電気学会論文誌 D（産業応用部門誌）
IEEJ Transactions on Industry Applications
Vol.143 No.2 pp.87–93　DOI: 10.1541/ieejias.143.87

論　文

内蔵LFSRを用いた乱数生成方法の評価

非会員　鴨狩　滉斗*　正　員　市川　周一**a)

Evaluation of a Random Number Generator based on an Internal Linear Feedback Shift Register

Hiroto Kamogari*, Non-member, Shuichi Ichikawa**a), Member

（2022年3月16日受付，2022年8月24日再受付）

A previous study presented a random number generator based on the fluctuation in the sampling interval of an internal linear feedback shift register (LFSR). They reported that the derived random numbers passed the Diehard test; however, the design guidelines were not specified. The previous study examined two configurations of LFSR, which are far from satisfactory to determine the appropriate length and feedback polynomial. Moreover, the minimal sampling period was not known. The present study focuses on these aspects and elucidates the requirements of LFSRs to generate high-quality random numbers. Extensive experiments were conducted, and the following results were derived. (1) The selection of the characteristic polynomial of an LFSR affects the quality of random numbers. (2) The length of an LFSR should be 48 bits or greater with a constant sampling period. (3) The sampling period should be 32 cycles or larger. (4) The quality of random number sequence is considerably improved by a fluctuating sampling period. (5) The properly designed random number generator passes the NIST test.

キーワード：乱数，URNG, TRNG, LFSR
Keywords: random number, URNG, TRNG, LFSR

1. はじめに

乱数生成器は，一般に，真性乱数生成器（True Random Number Generator; TRNG）と擬似乱数生成器（Pseudo Random Number Generator；PRNG）に分類される。TRNGは熱雑音等の物理現象を利用して乱数を生成するため，出力の予測は不可能であるが，専用ハードウェアが必要になる。PRNGは確定的アルゴリズムによって乱数の持つ統計的性質を再現するもので，ソフトウェアだけで実装できるが，アルゴリズムと内部状態から出力を予測可能である。

Suciuら[1]は，Intel社のCPUがもつパフォーマンスカウンタ（Performance Counter；PFC）を内部状態として利用することにより，実質的に予測不可能な乱数生成手法を提案した（Unpredictable Random Number Generator；URNG）。プロセッサ自体は確定的に動作するが，動作時の外部入力に存在する不確定要因から乱数を生成するため，その出力は実質的に予測不可能になる。しかしPFCには充分な乱数性（エントロピー）がないため，高い乱数品質を得ることは難しかった[2]。

正岡ら[2]は，プロセッサ内部に設けたLFSR（Linear Feedback Shift Register）をPFCの代りに用いることにより，高品質な乱数が生成できると述べた。正岡らは提案手法をFPGAで実装し，生成される乱数列がDiehardテスト[3]を通過することを示した。しかし正岡らの研究には以下の課題が残されている。

（1） 検討したのは2種類のLFSR構成だけで，実装するLFSRが満たすべき条件が明らかでない。
（2） 実装上の制約で生成速度が低く，乱数生成速度の上限が明らかになっていない。
（3） DiehardテストよりNISTテスト[4]を通過

a) Correspondence to: Shuichi Ichikawa. E-mail: ichikawa@ieee.org
* 豊橋技術科学大学 電気・電子情報工学課程
〒441-8580　愛知県豊橋市天伯町雲雀ヶ丘1-1
Electrical and Electronic Information Engineering Course, Toyohashi University of Technology
1-1, Hibarigaoka, Tampaku-cho, Toyohashi, Aichi 441-8580, Japan
** 豊橋技術科学大学 電気・電子情報工学系
〒441-8580　愛知県豊橋市天伯町雲雀ヶ丘1-1
Department Electrical and Electronic Information Engineering, Toyohashi University of Technology
1-1, Hibarigaoka, Tampaku-cho, Toyohashi, Aichi 441-8580, Japan

するか，検証されていない。

本研究では，課題 (1) と (2) について詳細なシミュレーションで検討し，正岡らの URNG について具体的な設計指針を確立する。最後に課題 (3) についてシミュレーションを行い，設計指針に従った URNG が NIST テストに合格することを確認する。

