ELECTRONOTES
1016 Hanshaw Road
Ithaca, NY 14850

APPLICATION NOTE NO. 402

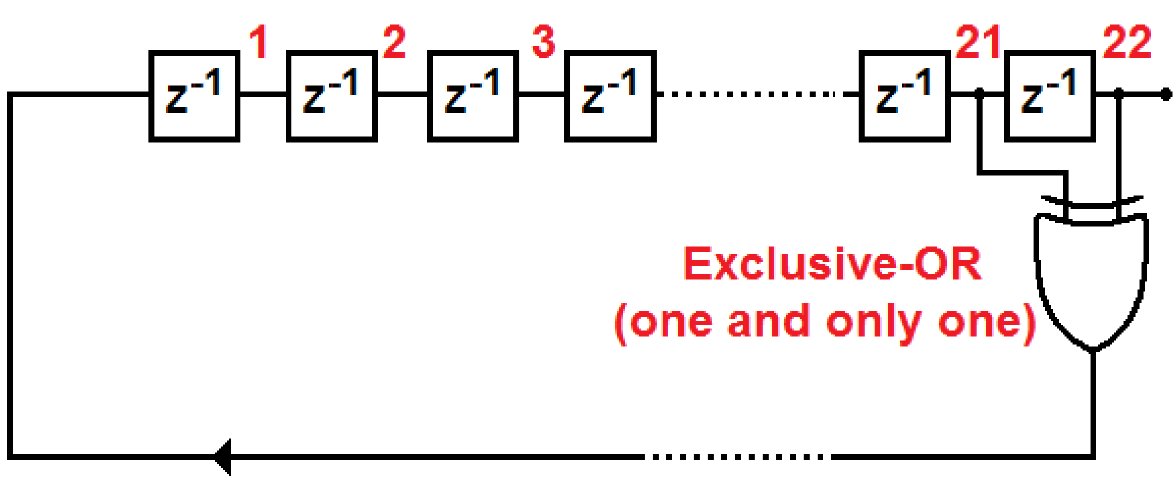
Jan 10, 2014

ANOMALIES IN PSEUDO-RANDOM GENERATORS

The use of Psuedo-Random Binary Sequence (PRBS) generators is widespread, and generally works well [1-6]. One primary use has been in the generation of sounds where a random signal ("noise") is the desired source material [1-3]. Some high-feature implementations using techniques such as adding up PRBS samples to achieve various amplitude distributions, and burst testing with repeatability, have been shown [4]. Reader here probably first saw these as offered by our highly-admired friend Don Lancaster [5,6].

These are highly suitable for sound generation purposes and for games, that sort of thing. But keep in mind the "pseudo" part of the name. Quite likely there are better choices for true "random number" generators for computer languages (which are still not random!). Here we want to examine some properties of the PRBS generators: things that we have noticed but not carefully studies before. Fig. 1 shows a typical generator (length N=22 stages) which repeats after 2^N-1 = 4,194,303 steps. [Note well the difference between the number of shift registers (N, relatively small) and the much larger length of the sequence (2^N-1)]. Fig. 1 is just a length 22 shift register with Exclusive-OR (modulo 2) feedback from the last two stages.

