Random Reading List

1. From Electronic Music: Systems, Techniques and Controls by Allen Strange
2. Don Buchla interview transcript from http://www.vasulka.org/
3. From: Musical Applications of Microprocessors by Hal Chamberlin
4. Wiard Model 1210 Noise Ring manual
5. Grant Richter’s notes on the Noise Ring Entropy Expander
6. Doepfer A-149 manual

I @2 _ ©§{u2s§2s2Y¦sv©s~2b¦£uw§§2 t ®2e¨wªw2dw{uz

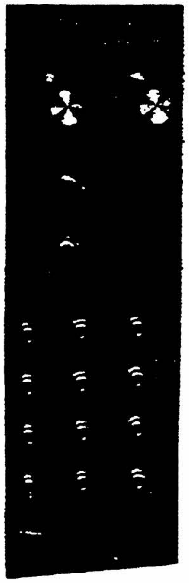


Figure 6.67. Aries AR-318 Sample/Hold (Photograph Courtesy of Aries Music Inc. Used by permission.)

Track-and-Hold. The *Track-and-Hold* (T/H) is a variation often accompanying the Sample-Hold and will usually be built into the same module, such as the Eu 2410 or the Aries AR-318 Sample and Hold (see figure 6.67). A track-and-hold operation relies on a gate input. When a gate voltage is present, it will exactly follow or duplicate the incoming voltage and as soon as the gate is released the output will hold at the last voltage level tracked. Consider the patch in figure 6.68. A triangle LFO or repetitive envelope is determining the period of a pulse source. Normally the period of the clock would vary with the rise and fall of the programming voltage. In this case the voltage is processed through a T/H. The voltage of the triangle function will appear unaltered at the output until a gate is present, perhaps from a keyboard. As soon as the T/H receives the gate, the output voltage will hold at the level present on the input at that time. The result is that the "tempo" of the clock will also hold at a proportional speed. As soon as the gate is released, the fluctuating input voltage will be outputted and the clock speed will immediately switch to the proportional speed and continue to follow the voltage function. This patch allows the player to grab and hold a fluctuating tempo. The clock can be used to trigger sequences, envelope generators, switches, etc. The T/H gives the player the ability to "freeze" an ongoing voltage at any point in its evolution, apply that voltage to any parameter, then to return to the ongoing voltage at will. This process can be applied to any transient voltage—sequencers, envelope generators, etc.

Random Voltage Sources. The most common application of the S/H is in the generation of random voltages. Most S/H circuits can sample an AC or DC voltage, independent of its frequency range. White or pink noise, when viewed on an oscilloscope, can be seen as rapid and random fluctuations in amplitude or voltage level. The S/H can grab one of these random voltages at any time so that the output is a stepped random voltage. The unattenuated output ideally is a random selection of control voltages throughout the control voltage range of the instrument (see figure 6.69). I say "ideally" because there is a probability factor involved with this method. As

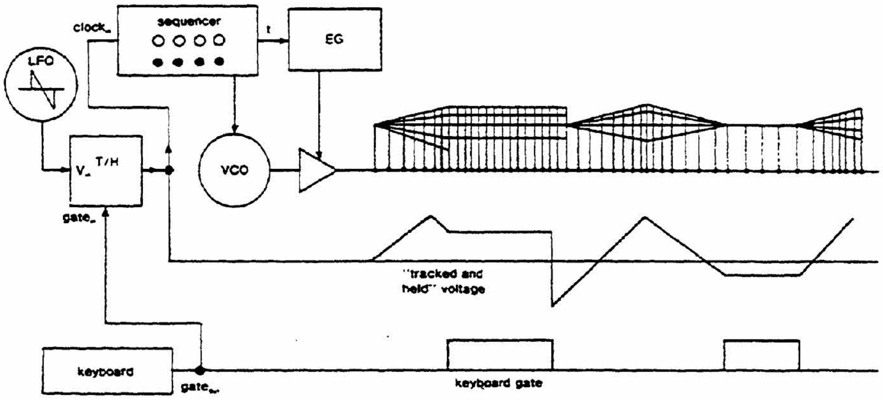


Figure 6.68. Track/hold application

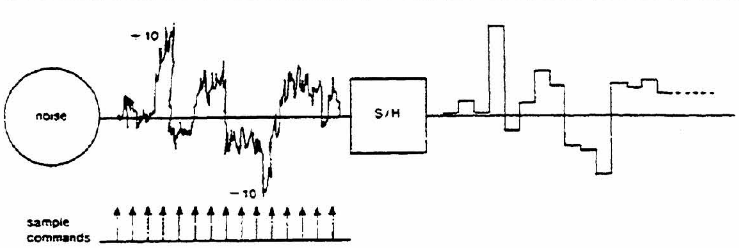


Figure 6.69. Sampling noise to produce random voltages

pointed out in Bernie Hutchins's *Theory and Application of Noise Generators in Electronic Music*,[8] a long term record of the voltages produced will show that there is a tendency for the voltages to gather around the middle of the voltage range. If the control range is −5 to +5 volts, a good portion of the "random" voltages will be near 0 volts. If the control voltage range is 0 to +10 volts, many of the voltages will be around +5 volts. This is due to what is known as the Gaussian distribution, which is a normal characteristic of white noise. If this type of sampling is applied to a VCO, there will be more pitches near the middle of the total available pitch range, and very high and low pitches will be less frequent. If this confuses you, there is another method which might be applicable and this is illustrated in figure 6.70. A sawtooth oscillator is modulated by pink or white noise (see chapter 8). This means that the frequency of the sawtooth wave is undergoing random changes at an audio rate. This sawtooth wave is then used as the sample voltage. What is being sampled is the descending (or ascending) edge of a voltage function which is changing randomly in time. With this method one has the same chance of sampling a high, medium, or low magnitude voltage. The amount of effect a random voltage has on a parameter is not the same as distribution probability. With no attenuation a random voltage source hypothetically will drive a parameter throughout its entire operational range. Attenuation of that voltage only restricts the range but does not change the distribution of events within that range. This technique of sampling noise to produce random voltage can be reversed to produce often surprising results. Use a stepped or triggered RVS clocked at a very fast rate, well into the audio range, to control a VCO. The VCO is then randomly changing frequency so fast that the VCO itself sounds like some undefinable noise source. The texture of the sound can be controlled by either the frequency of the timing pulses or the attenuation of the RVS.

Most dedicated random voltage sources (RVS) are based on the previously described techniques. Buchla instruments have done the most extensive developments in random voltage functions. The Buchla 266 Source of Uncertainty pictured on page 82 (figure 6.66) makes it possible to define these random voltages in a variety of ways. The upper section of the module provides three "flavors" of noise: white (+3 db/octave), pink (musically flat), and reciprocal white noise (−3 db/octave) which sounds like lowpass filtered pink noise but really involves a redistribution of energy rather than band limiting.

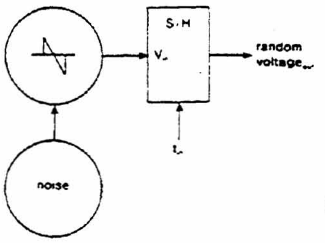


Figure 6.70. Alternate technique for generation of random voltage

The second section produces two fluctuating random voltage sources. This is analogous to "low" noise on most other random voltage sources. The probable speed of fluctuation can be defined anywhere between .05 and 50 times per second. The direction and maximum magnitude of the change cannot be predicted but the rate can be controlled. This function is voltage controllable so that various random rates can be pre-programmed by a sequencer, played by a keyboard or a brainwave, or even determined by another random voltage.

The "quantized random voltage" output is the most interesting, and a minimal amount of math is involved. There are two outputs that generate a voltage whenever a pulse is received. One output is marked "n + 1" and the other is marked "2^n". "n" is a numerical value from 1 to 6 and may be defined by a front panel offset or determined by an external control voltage. If n = 1, then the n + 1 output gives two random voltages, chosen in random order at the rate of the input pulse. The 2^n output also generates the same two random voltages (2 × 1 = 2) but in a different random order. As n is set to 2, the n + 1 output is 3 voltages and the 2^n output is 4; when n = 3, the n + 1 output is 4 random voltages and the 2^n output gives 8 random voltages, and so on. Thus it may be that this means linear or geometric access to the number of random voltages. As n increases, n + 1 increases linearly and 2^n increases geometrically. This leads to some interesting correlations, as illustrated in figure 6.71. In this instrument the n + 1 output is patched to a VCO and to the "n" input of the RVS. The 2^n output is patched to control a filter and to the period input of the pulser. Before reading further, make your own analysis and prediction about how this random instrument will behave! A high magnitude random voltage will generate a high pitch (depending on the attenuation setting on the VCO) and simultaneously set the value of n higher so that the next pitch can be randomly selected from a greater range of possibilities. If the voltage is low, the pitch will be correspondingly low and set the value of "n" so that the next pitch will have more restricted range of possibilities. Simultaneously the 2^n output controls the filter. Thus as the range of pitch selection increases, the number of possible spectral ranges becomes greater, but in a geometric relation. The speed of the pulser providing the triggering information is also controlled by the 2^n output so that bright

[8] Published in *ElectroNotes*, #64, vol. 8, April 1976, p. 3.

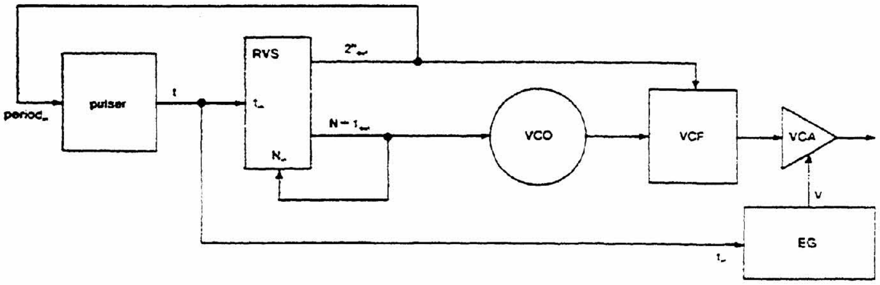
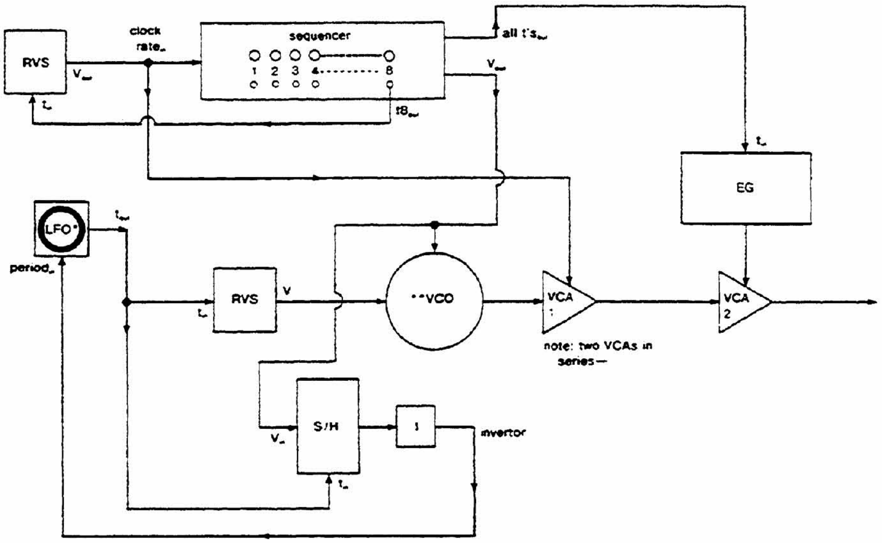


Figure 6.71. Correlation of random voltages in an instrument



*Any controllable pulse source.
**The sequencer and random voltage source are equally mixed to control the pitch of the VCO.

Figure 6.72. "Dream machine"

timbre is accompanied by longer events, longer events are accompanied by greater range probabilities for pitch, and the number of range probabilities for pitch selection is correlated geometrically with the number of possible spectral choices! This tail-chasing configuration can really consume many hours in the studio. However, it is worth exploration, therefore just keep it under control according to your compositional and performance interests.