以下，2 章では正岡らの URNG について説明し，3 章で LFSR 構成と乱数品質の関係について検討する。次に 4 章で，サンプリング周期の揺らぎの影響と，期待される生成速度について述べる。5 章では，NIST テストを用いて乱数品質の検証を行う。最後に 6 章で，正岡らの URNG を実装する際の設計指針をまとめる。

なお本稿は，著者らによる研究会発表[5]に加筆修正を施したものである。

2. 背　景

〈2･1〉 LFSR　LFSR は簡易な PRNG として，通信やテストパタン生成などに広く用いられている。LFSR は入力ビットが前状態の線形写像になっているシフトレジスタである。この線形写像は特性多項式（帰還多項式）によって表現され，特性多項式が原始多項式である場合に LFSR の周期は最長になる（M 系列）。n-bit LFSR の最長周期は $2^n - 1$ であり，このとき 0 以外の全ての状態を経由する。

Fig. 1 は，8 ビットのフィボナッチ型 LFSR の例である。この LFSR の特性多項式は $x^8 + x^6 + x^5 + x^4 + 1$ であり，ビット 8，6，5，4 の排他的論理和（XOR）が次の入力となる。以下の議論では，特性多項式の代わりにタップシーケンス [8, 6, 5, 4] を用いて LFSR を表現する。

〈2･2〉 正岡らの URNG　正岡ら[2]は，システムクロックに同期した LFSR を専用レジスタとしてプロセッサに追加し，ソフトウェアで LFSR の値をサンプルすることにより，乱数列が生成できると報告した。正岡らの提案方式は，外乱によるサンプリング周期の揺らぎにより実質的に値の予測が不可能になるため，URNG の一種に分類される。PFC の代わりに疑似乱数生成器（LFSR）を使用することにより，Suciu ら[1]の URNG と比べて乱数品質が向上する。

正岡らは 32-bit LFSR [32, 7, 5, 3, 2, 1] をシミュレーションで検討し，乱数検定に合格しないことを報告した。さらに 128-bit LFSR [128, 7, 2, 1] の下位 32 ビットが乱数検定に合格することを示し，RISC-V プロセッサに実装した。

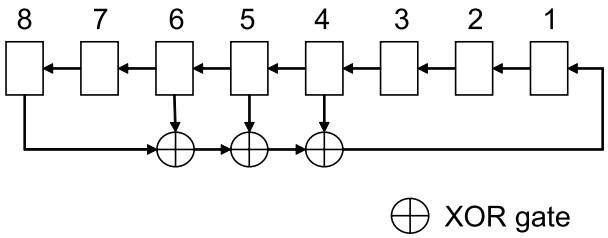

Fig. 1.　An example of 8-bit LFSR [8,6,5,4].

Xilinx Zync-7000 を用いた評価では，ハードウェアコストの増加はわずか（FF +1.4%，LUT −1.5%）で，生成速度は 125 kbit/s であった。生成された乱数列は Diehard テスト[3]に合格することが示された。

正岡らの URNG では，プロセッサに LFSR を追加する必要があるが，LFSR の回路規模は小さいため実装コストが非常に低い。また，ソフトウェアで LFSR を読み出すだけで乱数列が生成でき，後処理なしで Diehard テストに合格するなど，利用も極めて簡単である。一方，正岡らの実装では平均サンプリング周期が 5.1×10^3 サイクルと長く，乱数生成速度は 125 kbit/s に留まっていた。

本研究の目的のひとつは，乱数生成速度の改善方法を検討することである。サンプリング間隔と乱数品質にはトレードオフがあると考えられるので，その点も含めてシミュレーションで検討する。

〈2･3〉 Diehard テスト　本研究では，先行研究[2]と比較するため，乱数品質の検査に Diehard テスト[3]を用いる。Diehard テストは全 18 種のテストからなり，各テストで 1〜100 個（合計 313 個）の p 値を出力する。合格判定の基準は定められておらず，利用者の判断に委ねられているので，本研究でも先行研究[2]の基準に従って，以下のように乱数品質を評価する。