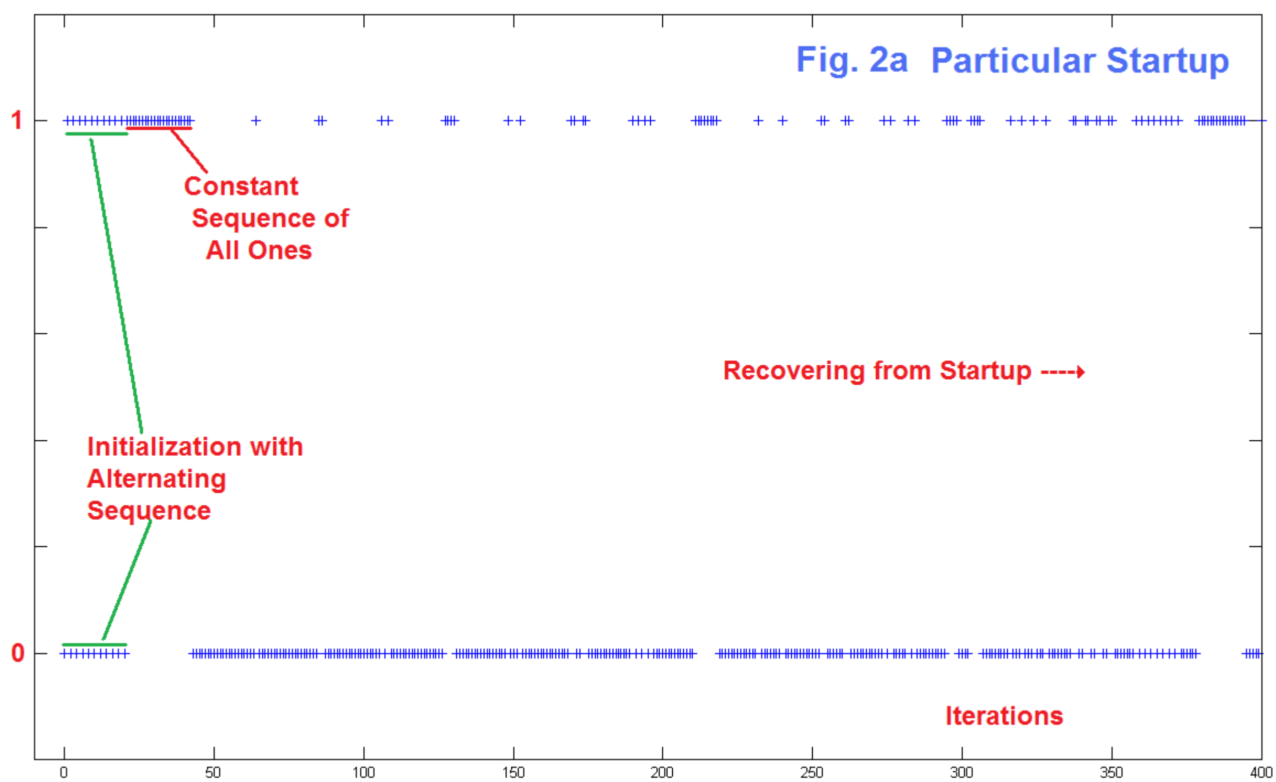
Fig. 1 Length 22 PRBS

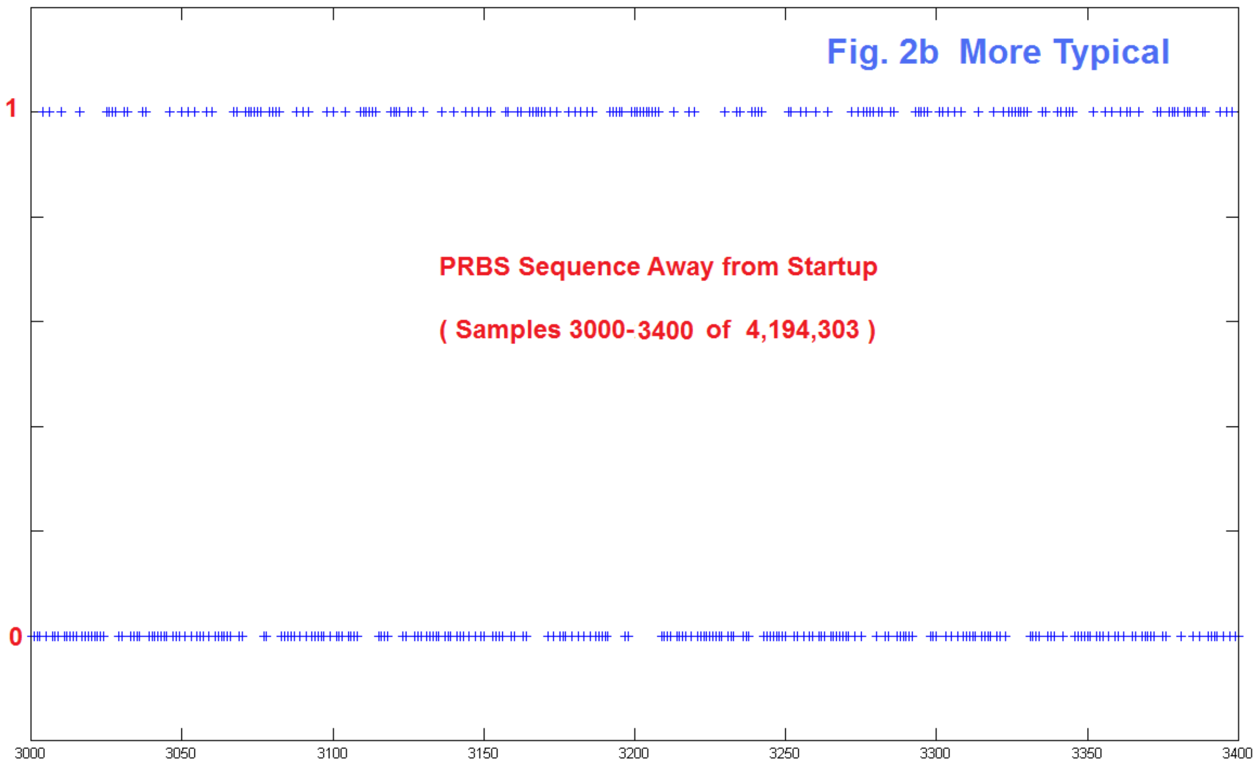


These configurations are not generated arbitrarily for any length, but are derived from some involved mathematics involving “primitive polynomials” as outlined ever so briefly in the ***Musical Engineer’s Handbook*** [3]. We could work endlessly with different lengths and selections of modulo-2 taps. Here we have consulted tabulated data for a “Maximal Length” generator. Yes – magic! The generator has length 2^N-1, one less than 2^N because the all zero state of the shift register is not allowed – by observation it stalls the generator at all zeros. The procedure here is that the delay line advances one step left to right; the modulo-2 of stages 21 and 22 (for this simple choice) is computed by the Exclusive-OR (high if there is one high and one low input, low if both are high or both low); and fed back to the outdated (because it was shifted up) stage 1. It just runs like this. Totally deterministic, but statistically it looks quite random in many aspects.

To be discussed here are some related issues. First, how do we select an initial state – other than avoiding the outlawed all-zero state – and does it matter. Second, what about the so-called “heartbeat” where the ear (brain as a pattern detector) somehow can recognize the existence of a repeat in millions of samples? Thirdly, are there any implications to the fact that all possible sequences (other than all-zero) are guaranteed by a maximal length PRBS?

Let’s begin with the heartbeat issue. When we listen to the output of the PRBS generator directly we expect to hear a crisp white noise, even though there are only two





