any one of the infinite timbres you can input to a VCF, an infinite number of new timbres is available at the output.

Probably a more useful application of the VCF than merely generating new timbres is the ability to voltage control a timbre. Consider the following patch:

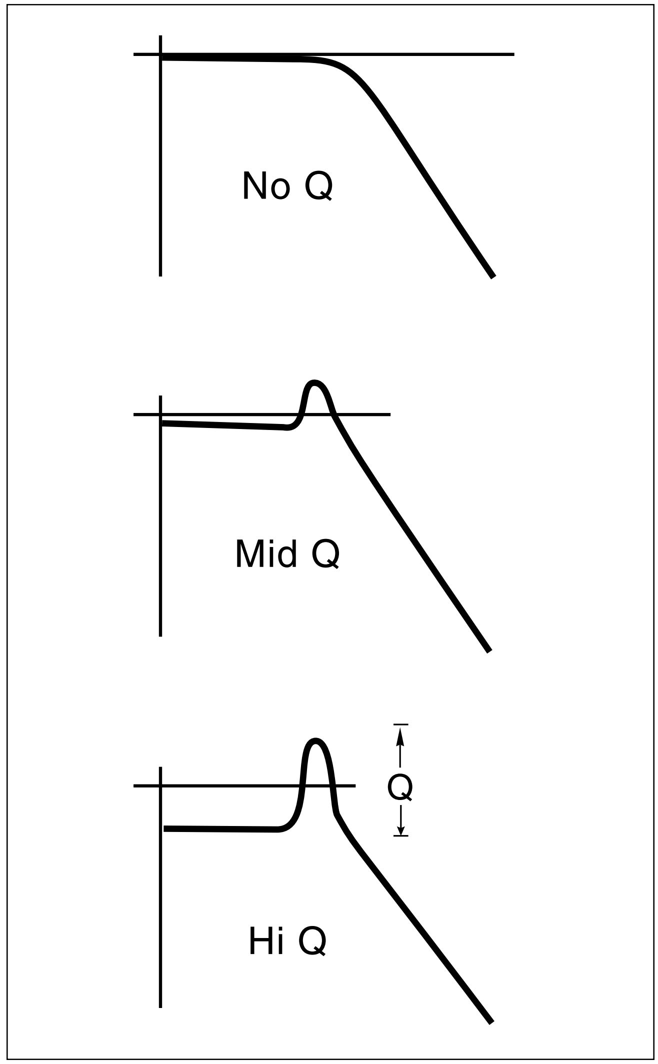


It is assumed in this patch that the VCO, VCF, and TG's are all on one KYBD bus, that the VCA is in linear mode with the initial gain set to squelch the output when the transient is off, and that the TG's are patched into attenuable control inputs. (These assumptions will be made without comment in the future. Exceptions, of course, will be discussed). Notice that the timbre of every newly-attacked note goes through a reproducible evolution as a result of the transient an the VCF. Experiment to see the effects of initial f_c, attenuation of the transient, and transient parameters. A fast attack and decay on the VCF transient give a sound suggestive of a plucked string; a slow attack on the VCA and the VCF sounds like the attack of a brass instrument (brass instruments are characterized by their spectral bandwidth increasing as the volume increases). Try inverting the VCF transient (if you've played in the same orchestras I have, it shouldn't sound familiar). Sub-audio sine or triangle control of a VCF gives an interesting variety of vibrato. Try this on a sine wave input for starters; you might expect the effect of a VCF on a sine wave to be similar to a VCA—you will notice, however, an apparent pitch vibrato as well as the expected tremolo. This is due to the fact that a filter introduces phase changes in frequencies close to f_c (phase change = 180° at f_c for the 2100).

As such, the VCF vibrato is an effect unlike either a true vibrato or a true tremolo. Try synchronous audio-rate f_c -modulation. This is a powerful effect, capable of even more versatile timbral changes than audio-rate A.M. If a transient is applied to one attenuable control input while the modulation frequency is applied at the other, very complex time-evolutions of timbre can be obtained. You can also control the modulation index with a VCA.

The 'Q' pot allows a resonant peak at f_c to be superimposed on the attenuation curve of the VCF. The numerical value of Q is proportional to the increase of gain of the VCF at f_c as a result of this resonant peak. As the Q is increased, the height of the resonant peak is increased, and the rest of the attenuation curve is depressed (the overall volume goes down). The attenuation curves at right with increasing values of Q show the effect.

In frequency space, a filter can be viewed as a module with a frequency- dependent gain. This is the view we have taken so far. At extremely low values of f_c, it is more intelligible to look at the VCF in the time domain—as a module which limits how fast its output voltage will change. Turn the initial f_c pots full left. This value of f_c is around 2-5 Hz. Now apply the control voltage from a keyboard or other source to the signal input and patch the VCF output to a VCO F.M. input. Notice the exponential portamento (q.v.) which results. By applying a control voltage to the f_c control inputs, the rate of the portamento can be voltage controlled. In fact, f_c will go well below 0.1 Hz with -10V on the 1V/octave



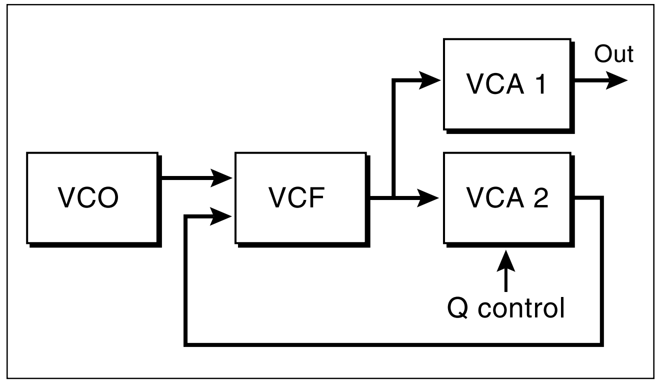
input. Higher values of f_c (around 1 Hz and up) may find many applications whenever sharp voltage transitions need to be smoothed out. See the chapter on the lag processor for more ideas.

In the 2100 VCF, the Q can only be increased to about 20. Past this point, the filter breaks into oscillation. If the Q is Set barely into the oscillation region, the oscillation is purely sinusoidal, and at precisely f_c (as the Q is increased, the distortion increases and the frequency goes down). The fact that this oscillation occurs at f_c makes it possible to set f_c at a particular value; it also makes it possible to trim the full-level control input to precisely 1V/octave. Thus, to assure that several VCF's are set to the same initial f_c, simply turn off the signal inputs, turn up the Q until oscillation first occurs, and zero-beat the VCF's at the desired frequency. Then turn the Q back down. Similarly, to set up the patch used earlier, which cut out all harmonics above the third from a sawtooth input, just tune the VCF to oscillate one octave above the VCO.

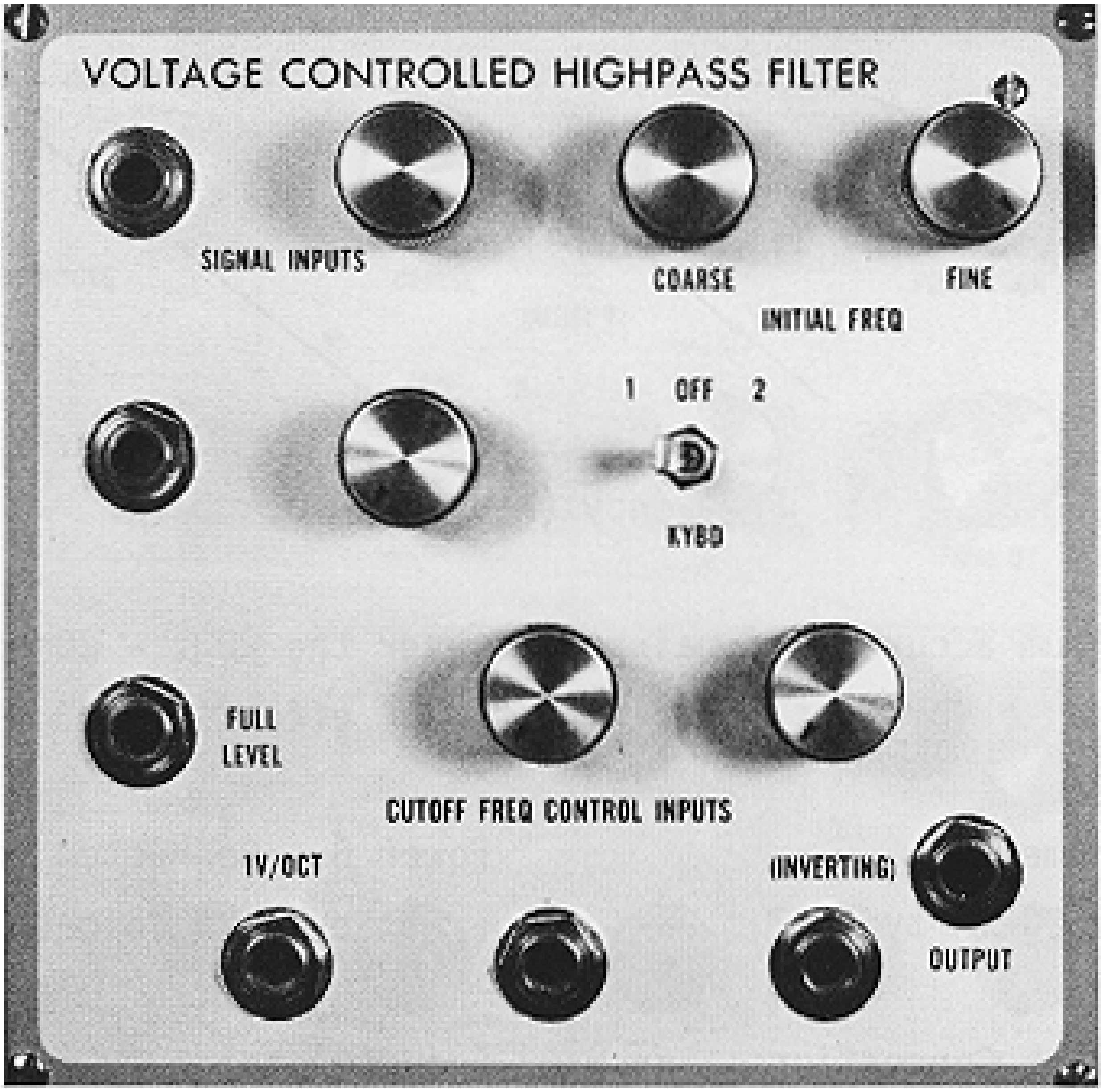
The general use of Q, however, is not to produce oscillation. The presence of a resonant peak at f_c not only makes for a new timbre, but it makes any movement of f_c more obvious to the ear. Without Q, the moving shoulder of the cut-off curve is not directly perceptible—only the rolling in and out of harmonics is noticeable, and when f_c is high compared to the input fundamental, the amplitude of the affected harmonics is not very great. With a high Q, however, sweeping f_c causes every harmonic present to be sequentially boosted to a high level. You can hear this happen by applying a sawtooth wave to the signal input of a VCF, turning the Q up just short of oscillation, and sweeping slowly from low to high f_c with the pot. If you listen closely and keep a light touch on the pot, you can isolate every harmonic up to at least the 10th before they get too close together to separate. When this sweeping is done quickly say, with a transient on a control input, the result is a characteristic “boing” or “wah” sound. (Two in quick succession would give a “wah-wah.”)

An electronic description of Q may be of interest at this point. A portion of the VCF submodule output is fed back, in an inverted sense, to the input circuitry through the Q pot. As the Q is turned up, the amount of this feedback increases. At most frequencies, the result is merely a cut in amplitude, since the feedback is negative. At frequencies close to f_c, however, the phase of the output begins to change relative to that of the input, as was mentioned earlier. At f_c precisely, the phase change is 180°; as such, any signal present at f_c gets inverted once by the filter and again by the feedback loop, thereby feeding back positively on itself and giving a resonance. Oscillation occurs when the gain around the feedback loop reaches unity.

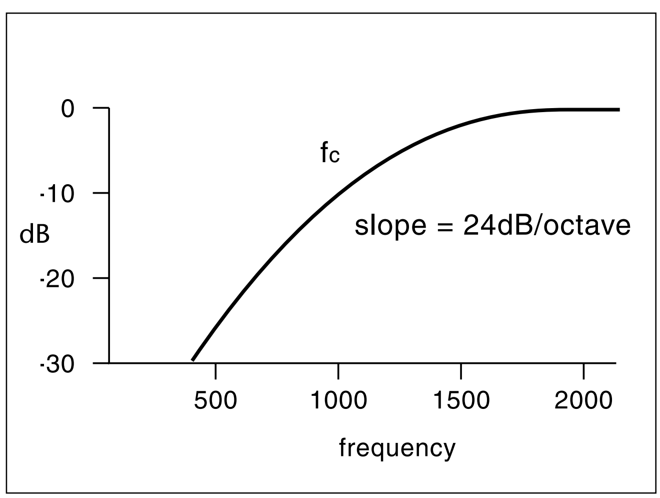
Voltage controlled Q can be obtained with the 2100 VCF by creating this negative feedback externally through a VCA. For higher values of Q, a Dual Preamp (q.v.) can be used to provide more gain in the feedback loop.



Voltage Controlled Highpass Filter

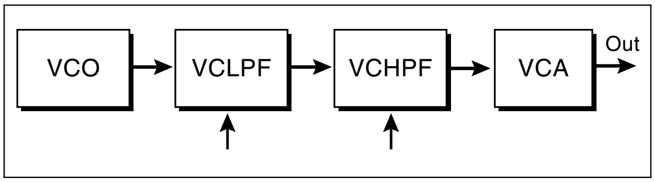


The function of the 2110 Voltage Controlled Highpass Filter is analogous to that of the 2100 VCLPF, except the VCHPF passes frequencies higher than its f_c and cuts out (at 24 dB/octave), frequencies lower than f_c. The VCHPF has an inverting attenuable control input whose use will be discussed later. In addition, the VCHPF does not have the Q control; details of the electronics dictate that a negative feedback loop which will produce Q in the VCLPF would give rise to distortion in the VCHPF. The following is a gain curve for the 2110 VCHPF when f_c = 1000 Hz:



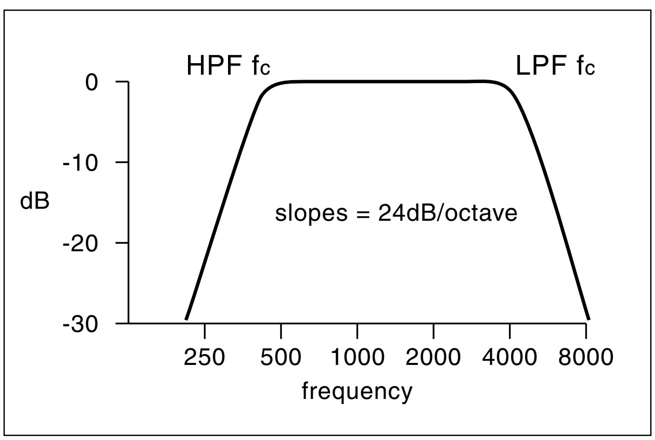
Notice that the HPF can filter out the fundamental frequency of an input while leaving the harmonics—something the LPF can never do. As a result, the timbres produced by an HPF tend to be unnaturally razzy, and the pitch of the output is often not clearly definable.

Though a useful source of new timbres in its own right, the HPF is particularly interesting when used in conjunction with an LPF. Consider an HPF and an LPF in series, as in the following patch.



Any frequency which reaches the VCA must be able to pass through both the LPF and the HPF—thus, we obtain a band-accept filter (a band pass filter is a special case of the band-accept filter where the f_c's of the LPF and the HPF are the same). Clearly, if the f_c of the HPF is above the f_c of the LPF, no signal of any frequency can get through the series pair.

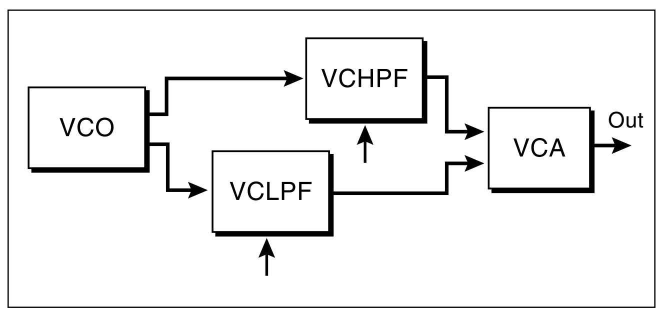
Here is an attenuation curve for a band-accept LPF/HPF pair:



Apply a control voltage simultaneously to a control input of the LPF and to a non-inverting control input of the HPF, and adjust the attenuators to about the same control level. The band-accept filter will now respond to a change in this control voltage by changing the center frequency of the band without affecting the bandwidth. If the band is fairly narrow (LPF f_c is slightly higher than HPF f_c), the effect is much like Q, in that narrow bands are sequentially heard as the center frequency is swept up and down. Experiment with a transient generator as the source for such a control voltage. Notice the effects of HPF f_c, LPF f_c, transient attenuation, and Q from the LPF. With either filter used alone, changes in f_c will affect the overall volume considerably. However, in the case of a band-accept filter, certain settings of the two control attenuators will allow nearly constant volume to be obtained over a wide range of center frequencies.

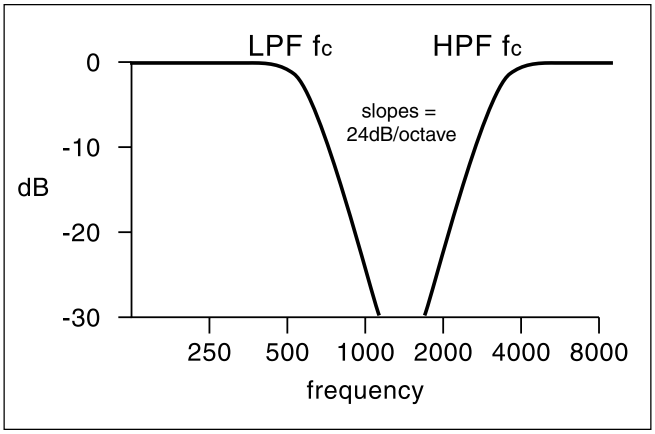
In the same patch, switch the transient on the HPF to the inverting control input and set the attenuators the same as before. The response to a transient is now to vary the bandwidth without appreciably affecting the center frequency, since the HPF shoulder moves down when the LPF shoulder moves up. A variable bandwidth filter such as this does affect the overall volume. The signal will be completely cut out if the control voltage goes too low, since the shoulders will then cross.

An LPF and an HPF can also be patched in parallel, as follows:



Here, the only frequencies which cannot reach the VCA are those which can't pass either the LPF or the HPF—the result is a band-reject filter. In this case, the LPF f_c is lower than the HPF f_c. If vice-versa, all frequencies are passed and some are boosted (this could be called a 'band-boost' filter; not really a filter at all, and we haven't looked into it much (maybe it's just the thing for your application).

A sample band-reject attenuation curve is as follows:

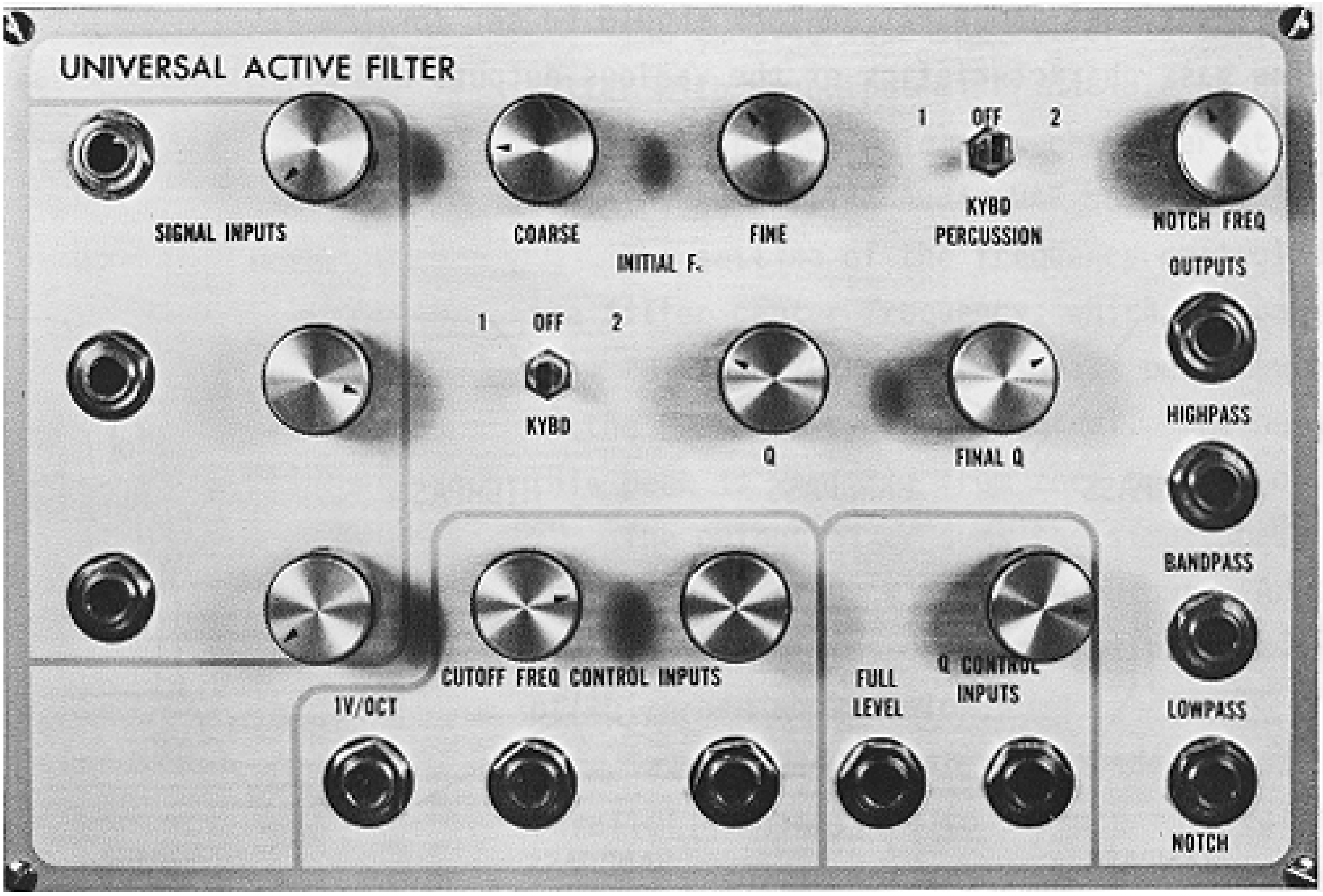


The notch filter is a special case of the band-reject filter where the LPF f_c is the same as the HPF f_c. You might guess that the output at f_c would only be down 6 dB, since f_c is attenuated 3 dB by each filter. However, the LPF introduces negative phase shifts in frequencies close to f_c, while the HPF introduces positive phase shifts; in the case of a notch filter, the phase shifts from the two filters are such that f_c itself is entirely cut out. (More about notch in the UAF chapter!). In any case, true notch can be produced with an LPF and an HPF in parallel by tuning both to the same f_c and inverting one of their outputs before mixing. This inversion occurs automatically if the two attenuable inputs on the VCA are used for the mix, since the lower of the two inverts. The f_c of the HPF is hard to set precisely, since the Q feature is not available. However, in the case of notch, the final effect is so unmistakable that the HPF can simply be tuned until the effect occurs.

Both notch and the more general band-reject function give rise to very interesting timbral changes as the center frequency is swept up and down. The result is much like the “phase-shift/flanger” effects produced by tape delays and commercially available phase-shifters. (Commercial electronic phase-shifters typically produce three notches in the spectrum, while a tape delay can produce a tremendous number of notches in the audio portion of the spectrum). Try using a transient to move the center frequency of a band-reject filter. A sub-audio sine wave used instead will give an effect similar to the phase-shifter vibrato. A variable bandwidth band-reject filter is obtained by using the inverting control input on the HPF. The effect is somewhat obscure, but you may want it someday!

Clearly, very complex spectra and changes in spectra can be obtained when several LPF’s and HPF’s are used in various series/parallel combinations. If you familiarize yourself with these filter functions now, the UAF and resonant filter will be more intelligible.

Universal Active Filter



The 2120 Universal Active Filter is a multi-mode resonating filter with simultaneous highpass, bandpass, lowpass, and notch outputs (all at the same f_c). It has three attenuable signal inputs, two attenuable f_c control inputs and one at 1V/octave, one attenuable Q control input and one at 1V/factor of two, and several features which will be discussed in a moment.

First, though, something about filters which hasn't been mentioned yet—any filter, whether highpass, lowpass, or a combination is characterized by a certain number of "poles" Electronically, a pole is produced by one resistor/capacitor pair. A one pole filter produces a cut-off slope of 6 dB/octave and an attenuation at f_c of 3 dB. When several poles of the same type are strung in series, the chain acts in a multiplicative way; for example, the 2100 VCLPF has four lowpass poles in series whose f_c's are always at the same value—as such, when this value of f_c is, say, 1000 Hz, the f_c of the whole filter is also 1000 Hz, but the cut-off slope is 6 dB/octave per pole, or 24 dB/octave total. Similarly, the attenuation of the VCLPF is 12 dB at f_c, since each pole attenuates 3 dB at f_c.

The UAF, on the other hand, is a two-pole filter. The lowpass and highpass outputs effectively use two poles in series, so their cut-off slopes are both 12 dB/octave. The bandpass output effectively uses one pole for each side of the band, so its two cut-off slopes are both 6 dB/octave. Notch, being a special case because of the phase considerations discussed in the previous chapter, has a very steep 60 dB notch centered on or near f_c.

Notice the difference in sound resulting from a UAF's lowpass and a VCLPF as a result of the different cut-off slopes. Compare the UAF's highpass with the VCHPF. Sharper cut-off slopes are popularly regarded as being more desirable; we tend not to make such value judgments, since you may like the shallower cut-off better for certain effects.

Q in the UAF is quite different from Q in the VCLPF. Notice that the UAF will not oscillate even at maximum Q, which can be over 500. Also, turning up the Q does not lower the overall volume of the output as it does in the VCLPF. Notice, too, that lowpass, bandpass, and highpass can all have Q with the UAF. The explanation for these features is that, whereas a VCLPF or a VCHPF just performs a single filter function on its input, the UAF actually solves several simultaneous filter equations to give its several outputs. With the extremely high Q available from the UAF, the effect of sequentially sweeping through the harmonics of the input by varying f_c is really spectacular. Using a sawtooth for the input and Q at midrange or above, you should be able to isolate harmonics well above the 50th, (as they start to blend, just turn up the Q some more).

An impressive patch with the UAF is simply to input a signal from a synthesizer voice or an external instrument, turn the Q way up, and sweep f_c around with a slow triangle wave, say, 0.1 Hz. Listening to any of the outputs (lowpass may be most pleasant), you will hear an ascending and descending arpeggio which can have very intricate intervals if the input signal is a chord. If a keyboard-controlled voice is used for the input and the UAF is switched to track the keyboard, the arpeggio will maintain the same interval sequence regardless of the key depressed.

You may find certain "wolf notes" in this patch which result when the Q gives a lot of gain to a frequency which is already present at high amplitude. The voltage-controlled Q is ideally suited for eliminating these wolf notes by lowering and raising the Q as needed. In the above patch, you might hear wolf notes when the f_c of the UAF coincides with the lower harmonics of the input, since they are much higher in amplitude than the high harmonics. You could eliminate the "booming" of the low harmonics by using the same triangle to control the Q (through the attenuated Q control input)—thus, at lower f_c's the Q would also be lower.

Another application of voltage-controlled Q is in the synthesis of human speech, which is discussed in detail in a later chapter.

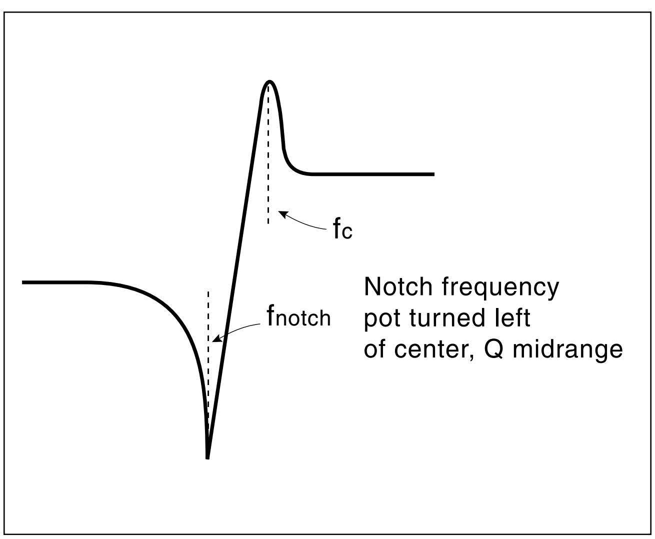
The standard howling wind patch is done very well on the UAF by the lowpass filtering of noise from a noise source with high Q. When you want more howling, turn up the Q control voltage—the UAF will boost any of f_c present in the noise, and since pink noise contains all frequencies, howling at f_c will always be heard if pink noise is used.

If white noise is used, the howling will sound louder at higher frequencies. Pink noise will give constant volume over all frequencies.) Similarly, a sharp tick (say, a keyboard trigger) applied to a signal input of the UAF will cause a "ringing" at f_c when the Q is high. As the Q is raised, the ringing has a longer duration.

Don't mistake these effects for oscillation—the UAF merely boosts any of f_c present in the input signal; when this signal is prolonged noise, a prolonged howling is heard at f_c. When the noise is very quick (a tick), the pure ringing at f_c is all that can be heard. The ability of the UAF to ring is made available as a pre-patch by means of the "keyboard percussion" switch. When switched to one of the keyboards, this switch patches the trigger from that keyboard through a shaping circuit into the signal input summing node, and the gate from the keyboard into a transistor switching network which selects which of the two Q pots is active at a particular time. When the gate is low (no keys depressed), the Q which is in effect is set by the "final Q" pot (this is only true in keyboard percussion mode. In normal mode, the final pot is inert, and the Q is set by the pot). The "Q" pot, in keyboard percussion mode only, determines the which is in effect when the gate is high.

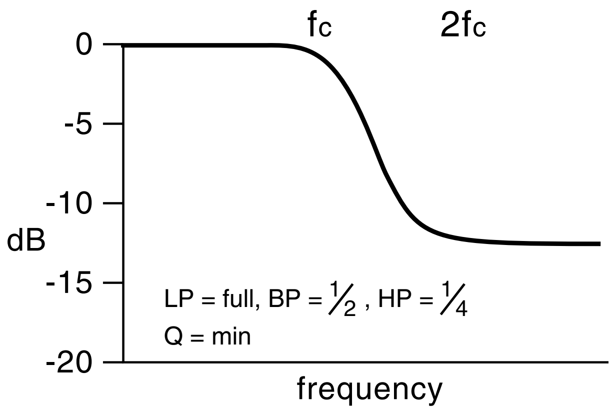
Thus, when a key is depressed, the trigger from the keyboard introduces a tick into the signal input (this occurs even with all the attenuators full left), causing the UAF to ring at f_c if the Q is set high enough. Since higher Q gives a longer ring, the Q pot effectively determines the initial decay time of the ring. When the key is released, the Q jumps to the value set by the final Q pot (either higher or lower Q is possible), and the UAF rings down at that value of Q. Thus, the final Q pot effectively determines the final decay time of the ring. In its usual use, the f_c is controlled by the keyboard voice voltage through the KYBD switch, making the keyboard percussion mode a musically playable patch. It can be used with all the signal inputs off, or the percussion effect can be superimposed an input signal. (We use it most often to calibrate the 1V/octave control!)