入力が理想的乱数であれば p 値は区間 [0,1) で均等に分布することが期待されるので[6]，Table 1 に示した基準で各 p 値の成功（PASS）／弱成功（WEAK）／失敗（FAIL）を判定する。FAIL の発生確率（期待値）は 2×10^{-6} なので，入力が乱数であれば（ほぼ）発生しない。WEAK の発生確率（期待値）は 1×10^{-2} なので，WEAK が 1%程度発生することは正常である。

単純に 313 個の p 値について PASS/WEAK/FAIL の個数を示すと，多くの p 値を出力するテストのウエイトが大きく見えてしまう。そこで以下の方法により，各テストの結果を判定する。各テストで出力される p 値の個数が 9 個以上であれば，得られた p 値の分布が一様であるかどうかの判定を Kolmogorov-Smirnov 検定により行い，得られた p 値を Table 1 に示した基準で判定する。テストの出力する p 値が 9 個未満であれば，以下に述べる方法で結果を判定する。各テストで出力される p 値に，ひとつでも FAIL が含まれれば，そのテストの結果は FAIL とする。出力される p 値に FAIL はなく，かつ WEAK が含まれれば，そのテストの結果は WEAK とする。出力される p 値が全て PASS であれば，そのテストの結果は PASS とする。

こうして計算した全 18 テストの結果（PASS/WEAK/FAIL の内訳）により，乱数列の品質を評価する。

Table 1.　Diehard evaluation criteria[2].

Decision	Condition
PASS	$0.005 \le p < 0.995$
WEAK	$0.000001 \le p < 0.005$, or $0.995 \le p < 0.999999$
FAIL	$p < 0.000001$, or $0.999999 \le p$

〈2･4〉 **NIST テスト**　NIST SP 800-22 [4] は，NIST（National Institute of Standards and Technology）により規定された乱数および疑似乱数の統計検定スイートであり，社会で広く認知され使用されている。NIST テストは 15 種のテストからなり，結果の解釈方法についても明確に定義されている。

NIST テスト [4] では 1 回のテストで約 10^6 ビットを使用し，そのテストを 1000 回以上行うことが推奨されている。Diehard テスト [3] では約 10^8 ビットのデータ量が必要であるが，NIST テストには合計 10^9 ビット程度が必要になる。従って本研究では，研究上の試行錯誤を行う際に簡易的テストとして Diehard テストを用い，最終的な乱数品質の検定には NIST テストを用いることにする。

3. LFSR の構成

正岡ら [2] の研究では，LFSR が満たすべき条件が明らかになってない。そこで本章では，様々な LFSR 構成の乱数品質を検討し，LFSR を用いた URNG を設計する際の設計指針を明らかにする。

〈3･1〉 **特性多項式と乱数品質**　M 系列を与える特性多項式（原始多項式）は複数あり，その周期は同じであるが，URNG に実装した際の乱数品質は異なる。Table 2 は，M 系列を生成する 64-bit LFSR のタップシーケンスの例である。M 系列のタップ数は偶数であるが，タップ数の多寡が品質に及ぼす影響を調べるため，Zivkovic [10] による 4 タップと 6 タップの多項式を取り上げた。また Rajski [11] の多項式は ring generator（LFSR の一種）のために選ばれたもので，タップがほぼ均等に配置されていることが特徴である。

これらの LFSR の下位 32 ビットを一定周期でサンプルし，〈2･3〉節の方法に従って乱数品質を評価した結果を Fig. 2 に示す。

Fig. 2 から，タップシーケンスの選択は乱数品質に大きな影響を及ぼすことがわかる。[64, 4, 3, 1] と [64, 63, 61, 60] は全般に乱数品質が低いが，特に周期 64 前後の品質低下が著しい。一方，Zivkovic および Rajski のタップ配置では乱数品質が高く，周期 64 前後でも品質は低下しない。タップ配置が適切であれば，32 サイクル以上のサンプリング周期で概ね全てのテストに合格する†。

周期 64 での品質低下は，以下のように説明できる。N-bit LFSR は N ビットの巡回構造をもつので，サンプリング周

Table 2. Example Configurations of 64-bit LFSR.

