Hideo Okawara's
Mixed Signal Lecture Series

DSP-Based Testing – Fundamentals 50
PRBS (Pseudo Random Binary Sequence)

ADVANTEST Corporation
June 2013

Preface to the Series

ADC and DAC are the most typical mixed signal devices. In mixed signal testing, analog stimulus signal is generated by an arbitrary waveform generator (AWG) which employs a D/A converter inside, and analog signal is measured by a digitizer or a sampler which employs an A/D converter inside. The stimulus signal is created with mathematical method, and the measured signal is processed with mathematical method, extracting various parameters. It is based on digital signal processing (DSP) so that our test methodologies are often called DSP-based testing.

Test/application engineers in the mixed signal field should have thorough knowledge about DSP-based testing. FFT (Fast Fourier Transform) is the most powerful tool here. This corner will deliver a series of fundamental knowledge of DSP-based testing, especially FFT and its related topics. It will help test/application engineers comprehend what the DSP-based testing is and assorted techniques.

Editor's Note

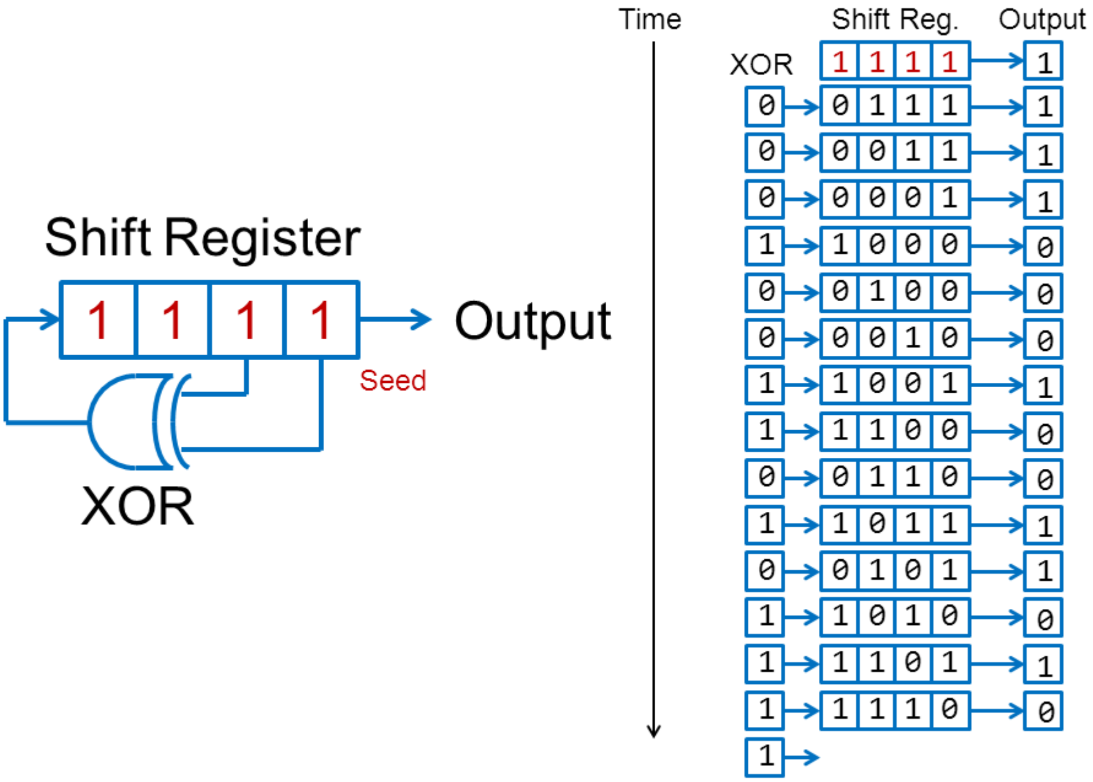
For other articles in this series, please visit the Advantest web site at
http://www1.verigy.com/ate/news/newsletter/index.htm

Preface

A pseudo random binary sequence (PRBS) is often used as a model data to test a high-speed serial interface devices for emulating a mission mode. It is mathematically randomized bit stream so that it is well neutralized and balanced data. It is employed not only in pure digital functional tests but also in mixed signal tests as an embedded data in a modulation signal. Even if your ATE is capable to generate PRBS data automatically, it is useful to understand the features of PRBS, so let’s look at how a PRBS data can be generated and its characteristics in this issue.