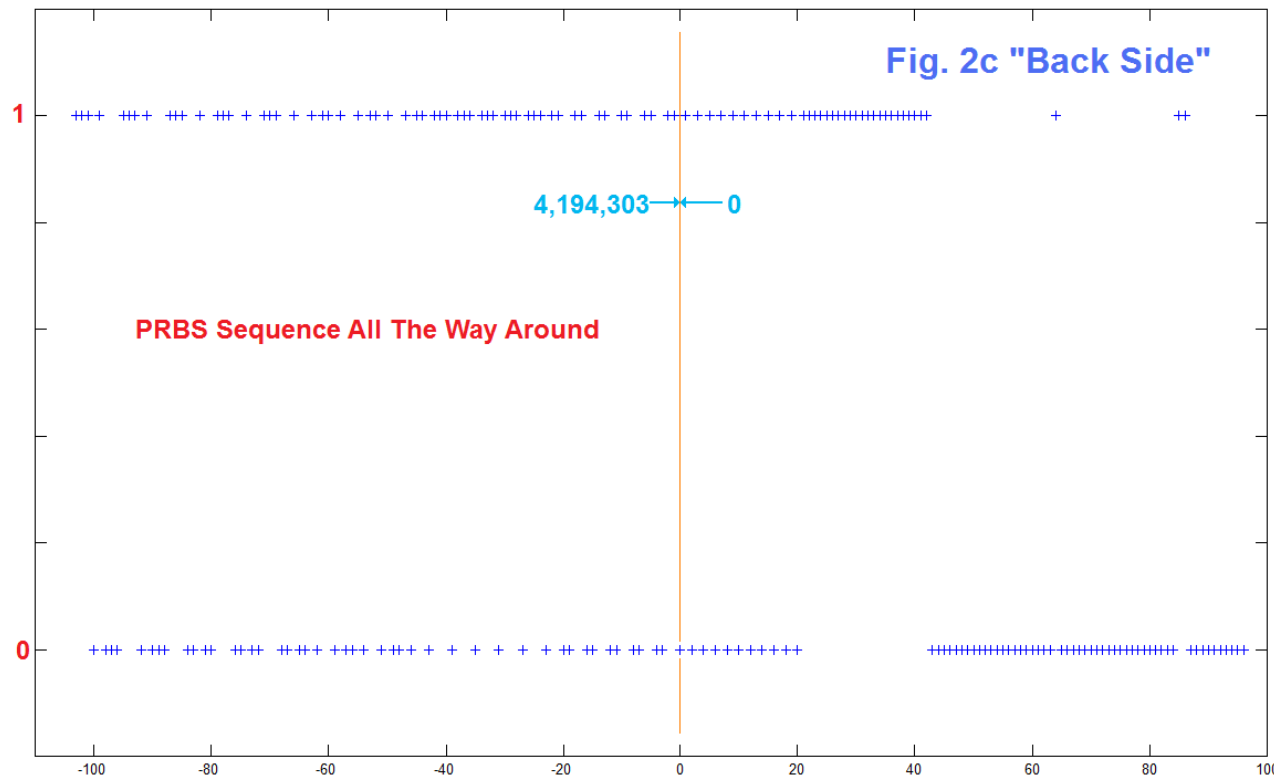
amplitude levels. For the most part, this is true. It is also true that the brain can be bored and start to hear various things in the noise. But in fact in the case of the PRBS noise, there can be a very discernible repeat pattern. We can become aware of the cycling of the sequence even after many many samples (perhaps millions). I thought this had been discussed a number of time, but I only located one [1]. This effect, which varies from a swish to a thump or a series of clinks and clanks, has been called a “heartbeat”. The speculation was that it was due to an untypical region or regions in the output, due to the fact that all possible sequences (some highly patterned) are guaranteed. That is, certain landmark features are embedded unavoidably. In fact, anomalies do exist, and these relate to the question of initialization (Fig. 2a).

We know that we must avoid the all-zero state. In consequence, we usually have used R-C delay to affirmatively load all-ones instead [2]. In working on this current study, it is convenient to use simulation and to initiate the simulation program in unique states. I wanted to examine what happened when the generator found itself in an alternating state, as I had speculated this was responsible for some sort of embedded audible feature that had to work its way out. In consequence, I initiated the simulation to the 22 states:

[1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]

while keeping in mind I also wanted to sometime look at the case of initializing to all 1’s.

Fig. 2a shows the result of starting in the alternating state. This we can compare with the situation much further along (Fig. 2b typical) where a more expected result is apparent. For one thing there are few longer runs – we must have some of course – and the balance of 1's and zeros is more even than in Fig. 2a. And, we see no obvious instances of extended periodicity. To my delight (it should NOT have been a surprise if I had thought an extra moment), directly following the alternating state is the state of all ones. So the simple structures tend to cluster – perhaps it is this that directly indicates the heartbeat of the sequence.



So why would we worry too much about the anomaly at the start if it goes away? Simply because it returns. In as much as we might be concerned with the return, we might be equally curious about how it set up. That is, what happened in Fig. 2a to the left of zero. We can probably "back up" the process and compute these samples. However, by simulating some four million plus iterations we show both the repetition and the "back side" as seen in Fig. 2c. This shows how the pattern is simplifying (groups of 3 and 1, groups of 2 and 2, etc) and is about to blunder into the anomaly. The sample 4,194,303 becomes sample 0 of the next full run.

