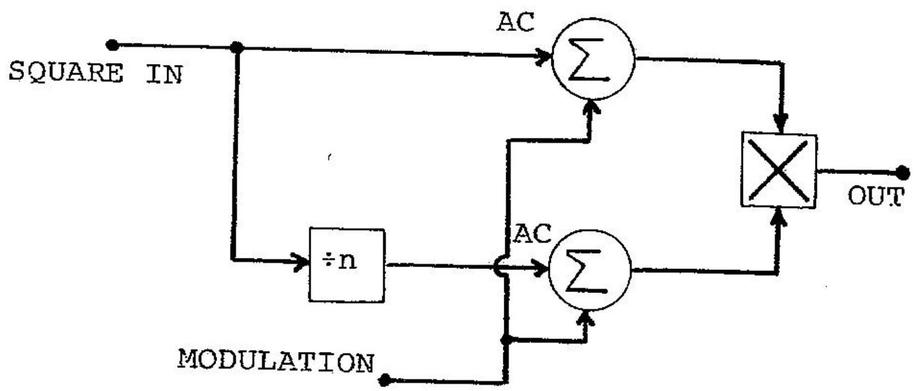
This suggests the following form of pulse modulation. The square wave input is frequency divided and both signals are then AC coupled to seperate summing amps. With no modulation input, the output is an EXOR operation. For either ±1/2 of the square wave p-p voltage as modulation input, the output is an AND operation with stepped waveform outputs for intermediary modulation values.

Fig. 2



* * * * * * * * * * * * * * *

SOME NOTES ON PSEUDO-RANDOM NOISE GENERATORS:

-by Bernie Hutchins

1. LOW CHIP NUMBER REALIZATION WITH 4006 CMOS:

Pseudo-Random (PR) noise generators are attractive because they are digital and offer a well defined level of noise at the output (unlike diode or transistor based noise generators which have to be adjusted for the noise level of the individual device). They also have certain advantages for random (pseudo-random) control devices.

One problem with PR generators is that they can involve a large number of digital IC chips. While there are single eight-pin mini-DIP noise generators, their output levels may not be standard for interface, and the clock is internal to the chip. A good choice for the shift register needed (see Musical Engineer's Handbook for more information) is something like the 74C164 eight-stage shift register. However, for a reasonably long noise sequence, you need at least three of these. An attractive alternative is to consider using the standard CMOS type 4006, which is an 18 stage shift register, which is available at about a dollar. The chip is composed of four sub-registers as shown below:

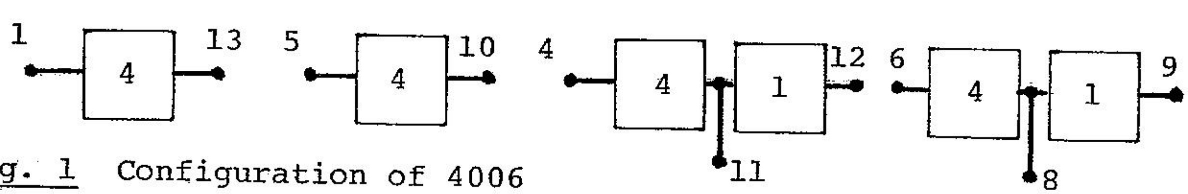


Fig. 1 Configuration of 4006

The problem can be posed as follows: Use only the 4006 and exclusive or gates as needed. If possible, make an 18 stage PR generator (feedback taps on stages 11 and 18), and if not, try a 17 stage PR generator (feedback taps on stages 14 and 17). It is easy to see that if we are to make an 18 stage PR generator, we need to use all the units of Fig. 1 in some sort of series. To get the required 11th tap, you need to get a tap seven back from the end. A bit of study will show that this is not possible, although you can get taps at 10, 12, 13, 14, and 17. So forget about an 18 stage PR generator. What about 17 stages?

A 17 stage PR generator requires a tap at stage 14, in the manner of Fig. 2 below. Let us say right here that it is possible to achieve this if one is sufficiently clever [The solution we will give here is one devised by Tim Fischer while he was a student at Cornell in 1980].

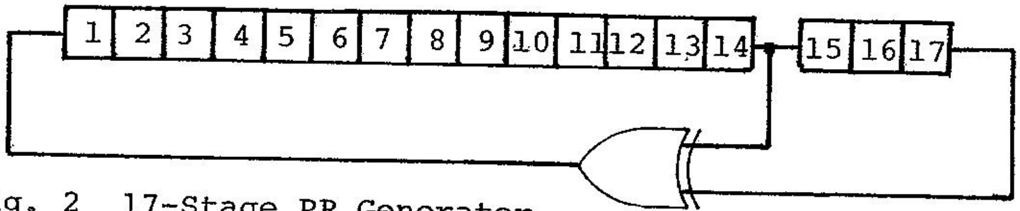


Fig. 2 17-Stage PR Generator to be realized with the 4006 chip.