Source	Tap sequence
Schneier [7]	[64, 4, 3, 1]
Xilinx [8][9]	[64, 63, 61, 60]
Zivkovic [10]	[64, 61, 34, 9]
Zivkovic [10]	[64, 61, 56, 31, 28, 23]
Rajski [11]	[64, 45, 31, 14]

† [64, 61, 34, 9] の周期 40 で FAIL が発生しているが，それについては〈3･3〉節で改めて議論する。

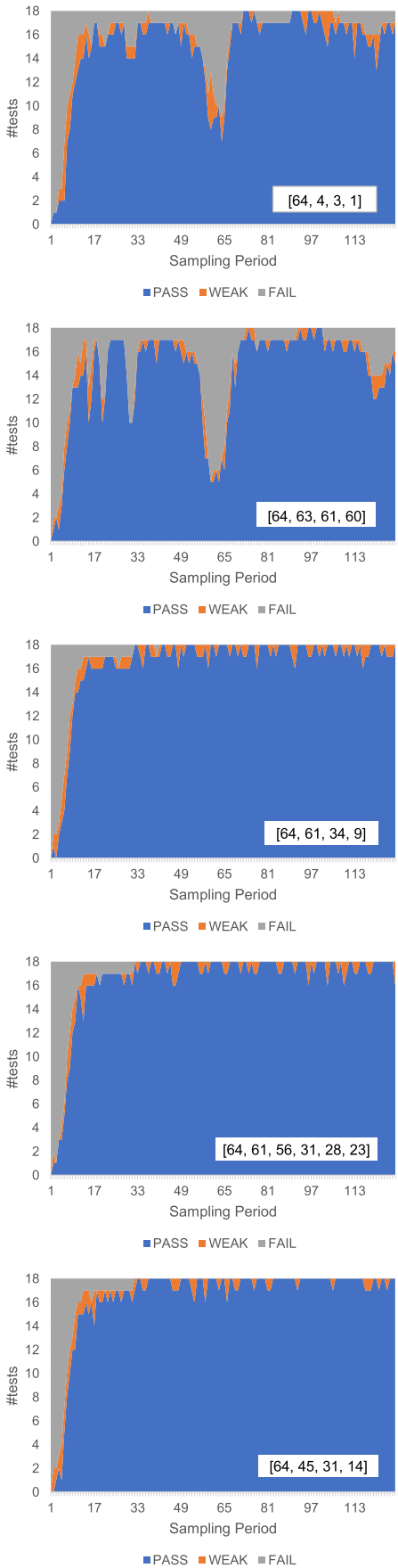

Fig. 2. Diehard results of 64-bit LFSR with various characteristic polynomials.

期が N サイクル前後では連続したサンプルに相関が生じやすい。相関が生じると，その点をチェックしているテストに失敗する。このときタップの配置が適切であれば，相関が抑制されて乱数品質が向上し，テストに合格する。

次に，32，48，64，96，128 ビットについてタップ数 4 と 6[10]の乱数品質を比較した。64 ビットの結果は Fig. 2 に含まれている。我々の実験の範囲内では，タップ数による品質の差は発見できなかった。

正岡ら[2]は，32-bit LFSR [32, 7, 5, 3, 2, 1] と 128-bit LFSR [128, 7, 2, 1] を評価に使用した。これらはいずれも Schneier[7]が出典である。我々のシミュレーションによれば，これらは周期 32 前後・128 前後で乱数品質が低下し，良いタップシーケンスではない。一方，Zivkovic および Rajski のタップ配置を採用すると，このような品質低下は現れなかった。これらの結果は，LFSR 長が異なるものの Fig. 2（64-bit LFSR）と本質的に同じなので，詳細は省略する。

〈3･2〉 LFSR 長と乱数品質　本節では，適切なタップシーケンスの採用を前提として，乱数検定を通過するために必要な LFSR 長を検討する。正岡ら[2]は，32 ビットでは不合格，128 ビットでは合格と述べたが，それ以上詳細な結果は示していない。本研究では〈3･1〉節の結果に基づき，Rajski[11]による 32～128 ビットのタップシーケンスを評価する（Table 3）。

