IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

Comparative Analysis of Pseudo-Random Binary Sequence Patterns for the High-Speed Digital Systems

Jaechan Lee

Liberal Arts and Science Academy, 1012 Arthur Stiles Road, Austin, TX, 78721, USA; jaechanlee0924@gmail.com

ABSTRACT: Pseudo-random binary sequences (PRBS) are widely used in high-speed digital systems to evaluate signal integrity and transmission reliability. However, the statistical characteristics of different PRBS degrees have not been as comprehensively analyzed as those of other specialized random sequences. In this study, we investigated the characteristics of PRBS7 through PRBS31, which is commonly used in practice, using a Python-based model. The analysis covered randomness properties, including run-length distribution, frequency balance, entropy, and power spectral density (PSD), as well as how these characteristics vary with noise intensity. The results provide objective criteria for selecting an appropriate PRBS generator, taking into account both statistical quality and implementation cost.

KEYWORDS: Engineering, Digital Electronics, Digital Logic System, Pseudo Random Binary Sequence.

Introduction

With the rapid advancement of technology, the importance of high-speed serial interfaces such as Universal Serial Bus (USB), Peripheral Component Interconnect Express (PCIe), and Ethernet in large-scale data transmission continues to grow. In particular, recent trends in artificial intelligence have explosively accelerated the demand for high-bandwidth input/output (I/O). However, channel insertion loss and noise through cables or printed circuit boards (PCBs) make it difficult to scale the bandwidth of high-speed interfaces and significantly contribute to an increase in bit error rate (BER).[1] Although semiconductor and circuit design technologies have substantially improved the operating speed of integrated circuits, the analysis and verification of such high-speed circuits have become increasingly challenging.

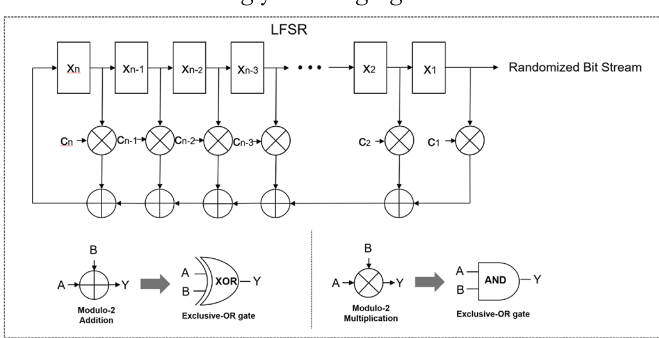

Figure 1: Run length distribution of the PRBS polynomials. The figure illustrates how the feedback shift registers defined by each primitive polynomial generate deterministic pseudo-random bit sequences through modulo-2 addition. Differences in the polynomial coefficients result in distinct sequence characteristics, which are reflected in the observed run-length distributions.

For efficient validation, it is desirable to test circuits at the same speed as their actual operating frequency. PRBS provides a cost- and time-effective solution since it can be integrated directly into digital circuits without the need for expensive test equipment, thereby enabling at-speed testing.[2] Consequently, PRBS-based BER measurement has become a fundamental requirement for verifying the data transmission quality of various high-speed interfaces. Although PRBS appears to be a random sequence, it is in fact a deterministic bit stream generated by a Linear Feedback Shift Register (LFSR), where the previous state of the register determines each output bit.[3] A sequence is defined by a register length N and a corresponding primitive polynomial. For example, when the highest degree of the primitive polynomial is 7, the sequence is denoted as PRBS7, whereas a 15th-degree polynomial defines PRBS15. Figure 1 illustrates an N-degree PRBS generator based on the polynomial $C_n x^n + C_{n-1} x^{n-1} + \cdots + C_2 x^2 + C_1 x^1 + 1$.

LFSR consists of N linearly connected registers X_n and a coefficient set C_n. The feedback function computes the new state based on the initial values of the shift register using modulo-2 addition and multiplication.[4,5] In this operation, addition can be replaced with an exclusive OR (XOR) gate, and the multiplication can be replaced with an AND gate. As an example, the operation of PRBS7 can be explained using the primitive polynomial $x^7 + x^6 + 1$. The corresponding coefficient vector is

$$(C1, C2, C3, C4, C5, C6, C7) = (0, 0, 0, 0, 0, 1, 1)$$

Assume the initial register state is

$$(x1, x2, x3, x4, x5, x6, x7) = (0, 1, 0, 1, 1, 0, 1)$$

The new feedback bit x0 is calculated as a modulo-2 linear combination of the tapped registers