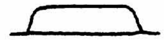
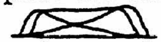
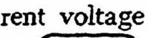
The fourth section of the Buchla 266 also has two outputs: one marked  and the other marked . Both outputs generate stepped random voltages when an input trigger is received. But each output has a different voltage distribution probability. The top output  generates random voltages according to even voltage distribution. The bottom output has a pot and control voltage input for establishing random voltage distribution probabilities. With the distribution pot to the "low" setting, most of the random voltages will be low magnitude, with occasional medium and even less frequent high magnitude voltages. This is not the same as attenuating, because the total magnitude is not compressed but rather involves a redistribution of energy. As the distribution pot is turned to the right, or as a control voltage is applied, the distribution moves through medium to high magnitude random voltage distribution. Voltage control of this distribution also allows one to program or play in real-time the center areas of random activity. The two lower sections of the Buchla 266 are a voltage controlled integrator with the time constant defined by an offset, or a control voltage and a polyphonic sample-hold as described on page 82).

Figure 6.72 is a "dream machine" in which the musical structure is defined on a moderately high level but two RVS's are used to affect the behavior of the fixed structure. Like Douglas Leedy's *Entropical Paradise* (see page 224), once this instrument is turned on, it will totally control its own performance. Analyze this instrument and see how much of this "random" instrument is actually predictable.

Patch Analysis. At this point the reader is dealing with fairly complex instrument configurations and it is very easy to be lost in the maze of connections and correlations. I would suggest following a set analytical procedure. You may already have developed a favorite method of taking patches apart, or an instructor may have some valuable hints. The following method is one that I have used with success with my own students and I would recommend that you try it.

The Rapout* Approach to Electronic Instrument Configurations

*Reasonable Analytical Procedure for Observing Usable Techniques

1. Locate and trace all audio signal routing and their offsets.
   a. How many sound sources are used (VCO's, noise sources, external sources via mikes, tape recorders, etc.)?
   b. Are these signals detected via ED's or PVC's to generate controls, or are they used as real voices within the instrument?
   c. Locate and trace all signal routing from their source to the final output via sub-mixes and final mixes.
2. Identify all variable audio parameters.
   a. In what ways are the audio signals processed (filters, amps, reverb, etc.)?
   b. Are these processing modules variable manually as specified by the composer, or are they voltage controlled by means of an active input? Note the control voltage attenuation level and predict how much effect a voltage (positive or negative) will have on the parameter—how much will the pitch of a VCO change with an applied voltage, what will be the maximum gain of a VCA, how much spectral change will be caused by the control of a filter, etc. At this point you will be aware of the number of voices involved and the number and degree of possible change in those parameters which can contribute to the basic sonic nature of the sounds.
3. Locate all control voltage sources, routing, and processing:
   a. to audio sources and any external processing onvolved—what controls the VCO's, and is the control inverted, integrated, sampled, quantized, etc., before it reaches the VCO? Do the same for each sound source or audio processing module (mixers, VCA's, etc.).
   b. to function generators. If a function generator has a manually variable or voltage controllable sub-function, what is doing the controlling, what is its range of effect in relation to the established offset, etc.?
4. Locate all timing pulse sources, their period and any processing.
   a. are the pulses manually activated or automated?
   b. if automatically generated, what is their period or speed and is this function voltage controlled? If voltage controlled, what is doing the controlling and what is the expected range of variation in relation to the offset period?
   c. is there any timing pulse processing involved—electronic switches, gate inversion, etc.?
5. Identify all structural correlations—audio and control.
   a. Within a single voice, is pitch related to loudness by means of common control sources? Are filter sweeps related to VCA control, etc.?
   b. Are separate voices related by common controls? Does the loudness of one voice have anything in common with the pitch of another voice?
   c. What are the relationships between control voltage sources? Is the speed of an LFO or the period of a pulser related to the selection or control of another function?
   d. What are the relationships between the behavior of any audio sources and processing and a voltage controlled function? Is the gain of a VCA related to the decay time on an EG?
6. Now try to describe verbally what the instrument will do.

This is a lengthy process but some configurations can become very complicated.

Miscellaneous Controllers. Another method of producing constant or varying control voltages is with "photosensitive controls." In electronic circuitry there are many components that are used to limit or block voltage. Their rating (how much voltage or current they are capable of blocking) may be permanently fixed or may be manually controlled as with a potentiometer. Photosensitive devices will vary their rating in relationship to an applied light source. In simpler terms, a photosensitive controller is a light-controlled pot. A "photosensitive oscillator" will usually generate zero Hz when no light is applied to its light controlled resistor (pot). The photo-oscillator will generate a maximum frequency when a maximum amount of light is applied. A photo-amplifier will provide signal amplification in direct relationship to the amount of light applied to its photo-controller. (Photosensitive controls are very applicable to spatial modulating devices and will be discussed in detail in chapter 13.)

Direct voltage can be controlled with light by using "photodiodes." An absence of light striking the photodiode may result in zero volts DC, while an increase in the amount of light (usually measured in lumins) will produce a proportional increase in DC voltage. The amount of light can be controlled in two different ways. A change in voltage to the light source will change the intensity, but this method is usually inadequate because the control of voltage is the desired outcome. The most useful methods of controlling light domain at the same time as the audio domain made compromises. DC offset doesn't make any difference in the sound domain but it makes a big difference in the structural domain whereas harmonic distortion makes very little difference in the control areas but it can be very significant in the audio areas. You also have a matter of just being able to discern what's happening in a system by looking at it, by observation. If you have a very complex patch it's nice to be able to tell what aspect of the patch is the structural part of the music versus what is the signal path and so on. Another factor is that perceptually we hear quite differently than we assess things. An example would be if you have an input device on a control device you want to deal with attenuation in a linear fashion and you might even want to turn it upside down because phase is an important difference in terms of how something is controlled. Whether the pitch is going up or down is important to us but in an audio device we hear things exponentially. So if you're dealing with pitch and volume, you want to deal with exponential functions. So there's a big difference in whether you deal with linear versus exponential functions at the control sound level and that was a very inhibiting factor in Moog's more general approach.

Could you talk a little about your interest in randomness because to me that was one of the big advantages of your system? Compositionally, that capability is what interested me. It informed a lot of the structural basis of Mort's work as in "Silver Apples" and a lot of these pieces. What was your rationale behind that?

Uncertainty is the basis for a lot of my work. One of the important dichotomies of music to me is predictability versus uncertainty. One always operates somewhere between the totally predictable and the totally unpredictable and to me the source of uncertainty, as we called it, was a way of aiding the composer. It's not fair to say that it's totally random because we allowed constraints. We could say the deviations from one event to another in terms of the parameters we were dealing with would be constrained to a small number or the choices from a, say a group of pitches would be restrained to these pitches rather than just a random walk. The predictabilities could be highly defined. You could have a sequence of totally random numbers but you could have a sequence in which the numbers were only allowed to change by a certain range or interval. We had voltage control of the randomness and voltage control of the rate of change so one could randomize the rate of change. In this way you could make patterns that were of more interest than patterns that are totally random. So we got quite involved in randomness. An interesting story I could tell now is that Yusochevsky bought three identical systems from us very early in the game, in '65 or so, to outfit the Columbia/Princeton electronic music studios. He was very disturbed by the random module and taped them over. He didn't actually disassemble them but in the two graduate studios he taped them over. In his own studio he allowed the randomness to be used but he did not want to assess compositions made with the random voltage generator.

Did you have something against the keyboard in the early days?

No, that's a myth. In the late 60s Rolling Stone published something that said I hate keyboards and hate keyboard players. I countered by pointing out that some of my best friends were keyboard players. It's often concluded that because I don't happen to choose to build keyboards into my instruments that I'm somehow against them and that's not true. I even studied keyboard myself but I don't choose to adapt the organ keyboard to a means of control of electronic vocabularies. I'm not saying there's anything wrong with that at all. It's simply that others are doing it and I'm not here to duplicate other people's work. To me the alternatives are much more interesting.

In the beginning you didn't know there were others. There weren't others.

No, but that's a funny one. In the beginning we were all dealing with monophonic structures. I did actually build a very early keyboard and decided this is not very interesting. To me, as a keyboard player, the concept of a monophonic keyboard was very strange. I never even thought of it. When I built my first keyboard it was polyphonic. It four-voiced polyphonic and then later on Arp came into the game much, much later and said, "We have the first polyphonic keyboard." Then I said, "I never thought of a monophonic keyboard." Then I got acquainted with Moog's equipment and realized that people were building keyboards with which you could only play one note at a time. To me that was absolutely preposterous. The only virtue of a keyboard is that you can simultaneously access a number of notes. I used a keyboard in some of my larger systems. In fact, very expensive hybrid systems that came out about '70-71 and found in my own work that the keyboard just was too dictating. I'd look at it and it shouted "twelve tone" back at me. I couldn't deal with it in an abstract way. I actually wrote pieces and more often adapted others. I remember a Daniel Lenz piece that was thought to be impossible to play , because it involved modifications to four strings and a voice. It might have been called "Sermon." It had a beautiful graphic score which I photographed and projected and put the entire score on the keyboard and then called a pianist and said, "Here, play this." Well, we wound up with five people having to simultaneously play the keyboard because it was impossible to do otherwise. I used the natural groupings of the keyboard. I would use the three black keys for different ring modulations and four of this for reverberation and two of this for switching on and off some other function. It was a five octave keyboard with one octave devoted to the control of each instrument. So there's my experience with keyboards. It's a one dimensional system for throwing hammers at strings and getting things into vibration and it's very good for that and very good for a particular kind of music but not so much for the aesthetic that Mort was into and numerous other composers on the west coast

Another novel input method is the *breath control* transducer. Essentially these are nothing more than pressure transducers with a tube that can be inserted into the user's mouth. Variations in breath pressure, which may be positive or negative, are converted into output variations. Such devices are most useful when the user's hands and feet are already tied up manipulating other input devices.

Modified Musical Instruments

Many potential users of a computer-based synthesis system may have spent years perfecting playing techniques on various instruments such as the guitar or clarinet. Accordingly, it has become common to fit contacts and other sensors to these instruments for easy input into a synthesizer or computer. For example, it is relatively simple to install contacts under the keys of a clarinet. This coupled with a rough amplitude analysis of the actual clarinet sound gives the functional equivalent of a keyboard. Translating key closure patterns into equivalent notes is not so simple because it is the *pattern* of keys that are pressed that is important. When going from one pattern to another, it is unlikely that all of the keys will make or break simultaneously so some intelligence is required to prevent spurious note outputs. Also many notes have "alternate fingerings." Thus, even if the clarinet is to be used to control a synthesizer directly, a microprocessor would be useful as part of the interface.

Guitar controllers are another popular item. These are actually an application of source-signal analysis, but the guitar and associated pickups are usually modified to simplify the analysis task. For example, since simultaneous tones are very difficult to separate, an independent magnetic pickup is provided for each string. Also, since strong harmonics can lead to pitch errors when the signal is analyzed, the pickups are placed near the center of the string length. If such a guitar were simply connected to a conventional amplifier, the sound would be quite dull and lifeless.

Typically, the guitar's audio signal is analyzed into amplitude and frequency parameters for each of the six strings. Often the amplitude channel is used merely as a trigger source for an envelope generator; thus, the synthesized sound may have any amplitude envelope desired. One of the attractions of guitar controllers is the fact that they are inherently polyphonic. Not only can up to six nearly independent tones (each string has a somewhat different frequency range) be simultaneously controlled, there is no problem in the assignment of notes to voices; the string corresponds to the voice.

Algorithmic Input

Certainly everyone has been exposed in one way or another to the often beautiful images created from mathematical equations. The spiragraph, which is a machine for drawing cycloids, and its output is such an example.

In fact, much computer art is done by evaluating various equations. Often "random-number generators" are used to set the parameters of the equations and then the computer takes off and generates an image. The "artistic" part of the process is the knowledge of what equations make good "material" and the judgment to modify or reject unsuitable results.