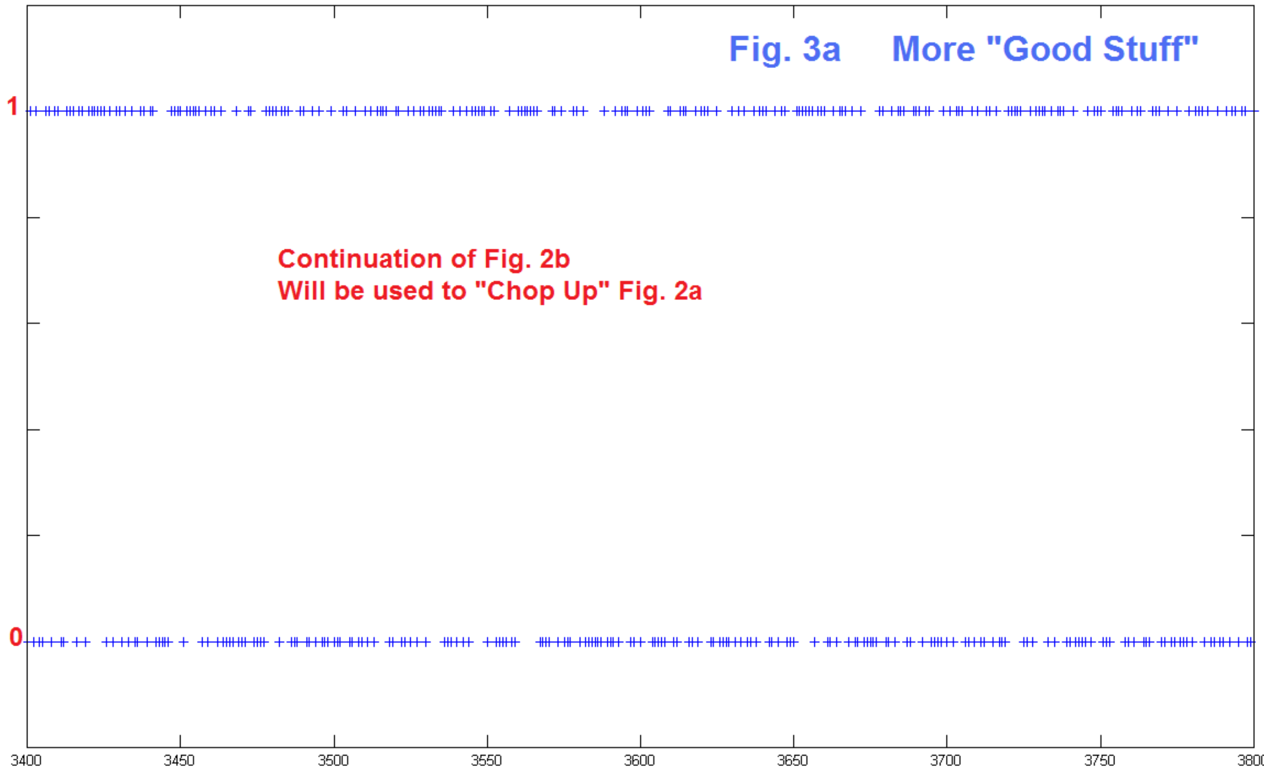


Fig. 3a More "Good Stuff"
1
0
Continuation of Fig. 2b
Will be used to "Chop Up" Fig. 2a
3400
3450
3500
3550
3600
3650
3700
3750
3800