$$x_0 = (C_6 \cdot x_6) \oplus (C_6 \cdot x_7)$$

Substituting the value $x_6 = 0$ and $x_7 = 1$, we obtain

$$x_0 = 0 \oplus 1 = 1$$

This feedback bit is inserted into the first register, while all other register values are shifted to the right. The updated state becomes

$$(x1, x2, x3, x4, x5, x6, x7) = (1, 0, 1, 0, 1, 1, 0)$$

By repeating this process, the PRBS7 generator produces a deterministic bit sequence that appears random but has a maximal length of $2^7-1=127$ before repeating. Using this principle, PRBS patterns of different degrees can be generated. Although several studies have investigated pseudo-random number generators,[6-9] most have focused on specialized generators used in applications such as cryptography, and a comprehensive understanding of the statistical and structural properties of PRBS of various degrees used in high-speed systems is still not fully understood in practice. To address this gap, the main objective of this study is to systematically compare and analyze the properties of PRBS sequences of different polynomial degrees employed in high-speed interface designs. Specifically, we implement various PRBS patterns using a Python model and evaluate them against widely used pseudo-random sequence generators. The analysis includes statistical randomness properties, sequence performance, and hardware implementation cost. Through this investigation, we aim to provide useful criteria for selecting the most suitable generator depending on specific design requirements.

■ Methods

PRBS Polynomials:

In this study, we reviewed several primitive polynomials with PRBS degrees ranging from 7 to 31 in relevant literatures10-12 and selected 9 PRBS polynomials representing short, medium, and long sequences as shown in Table 1. This selection covers a wide range of sequence lengths and reflects the PRBS patterns widely used in digital systems. A Python model was designed for statistical analysis. Each PRBS polynomial was implemented in Python, and the formulas required for the respective statistical analyses were simulated accordingly.

Table 1: Different degrees of PRBS polynomials. This table shows a list of commonly used PRBS patterns, their corresponding primitive polynomials, and degrees N. Using the PRBS patterns defined in the table, various statistical tests are performed to compare their characteristics.

PRBS Pattern	PRBS Polynomial	Degree (N)
PRBS7	X^7+X^6+1	7
PRBS9	X^9+X^5+1	9
PRBS11	$X^{11}+X^9+1$	11
PRBS13	$X^{13}+X^7+X^3+X^2+1$	13
PRBS15	$X^{15}+X^{14}+1$	15
PRBS19	$X^{19}+X^5+X^2+X+1$	23
PRBS23	$X^{23}+X^{18}+1$	23
PRBS27	$X^{27}+X^5+X^2+X+1$	40
PRBS31	$X^{31}+X^{28}+1$	31

Statical Analysis:

Run Length Distribution:

All selected polynomials have been proven in previous research to be irreducible primitive polynomials.[3-5] This means that the maximum length of the selected polynomial of degree N is 2^n - 1 bits for a given degree N. Run tests were performed to analyze and compare the statistical randomness and maximum run length properties of each generated PRBS sequence.[13] For a PRBS of degree N, the expected run count for run length L is given by:

$$Run\ Count\ =\ \frac{2^{N+1}}{2^{L+1}}\ \ (1\leq L\leq N-1)$$

Figure 2 shows the run length distributions for each polynomial. The tests were conducted over two full periods of each PRBS sequence. As observed, the run count decreases exponentially with run length, corresponding to a near-linear decline on a logarithmic scale, which indirectly confirms the primitivity of the polynomials.

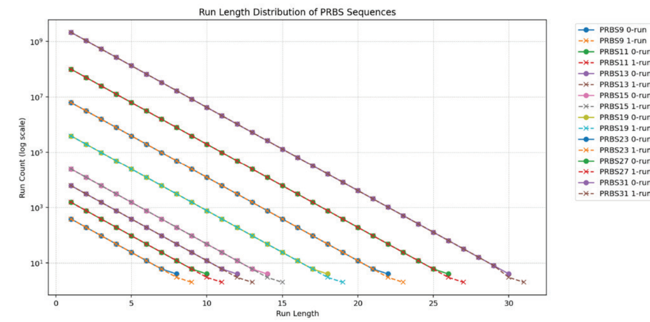

Figure 2: Run length distribution of the PRBS polynomials. This figure compares the run length statistics of different PRBS orders, showing that higher-degree sequences exhibit longer maximum runs and a wider distribution range.