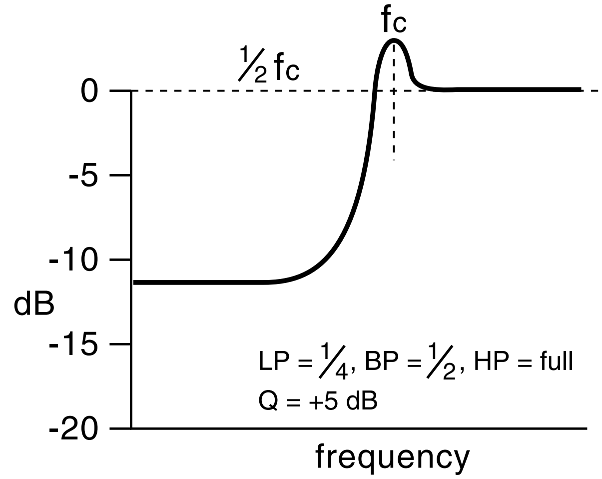
So what do you do with the four outputs? You can, of course, use them one at a time; you have already heard the difference between the two-pole functions of the UAF and the four-pole functions of the other filters. In addition, the UAF bandpass and notch are more convenient to use than the LPF/HPF pairs discussed earlier. In the case of notch, the relative mix of highpass and lowpass can be controlled with a single pot, the "notch frequency" pot. As the pot is turned left of center, the fraction of highpass in the mix increases, causing the center of the notch to be at a lower frequency than f_c. As it is turned right of center, the fraction of lowpass increases, causing f_{notch} to exceed f_c (f_{notch} can differ from f_c by at most a few semitones either way). With the notch precisely at f_c, high Q causes the notch to be defeated, while lower Q values are themselves swallowed by the notch. When f_{notch} is different from f_c, however, both effects can occur, the notch at f_{notch}, and the Q peak at f_c. An attenuation curve for such a knob setting might be helpful:

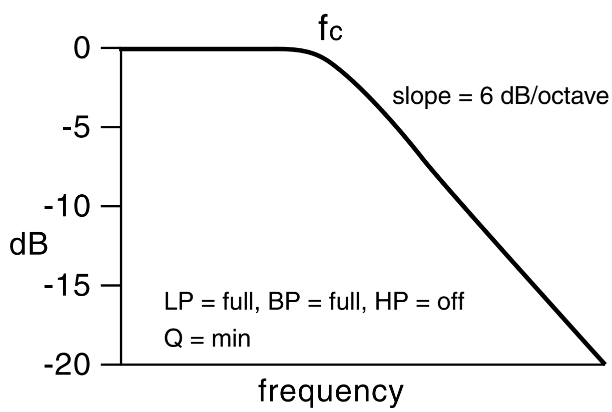


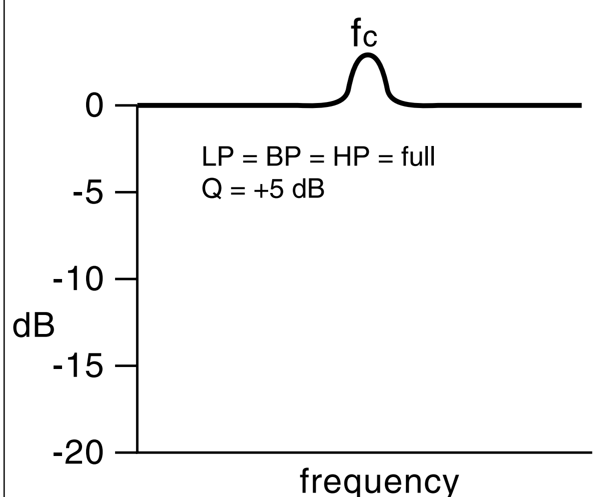
As this formation is swept up and down in frequency space, the effect is quite strange, being sort of a combination of "wah-wah" (Q) and "phase-shifter" (notch).

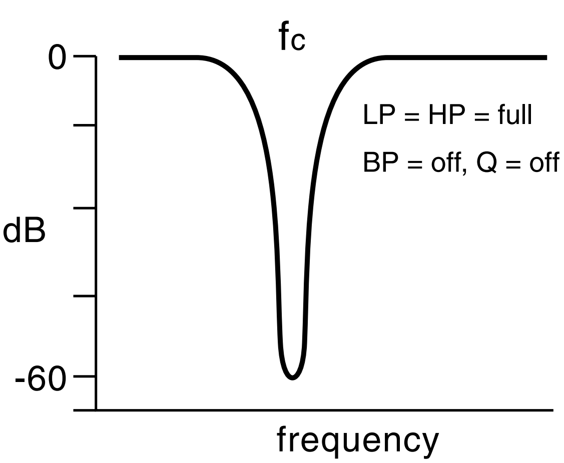
Given several different filter functions all with the same f_c, it is natural to think of mixing them, and the possibilities are indeed musically useful. Try patching highpass, bandpass, and lowpass into three channels of a Quad VCA. (Notch can be left out since it is the sum of H.P. and L.P.) With the initial gains of the three channels set the same, the mix output of the QVCA should be equivalent in timbre (though maybe not in amplitude) to the original signal input to the UAF. This expresses one of the basic equations solved by the UAF: H.P. + B.P. + L.P. = original signal. With the Q minimal, no change will be apparent in the output as f_c is varied, since all frequencies are attenuated the same amount. Q can still be added, however, and the result is a resonant peak at f_c in an otherwise flat response, very useful for producing formants (q.v.) in an output spectrum. Experiment with the different mixes possible from the QVCA—sweep f_c around and vary the Q. Remember that f_c, Q, and the mix proportions can all be voltage-controlled. The drawings on the following page show some of the attenuation curves available from the UAF/QVCA pair:

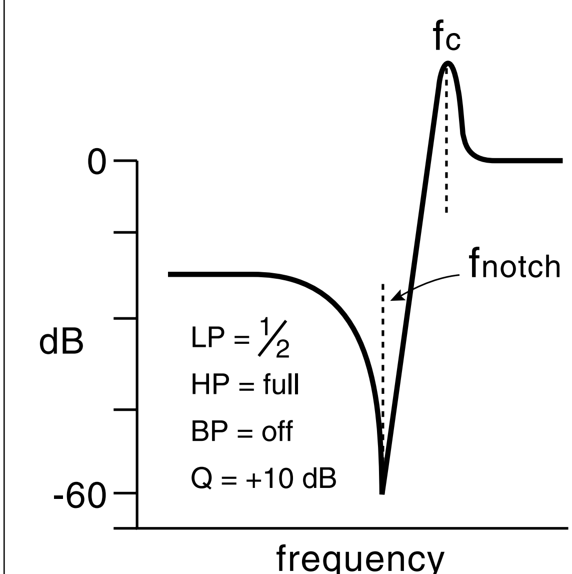

fc
2fc
0
-5
-10
dB
-15
-20
LP = full, BP = ½ , HP = ¼
Q = min
frequency


fc
½ fc
0
-5
-10
dB
-15
-20
LP = ¼, BP = ½, HP = full
Q = +5 dB
frequency

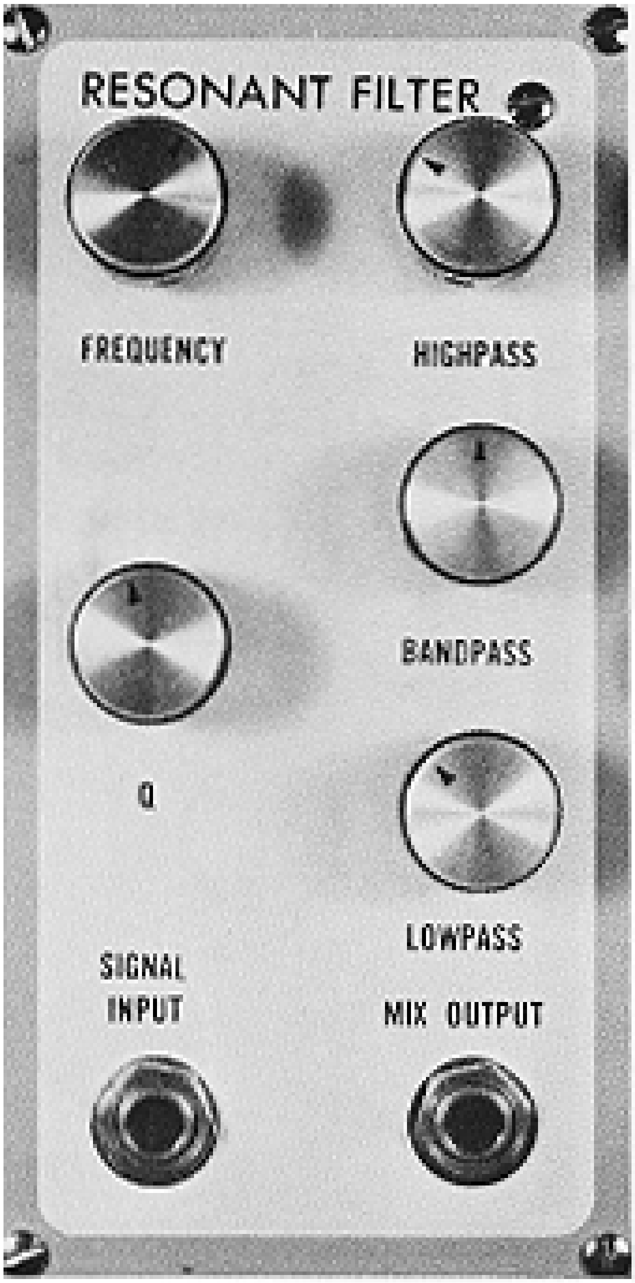

fc
0
-5
slope = 6 dB/octave
-10
dB
-15
-20
LP = full, BP = full, HP = off
Q = min
frequency


fc
0
LP = BP = HP = full
Q = +5 dB
-5
-10
dB
-15
-20
frequency


fc
0
LP = HP = full
BP = off, Q = off
dB
-60
frequency


fc
0
fnotch
dB
LP = ½
HP = full
BP = off
Q = +10 dB
-60
frequency

Resonant Filter

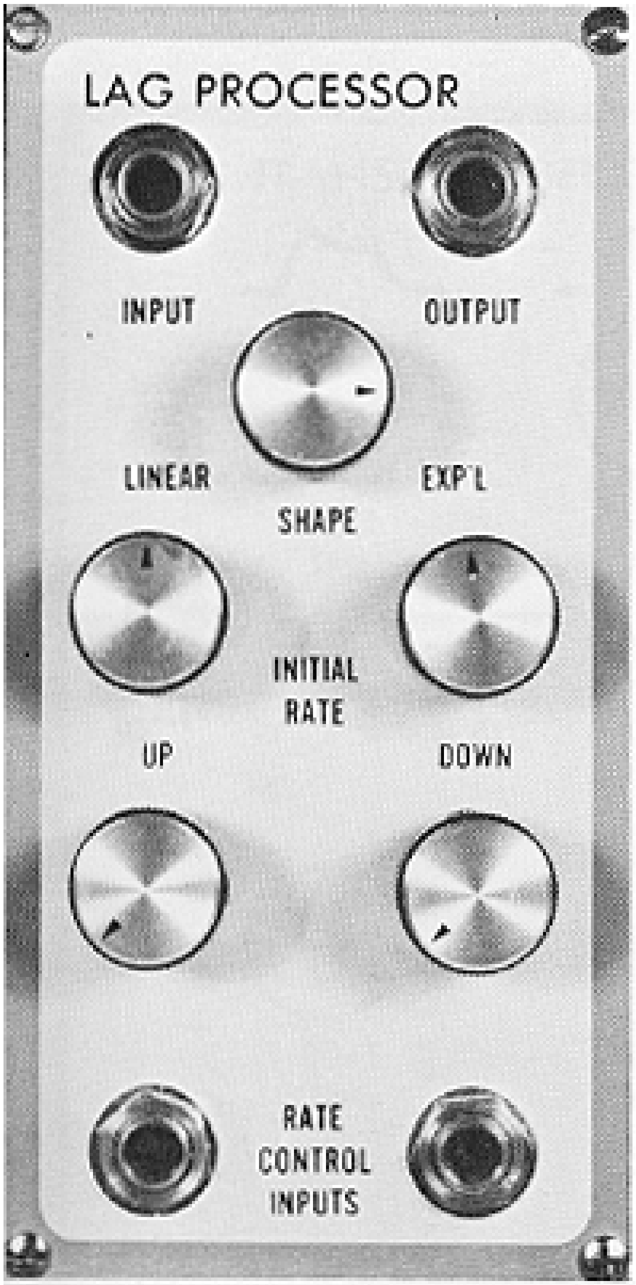


The 2140 Resonant Filter is one module which is capable of producing the same two-pole mixed-function filter characteristics as the UAF/QVCA pair just discussed, but lacking the voltage-controllability features.

Fc is determined on the front panel only by a single pot (though there is an approximate 1V/octave Burndy input in the back), Q is not voltage-controllable, and the mix is not voltage controllable. In addition, the f_c of the 1140 Audio UAF submodule used in the RF will not go as low as the UAF's f_c, and the Q will not go as high. Nevertheless, as a fixed filter or formant filter, the resonant filter is just as capable, much cheaper, and generally easier to use than the UAF/QVCA pair.

Several RF's can be used in series or parallel to give very intricate formant spectra. We regularly use two RF's in parallel (stereo) on the final output of our polyphonic system. By using moderate Q and adjusting the f_c's by ear, we achieve an open "vowel" sound in the output timbre (in human speech, vowels are characterized by two major formants in the spectrum). An amazing range of timbres is available just by changing the two f_c's. We've been told that a violin can be duplicated almost precisely with twenty resonant filters set to the right f_c values! *(There must be an easier way).*

Lag Processor

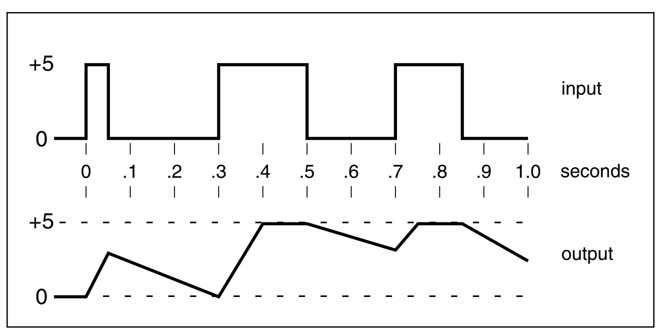


The 2340 Voltage Controlled Lag Processor performs a rate-limiting function on its input; it introduces a linear or exponential slide in the output voltage if the input voltage changes faster than a certain rate. Typically it is used to process control voltages—for example, it can give voltage controlled portamento when its input is a keyboard control voltage, it can turn a gate into a voltage controlled attack/release transient generator and it can take the sharp jumps out of the output of a memory, a VSOU, or a sample & hold.

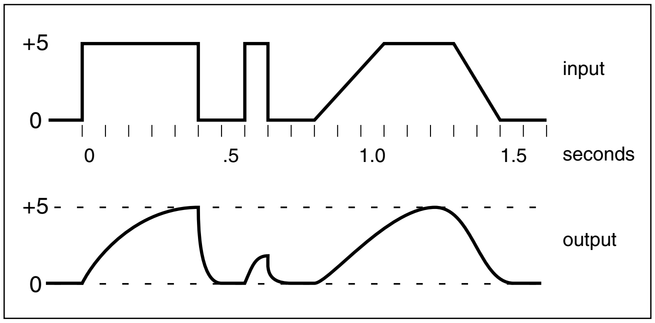
In linear mode (“shape” pot full left), a fast upward change in the input voltage produces an upward linear ramp in the output whose rate is determined by the sum of the “up” initial rate pot and the attenuated “up” rate control input. For larger values of this sum (initial rate pot clockwise, rate control input more positive), the rate of the up ramp is made faster. Similarly, a fast downward change in the input voltage gives a downward linear glide, whose rate is set by the “down” rate pot and the “down” control inputs. In linear mode, a change in the input voltage which happens more slowly than the rate set by the appropriate controls is completely unaffected by the LP, and appears at its own rate at the output.

In exponential mode (shape pot full right), the LP acts somewhat like a VCLPF with a low f_c; the use of the VCF for introducing exponential glides in control signals was discussed in the VCF chapter. The big difference is that the LP can have different f_c’s for upward and downward changes, and they can be separately voltage controlled.

A few diagrams will explain the function of the LP completely. Consider the case where the linear slew rates are 50V/second up and 10V/second down. (With no rate control inputs, these are achieved at roughly midrange and 3/4 left, respectively). The following shows a sample input and the output which results in linear mode.



In exponential mode, with the up rate at midrange and the down rate about 3/4 right, the LP responds approximately like this:

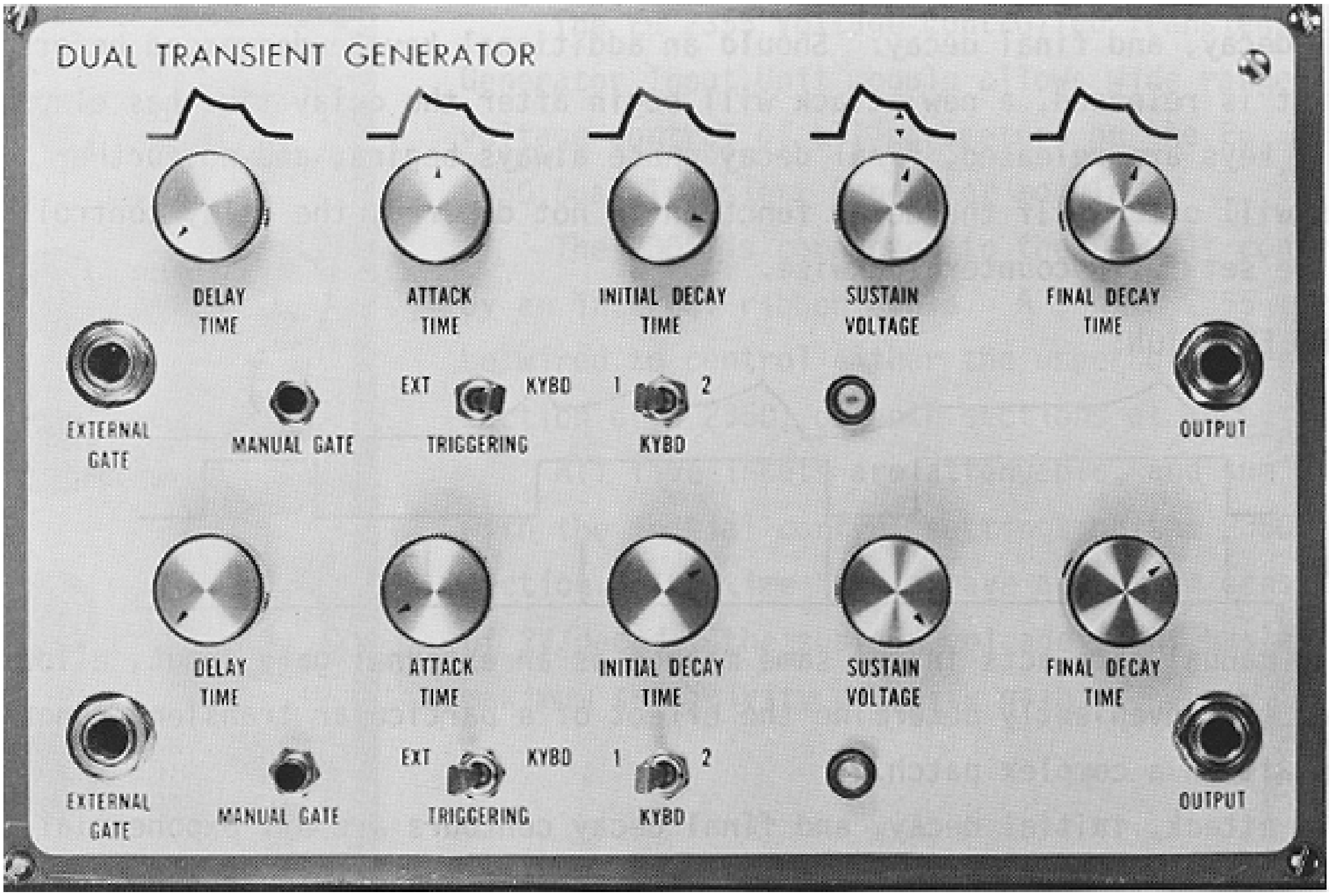


The response to a gate, as shown in this diagram, is useful as a voltage controlled exponential attack/release transient generator. (With the shape pot in between the two extremes, the response to a sudden change in input voltage is a linear ramp initially, which becomes exponentially rounded, starting part-way through the slide).

With the LP in exponential mode, you may have noticed that the output voltage does not settle out at precisely the same value as the input voltage. This can best be observed by using the LP to give exponential portamento to the keyboard. Patch the keyboard control voltage into the LP input and use the output to control a VCO at 1V. You'll find the VCO to be out of tune with respect to the keyboard. This is an inherent limitation of the LP which can be minimized by selection of the components, but can't be eliminated. In linear mode, however, the portamento patch works very well—the LP is, in fact, the only way to obtain accurate voltage controlled portamento. Try this patch with different up and down rates; try sinusoidal rate control inputs, or patch the keyboard control voltage into one or both of the rate control inputs as well as into the signal input, thus giving different portamento rates at the low and high ends of the keyboard.

Other applications of the VCLP will be encountered in later chapters in connection with other control sources; rather than mention them all here, they will be discussed as they come up.

Dual Transient Generator



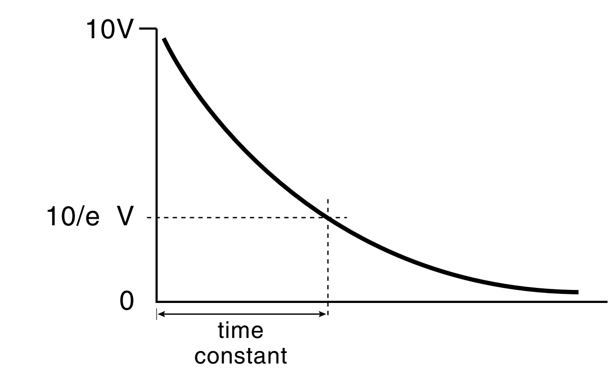
The 2350 Dual Transient Generator contains two identical, completely independent four-phase transient generators in one module. Further discussion will concern only one of them; they are doubled up only because the panel layout is most convenient that way. The DTG may be confusing at first, particularly if you aren't too keen on voltage. It's the first module you've encountered that has no signal input or signal output—its use is strictly as a source of "transient" control voltages: reproducibly varying, synchronously gatable fluctuations in voltage.

A TG (half of a DTG) performs a fairly straightforward function: it has two inputs, a gate and a trigger; and one output, a transient. In the "external" position of the triggering switch, the "external gate" jack is connected to both the gate and the trigger inputs of the appropriate VCDTG submodule (for "voltage controlled delayed transient generator").
In the "KYBD" position of this switch, the gate and trigger from the selected keyboard are pre-patched separately into the gate and trigger inputs of the submodule. For the moment, we'll consider the use of a keyboard to give the gates and triggers.

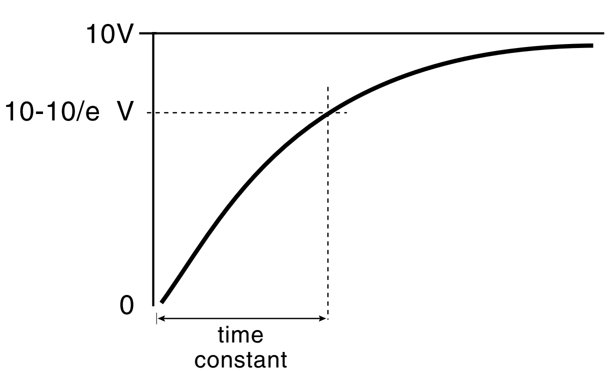
When the gate goes high and the trigger occurs, the transient remains at zero volts until the delay time set by the "delay time" pot elapses, and then begins an upward inverse exponential increase at a rate determined by the "attack time" pot. If the gate falls before the delay time elapses, no transient is produced. Assuming the gate stays high indefinitely, the transient, now in the attack phase, exponentially approaches a level set internally which we will call the "attack approach voltage" (about 12V) Being exponential, the transient technically would never reach the approach voltage, but it doesn't have to— at another internally set voltage, the "attack cut-off voltage" (about 10V), a-comparator is fired, causing the attack phase to end and the initial decay phase to begin.

In the initial decay phase, the transient falls exponentially at a rate set by the "initial decay time" pot, leveling off at the sustain voltage set by the "sustain voltage" pot. This downward approach to the sustain voltage continues indefinitely as long as the gate stays high. (For all practical purposes, the transient can be said to reach the sustain voltage and stay there as long as the gate is high). When the gate falls, the transient enters the final decay phase, in which it exponentially decays to zero volts at a rate determined by the "final decay time" pot.

Since the decays are exponential, they can't be timed from start to finish— there is no finish. The attack can be timed precisely, since it is cut short at a definite voltage; nevertheless, all these exponential approaches are conveniently described by a "time constant", which is defined as the time required for the exponential approach to get within 1/e (1/2.71828...) of its approach value. The following drawings illustrate the concept of time constant:



EXPONENTIAL APPROACH

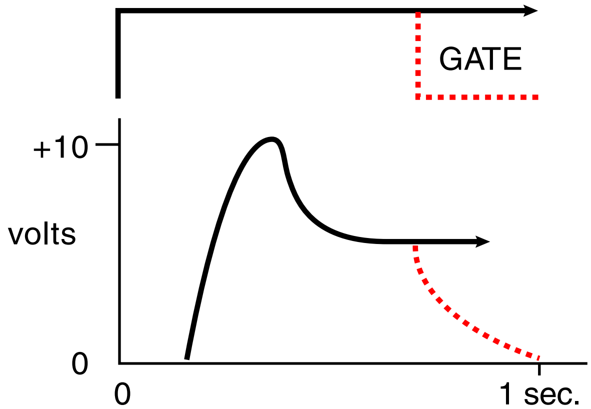


INVERSE EXPONENTIAL APPROACH

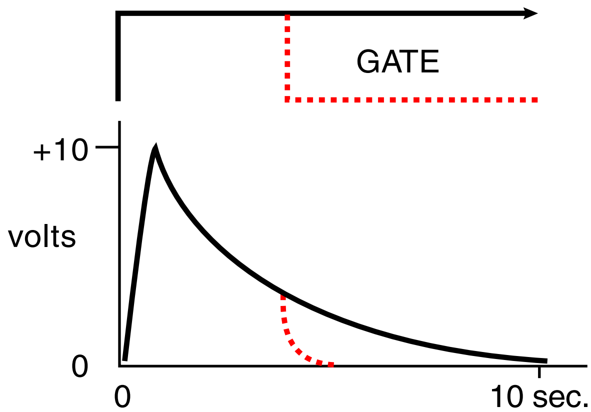
Clearly, then, a decay is still significantly short of its approach value well after the time constant elapses. For example, a final decay with a time constant of 10 seconds will give a total glide time of about 30 seconds. (The time constant is precisely definable, however, while the "total glide time" depends on the sensitivity of your ear). The attack phase, being cut off at 10/12 of its approach value, lasts about 1.7 time constants.

The range of time constants available from the pots for attack, initial decay, and final decay is, in each case, about 1 msec. to 10 seconds. You may notice, particularly with a 'scope, that a certain rotation which gives a change in time constant from 1 msec to 5 msec at the far left extreme will give a change from 2 seconds to 10 seconds at the far right extreme. This is an exponential response to the knobs; it allows you to achieve equal sensitivity at very short and very long time constants. Electronically, it is accomplished by making the submodule produce an exponential change in time constant in response to a changing control voltage; this control voltage is then obtained from a linear pot, giving an overall exponential-response to the pot. The delay time is also exponentially controlled by its pot, and can vary from about 3 msec to 3 seconds. The sustain voltage responds linearly to its pot, and can be set between zero and +10V.

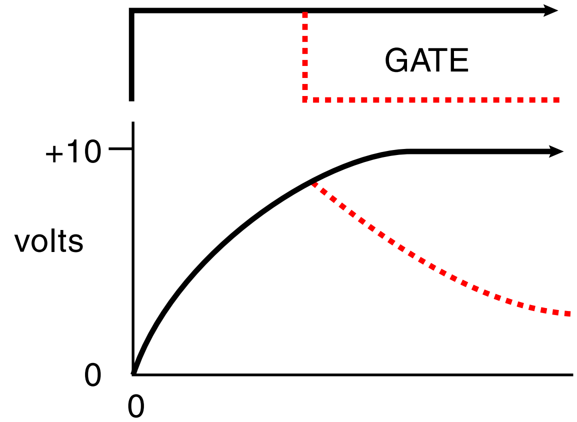
The following drawings show some of the transients that can be obtained from TG's; first, the simplest cases, where a keyboard is used to initiate the transient, and only one key is depressed at a time. Notice that if the gate falls at any time during the transient, the TG immediately enters final decay phase:



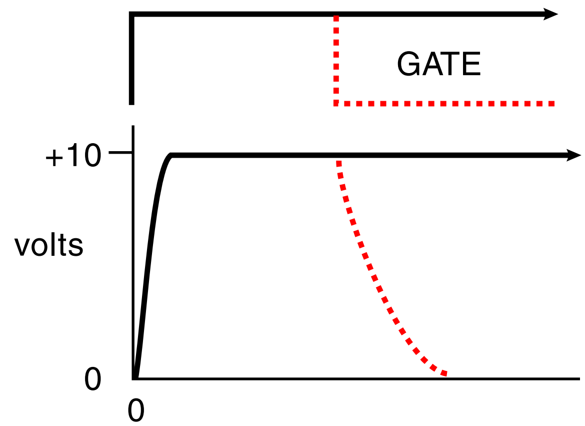
all controls midrange



no delay, fast attack
slow I.D., S.V. = 0, fast F.D.
(piano envelope)



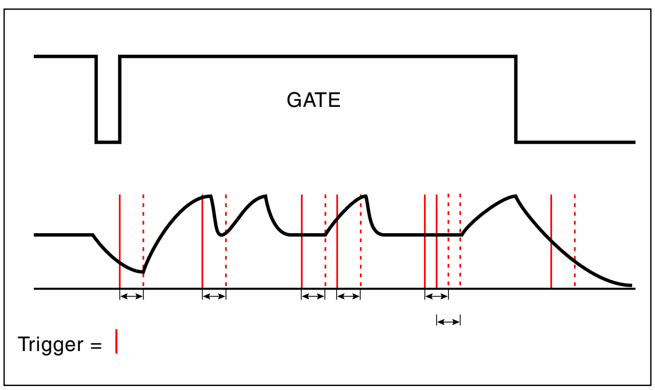
no delay, slow attack
fast I.D., S.V. = 10V, slow F.D.



no delay, fast attack,fast I.D.
S.V. = 10V, fast F.D.
(organ envelope)

When a monophonic keyboard (q.v.) is used, additional triggers are often produced without the gate ever going low; specifically, this happens either when a lower key is depressed without releasing an upper key, or when two keys are depressed and the lower is then released. It is necessary that a transient be able to respond (with the appropriate delay) to such additional triggers.

The diagram at right shows most cases that might arise, and the response of the TG in each case. The trigger applied by the keyboard is shown in solid red, while the trigger seen by the circuitry after the delay time elapses (the "delayed trigger") is shown in dotted red. The delay time remains constant throughout, and is shown by the double-headed arrows:



Study carefully the conditions under which new attacks are initiated. Notice that a delayed trigger has no effect if it occurs during an attack phase (or without a high gate). Also, many triggers in quick succession may have no effect if the delay time is longer than their spacing—only the last delayed trigger is seen.

The manual gate, like the external gate input, is patched to both the gate and the trigger input of its TG. When you can't tell what's going on with a TG, this push-button can narrow down the range of your confusion considerably by allowing you to gate the TG by hand. Because of the extremes at which the rate pots can be set, you may sometimes get no apparent transient even though you can see the gate lamp lighting (indicating the gate is high). First make sure the transient is patched where you think it is, and that it is not attenuated too far. Next check to see if the attack is set extremely slow, or the delay very long—alternatively, the attack and initial decay may be so short that the entire transient becomes a "blip" (only a problem if the sustain voltage is low). You may also have problems with "popping" sounds if the attack or decay is set very fast; these can be eliminated by turning the appropriate pots slightly clockwise.

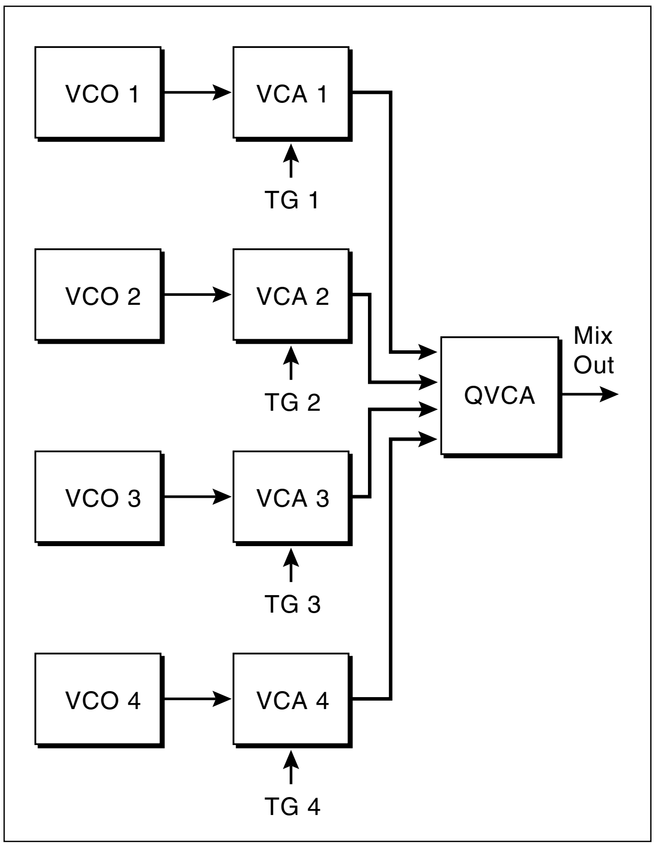
The TG need not receive its gate and trigger from a keyboard. In the "external" position of the triggering switch, an input applied at the "external gate" jack is sensed as a high gate within the submodule whenever it exceeds 2.0V, and as a trigger whenever it passes upward through 2.5V. The difference in threshold voltages assures that a trigger will never occur without a high gate. It thus becomes possible to initiate transients with a pulse wave, a sine wave, or any other varying voltage with excursions above 2.5V. Slowly rising voltages will, of course, give non-simultaneous gate and trigger—attacks will begin with the delayed trigger. A very important use for the external gate is the initiation of transients with digital signals (q.v.). By patching a digital output of a memory into a TG's external gate, it is possible to have attacks only on certain notes of a sequence—this is how timing in a sequence is achieved. In polyphonic systems, the external gate is often the most convenient way to apply keyboard gates to their respective TG's. This can be done with the firm-wire external gate input, if desired.