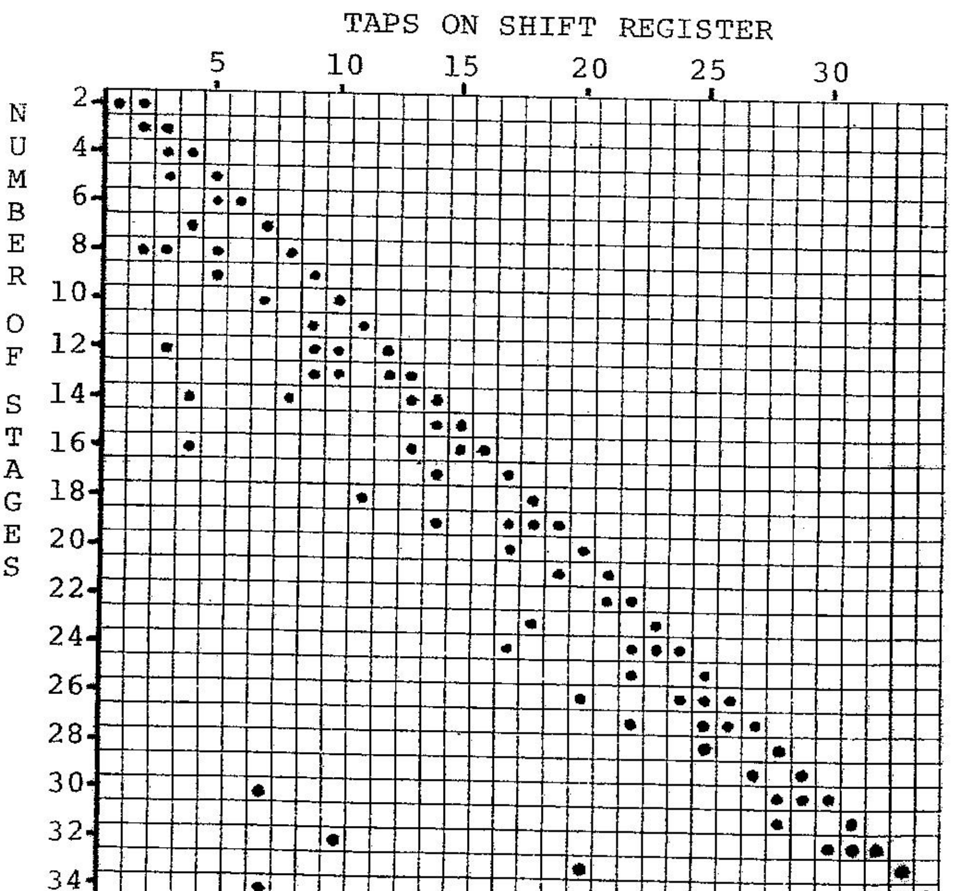
At first sight, it looks as though you are not going to get this one either, as it requires a difference of three stages between taps, and the 4006 offers only differences of one or four. Can you figure it out? Take a shot at it first, and if you can't get it, the answer is printed at the bottom of the next page. Note that the solution ends up with only a 4006 chip and 1/4 of an exclusive-OR gate package (such as a 4070) so three exclusive-OR gates are left over. Perhaps these extra three gates can be used to form a clock to drive the shift register. [I tried this and it did not quite work. The standard clock circuits seemed to slow to trigger the register. (?)]

2. POSSIBLE BENEFICIAL EFFECTS OF EARLY TAPS

One of the fascinations about PR generators is that they have many of the properties of random generators, but are not really random. In fact, if one listens closely to some PR generators, it is possible, and at times quite easy, to hear the "heartbeat" of the generator. This is due to the actual repeating of the PR sequence.

Fig. 3

Tap feedback (for modulo-2) data for the setup of PR generators of length 2 to 34. Dots in boxes indicate the position of a tap. Compare with diagram for 17 stage PR generator in Fig. 2 above.



The cause of the "heartbeat" is not clear, but the fact that the ear-brain is apparently processing very large numbers of bits and then coming up with the notion that the sequence is actually repeating is extremely interesting from a psycho-acoustic point of view. Evidently the repetition may be detected directly for short sequences (lenghts of 10 stages or less, where the sequence length is only about 1000 bits). In the case of longer register lengths, say 17, where the sequence length is 131,071, it is probably some series of "markers" in the sequence (short structured sub-sequences detected by the ear) that make the repetition detectable. From a synthesizer design point of view, it is probably a question of just how evident a heartbeat effect is that will be our major interest. [We don't want users to notice a pulsation in their noise. Often this is not a problem, as when the noise serves as raw sound for synthesis of short notes. Note also that even when a heartbeat is not heard in the raw sound, a PR noise generator may make itself evident, as with the use of a sample-and-hold to generate PR tunes for example.]