The same concepts can be applied to sounds and music. In fact, many purely analog synthesizers have the means for automatically generating sequences of control voltages that may be highly ordered, totally random, or anything in between. Likewise, a computer user has at his disposal the ability to evaluate equations of any complexity for ordered sequences and a random-number generator for disordered sequences. Algorithms for averaging and selecting random data can also be easily set up. Since this whole discussion crosses the line between music performance and music composition, it will be kept brief and no value judgments will be made about the various techniques.

A complete electronic music performance involves many sequences of events and dozens of time-varying parameters. On the other hand, a simple melody really only requires two parameters, the pitches and durations of the notes. Since conventional music starts with a melody and adds accompaniment, algorithmic composition efforts usually concentrate on generating melodies. Depending on the application, the "melody" may be as simple as a repeating sequence of notes or a genuine attempt at automatically composing a true melody.

Sample-and-Hold Module

One very useful device for sequence generation that is present on many analog synthesizers is a sample-and-hold (SAH) module. Functionally, it is the same device that was discussed in the section on analog-to-digital conversion. For synthesizer use, it has a signal input, a trigger input, and a signal output. When a trigger occurs, the output is immediately updated to match

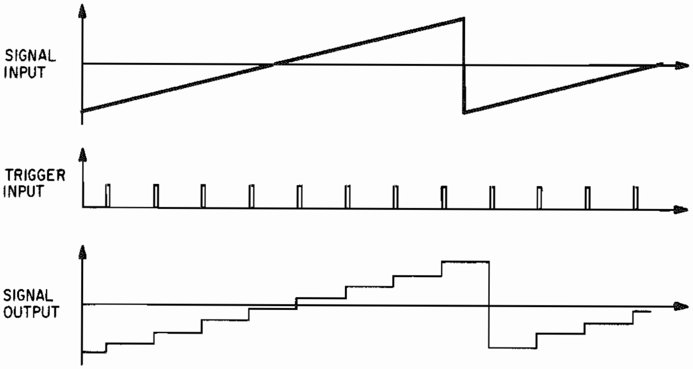


Fig. 10–3. Sampling a low-frequency sawtooth wave

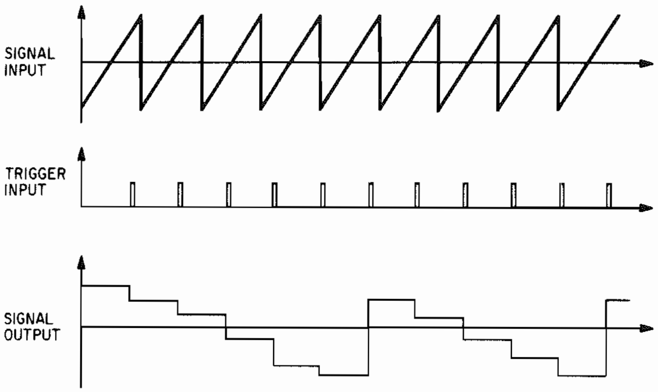


Fig. 10–4. Sampling a high-frequency sawtooth wave

the input. Between triggers, the output remains constant at its last value. The trigger input is usually designed so that any kind of waveform can drive it with perhaps the positive-going zero crossings of the wave being the trigger points. In essence, the SAH module accepts a continuously varying input and produces a stepwise output where the trigger initiates each step. Thus, if the SAH output drives a VCO and the trigger also drives an envelope generator, then a distinct "note" is produced for each trigger.

One of the simplest applications of the SAH module is in producing arpeggios. If a very-low-frequency (0.2 Hz) sawtooth wave is fed into the signal input and a 2 Hz pulse is fed into the trigger input, the output will be a staircase wave as in Fig. 10–3. When controlling a VCO, the staircase will produce an ascending sequence of 10 notes that repeats indefinitely. The actual notes depend on the amplitude of the sawtooth and the VCO settings.

If the pulse and sawtooth frequencies are not in a precise integral ratio, then each repetition of the sequence will be different. It is not difficult to adjust things to produce a scale of fifths that increases (or decreases) a half-step each iteration for six iterations and then repeats. If the sawtooth frequency is increased so that it is slightly higher than the trigger frequency, a *descending* series of notes is produced as illustrated in Fig. 10–4. Note that a slight change in the relative frequencies of the two waves can have a profound effect on the output sequence. This sensitivity increases as the sawtooth frequency increases. In the kilohertz range, interesting patterns of sequence evolution are produced as the sawtooth frequency drifts slightly due to imperfections in the VCO generating it. One can become completely absorbed in knob twiddling using a such a setup.

For truly random sequences, the SAH module can be set up to sample white noise. One would feed white noise into the signal input and a constant frequency into the trigger input. The output then would be a random series of steps. When using a typical analog white noise generator (diode junction

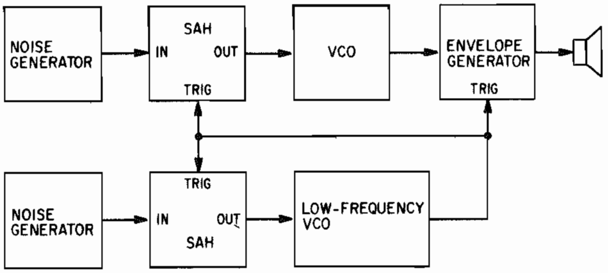


Fig. 10–5. Generator for random notes of random duration

noise), the steps are completely random and will never repeat. When this signal drives a VCO (use the sine wave output), the resulting series of random pitches of identical duration is the typical Hollywood conception of a computer hard at work.

Two SAH modules set up according to Fig. 10–5 will produce a random pitch sequence with random durations. Essentially, the first SAH determines the pitches, while the second determines the durations by controlling the VCO that provides the trigger pulses. The result is more interesting but still completely disordered.

Statistics

Although random sequences are unpredictable, they do have definite statistical properties. The most important ones are the mean or average value, the standard deviation, and the probability density function. The output of virtually any noise generator will have an average value of zero. If a mean of +5 V is desired, all that is necessary is to add a dc voltage of that magnitude. The standard deviation is equivalent to the rms voltage of the noise; thus, it may be changed with a simple gain control. Most noise sources also have a gaussian (bell-shaped normal curve) density function, which is not quite as easy to change. Even though a SAH module converts white noise into a random series of steps, the sampling process does not change any of these statistical properties.

The probability density function can be changed by using the SAH module differently. The idea is to *randomly* sample a *periodic* waveform. The resulting probability density function depends only on the shape of the sampled waveform, not on the properties of the noise source. Figure 10–6 shows a setup to do this. If the waveform to be sampled is in the kilohertz range, then only slight random variations in the sampling interval are needed. Thus, for practical purposes the step durations can still be controlled as desired.

Fortunately, the standard synthesizer waveforms give desirable density functions. Both the sawtooth and the triangular wave give a *uniform* (flat-

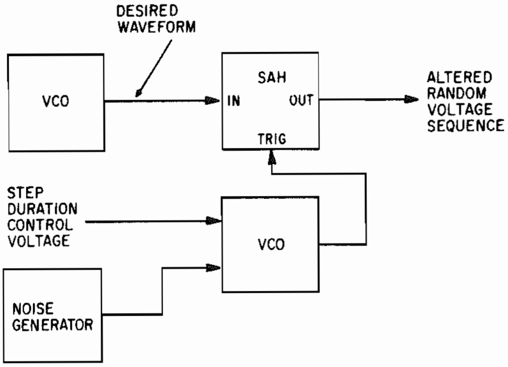

DESIRED WAVEFORM
VCO
SAH
IN
OUT
TRIG
ALTERED RANDOM VOLTAGE SEQUENCE
STEP DURATION CONTROL VOLTAGE
VCO
NOISE GENERATOR

Fig. 10–6. Altering the probability density function

topped) distribution. A sine wave gives a reasonable likeness of a normal distribution, which might be useful for cleaning up the output of an otherwise poor noise source. Note that unlike a true normal distribution, there is an upper limit on peak deviation from the mean. A square wave gives two spikes, which means that only two different output voltages are possible and there is a random selection between them. (Actually this only applies to perfect square waves and SAHs; in a real situation one would occasionally get an intermediate output.) A rectangular wave gives similar results, but one value will be more probable than the other according to the duty cycle of the wave.

So far the SAH module was assumed to be perfect, that is, the input was instantly sampled and held. In a real SAH module, each occurrence of the trigger fires a single shot, which closes the sampling switch long enough for the hold capacitor to charge up to the instantaneous input signal voltage. Typically, this time is in the low microsecond range and is constant. If a resistance is inserted in series with the analog switch then the output will move *toward* the input during the sample interval but will not reach it. The effect is sometimes called "slew limiting" of the SAH.

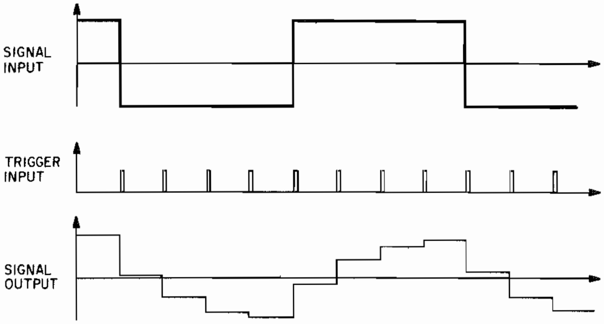


Fig. 10–7. Slew-limited sampling of square wave

Figure 10–7 shows the effect of sampling a low-frequency square wave with such a degraded SAH. The output is a series of rising and falling arpeggios, but the pitch intervals start large and then decrease after each direction reversal. Altering the series resistor changes the step size and rate of interval decrease considerably. The effect on sampled white noise is also interesting. As the resistor is increased, the random output changes from total disorder toward a more correlated result. Unfortunately, the standard deviation also decreases, which would have to be counteracted with a VCA in a practical application. Obviously, voltage control of slew limiting would be a useful SAH feature.

Controlling Randomness

As was mentioned, a slew-limited SAH module is capable of imparting a degree of order in random sequences. Actually, it is possible to get exactly the same results by passing the signal to be sampled through a single-pole R-C low-pass filter first. If white noise is to be sampled, then this amounts to filtering the noise. The slew-limited SAH module is actually a discrete-time low-pass filter, which is the first step toward a digital filter!

A sequence of random numbers is actually sampled white noise. Thus, one can easily write a program to simulate analog sampling of white noise by using the RND (random number) function available in the BASIC programming language. If a synthesizer is interfaced to the computer, then random numbers can be fed to an oscillator to produce the same kinds of note sequences available with analog setups. One point to be aware of is that most random number generators have a *uniform* probability distribution, generally between 0.0 and 1.0. A good approximation to a gaussian distribution may be had by adding up 12 random numbers (distributed uniformly between 0.0 and 1.0) and subtracting 6.0 from the sum. The mean of the result will be 0 and the standard deviation will be 1.0.

The name "stochastic music" refers to music (melodies) that orginates from sequences of random numbers. It should be apparent that raw random numbers, regardless of the probability distribution function, would create rather uninteresting music. Each note is an independent entity, with no relation to what came before and no influence on what follows.

A very simple algorithm can be applied to a random sequence, however, to produce a highly correlated sequence that might be more interesting. The basic idea is to use random numbers to determine the direction and magnitude of pitch *movement* rather than the pitches themselves. As a simple example, let's say that the pitches are to be notes on the chromatic equal-tempered scale and that the maximum allowable interval between successive notes is an octave. Thus, a sequence of random integers falling between −12 and +12 inclusive is needed. The BASIC expression INT(25*RND(1)) −12 will produce such a sequence. To produce a note sequence, numbers would first be assigned to the notes on the scale. Next, a starting point, such as middle C, must be selected. To determine what the next note should be, one simply evaluates the above expression and *adds* the random number to the numerical equivalent of the previous note.

One undesirable side effect of this process is that the notes can run off the ends of the scale. One solution is to treat the ends of the keyboard as "reflecting barriers," which would "bounce" the sequence back toward middle C. For a gentle reflecting action, one might alter the split of up/down probabilities to favor down when the current note is high and vice versa.