LFSR (Linear Feedback Shift Register)

A PRBS bit stream can be generated by using a linear feedback shift register (LFSR). Figure 1 illustrates an example of a 4-bit LFSR and its shifting data pattern. When the shift register is filled up with a seed pattern of all 1’s here, the table in the right hand side depicts how the register contents change and put out a series of PRBS. Right after the final bit, it returns to the top of the bit stream. There are 15 bits of pseudo random bit stream generated. An L-bit LFSR generates 2^L-1 bits of PRBS. With carefully looking at the bit pattern in the shift register, you can see there are all 4-bit combinations appeared except all 0’s. If you feed the pattern of “0000”, the shift register would be stuck and it generates only 0’s infinitely. A seed pattern must not be all 0’s. So one of the 15 4-bit patterns can be accepted as a seed.



XOR	Shift Reg.	Output
	1 1 1 1	1
0	0 1 1 1	1
0	0 0 1 1	1
0	0 0 0 1	1
1	1 0 0 0	0
0	0 1 0 0	0
0	0 0 1 0	0
1	1 0 0 1	1
1	1 1 0 0	0
0	0 1 1 0	0
1	1 0 1 1	1
0	0 1 0 1	1
1	1 0 1 0	0
1	1 1 0 1	1
1	1 1 1 0	0
1		

Figure 1: Linear Feedback Shift Register (LFSR)

PRBS Waveform, Bits and Edges

Figure 2 illustrates how the 15-bit PRBS pattern looks like as a waveform. There are 8 edges toggling in the 15 bits NRZ pattern, containing eight 1's and seven 0's. This is a general feature of a PRBS; there are $2^{(L-1)}$ bits of 1's and $(2^{(L-1)}-1)$ bits of 0's in a $(2^{L}-1)$ bits of PRBS. These are summarized in Table 1, which shows examples of PRBS4 and PRBS7.

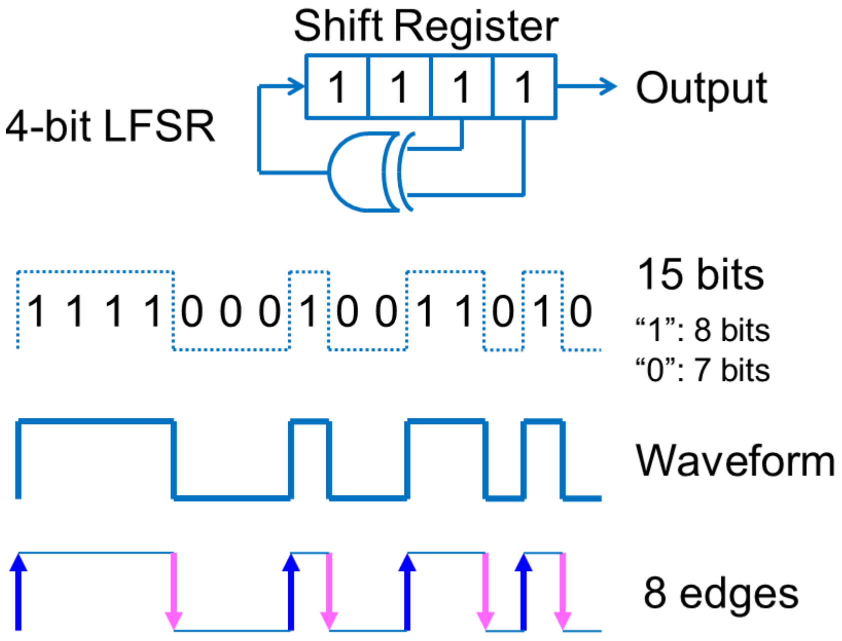
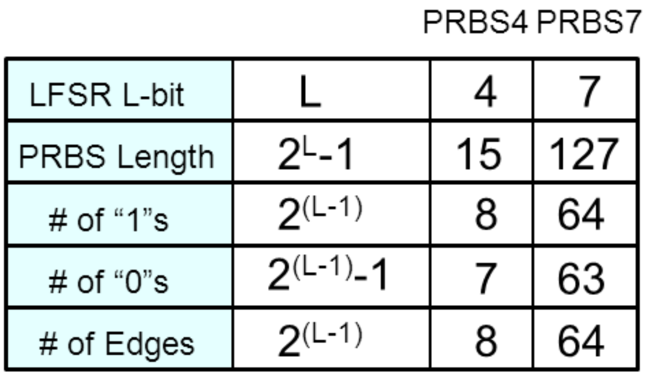