Still another way to apply gate and trigger signals to a TG is through the keyboard busses. If a keyboard bus is free, the TG can be switched to that bus and can then receive its gate and trigger from the gate and trigger jacks on the power supply front panel. It is thus possible to use two completely independent control voltages to provide independent gate and trigger to a TG (the threshold voltages are still 2V and 2.5V for gate and trigger, respectively). The result is a type of "and" function performed on the two inputs, in that the conditions for a gate must be met at the same time that a delayed trigger is generated for a transient to occur. The effect can be very complex—see if you can find a good application!

The standard uses for a transient are pretty obvious; in fact, you already know a number of them. As an envelope, or loudness contour, a transient is indispensable. It was because of its use as an envelope generator that it got that name—we have avoided the term because it refers to only one of the many possible applications for sub-audio transients. The use of the TG for envelopes also explains the choice of an A.D.S.R.-type function (attack, decay, sustain, release); fast attacks with long decays give envelopes with a percussive effect (like a piano, for example), while slow attacks with high sustain voltages can give brass-like envelopes. Generally speaking, the ADSR transient is very versatile. In addition to giving good likenesses of standard instrument envelopes, other envelopes not available from any conventional instrument can be obtained from the TG.

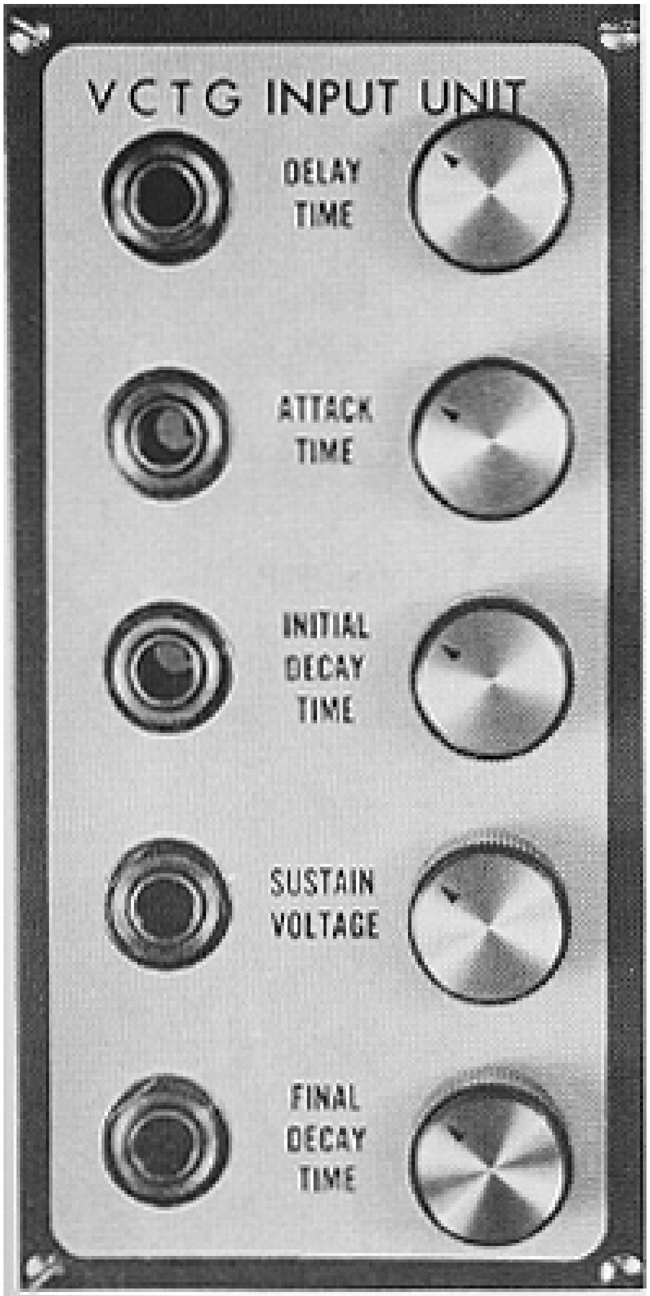
There are a few uses for transients that are less obvious—transients on filters, VCO frequency controls (perhaps the most graphic way to hear the transient pattern), or using a transient as the signal input to a Sample & Hold. It is sometimes useful to invert a transient with a Quad Inverter (q.v.) before applying it to a control input. (One application of an inverted transient was already discussed in connection with the VCHPF, but a quad inverter was not necessary in that case). When you try this, remember to raise the initial setting of the controlled parameter, since the inverted transient is always negative. Multiple transients can be superimposed to control the same module, with very complex fluctuating functions resulting. Try patching two TG's into two of the control inputs of a VCA—use the delay time to offset one transient relative to the other. Several transients can also be used to simultaneously control different parameters within a single voice. We've already mentioned the use of separate transients on a VCF and a VCA.

Consider the following patch:



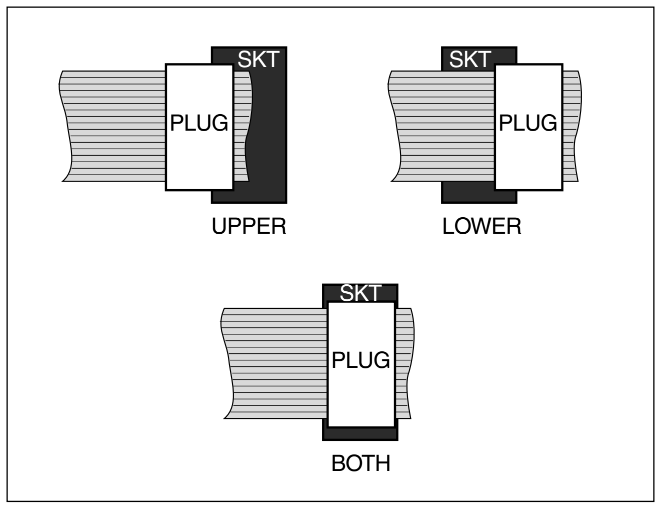
If the four VCO's are tuned to track together, say, at intervals corresponding to the fundamental and the next three harmonics of a note, the envelope of each harmonic can be independently controlled by its own TG. If all the TG's are triggered from the keyboard, the delay time allows any of the envelopes to have the same shape but to be displaced in time, the only limitation being that the final decays will all begin simultaneously when the gate falls.

VCTGIU

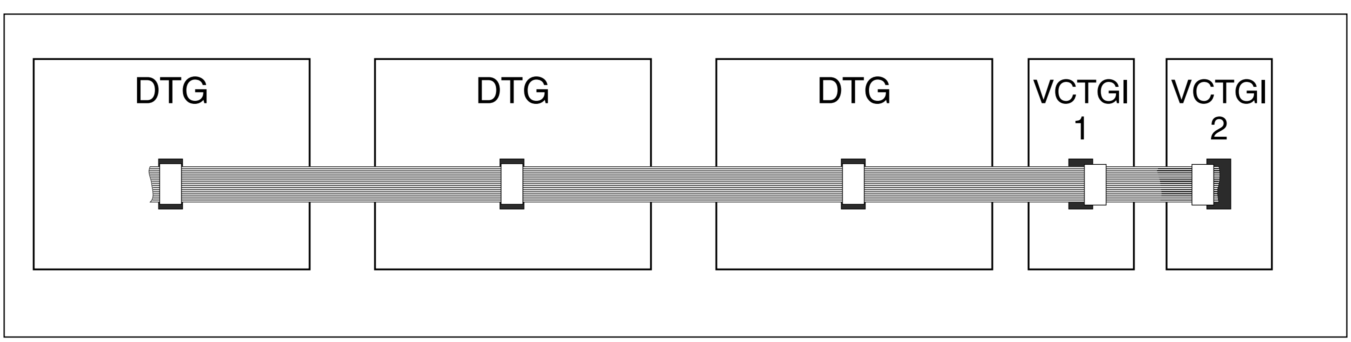


The 2355 Voltage Controlled Transient Generator Input module makes it possible to control all five of the TG parameters with input voltages; the VCDTG submodule is always voltage controlled, but it receives its control voltages only from the pots on the module when the VCTGI is not used. The VCTGI simply adds attenuated control voltages to the voltages obtained from the appropriate pots.

The control inputs applied to the VCTGI are carried by a length of ribbon cable (completely independent of the power and busses ribbon cable) to a DIP socket on the TG module board. The VCTGI inputs can control either the upper TG in the DTG, or the lower, or both, depending on how the DIP plug is plugged into the DIP socket on the VCTGI:



The DIP plug on the DTG itself is plugged into its socket in the normal way. One VCTGI can control any number of TG's; here is a wiring scheme which gives VCTGI 1 control over the upper TG's in three DTG'S, and gives VCTGI 2 control over the lower TG's;



Notice only one piece of ribbon cable is required. The responses of the time constants of the TG to the VCTGI inputs are exponential (as you would expect, since the responses to the pots are exponential). With the attenuators on the VCTGI full right, the delay, attack, and decay times vary roughly by a factor of ten with a 2V change in the input. Thus, if the final decay pot an the TG is turned full left, giving a time constant of about 1 millisecond with no input from the VCTGI, the time constant can be increased to 1 second by applying about 6V to the unattenuated "final decay" jack on the VCTGI. As the attenuators are turned left, the response to voltage remains exponential but becomes less sensitive.

With the sustain voltage attenuator full right, a voltage applied to the "sustain voltage" jack simply adds its value (which may be negative) to the value set by the sustain pot on the TG. The response is thus linear, and may be made less sensitive with the attenuator. The sustain voltage of the transient will not exceed about 12V regardless of the voltage applied to the VCTGI jack. It will go below zero, however, and this may be useful in some applications. (Though we haven't thought of one!)

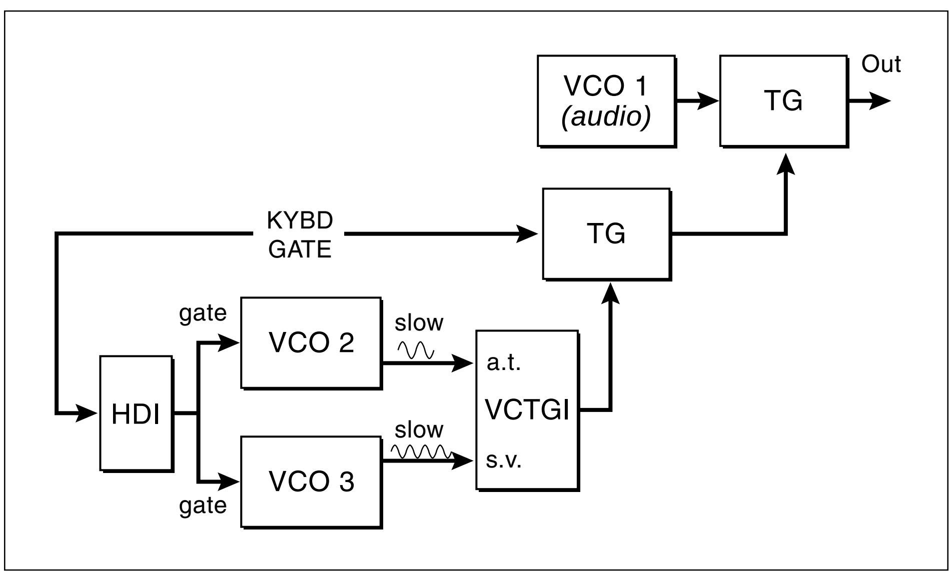
Notice that the time constants can be made much longer with the VCTGI than they can with the TG pots alone. In fact, with the final decay pot turned full right and +10V on the unattenuated final decay input of the VCTGI, the time constant for the final decay is theoretically about 11 days! Less extreme time constants are perhaps more valuable.

The patches that come to mind with voltage control of TG parameters are numerous. A foot pedal can be used as a "sustain" pedal by having it control final decay time. A foot pedal on the sustain voltage gives a type of sustain, also.

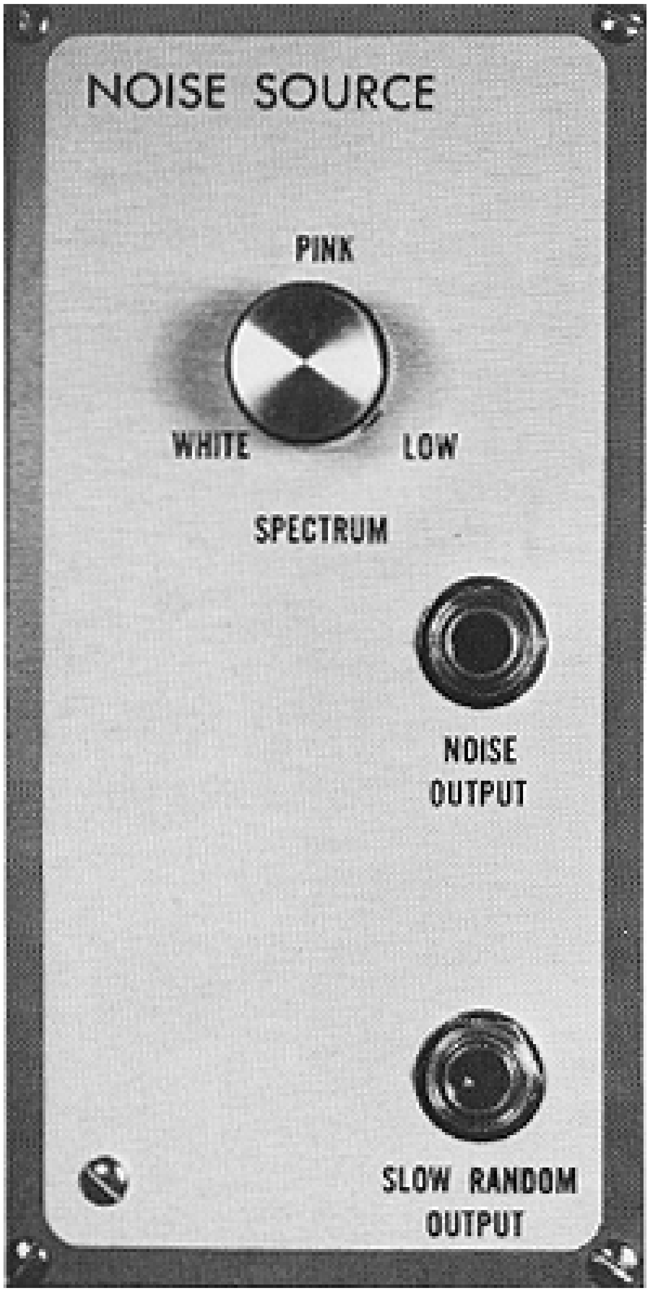
Try patching the inverted keyboard control' voltage into the final decay input. (Turn the final decay pot on the TG high, since the output of the inverter is always negative). The decays are now shorter for high notes and longer for low notes—much like the behavior of a piano keyboard. Use the four outputs of a VSOU (q.v.) to control four of the parameters of a TG. (The TG pots can all be turned full left, so that only the VSOU pots determine the control voltages). By setting up the eight columns of the VSOU appropriately, eight completely different transients are available, and the desired transient can be instantly selected with the pushbuttons on the 8AG. In this same patch, any number of additional TG's could be controlled by the same VCTGI, and all would have equal parameters if the pots on every TG were turned full left. This use of the VCTGI as a "gang" control for several TG's is very powerful.

Try using sub-audio waveforms from VCO's to control various TG parameters. For example, a sine wave on sustain voltage will give a tremolo (if the transient is used as an envelope) which begins only after the attack and initial decay phases. (In this patch, initial decay rate will have an effect on the depth of the tremolo). If the VCO's producing the control waveforms are gated by the keyboard, complex but reproducible transients can be obtained.

Here's a sample patch:

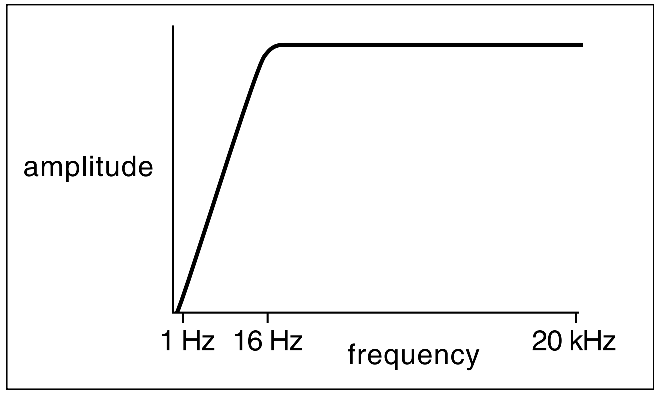


Noise Generator



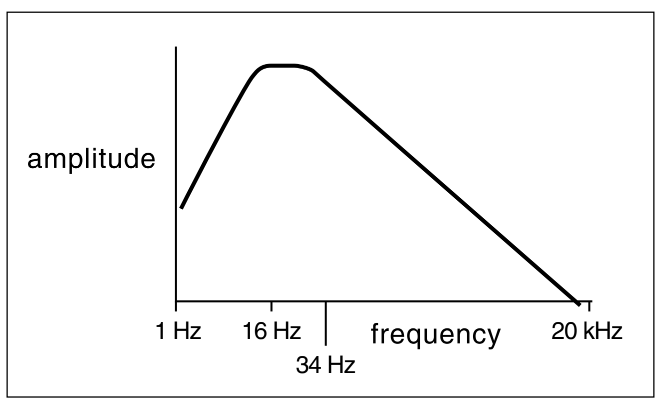
"Noise", in electronic terminology, has fairly specific definitions and varieties; "white" noise is a mixture of all frequencies in the audio spectrum at the same average level (analogous to white light in the visible spectrum).

The spectrum of white noise available from the 2400 Noise Source looks like this:



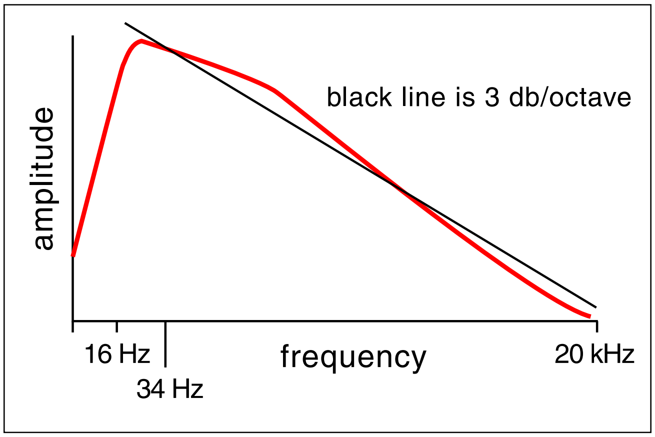
This is obtained from the "noise output" jack when the "spectrum" pot is full left. The attenuation of the theoretically flat spectrum at low frequencies is due to the necessity of filtering out any DC component in the output. Listen to white noise through a VCF with the Q set just short of oscillation. As you sweep f_c, you will hear the VCF output whistle smoothly through the entire audio spectrum, indicating the presence of every frequency in the noise.

Using a VCA as an attenuator, listen to the NS noise output and turn the spectrum pot full right. The output is now called "red" or low filtered noise. In this position of the spectrum pot, the noise output is white noise which has been lowpass filtered at 6 dB/octave. (This is done with a single pole filter whose f_c is about 34 Hz.) Its spectrum is as follows:



Notice that the low filtered noise is no lower in volume than the white noise, even though it has been filtered. The circuitry boosts the low filtered output to accomplish this, the result being that low frequencies are actually louder in the low filtered output than in the white noise, while high frequencies are attenuated.

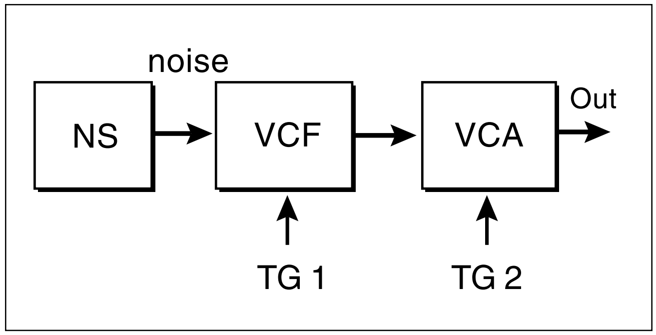
Intermediate positions of the spectrum pot give the full range of mixes of white and 6 dB/octave low filtered noise. At approximately midrange, the mix is 50:50; this gives a noise spectrum with a complex cut-off curve which is a close approximation of a 3 dB/octave slope:



Noise with precisely a 3 dB/octave cut-off slope is called “pink noise”, and the significance of this particular slope is that it gives an equal power distribution per octave in the noise output. (White noise, with equal average power at all frequencies, contains more power within one high octave than within one low octave, since the high octave spans a wider frequency range). Actually, a precise 3 dB/octave cut-off is difficult to obtain, since a simple one-pole filter gives 6 dB/octave. Some pink noise sources, use complex filters to give a “step-type” cutoff which closely approximates 3 dB/octave; the 2400 NS uses the mix method to achieve a different approximation.

The slow random output is a randomly varying sub-audio control voltage with typical excursions of -5V to +5V, and frequency components maximized between 0.1 Hz and 10 Hz. (Since the slow random output is inaudible, it certainly doesn’t have significant components above about 30 Hz. The slow random output is completely unaffected by the spectrum pot.

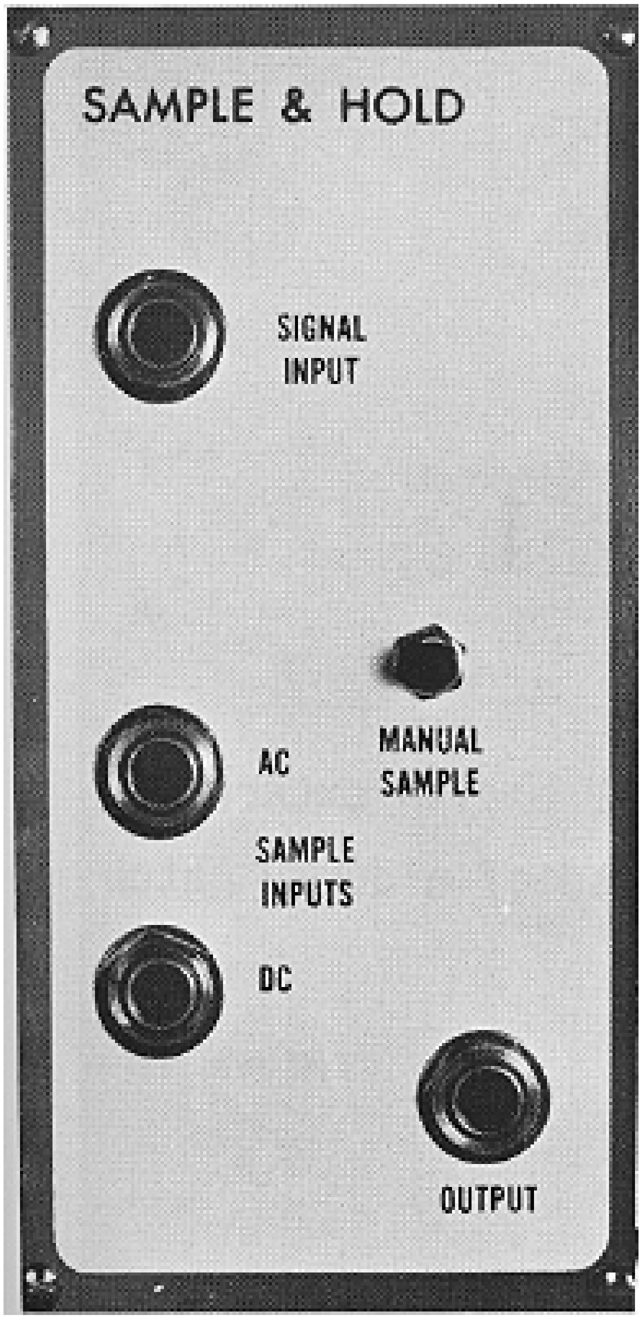
One use for noise which has already been discussed is the howling wind patch using a filter with high Q. Noise is also good for achieving percussive effects—drums, cymbals, etc. Try the following patch:



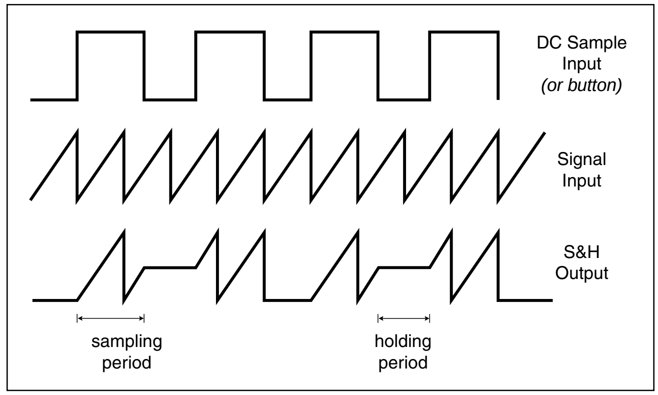
With this simple patch, appropriate choices of knob settings will give a very wide range of conventional percussive sounds—a high hat cymbal imitation should be pretty easy, as should tom-tom, bass drum, and wood blocks. (These last three require high Q to give a tone to the sound). Snare drum is very difficult, and certain other percussive sounds like ride cymbal and bells are better imitated with the ring modulator. Of course, imitation is not the name of the game; many unique percussive effects are available with this patch that have no counterpart in conventional instruments. The noise source is also useful for adding a “breathy” quality to the attacks or overall timbre of a musical voice. It can also be used for simple consonant sounds in human speech synthesis, and excellent renditions of pounding surf and thunder.

The slow random output is a special purpose control voltage which is sometimes valuable for giving life-like random quality to a sound—for example, it is indispensable to the thunder patch and the synthesis of animal sounds like bird chirps. Often, slow random control is more of a comical effect, as in the “ticklish alien” patch, in which slow random output controls the frequency of a VCO. Both noise and slow random output can be used as inputs to a sample and hold for creating random step functions—this will be discussed in the next chapter.

Sample & Hold

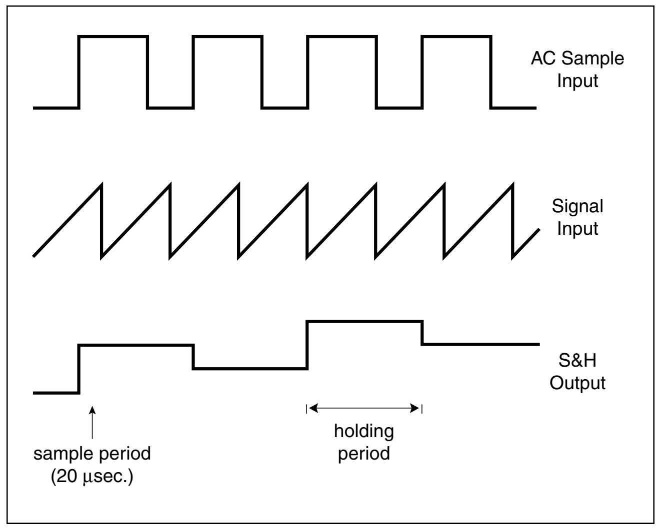


The 2410 Sample & Hold has an output which will track the input while a sample command is given, and will hold the input voltage which is present when the sample command is removed. Two sample input jacks are available: the DC jack allows continuous sampling as long as its input voltage exceeds 2.5V; the AC jack is sensitive only to rising edges, and causes the input to be sampled for only 20 microseconds when a rising edge is applied. Diagrams will show these effects more clearly. Consider first the DC sampling:



(When the 'manual sample' pushbutton is depressed, its effect is the same as a high level an the DC sample input).

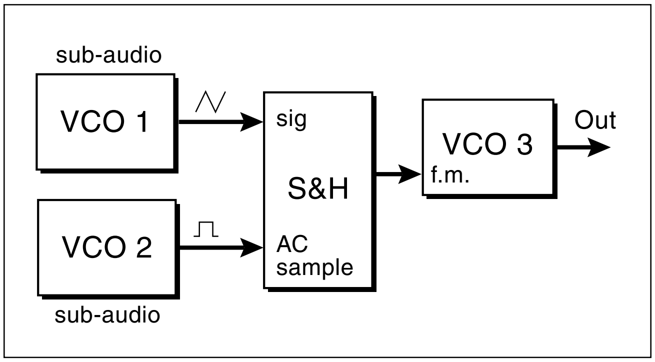
Here's a diagram of AC sampling:



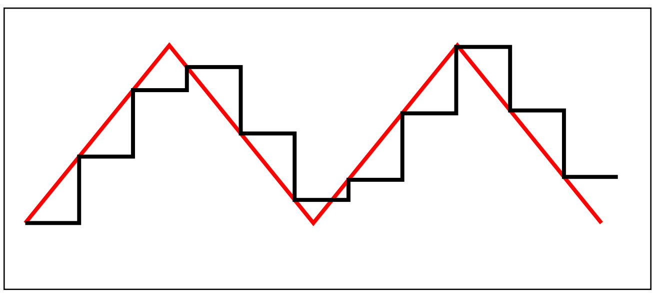
The holding stability of the S&H is excellent, but not perfect. The output voltage will begin a very slow exponential fall as soon as the holding period begins. This decay amounts to about a millivolt every few seconds for a held voltage around 5V. Being exponential, of course, the decay is even slower at lower voltages. This can be minimized if the output can be "refreshed" periodically with a new sample command. This will be discussed later in reference to sampled keyboard voltages.

S&H can be used for a wide range of very different effects.

Consider the following patch:



If the triangle is much lower in frequency than the pulse wave, a step function results at the S&H output:

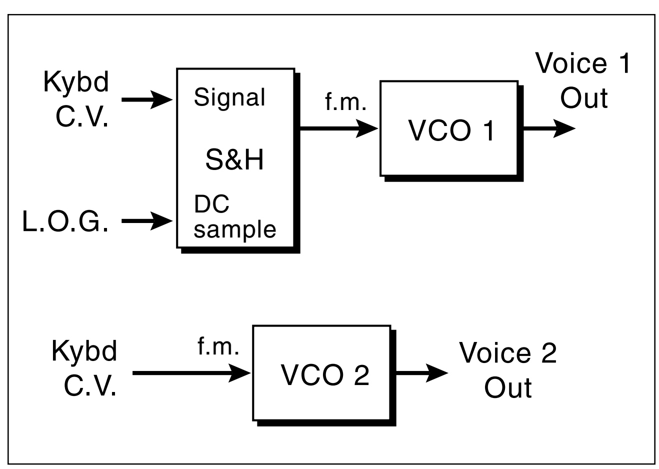


This gives a stepwise ascending and descending glissando at VCO 3's output. If VCO's 1 and 2 are sync'd, the glissando will be repetitive.

With the sync off, gradually raise the triangle frequency. As the period of the triangle decreases, the fractional portion of one triangular cycle covered in each step of the sampled output increases, causing the output to lose any resemblance to a triangular step function. Instead, we obtain a sequence of voltages which typically gives a pseudo-repetitive sequence of pitches from VCO 3: for example, an identifiable pattern may repeat itself with a gradual upward or downward drift. The sequence becomes perfectly repetitive if the triangle and pulse are sync'd. With the sync off, adjust the triangle and pulse to almost the same frequency; a slowly drifting tone will be heard from VCO 3. This is actually still a step function, but the voltage change from step to step is so small as to seem continuous. If the two waveforms are now sync'd to exactly the same frequency, a single pitch will result, since the pulse rising edge will always occur at the same part of the triangle cycle. If the inverted pulse of VCO 2 is used, the pitch of VCO 3 will depend on the pulse width setting of VCO 2, since the pulse width controls the location of the rising edge in the inverted pulse wave.