In any case, the resulting "melody" is highly correlated because the pitch of the current note depends on *all* of the preceding notes as well as a random input. Likewise, the current note will influence all future notes. The audible effect of such a sequence (particularly if the maximum allowed interval is small) can be described as an aimless wandering with few surprises. Most listeners would say that the sequence is too correlated to be really interesting.

Various schemes have been tried to produce sequences that are more correlated than raw random numbers but less correlated than the method just described provides. Just as white noise has a flat spectrum, the sampled white noise associated with raw random numbers also has a flat spectrum. The algorithm just discussed is actually a simple digital filter; an integrator to be exact. An integrator is simply a low-pass filter with a 6-dB/octave cutoff slope. Unlike the typical low-pass filter, however, the response curve continues to increase as frequency decreases without limit. The random numbers emerging from the process then have a filtered spectrum that increases by 6 dB for each octave of frequency decrease. Thus, it would seem that other digital filters would be useful for modifying random sequences.

More Sophisticated Techniques

One such filter that has been studied is a so-called "pink noise" or "1/F" filter, which has a slope that rises *3 dB/octave* as frequency decreases. The 1/F designation is used because the spectral *power* per hertz of bandwidth is inversely proportional to frequency. Since this is midway between 0 dB and 6 dB, the degree of correlation should also be intermediate. Listening tests bear this out; most people rate 1/F sequences as more pleasing than raw or integrated sequences. Unfortunately, a good 1/F digital filter is quite complex.

Another idea is to provide a mechanism whereby the influence of past events either ceases or diminishes as the sequence continues. For example, one might specify that the next note will depend on the previous three notes and a random input. One implementation method involves a large table that lists every possible combination of the three previous notes. Each entry in the table specifies a percentage probability for the next note. The random-number generator is used to select the next note based on the specified probabilities. The character of the music generated thus depends on the table entries and the number of prior notes considered.

One method for filling the table is analysis of existing music. For example, one might perform a statistical analysis of all four note sequences in the most popular Bach organ fugues. The data obtained could be compiled into a table like the one just described. There would probably be numerous combinations that did not occur in the music analyzed, so one might have to add a "back-tracking" capability to the program. One problem with extending the technique to consider longer sequences of notes is the tremendous increase in table size. The analysis of most conventional music, however, would result in a large proportion of empty (zero probability) table entries. Thus, it may be more compact to formulate the data into a set of rules. Besides memory savings, it is usually easier to experiment with the rules than thousands of probability table entries.

The results of such efforts have been mildly successful in producing interesting sequences. Pieces produced by analyzing Bach's music, for example, may sound Bach-like for a short run of a few notes. However, after listening for awhile, it becomes apparent that the music is just drifting aimlessly and getting nowhere. Overanalysis is likely to result in whole phrases from the analyzed material appearing in the output.

Analog Feedback Techniques

Another method of producing sequences is to use the principle of *feedback.* The sequences produced, while definitely not random, are complex and often unpredictable. The basic idea is to set up a collection of devices or modules, each of which has an input, an output, and performs some processing function. The modules are strung together and the output of the last module is fed back into the input of the first. Multiple-feedback paths can also exist. A simple sequence, even a single event, is then fed into the chain and gets processed over and over changing some on each trip. With multiple feedback paths, the sequence may be split and duplicated on each evolution.

One of the simplest setups is a series of SAH modules, all driven by the same trigger as in Fig. 10–8. A multiple-input VCA is used to selectively mix an input from outside and one or more feedback loops. With only the input enabled, the final output from the system is simply a delayed, sampled version of the input. Outputs taken from intermediate states would be identical but with differing delays. This might be useful in creating sequence echo effects or even have a sequence play a "round" with itself.

With the end-around feedback path enabled, many possibilities exist. One could, for example, fill the SAH chain with a short sequence of notes (five SAHs could hold a five-note sequence), disable the input, and recircu-

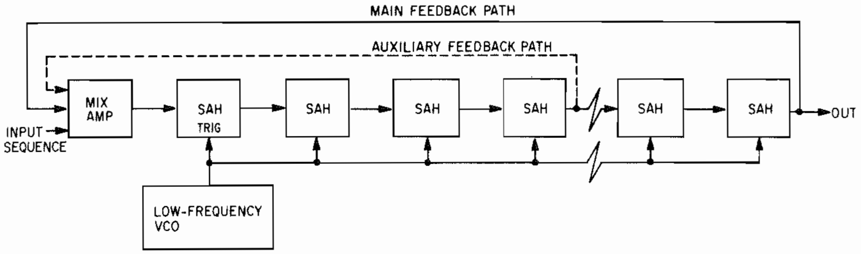


Fig. 10–8. SAH module feedback sequence generator

late the same sequence. If the SAH modules were perfect, the sequence would repeat indefinitely, but in reality analog errors would accumulate and the sequence would evolve in perhaps interesting ways. If, instead of removing the external input, it were fed a constant 1/12 V, the sequence would shift upward a half step on each interaction. If the feedback gain were greater or lesser than unity, the pitch intervals in the sequence would progressively increase or decrease, respectively. Enabling a second feedback path would create such complex patterns that they may be difficult to predict beyond the first repetition. The use of slew-limited SAH modules adds yet another dimension of possibilities.

Digital Feedback Techniques

A digital feedback system can be set up using flip-flops connected as a shift register. Figure 10–9 shows such a system. The input summer that drives the register is a parity generator that actually computes the "modulus 2 sum" of all its inputs. The switches enable a particular feedback path if closed. A low-frequency VCO provides trigger pulses to drive the system.

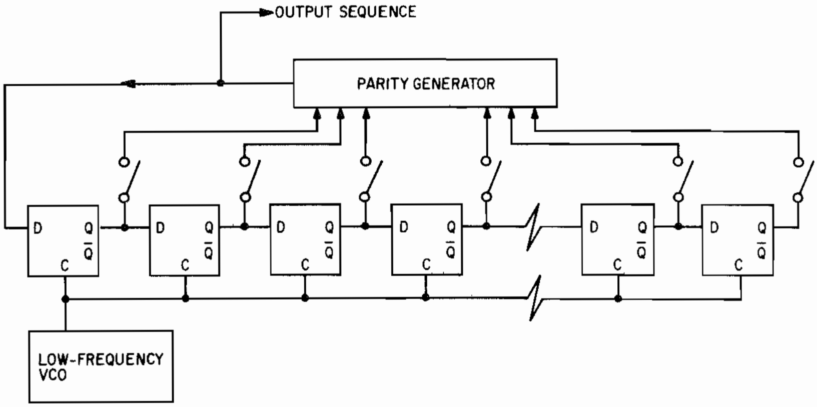


Fig. 10–9. Feedback shift register sequence generator

```
0000001100000101
0000111100010001
0011001101010101
111111100000001
```

(A)

```
0000011100010101
0110101000010110
0110001100101001
1101110100010011
0111100001101001
0001111101011100
1001011111100111
101101100000001
```

(B)

```
0000100101000011
0101100110010101
0011110011011010
0000101001011000
1001110001111111
1000001110011111
0110010001000110
0111010111011011
0000001100011011
1100010111101010
1011111101000101
0110111000011110
1110100110100100
100000001
```

(C)

Fig. 10–10. Some 8-bit feedback shift register sequences. Note: Switch states read left to right as in Fig. 10–9. One cycle of the output sequence from the parity generator is shown. (A) Switches = 00000011. (B) Switches = 00000111. (C) Switches = 00001001.

The output, of course, is simply a two-level digital signal that may change only in conjunction with a trigger pulse. As such, it is useful for rhythm generation, but there are methods for controlling multiple-frequency tones also.

The sequence generated depends entirely on the configuration of open and closed switches. Since there are 2^N possible switch combinations, a fairly

```
1111111010101011
0011001000100011
1100001010000011
000000100000001
```

(D)

```
0001110001001011
1000000110010010
0110111001000001
0101101101011001
0110000111110110
1111010111010001
0000110110001111
0011100110001011
0100100010100101
0100111011101100
1111011111101001
1001101010001100
0001110101010111
1100101000010011
1111110000101111
000110100000001
```

(E)

Fig. 10–10. Some 8-bit feedback shift register sequences *(cont.)* (D) Switches = 10000001. (E) Switches = 00011101. This is the longest possible sequence using an 8-bit register (255 bits).

small number of stages can create a nearly infinite number of different patterns ranging from highly structured to virtually random. The sequence length (number of clock cycles necessary to cause the sequence to repeat) varies from just 2 to $2^N - 1$, where N is the number of shift register stages. From this vast array of sequences, Fig. 10–10 shows a few of those possible with an 8-bit register.

The Muse

At least one interesting device has been marketed that is based on the feedback shift register principle. It is called the "Muse" and is advertised as a music composition machine with which the user, by setting several levers and switches, controls a sequence generator, which in turn controls a single oscillator to produce notes.

A simplified block diagram of the device is shown in Fig. 10–11. Thirty-eight different digital signals are generated by several counter stages and a 31-stage shift register. These signals along with constant 0 and 1 are

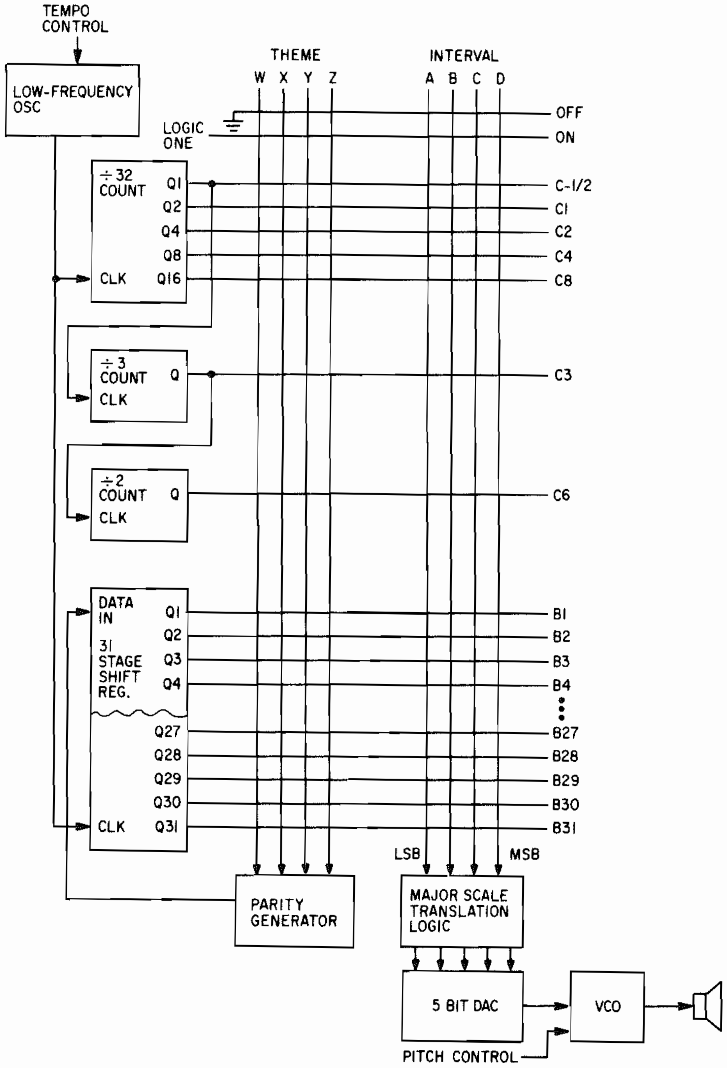


Fig. 10–11. Block diagram of Muse

connected to 40 signal rows. Eight 40-position slide switches divided into two groups of four switches act as columns and can select any individual row signal. Four of the switches, which are called "theme" controls, feed a parity generator whose output feeds the 31-position shift register. The other four switches, designated "interval" controls, are connected through some translation logic to a 5-bit DAC, which drives a VCO tone generator and output speaker. The VCO and DAC are adjusted so that the step size is a semitone on the equally tempered scale and the translation logic converts its 4-bit input into a 5-bit output according to the conventions of the major musical scale. An adjustable low-frequency oscillator clocks the counters and shift register.