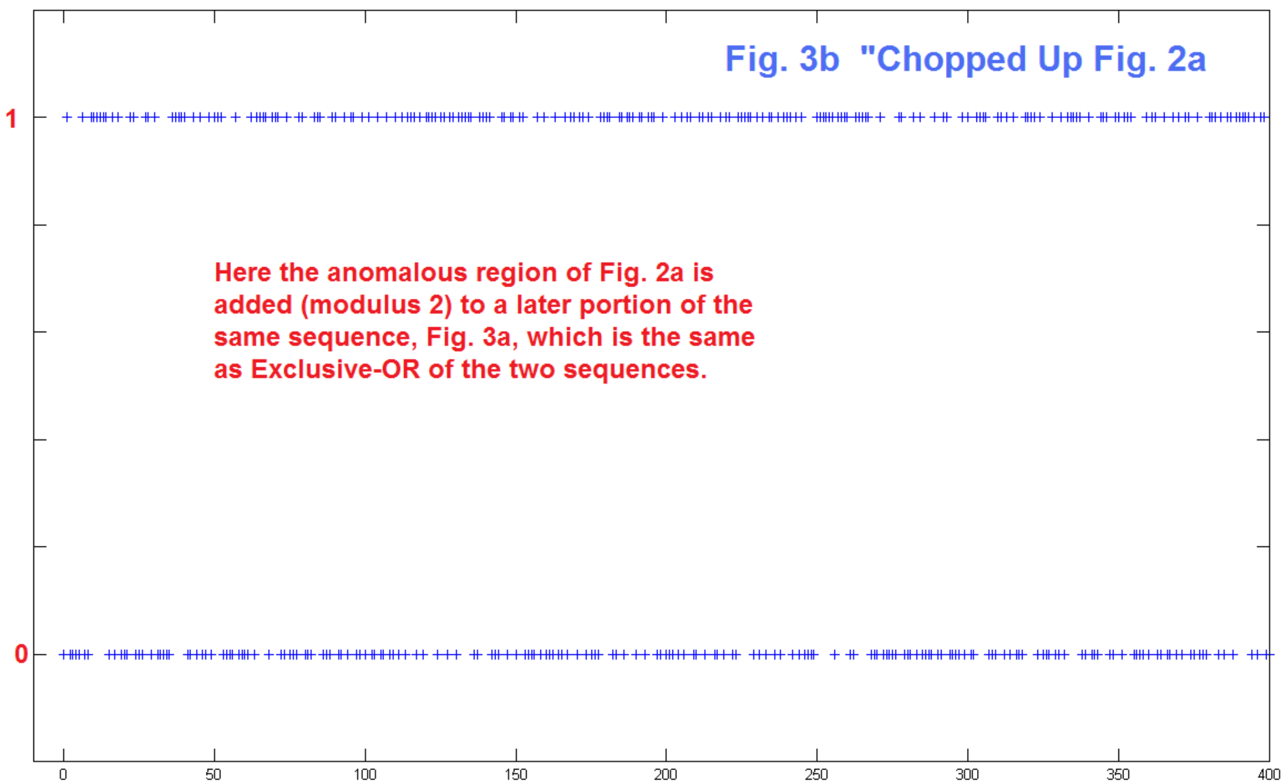


Fig. 3b "Chopped Up Fig. 2a
1
0
Here the anomalous region of Fig. 2a is
added (modulus 2) to a later portion of the
same sequence, Fig. 3a, which is the same
as Exclusive-OR of the two sequences.
0
50
100
150
200
250
300
350
400