Try the whole range of frequency settings for VCO's 1 and 2. Many complex and bizarre sequences can be had from this simple patch. Try applying the sampling pulse wave- to the DC sample input—the effect will be similar but will include glides within each high portion of the pulse wave, since, during that time, the triangle appears unchanged at the S&H output. Notice the effect of pulse width on the DC sampled output; if the high portion of the pulse is short enough, the result is just like AC sampling. Truly random sequences can be obtained by sampling noise or slow random output. Very interesting timbral sequences are available when the output of an S&H is used to control the f_c of a filter (particularly a UAF). Different sampling signals can be applied to the AC and DC jacks simultaneously—the results can be complicated almost to the point of unpredictability, but it could be powerful. Give it a try! Remember, too, that any voltage with excursions above 2.5V will operate the DC sample function; similarly, any fast rising edge, such as a digital output or a gate, will perform an AC sampling.

One final patch that may be of interest is the use of an S&H to remember keyboard control voltages:

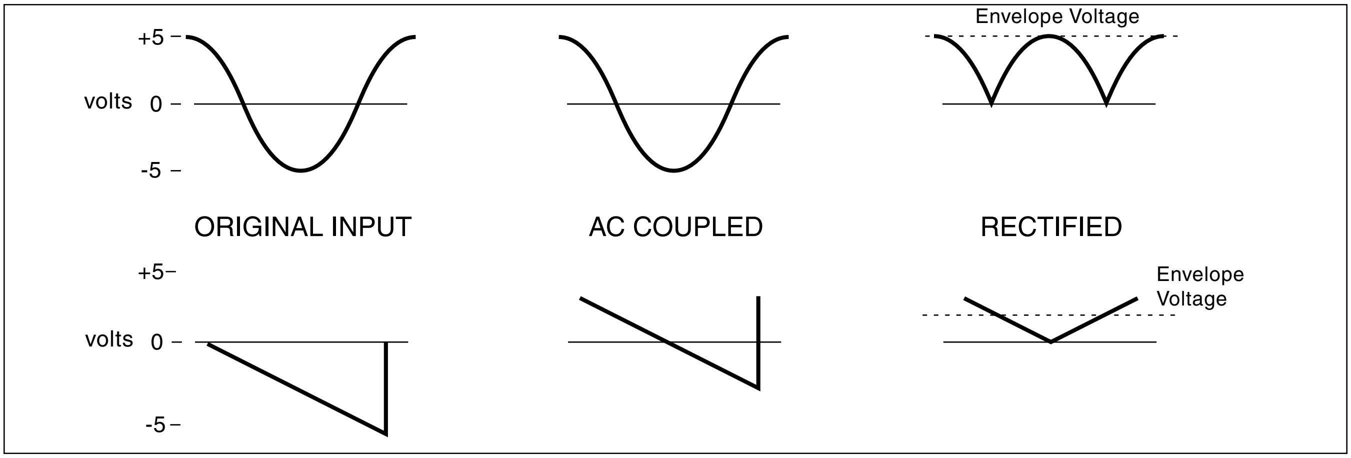


In this patch, voice 2 responds to the keyboard directly, while voice 1 is a “drone” voice whose pitch can be reset with the low octave gate. (Be sure the low octave is turned off). As long as the LOG is high, voice 1 tracks with the keyboard just like voice 2. As soon as the low octave key is released, however, the pitch of voice 1 remains fixed at its present value; voice 2 can now harmonize with the drone. This patch, cleverly used, can make a monophonic keyboard almost as powerful as a polyphonic one, especially if several independent drone voices are used. Its only limitation is the slow downward drift of the S&H output. If the drone note is changed every few seconds, the drift won’t cause any problems. If a drone note must continue for quite a while, it has to be “refreshed” every so often by depressing the low octave key while the desired drone key is also depressed.

Envelope Follower

The 2440 Envelope Follower performs three different functions in one module: envelope detection, voltage comparison, and slope-sensitive trigger generation. The three respective outputs are "envelope", "gate", and "trigger". When an audio-rate signal is applied to the signal input, the output at the envelope jack is a positive voltage proportional to the amplitude (loudness) of the input signal . When the voltage applied to the "external comparator in" jack exceeds the sum of the voltages from the "initial gate threshold" pot and the "threshold input" jack, the gate output goes high (about +5V). (Otherwise, the gate output is zero volts). When the voltage applied to the comparator input increases at a rate faster than the rate set by the "trigger sensitivity" pot, a 5V pulse (trigger) lasting 100 µsec is output on the trigger jack. With no patch cord in the comparator input jack, it is a pre-patched input from the envelope output jack, thus allowing gate and trigger to be extracted from the envelope of an applied signal.

Consider first the envelope output. Envelope detection is usually not performed on a synthesizer-generated signal; the synthesizer can control its own envelopes with VCA's, so the envelopes are already available in the form of VCA control inputs. The usual application of an EF is to extract the envelope of an externally generated signal, like the output of a microphone or electric guitar. Nevertheless, for simplicity, we'll start by using a VCO as the signal input to an EF. Patch the mix output of a VCO to the signal input of an EF, and look at the voltage from the envelope output with a voltmeter, oscilloscope, or, if nothing else, a VCO F.M. input. Set the input VCO frequency in the audio range, and, starting with all the waveforms fully attenuated, turn up the sine pot. The envelope output will follow the level of the VCO output, reaching about +5V when the sine output is maximum. (This is one of the uses for envelope detection on a synthesizer-generated signal—obtaining a control voltage from an attenuation pot). Turn off the sine pot and turn up the sawtooth pot. You'll see two noteworthy things; first, the envelope output is still always positive, even though the inverted sawtooth is always negative, and second, the envelope output with the sawtooth maximized is only about 2V. Both observations are understood by seeing how the envelope is detected. First, the input signal is AC coupled, which means that any DC offset is removed. In the case of sine, this has no effect; however, inverted sawtooth, since it ranges linearly from zero to -5V, has a DC component of -2.5V. Without this offset, the sawtooth has the same shape, but ranges from +2.5V to -2.5V. After being AC coupled, the signal input is rectified, which means all



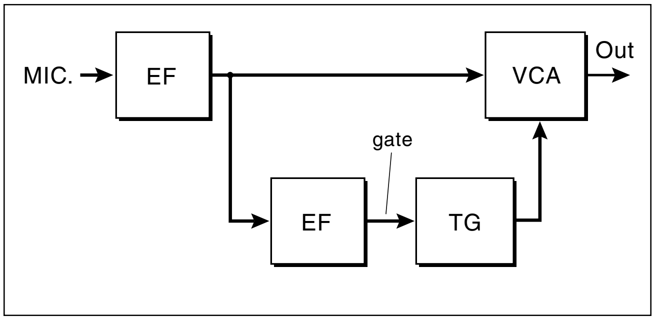
negative portions of the waveform are reflected through zero volts, thus becoming positive. After rectification, the resulting DC offset (the envelope) is extracted by a simple four-pole lowpass filter (f_c about 25 Hz) Notice that these three functions (AC coupling, rectification, and lowpass filtering) could be performed in a patch with a VCHPF, a ring modulator, and a VCLPF. The EF simply performs these functions (and others, as well) in one module. See the diagram above.

Notice that the envelope output for a sinusoidal input is actually somewhat higher than the peak voltage of the rectified sine wave, while the envelope of a triangle or sawtooth is less than the peak rectified voltage. The reason for this is that the voltage level of the DC component present in a rectified waveform depends upon the waveform as well as the peak level of the waveform. As you can see from the drawing, the rectified full-level sawtooth spends equal time above and below 1.25V, so the DC component in that waveform is 1.25V. This is amplified to about 2V by the output circuitry. In the case of the rectified sine wave, the waveform spends equal time above and below $10/\pi$ volts (3.18V), so this DC component, amplified by the same factor, gives an envelope output of about 5V.

The envelope output begins to have a significant amount of ripple when the input signal frequency is below 50 Hz. (For a 50 Hz sine input, the peak-to-peak variation in the envelope is about 10% of the average value of the envelope). As the input signal frequency gets even lower, the envelope drops off toward zero volts, since the AC coupling (highpass filtering with f_c =16 Hz) begins to attenuate the input at low frequencies.

As was mentioned earlier, the envelope output is commonly used when the envelope of an externally generated signal is desired. To do this with, say, a microphone output, patch the microphone into a Dual Preamp module (q.v.) and the output of the preamp to the signal input of the EF. Adjust the preamp gain until the envelope output is about 10V when the volume level into the microphone is at a maximum. The envelope output can now be used as a control voltage for any purpose: try running the microphone signal (from the DP) through a filter while its envelope controls the filter's f_c. Try using the envelope as the only control voltage to a VCA, making the volume of a synthesized signal dependent on the level of the microphone output. With this patch, if you sing into the microphone while a synthesized voice is sent through the VCA, the envelope of the synthesized signal will follow the envelope of your voice.

Since the envelope output is pre-patched into the comparator input through the shunt of the comparator input jack, the gate output goes high every time the envelope goes above the threshold value, and stays high until the envelope drops below that value. This threshold can be varied from zero to +10V with the initial gate threshold pot alone; as well, any voltage applied to the threshold input jack adds its value to the pot value to give a total gate threshold value. One application of the gate is to initiate transients. Consider the following patch:



It is possible with this patch to remove sharp attacks or long decays from the microphone signal by setting the parameters on the TG appropriately. If an electric guitar is used instead of a microphone through the DP still), the plucked attacks can be removed from the guitar envelope, giving a radically different voice to the guitar.

In the above patch, the TG won't respond to changes in the microphone or guitar envelope once the transient is initiated. If this feature is desired, it can be obtained by will be independent, a keyboard bus will have to be used to input them to the TG). Now, an increase in the level of the microphone or guitar output can produce a trigger from the EF, thus initiating a new attack in the transient, even if the gate output of the EF doesn't fall. The sharpness of the envelope increase necessary to give a trigger is adjustable by the trigger sensitivity pot. With this pot full left, even the sharpest risetime available from the synthesizer (the VCO pulse rising edge) won't produce a trigger when applied to the comparator input. With the pot full right, the trigger is easiest to generate, requiring about a 50 Volt/second slope in the comparator input to give a trigger.

With a patch cord plugged into the comparator input, the pre-patch from the envelope output is defeated, and the gate and trigger functions are performed on the applied comparator input. The gate output is particularly useful here, since its value answers the question, "which of two voltages is greater?" (The two voltages in question are applied to the comparator and threshold jacks; the initial threshold pot can be regarded as an offset on the applied threshold voltage). One of many applications for the comparator function is to time the interval between events. If one event initiates a slow upward sawtooth ramp (using a VCO gate input), another event can be initiated when the ramp reaches a predetermined level by comparing the ramp to the desired level with the EF. This is the basis for a patch which allows the timing of a sequence to be stored in memory as well as the pitches of the sequence. This patch is discussed in detail later.

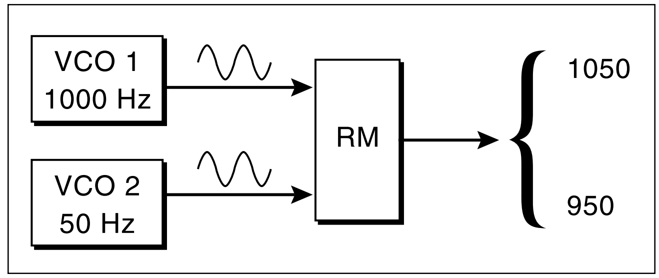
The trigger output will not respond at audio rates, since the trigger circuit contains a lowpass filter with f_c=16 Hz. The gate output, however, will run at audio rate, and this can be useful for generating new waveforms.

If an externally generated signal, say, an electric guitar output, is amplified by a DP and applied directly to the comparator input of an EF, the EF gate will go high during that portion of the waveform which exceeds the threshold voltage, and low otherwise. Thus, the input signal is digitized, or "squared" (not mathematical squaring, but shape squaring)—the effect is much like a "fuzz tone". In the case of the guitar, some lowpass filtering is necessary before squaring. Another use for audio rate comparison is changing the shape of a waveform in mid-cycle. If the EF gate operates an Analog Switch (q.v.), it is possible to obtain a waveform which is, say, sinusoidal for part of its period, and zero or, say, triangular for the remainder of its period. By adjusting the gate threshold voltage, the level at which the change occurs can be varied. (This same result can be obtained by using a separate sync'd VCO to provide the audio rate gate).

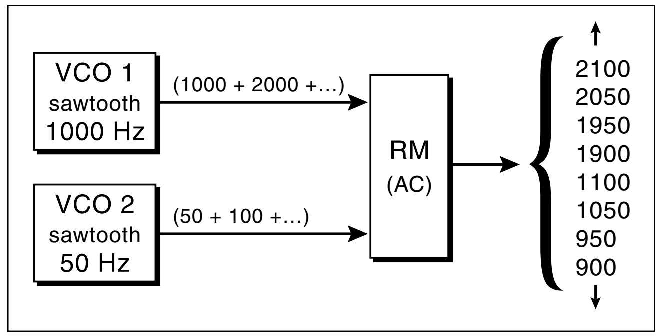
Another application of the EF is to convert a "click track" on recording tape into digital signals usable by the synthesizer. A click track is a recording of a sub-audio pulse wave, much like a metronome-in function, which enables sequences to be synchronized together on tape. Sometimes the clicks correspond to attacks on a transient generator, or digital signals used for other purposes. If the click track is amplified and applied to an EF comparator input, both the gate and the trigger go high when a click occurs. Either the gate or the trigger, possibly applied to a Dual One-shot (q.v.), will give a gate suitable for use by a TG external gate input or other digital input.

Ring Modulator

The ring modulator is typically used as a source of enharmonic spectra—complex spectra whose components are not harmonically interrelated. The name comes not from the sounds it makes, but from the diode ring used in early circuits. The output of the ring modulator is the algebraic product of the two inputs. The product of two sine waves is the sum of two new sine waves, with frequencies equal to the sum and difference of the original frequencies:

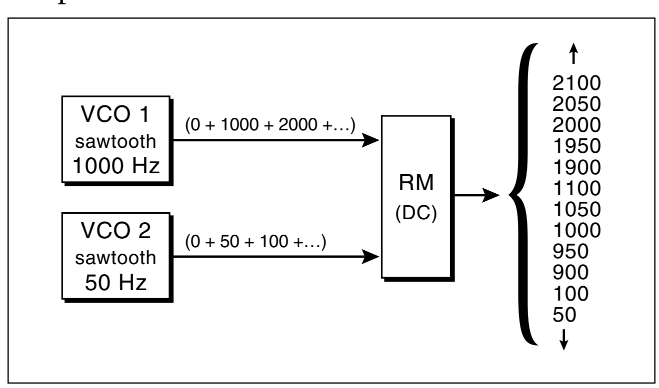


If more razzy waveforms are ring modulated together, the spectrum quickly becomes very complex. For example, with AC coupled sawtooth waves (which contain all harmonics), the above patch would look like this:



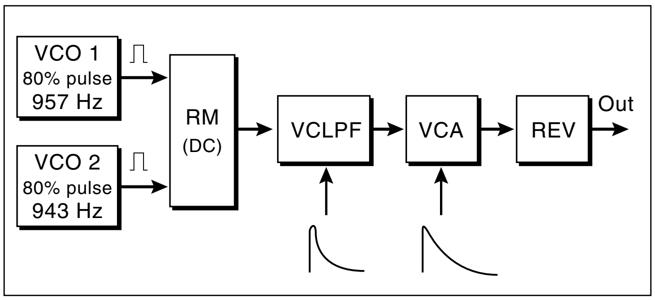
This is why the ring modulated output has a characteristic gong-like sound. A gong has non-harmonic resonances at different places along its surface, giving a similar effect.

Since Eμ sawteeth are 0-5V, they actually contain a 2.5V DC component In addition to the sinusoidal components. In DC mode, this 0 Hz component participates In the multiplication function, giving all the components of the inputs in the output, as well as their sums and differences. With either input in the AC mode, it is AC coupled before multiplication, causing the original components in the other input to be eliminated from the output. Compare the spectrum shown above for AC coupled sawtooth inputs with this one for DC coupled sawteeth:



Note that the 50 Hz output present in DC mode would theoretically appear even in AC mode as the difference between VCO 1's fundamental and VCO 2's 19th harmonic. In fact, though, the 19th harmonic of a sawtooth is so low in amplitude (about 25 dB below the fundamental) that the 50 Hz component in the output would be negligible in AC mode.

Here's a patch with two VCO's that gives a simple gong tone:

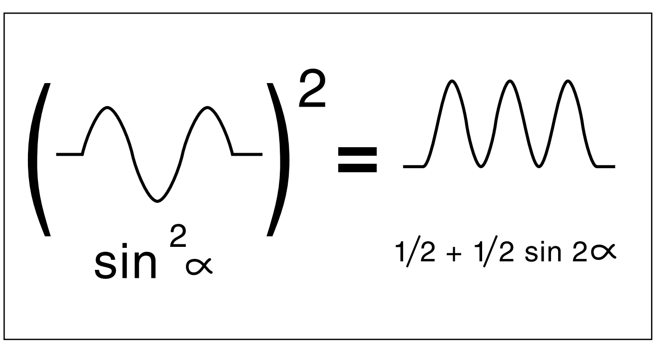


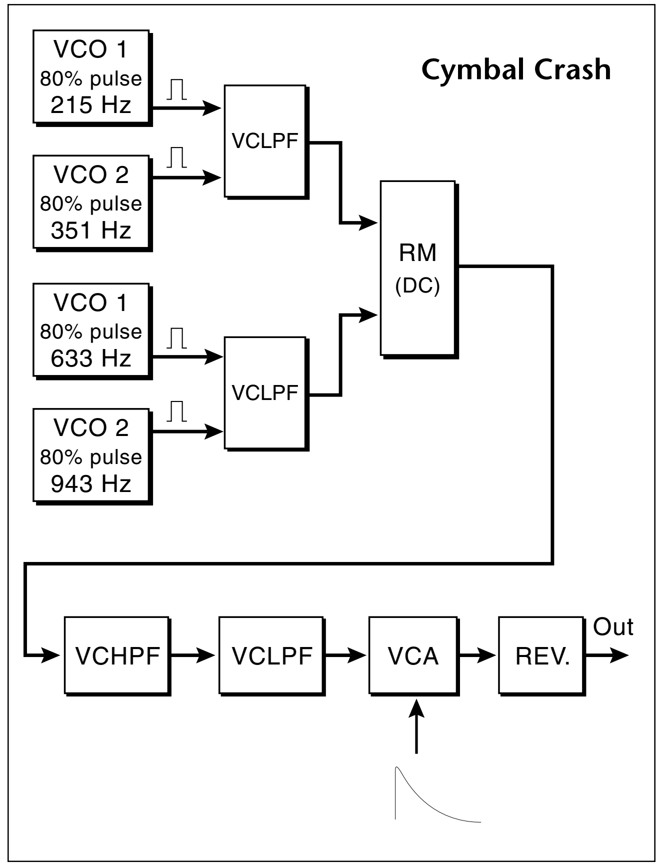
The initial settings were experimentally chosen for a good sound. Many other frequencies in similarly complex ratios will also produce bell-like tones.

The patch above with four VCO's gives a tone reminiscent of a cymbal crash:

Because of the non-harmonic ratios among initial frequencies and the razzy waveforms, the ring modulated output in this patch is almost as atonal as white noise, giving a more cymbal-like quality to the usual gong tone.

The ring modulator has many special-purpose uses, plenty of which we haven't found yet. By patching the same signal into both inputs, the square of the signal is outputted:



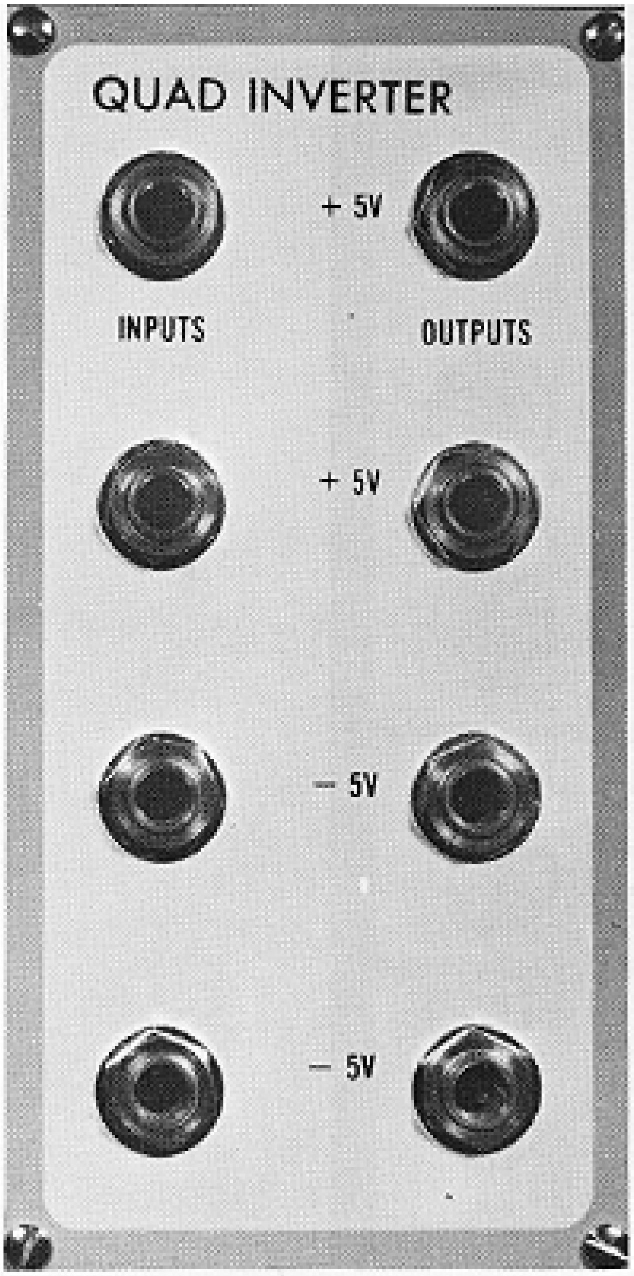


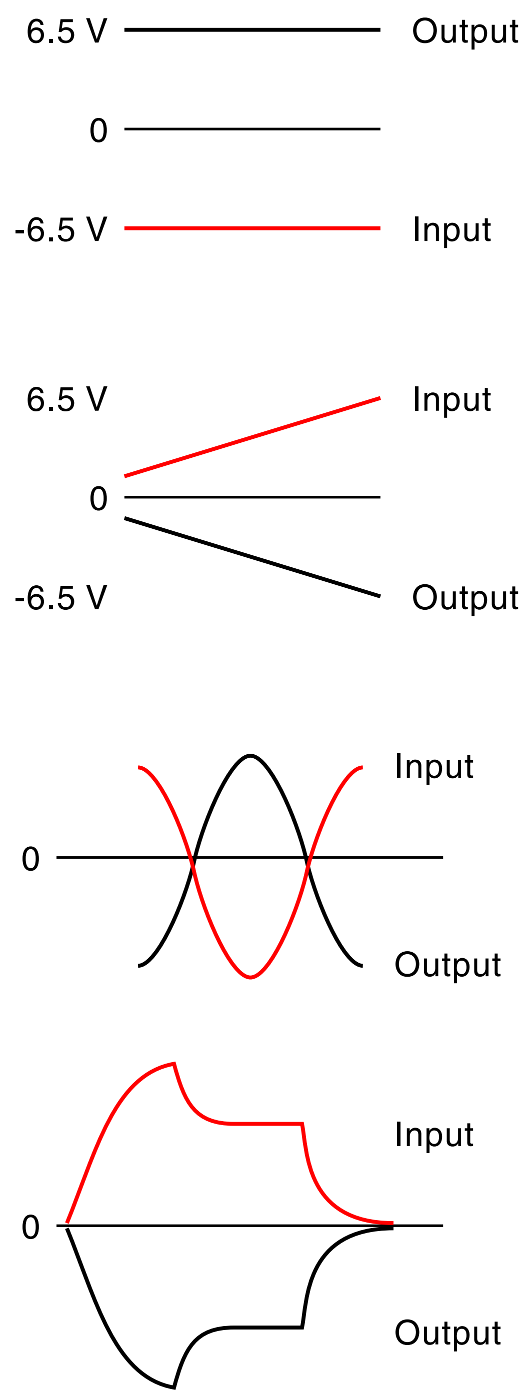
This has applications both for rectification and frequency doubling. By patching a signal in the carrier input and a control voltage on the modulation input, a special-purpose VCA is obtained. More precisely, this is a four-quadrant multiplier, since it accepts negative as well as positive control voltages (the output is inverted for negative control voltages).

The difference between the modulation and carrier inputs is that the carrier input is rejected by about 100 dB when the modulation input is at 0V, but the modulation input is rejected by only about 50 dB when the carrier input is at 0V. Clearly, for a VCA-type application, minimum signal leakage at 0V control voltage is desirable.

In general, ring modulation tends to be an unpredictable effect. Very slight changes in the inputs can make dramatic changes in the sound of the output; therefore, it is very open-ended for experimentation Try patching transient generators into inputs, or output into the frequency control of a VCO.

Quad Inverter





The 2450 Quad Inverter contains four independent analog inverters. When a voltage V is applied at an input, -V appears on the corresponding output jack. Since the inverters respond at any frequency from zero Hz to super-audio, they will invert any audio or sub-audio input. These drawings show a few possibilities:

The input jacks are pre-patched (via their shunts through precision resistors) to the +15V and -15V power supplies. The supply connections and resistor values are such that the upper two input jacks receive -5V through their shunts, and the lower two inputs receive +5V. Thus, +5V is available from either of the upper two outputs, and -5V from either of the lower two outputs when no patch cord is in the corresponding input jack. These fixed voltages have many uses; for example, they can be patched into a VCTGI module to offset the available range of one or more TG parameters, or -5V can be applied to an EF threshold input to give negative gate threshold values. The fixed +5V outputs are also convenient sources of digital “ones” (q.v.). If fixed voltages greater than +5V or less than -5V are needed, they can be obtained, respectively from a VSOU pot or the inverted output (Quad Inverter, of course!) of a VSOU pot. The QI is generally used to invert control signals, since the ear can’t distinguish an audio signal from its inverse. You have already been exposed to a number of patches that need an inverted control signal—variable bandwidth filtering, for one. Further applications will be encountered later.

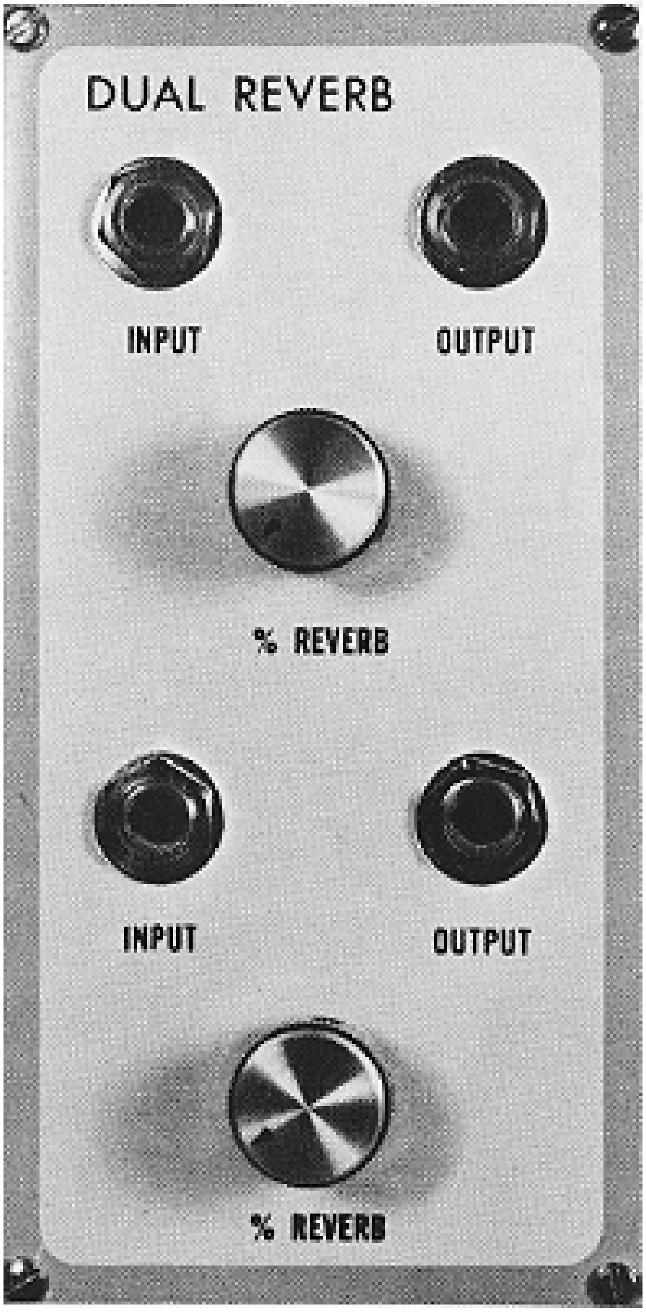
Remember that the inverse of an all positive signal (say, a transient) is an all negative signal; if such an all negative signal is used to control some parameter in the synthesizer, it may be necessary to turn up the initial setting of that parameter to keep its excursions within the desired range.

Dual Preamp

The 2420 Dual Preamp contains two completely independent preamplifiers for raising low-level signals (usually externally generated) to voltage levels usable by the synthesizer. If a low-level signal, say, from a microphone, is applied to the input jack of one of the preamps, it is amplified by a factor selected by the "gain" switch, either 20 dB (a factor of 10 in voltage), 40 dB (factor of 100), or 60 dB (factor of 1000). The maximum voltage available from this amplification stage is about 12V; thus, clipping will occur in the output if the selected gain is too high for the applied input level. After amplification by the switch-selected gain, the signal is attenuated by the "level" pot; the attenuated output is available from the output jack. With the attenuator turned up all the way, no attenuation occurs, and the total amplification factor is equal to the gain set by the gain switch. As the attenuator is turned down, the overall gain is lowered, until the output is completely off at the full left position. Notice that clipping, if it occurs, can't be eliminated by turning down the level pot, since the pot comes after the amplification stage; clipping must be eliminated by switching to a lower gain. (Sometimes, clipping is desirable, for certain timbres or for shape-squaring of a waveform. The use of an EF for this purpose was discussed earlier).

The DP has other applications when used in patches involving only synthesized signals. Clipping of synthesized signals can be interesting—clipped noise is noticeably different from the normal noise output; as well, filter oscillations (like the UAF percussion mode), which are only sinusoidal, can become razzy timbres if they are clipped with a DP. The VCLPF description mentioned a patch for voltage-controlling the Q of the VCLPF using a negative feedback loop. If a DP is used to increase the gain in this loop, very high Q values (even into oscillation) can be produced and voltage-controlled.

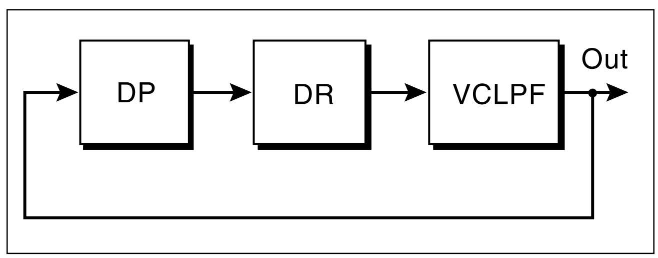
Dual Reverb



The 2460 Dual Reverb contains two independent spring reverberation units. The two reverb trays are located separately (generally wrapped in foam in the back of the synthesizer cabinet), and they are connected with shielded cables to the DR module board. The DR uses the High Gain Amplifier submodule as does the DP, with the exception that the gain of the amplifier is fixed in the DR. When a signal is applied to one of the DR input jacks, it is AC coupled and sent to the appropriate reverb tray. The output of the reverb tray is amplified by one section of the HGA submodule, and the result is mixed with the original signal in a proportion set by the "% reverb" pot. The mix is available at the DR output jack. With the "% reverb" pot full left, no reverb output is included in the final mix output; at the full right position, no original signal is included.