In the Muse, the rows driven by the counters and constant 0 and 1 are designated as the "C" (counter) region. Five of these rows are connected to a simple 5-bit counter, while two more connect to a divide-by-6 and divide-by-12 counter. The outputs of the various counters are normally used for short, highly ordered sequences. For example, if the "A" switch is set to row "C1", B to C2, C to C4, etc., the device will generate an ascending major scale. Essentially, a binary counter has been connected to the DAC, which would be expected to generate an ascending staircase waveform. If switch A is moved to the C1/2 position, the scale will still ascend but by alternate intervals of one note and three notes. Moving B and D back to the off position (constant 0 row), results in a pair of trills: C-D-C-D-G-A-G-A-C-D . . . Many other combinations, of course, are possible, but the sequence length will never be more than 64 notes using the C6 row or 32 notes otherwise.

The 31 rows in the "B" (binary) region are driven by the 31 stage shift register, which shifts downward from row 1 to 2 to 3, etc. The four "theme" switches are used to control the shift register by determining what will be shifted into the register's first stage input. If they are set in the C region, then the register acts merely as a delay line. This can be useful in creating cannon effects. However, if one or more are set in the B region, then a feedback path into the shift register is created and some complex sequences indeed can result. One possibility is to set the theme switches for a complex sequence, set three of the interval switches in the C region for a repetitive tone pattern, and set the fourth somewhere in the B region. The result is that the repetitive pattern is modified according to the shift register pattern. Although one can think through what the effects of a particular switch setting might be, there are so many degrees of freedom that one usually succumbs to random tinkering. The number of unique combinations is for all practical purposes infinite.

Obviously, the concept can be easily expanded to more stages, more notes, more voices, rhythm control, and even scale changes. Of all the algorithmic "composition" methods discussed thus far, the author feels that this one holds the most promise and is the most fun to use. It is obvious that the Muse can be easily simulated on any microcomputer system using any language desired. Although user interaction may not be as convenient as the many slide bars and switches on the real thing, it becomes easy to expand or restructure the device with minor program changes. Also, with the user interaction techniques discussed in the next chapter, even the user interface can be improved upon.

Word length	*Sequence length*	*A*	*B*
8	256	77	55
12	4096	1485	865
16	65536	13709	13849
24	16777216	732573	3545443
32	4294967296	1963 14165	9076335 15

Shift Register Method

Another method that is superior to the linear congruential method in some respects can be called the feedback shift register random bit generator. As the name implies, the method generates random *bits* that are grouped to form random numbers. A feedback shift register similar to that discussed in Chapter 10 is used. By proper selection of the taps to be exclusive or-ed and fed back, the register can generate a sequence of $2^N - 1$ bits before repeating. To form an N-bit random integer, where N is equal to or less than the register length, one simply iterates the register at least N times and then reads the result directly from the register.

One advantage of the method is that all of the bits are random and therefore can be used indiscriminately for random control functions. Another advantage is that only logical operations are needed. On the other hand, the method as presently formulated is not as efficient at generating numbers with large N compared with the previous method even if the computer does not have a multiply instruction. Also, when set up for iterating N times to get an N-bit number, it fails some statistical randomness tests. This fault may be minimized by iterating somewhat more than N times. On the other hand, quite satisfactory white noise samples are generated with only a few iterations, such as five, for full 16-bit noise samples.

The shift register method is ideally suited for hardware implementation. With very little logic (three IC packages costing less than three dollars), one can set up a random number peripheral that will pass anyone's test for randomness including nonrepeatability of results. The circuit in Fig. 15–3 shows a 26-stage shift register with feedback logic that would be iterated by the microcomputer's clock. Up to 14 (8 are shown) of the register bits are available for connection to an input port. If the clock phase used to trigger the register is chosen properly, there is no danger of the register changing while the computer is reading it. The 2^{26} sequence length would run over 30 sec alone, but variations in program execution time make it highly unlikely that any repetition could ever be detected. Skeptics can substitute a type 4031 64-bit register for the type 4006 18-bitter and have a period of nearly 150 million years at 1 MHz.

Digressing for a moment, such a circuit makes an excellent analog noise generator as well. One simply clocks it at 400 kHz or more (an R-C oscillator

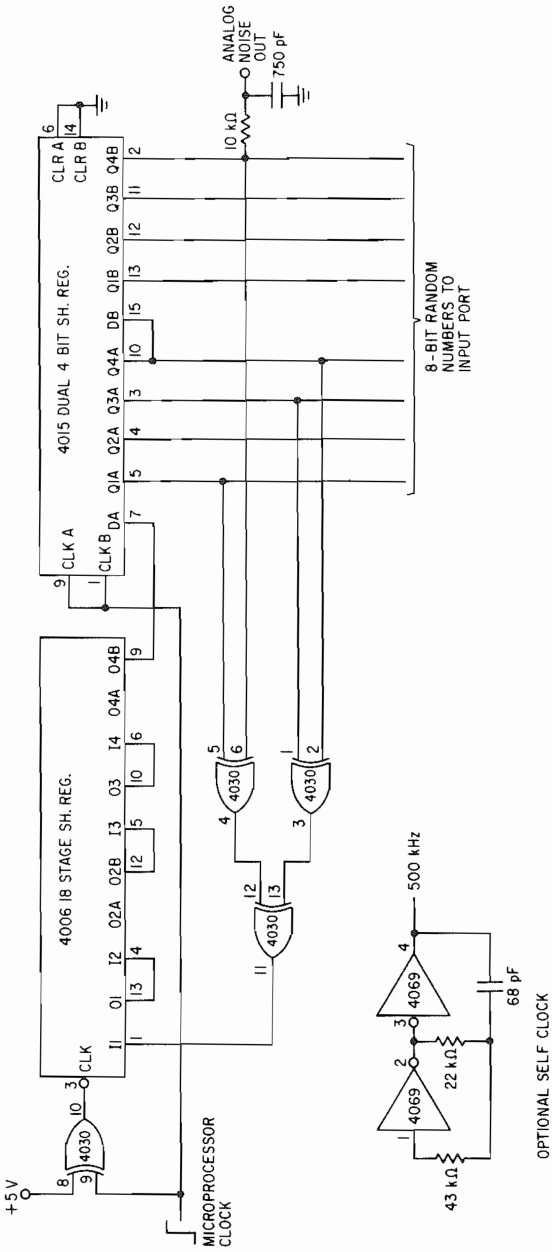


Fig. 15-3. Random-number peripheral

is fine) and runs one of the register bits through an R-C low-pass filter with a cutoff around 20 kHz. The output sound and waveform are indistinguishable from diode noise sources and are much less susceptible to hum pickup. Also, the output amplitude is repeatable and stable, a virtue not shared by diodes.

A modification to the shift register method will greatly increase its efficiency as a white noise subroutine. Essentially, the register and exclusive-ors are turned inside-out, which results in several of the register bits changing in an iteration rather than one. Assuming that the register length is the same as the word length, the following steps are performed for an iteration:

1. Shift the registêr left bringing in a zero on the right and putting the overflow bit into the carry flag.
2. If the carry flag is off, the iteration is complete.
3. If the carry flag is on, flip selected bits in the register. This may be accomplished by exclusive-oring a mask word with the register contents. The iteration is now complete.

In a 16-bit machine, these steps may require as few as three instructions to generate a sample of quite acceptable, if not statistically perfect, white noise. The table below lists mask words for common computer word lengths.

Word length	*Sequence length*	*Mask in hexadecimal*
8	265	1D
12	4095	1D9
16	65535	1D87
24	16777215	1D872B
32	4294967295	1D872B41

Using the Random Numbers

The output of the random number generators just discussed is a string of unsigned integers. However, since they are random, one can interpret them as standard twos-complement numbers as well or even as binary fractions. Since twos complement is slightly asymmetrical (there is one more possible negative value than possible positive values), the mean of the sequence will be −0.5 of the least significant bit rather than 0. This almost never causes problems unless the sequence is integrated for long periods of time.

Although the output of a random number generator when sent through a DAC sounds like white noise and in fact gives a white Fourier transform, it does not look at all like natural white noise. The difference is its probability density function, which is uniform rather than gaussian. As was mentioned in Chapter 10, one can easily convert uniformly distributed random numbers into near-gaussian distributed numbers by adding up 12 of them (assuming a range of 0 to 1.0) and subtracting 6.0 from the sum. The mean of the result will be 0 (except for the error described above) and the standard deviation will be 1.0. The simulation is not exact because the probability of a result greater than 6 standard deviations from the mean is 0 when it should be about 2×10^{-9}. Actually, since uniformly distributed and gaussian-distributed numbers sound the same and even may look the same after filtering, there is little reason to perform the gaussian conversion.

In many synthesis or modification applications, one may need several uncorrelated sources of white noise simultaneously. If the random number generator is reasonably good, one can simply distribute successive numbers to the separate processes requiring them. Thus, one random number generator can be used to simulate any number of random processes.

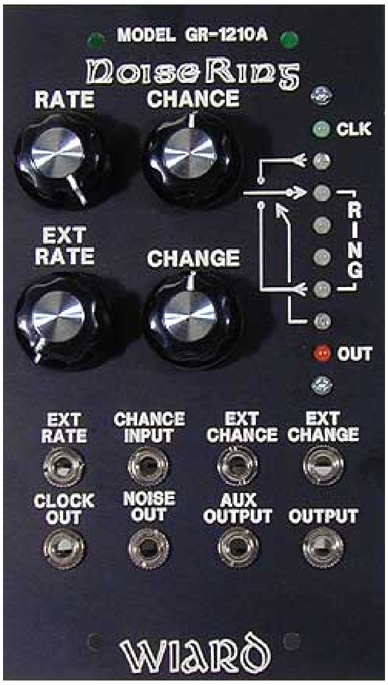
Type 3 Percussive Sounds

Now that we have a source of white noise lets discuss how it can be used to synthesize Type 3 percussive sounds. A pure Type 3 sound may be generated by filtering the noise and applying a suitable amplitude envelope. Very simple filters are adequate for many common sounds. For example, a quite deadly sounding gunshot may be produced by high-pass filtering the noise with a 300 Hz to 500 Hz cutoff and applying an envelope with zero attack time and decay to −30 dB in 100 msec to 200 msec. Conversely, a cannon boom is noise low-pass filtered at 100 Hz to 200 Hz with a somewhat longer attack and decay than the gunshot. Returning to musical instruments, high-pass filtering above 1 kHz with a 50 msec to 100 msec attack and decay simulates brushes (lightly on a snare drum) quite well. Maracas sound best with a little lower cutoff frequency and shorter attack and decay. Cymbal crashes have about the same frequency distribution as maracas but a fast attack and long decay in the 0.5-sec to 1-sec range as well as high overall amplitude.

Some sounds are not pure Type 3. A standard snare drum beat is the best example and consists of a burst of virtually white noise (only the very lowest frequencies are missing) combined with a Type 1 or Type 2 drum sound. A very realistic close range (remember how it sounded when one passed just 5 feet away in a parade?) bass drum sound can be produced by amplitude modulating 500-Hz low-pass filtered noise with the primary 50 Hz damped sine wave and mixing the two together.

Noise may also be bandpass filtered to simulate other classes of sounds. Many types of drums are more easily synthesized with filtered noise than several damped sine waves and sound just as good if not better. The bass drum, for example, can be done by bandpass filtering in the 50-Hz range as in Fig. 15–4. Pitched drums such as tom-toms also come out well if somewhat higher center frequencies are used. Sometimes it may be necessary to cascade two bandpass filters to provide greater attenuation of frequencies far removed from the center frequency, which would otherwise detract from the naturalness. It should be noted that bandpass filters not only take time to die out after the input is removed but also take time to reach full output if the input is suddenly applied. In some cases, the filter itself may generate a suitable envelope simply by turning the noise input on and off. Single-pole R-C low-pass filters also have a finite build-up time, but it is relatively short.