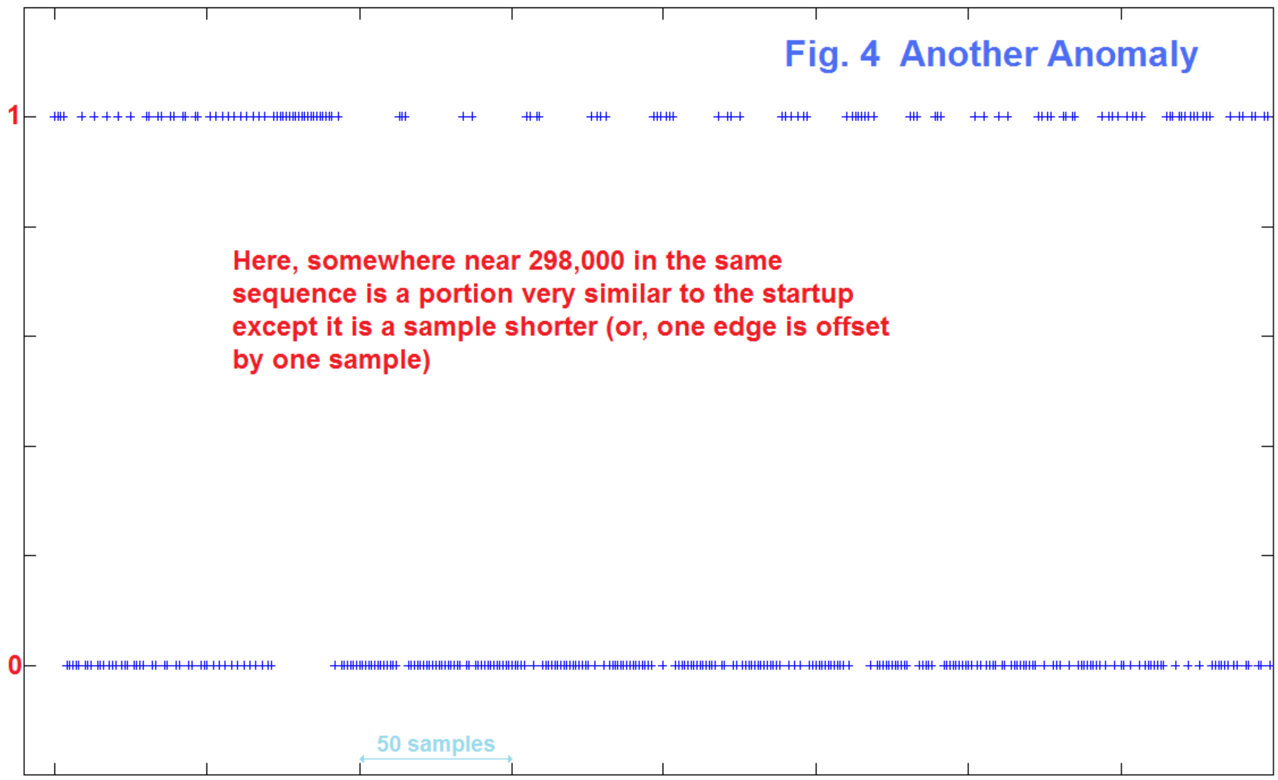
Given that what goes around comes around, we need an alternative strategy, and this seems apparent if we observe (as has been done before) that we can always break up a poor section by multiplying (Exclusive-OR) two different sequences. In Fig. 3a we show a portion of the sequence from samples 3400 to 3800 which appear "good". If we multiply these by the original start-up anomaly (Fig. 2a) we arrive at Fig. 3b, which looks good. So this just uses a later portion the same sequence, not as a substitute (how would one know where to make the substitution) but as scrambling. Clearly the start up problems in the delayed sequence would correspondingly be chopped up by the original.

In practice, there is probably no advantage to using the same sequence, as the construction of two separate side-by-side generators with different starts would seem to be required anyway. So why not use different length generators, which would seem to make the mod-2 combination have a length that is the product of the two sequence lengths: infinite for practical purposes.

One question that needs addressing is if there is just one, or are there perhaps many anomalies. It seems that there are several at least, not all of equal severity. Indeed, some degree of anomaly may be everywhere but usually manifests itself as too small to be even noticed. One reason I believe this is from the audio experiments, a description of one which I did locate (I recall others). Here is what it said in EN#64 [1], Page 10:

> *We will want to consider how the PRBS can be modified and processed, but first we should consider its direct use. If the PRBS is clocked along at 50 kHz to 100 kHz, the output is-a good "crisp" white noise, even though there are only two amplitude levels in the signal. This assumes that there are enough stages in the sequence so that the repetition rate is not directly audible as a pitch. At 100 kHz clock rate, this will be true for n = 13 or higher. For n = 20 or higher (at 100 kHz clock), the output sounds like random white noise. There are some fascinating cases between n = 13 and n = 20. The common setup for n = 15 is feedback from stages 14 and 15. This has a sequence length of 2^{15}-1 = 32,767. If this is clocked at about 32 kHz, it cycles approximately once per second. It is possible to hear this cycle; probably due to a "chink" and a "clugg" that occur during the sequence which seem to serve as markers. The same is true of a 17 stage sequence fed back from stages 14 and 17. This is a 2^{17}-1 = 131,071 stage sequence. If this is clocked at about 130 kHz (as in a commercial IC which is available), it cycles every second, and this cycling is just audible, although not as clearly as in the 15 stage case. Additional cases deserve study.*