Fig. 3 は Diehard テストによる評価結果である。[64, 45, 31, 14] の結果は，Fig. 2 に含まれるものと同じである。

32 ビット長では，サンプリング周期を伸ばしてもテストに合格できない。合格できないテストは，Binary Rank (31x31) と Binary Rank (32x32) である。32-bit LFSR の周期 $2^{32}-1$ が，これらのテストに対して短すぎるためと考えられる。48 ビット以上の LFSR では，サンプリング周期 32 以上で概ね全てのテストに合格する†。

32 ビットと 48 ビットの間を詳しく調べることも可能だが，本研究ではこれ以上調べていない。LFSR 長を削るとハードウェアの実装コストは下がるが，そもそも LFSR の論理規模は小さいため，これ以上の調査は必要ないと判断した。

〈3･3〉 LFSR の初期値　Fig. 2 では，[64, 61, 34, 9] の周期 40 で FAIL が発生している。同様に Fig. 3 では，[48, 38, 26, 13] の周期 53 と 61，および [128, 105, 83, 62, 42, 21] の周期 48 と 49 で FAIL が発生している。これらの現象

Table 3.　Primitive Polynomials by Rajski[11].

Length	Tap sequence
32	[32, 25, 15, 7]
48	[48, 38, 26, 13]
64	[64, 45, 31, 14]
96	[96, 79, 64, 49, 33, 16]
128	[128, 105, 83, 62, 42, 21]

† [48, 38, 26, 13] の周期 53 と 61，および [128, 105, 83, 62, 42, 21] の周期 48 と 49 で FAIL が発生しているが，それについては〈3･3〉節で改めて議論する。

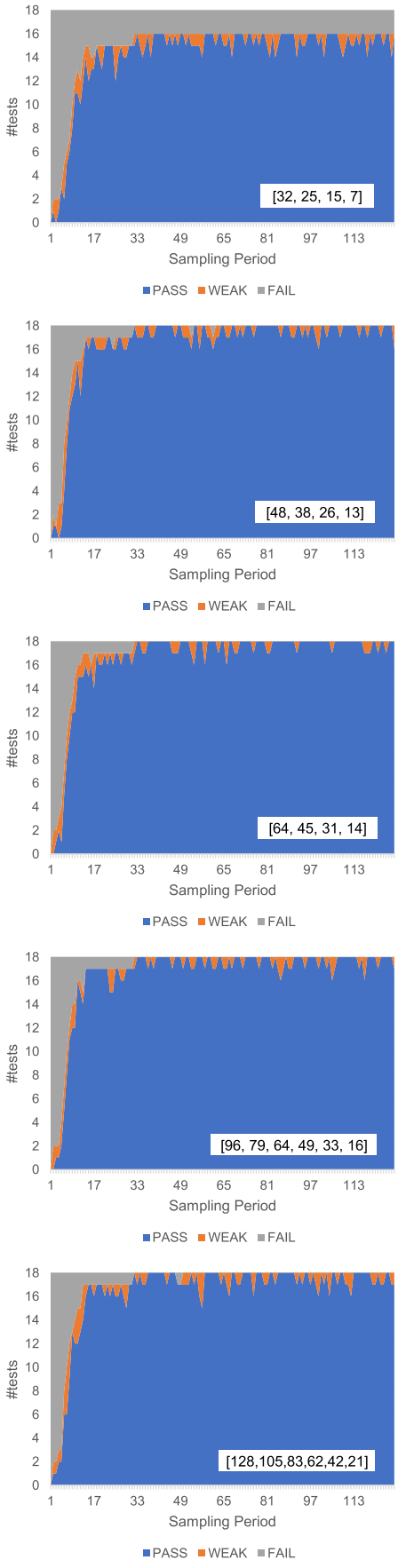

Fig. 3.　Diehard results for various LFSR lengths.