The study of the periodic properties of PR sequences is not particularly easy, as it may be principally subjective, as when we listen carefully to a heartbeat effect, or it may be very time consuming if we play the sequence slowly enough to examine individual steps. One method of studying sequences fairly slowly is to clock them down to perhaps 5 - 10 Hz, and feed the output to a VCO as suggested in Fig. 4. Various adjustments of the depth of modulation and the initial frequency can be made, but you will probably end up with something that sounds more like Morse Code than anything else. The bits of the sequence are converted to a series of frequency shifts - a two pitch PR tune. It is boring to listen to of course after a few moments, but it is easier to monitor that, for example, watching a voltage level change on a scope face.

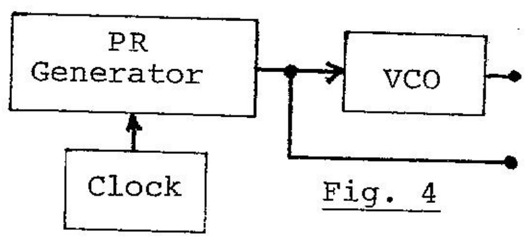


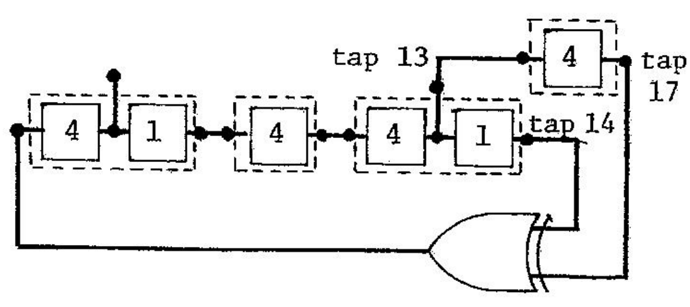
Fig. 4

In a typical study, the generator would be made to first run fast, and events in its cycle would be noted (e.g., "bell" 1/3 of way through, "chirp" 1/2 way through, "thud" almost at end, etc.). Then the sequence is slowed down and you listen to the VCO output. To avoid fatigue, take steps to move rapidly to the general area of the events. A typical event that you might then come across would be a long series of high bits, followed by a series of lows, and so on for a portion of the sequence.

Now, as we have noted before, these "events" may be due to the fact that a unique subsequence gets on the register. Once on, it comes off slowly, and often will then generate nearly the same, but complimentary, subsequence, which will then in turn have to work its way off. The problem is the fact that feedback taps tend to be located toward the far end of the shift register, as can be seen by a glance at the chart

---

Tim Fischer's Solution to the 4006 17-Stage PR Generator Problem



of Fig. 3. You always have to have the last stage tapped (or else how would you know it was there?), and there is then either one other tap, or three other taps, which tend to be near the far end. Recall that the choice of taps is made first so that the sequence is maximal length, and secondly so that there are relatively few taps (so that we don't need a lot of exclusive-OR gates). There are many many other tap choices possible. Perhaps we should look at some that are more complicated in terms of taps, if they will give us "smoother" noise.

We can speculate that unique subsequences are "captured" on the register because the taps are near the end. Thus no change at the input is possible until this unique subsequence passes by the feedback taps. If the taps are near the end, they are not reached until after the unique sequence has had considerable opportunity to regenerate itself or its compliment. Eventually it will work off as it must, but the "mixing" is slow. What if there were more taps, and in particular, if there were earlier taps. The mixing would be better. How can we check this idea?

Ideally, we would look at a number of different tap choices for a shift register length that is fixed. We have not done this here since calculations must be made, and hardware must be built. It is possible however to look at those few cases where there are taps early along the register to see if these are devoid of obvious features. The choices are (see Fig. 3) 16, 30, 32, and 34 stages. [Choices of fewer that 16 stages are probably just too short to avoid a direct detection even without features.] Of the available choices, 30, 32, and 34 are probably too long to examine this problem. They make excellent noise generators, but the sequence lengths are over one billion, four billion, and 17 billion respectively for 30, 32, and 34, so even at MHz clock rates, you aren't going to hear them cycle. This leaves 16 stages as the only practical case to examine with available tap data. We can compare it with 17 stages, which does have a heartbeat. There is some apparent improvement in going from 17 to 16, but the case is not clear. Even if it were clear, we can not judge just from this one case as to whether the improvement is due to the early tap on the 16 stage register or perhaps due to something else we do not know about yet.

While no strong conclusions can be made, we can see that a 32 stage shift register PR generator is attractive. First, it has over four billion bits before a repeat. Secondly, it has an early tap (at 7). Thirdly, it would use all of four 74C164 8-stage shift register IC's efficiently.