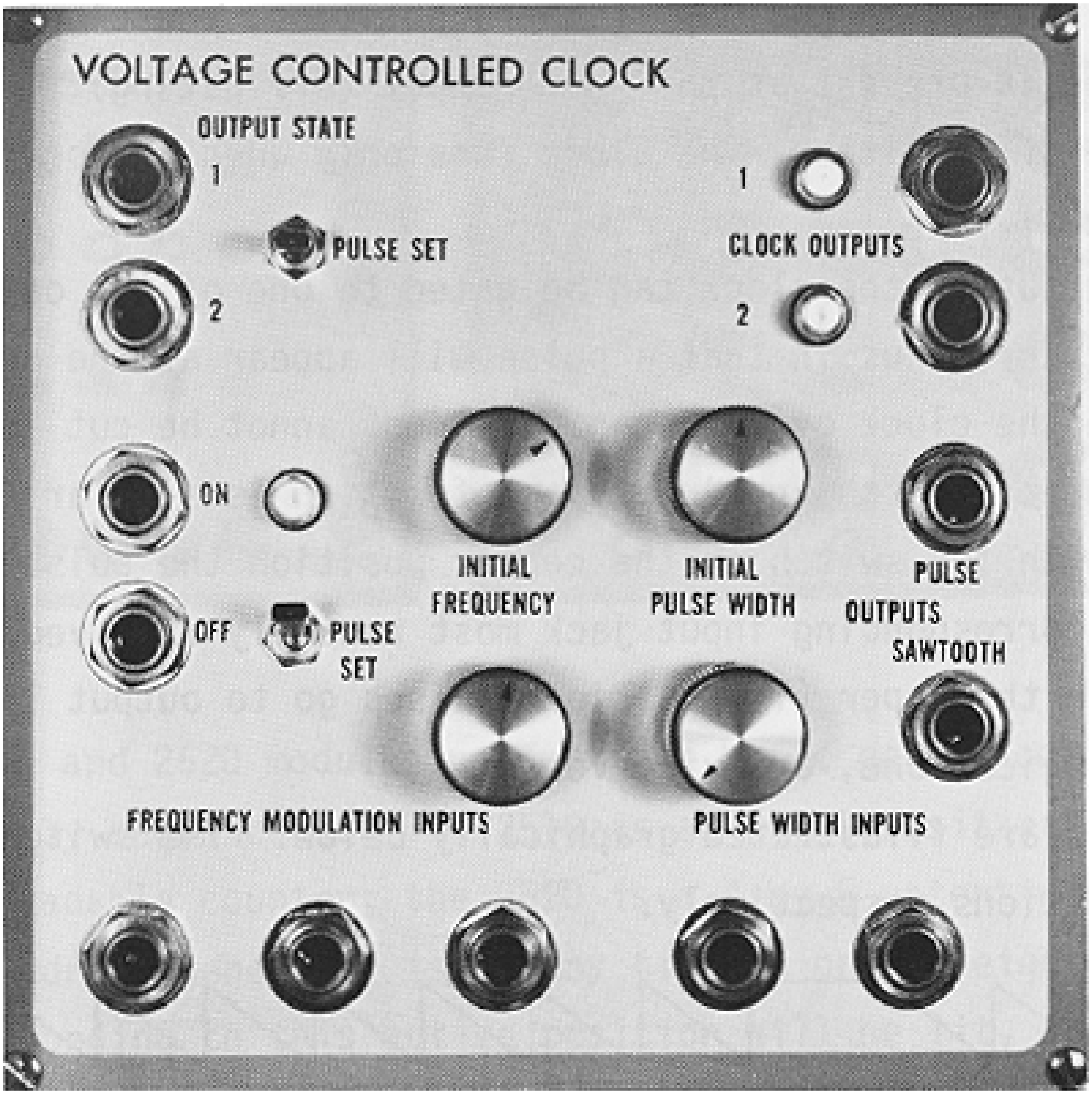
Notice that the overall loudness of the DR output falls off somewhat as the percentage of reverb in the mix is increased. The reason for this is the uneven frequency response of the reverb units—the gain of the HGA must be adjusted for the frequencies with the highest amplitude in the reverb output, so that these frequencies won't be distorted by too much gain. As a result, the amplitude at other frequencies will be lower, and an overall decrease in power will occur.

Aside from the obvious use of the DR for producing reverb, it has an obscure application that may be of interest. The frequency response of the reverb contains a large number of resonant peaks scattered throughout the audio spectrum. For this reason, the reverb makes a good "resonant cavity", which can oscillate if fed back upon itself. Consider the following-patch:



The output of this patch can be a very bizarre sound whose pitch and timbre are drastically affected by the f_c of the filter and the "% reverb" setting on the DR. If you like the feedback effect, try voltage-controlling it. Remember that you can obtain voltage controlled reverb percentage simply by mixing the 100% reverb output with the original signal in a QVCA.

Voltage Controlled Clock



The 2500 Voltage Controlled Clock is a sawtooth/pulse VCO designed primarily for low frequency (0.001 Hz to 500 Hz) clocking of sequences. It contains a 1500 Voltage Controlled Clock submodule with the following unique features: a digital pulse output, synchronous on/off gating, and synchronous routing to either of two output channels. The module also outputs routed pulse and sawtooth waveforms. The frequency of the VCC oscillation is voltage controllable at roughly 1V/octave on the unattenuated control inputs.

Consider first the on/off gating: this can operate in three different modes, determined by the lower of the two switches. With this switch in the upper position (normally on), the VCC is on unless a digital one is present on the “off” jack. With the switch in the lower position (normally off), the VCC is off unless a digital one is present on the “on” jack. With the switch in the center position (pulse set), the VCC remains on or off depending on which jack received the most recent digital one. (If a digital one is present on both inputs simultaneously, in any of the three switch positions, the VCC is off).

When the VCC is first turned on, the LED above the switch lights up, a falling edge occurs at the sawtooth output, and a rising edge occurs at the pulse outputs. (The unrouted pulse output and the selected output channel are completely identical). The pulse width, or time spent in the digital one state, is pot- and voltage-controllable (10% per volt maximum). The pulse width will not go to the 0% and 100% extremes as will the VCO pulse width. As long as the VCC is on, a minimum width pulse is always issued every cycle, even at very negative total pulse width control voltage; similarly, even at very positive PW control voltages, a minimum width low time will always occur. When the VCC is first turned off, the LED immediately goes off, but the waveforms finish the cycle in progress. After completing the final cycle, the sawtooth output remains at its peak value (about +5V). and the pulse outputs stay in the digital zero state. (If you try to make the width greater than 100%, the pulse outputs remain at digital one when the VCC is turned off. In that case, the falling edge, and an immediate rising edge, occur when the VCC is turned on again.)

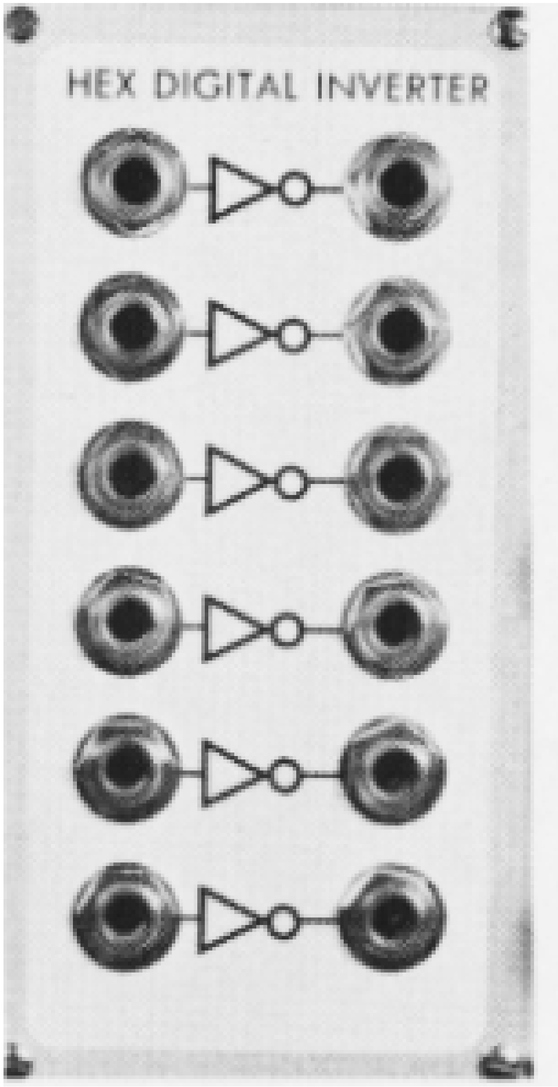
The channel selection switch and jacks operate in a manner similar to their on/off counterparts—channel 1 (upper), channel 2 (lower), or pulse set (center). When the pulse output is switched from one channel to the other, the cycle in progress is allowed to finish on the original channel. The next rising edge appears on the other channel.

For many simple sequencing operations, a VCO pulse output is just as good as a VCC output. Often, the high frequencies available from the VCO make it the only alternative. However, certain applications will require the on/off, or routing features of the VCC; other applications will require the TTL digital type output (q.v.) For example, the digital zero, as was mentioned in the previous chapter, is a current sink; as such, the clock output can be used in wire-and functions (q.v.). The VCO pulse output, of course, will not perform a wire-and function, since its output stage is not active-low. The routing feature has several interesting applications that we have found, plus all those we haven’t found: try patching one channel into the “up clock” and the other into the “down clock” of an MAG or 8AG. With this patch, you can clock either up or down through the same sequence; or synthesize step-wise waveforms that are symmetrical. Try patching one clock channel into one MAG or 8AG, and the other channel into a separate MAG or 8AG—by re-routing the clock output, you can hold one note of either sequence while continuing the other sequence.

Details of these and other patches are best left for later, when you know more about all the sequencer modules. (You’ll often find that patches that seem pretty simple to a fancy computer like your brain will turn out to require a maze of patch cords to execute in actual hardware!)

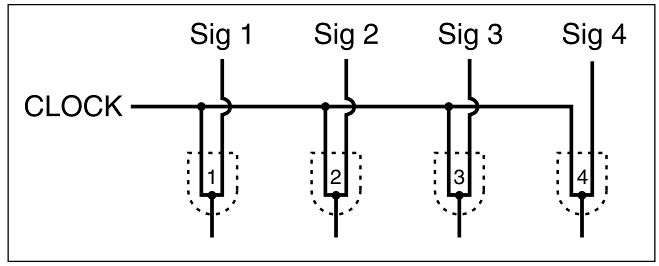
One special-purpose application of the VCC is its use as a digital “one-shot” (q.v. Dual One-Shot). Because of the synchronous on/off gating, a fast pulse or trigger applied to the “on” jack (while the VCC is normally off) will cause the VCC to run for one whole cycle. The duration of the resultant high output can be varied with the frequency and pulse width controls; remember that the pulse output stays high until the beginning of the next cycle if set over 100%. The one-shot obtained with this patch can have a much longer output pulse than can the DOS module—as well, the output pulse width is voltage controllable.

Hex Digital Inverter



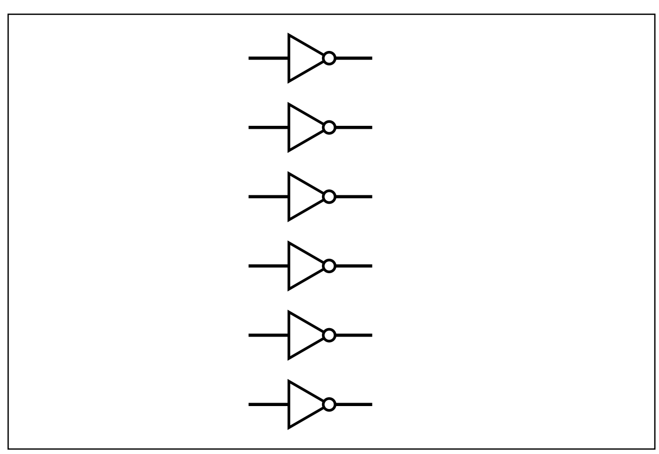
The 2550 Hex Digital Inverter contains six identical logical inversion circuits. If a digital one (any voltage greater than 2.5V) is applied to an input, a digital zero appears on the corresponding output; conversely, an input voltage less than 2.5V gives a digital one at the output. (As such, when no patch cord is plugged into an input, the output of that channel is high). The inputs are pre-patched together as in the QVCA, with the tip of each input jack wired to the shunt of the jack below it. This allows the HDI to serve as a digital multiple with buffered, or isolated, outputs. To understand this feature, imagine a digital patch in which a VCC output has to be wire-anded separately with several other independent digital signals.

Suppose we patched it like this:

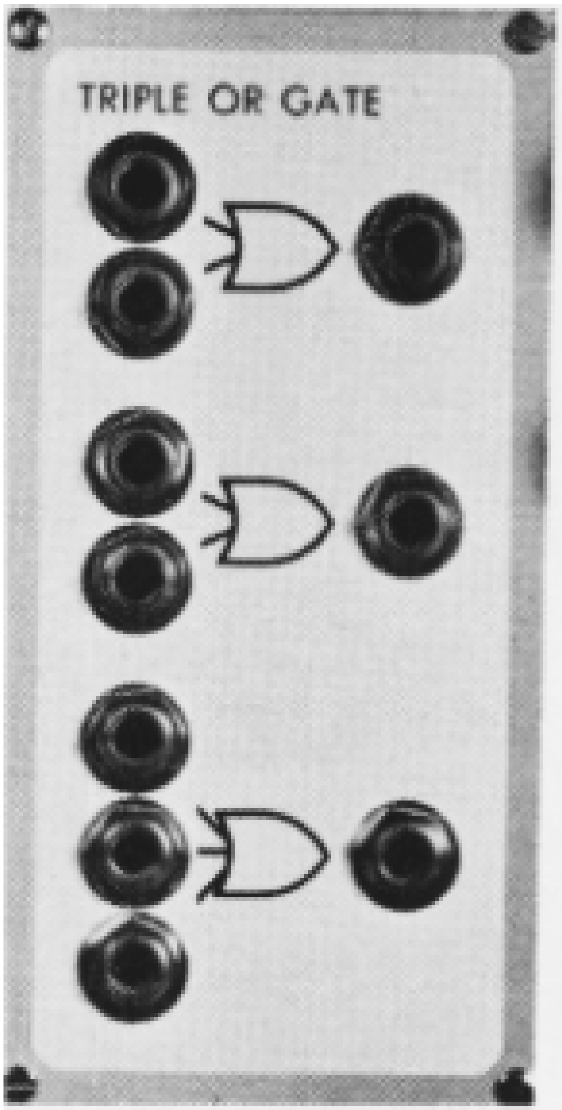


If a simple multiple is used to distribute the clock output, a problem arises: if any of the input signals 1 through 4 go low, not only does the output the corresponding “and”-gate fall, but the clock itself and all the other and-gates are pulled low as well. To make the four “and”-gates independent, the clock input to each one must be isolated from the clock itself. This is done with an HDI as follows: patch the VCC output into the uppermost channel of the HDI, and patch the output of that inverter into the input of the second inverter. The remaining five HDI outputs are exactly equivalent to the VCC output (since the VCC output is inverted twice), but any of these five clock signals can be pulled low by a wire-and function without affecting the clock itself. An actual application of the HDI as a buffered multiple will be discussed in connection with the 2545 Memory.

The HDI can also be used as a shape-squarer for analog inputs; consider that a 10V peak-to-peak sinusoidal input to one of the digital inverters would give a 5V pulse waveform with 33% pulse width at the output. The symbol for a digital inverter is , thus an HDI can be diagrammed like this:



Triple OR Gate



The 2551 Triple Or Gate (accent on the or) contains three independent “or”-gates; two have two inputs, and the third has three. The output of each .”or”-gate is a digital one if any of the inputs is a digital one, and zero only if all the inputs are zero. Specific applications will be discussed as they arise—in general, the TO is used whenever a digital event is to occur in response to one of several other digital events.

Notice that digital signals require an “or”-gate to perform the “or”-function, but can use a multiple to perform the “and”-function. Conversely, analog signals require an “and”-gate to perform an “and”-function (a ring modulator will work for pulse waveform), but can use a multiple to perform the “or”-function (commonly called “mixing”).

As an amusing exercise, try building an”exclusive-or”-gate from modules. The exclusive-or function gives a low output when the two inputs are the same (either both zero or both one), and a high output only when the inputs are different (one low and one high). Hint—you’ll need a TO, a multiple, an HDI and 11 patchcords.

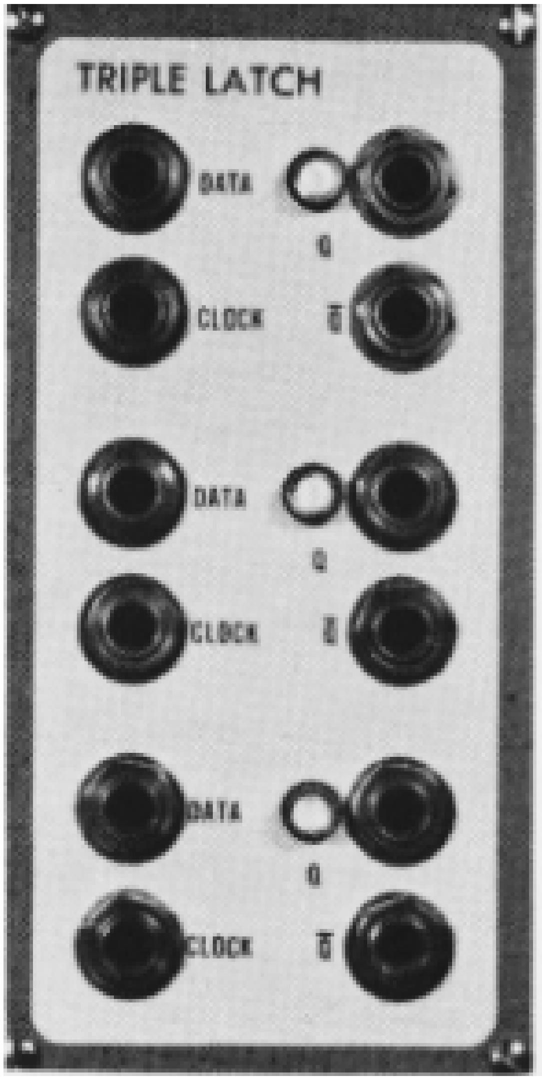
OR

	0	1
0	0	1
1	1	1

Exclusive - OR

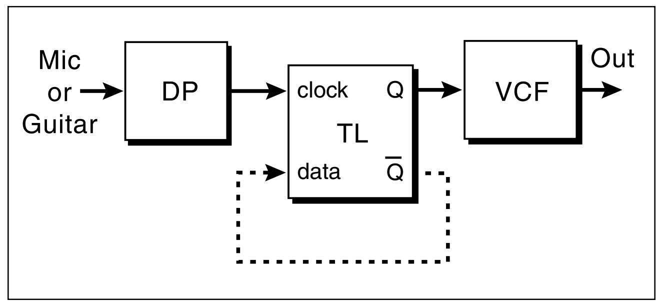
	0	1
0	0	1
1	1	0

Triple Latch



The 2552 Triple Latch contains three identical digital latches. A latch is essentially a digital sample & hold; when the "clock" input changes from digital zero to digital one, the digital value currently applied to the "data" input is sampled and held, or "latched", at the Q output; and its digital inverse (its "complement"), is latched at the $\overline{Q}$ output (pronounced "Q-bar'). These outputs remain indefinitely at the digital values most recently latched.

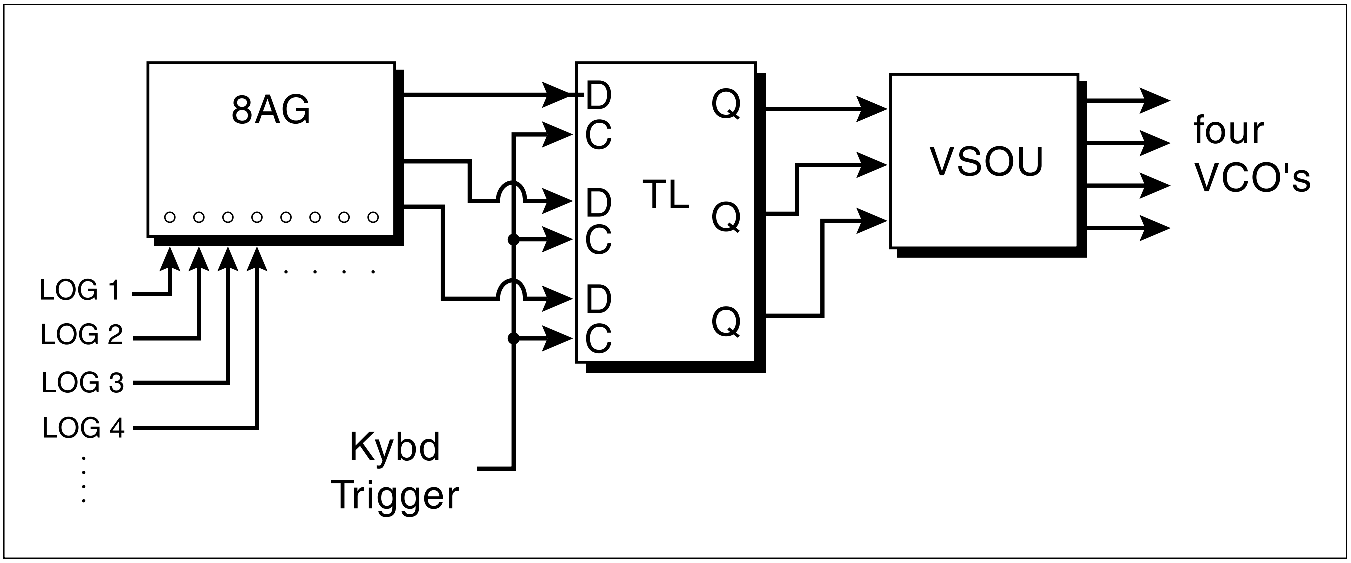
A pre-patch exists from each $\overline{Q}$ output to the shunt of the data input in the same latch. This allows each latch to be used as a divide-by-two circuit for a train of regularly spaced pulses applied to the clock input, say, from a VCC, VCO, or an external source. The explanation for the divide-by-two function is clear with a little study—say the latch has Q high ($\overline{Q}$ low) and "clock" low. If the pre-patch is in effect, "data" is also low. At the next rising edge on the clock input, the digital zero on the data input will be latched—that is, Q will go low and $\overline{Q}$ will go high. Because of the pre-patch, "data" is now high, and the next clock rising edge will latch a one at Q and a zero at $\overline{Q}$. Thus, Q goes high (and $\overline{Q}$ low) every other time the clock input goes high—this is equivalent to dividing the frequency of the clock input by two. Try the following patch:



The output signal will be one octave lower in pitch than the input signal, provided the gain of the DP is high enough to give excursions greater than 2.5V in its output. If either Q or $\overline{Q}$ is patched into the "clock" of the second latch, its Q and $\overline{Q}$ outputs are then two octaves below the original signal. This division can be performed indefinitely with more latches.

The triple latch can also be used as a ring counter (000, 001, 010 100, repeat) for sequencing through the four channels of an Analog Switch. A three-stage ring counter (001, 010, 100, repeat) is quite easy to obtain—patch the clocking signal into all three clock inputs, and the Q output of each latch into the data input of the next latch. (Bring the third Q back to the first "data"). Getting the all-zero condition necessary for a four-stage ring counter is a little trickier—try figuring it out yourself. Hint—wire-and the three $\overline{Q}$ outputs to detect the all-zero condition, and feed a digital one back into the beginning of the ring counter only when the all-zero condition occurs. (Notice that a ring counter has to be started in a particular state for it to operate properly.)

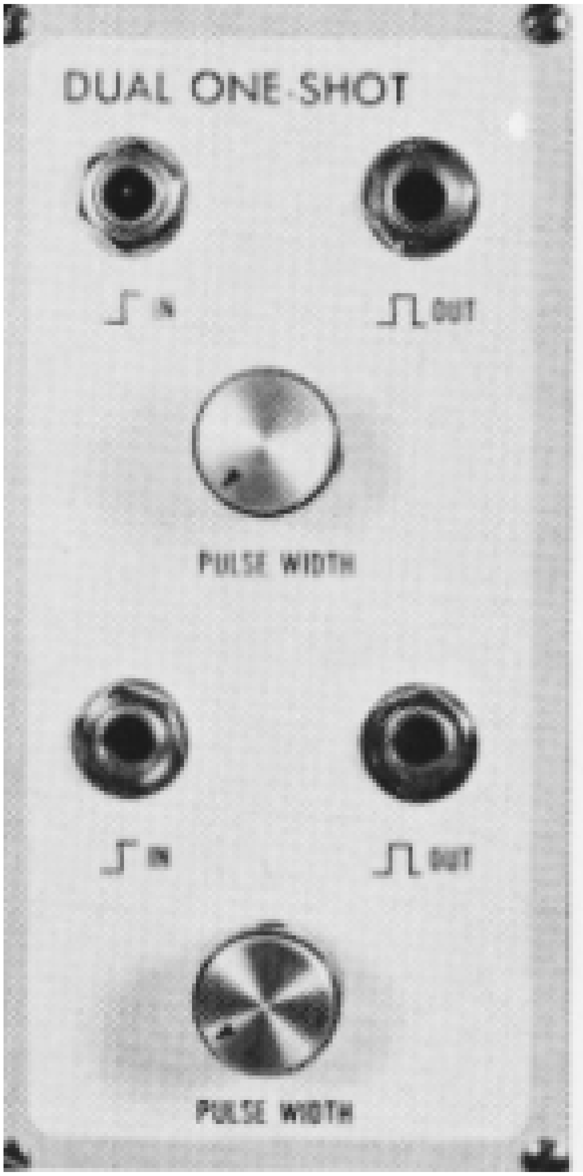
The TL can, of course, be used simply as a three-channel digital sample & hold. Consider the following patch:



This patch turns a monophonic synthesizer into a four-note chord organ. The intervals in up to eight desired chords are set on the VSOU, and the keyboard control voltage is summed at each VCO with the appropriate VSOU output to transpose the selected chord into the desired key. Of course, the address from the 8AG could be patched directly into the VSOU. However, if you wanted to change chords simultaneously with a key change, you would have to coordinate the LOG depression perfectly with the depression of the desired key.

Interposing the TL between the 8AG and the VSOU allows the next chord change to be keyed in at any time; the next keyboard trigger then performs the actual change of chords synchronously with the key change.

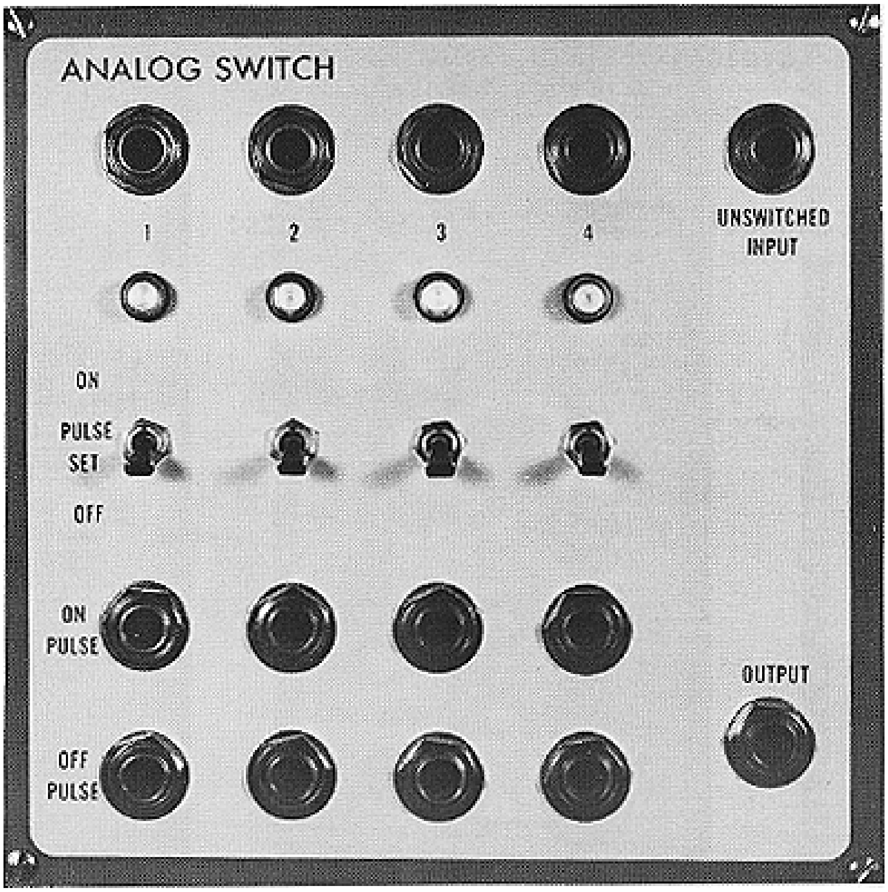
Dual One-Shot



The 2553 Dual One-Shot contains two identical independent pulse forming circuits. When the voltage applied to an input rises through about 4V, the output goes to a digital one, and stays at the high level for a time determined by the "pulse width" pot. The pot varies the pulse width exponentially from about 10 msec full left to about 10 sec full right. Since the DOS output is digital, its low state is a current sink, so it will perform the wire-and function with other digital signals. The DOS is completely oblivious to the falling edge of its input, and to rising edges which occur while the output is high. In other words, the 2553 has "non-retriggerable" one-shots.

Consider this application of a DOS—suppose one address of a memory has digital output 1 high, say, for the purpose of gating a TG. If this memory is clocked rapidly, the high level at digital output 1 may last only a fraction of a second. Of course, one way to prolong the TG gate is to program a digital output to be high in several consecutive addresses; however, if a variable-width gate is desired, it can be obtained by patching digital output 1 into a DOS and using the one-shot to gate the TG.

Analog Switch



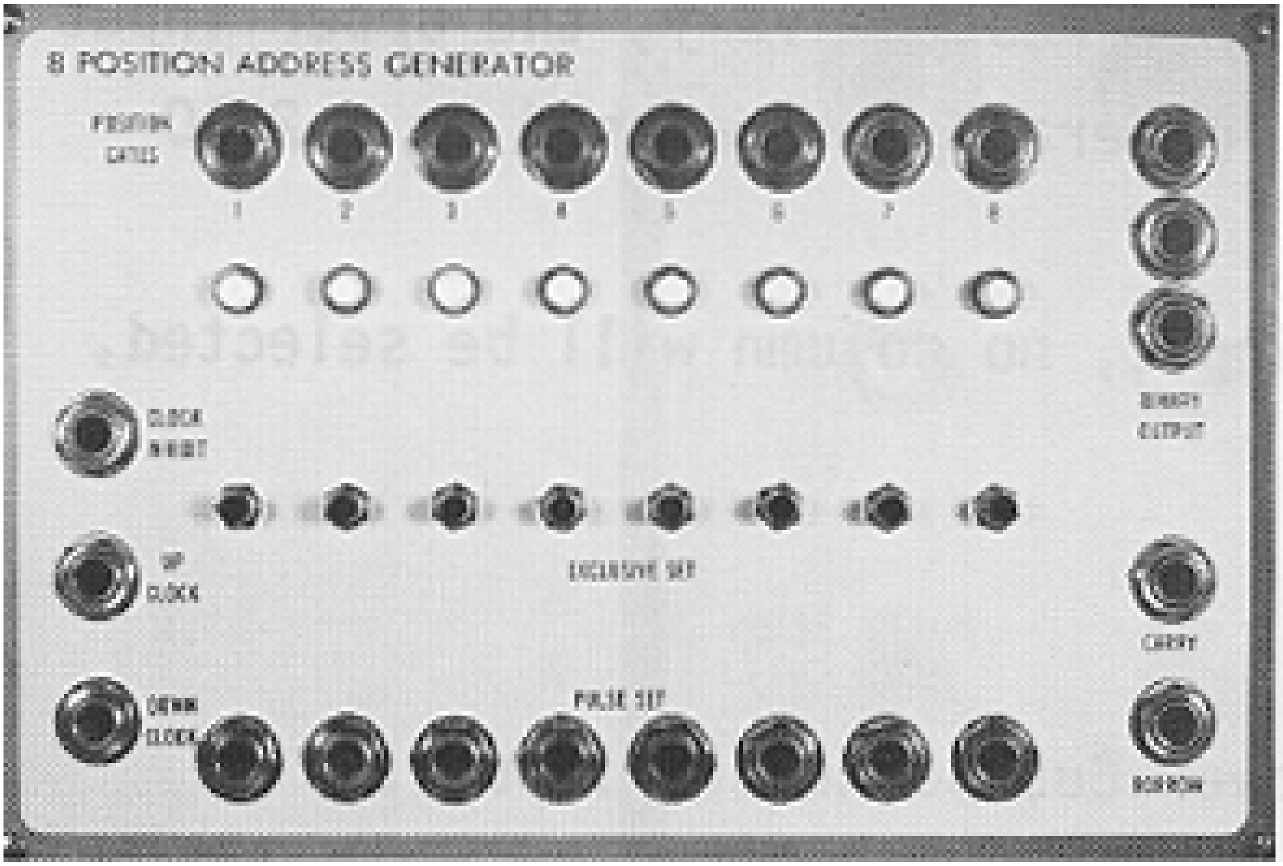
The 2530 Analog Switch allows up to four analog input channels per module to be summed in any digitally selected combination. The analog signal present on the “output” jack is the sum of the analog signals on the “unswitched input” and the inputs of all activated channels; a lit LED indicates that a channel is activated. The digital on/off control of each channel operates like the on/off and output select controls of the VCC: the switch has three positions, normally on (up), normally off (down), and pulse set (middle). A normally on channel is off only if a digital one is present on the “off pulse” input; a normally off channel is on only if a digital one is present on the “on pulse” input; a channel in pulse set mode remains in the state most recently pulsed.

The AS is particularly useful for cascading sequences or digitally selecting among several simultaneous sequences—these applications will be discussed as you learn more about sequences.