Figure 2: Waveform of PRBS by 4-bit LFSR

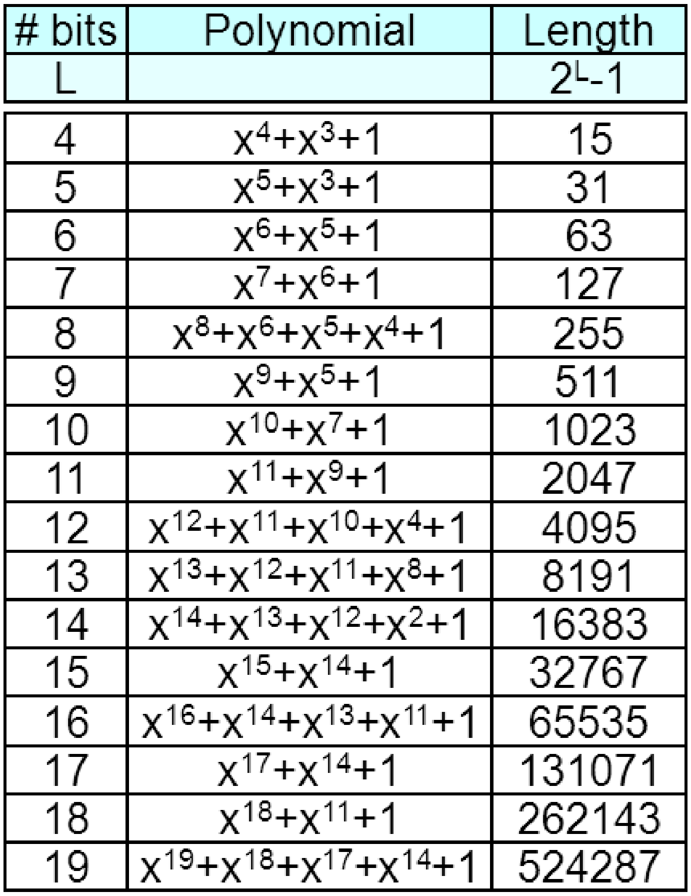


		PRBS4	PRBS7
LFSR L-bit	L	4	7
PRBS Length	$2^{L}-1$	15	127
# of "1"s	$2^{(L-1)}$	8	64
# of "0"s	$2^{(L-1)}-1$	7	63
# of Edges	$2^{(L-1)}$	8	64

Table 1: Features of LFSR and PRBS

LFSR Polynomial

The input of the XOR (exclusive OR) in the 4-bit LFSR is the 3rd and 4th cells of the LFSR. This LFSR is mathematically described as a polynomial of x^4+x^3+1. Table 2 shows more LFSRs longer than 4 bits, so they can generate PRBS patterns longer than 15 bits.



# bits	Polynomial	Length
L		2^L-1
4	x^4+x^3+1	15
5	x^5+x^3+1	31
6	x^6+x^5+1	63
7	x^7+x^6+1	127
8	$x^8+x^6+x^5+x^4+1$	255
9	x^9+x^5+1	511
10	$x^{10}+x^7+1$	1023
11	$x^{11}+x^9+1$	2047
12	$x^{12}+x^{11}+x^{10}+x^4+1$	4095
13	$x^{13}+x^{12}+x^{11}+x^8+1$	8191
14	$x^{14}+x^{13}+x^{12}+x^2+1$	16383
15	$x^{15}+x^{14}+1$	32767
16	$x^{16}+x^{14}+x^{13}+x^{11}+1$	65535
17	$x^{17}+x^{14}+1$	131071
18	$x^{18}+x^{11}+1$	262143
19	$x^{19}+x^{18}+x^{17}+x^{14}+1$	524287