All sequences maintained a near-perfect balance between zeros and ones, and the maximum run lengths matched theoretical predictions. The slight decrease in the number of 1-run near the maximum run length occurs because 1-run reaches the same length as the polynomial's degree when sampling runs, while relatively reducing the number of 1-run of degree N-1. Overall, the simulated results satisfy theoretical expectations, and it can be confirmed that all nine polynomials can generate PRBS sequences of maximum length in practical implementations.

Frequency Test

The frequency test is used to verify whether a PRBS exhibits randomness and whether 0 and 1 occur with equal probability in each bit sequence.[13] S_n, indicating a bias toward bit 1 or 0, can be measured by summing over the entire binary sequence B_k and is expressed as:

$$S_n=\sum_{k=1}^{n}B_k$$

For example, a non-zero positive value for S_n means that 1s are observed more often than 0s in a binary sequence. With S_n, the randomness of a binary sequence can be verified using a probabilistic concept called the *p-value*. This represents the probability that the currently observed result will occur, assuming that the null hypothesis is true.[13] Therefore, the closer the *p-value* is to 1, the more likely it is that the observed data would be observed very commonly under the null hypothesis, indicating a higher probability that the result occurred by chance. P-value is given by:

$$p\_value\ =erfc\left(\frac{|S_n|}{\sqrt{2N}}\right)$$

where N is the maximum degree of the given polynomial, the *erfc* function is a complementary error function, which calculates the probability that a value from a standard normal distribution will exceed a given threshold.[13] The Python math library was used to calculate the complementary error function. Table 2 shows the p-values obtained under two different conditions. First, all PRBSs were simulated with the same number of bits, and second, each PRBS was simulated with a full cycle.

Table 2: Frequency Test P-values for the PRBS Polynomials. This table highlights that insufficient test length can lead to misleadingly low p-values for long PRBS sequences. On the other hand, when performing a Frequency Test using one period, a very distinct pattern appears in which the p-value gradually increases as the PRBS order increases. This indicates that the random characteristics converge to the ideal quickly as the PRBS order increases.

Test Period	PRBS7	PRBS9	PRBS11	PRBS13	PRBS15	PRBS19	PRBS23	PRBS27	PRBS31
10M bits	0.0	0.0	0.0	0.0	6.81e-41	0.909834	0.839661	0.9181876	0.6233868
One period (2^N-1)	0.929291	0.964715	0.982366	0.991184	0.995592	0.998898	0.999724	0.999931	0.999992

As mentioned earlier, a p-value closer to 1 indicates higher randomness. For lower degree PRBSs, the same pattern repeats across 10 million bits, resulting in a deterministic pattern, which increases statistical deviation and results in a p-value of 0. This demonstrates that PRBSs are not completely random but rather closer to deterministic patterns. In contrast, when a single cycle is simulated without repetition, the p-values are significantly higher because no recurring patterns are present since higher degree PRBS can generate better randomized patterns, so it shows a better p-value than lower degree PRBS.

Entropy Analysis:

To assess the uncertainty of a binary sequence, we analyzed entropy. Entropy can measure the uncertainty of a random binary sequence.[14] Shannon entropy (E) is widely used for the binary sequence randomness and is given by:

$$E = -\left(p_0 \, X \, log_2(p_0)\right) - \left(p_1 \, X \, log_2(p_1)\right)$$

where p0 and p1 are the ratios of 0 and 1 in the sequence, to calculate Shannon entropy, we first calculate the Shannon information for each possibility of each bit, p0 and p1, on a logarithmic scale.[14] Then, the entropy can be obtained by adding all the information for each bit as shown above. The closer the 0s and 1s occur with equal probability, the closer the entropy value is to 1. Shannon entropy measures the overall balance of the binary bit distribution but does not account for the sequence's periodicity or pattern repetitions.

Table 3: Shannon Entropy for the PRBS Polynomials. This table shows that the Shannon entropy of all PRBS sequences is extremely close to 1, and it increases with the PRBS order as longer periods provide more statistically balanced bit distributions.

PRBS	Entropy (≤ 1.0)
PRBS7	**0.9999**552765354263
PRBS9	**0.99999**72375155557
PRBS11	**0.999999**8278491158
PRBS13	**0.9999999**892458553
PRBS15	**0.99999999**23594146
PRBS19	**0.99999999**85234149
PRBS23	**0.9999999999**752293
PRBS27	**0.9999999999**78499
PRBS31	**0.9999999999991**342

Table 3 shows the simulated entropy value for all PRBS sequences. Lower degree PRBS has shorter periods, which can cause statistical bias and slightly lower entropy. Higher degree PRBS maintain almost perfect balance within a single period, producing entropy values very close to the theoretical maximum.