について調べたところ，全て Diehard テスト内の OPERM5 で FAIL していた。

しかしこれらの FAIL が発生する条件下で，シミュレーションにおける LFSR の初期値を変えると，全てのテストが PASS か WEAK となる（FAIL は発生しない）ことが確認できた。本研究ではシミュレーションの条件を統一するため，全てのタップシーケンス，全てのサンプリング周期について，同じ値で LFSR を初期化している。しかし M 系列の LFSR は 0 以外の全ての内部状態を巡回するため，こうした FAIL はタイミング次第で発生する，ということになる。

本研究の評価において，こうした FAIL の観察例は多くないが，こうした状況の発生確率については現在まで明確になっていない。また，こうした FAIL の有無（発生確率）が特性多項式によって異なるのか，その点も分かっていない。本研究を踏まえて，今後検討を進める必要がある。

4. サンプリング周期の揺らぎ

3 章までの評価では，サンプリング周期を一定として評価した。実機では，割込み・キャッシュミス・パイプラインフラッシュなどの諸要因により，サンプリング周期が影響を受けて一定にならない。このサンプリング周期の揺らぎにより，提案手法の出力は「実質的に予測不可能」になる。〈3·1〉節の例でも見た通り，サンプリング周期が一定であると（LFSR の性質上）乱数品質が低下する可能性がある。その意味で「サンプリング周期一定」という条件は，悪条件での評価になると思われる。

そこで本章では，サンプリング周期に一定の揺らぎを加えてシミュレーションを行い，乱数品質に現れる変化について調査する。LFSR が長いと悪条件でも Diehard テストに合格するため，評価結果に差が表れない。そこで，〈3·2〉節でテストに合格できなかった 32 ビット長で評価を行う。タップシーケンスは [32, 30, 17, 12, 3, 1] [10] を採用した。

1 サンプル毎のサンプリング間隔 P を，基準周期 C（定数），一様乱数 x（$0 \le x < 1$），重み α を用いて，$P = \lfloor C(1+\alpha x) \rfloor$ で決めるものとする。一様乱数の生成にはメルセンヌツイスタ MT19937 [12] を使用した。$\alpha = 0.00, 0.05, 0.25$ の結果を Fig. 4 に示す。

$\alpha = 0.00$ は，サンプリング周期一定という条件なので，〈3·2〉節で示した通り C が大きくても Diehard テストに合格できない。

$\alpha = 0.05$ では，$C \ge 32$ のときテストに合格する。$C = 32$ のとき $32 \le P \le 33$ となり，P の期待値は 32.375 である。すなわち揺らぎは 1 サイクル程度と僅かであるが，明かな乱数性の向上が認められる。

$\alpha = 0.25$ でも，$C \ge 32$ のときテストに合格する。$C = 32$ のとき $32 \le P \le 39$ となり，P の期待値は 35.5 である。$\alpha = 0.00$ の $C = 36$ ではテストに合格できないことから，乱数性の向上はサンプリング周期の揺らぎに起因することは明らかである。特に $10 \le C \le 31$ では $\alpha = 0.05$ より品質の向上が見られる。