http://en.wikipedia.org/wiki/Linear_feedback_shift_register

Table 2: **4 to 19-bit LFSR Polynomials**

When you use a PRBS pattern in a test which is required to be repetitive with 2^L bits for some reason, the natural PRBS has 2^L-1 bits so that it is not convenient for the test requirement. Then you may want to put additional 0 to the end of the PRBS pattern. (Table 3) When you fill up 16-bit buffer memory with the 15-bit PRBS patten, the final cell becomes empty so that this would be incovenient in some situations. Then you may want to repeat 15 iterations of the PRBS pattern continuously in order to fill up the 16-bit buffer. (Table 4)

1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	0

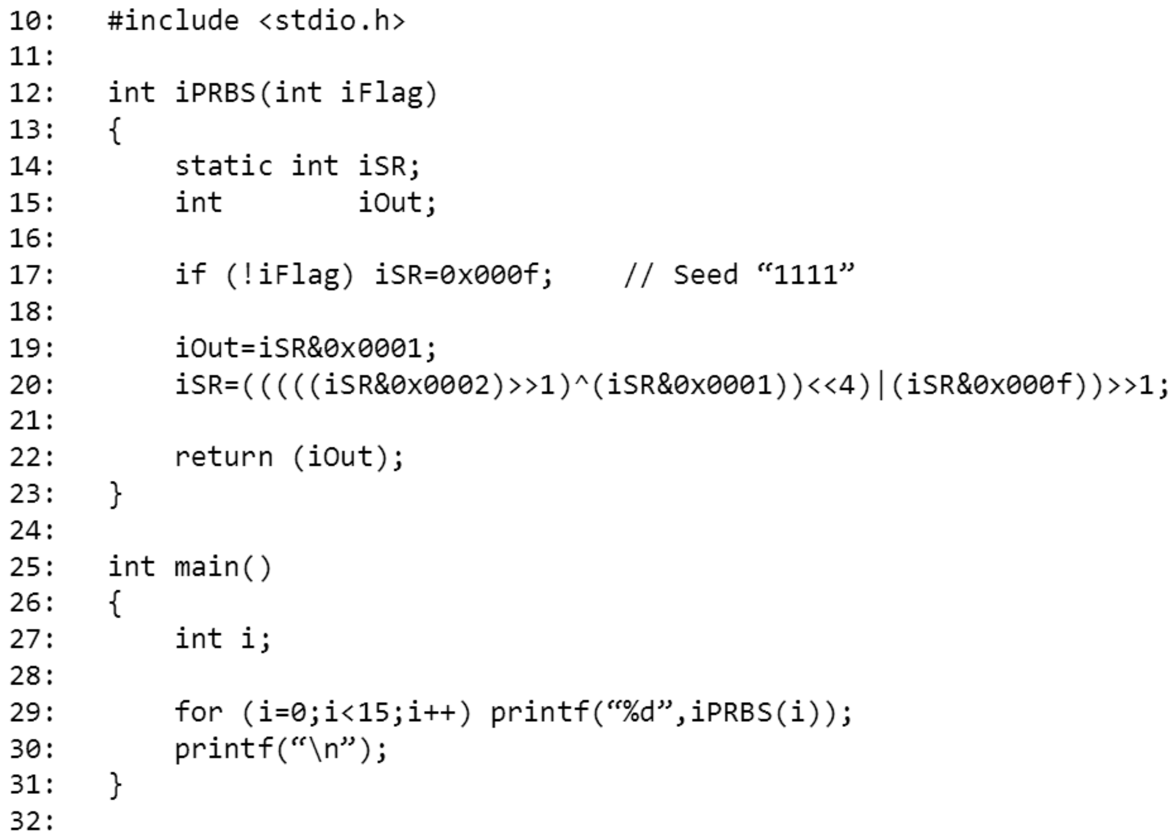
Table 3: Padding “0”