Autocorrelation and PSD Analysis:

Autocorrelation measures how much the sequence is correlated by comparing it to itself at different lags.[15,16] For PRBS sequences, the autocorrelation is very low where lag is not zero and shows a single sharp peak occurs at lag = 0, indicating that the sequence behaves as a random signal without correlation across different time instances, thereby showing good randomness. Figure 3 shows the results of simulating four periods of PRBS7, PRBS11, PRBS15, and PRBS23, with the autocorrelation values computed up to lag 500 and visualized as plots. Here, the purpose was not numerical comparison but to provide a simple graphical comparison of the autocorrelation characteristics among PRBS sequences. The normalized correlation values are 1 at lag = 0, and the y-axis was scaled to visualize non-zero lag regions better.

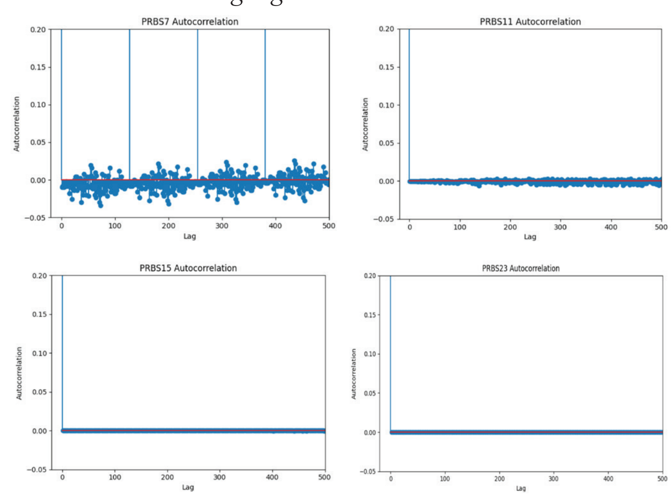

Figure 3: Autocorrelation plots for various PRBS orders. As the PRBS order increases, the autocorrelation rapidly flattens for all non-zero lags, resulting in an ideal response with a single peak at lag 0. This indicates improved randomness and reduced periodic structure in higher-order sequences.

In Figure 3, PRBS7, having a short period of 127 bits, exhibits periodic patterns when the same number of samples is used. In contrast, PRBS23 has a much longer period and shows much lower correlation and behaves more randomly. Using the same sample length, shorter-period PRBS7 demonstrates periodic autocorrelation, while longer-period PRBS sequences appear more random, clearly illustrating that higher-degree PRBS provides a higher level of randomness.

To further analyze the statistical properties, the Power Spectral Density (PSD) of PRBSs was also simulated. The PSD represents the power distribution of a binary sequence, so truly random binary sequences should distribute power uniformly across all frequency bands, similar to white noise. However, PRBSs are not perfectly random, and deviations from uniformity occur. The following three metrics were used to analyze how closely the PSD of the PRBS resembles white noise.

DOI: 10.36838/IJHSR810.37

PSD Average:

The PSD average is defined as the mean PSD value across all frequency bins. For a PRBS sequence where 0s and 1s appear randomly, converting 0 to -1 should yield a PSD average approximately equal to 1. The PSD was computed using Python's Welch method.[17] Denoting the normalized PSD obtained from the Welch function as $X(n)$, the PSD average can be defined as follows, where N is the length of the measured sequence.

$$PSD\_avg \ = \frac{1}{N}\sum_{r=0}^{n-1}|X(n)|^2$$

Table 4 shows that the PSD average for all PRBS sequences is 1. It indicates that the mean power is constant at 1 when binary values are mapped to ±1.

Variance of PSD:

The variance of the PSD quantifies the fluctuations around the mean. If the PSD average is 1 and the normalized PSD values are $X(n)$, the variance can be defined using the following equation.

$$Variance \ = \frac{1}{N}\sum_{r=0}^{n-1}(X(n) - PSD\_avg)^2 = \frac{1}{N}\sum_{r=0}^{n-1}(X(n) - 1)^2$$

A higher variance indicates greater spectral fluctuations and greater variability. As shown in Table 4, lower degree PRBS sequences with shorter periods have higher variance because the power is unevenly distributed across frequency. In contrast, higher-degree PRBS sequences with longer periods, which have characteristics closer to white noise, result in lower variance.