Wiard Model 1210 "Noise Ring" Data Resonator



Faceplate of the Model 1210
3" Frac-Rac compatible module with Blacet power connector
A source of random voltages and noise-tone hybrid sounds

Introduction:

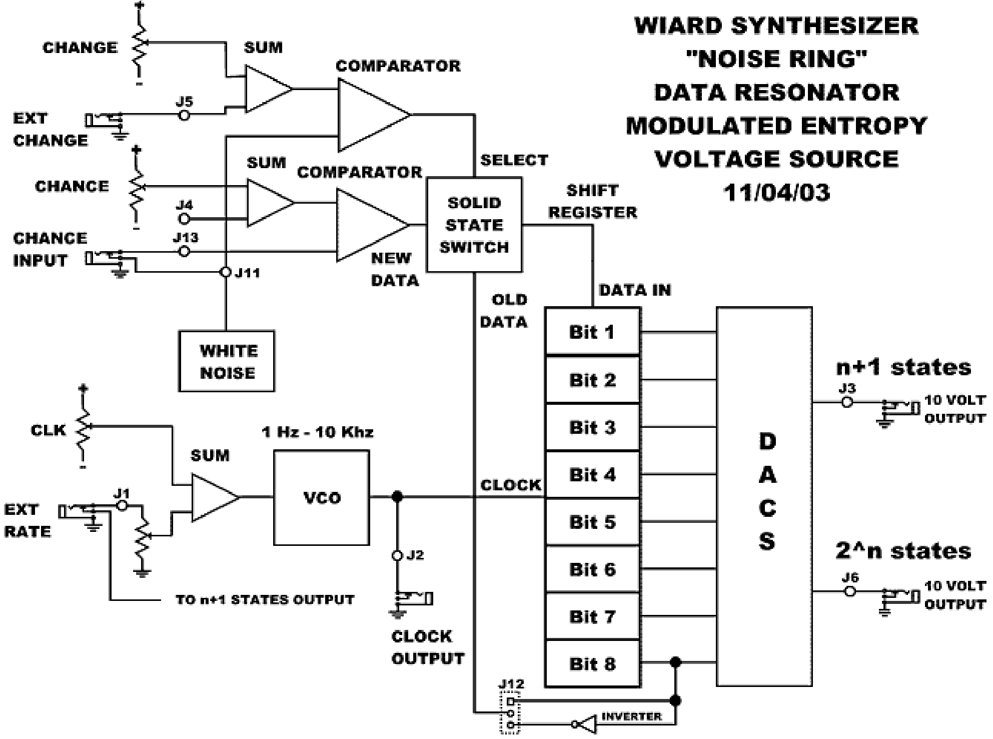
NOTE: The empty socket on the PCB is for test or expansion. It is normally empty! The pinout for the socket J10 is at the end of the manual.

In latter model synthesizers, digital noise sources began to appear in place of analog ones. Traditionally, a psuedo-random shift register set up for optimal length. By optimal length, it is meant that every state of all available bits will appear at some time, but the order is unknown. Essentially a counter that counts in an unknown order. This represents the maximum state of information "entropy" available for that number of bits.

But music has close self-similarity over short periods of time. That is, it repeats itself with changes appearing slowly. This shift register generator is designed to give control of the rate of appearance of new information. It has a tight set of controls over how random it actually is and how fast change occurs.

Knob Twiddling:

A longer sample involving just adjusting the various settings on the module by itself: Playcontrol1.mp3



Block Diagram of the Model 1210

Tone Wheels and Noise Rings:

Clocked at audio rates, the shift register forms a kind of synthetic "tone wheel", similar in idea to the electromechanical tone wheels in organs. But with very crude quantization, and hence very "noisy". To some extent you can visualize the resistor summers (DACS - digital to analog converters) like the lines of magnetic force around a metallic tone wheel. Different summers form different kinds of pickups. It is actually a digital transversal filter, but the "pickup" idea is helpful.

All voltage ranges are from 0 to 10 volts. The Aux Output (n+1) will change between one of 9 voltage levels. The Output (2^n) will change between one of 256 voltage levels.

Noise Source:

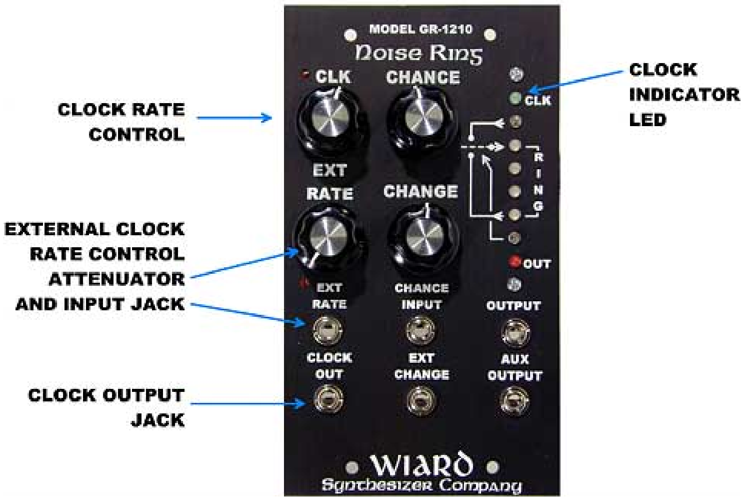
The module contains an internal analog white noise generator. Random digital data is generated by using two comparators to compare the noise voltage against two DC voltages. One comparator generates a controllable density of zeros or ones. The second comparator controls the solid state switch which which selects between recycling old data in the shift register, or getting new data from the outside. If the "CHANGE" control is set to 100%, ALL new data is shifted through the register. With the "CHANGE " control set to 0%, only old data is recycled through the shift register without change.

CLOCK:

A wide range voltage controlled oscillator is used to clock the shift register. The 4 decade range from 1 Hertz to 10 Kilohertz, allows the generation of audio rate sound and also control voltage sequences. The VCO has a coarse set front panel control and an external control voltage input with attenuator.

With the "CHANGE" control set to 100% and the "CHANCE" control set to 50%, sweeping the clock from high to low sounds like this at the 2^n output: Clocksweep1.mp3
With the "CHANGE" control set to 100% and the "CHANCE" control set to 50%, sweeping the clock from high to low sounds like this at the n+1 output: Clocksweep2.mp3

NOISE RING CLOCK CONTROLS

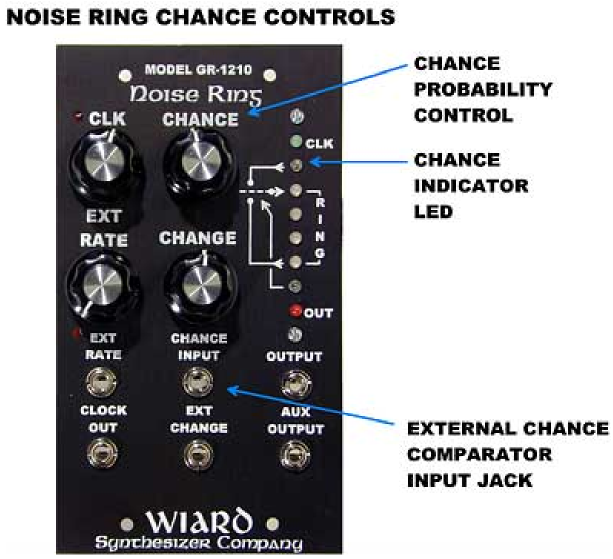


Clock controls on the Model 1210.

Clock frequency is nominally 1 Hz to 10 kHz. 0 to +10 V square wave.
Green "CLK" LED flashes when "CLOCK OUT" jack is at +10V.
The "n+1" output is normalized to the "EXT RATE" input. Increasing the "EXT RATE" control randomizes the clock time.
If an external voltage source is connected to the "EXT RATE" input jack, the "n + 1" connection is broken.

"CHANCE" Control

Controls the balance between ones and zeros in the new data. With the "CHANGE" control is set to 100% (all new data), sweeping the "CHANCE" control sounds like this at the 2^n output: Prob1sweep.mp3
With the "CHANGE" control is set to 100% (all new data), sweeping the "CHANCE" control sounds like this at the n+1 output: Prob1sweep2.mp3



NOISE RING CHANCE CONTROLS

CHANCE PROBABILITY CONTROL

CHANCE INDICATOR LED

EXTERNAL CHANCE COMPARATOR INPUT JACK

Chance controls on the Model 1210.
Chance controls the number of zeros and one extracted from the noise source.
At 7 o'clock position, all zeros are output and the "Chance indicator" LED is always off.
At 5 o'clock position, all ones are output and the "Chance indicator" LED is always on.
At 12 o'clock position, equal numbers of zeros and ones are output and the "Chance indicator" LED flickers.

Processing External Data

Oscillator:

Another oscillator can be used to supply data into the shift register. Connect any waveform into the "CHANCE INPUT" jack. In this case we hear the familiar "phased" sound of a swept shift register: Phaseshifter.mp3

At ultrasonic frequencies the heterodynes do a good approximation of a shortwave radio: ShortWave.mp3

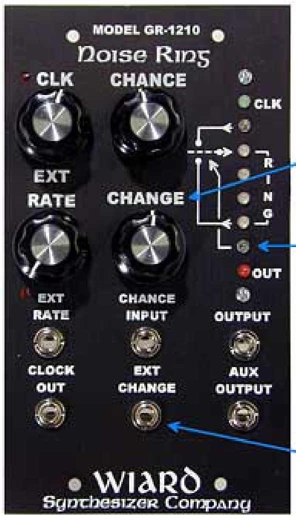
Drum Machine:

Using the Envelope Follower from an ARP 2600 controlling a Borg Filter. The drum machine input is processed through the noise ring, then has the envelope reapplied by the Borg Filter.
drum1.mp3
drum2.mp3
drum3.mp3
drum4.mp3

"CHANGE" Control

With the VCO set to the audible range, sweeping this control from 100% to 0% sounds like this at the 2^n output: Prob2sweep.mp3
With the VCO set to the audible range, sweeping this control from 100% to 0% sounds like this at the n+1 output: Prob2sweep2.mp3

NOISE RING CHANGE CONTROLS



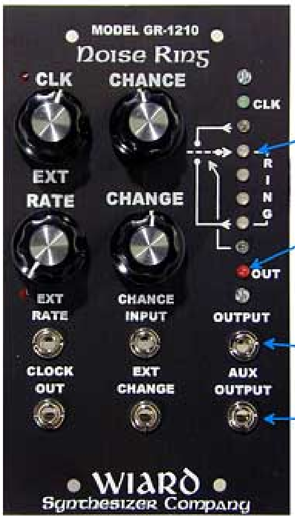
CHANGE PROBABILITY CONTROL

CHANGE INDICATOR LED

EXTERNAL CHANGE CONTROL VOLTAGE INPUT JACK

Change controls on the Model 1210.
Change controls the number of new zeros and one which are let into the ring.
At 5 o'clock position, all old data is used and the "Change Indicator" LED is always off.
At 7 o'clock position, all new data is used and the "Change Indicator" LED is always on.
At 12 o'clock position, equal numbers of old and new data are used and the "Change Indicator" LED flickers.

NOISE RING OUTPUTS AND RING LIGHTS



n INDICATOR LIGHTS

2^n STATES INDICATOR LED

2^n STATES OUTPUT JACK

n + 1 STATES OUTPUT JACK

The "n" ring lights flicker red and green as data passes through the ring.
The orange "OUT" LED will change brightness with the 2^n output voltage.