0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1
1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1
2	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1
3	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
4	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0
5	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0
6	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
7	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1
8	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	0
9	0	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0
10	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1
11	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1	1
12	0	1	0	1	1	1	1	0	0	0	1	0	0	1	1	0
13	1	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1
14	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0

Table 4: 15 Iterations of 15-bit PRBS

PRBS Generation Program

The bit stream of PRBS4 (4-bit LFSR) shown in Figure 2 can be generated by executing the “C” program in List 1. When the for-loop begins with i=0, the register “iSR” is initialized with the seed pattern “1111” or 0x000f in this example.



```c
#include <stdio.h>

int iPRBS(int iFlag)
{
    static int iSR;
    int        iOut;

    if (!iFlag) iSR=0x000f;    // Seed “1111”

    iOut=iSR&0x0001;
    iSR=(((((iSR&0x0002)>>1)^(iSR&0x0001))<<4)|(iSR&0x000f))>>1;

    return (iOut);
}

int main()
{
    int i;

    for (i=0;i<15;i++) printf(“%d”,iPRBS(i));
    printf(“\n”);
}
```

List 1: **Source Code of PRBS Generation by 4-bit LFSR**

PRBS Spectrum and Bandwidth

When the bit pattern of PRBS4 is running at the rate of 1 Gbps, its waveform would be the top picture in Figure 3. By applying the FFT (Fast Fourier Transform) to the 1 Gbps PRBS waveform, you can see the frequency spectrum of the PRBS waveform, which is shown in the bottom picture in Figure 3.

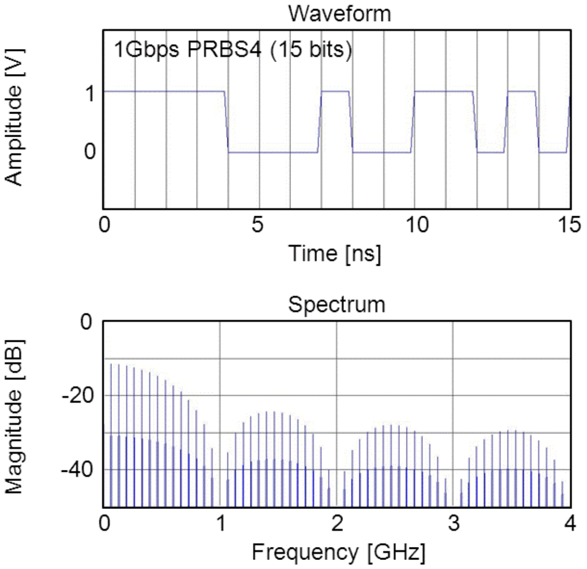
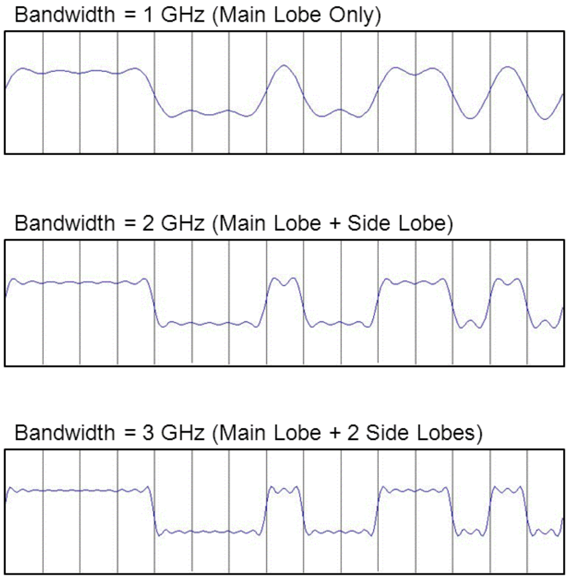


Figure 3: Spectrum of 15-bit PRBS



Bandwidth = 1 GHz (Main Lobe Only)

Bandwidth = 2 GHz (Main Lobe + Side Lobe)

Bandwidth = 3 GHz (Main Lobe + 2 Side Lobes)

Figure 4: Waveforms vs. Bandwidth

You may notice some specific features in the spectrum; the main lobe of 1 Gbps bit stream occupies 1 GHz which corresponds the bit speed. The spectrum is constructed with line spectral components. The main lobe contains 15 lines of components including DC location. So the spectrum spacing becomes 66.666… MHz (=1 GHz/15). Each one of the side lobes is constructed with 15 spectrum spacing as well. The line spectrum spacing equals the bit rate/(2^L-1).