Flatness:

Flatness measures how uniform the PSD distribution is. Defining the minimum and maximum PSD values as PSD_{min} and PSD_{max}, the flatness is calculated on a dB scale as follows.

$$Flatness\ (dB) \ = \ 10 \times log_{10}(\frac{PSD_{min}}{PSD_{max}})$$

Table 4: PSD average, Variance, and Flatness for the PRBSs. This table clearly shows that as the PRBS order increases, the variance of the PSD decreases monotonically, resulting in significantly improved spectral flatness. PRBS31 exhibits the lowest PSD variance and the best flatness, indicating that longer PRBS sequences approximate white-noise behavior more closely.

PRBS	PSD Avg	Variance	Flatness (dB)
PRBS7	1.000	1.569e-02	-13.630
PRBS9	1.000	3.910e-03	-14.232
PRBS11	1.000	9.768e-04	-14.838
PRBS13	1.000	2.442e-04	-15.436
PRBS15	1.000	6.104e-05	-16.038
PRBS19	1.000	3.815e-06	-17.242
PRBS23	1.000	2.384e-07	-18.447
PRBS27	1.000	1.490e-08	-19.651
PRBS31	1.000	5.001e-09	-20.125

If the power is similar across all frequency bands, flatness approaches 0 dB; otherwise, it takes a negative value. Table 4 demonstrates that flatness improves with increasing PRBS degree, indicating that PRBS31 approximates ideal white noise more closely than PRBS7.

BER and Jitter Analysis under AWGN:

In this study, a simple numerical model was employed to emulate the signal transmission environment, and artificial interference was applied to PRBS signals. The original and interfered signals were then compared to measure and analyze the AWGN signal-to-noise ratio (SNR) and bit error rate (BER).[18]

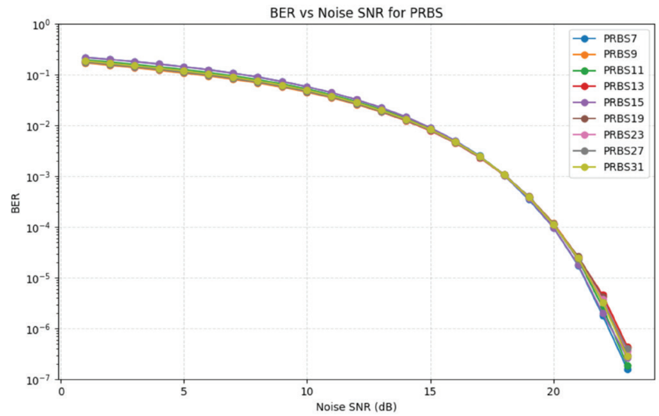

Figure 4: BER under AWGN SNR. This figure shows the BER performance of various PRBS sequences under AWGN conditions. All PRBS lengths exhibit nearly identical BER curves, indicating that sequence length does not significantly affect. Small deviations appear at high SNR regions, but the overall results confirm that BER performance is dominated by channel noise rather than PRBS length.

As shown in Figure 4, although the BER degraded as AWGN noise increases, all PRBSs show almost identical BER performance regardless of their degree or pattern length. Prior to the simulation, it was anticipated that sequences with minimized pattern repetition, such as PRBS31, would distribute the noise impact more evenly. However, higher degree PRBS sequences did not show any noticeable difference in BER characteristics. These results confirm that BER performance is largely dependent on the structure and characteristics of the receiving system rather than the PRBS degree. Nevertheless, higher degree PRBS sequences, with their longer periods and more diverse patterns, are advantageous for testing various characteristics of the receiver system.

Random Jitter (RJ) refers to unpredictable variations in the timing of signal transitions, which are generally caused by thermal noise, interference, or other stochastic factors.[19,20] Measuring RJ allows for the evaluation of timing stability in communication systems under noisy environments and provides insight into how different PRBS (Pseudo-Random Binary Sequence) patterns affect the robustness of timing extraction. The jitter of PRBS was measured based on variations in signal transition points. The rising edges of the reconstructed signal were detected, and the intervals between consecutive edges were calculated. The standard deviation of these intervals relative to the average bit period was defined as the RJ value. Figure 5 shows the RJ of PRBS sequences measured under various AWGN SNR conditions. As the SNR decreases in an AWGN environment, the RJ of all PRBS sequences tends

DOI: 10.36838/IJHSR810.37

to increase gradually. This is because noise makes the signal's zero crossings more ambiguous, thereby increasing the uncertainty of timing detection. In particular, as the PRBS degree increases, the magnitude of RJ growth under the same SNR conditions becomes more pronounced. This result is related to the fact that higher degree PRBS sequences have longer periods, and their frequency spectra become closer to white noise. In other words, spectral components distributed uniformly across the frequency band interact with noise, making the timing extraction process more sensitive and resulting in greater RJ growth.