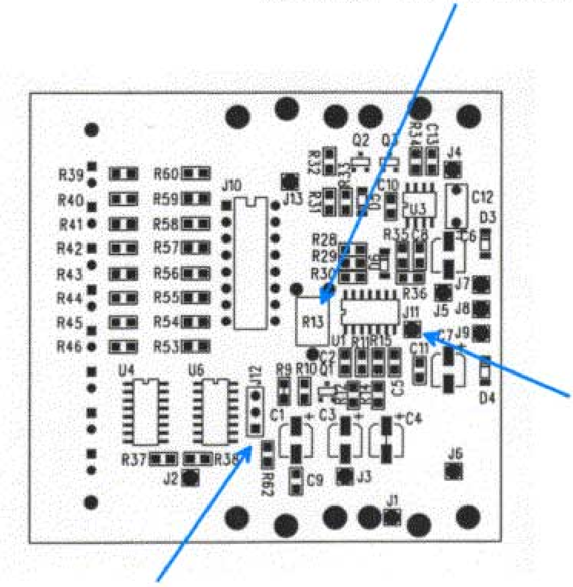
Used with a Sequencer

The following mp3s all use the same patch. The Noise Ring is used as an audible source while the parameters are varied in real time by a multi-rank sequencer. A Wiard Borg filter module is used as VCF and VCA swept by an Wiard Envelator module. This demonstrates the mixture of noise and tone that can heard coming from the hybrid oscillator.

Noise Seq 1.mp3
Noise Seq 2.mp3
Noise Seq 3.mp3
Noise Seq 4.mp3
Noise Seq 5.mp3
Noise Seq 6.mp3
Noise Seq 7.mp3
Noise Seq 8.mp3
Noise Seq 9.mp3

TECHNICAL INFORMATION

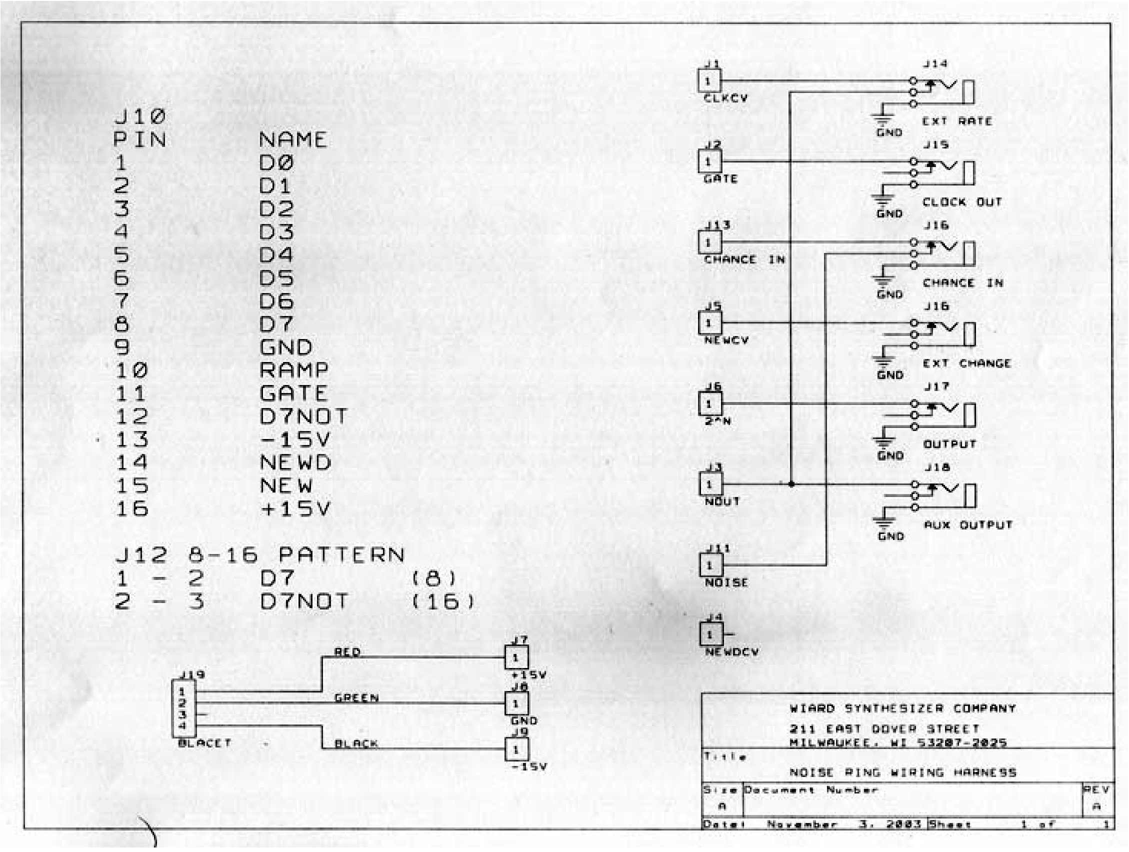
NOISE AMPLITUDE ADJUSTMENT R13
ADJUST TO 10 VAC PEAK TO PEAK



NOISE TEST POINT J11

J12 8-16 SELECT
PIN 1

Only 1 trim point on module. Move red jumper on J12 to invert data recirculated through shift register.



J10

PIN	NAME
1	DØ
2	D1
3	D2
4	D3
5	D4
6	D5
7	D6
8	D7
9	GND
10	RAMP
11	GATE
12	D7NOT
13	-15V
14	NEWD
15	NEW
16	+15V

J12 8-16 PATTERN

1 - 2	D7	(8)
2 - 3	D7NOT	(16)

WIARD SYNTHESIZER COMPANY
211 EAST DOVER STREET
MILWAUKEE, WI 53207-2025

Title
NOISE RING WIRING HARNESS

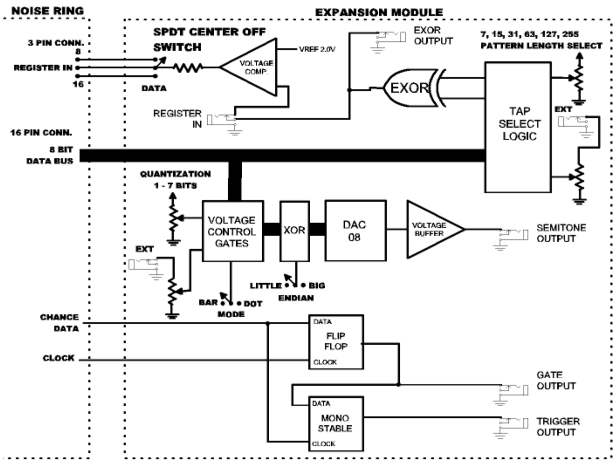
Size A Document Number REV A

Date: November 3, 2003 Sheet 1 of 1

Wiring harness of the Noise Ring Module. J10 is test connector.

[home] [modular] [accessories] [purchase]

Return to Wiard Homepage


BLOCK DIAGRAM OF ENTROPY EXPANDER FOR NOISE RING

The reason it exists is because I figured out another algorithm that can run on the same set of hardware. The mathematics of information theory as applied to music predict that it is possible that some portions of this new algorithm MAY have musical usefulness.
That is all we had to go on the first time, so what the heck.
Our story begins with the "Quantized Random Voltages" section of the Buchla Model 266.
http://www.musicsynthesizer.com/Buchla/source_of_uncertainty.htm

It was absolutely essential that the Noisering NOT be a 266, because Mr. Buchla had done that already, so repeating his work would not advance the state of the art.
I did however incorporate the 2^N and N+1 output structure. But there is no psuedo-random sequence generator (PRSG) in the Noise Ring currently.

A PRSG is one kind of way to control the contents of a shift register. And if you grew up on Don Lancaster, only a maximal length PRSG is desirable.

Then I find a paper by Ralph Burnams "Harmonic Content of Sub-Optimal Psuedo Random Generator Sequences". And it turns out that that the original CD4006 implementation of an optimal PRSR is only one of a spectrum of solutions, each of which has a unique audio harmonic signature.

So there is one way to spice up Mr. Buchla's original algorithm, add voltage selectable sub-optimal sequence control. So the length control selects sequences of 15,31,63,127,255 under voltage control. They values are shown on LEDs. Don't worry, beat one of the repeat is at 16, 32 etc.

In beta testing the good Dr. Mabuse (aka Mike Murphy) detected that any static sequence sounds, well, static. HOWEVER by dynamically modulating the count length, you got an effect at some settings, that were musical "keepers". That is a good enough recommend to me that I am willing to go ahead with the rest of it. "Sometimes the Universe gives you clues about your destiny, you should listen"- George Clooney

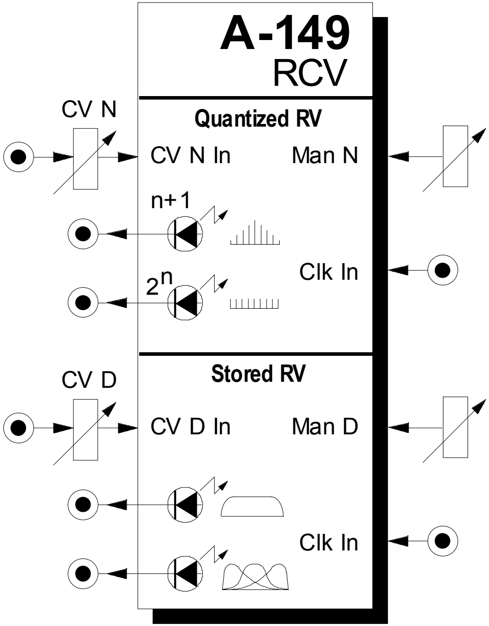
OK, so we have something new for sequence control and animation in the audio domain, what about as a controller?

The "quantization" knob on the 266 is just Shannon's Measure of Information Entropy. Which is just "how many levels in the signal, expessed in bits, dude".

Now if we make each level be the 1 volt per octave voltage of a semi-tone, we have a musically quantized output. But if we copy the 266 design exactly, then we can only control the entropy from big to small. What if we want to control it from small to big, like an expanding range of semi-tones.

So we need an "Endian" control. Technically that is the term for the order of bits. Microsoft being the reverse of MACs (of course).

The "Gate" and "Trigger" circuits came from "Musical Applications of Microprocessors" by Hal Chamberlain.



Remark: The A-149-1 can be expanded by the **A-149-2** module (8 digital random outputs with LED displays)

1. Introduction

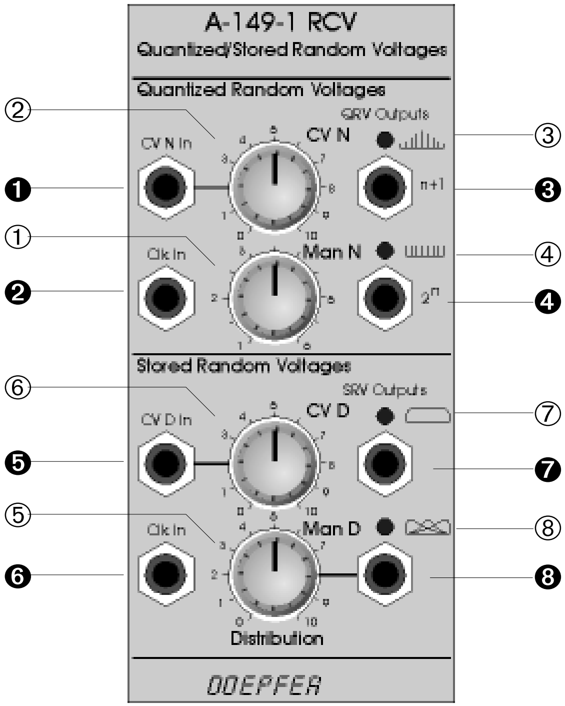
Module **A-149-1** is a **Random Control Voltage Source** based on the idea of Don Buchla's "**Source of Uncertainty 265/266**" modules. It has available **4 analog random voltages,** that are generated in different ways.

The ***Quantized Random Voltages*** section has available the outputs ***N+1*** and ***2^N States***. **N** is an integer number in the range 1...6 that can be adjusted manually (Man N) and by an external control voltage CVN. The voltage steps are 1/12 V for the 2^N output (i.e. semitone intervals) and 1.0 V for the N+1 output (i.e. octave intervals).

The ***Stored Random Voltages*** section has available an output with ***even voltage distribution*** with 256 possible output states and another output with ***adjustable voltage distribution probability.*** The distribution of this output can be adjusted manually (Man D) and by an external control voltage CVD. The voltage range is 0...+5 V for both stored random outputs.

The rising edge of the corresponding **Clock input** signal triggers a new random voltage value at the outputs.
Each output is equipped with a **LED** that displays the current output voltage.

