令和２（２０２０）年度　卒業研究報告書概要

課程，学籍番号，氏名	課程：電気・電子情報　工学課程，学籍番号： B193226，氏名： 鴨狩　滉斗
工学分野名：情報通信システムコース	指導教員名：市川　周一

題　　目：和　内蔵 LFSR を利用した乱数生成手法の設計指針

（英　Design Guidelines for Random Number Generation Methods using the Built-in LFSR　）

Abstract

This study investigates a low-cost and high-performance URNG (Unpredictable Random Number Generator), which was proposed by Masaoka et al. (2020). This study examines various design parameters, and shows a design guideline to derive high-quality random numbers.

The proposed guideline includes four requirements. First, LFSR bit length should be longer than or equal to 48. Second, four LFSR taps are enough. Third, the tap positions of LFSR should be balanced. And fourth, the sampling interval should be longer than or equal to 32-cycle.

Further verifications are desired, particularly using NIST 800-22 randomness test.

概　　要

暗号技術の運用に必要となる乱数生成器には、TRNG (True Random Number Generator：真性乱数生成器)と PRNG (Pseudo Random Number Generator：疑似乱数生成器)の 2 種類があり、PRNG は低コストであるが生成列が予測し易いという問題を抱えている。これを解決するために Suciu ら(2011)は URNG (Unpredictable Random Number Generator)という乱数生成手法を提案した。URNG は、CPU の内部状態をエントロピ源として疑似乱数生成をすることにより、実質的に予測不可能な乱数生成を実現する方法である。

正岡ら(2020)は、組込みシステムに向けた低コストかつ高性能な URNG を提案するという目的から、LFSR (Liner Feedback Shift Register：線形帰還シフトレジスタ)を用いて URNG を実現するという手法を提案した。実機に実装した結果、ビット長さ 128、サンプリング間隔 5000 サイクルで DIEHARD テストを PASS すると報告した。しかし正岡らの提案には、LFSR の仕様 (タップシーケンス)が限定的、必要最低限のサンプリング間隔が明確でない、高度なテストである NIST テストによる乱数評価ができていないといった課題点が残っている。

本研究の目的は、組込みシステムに向けた低コストかつ高性能な URNG を提案することである。そのため、正岡らの手法に残った課題点を解決するために、シミュレーションによって LFSR の設計および乱数生成,評価を行った。シミュレーションでは、C 言語で設計,記述した LFSR を用いて乱数生成を行い、生成された乱数列を乱数テストに通して乱数品質を確かめた。乱数テストに用いるテストスイートは DIEHARD を用いた。テスト結果の合格基準は正岡らの先行研究に倣って決定した。

検証内容として、(1) サンプリング間隔を長くする、(2) LFSR の長さを変更する、(3) LFSR のタップ数を変更する、(4) (1)～(3)の仕様が同じである別のタップシーケンスを用いる、(5) サンプリング間隔に揺らぎを与える、といった操作をした場合に乱数品質はどうなるのかを検証した。その結果、LFSR による乱数生成で DIEHARD テストを PASS するための必要条件が明らかになった。条件は、① LFSR の必要なビット長さは 48 以上、② タップは 4 つで十分、③ タップ位置を偏らせない、④ サンプル周期は 32 サイクル以上、の 4 つである。上記 4 つの条件を満たすことにより、正岡が提案した LFSR よりも理論上 100 倍以上の速度で乱数生成をしつつ実装コストを抑えることができる。

今回の検証はシミュレーションのみで行ったため実装評価を行っていない。信頼性を高めるために、実装評価が必要である。また、乱数テストに用いるテストスイートとして、DIEHARD 検定よりも高度である NIST 800-22 検定を採用し評価を行うべきである。

発表する際の課程を記入　　電気・電子情報工学　課程　　発表番号　16

（学籍が他課程所属の学生も発表する課程を記入すること）