Here I mention two "markers" that are audible, and I recall three event in another hearing. At present, what I have added is a search for other anomalies similar to that of Fig. 2a. With over 4 million samples in the sequence, this was not a matter of plotting and searching the plot! Instead segments of the sequence (length 400) were computed and examined for a number of zeros or of ones out of range (fewer than 135) of the expected 200. Curiously, over abundances of ones were not found! However, on about samples 298,000 (of 4,194,303) the sequence of Fig. 4 is found. It looks a lot like Fig. 2a, but of course it has to be different or the sequence would have been shorted back at that point. Indeed, instead of 22 one in a row there are 20. But the shuffling in and out is quite similar.



We are comfortable with software random number generators and may well agree that we could trust them as much as we might some sort of “real” noise such as the arrival of cosmic rays or the thermal motion of electrons in a resistor. PRBS generators should probably not be placed in a similar status. At least, we would be well advised not to use one to determine the winner of a multi-million lottery.

The output of the PRBS generator is a single bit, traditionally taken to be the rightmost shift register position (Fig. 1). It takes on only one of two values, 0 or 1, and it is not unreasonable to contend that given no idea about the generator, we don’t know if the next output will be a 1 or a 0**:** the probability is 0.5 for either case. We have methods of achieving multiple amplitude values, distributions, and correlation properties of random-like signals derived from a PRBS output [1, 4].

Another way to obtain a two-level random sequence would be to take a “real” analog random generator, sample it at equal time intervals, and quantize to two levels (like take the sign of the analog samples). In many ways, these would look like a PRBS sequence. So we can propose a game in which we have a sequence and we want to determine if it is most likely a PRBS or is derived from a random analog source.

Most obviously, we could look for periodicity. It would do little good to look for a repeat of a short sequence (perhaps [1 0 1 1 1 0 0]) as this would occur far too often. Choosing a test sequence of length 100 might be more persuasive. If this reoccurred after say a million iterations, we might be onto something. This would be very convincing if it reoccurred again after the same interval, and as different test sequences were found to have the same interval of periodicity. Such tests are not that hard to do: one simply latches a test sequence and runs the generator against the latched values with AND gates, counting as we go.

Another thing we could say is that if we find a long run of N 0’s, that the sequence was not generated by a PRBS generator of the length N of that run, or of a shorter length than N. Yet another thing to look for is that if we know we may have a certain length N PRBS generator, maximal length, that if we find the recurrence of any sequence of length N that occurs with spacing other than 2^N-1, we do not have a PRBS generator. That is, no length-N sequence can occur twice within the 2^N-1 length. For example, 2^{17}-1 is 131,071 so if you found a length 17 sequence of all 1’s occurring and then reoccurring before 131,071 samples (or any occurrence of 17 0’s) we know it is not PRBS.

Consideration of these and similar questions are entertaining and possible exam questions, but probably not too important, although they do indicate that a person understands what is going on, and some subtle distinctions.

By far, the most useful observation is that using two (or more) rather than just one PRBS, combining the results mod-2, should assure that artifacts and anomalies get well-hidden.

REFERENCES

[1] Hutchins, B, "Theory and Application of Noise Generators in Electronic Music," ***Electronotes,*** Vol. 8, No. 64, April 1976, pp 3-17.

[2] Hutchins, B., "The ENS-76 Home-Built Synthesizer System – Part 8, Random Sources" , ***Electronotes,*** Vol 9, No. 76, April 1977, pp 3-16

[3] Hutchins, B., ***Musical Engineer's Handbook***, Chapter 5h, Noise and Random Source Design

[4] Neuvo, Y. and W. Ku, "Analysis and Digital Realization of a Pseudorandom Gaussian and Impulsive Noise Source", ***IEEE Trans. on Communications***, Vol. 23, No. 9, Sept 1975, pp 849-858

[5] Lancaster , D., "Understanding pseudo-random circuits," ***Radio-Electronics*** April 1975 pg 42

[6] Lancaster, D., "Build the Psych-Tone: Melody Synthesizer with 28 Controls and 63-Note Melody", ***Popular Electronics***, Vol. 34, No. 2, Feb 1971, pg 25