You may notice obvious similarities between the AS and the QVCA (q.v.), with the primary difference being that an AS channel is either on or off, while a QVCA channel has variable gain. Of course, a QVCA can also respond in an “all or none” fashion to digital signals an its control inputs. As well, the QVCA has the lag time feature which can remove the pops caused by the rising and falling edges of digital control signals. However, the QVCA lacks the built-in latches of the AS pulse set node; in addition, the QVCA has only three channels in each module available for cascading, since one channel must be used to pass on the output of the previous QVCA. (The AS has the unswitched output for this purpose.)

In any case, the digital pops don’t occur unless digital controls are changed in the middle of a note—in sequencer patches, this rarely needs to happen. The convenience of the AS switching scheme (no need to adjust pots and set different control polarities), and its lower cost, make the AS attractive for many applications.

Eight-Position Address Generator



The Eight-Position Address Generator and the Voltage Source Output Unit together comprise the type of sequencer commonly used in most synthesizers— the potentiometer matrix. However, because of the separation of the two modules, and certain special features, the 8AG/VSOU pair is somewhat more versatile than other pot matrix sequencers.

Consider first the 8AG: in a typical/sequencing application, its primary function is to count from one to eight (actually 0 to 7 in binary) in response to a clock input. Every time the “up clock” input rises through 2.5V, the 8AG advances to the next address. To understand what this really means, suppose we start the 8AG in position 1: only the left-most LED is lit, only the left-most “position gate” is at a digital one, and the three “binary output” jacks are all at digital zero. Now apply a voltage greater than 2.5V to the up clock input—the first LED goes off and the second from the left lights up; the first position gate goes low and the second goes high; and the binary output (from top to bottom) changes to 0,0,1. At each successive rising edge of the up clock input, the LED and position gate previously high go low, the next LED and position gate to the right go high, and the number represented by the 3-digit binary output is increased by one:

Position	Binary Output (top to bottom)
1	000
2	001
3	010
4	011
5	100
6	101
7	110
8	111

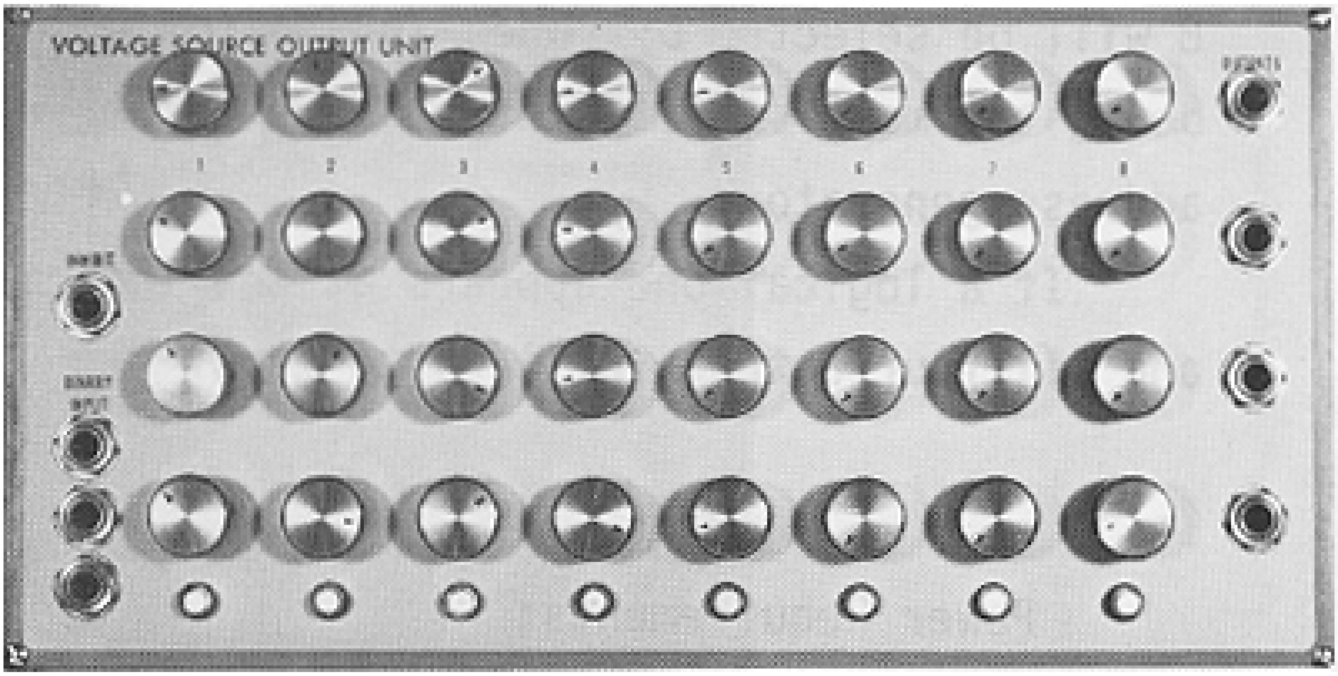
Notice that the digital bit on the bottom jack (the "least significant bit") changes every clock pulse, the middle bit changes every other clock pulse, and the bit on the upper jack (the "most significant bit") changes every four clock pulses. After position 8, the 8AG goes back to position 1 on the next up clock rising edge. It is thus possible to clock the 8AG with a periodic waveform (typically pulse from a VCC or VCO) and repeat a regular sequence of addresses indefinitely. By applying such a periodic clocking signal to the "down clock" input, the sequencing will occur from right to left (higher to lower addresses) instead. If a digital one is present on the "clock inhibit" input, rising edges applied to either the up clock or the down clock input will be ignored.

The clock inputs are not the only way to change the address of the 8AG. The "exclusive set" pushbuttons (so called because they override the clock inputs while depressed) allow the desired address to be selected manually. Alternatively, a digital one on a "pulse set" input holds the 8AG on that address. Notice that if the position gate at address 5 is patched to the pulse set input for address 1, the 8AG will respond to a periodic up clock input by sequencing through the first four addresses only. Similarly, if position gate 3 is patched to pulse set 6, the sequence will skip addresses 3,4, and 5.

When the 8AG is in position 8 and an up clock pulse is received, a single short 5V pulse is issued on the "carry" output. Similarly, when the 8AG is in position 1 and a down clock pulse occurs, a short pulse appears on the "borrow" output. These pulses can be taken off one 8AG for clocking a second 8AG, or they might be used to clock a ring counter which cycles through the four channels of an Analog Switch. This last patch is one way to use a VSOU as a 32x1 pot matrix instead of 8x4—more on that later!

Aside from its use in conjunction with a VSOU, the 8AG has a few interesting applications in its own right. The position gates allow the 8AG to be used as a ring counter; compared to the TL used as a ring counter, the 8AG has the advantages of being simple to use and expandable to eight positions, whereas the TL has a maximum of four channels and is quite cumbersome for more than three. The 8AG can also perform a divide-by-n function (n up to 8). For example, if the 8AG is patched 'to sequence through five positions, any one of the included position gates will go high once every five clock pulses. The 8AG also has a rather obscure application as a type of latch, in that a pulse applied to one of the pulse set inputs will cause the associated position gate to remain high indefinitely until a different position is pulsed.

Voltage Source Output Unit



The VSOU is a potentiometer matrix having eight columns with four pots in each column. At any point in time, only one column of the VSOU is selected—the voltage output from each of the four pots in the selected is available from the output jack located to the right of its row. The voltage output of the pots is linearly variable from zero to 10V. Which column is selected (indicated by a lit LED) is determined by the binary address applied to the 3-bit "binary input". (The order is the same as on the 8AG—least significant bit on the bottom). If a digital one is applied to the "inhibit" jack, all the LED's go off and all four outputs go to zero volts.

Suppose the binary address output of an 8AG is patched bit by bit to the binary 1 address input of a VSOU—as the 8AG is clocked, each column of the VSOU will be selected in turn, and each of the four VSOU outputs will go through an independent repeating sequence of eight voltages determined by the pot settings in each row. If these four outputs are patched to the F.M. inputs of four VCO's, four independent 8-note ostinato's (pitch sequences) will occur. If VCA's or a QVCA are placed after the four VCO's in such a patch. TG's controlling the VCA's can be gated by some of the position gates of the 8AG, thus accentuating and/or eliminating certain notes in each ostinato. All four channels need not be used for VCO's; try controlling the 8AG clocking rate with one of the rows of the VSOU. At each new position, the clock will run at a different preset rate, giving the sequences adjustable rhythm. (All four rows will have the same rhythm, of course.) Using the inhibit input of the VSOU, you can, at any time, substitute the initial setting of each VCO for the preset pitch in the sequence without having to change any pot settings.

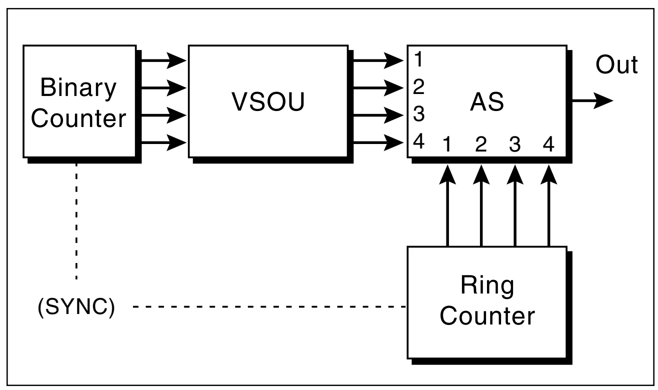
Often you may want to sequence through a set of eight chords in some pattern other than simply left to right or right to left. Because of the binary address feature, this is possible in any conceivable way. As an extreme example, you might patch the pulse outputs of three VCO's into the three binary input jacks; the result can be very nearly random in pattern and rhythm, particularly if the VCO's aren't sync'd. Perhaps a musically more useful method of changing the order of a sequence is to process, in some fashion, the binary output of an 8AG before applying it to a VSOU binary input. Try patching the most significant bit of the 8AG output into the least significant binary input jack of the VSOU, and vice versa. Leave the middle bit patched the same.

The result is as follows:

8AG Position	8AG Output	VSOU Output	VSOU Position
1	000	000	1
2	001	100	5
3	010	010	3
4	011	110	7
5	100	001	2
6	101	101	6
7	110	011	4
8	111	111	8

Similarly complex hashing of a sequence can be accomplished by inverting (with an HDI) one or more of the 8AG output bits before applying it to one of the VSOU input jacks. For example, if all three bits are inverted and the complements applied to the input jacks in the correct order, the VSOU will sequence backwards when the 8AG sequences forward. Horrendous modifications are possible when "wire-and" and/or "or" functions are included. With a clever patch, well-programmed VSOU pots, and some attention paid to coherent composition rules, the synthesizer can compose listenable music by itself!
(Well almost.)

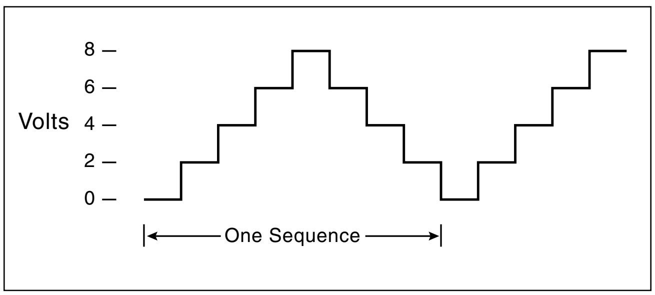
So far, we've treated the VSOU only as an 8 x 4 matrix. There are several patches that turn it into a 32 x l matrix (that is, one channel 32 positions long). An examination of a few of these patches will illustrate the principle that there are often several ways to perform a given function, and some may be much simpler than others. Generally speaking, the 32 x 1 matrix is obtained by sequencing columns 1 through 8 while looking at row 1 only, then sequencing all eight columns again while looking at row 2 only, then the same for rows 3 and 4. After all four rows have been scanned, the patch must start counting over again from column 1, row 1. To accomplish this feat, some form of binary counter must be patched into the binary input of the VSOU, and some form of ring counter must sequentially scan the four VSOU outputs using an Analog Switch. Clearly, the row selection must change synchronously at the end of each cycle of the VSOU:



If you recall, the Triple Latch can be used either as a binary counter or as a ring counter. Similarly, the 8AG can be used for either function. Four methods immediately come to mind to implement our patch—two 8AG'S, two TL's, or one 8AG and one TL. Say we decide to use a TL as the binary counter; its all zero condition must be the signal to advance the ring counter one step. There is a very easy way to detect the all zero condition; the most significant bit of an 8-position binary counter (look at the binary table for the 8AG), goes low every time the count starts over. As such, its complement goes high at the same time. Thus, the third Q output of the TL in our patch can be used to clock the ring counter directly. If the ring counter is an 8AG (with position gate

5 patched to pulse set 1), the third Q output of the TL need only be patched into the "up clock" input of the 8AG. If two 8AG's are used instead, the carry output of the binary counter 8AG is used to advance the ring counter 8AG. Study all four methods. Which one requires the fewest changes to make the 32-position sequence reverse direction? How would you make a 16 x 2 pot matrix?

An additional application of the 8AG/VSOU pair is stepwise waveform synthesis. Suppose the following sequence of voltages is preset on one row of a VSOU: zero volts, 2V, 4V, 6V, 8V, 6V, 4V, 2V. If a clock rate of 4000 Hz is applied to the associated 8AG, the signal on the VSOU output jack will be a stepwise approximation of a 500 Hz triangle wave:



With slight filtering or linear lag processing, the corners can be largely smoothed out, if desired. The VSOU so used is a musically powerful sound source capable of generating timbres unavailable from any other module. The frequency of the clocking VCO can be controlled with a keyboard to make the effect playable; and the 8AG can be clocked as fast as a VCO will oscillate. If the waveform is defined by 16 or 32 points instead of 8, still more complex waveforms can be synthesized, though the maximum output frequency is progressively lowered as the number of steps per cycle increases.

Digital Sequencer

Memory Address Generator

In the digital sequencer, the 2540 Memory Address Generator has a function analogous to the 8AG in the 8AG/VSOU analog sequencer: to output an address corresponding to one of 256 (or 512 with expanded memory) positions in a memory, and to sequentially increment (or decrement) its address output in response to up (or down) clock pulses. The MAG also contains internal up and down clocks, which can be activated with the pushbuttons on the front panel; holding the appropriate pushbutton depressed causes the MAG to "fast forward" or "fast reverse", allowing you to reach a desired address easily. Quick taps on these pushbuttons will cause the MAG to step through addresses one at a time.

The current address of the MAG is indicated by the LED display. Notice that this display is in octal, or base eight; thus, after 7, the counter goes to 10, after 77 it goes to 100, etc. Position 255_{ten}, the highest address, is 377 in octal (written 377_{eight}). We've heard a few groans already, so some explanation of this feature is in order. Computers (and memories) are, of course, binary devices. Binary numbers are hard to read, however, so a higher base is necessary for displays. Base ten, though ideal for our convenience, is not at all convenient for the electronics, since ten isn't a power of two. Alternatively, if octal is used for the display, the conversion of binary into the higher base is simple:

base two:	1	0 1 0	1 1 1	0 0 0	1 0 0
octal:	1	2	7	0	4

Each octal digit represents exactly three binary digits ($2^3 = 8$), so the conversion is electronically easy—a decimal read-out would substantially increase the cost of the MAG. In any case, you can often use the octal read-out to good advantage, since traditional Western music is often written in two, four, or eight. Say you put a composition into memory in 4/4, with eighth notes being the shortest notes in the sequence. If you started the composition in address 000, the first beat of each successive measure would fall in addresses 010, 020, 030, 040, etc. This can greatly aid the programming of sequences, as you will see later. When necessary, the octal read-out can easily be converted into decimal, and vice versa. The right-most digit in the read-out is the "ones" column, the middle digit is the "eights" column, and the left-most digit is the "sixty-fours" column.

Thus,

$273_{eight} = (2 \times 64) + (7 \times 8) + (3 \times 1) = 128 + 56 + 3 = 187_{ten}$

and

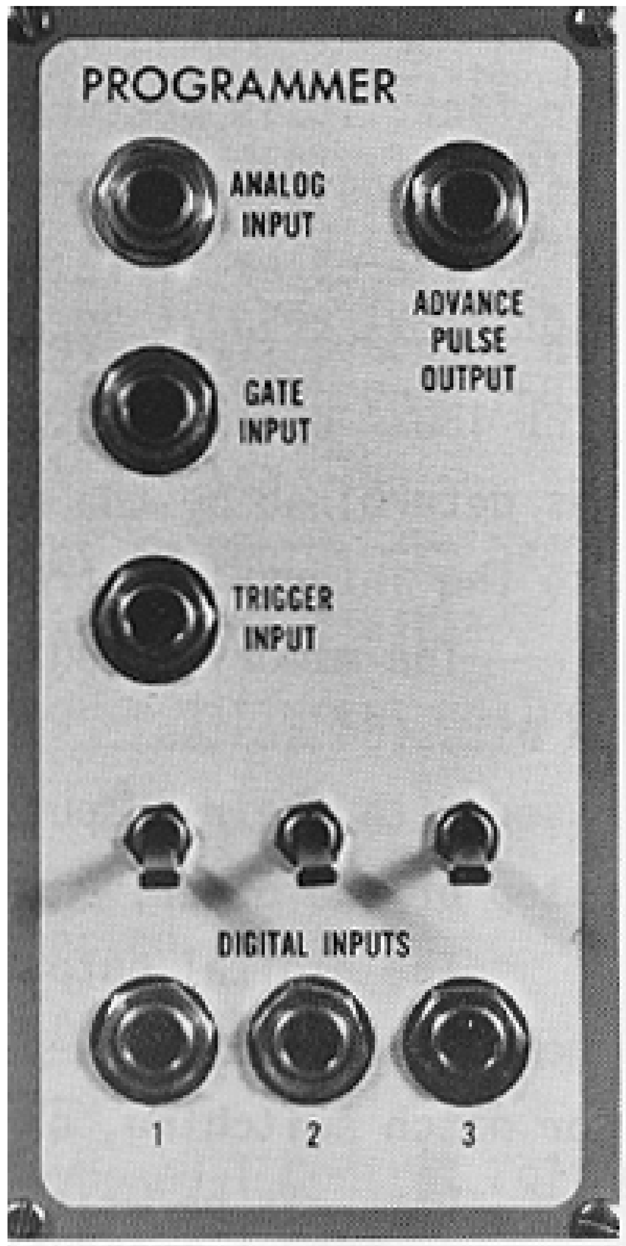
$200_{ten} = (3 \times 64) + (1 \times 8) + (0 \times 1) = 310_{eight}$

The MAG "clock inhibit" input, when high, causes up and down clock pulses to be ignored. The "reset" input, when high, and the reset pushbutton, when depressed, cause the MAG to reset to address 000. The MAG remains at 000, regardless of clock inputs, as long as the pushbutton is depressed or the reset input is a digital one. The reset input is the means by which a sequence occupying less than 256 addresses can start itself over without a break in the rhythm-more about that later!

The output of the MAG is an eight-bit binary address ($2^8 = 256$), which is conveyed to all the associated Memory modules (up to about 50) by means of a ribbon cable running from the MAG submodule to the memory submodules. At each of its 256 addresses, a memory is capable of storing eight bits of information, or one "eight-bit word". The eight-bit word in each address is actually viewed by the circuitry in two portions: one portion of six bits, which determines an analog voltage output for that address, and another portion of two bits, which determines which one of four digital outputs will be high at that address. First, consider the six bits worth of analog information in each address. Six bits allows the memory to specify only 64 different voltages, since $2^6 = 64$. Since the 4000 Keyboard has 61 notes, it is logical to limit the voltages which can be stored in a memory to the 61 semitone voltages of the keyboard, plus the next three semitone voltages higher than the 61st note (which aren't available from the keyboard unless its output is offset). Since the memories are generally programmed from a keyboard, this limitation makes sense. However, the memories could also be programmed from continuous voltage sources (say, pots on a VSOU); if the voltage you try to program falls between two semitone voltages of a keyboard output, the voltage actually programmed will be the semitone voltage closest to the applied voltage. If you try to program a voltage less than zero volts or greater than 5.25V (the "64th key" of the keyboard), the voltage actually programmed will be zero volts.

In fact, since the keyboard is originally a digital device anyway, the information stored in a memory is best regarded as an equivalent key position

Programmer



The 2546 Programmer has three inputs which are used in the programming of control voltages from a keyboard. These are the “analog input”, “gate input”, and “trigger input”. These inputs are pre-patched (through the shunts) to make programming easier, as follows: the keyboard control voltage and gate are patched to the programmer’s analog input and gate input, respectively. Also, the programmer’s gate input is internally pre-patched to the shunt of its trigger input. For the moment, defeat this last pre-patch by patching the keyboard trigger into the programmer’s trigger input. (You’ll understand the purpose of the pre-patch later). Patch the “advance pulse output” of the programmer into the “up clock” input of the MAG. Switch the “write enable” switch of one or more associated memories into the enabled position (up)— the LED above the switch will light, indicating the memory can now be programmed. So that you can hear what you’re programming, either directly or from the memory, set up a voice on the synthesizer and patch the “fixed 1/12 V” output of the memory into the 1V/octave input of the VCO in the voice. (Patch it into the VCF, too, if you want the VCF to track with the VCO). By switching the VCO off the keyboard bus, you can now play back the programmed memory only; alternatively, by switching the VCO to the keyboard bus and pulling the patch cord from the memory, you can use only the keyboard output.

The functions performed by the programmer in this patch are as follows: when a trigger occurs, the programmer looks at the voltage present on its analog input, converts that voltage into a binary word corresponding to the equivalent key position, and programs that binary word (through a ribbon cable) into any write-enabled memories, in the address currently specified by the MAG. When the gate falls, a short pulse is issued on the advance pulse output, causing the MAG to advance to the next address. Before you can actually program the memories, the keyboard must be switched to “preset” mode, the portamento must be off (we still-get caught on this one, occasionally), and the keyboard offset must be exactly zero volts.

The easiest way to set the keyboard offset to zero is to compare the frequency of a VCO off the keyboard bus to its frequency on the keyboard bus with the lowest key depressed—adjust the offset pot until they are the same. You are now ready to program. Switch the VCO to the keyboard bus, unplug the F.M. input from the memory, reset the MAG, and start playing!

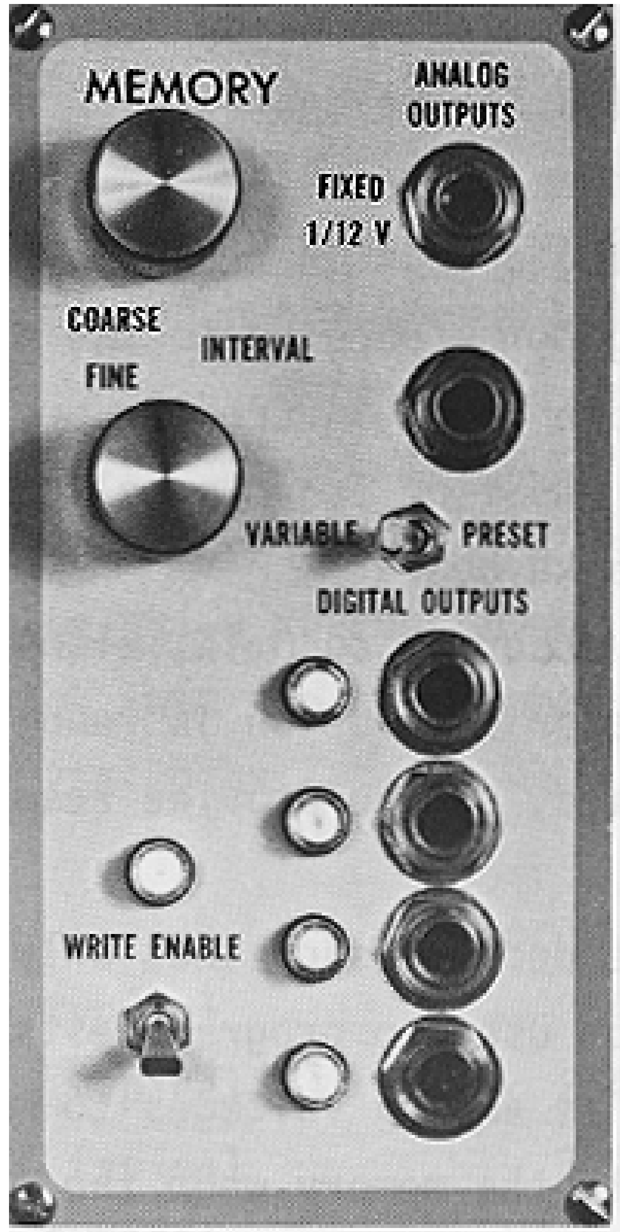
Notice that each time the gate falls, the MAG advances to the next address; thus, when you let up all the keys, the address shown by the LED display is the next address in which you will write. Because the gate and trigger inputs to the programmer are independent, you can stay on the same address and change notes as often as you wish, just as long as you don’t let the gate fall. If you don’t want this feature, you can save a patch cord, since the gate input is pre-patched to the trigger input. Programming then occurs only at a gate rising edge. In either case, if you make a mistake while programming you can always go back with the down clock pushbutton and correct it.

After you’ve programmed, say, 20 addresses (the MAG display will read 24_{eight} at this point), turn off the write enable switch to protect your sequence. Reset the MAG, switch the VCO off the keyboard bus, plug the memory output into the VCO, and clock through the first 20 addresses by hand. You will probably hear nothing, since there is no signal gating the TG’s in your voice patch; either listen directly to the VCO, turn up the VCA, or use a clock signal to simultaneously clock the MAG and gate the TG’s. You should now hear the sequence you just programmed in addresses 000-023, and whatever you last programmed in 024 and above. (If nothing has been programmed in a memory since the machine was last powered up, it stores random garbage. In other words, the memories are “volatile”—they forget everything when the power goes down.)

Instead of using the “fixed 1/12 V’ output of the memory, try patching the lower analog output into the 1V/octave input of the VCO. This output has two modes, selectable by the adjacent switch. In “preset” mode, the output is identical to the “fixed 1/12 V” output—

that is, two adjacent key positions programmed into memory give respective analog outputs exactly 1/12V apart, so that the output of memory can give even-tempered tuning with a VCO's 1V/octave input. In "variable" mode, however, the analog output voltage interval between the same two programmed key positions can be varied from zero volts to about 2/12V. Since the programming of a memory must be done with an even-tempered keyboard, sequences using intervals other than multiples of a semitone must be done using this variable interval output. Notice that the fixed 1/12V output could be patched into an attenuable F.M. input of a VCO instead of the 1V/octave input for intervals smaller than a semitone, and it could be patched into two F.M. inputs on the same VCO for larger intervals; the variable interval output is just a simpler way to accomplish the same thing, and it allows the output of a memory to be used by several VCO's and/or VCF's without requiring delicate adjustment of each module.

Memory Module/ Digital Outputs

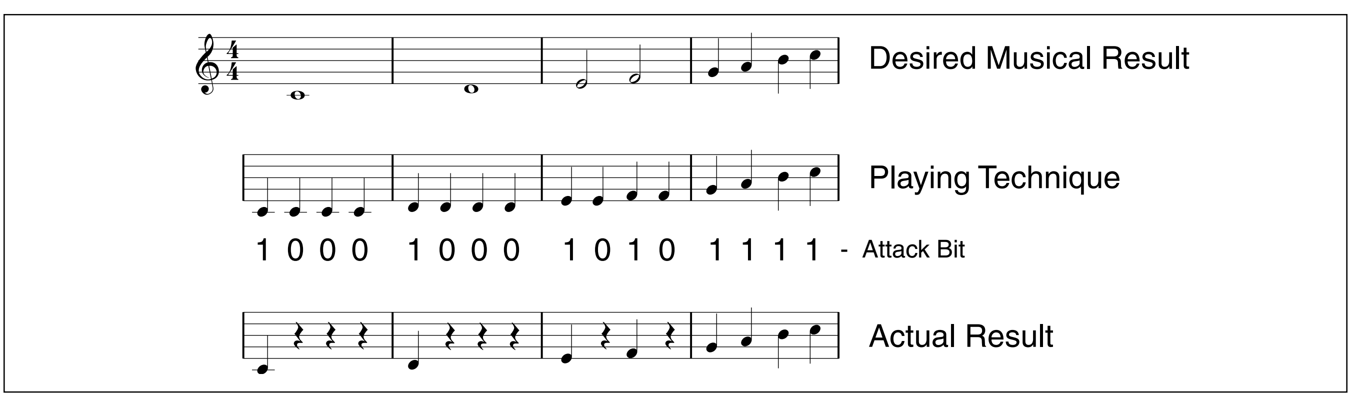


As was mentioned earlier, six bits of the eight-bit word in each address are converted into an analog voltage output for that address, and are a digital representation of the programmer's analog input at the time of programming. The remaining two bits of the eight-bit word in each address determine which of the four digital outputs is high at that address ($2^2 = 4$), and are a digital representation of which one of the programmer's "digital input" jacks was high at the time of programming. A high level (>2.5V) on digital input 1 when a trigger is received causes any write-enabled memories to be programmed with the uppermost digital output high-at that address. (An analog voltage will, of course, be programmed at the same time). Similarly, digital output 2 (or 3) can be programmed high by triggering the programmer while a digital one is simultaneously applied to its digital input 2 (or 3). The lowest digital output jack, digital output 4, is the default position, and is programmed high if digital inputs 1, 2, and 3 are all low when the trigger occurs. Thus, in your sequence of 20 voltages, all 20 addresses 000-023 will have digital output 4 high (indicated by a lit LED). If you want the sequence to reset after address 023, you could program digital output 1 to be high in address 024, and patch this digital output to the reset input of the MAG.

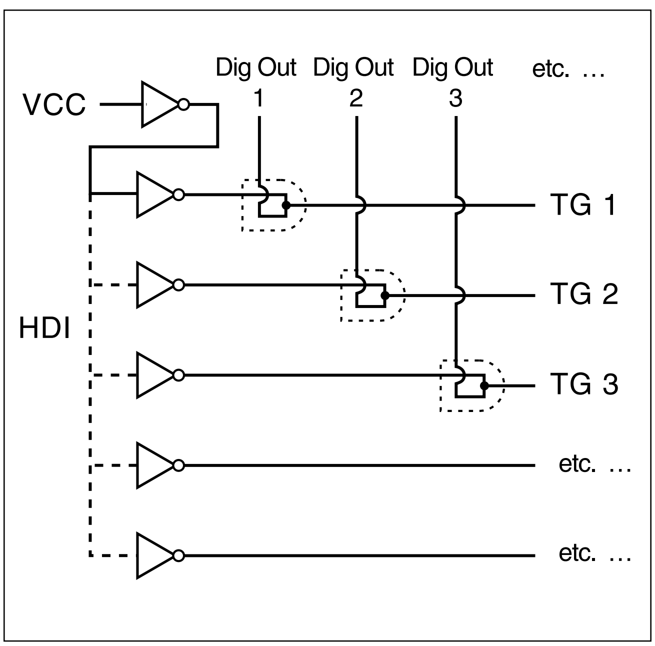
To program digital outputs, we generally switch the low octave of the keyboard out and patch three low octave gates to the three digital inputs of the programmer. Thus, control voltages are programmed with keys in the upper four octaves, and digital outputs are programmed by simultaneously depressing a low octave key. To program the desired reset bit in address 024, simply depress the appropriate low octave key while the MAG is in address 024, and tap any high key to give the required trigger to the programmer. (Since the memory will stay at 024 for only about 10 µsec, you don't care what analog voltage is stored there). Make sure you don't have digital output 1 patched to the reset input when you program it high, or you'll reset to 000 in the middle of the programming operation and strange things will result. Make sure also, that you never have a high resetting output in address 000, or the MAG gets stuck there until you either pull out a patch cord or reprogram the offending memory.

Digital outputs can be used for a variety of other purposes besides resetting the MAG. We mentioned earlier the idea of externally gating the TG in your sequencer voice with the same clock that clocks the MAG; but suppose you don't want attacks on every note of a sequence. Clearly, you need to store in each address, in addition to the control voltage, the information as to the presence or absence of an attack on that note. If digital output is used for the reset bit, another digital output, say, #2, can be used for this attack bit. Try going back over your 20-note sequence and programming attack bits at digital output 2. You will, in fact, be programming a rhythm in the sequence by appropriate placement of attacks. If the voice includes several transients on different parameters, use a multiple to gate them all with digital output 2, or gate some with digital output 2 and others with digital output 3 (programmed differently, of course). To keep from losing the already stored analog information when you program the digital outputs, patch the analog output of the memory back into the analog input of the programmer. Thus, when you program the digital outputs, each analog voltage will be reprogrammed into the same address. (Similarly, to reprogram analog values without losing digital cues, patch the upper three digital outputs into the programmer's three digital inputs).