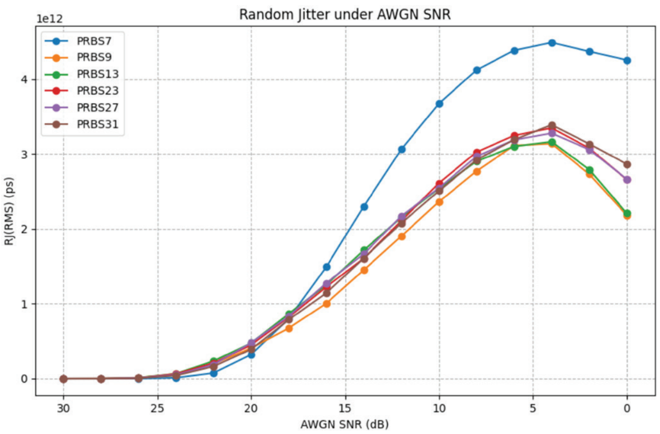

Figure 5: Random Jitter under AWGN SNR. This figure illustrates the RMS random jitter measured for several PRBS sequences as a function of AWGN SNR. Shorter sequences such as PRBS7 exhibit significantly higher jitter, especially in moderate-to-low SNR regions, due to their poorer spectral flatness. In contrast, longer sequences exhibit more noise-like spectra and therefore show a more gradual and stable RJ increase.

In contrast, PRBS7 exhibited distinctly different behavior from other PRBS sequences. Since PRBS7 has a short period, it produces prominent tonal components in the frequency domain. Such periodic spectral structures are interpreted to interact with AWGN and abnormally amplify the instability of timing extraction. As a result, PRBS7 does not follow the general gradual increase pattern of RJ but instead shows excessive amplification. On the other hand, longer-period sequences such as PRBS23 and PRBS31 have spectra closer to white noise, thereby exhibiting a more gradual increase in jitter under similar conditions. In addition, when SNR decreases, RJ initially increases gradually, but at very low SNR levels, it is observed to decrease. This is because, at moderate SNRs, noise blurs the signal zero crossings and increases timing uncertainty. In contrast, at extremely low SNRs, the signal is almost completely buried in noise, and timing information is lost. In this case, the observed RJ is governed more by random noise than by meaningful signal characteristics, resulting in saturation or even reduction. Therefore, the degree and period of PRBS sequences influence RJ characteristics, and it can be concluded that short-period sequences, such as PRBS7, exhibit disadvantageous properties in terms of timing extraction stability under noisy conditions.

Hardware cost:

While PRBS generation speed remains largely independent of sequence degree, the complexity of the checker logic increases with PRBS degree due to the larger number of LFSR registers and feedback XOR gates, as shown in Table 5. Consequently, higher degree PRBS sequences require more hardware resources and may incur slightly longer synchronization times in receiver systems, although bit-level computation speed per cycle remains similar.

Table 5: Hardware cost for PRBSs. This table compares several PRBS generators in terms of hardware cost and implementation complexity. Higher-order PRBS sequences require more registers and XOR gates, which increases hardware overhead and checker complexity. However, the computation speed remains similar across all PRBS types.

PRBS	Registers	XOR gates	Hardware Cost	Checker Complexity	Computation Speed
PRBS7	7	1	100%	Low	
PRBS9	9	1	122%	Low	
PRBS11	11	1	157%	Low	
PRBS13	13	3	189%	Medium	
PRBS15	15	1	214%	Medium	Similar
PRBS19	19	3	271%	Medium	
PRBS23	23	1	328%	High	
PRBS27 [18]	27	3	385%	High	
PRBS31	31	1	442%	High	

The proposed design was evaluated using Python-based simulations. While hardware implementation using FPGA platforms and hardware description languages such as Verilog or VHDL was beyond the scope of this study, such an implementation could enable direct measurement of real-world performance metrics, including power consumption, logic utilization, and timing characteristics, and is considered an important direction for future work.