3. SOME IDEAS USING SAMPLE-AND-HOLD WITH PR GENERATORS:

It is sometimes surprising to learn that a PR generator which produces a sequence consisting of random (apparently) bits that are either 1 or 0 still sounds like white noise. We generally expect noise to have random levels, and not be restricted to just two levels. Yet it is true that the PR generator sounds white and is just as good as a noise-junction type of generator for electronic music applications where the noise is raw material.

However, in electronic music, noise is often used for other purposes, in particular, for the control of other devices. The sampling of noise and holding of its instantaneous voltage level is one such application. Above we suggested that there could be a problem with using a PR generator in this application, and this problem is obvious if the PR sequence is used directly, since sampling of two voltage levels only results in two possible held levels. Thus it is usually the practice to sum numerous stages, or somewhat equivalently, to low-pass filter the PR sequence, and this produces a much larger number of levels, usually a satisfactory number. Yet a problem may still remain. Suppose that the repeat time of the sequence is exactly 5 seconds say, and that the sample-and-hold is being clocked at exactly 2 Hz. It is then clear that a periodic waveform is being periodically sampled, and the result would be a constant 10 note sequence. This is bad if we are in fact looking for a subjectively random note sequence, but it has some obvious possibilities. Certain locking techniques might be used to lock the sample-and-hold into an integer ratio with the PR generator. Actually, it is somewhat difficult to do this, since the PR generator of n stages has a cycle length of 2^n-1, the (-1) being the proverbial fly in the ointment. Thus it would be simpler if they did not have to be locked. What can we do with this idea?

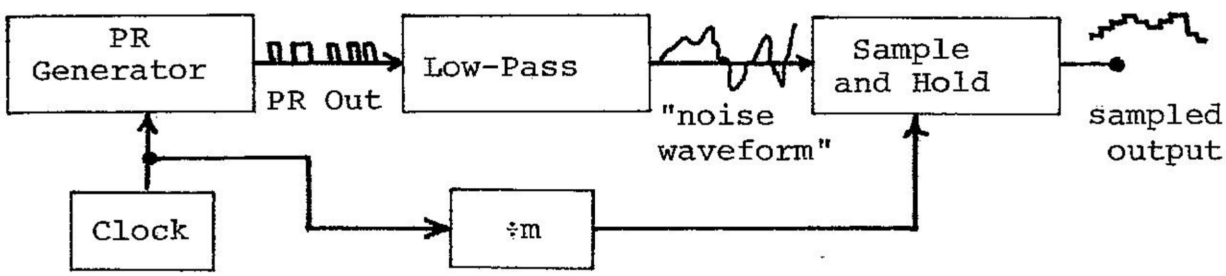


Fig. 5 A Possible Waveform Animator or Evolving Tone-Sequence Generator

Fig. 5 shows a possible setup where a clock drives a PR generator, and we can think of the PR generator as producing a "waveform" that is divided by 2^n-1, relative to the clock. This PR output is smoothened by the low-pass filter, which also serves to make available a variety of levels. At the same time, the clock is divided by a number m and this triggers the sample-and-hold. If $m = 2^n-1$ (not all that easy to do), the output sequence repeats exactly. If m is not restricted, the sequence evolves. Suppose for example $m = 2^n$ (easy to do) than the output sample moves along the noise waveform one step at a time. Interesting - possibly useful. However, if $m = 2^{n-i}$, where i is an integer, then the noise waveform is sampled 2^i times (approximately) each cycle. Yet the sampling is not exactly in sync, so the sampled sequence changes slightly on each repeat. Now, if the noise waveform were truly random, we would get basically noise at the sampled output. However, the PR nature makes a repeat possible, and the low-pass filter makes an approximate repeat possible, since it correlates successive portions of the waveform. Thus each repeat sampling of the noise waveform is slightly different, and this is true for each stage of the sampled output. You can think of it as having a 2^i step sampled output where each step is walking slowly along different portions of the noise waveform.

If the system of Fig. 5 is run slowly, the output samples can be applied to a VCO to produce an evolving tone sequence. If the clock is in fact a VCO, then the scheme may be a waveform animator of the type we have been considering lately. Here we have in mind that the PR generator is short, perhaps about n = 10. These ideas have not been tried yet.

ELECTRONOTES, Vol. 12, No. 119 [November, 1980](published May 1981)
Published by B. A. Hutchins, 1 Pheasant Lane, Ithaca, NY 14850
Phone (607)-273-8030
Please use above address and phone for subscription information, technical questions, and for all other business. Best phone hours - Wednesday evenings, 8PM through 2 AM, Eastern time. Bad times to call (unless you don't mind talking to the answering machine machine) - any morning, Monday, Tues., and Thurs. afternoons. Feel free to call any evening until midnight. To avoid the answering machine, hang up after the fifth ring if a live person does not answer.