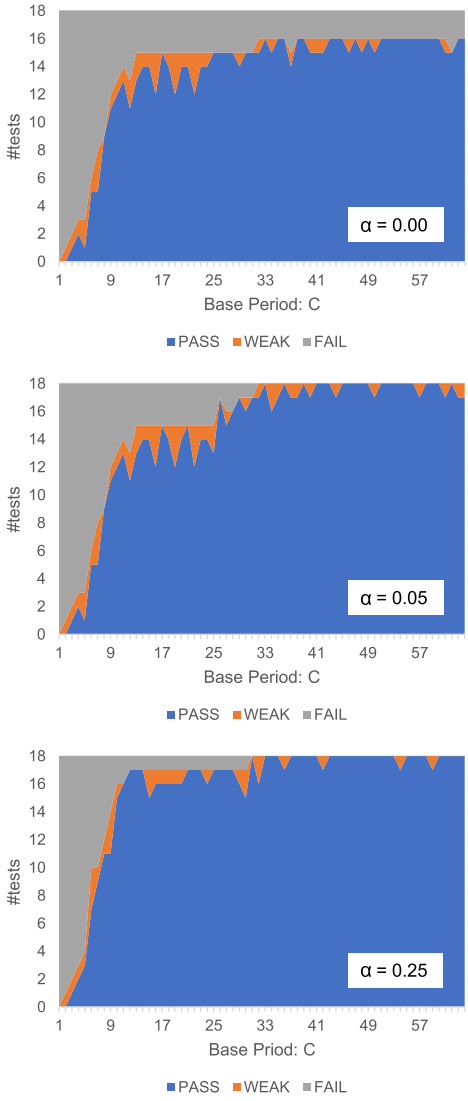

Fig. 4. Diehard results for fluctuated sampling interval.

以上より，サンプリング周期に揺らぎがあれば乱数品質が向上することが確認された。3 章のように，サンプリング周期の揺らぎなし（最悪条件のシミュレーション）で乱数検定に合格すれば，揺らぎのある実機においても乱数検定に合格することが期待できる。

また，以上の評価結果より，LFSR を用いた URNG の乱数生成速度の上限を見積もることができる。システム（LFSR）の周波数を f [MHz] としたとき，32 サイクル毎に 32 ビットの乱数を生成することができるので，期待される乱数生成速度は f [Mbit/s] となる。

正岡ら [2] の実装では，RISC-V プロセッサ（PULPino）の動作周波数が 20 MHz であるため，適切な実装を施せば 20 Mbit/s の生成速度が期待できる。正岡らの実装では実測値で 0.125 Mbit/s なので，100 倍以上の性能改善が可能と思われる。

Table 4. Results of NIST test.

	LFSR		Sampling Period		
	Length	Tap sequence	32	64	128
This study	32	[32, 16, 7, 2]	(Binary Matrix Rank)	(Binary Matrix Rank)	(Binary Matrix Rank)
	48	[48, 19, 9, 1]	(Overlapping Template Matching)	All passed	All passed
	64	[64, 61, 34, 9]	All passed	All passed	All passed
Masaoka [2]	128	[128, 7, 2, 1]	(Maurer's Universal Statistical)	All passed	(Non-overlapping Template Matching)

5. NIST テストによる検証

Table 4 に，NIST テストの結果をまとめる。不合格の場合は，カッコ内に失敗したテスト名を示している。本研究では 32，48，64 ビットの LFSR について，サンプリング周期 32，64，128 で乱数列を生成し，NIST テストで検定した。また，先行研究と比較するため，正岡ら [2] が使用した 128-bit LFSR についても同様に検定した。なお本研究で用いたタップシーケンスは Zivkovic [10] によるものであり，正岡らのタップシーケンスは Schneier [7] によるものである。サンプリング周期の揺らぎは加えていない。

Table 4 から明らかなように，LFSR が 48 ビット以上かつサンプリング周期 64 以上で NIST テストに合格している。32-bit LFSR では内部状態が少ないため Binary Matrix Rank テストに失敗する。これは Diehard テストの結果と整合している（〈3·2〉節）。48-bit LFSR でも，サンプリング周期が短い場合 (32) には Overlapping Template Matching テストで失敗するが，周期が 64 以上であれば全てのテストに合格する。

正岡の設計 [2] では，LFSR は 128 ビットと大きいが，タップシーケンスに問題がある。そのためシミュレーションにおいて，サンプリング周期 128 で NIST テストに合格することができない。このシミュレーションでは揺らぎ無し（最悪条件）で評価しているので，実機では外部要因による揺らぎでテストに合格できる可能性もある。その意味で，不適切な設計とまでは言えないが，本研究の成果に照らして考えると改良の余地がある。今後，本研究の成果を踏まえた再設計と，実機での再評価が望まれる。

以上の結果から，適切な設計を行えば，提案する URNG は NIST テストに合格することがわかった。本研究により，適切な設計を選択するための基準が新たに明らかになった。