If you limit the bandwidth of the spectrum and reconstruct the waveform by applying the IFFT (Inverse FFT), the waveform looks as illustrated in Figure 4. Considering the signal integrity of the waveform, a good digital signal should contain at least 3 lobes (the main + 2 side lobes) in general. So an analog bandwidth of signal path should be greater than 3 times of the bit rate frequency. This is a rule of thumb for designing a load board for digital signal. This is greatly important in testing high-speed digital interface signals. An ATE resource to test such signals should be required to have more than 3 times of the bit rate as well.

TMU Consideration

The PS9G and PS1600 digital channels in the SOC Test System V93000 integrate the TMU (Time Measurement Unit), which is an excellent resource to measure time events. When testing jitters of a PRBS signal by using the TMU, it is useful for you to understand a feature of PRBS. For detailed information about the TMU and its programming, you should consult the specific system manual and documents.

There are some programmable parameters available in the TMU. At first, you should set the edge parameter "RISE_FALL" for capturing a PRBS signal. Then "Prescaler" and "InterSampleDiscard" are important for capturing high frequency waveform or high-speed digital signal. For simplicity, "Prescaler" is not discussed here. For detail, refer the appropriate document.

Now "InterSampleDiscard" is a point in this article. It should be set as an even number. Then let's look at how the 15-bit PRBS4 waveform would be treated in Figures 5 and 6. When setting "RISE_FALL", rising edges and falling edges are alternately captured, marked red in the figures. Looking at the time intervals of every rising and falling edges, there are 8 unique patterns found in this case. This number "8" correponds to the number of edges of the PRBS waveform. When PRBS4, there are 8 edges in the waveform so that there are 8 unique patterns available. As seen in Figures 5 and 6, when "InterSampleDiscard" is different, the unique pattern waveforms are different but the number of unique patterns is always constant. This unique pattern concept would influence the data dependent jitter (DDJ) in the actual jitter measurement.

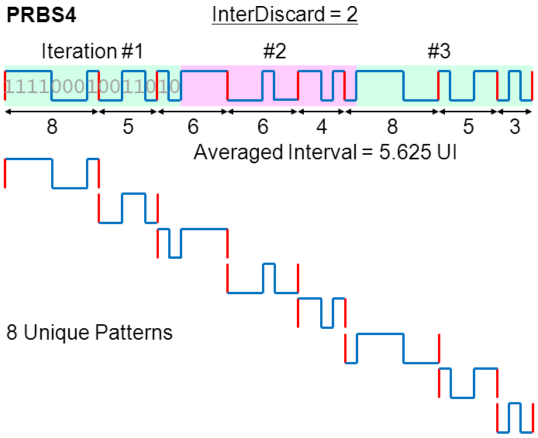


Figure 5: InterSampleDiscard=2

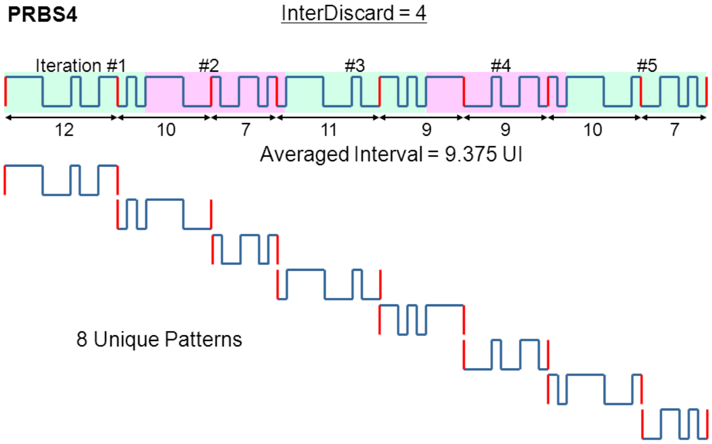


Figure 6: InterSampleDiscard=4