Results and Discussion

This study systematically analyzed PRBS sequences with respect to sequence degree, statistical properties, and hardware implementation cost. From a statistical perspective, the run length distribution tests confirmed that all selected PRBS sequences closely follow the expected patterns of maximum-length sequences and maintain an almost perfect balance between 0s and 1s. Frequency simulation showed that a higher degree of PRBS shows particularly good balance and randomness within a single sequence cycle, findings further supported by entropy analysis. So, higher degree PRBS shows the theoretical maximum entropy; however, lower degree PRBS shows slightly lower entropy due to shorter periods. PSD analysis revealed that long PRBS sequences distribute energy more evenly across frequencies, with lower variance and improved spectral flatness. In contrast, short sequences such as PRBS7 show pronounced periodic spectral components. The simulation results indicate that a higher degree of PRBS provides pseudo-random characteristics close to ideal white noise.

All PRBS sequences were generated using LFSRs with predefined primitive polynomials and non-zero initial seeds. For each PRBS degree, maximum-length sequences were generated and evaluated under identical simulation conditions to ensure fair comparison across different orders. Jitter and BER measurements were obtained through repeated simulations under controlled AWGN levels to ensure statistical consistency and reproducibility.

In an AWGN environment, BER measurements indicated that error rates were largely unaffected by the PRBS degree.

DOI: 10.36838/IJHSR810.37

Their BER is primarily determined by channel conditions and receiver architecture rather than the choice of test pattern.

Jitter measurements revealed that, excluding PRBS7, sequences generally exhibited a gradual increase in RJ as the SNR decreased, with higher degree PRBS sequences showing a larger increase compared to lower degree sequences. In contrast, PRBS7 shows poor jitter performance due to a very short period and repeatability. This interacted with AWGN, resulting in an abnormal amplification of timing instability. For longer-period sequences such as PRBS23 and PRBS31, no significant differences in jitter were observed among different PRBS degrees. This suggests that when the sequence period is sufficiently long, the effect of noise on jitter becomes largely independent of the sequence degree.

From a hardware perspective, PRBS generation requires only a single feedback operation per clock cycle, so generation speed is independent of sequence length. However, implementation complexity increases with sequence degree. Implementation for higher degree PRBS requires more registers and XOR gates, which can increase the logic area and power consumption. Based on the analysis, lower-degree PRBSs can provide a reasonable balance between randomness and hardware efficiency. Higher-degree PRBSs are particularly useful for evaluating receiver performance and jitter tolerance, but their implementation costs are high.

Ultimately, the optimal PRBS degree should be selected based on the design objectives. Low-degree PRBSs are sufficient for rapid validation and low complexity testing. High-degree PRBSs are preferred for system-level verification when timing stability and signal quality are of concern, due to their superior statistical properties and white-noise-like behavior.

Conclusion

This study presented a comparative analysis of various kinds of PRBS sequences ranging from PRBS7 to PRBS31, simulating statistical properties and noise robustness, and also analyzing hardware implementation requirements. Run length, frequency, entropy, autocorrelation, and PSD analyses demonstrated that higher degree PRBS closely approximate ideal white noise, while lower degree sequences show predictable periodic features. BER measurements under AWGN confirmed that bit error performance was largely independent of PRBS degree. Furthermore, jitter analysis results showed that noise-induced jitter increased with increasing PRBS degree. However, for very short sequences, such as PRBS7, noise-induced jitter characteristics were significantly worse due to the periodic spectral components. Hardware analysis showed that implementation complexity increases with sequence degree, although computation speed remains unaffected. By systematically evaluating these factors, low-degree PRBSs are found to be suitable for efficient testing and low complexity validation, and high-degree PRBSs are found to be suitable for system-level verification where statistical randomness and receiver performance assessment are important. This study provides objective criteria for selecting appropriate PRBS sequences based on design priorities. So, a balanced tradeoff can be achieved between statistical quality, timing robustness, and implementation cost.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Bongkyu Kim, at Synopsys Inc., for his guidance, valuable insights, and helpful discussions throughout this research.