6. おわりに

本研究では，実質的に予測不能な乱数生成器（URNG）を正岡ら [2] の手法で実現するため，内蔵 LFSR の具体的設計指針をシミュレーションにより検討した。

その結果，以下の各項目が明らかになった。

- LFSR の特性多項式（原始多項式）の選択は，乱数品質に影響を与える。
- LFSR の長さは 48 ビット以上必要で，64 ビット以上が望ましい。（サンプリング間隔が一定の場合）
- サンプリング間隔は 32 サイクル以上必要で，64 サイクル以上が望ましい。
- サンプリング間隔の揺らぎは，1 サイクル程度でも乱数品質が向上する。
- 設計指針に従った URNG が NIST テストに合格することを確認した。

本研究ではシミュレーションによる検討を行ったが，今後は実機上で乱数品質を評価し，シミュレーションとの一致を確認する予定である。さらに正岡らの URNG において乱数生成速度が改善されることを確認し，本方式の URNG の実用化と普及を推進したい。

謝　辞

本研究の一部は JSPS 科研費 20K11733 の支援による。

文　献

（1） A. Suciu, S. Banescu, and K. Marton: "Unpredictable random number generator based on hardware performance counters", Digital Information Processing and Communications (ICDIPC 2011), pp.123–137, Springer-Verlag (2011)

（2） H. Masaoka, S. Ichikawa, and N. Fujieda: "Random Number Generation from Internal LFSR and Fluctuation of Sampling Interval", *IEEJ Trans. IA*, Vol.141, No.2, pp.86–92 (2021) (in Japanese)
正岡秀崇・市川周一・藤枝直輝：「内蔵 LFSR とサンプリング間隔の揺らぎを利用した乱数生成手法」, 電学論 D, Vol.141, No.2, pp.86–92 (2021)

（3） G. Marsaglia: "Diehard battery of tests of randomness (Archived)", https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/

（4） A. Rukhin, et al.: "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", NIST SP 800-22 (Rev. la) (2010)

（5） H. Kamogari and S. Ichikawa: "An investigation of random number generation based on the internal LFSR", The Papers of Technical Meeting on Innovative Industrial System, IEE Japan, IIS-21-011 (2021) (in Japanese)
鴨狩滉斗・市川周一：「内蔵 LFSR を用いた乱数生成に関する検討」, 電学次世代産業システム研, IIS-21-011 (2021)

（6） IPA ISEC：「電子政府情報セキュリティ技術開発事業擬似乱数検証ツールの調査開発 調査報告書」, 情報処理振興事業協会 セキュリティセンター (2003)

（7） B. Schneier: Applied Cryptography, John Wiley & Sons (1996)
B. Schneier：暗号技術大全, ソフトバンクパブリッシング, 東京 (2003)

（8） P. Alfke: "Efficient Shift Registers, LFSR Counters, and Long Pseudo Random Sequence Generators", XAPP 052, ver.1.1, Xilinx (1996)

（9） M. George and P. Alfke: "Linear Feedback Shift Registers in Virtex Devices", XAPP 210, ver.1.3, Xilinx (2007)

（10） M. Zivkovic: "A table of primitive binary polynomials", Mathematics of Computation, Vol.62, No.205, pp.385–386 (1994)

（11） J. Rajski and J. Tyszer: "Primitive Polynomials Over GF(2) of Degree up to 660 with Uniformly Distributed Coefficients", Journal of Electronic Testing, Vol.19, pp.645–657 (2003)

（12） M. Matsumoto: "Mersenne Twister Home Page", http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/mt.html

鴨 狩 滉 斗（非会員） 2019 年都立産業技術高等専門学校ものづくり工学科卒業。同年豊橋技術科学大学電気・電子情報工学課程編入学。2021 年同大学同課程卒業。

市 川 周 一（正員） 1985 年東京大学理学部卒業。1987 年東京大学大学院理学系研究科修士課程修了。1987 年新技術事業団，1991 年三菱電機（株），1994 年名古屋大学工学部助手。1997 年豊橋技術科学大学工学部講師。同助教授，准教授を経て，2011 年沼津工業高等専門学校制御情報工学科教授。2012 年より豊橋技術科学大学大学院工学研究科教授。現在に至る。理学博士。IEEE（senior member），電子情報通信学会（シニア会員），ACM，情報処理学会，各会員。