2. Overview



Controls and Indicators:

① **Man N** : manual control of “N”

② **CV N** : attenuator for CVN at input ❶

③, ④ **LED** : display for output ❸ resp. ❹

⑤ **Man D** : manual control of distribution "D"

⑥ **CV D** : attenuator for CVD at input ❻

⑦, ⑧ **LED** : display for output ❼ resp. ❽

In - / Outputs:

❶ **CV N In** : CV input for “N”

❷ **Clk In** : clock input for Quantized RCV section

❸ **n+1** : N+1 states output

❹ **2^n** : 2^n states output

❺ **CV D In** : CV input for distribution “D”

❻ **Clk In** : clock input for Stored RCV section

❼ : output with equal probability distribution

❽ : output with adjustable probability distribution (D)

3. Controls

3.1 Quantized Random Voltages

① Man N

This is the manual control for the integer **number N** in the **range 1 to 6.** It defines the **number of possible states** at the outputs ❸ and ❹:

	Possible states of	
N	**Output n+1**	**Output 2^n**
1	2	2
2	3	4
3	4	8
4	5	16
5	6	32
6	7	64

Remark:

As N increases **n+1** increases **linear** too but **2^n increases exponentially.**

☞ The **final value of N** is the sum of the manual control ① and the external (attenuated) control voltage applied to input ❶.

② CVN

The external control voltage CVN fed into input ❶ is attenuated with this control.

③ LED • ④ LED

The brightness of each LED is proportional to the output voltage at the corresponding output ❸ resp. ❹.

3.2 Stored Random Voltages

⑤ Man D

This is the manual control for the **probability distribution** of the **256 states** appearing at output (.

With the control set fully counterclockwise most of the random voltages will be low magnitude but even medium and high magnitude voltages may appear but with smaller probability. As the control is turned to the right (or a positive control voltage appears at the CVD input) the distribution moves through medium to high magnitude voltage probability. The symbol at the lower jack ❽ socket shows this coherence graphically (see also fig. 1).

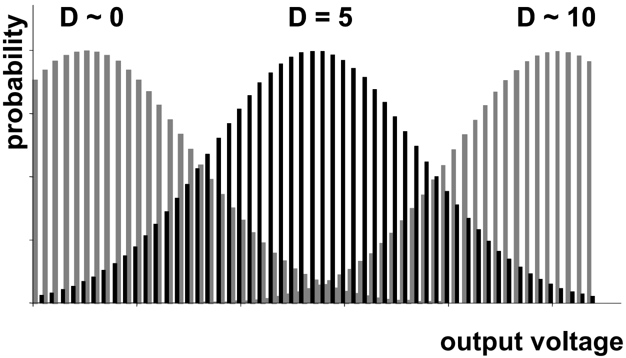


Fig.1: Probability distribution for different settings of "D"

☞ The **final value of D** is the sum of the manual control ⑤ and the external (attenuated) control voltage applied to input ❺.

⑥ CV D

The external control voltage CVD fed into input ❽ is attenuated with this control.

⑦ LED • ⑧ LED

The brightness of each LED is proportional to the output voltage at the corresponding output ❼ resp. ❽.

4. In- / Outputs

4.1 Quantized Random Voltages

❶ CV N In

This socket is the **Control Voltage input** for the parameter **"N"**.

❷ Clk In

This socket is the **Clock input** for the Quantized Random Voltages section. Each rising edge of this signal causes the generation of a new random voltage at the outputs ❸ resp. ❹. Any clock or gate signal can be used to control this input.

❸ n+1

This socket outputs the random voltage with **n+1 states**. The **voltage range** for this output is **0 to +5 V**, the **voltage steps** are **1.0 V** (i.e. 1V quantization). This corresponds to octave intervals when used to control the pitch of a VCO.

❹ 2^n

This socket outputs the random voltage with **2^n states**. The **voltage range** for this output is **0 to +5.25 V**, the **voltage steps** are **1/12 V** (i.e. 1/12 V quantization). This corresponds to semitone intervals when used to control the pitch of a VCO.

4.2 Stored Random Voltages

❺ CV D In

This socket is the **Control Voltage input** for the **probability distribution "D"**.

❻ Clk In

This socket is the **Clock input** for the Stored Random Voltages section. Each rising edge of this signal causes the generation of a new random voltage at the outputs ❼ resp. ❽. Any clock or gate signal can be used to control this input.

❼ • ❽

These sockets output the random voltages of the Stored Random Voltages section. Socket ❼ is the output with **equal probability distribution**, socket ❽ outputs the voltage with **adjustable distribution "D"**. The **voltage range** for both outputs is **0 to about +5.3 V**, the **voltage steps** are about **1/48 V** (i.e. 1/48 V quantization). This corresponds to about 1/4 semitone intervals when used to control the pitch of a VCO.

5. User examples

The **Doepfer web site** www.doepfer.com shows some **typical examples** of the A-149-1, including sound examples in the mp3 format. Even more details concerning the technical realization of the module can be found. An excellent description of several applications of random voltages like those generated by the A-149-1 can be found in the Allen Stranges "Electronic music - systems, techniques and controls" from page 82. The examples in this book are based on Don Buchla's modules 265/266 but are valid for the A-149-1 too.

The following patch is taken from this book and shows how to create very complex permanently changing sound structures by means of the A-149-1 in combination with the voltage controlled LFO A-147 and some additional standard modules (VCO, VCF, VCA, ADSR):

A high magnitude voltage at the N+1 output of the A-149-1 causes a high VCO pitch and simultaneously sets the value of N higher so that the next pitch is taken from a greater range of possibilities. If the N+1 output is low the VCO pitch will be low too and sets the value of N so that the next pitch will have a more restricted range of possibilities. Simultaneously the 2^n output controls the frequency of the filter and the tempo of the VCLFO A-147. Thus as the range of pitch selection increases the number of possible spectral ranges becomes exponentially (or geometrically) greater. As the tempo of the VCLFO is controlled by the 2^n output too, bright sounds are accompanied by longer events, longer events are accompanied by greater range pitch range possibilities and the number of of range probablities for pitch selection is correlated exponentially. This tail-chasing configuration may last a few hours (to obtain Allen Strange's original patch a voltage inverter A-175 has to be inserted between the 2^n output and the control input of the VCLFO as the CV input of A-147 controls the tempo rather than the period).

More examples with random voltage sources can be found in Allen Strange's book from page 80 (e.g. the "Dream machine" on page 85).

Some additional ideas:

- Use the RND Clock output of an **A-117** Digital Noise Generator as clock input for the A-149-1 to increase the randomness of events.
- Use the Quantizer module **A-156** to obtain more restricted pitch voltages (e.g. only notes from major/ minor scale/chords)
- Combine the A-149-1 with a A-155 sequencer (common clock) to obtain random envelopes (A-142), timbre (filters), loudness (VCA) or stereo position (VC panning A-134), frequency shifting (A-126)

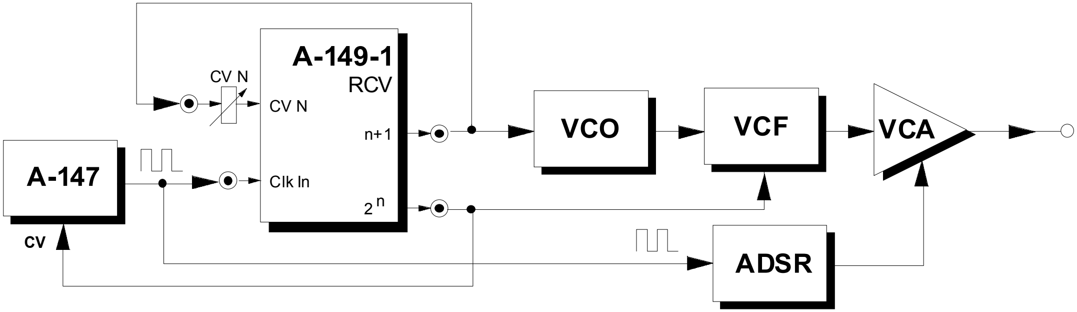


Fig. 2: "Random patch" adapted from Allen Strange's book "Electronic music - systems, techniques and controls"

Music as a Gradual Process

Steve Reich, 1968

I do not mean the process of composition, but rather pieces of music that are, literally, processes.

The distinctive thing about musical processes is that they determine all the note-to-note (sound-to-sound) details and the over all form simultaneously. (Think of a round or infinite canon.)

I am interested in perceptible processes. I want to be able to hear the process happening throughout the sounding music.

To facilitate closely detailed listening a musical process should happen extremely gradually.

Performing and listening to a gradual musical process resembles:

- pulling back a swing, releasing it, and observing it gradually come to rest;
- turning over an hour glass and watching the sand slowly run through the bottom;
- placing your feet in the sand by the ocean's edge and watching, feeling, and listening to the waves gradually bury them.
- Though I may have the pleasure of discovering musical processes and composing the musical material to run through them, once the process is set up and loaded it runs by itself.

Material may suggest what sort of process it should be run through (content suggests form), and processes may suggest what sort of material should be run through them (form suggests content). If the shoe fits, wear it.

As to whether a musical process is realized through live human performance or through some electro-mechanical means is not finally the main issue. One of the most beautiful concerts I ever heard consisted of four composers playing their tapes in a dark hall. (A tape is interesting when it's an interesting tape.)

It is quite natural to think about musical processes if one is frequently working with electro-mechanical sound equipment. All music turns out to be ethnic music.

Musical processes can give one a direct contact with the impersonal and also a kind of complete control, and one doesn't always think of the impersonal and complete control as going together. By "a kind" of complete control I mean that by running this material through the process I completely control all that results, but also that I accept all that results without changes.

John Cage has used processes and has certainly accepted their results, but the processes he used were compositional ones that could not be heard when the piece was performed. The process of using the *I Ching* or imperfections in a sheet of paper to determine musical parameters can't be heard when listening to music compsed that way. The compositional processes and the sounding music have no audible connection. Similarly in serial music, the series itself is seldom audible. (This is a basic difference between serial (basically European) music and serial (basically American) art, where the perceived series is usually the focal point of the work.)

What I'm interested in is a compositional process and a sounding music that are one and the same thing.

James Tenney said in conversation, "then the composer isn't privy to anything". I don't know any secrets of structure that you can't hear. We all listen to the process together since it's quite audible, and one of the reasons it's quite audible is, because it's happening extremely gradually.

The use of hidden structural devices in music never appealed to me. Even when all the cards are on the table and everyone hears what is gradually happening in a musical process, there are still enough mysteries to satisfy all. These mysteries are the impersonal, unattended, psycho-acoustic by-products of the intended process. These might include sub-melodies heard within repeated melodic patterns, stereophonic effects due to listener location, slight irregularities in performance, harmonics, difference tones, etc.

Listening to an extremely gradual musical process opens my ears to *it*, but *it* always extends farther than I can hear, and that makes it interesting to listen to the musical process again. That area of every gradual (completely controlled) musical process, where one hears the details of the sound moving out away from intentions, occuring for their own acoustic reasons, is *it*.

I begin to perceive these minute details when I can sustain close attention and a gradual process invites my sustained attention. By "gradual" I mean extremely gradual; a process happening so slowly and gradually that listening to it resembles watching a minute hand on a watch--you can perceive it moving after you stay with it a little while.

Several currently popular modal musics like Indian classical and drug oriented rock and roll may make us aware of minute sound details because in being modal (constant key center, hypnotically droning and repetitious) they naturally focus on these details rather than on key modulation, counterpoint and other peculiarly Western devices. Nevertheless, these modal musics remain more or less strict frameworks for improvisation. They are not processes.

The distinctive thing about musical processes is that they determine all the note-to-note details and the over all form simultaneously. One can't improvise in a musical process--the concepts are mutually exclusive.

While performing and listening to gradual musical processes one can participate in a particular liberating and impersonal kind of ritual. Focusing in on the musical process makes possible that shift of attention away from *he* and *she* and *you* and *me* outwards towards *it*.