Because of the programming of attacks by the use of the digital outputs, it is not necessary for the memories to store the duration of each note. The following diagram shows how a rhythmic passage can be reproduced from a memory:



Notice that the playing technique is such that the shortest note in the sequence (quarter note, in this case) occupies one address in memory. To play a whole note, the same key is depressed four times, but an attack bit is programmed only on the first quarter note. Similarly, a half note results from two quarter notes of the same pitch in adjacent addresses, with an attack bit only in the first of the two. (The reason this method works at all is that there is no discernible pop in the memory output when it advances from one address to the next). When you try to program attacks on the four quarter notes, a problem arises—if the TG is gated directly by the attack bit, it will see a high gate that lasts through all four addresses. The result is an attack on the first quarter note only, and a sustain through the others. To obtain the desired attack on each note, the attack bit can be "and"-ed with the clocking signal in a multiple, and the "and"-function used to gate the TG. Since the clock is low for part of each cycle (or -5V if it comes from a VCO), the "and"-function of the clock with the attack bit will also go low in the latter part of each address period, thus giving a rising edge (an attack) at the beginning of each address period in which an attack bit is present. If several digital outputs are all gating TG's, all will need to be "wire-and"-ed with the clock. To do this, you'll need to multiply and buffer the clock output with an HDI, as was discussed in the HDI chapter. Here's the patch:



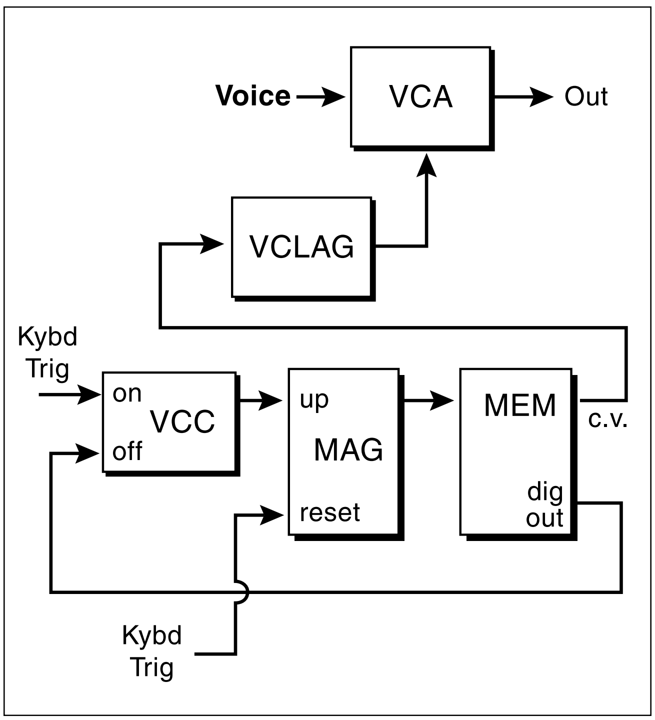
Sometimes, of course, you may want the option of programming a particular digital output high in several consecutive addresses for the purpose of sustaining a note or rolling over notes without attacks. To have both types of gating available, we generally use digital output 2, "and"-ed with the clock, to gate one TG, and digital output 3 to gate a second TG directly. Thus, staccato, legato, and sustains are all possible, depending on how you program the two gating bits.

Clearly, programming rhythmical sequences requires a modification of normal keyboard technique. Typically, the right hand plays the desired passage "metronomically", that is, at the rate of the shortest note in the sequence, while the left hand depresses low octave keys to program digital bits. It takes a bit of practice, but turns out to be quite natural after a while. The payoff comes both when you compose as you play (in which case, you don't want to have to keep perfect time), and also when you can't play what you have in mind (for example triplets and quintuplets in the same measure). Alternatively, there are patches that allow the duration of notes to be stored at the same time as the control voltages, either with two programmers or several addresses per note. These will be discussed in detail later in this chapter.

Resetting and TG gating are common uses for the digital outputs; other possible applications include squelching a VCA or cutting short a final decay (with a QI and a VCTGI) for the purpose of obtaining silent rests in a sequence. Also, digital outputs can be used to select channels of an Analog Switch; the AS can then sequentially scan several voices or several memories running simultaneously, just as it can scan through the four outputs of a VSOU. Arbitrarily long sequences are thereby possible. The four digital outputs can be programmed sequentially in addresses 000-003, and a reset signal can be obtained from the analog output (any voltage greater than 2.5V. programmed from the keyboard); the result is a four-position ring counter. (There are easier ways, however, as you know!) Digital outputs can also be used as analog control voltages fixed at 5V; for example, the clock may be made to run faster at certain addresses by patching the appropriate digital output into an attenuable F.M. input of the VCC or VCO.

Other Applications

Now that you basically know the standard applications of memories, a few more esoteric ideas deserve mention. First, realize that memories store control voltages, and these voltages can control anything. VCO's and VCF's are the most obvious (complex filter-f_c sequences can be interesting); a less obvious one is patching a memory output into a VCA control input to give a complex envelope. For this, you could specify, say, 50 voltages in a sequence, and cycle through all 50 during the evolution of each note. Here's a sample patch to give point-by-point transient synthesis:



The keyboard trigger does two things in this patch—it resets the MAG, and, through the on-pulse input of the VCC (VCC in pulse set mode), it turns on the clock. The end-of-sequence bit in memory, through the VCC off-pulse input, turns off the clock. Thus, a key depression causes the memory to output its sequence from 000 to the address containing the end-of-sequence bit, and then to stop. Retriggering in the middle of a transient causes the sequence to start over, just as retriggering a TG gives a new attack. Notice that the end-of-sequence bit need not be used to reset the MAG, since, in this patch, resetting is necessary only when a key is depressed. Depending on the clock rate and the length of the sequence, the overall transient duration can be varied (voltage-controllably) over the entire useful range. The VCLAG takes out the "ripple" in the transient resulting from its stepwise nature. Such processing of a memory output with a VCLAG is often useful; for example, to remove the pops from a filter-f_c sequence, or to give voltage-controlled portamento in sequences.

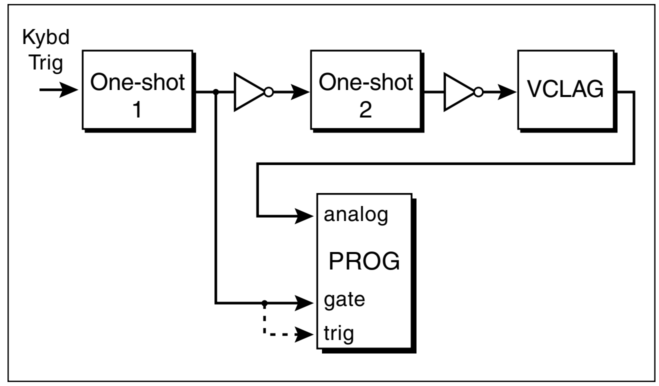
An obvious extension of the stepwise transient is the use of memory for stepwise waveform synthesis. This was discussed in connection with the VSOU already; the memory merely allows more points to be specified per cycle. Actually, if you try to specify 256 points, two problems will arise—first, even if the clocking VCO runs at 50 kHz, the synthesized waveform will be only about 200 Hz maximum; second, if you try to reset the memory after only 100 points or thereabouts (to allow for higher frequencies), you'll find that, at 50 kHz, the reset bit doesn't last long enough to perform the reset function. Clearly, fewer points should be specified.

Occasionally, you may find yourself limited by the fact that a memory stores only six bits of analog information. Suppose you want the full five-octave range in a sequence, but you want to be able to include quarter tones or eighth tones as well. This can be done with two memories whose outputs are summed (in an Analog Switch, for accuracy). Use one memory's fixed 1/12V output and the second memory's variable output, adjusted so that its entire analog voltage output range (from key 1 to key 64) is only 1/12V. Thus, the fixed-output memory is a "coarse tuning" control, and the variable-output memory gives 64 divisions within one semitone. The sum of the two memory outputs, patched into a VCO's 1V/octave input, can give all the 128th tones over the entire five octaves. Quarter tones, of course, would be easy!

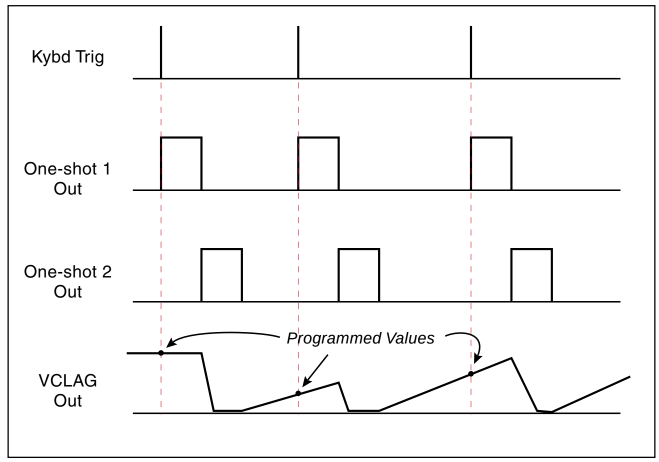
The ability of a memory to specify only semitone values is the basis for an interesting effect. Patch a keyboard control voltage into the programmer's analog input, and a VCO pulse waveform (about 30 Hz to start), into the trigger input. Leave the associated memory write-enabled in a single address (no clock). With the keyboard portamento on, control the frequency of a synthesizer voice with the memory's fixed output. As you change keys, the input to the programmer slews to the new value, but the output of the memory steps through each intervening semitone between the two keys. As long as the triggers are coming much faster than the time between semitone steps, the stepwise glissando obtained from the VCO will include every semitone. This patch is perhaps the only way to obtain a perfectly chromatic glissando.

The above patch should make it clear that you can simultaneously read and write into a memory—simply leave it write-enabled. Thus, you can obtain a sequence from memory which repeats the most recently played notes—sort of an endless tape loop. Make sure, when you try this, that you don't accidentally erase your reset bit, or the memory will reset at a later time than you expect (either after 377 or when the appropriate digital output happens to be high again.)

As was mentioned earlier, it is possible to program the duration of notes played on the keyboard, as well as the control voltages. The basic principle is to initiate a linear upward voltage ramp with every key depression, and to store in a memory the voltage level of the ramp when the next keyboard event occurs. In one form of the patch, one programmer and one memory are devoted to storing analog voltages which are linearly proportional to the duration of the notes played. Another programmer and memory are used in the usual way to record control voltages, and need no further discussion. The MAG in the patch is common to both memories. Here's a simple version of the timing portion of the patch:

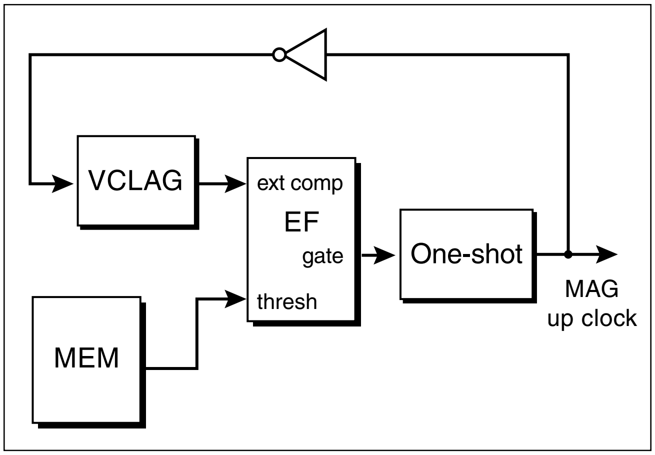


The following diagram shows the synchronization among the different signals in the patch:



One-Shot 1 serves to introduce a delay between the programming of the ramp voltage and the subsequent fall of the ramp. This is necessary because the programming requires a certain time to be completed; if the delay were not present, the programmer would still be trying to program the intended voltage while the ramp was falling quickly. (The difference between the programmed value and the actual peak value of the ramp for each note is not nearly as great as the drawing suggests. Both one-shots are typically set close to their shortest pulse width, about 10 msec.) The inverse of one-shot 2 serves to reset the ramp: by setting the up rate of the VCLAG very slow, and the down rate at the fastest possible rate, a short-lived fall in the VCLAG input (produced, in this case, by the inverted pulse from one-shot 2) causes the VCLAG output to fall rapidly and rise again slowly. The total rise time of the ramp should be adjusted to be slightly longer than the longest note you will program.

To play back the programmed sequence with timing, set up the following patch using the same VCLAG settings, the same timing memory you just programmed, and an envelope follower. (The control voltage memory, which you programmed simultaneously with a second programmer, will be used to control a voice, of course):



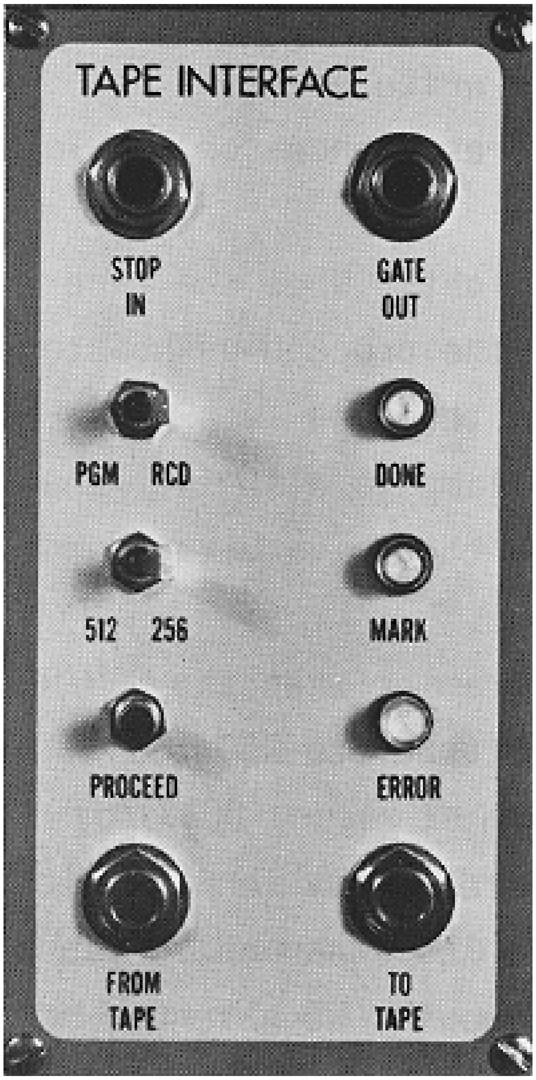
To start the replay, reset the memory and pull out the VCLAG input patch cord for a short time to make the VCLAG output fall. (This could be done more elegantly using, say, a low octave gate). When the rising ramp reaches the voltage programmed in address 000, the EF gate will rise, causing the MAG to advance to address 001 and the ramp to start over again. When the ramp reaches the voltage stored in 001, it falls again and advances the MAG to 002. Thus, the memory continues to clock its own MAG (and, hence, the control voltage memory) at the rate determined by its stored voltages, giving the original timing in the replay. Notice the effects of EF gate threshold and VCLAG up rate on the play-back timing.

This patch is very simple, and doesn't really work too well, for several reasons. The voltage stored in address 000 will always be +5V, since that is the voltage at which the VCLAG output levels off. To correct this, you would need a "start" key of some sort (a low octave gate, perhaps).

Also, the accuracy is not excellent, since the ramp must rise very slowly to include the longest note you will play. This could be corrected by setting the ramp quite fast, and then including a provision in the patch to program and advance the MAG every time the ramp peaks out—thus, long notes would 'overflow' into several addresses, and short notes would be more accurately converted. Also, the patch doesn't remember rests, but only the total time elapsed between successive key depressions.

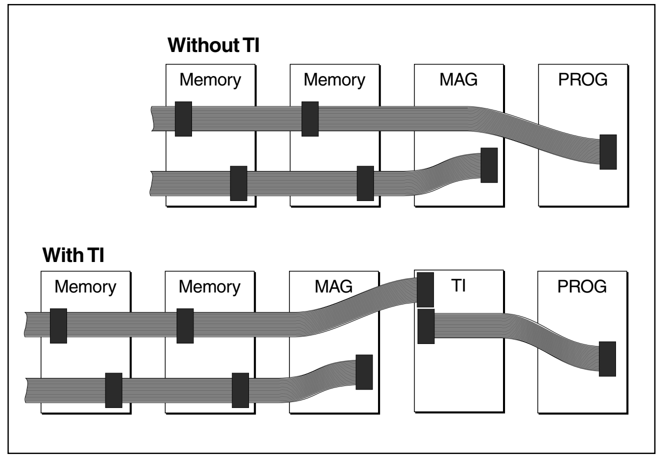
Of course, if you're willing to make a second pass of programming to store duration information, additional complexity is completely unnecessary. Simply control the frequency of the clocking VCC or VCO with the analog output of a second memory at 1V/octave—zero volts gives the base rate, 1V gives twice the base rate, 1.5 V gives three times the base rate, etc. This method is not only easy to program (the required voltages can be taken directly from the keyboard), but it is also very economical of memory. Consider a passage in four which you decide to modify by including a triplet. If duration is stored in the standard "metronomic" way, you'll have to completely reprogram the sequence, using three times as many memory locations. If duration is stored in a second pass with another memory, you'll end up using only two additional addresses after the change.

Tape Interface



The 2547 Tape Interface makes it possible to store the contents of memories as audio signals on recording tape, and to reprogram the memories from the tape recorder at any time. With this module, you can accumulate a library of sequences in permanent, non-volatile form.

Before discussing how you use the TI, an explanation of how it works would be helpful. In a system containing a TI, the 8-bit digital output of the programmer is an input to the TI, through a ribbon cable. The TI also has an 8-bit output which is conveyed to the memories with another ribbon cable:



The TI has two modes of operation, “record” and “program” (selected by the RCD/PGM switch). In record mode, the 8-bit output of the programmer appears directly on the output ribbon cable of the TI—thus, to program memories in the usual way from the keyboard, the TI is simply left in record mode and ignored. When recording the contents of memory on tape (record mode), the TI also converts the parallel output of the programmer (8 bits which are presented all at the same time) into a series string of bits (groups of 8 bits, plus beginning-of-word, end-of-word, and error-checking bits, stretched out in a string over a short time period). During the parallel-to-series conversion of each 8-bit word, a falling edge occurs on the “gate” output of the TI, which is used to advance the MAG. The next rising edge on this-gate output causes the programmer to convert the information in the new address into a new 8-bit word, which is, in turn, serialized by the TI. The resulting series string of ones and zeroes (remember—just voltages!) is used to control the frequency of an internal triangle-wave VCO; thus, the output of the VCO switches quickly between two frequencies (called “mark” and “space”, equal to 1800 Hz and 2400 Hz, respectively). When this signal, output on the “to tape” jack, is recorded on a tape recorder, the entire memory’s digital information is thereby stored on the tape as a sequence of the two frequencies.

When programming a memory from such a tape (program mode), the operation of the TI is reversed. As the tape is played back with its output applied to the “from tape” jack, the string of marks and spaces is converted into a string of zeroes and ones, the error-checking bits are read to assure that no errors occurred in recording or playback, and the series string of bits is converted into a sequence of 8-bit parallel words, which are sent to the memories on the output ribbon cable. A pulse is issued on the gate output after each word has been written into memory, allowing the MAG to be clocked to the next address.

To actually use the TI, a certain amount of external patching is necessary. Say you want to record the contents of memory on tape. First, make sure the 256/512 switch is in the 256 position. (“512” will be used if an entire 512-address memory is to be recorded.) Next switch the TI mode switch to “RCD”. Patch the memory output to the programmer analog input, and the upper three digital outputs into the programmer digital inputs. Patch the gate output of the TI into the gate input of the programmer, and the advance pulse output of the programmer to the MAG up clock. Patch the “to tape” jack into the line input of your tape recorder. Start recording on the tape recorder— if you monitor the tape recorder “source” or “tape” output, you will hear a mark signal (1800 Hz). Allow the mark signal to be recorded for several seconds to give the tape speed time to stabilize. Reset the MAG, and push the “proceed” pushbutton on the TI: the TI, through the programmer, will clock the MAG through all 256 addresses in about 10 seconds, unloading (non-destructively) the contents of the memory to the tape recorder. When the recording is finished, the MAG will return to 000, and the “done” LED will light. You can now turn off the tape recorder. Alternatively, if you don’t need to copy the entire memory, but, say, only the first 50 addresses, a digital output can be programmed high only in address 49_{ten} and patched to the “stop” input of the TI. The MAG will then stop after address 49_{ten} is programmed, and you won’t have to waste time recording unwanted material.

Try loading another memory with the information you just recorded on tape. To do this, switch the TI to program mode and patch the TI gate to the MAG up clock (or just leave in place the patchcords from TI gate to programmer gate input and advance pulse to MAG). Make sure the MAG is reset. Patch the line output of the tape recorder to the “from tape” input of the TI, rewind the tape, and play it. When the mark signal comes up on the tape, the “mark” LED will light. If you now push the “proceed” pushbutton, the TI is armed and will begin to load any write-enabled memories as soon as the information comes up on the tape. As before, the MAG will be clocked through all 256 addresses and will stop at 000; the “done” lamp will then light. (If a stop bit was recorded on tape, the MAG will stop and the done lamp will light after the contents of the address holding the stop bit have been programmed). If any errors occurred during recording or playback (drop-out, insufficient level, wrong playback speed, etc.), the “error” LED will light; you then have to find the problem! Assuming the error lamp does not light, the memories which were write-enabled will now be correctly programmed. If you want the error-checking feature as you record a tape, rather than having to wait until you play it back, the tape playback monitor can be patched into the “from tape” input during the recording procedure.

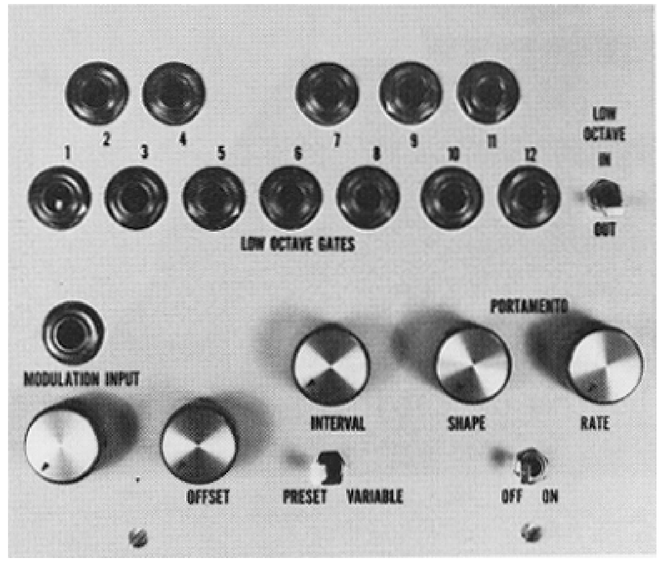
In this case, the mark lamp will light before recording, and the error lamp will indicate any errors as soon as they occur.

It may, at first glance, seem strange that the programmer is required in the patch to record a memory on tape, but is not necessary when programming a memory from tape. This is understandable when you consider that, in any of its patches, the programmer serves primarily as an analog-to-digital converter (an “ADC”). Though the memories store their information digitally, they output 3/4 of it as an analog voltage, so the programmer’s ADC must be used to redigitize the analog output for processing by the TI. However, when programming a memory from tape, the information taken from the tape is already in a digital form, so an ADC is not needed. When the programmer is used in programming from tape to memory, it is only to provide an advance pulse to the MAG, and this is only a convenience when the TI must be used for both recording and programming, and you don't want to have to change the patch to change modes. Otherwise, as was mentioned, the MAG can be clocked directly by the TI gate when programming.

The TI is an asynchronous device, which means that it doesn't care, within certain limits, whether the recording tape speed and the programming tape speed are exactly the same. (They don't need to be sync'd, hence, "asynchronous"). The allowable variation between the two speeds (and also, for the same reason, the cumulative "wow" of both tape recorders is about 4%. If you need to program from tapes recorded on a tape recorder with radically different speeds than yours, compensation of the TI's internal clock rate, by means of a trimmer on the submodule, can solve the problem.

In our experience, the TI works fine with even inexpensive cassette tape recorders, though good tape is a must. Absence of drop-out and splices are the most important considerations.

4000 Monophonic Keyboard



Historically, synthesizers have generally been keyboard-controlled. Though the collection of possible controllers is wide and expanding (guitar controllers, wind instrument controllers, slide controllers, joy sticks, foot pedals, knobs, etc.), keyboards are still one of the most graphic and versatile musician/synthesizer interfaces available. Several features of the Eμ synthesizer are particularly convenient for use with a keyboard, for example, the ribbon cable busses and the memory. As such, you should know basically what the keyboard does before delving into the various modules.

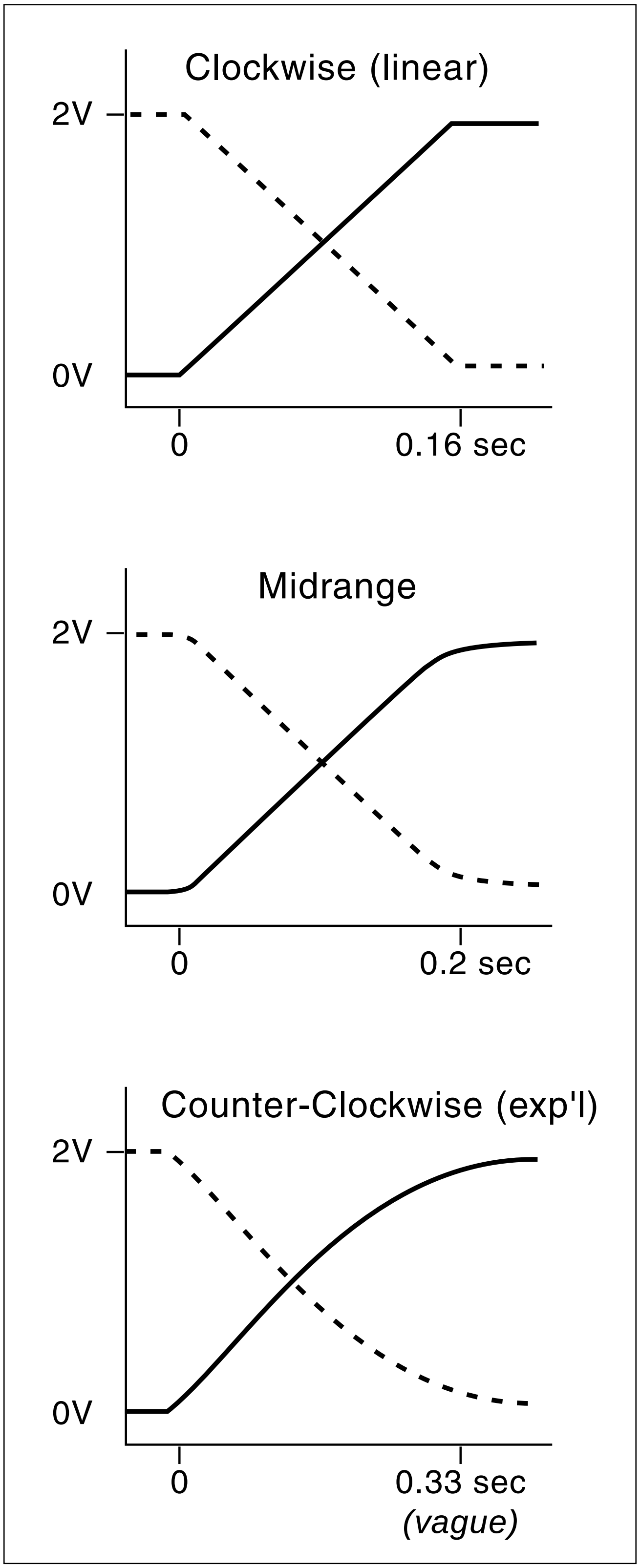
Synthesizers in the past have used keyboards which are monophonic, that is, capable of detecting only one key depression at a time. The reasoning behind this limitation is partially the relative ease of designing a monophonic keyboard compared to a polyphonic one; fundamentally, though, we're talking about the basic difference between a synthesizer and an organ. A synthesizer, in its simplest, theoretical form, has a single sound source (an oscillator) whose output frequency is varied to play melodies by varying an applied control voltage. An organ, on the other hand, has the equivalent of a separate oscillator for each key, and each of these oscillators is fixed at its one frequency. Thus, a synthesizer is analogous to a flute, and an organ is more like a "Hardart" (that strange instrument in which middle C is, say, a particular bicycle horn, middle D is a car spring, etc.) The advantage of the organ is its ability to play any number of its separate sound sources at the same time; the advantage of the synthesizer is the ability to continuously change its voice, even during the evolution of a single note. Eμ and others do make true polyphonic synthesizers, but these are best regarded as containing a separate synthesizer for each voice.

It is electronically simple, and fairly sensible, to limit the one depressed key which the keyboard can detect to the lowest depressed key when several are depressed. When all the keys are released, the keyboard should hold, indefinitely, the voltage output corresponding to the last key released. This is how all high quality monophonic synthesizer keyboards work, including Eμ's. While most synthesizers use a string of precision resistors to determine the "control voltage" output, the 4000 Keyboard is digital. Basically, the circuitry scans the entire keyboard every 10 msec, counting in base two through all the keys. Because of the 10 msec scan time of the circuitry, contact bounce is eliminated from all keyboard outputs.

For playing in the even-tempered chromatic scale, the keyboard has a "preset" mode, selected by the "preset/variable" switch. In this mode, the control voltage increases by exactly 1/12 V (trimmable) with each successive key—thus, one octave is a difference in control voltage of precisely 1V. Since the oscillators and filters have calibrated 1V/octave inputs, the preset interval allows perfect pitch chromaticity to be obtained. Because the keyboard is digital, drift can be essentially eliminated, or at least reduced to far below the maximum 3.5 mV (1/24 semitone) fixed error. Patch the control voltage output of the keyboard to the 1V/octave input of a VCO, and listen to the VCO. (The control voltage output is available either from the power supply front panel or the keyboard bus through the "KYBD" switch on the VCO). For the moment, switch the low octave "in", the interval to "preset", and portamento off. Play something! With the switch in "variable" position, the control voltage interval between two successive keys can be varied, with the "interval" pot, from zero volts (full left) to about 2/12 V (full right). Thus, the keyboard output is 2V or less per octave. The most obvious application of the variable mode is playing non-chromatic intervals, such as quarter tones. (Of course, the keyboard need not even he used on a VCO, so there are other applications). It is possible to preset a variable interval on the interval pot, and switch from 1V/octave tuning to your-preset interval whenever you want, without having to change the patch, simply by means of the preset/variable switch. Try this in the patch you just set up.

The "offset" pot adds a voltage to the final output of the keyboard without affecting the interval; the total range of this pot is about 0.2V full left to +0.2V full right, giving a swing of about 2.5 semitones either down or up. Notice that the interval pot does have a slight effect on the offset. (This can be seen only on the lowest key, and is quite irrelevant, since the keyboard is entirely "re-tuned" anyway when switched out of preset mode.

The monophonic keyboard has a "portamento" feature which introduces a variable slide in the transition from one control voltage to another, (With the portamento off, there is also a slide, but it is shorter than a microsecond per volt). Turn the portamento on, and experiment with the "rate" and "shape" pots. The shape pot controls the shape of the approach function; with the pot fully clockwise, the approach is linear, that is, the control voltage slews at a constant rate between the original value and the final value. Notice the illusion of a slight "pop" when the control voltage suddenly levels off. With the shape pot fully counter-clockwise, the approach is exponential, that is, the control voltage approaches its final value more and more slowly the closer it gets (In fact, it technically never gets there!) Notice the gradual approach instead of the sudden "corner" of linear mode. With the shape pot in between the two extremes, the approach starts out linear, but the final corner is rounded to an adjustable extent, allowing the "pop" illusion to be reduced. The following diagrams show the effect of the shape pot graphically . The particular timings were measured with the rate pot at midrange:



The rate pot controls the overall rate of the approach. In linear mode, the slew rate is adjustable from 1V/10 msec (counter-clockwise) to 1V/sec (clockwise). Notice that the overall glide time is longer for the same rate setting in exponential mode than in linear mode.

There is an important difference between the portamento of an Eμ digital keyboard and that of a standard analog keyboard: if you lift the approached key before the control voltage reaches that value, a digital keyboard continues the slide anyway, whereas an analog keyboard gets “stuck” in the middle of the slide, at the voltage present when the key was lifted.