References

1. Guin, U.; Chiang, C.-H. Design for Bit Error Rate Estimation of HighSpeed Serial Links. *Proc. IEEE VLSI Test Symposium* 2011, 1–6. https://doi.org/10.1109/VTS.2011.5783734
2. Zepernick, H.-J.; Finger, A. *Pseudo Random Signal Processing: Theory and Application*; pp 225–389, 2005.
3. Mishra, K. *Advance Chip Design*; CreateSpace Independent Publishing Platform, pp 131–136, 2013.
4. Naim, M.; Pacha, H. A.; Pacha, A. A.; Said, N. H. Lengthening the Period of a Linear Feedback Shift Register. *J. Eng. Technol. Appl. Sci.* 2021, *6*, 45–68. https://doi.org/10.30931/jetas.778792
5. Mioc, M. A.; Stratulat, M. Study of Software Implementation for Linear Feedback Shift Register Based on 8th Degree Irreducible Polynomials. *Int. J. Comput.* 2014, *8*, 46–55. https://www.naun.org/main/NAUN/computers/2014/a042007-097.pdf (accessed Feb 2025)
6. Kadir, R.; Maarof, M. A. A Comparative Statistical Analysis of Pseudorandom Bit Sequences. *Proc. Int. Conf. Inf. Assur. Secur.* 2009, 91–94. https://doi.org/10.1109/IAS.2009.242
7. Parol, M.; Dabal, P.; Szplet, R. PseudoRandom Bit Generators Based on LinearFeedback Shift Registers in a Programmable Device. *J. Meas. Autom. Monit.* 2016, *6*, 184–186. https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-ffcf8b68-e4e5-46b2-93bc-6731f865de42/c/Parol_pseudo-random_MAM_6_2016.pdf (accessed Feb 2025)
8. Mandrona, M. N.; Maksymovych, V. N. Comparative Analysis of Pseudorandom Bit Sequence Generators. *J. Autom. Inf. Sci.* 2017, *49*, 78–86. https://doi.org/10.1615/JAutomatInfScien.v49.i3.90
9. Sindal, S. S.; Paliwal, A. Analysis of PseudoRandom Binary Sequence and Gold Sequence to Achieve High Speed Optical Communication System. *J. Emerg. Technol. Innov. Res.* 2018, *5*, 51–55. [Online]. Available: https://www.jetir.org/papers/JETIRB006013.pdf (accessed Feb 2025)
10. Intel. 50G Interlaken Intel FPGA IP User Guide. https://www.intel.com/content/www/us/en/docs/programmable/683217/22-1/prbs-generation-and-validation.html (accessed Jan 2025)
11. Phabrix. Using PseudoRandom Binary Sequences to Stress Test Serial Digital Interfaces. https://phabrix.com/ftp/App_Notes/Stress_Whitepaper.pdf (accessed Jan 2025)
12. Okawara, H. DSPBased Testing – Fundamentals 50 PRBS (Pseudo Random Binary Sequence). https://www3.advantest.com/documents/11348/3e95df23-22f5-441e-8598-f1d99c2382cb/ (accessed Nov 2024)
13. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E.; Leigh, S.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; Dray, J.; Vo, S. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications; National Institute of Standards and Technology, 2010. https://dl.acm.org/doi/pdf/10.5555/2206233
14. Mertens, S.; Bauke, H. Entropy of Pseudo Random Number Generators. *Phys. Rev. E* 2004, *69*, 055702. https://doi.org/10.1103/PhysRevE.69.055702

15. Zucconi, A. The Autocorrelation Function. https://www.alanzucconi.com/2016/06/06/autocorrelation-function/ (accessed Mar 2025)
16. Orth, S.; Klingbei, H. Maximum Length Binary Sequences and Spectral Power Distribution of Periodic Signals. *EURASIP J. Adv. Signal Process*. 2024, *80*. https://doi.org/10.1186/s13634-024-01177-5
17. Solomon, O. M. PSD Computations Using Welch's Method; Sandia National Labs., 1991. https://doi.org/10.2172/5688766
18. Promise Elechi. Performance Analysis of BER and SNR of BPSK in AWGN Channel. *Int. J. Digit. Analog Commun. Syst*. 2022, *7*, 27–38. Available: https://www.researchgate.net/publication/358825659_Performance_Analysis_of_BER_and_SNR_of_BPSK_in_AWGN_Channel (accessed Mar 2025)
19. Schultz, A. Random Jitter – What Is Really Going On? *EE Times*, Oct 22, 2001. https://www.eetimes.com/random-jitter-what-is-really-going-on/ (accessed Mar 2025)
20. Tranchemontagne, M. Jitter Basics, Advanced, and Noise Analysis; Tektronix, 2016. https://www.scribd.com/document/468415514/jitter-basics-advanced-pdf (accessed Mar 2025)

■ Author

Jaechan Lee is a junior at LASA High School in Austin, Texas. He is passionate about electrical engineering and aerospace engineering, with a strong interest in developing technologies that digital logic design for pseudo-random number generation.

DOI: 10.36838/IJHSR810.37