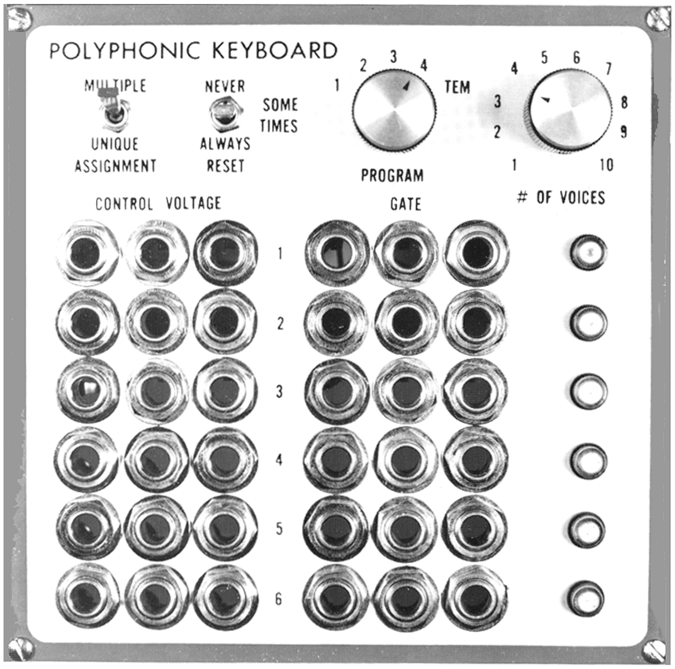
In addition to outputting a control voltage, the 4000 keyboard also outputs a gate and a trigger. The gate output is high (about 5V) as long as any number of keys is depressed, and low (zero volts with current sinking capability) when no key is depressed. The trigger is a short pulse which is issued either when the gate rises or when the control voltage changes. Thus, if a key is depressed, a trigger is issued. If that key is held down while a lower key is depressed, a second trigger occurs. If a higher key is depressed, no trigger occurs. If the lower key is released while the higher key is held down, a third trigger occurs.

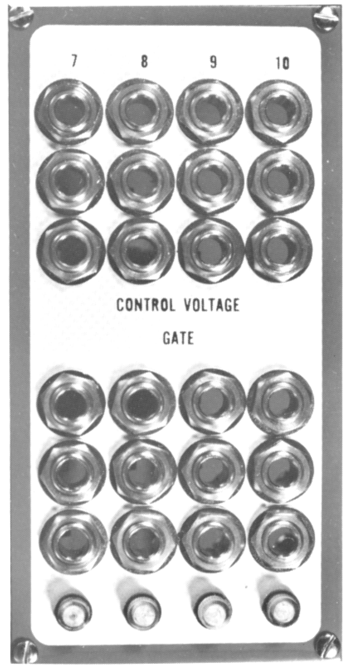
One application for the gate and trigger signals is the initiation of attacks by means of transient generators (q.v.) You will learn about that in more detail later; in any case, with both gate and trigger outputs present, all pertinent information about events on the monophonic keyboard is electronically available for any purpose. Don’t forget that gate and trigger are just voltages—the gate can be applied to any voltage-sensitive input, and the trigger can be used on any edge-sensitive input (for example, one-shots, latches, and pulse set inputs).

Another feature of the 4000 Keyboard is low octave gates. The lowest twelve keys, in addition to having the normal effects on control voltage, gate, and trigger, also function as twelve separate “sense switches”. When one of these keys is depressed, its corresponding low octave gate output goes high (5V) and remains high as long as that key is depressed. Any low octave gate output whose corresponding key is not depressed is at a digital zero (active low). The LOG's are very valuable for controlling assorted synthesizer functions from the keyboard. They can be used in the programming of memories, the initiation of special events (for example, synchronous waveforms or sequences), selection among several simultaneous voicings, gating of transient or sample & holds, activation of digital inputs, and an infinitude of other possibilities. They could also be used as analog control inputs fixed at 5V. Obviously, twelve separate gate functions of this nature can be patched simultaneously since the twelve low octave gates are independent. Often, you will want to use LOG's without their having an effect on the other keyboard outputs. To do this, the low octave can be switched out with the low octave switch; the twelve lowest keys then have no effect on control voltage, gate, or trigger, but function only as LOG switches. The twelve LOG's are always operative, regardless of whether the low octave is switched in or out.

The control voltage output of the keyboard is very low impedance (much less than 1Ω), unlike most other Eμ modules, whose output impedances are 1KΩ. This prevents the keyboard output from being loaded down when it is applied to a control input. Suppose the keyboard output impedance were higher, say, 1KΩ. Applied to the 1V/octave input of a VCO (100KΩ input impedance), such a keyboard output would be pulled down by 1%, throwing the VCO out of tune. By keeping the output impedance extremely low, the accuracy of the control voltage is maintained. You will see this principle applied to several other outputs (memory, for one) where high accuracy is essential.

4050 Polyphonic Keyboard





The 4050 Keyboard is a modular digital keyboard with true polyphonic capability up to ten voices. The physical keyboard is identical in outward appearance to the 4000 Monophonic Keyboard; however, in the 4050 series, several modularized functions are available as options, corresponding roughly to separate circuit boards interconnected within the 4050 keyboard cabinet.

The basic 4050 system consists of the keyboard, cabinet, controls (offset, interval, and switches), and the 4051 Keyboard Control board. The Keyboard Control activates the offset pot, interval pot, and preset/variable switch. As well, it contains the digital scanning circuitry and digital-to-analog converter, though it does not output a control voltage directly.

The addition of a 4052 Monophonic Voice option gives a lowest note monophonic keyboard which outputs control voltage, gate, and trigger. When a Monophonic Voice is included, the portamento circuitry, identical in effect to that of the 4000 Keyboard, is activated for the monophonic voice only. (Polyphonic portamento is somewhat hard to define, as you will see later. None of the several possible versions are available in the 4059 series). The monophonic voice of the 4052 has one subtle difference from the voice of a 4000 keyboard. To see it, patch the monophonic control voltage from a 4050 keyboard into the 1V/octave input of a VCO, and listen to the VCO. If you sweep the offset pot very quickly from one extreme to the other, the VCO pitch will change stepwise rather than continuously. The effect is even more noticeable if you switch the keyboard to variable interval, hit a high key, and sweep the interval pot very fast. In the 4000 keyboard, the same test will give a completely continuous pitch change, even at a fast rate, for the reason that, in the 4000 keyboard, the offset and interval controls affect the final processing of the control voltage. In the 4050 keyboard, however, these functions are performed by the 4051 in the initial processing of the control voltage - so that the offset and interval controls will affect all the voices identically. Since later processing in each voice includes a sample & hold stage, the stepwise effect is observed in all the 4050 control voltage outputs. (Of course, in normal usage, the offset and interval pots behave perfectly continuously). The control voltage, gate, and trigger of the 4052 monophonic voice are available, as in the 4000 keyboard, either from the power supply front panel or on the ribbon cable busses. Whereas the 4000 keyboard is accurate to 1/32 of a semitone (0.03 semitone, or 3 “cents”), the 4052 monophonic voice, as well as the polyphonic voices, is accurate to 1/128 semitone, or 0.8 cents.

The 4053 Low Octave Gates option adds the same low octave gates feature included in the 4000 keyboard. As in the 4000, the low octave can be switched in or out, but the twelve low octave gates are operational all the time.

The addition of the 4054 Polyphonic Control option, and up to ten 4055 Polyphonic Voices, gives polyphonic capability to the 4050 keyboard. In a polyphonic system, a Polyphonic Keyboard module panel is added to the system cabinet to hold the master controls and the output jacks for the first six voices. If the keyboard contains more than six 4055 voices, a 4057 7th Voice module panel is also included in the system cabinet to hold the output jacks for up to four more voices.

Each 4055 Voice outputs an independent control voltage and a gate. These are available from the appropriate row on the Polyphonic Keyboard panel (or column on the 7th Voice panel); three-jack multiples are provided for outputting the signals from each voice. When the gate for a particular voice is high, the LED on the front panel next to the appropriate gate output jacks is lit. Notice that the polyphonic keyboard does not output triggers; the reasoning behind this will become clear in a moment.

Consider the basic idea of the polyphonic keyboard: when a single key is depressed, the voltage corresponding to that key position is latched, in digital form, in one of the 4055 Voice boards (which voice is assigned to the key is determined by assignment rules which will be detailed later). The analog control voltage is thus held indefinitely on the appropriate control voltage output jacks, and the appropriate gate goes high for as long as the key is down. If the first key is held down while a second key is depressed, the second control voltage appears on the control voltage output of another voice, and the gate of that voice goes high. Similarly, as more keys are depressed, each one is given to a different voice in some sequence and the gate of each voice goes high for as long as its key is down. As long as a key is depressed, the voice assigned to it cannot be reassigned to a different key. When a key is released, the control voltage is held at the output, but the voice can then be reassigned to a different key at any time.

It should now be clear why triggers would be superfluous. The purpose of a trigger in a monophonic keyboard is to allow a lower note to give an attack while a higher note is still depressed. In a polyphonic system, additional key depressions are simply given to different voices, with different gates that go high. Thus, the rising edge of a gate is the only necessary condition for an attack on that voice. The one case where you might think a trigger is still necessary is when all but one voice are busy, and you play a single-note passage on the last remaining voice. In that case, you will find that the depression of a lower key while a higher key is held down cannot give an attack (since there’s no trigger), but an attack can be heard when the other is released. What actually happens is that the gate falls for an instant after the lifting of any key, even if another key is immediately given to the voice.

It should also be clear at this point why polyphonic portamento doesn’t make too much sense. On a monophonic keyboard, there is only one control voltage, so portamento is simply a matter of introducing a slew in that voltage. On a polyphonic keyboard, however, a second key depression often goes to a different voice, so there is no single output which can have knowledge of both voltages (clearly a prerequisite for slewing between them). Of course, separate portamentos could be used on each voice’s control voltage output; however, the result is quite unpredictable and very different in different assignment modes. (Besides, it’s easy to patch with a VCLAG in each voice).

The rules governing the assignment of voices to keys are a fairly complicated function of the settings of the “assignment” switch, “reset” switch, and “# of voices” rotary switch. For the moment, set the “# of voices” switch equal to or greater than (treated the same as equal to) the number of voices installed. With the assignment switch in “multiple” mode, voices are selected simply by an assignment counter , which looks sequentially through all the voices until it encounters a voice with a low gate. That voice will be the voice assigned to the next key depression. Thus, if the counter is never reset to voice 1 (“never” mode of the reset switch), successive single key depressions will be given to each voice in succession (in a four-voice system, 3,4,1,2,3,4, etc., or 1,2,3,4, etc.). In multiple/never mode, suppose a key is given to some voice, say, voice 3 of a four-voice system, and it is held down while several single keys are depressed in succession. These second key depressions will sequence voices 4,1,2,4,1,2, etc. If two keys are depressed during a succession of third key depressions, the third key will alternate between the last two voices. The fourth key depressed will go to the last remaining voice.

If the counter is always reset to voice 1 at the end of each keyboard scan (“always” mode of the reset switch), successive first key depressions will all be given to voice 1 (1, 1, 1, etc.). Once voice 1 is assigned to a depressed key, the end of every successive keyboard scan will reset the assignment counter to voice 1 as usual, but will find voice 1 busy as long as its key is down, and so, will count through the voices until it finds the next available voice (voice 2). Thus, second key depressions will all go to voice 2, third key depressions to voice 3, etc.

In the “sometimes” mode of the reset switch, the assignment counter is reset to voice 1 only after a keyboard scan in which no depressed keys are detected. Thus, first key depressions always go to voice 1, but additional key depressions are treated the same as in “never” mode.

Notice the differences among the three modes of the reset switch. In “never” mode, the assignment of voices is quite uncontrollable by means of keyboard technique. However, multiple/never mode gives the longest average cycle time before a given voice is reassigned to a different key, since the assignment counter won’t come back to any voice until it has cycled through all the others. The result is that long decays can be used on all voices without getting cut short by reassignment; when these decays overlap, a rich, “chorale” effect results from the beating of multiple voices. On the other hand, “always” mode gives the most frequent reassignment of voices, so the chorale effect does not occur as much (not at all on single-note passages). But multiple/always mode has the advantage of giving complete control over the assignment of voices to keys; for example, any triad can be played in six different ways depending only on the order in which the three keys are depressed:

Voice 1	C	C	E	E	G	G
Voice 2	E	G	C	G	C	E
Voice 3	G	E	G	C	E	C

Of course, this is only noticeable if all the voices sound different. In fact, if all the voices sound the same, the only difference among the three reset modes is the extent to which long decays give rise to the chorale effect. “Sometimes” mode is a very valuable compromise between the other two, sometimes. Consider first that the assignment of voice 1 is completely predictable, since you need only lift all the keys for an instant to guarantee that the next key depression will go to voice 1. Once voice 1 is busy, the rest of the voices are sequenced with each key depression, but always in the same order, and always starting with voice 2. Thus, any passage can be played over and over (with long decays, and, hence, chorale effect), with a completely reproducible sequence of voices, as long as it starts after all the keys are released and the first key in the passage is held down through the whole passage. This is musically very useful, particularly if voice 1 is different from the others, say, a louder bass voice.

So far, we've been considering only the "multiple" mode of the assignment switch. In multiple assignment mode, one key can be given to any number of voices; for example, in multiple/never mode simply depressing the same key several times can cause that key to be given to every voice. In "unique" assignment mode, however, once a key is given to a voice, further depressions of that key will not affect other voices, but will only cause the originally assigned voice to be repeatedly gated. In other words, that key will be uniquely given to the original voice every time it is depressed, until the voice is reassigned to a different key. (This is important to understand—the key belongs to the voice, but not vice versa).

Thus, in unique/never mode, repeated key depressions sequence through the voices only if they are all on different keys. If you hit one key several times, it doesn't change voices. The next different key you depress, however, will go to the next voice in sequence, since the assignment counter is superceded, but not clocked, when a note is taken by an out-of-sequence voice. Notice that, in never/unique mode, the unique associations of keys with their voices never last very long, since all the voices are constantly being sequenced. In always/unique mode, however, the possibility exists (especially with many voices) of one or more unique associations lasting through long passages, and then finally coming through at carefully planned moments as previously unheard voices sounding on previously untouched (but prearranged) keys. This works best, of course, if the prearranged associations involve high-numbered voices, since, even in always mode, low-numbered voices can get reassigned fairly often. Never/unique and sometimes/unique are particularly good for passages in which you don't want to get stuck too much in only a few voices (a possible problem in always mode), but you do want trills and ostenatos involving only a few different notes to repeatedly sequence through the same voices in the same order.

Clearly, the six different polyphonic modes available from the assignment and reset switches are quite different from one another, though often in subtle ways. Each has its own uses and its own optimum keyboard techniques for the effects you want. The situation is much simpler when all the voices are patched to sound the same, but even then the patterns of transient gating (attacks and decays) depend completely on the chosen modes. To basically understand the polyphonic keyboard, you'll probably have to play with it for several days or weeks. After that, you may keep finding out new details for several years!

If the "# of voices" rotary switch is set to a number less than the number of voices actually installed, all installed voices numbered higher than the switch setting are partially deactivated: they continue to hold, indefinitely, the control voltage which they were outputting when deactivated, but are not included in the sequencing of voices by the assignment counter. However, as was mentioned earlier, a key which is given to a voice in unique mode will be taken by that voice every time it is depressed, regardless of the position of the assignment counter. As such, even a deactivated voice can output a high gate if a key is depressed to which the voice was assigned in unique mode before deactivation. Since this is the only condition in which the gate of a deactivated voice will go high, deactivated voices can be used as unique key sensors for certain special applications. For example, the sounding of a particular note in a musical sequence can be the signal which initiates another synthesizer event—perhaps the start of a memory sequence, or a special transient. Such key-sensing could also be done in always/unique mode by the use of a high-numbered gate that was not deactivated, as long as you were careful not to use up so many voices that you reassigned the key-sensing voice to a different key. However, deactivation guarantees that the voice cannot be reassigned, thus allowing you to use any of the three reset modes.

The 4055 Programmability option enables the synthesist to instantly retune the entire 4950 keyboard to any one of four custom-tailored scales. Each of the four programs actually stores a value for each key position which represents the desired deviation (either up or down) of that key position from its normal value in even-tempered tuning. Different keys can be given different amounts of deviation, which can get pretty complicated, but the worst case can still have a half-semitone of deviation, and some keys can be given as much as 2.5 semitones of deviation. In any case, there's more than enough "slack" to retune the keyboard into any Just or Pythagorean intonation, to name only some traditional possibilities. Resolution of the programmed scales is 1.5 cents. When the programmability option is purchased, a five-position rotary switch is added to the Polyphonic Keyboard front panel—one position for tempered tuning, and a position for each of the four tuning programs. The four sets of deviation values are stored on an erasable, programmable read-only memory, which Eμ can program and reprogram at any time to user specifications.

A few general considerations are very important regarding polyphonic systems: for one, it's fairly easy to get a 10-voice 4050 keyboard, but a modular synthesizer capable of playing those ten voices would cost a lot more. The reason for this is, of course, that ten voices require a minimum of ten VCO's, ten VCF's, ten VCA's (or three QVCA's), and ten DTG'S. Any additional processors and signal or control sources (noise source, ring modulator, sequencer, etc.) would add still more to the final cost. Consider, too, that such a system still has very simple voices. Also, the patching of large polyphonic systems gets to be a real tangle unless extensive firm-wire patching is done. Often, for convenience, the busses for one keyboard can be cut into separate sections for each voice in a cabinet, and these sections can then be firm-wired to the appropriate voice outputs. Thus, the KYBD 1 bus can still be used for a whole-system monophonic voice, while the KYBD 2 busses give each polyphonic voice its appropriate signal. (Since the polyphonic system doesn't use triggers, TG's can't be gated from the keyboard busses, but must receive the polyphonic gates on their external gate inputs, either via patch cords or firm-wires.) Another observation about polyphonicity is that the wide-spectrum harmonic content of saw and pulse waveforms tends to muddy the sound when chords are played on multi-voice systems. To avoid this, careful attention must be paid to filtering and filter transients.

These thoughts are not intended to discourage anyone from pursuing polyphonicity—the richness of the sound from even a four-voice 4050 system makes it unlike any other keyboard instrument, and a six-voice guitar-controlled synthesizer can be truly astounding!

Applications

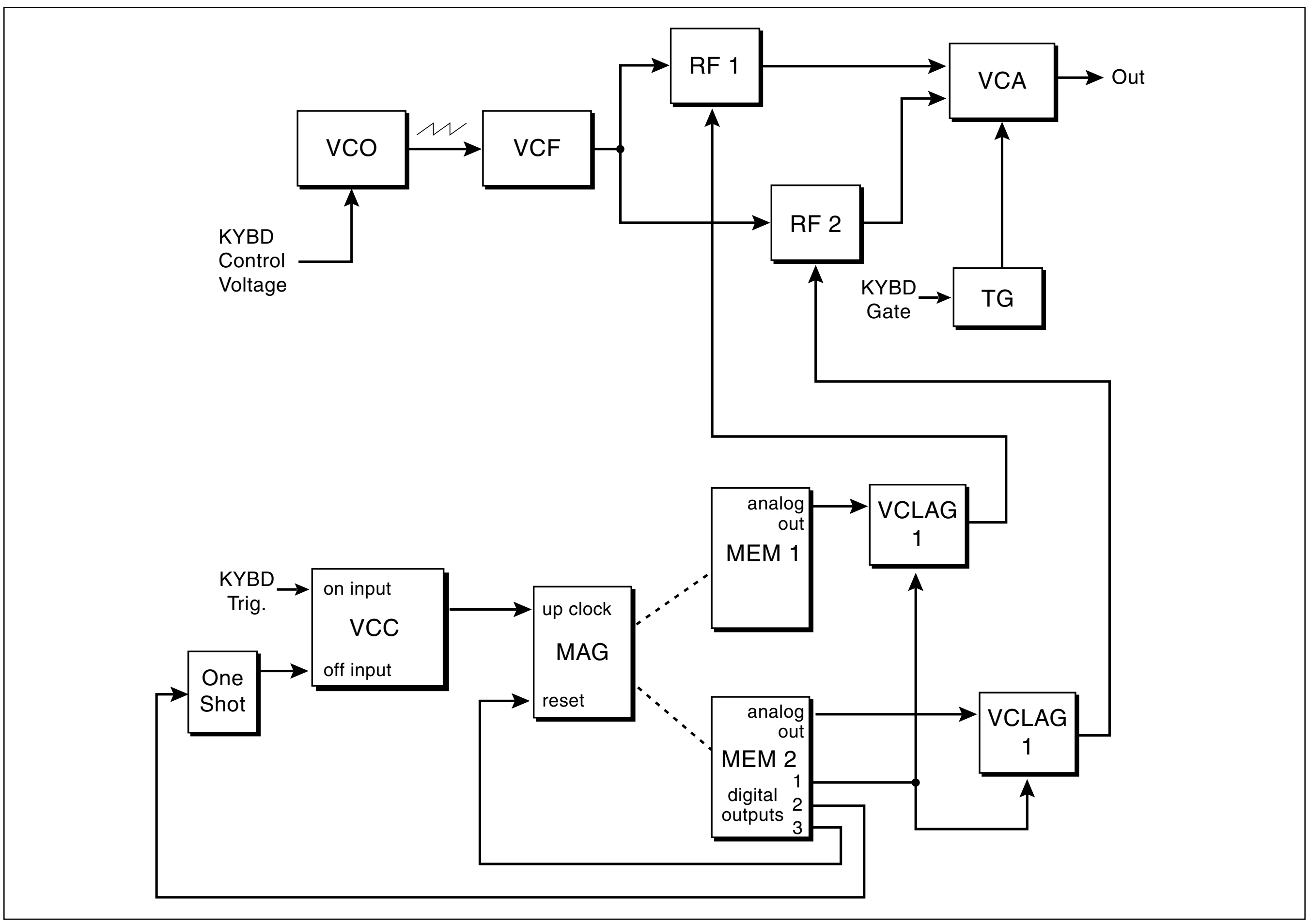
Speech Synthesis

Each different vowel sound in human speech is characterized by a unique pattern of fixed resonant peaks (formants) in the sound spectrum. In speech, these formant peaks result from excitation of the resonant cavities of the mouth, nose, and throat by the oscillations of the vocal cords. Substituting all or part of the natural apparatus with electronic analogues is the basis for such speech-like effects as "talking guitar", wah-wah, and the "bag" (an external sound source which excites the oral/nasal cavities through a plastic hose held between the teeth).

Vowel sounds can be generated on a synthesizer by passing a harmonic-rich waveform from a VCO through several resonant bandpass filters in parallel, to give the necessary peaks in the output spectrum. By choosing the number, locations, and Q of these peaks properly, all possible vowels and semi-vowels (1, m, n, r, ng) can theoretically be initiated. Actually, the requirements of our brains are not very stringent regarding certain of the parameters of speech; recognizable speech is often possible ,with just two equal-amplitude format peaks in the spectrum, though precise ratios between the frequencies are still essential. Thus, to crudely synthesize words or sentences containing only vowels, semi-vowels, and dipthongs, it is only necessary to program two precise parallel sequences of bandpass center frequencies (in addition to handling the comparatively simple problems of realistic intonation and dynamics).

One particular patch for the Eμ synthesizer can be programmed to pronounce fairly clearly, "How are you?", without the H. The same patch, with different programming, would work equally well for other vowel/semi-vowel/diphthong sequences, such as "emu", "mello yellow", or "I know why you lie on my ring."

Here's the patch:



There are two main sections in the patch. The upper section consists of the familiar VCO, VCF (24 dB/octave), VCA, Transient Generator (TG, our name for an “envelope generator”), and two Resonant Filters. (The RF is an audio-range universal active filter giving simultaneous highpass, bandpass, and lowpass functions. Only the bandpass with high Q is used in this patch). Different vowel sounds are achieved by controlling the center frequencies of the two Resonant Filters with two control voltage sequences which are previously programmed in the digital Memory modules shown in the lower portion of the patch. Each Memory module contains 256 “addresses” in a digital computer-type memory. In each address, a control voltage from zero to +5V can be stored (the “analog output voltage”), as well as a one-of-four digital code, whose use will be discussed later. The Memory Address Generator (MAG) is an associated module which allows any number of memories (in this case, two) to be clocked together either forward or backward through their addresses, giving a sequence up to 256 voltages long from each Memory’s analog output.

Clocking of the MAG is done in this patch with a Voltage Controlled Clock (VCC, basically a sawtooth/pulse VCO with on/off and multiplexing capability). To smooth out the stepwise jumps in the Memory analog outputs (essential for an imitation of free-flowing speech), two Voltage Controlled Lag Processors are interposed between the Memory outputs and the RF f_c-control inputs. (The VCLAG is similar to a low-f_c VCF, or a slew-rate limiter).

The first step in programming “How are you?” into the memories is to create a dictionary of all the phonemes (speech sounds) present in the sentence. Ignoring the H, there are four: AH, OO, ER, and E. The sentence can be written phonetically as:

AH $\overline{\text{OO}}$ AH ER $\bar{\text{E}}$ $\overline{\text{OO}}$

Next, the control voltages must be determined which, when applied to the RF control inputs, give the proper formant frequencies to imitate the required phonemes. This is best done empirically using a conventional pot-matrix sequencer, setting the pots by hand while their corresponding output voltages are applied to the RF control inputs. The Eμ 8-Position Address Generator and Voltage Source Output Unit (not shown in the patch) are well suited for making the dictionary, since they can be used as a 16x2 storage matrix. (16 formant pairs can thus be stored at the same time, corresponding to16 phonemes. In English, the vowels and semi-vowels together total just about 16 phonemes). The frequencies used in our imitation of “How are you?” are as follows:

	AH	**$\overline{\text{OO}}$**	**ER**	**$\bar{\text{E}}$**
f_1 (Hz)	909	250	385	250
f_2 (Hz)	1053	667	1000	3472

These are emphatically not the “correct” frequencies that you might find in tables of formant frequencies. They are only the closest empirical approximations that we could find in the patch we were using after a couple hours of tweaking! The AH and ER were quite good, while the OO and E left something to be desired.

Once the dictionary of voltages is made, it is only necessary to program them, in the correct order, into the two memories. This is done successively for each address, one memory at a time, using the Eμ Programmer module (not shown in the patch). Twelve addresses are used for “How are you?”, programmed in terms of phonemes as follows:

address	**0**	**1**	**2**	**3**	**4**	**5**
phoneme	AH	AH	$\overline{\text{OO}}$	$\overline{\text{OO}}$	AH	AH
address	**6**	**7**	**8**	**9**	**10**	**11**
phoneme	ER	ER	$\bar{\text{E}}$	$\overline{\text{OO}}$	$\overline{\text{OO}}$	$\overline{\text{OO}}$

Two addresses were used for most of the phonemes so that the E could be programmed in only one address, making it shorter in duration when the Memory is clocked. (Technically, Y is a transitional, not a vowel, since the E portion of YOO is so short).

The digital one-of-four code in one of the memories (in this case, Memory 2) is also programmed to control various portions of the patch. The digital outputs are very useful and fairly simple: whatever address the Memory is currently in, one of its four digital outputs will be at a digital "one" (+5V), and the other three will be at digital "zero" (grounded). Which output is high at each address can be programmed along with the control voltage for that address, or separately on a second pass. Here is how the first thirteen addresses of Memory 2 are programmed, showing the synchronization of the digital outputs with the phonemes:

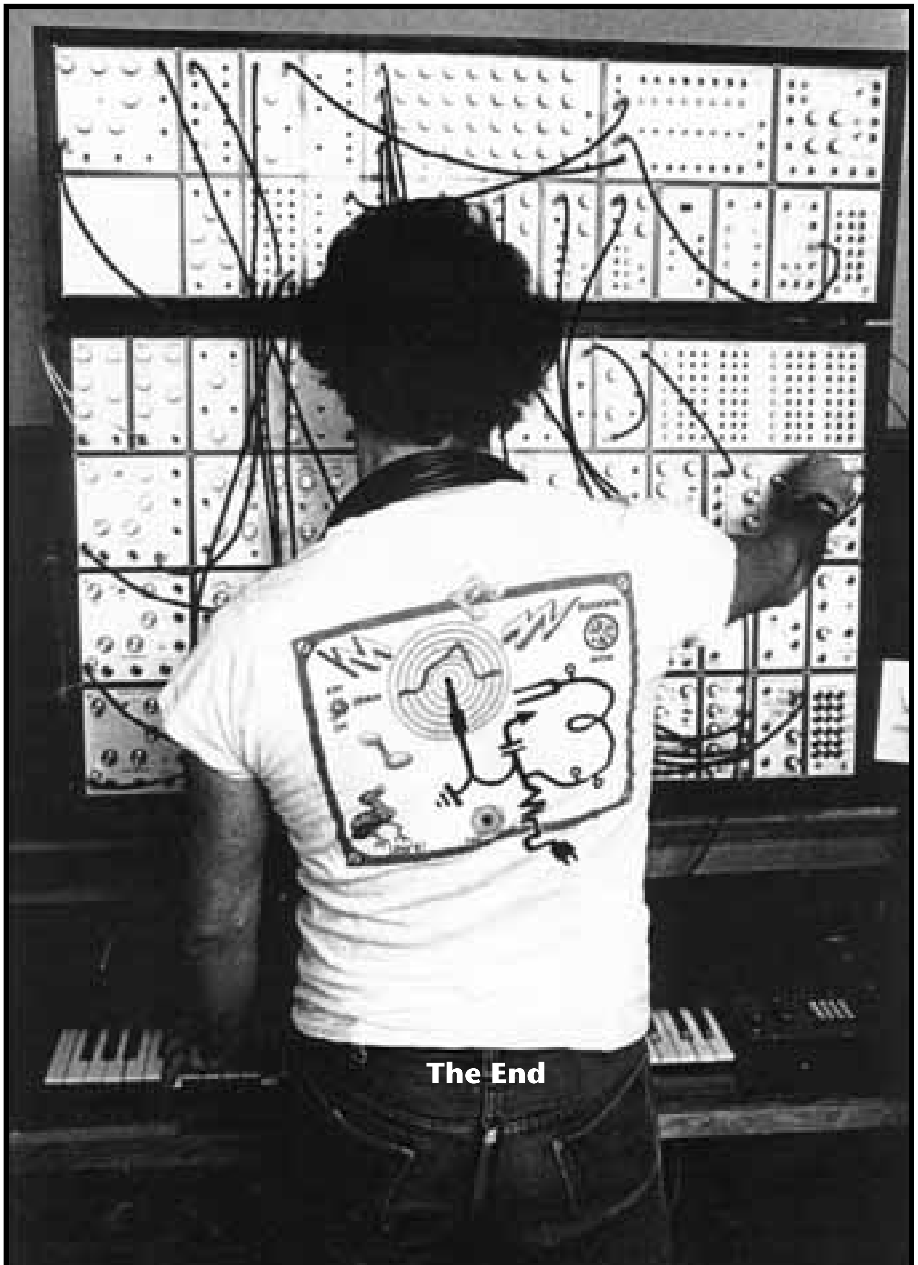
again, and the second word is pronounced. When digital output 2 goes high again in address 7, the VCC is turned off once again. A third key depression similarly causes the Memories to sequence through "you", stopping in address 11. When a fourth keyboard trigger occurs, the Memories advance to address 12, but a high level is then encountered (for the first time) on digital output 3, which causes the MAG to immediately reset the Memories to address 0. Since the clock remains, on after resetting, "how" is pronounced, and the Memories stop in address 3 as before. Thus, successive keyboard triggers cause the synthesizer to say, "how are you how are you how are..." Since the memories spend only about 10 µsec in address 12 before being reset to address 0, the analog contents of

address		0	1	2	3	4	5	6	7	8	9	10	11	12
analog output		AH	AH	$\overline{\text{OO}}$	$\overline{\text{OO}}$	AH	AH	ER	ER	$\bar{\text{E}}$	$\overline{\text{OO}}$	$\overline{\text{OO}}$	$\overline{\text{OO}}$	
digital output	1	1	0	0	0	1	0	0	0	1	0	0	0	0
"	2	0	0	0	1	0	0	0	1	0	0	0	1	0
"	3	0	0	0	0	0	0	0	0	0	0	0	0	1
"	4	0	1	1	0	0	1	1	0	0	1	1	0	0

Referring to the patch, the function of each digital output should become clear. When a key is depressed on the keyboard, the keyboard trigger turns the VCC on, and the Memories start sequencing from address 0, assuming the MAG is first reset. (A VCC clock rate of about 35 Hz , or 35 addresses/sec, seems about right). When the Memories reach address 3, a high level is encountered at digital output 2 of Memory 2. This high level activates the One-Shot module, turning off the VCC. (In this case, the One-Shot turns a gate into a trigger). Thus, the first word is pronounced, and the sequence stops. The transient on the VCA should, of course, be set so as to decay quickly at that point. With the next keyboard trigger, the VCC is turned on

address 12 are irrelevant. Digital output 1 is necessary to eliminate the slide which would otherwise occur at the beginning of each word as the Memory outputs change from their values at the end of the previous word. By programming digital output 1 high in the first address of each word and patching it to the slew-rate control inputs of the VCLAG's, the objectionable slides can be prevented.

Intonation is achieved by playing the keyboard, Since the VCO is controlled by the keyboard control voltage. With the keyboard portamento on, very life-like intonation is possible. For the formant frequencies listed above, we found a VCO fundamental frequency around 80-150 Hz to be the most human.



The End