令和2年　1月　8日

電気・電子情報工学専攻	学籍番号	M153269
申請者氏名	正岡 秀崇	

指導教員氏名	市川 周一

論　文　要　旨(修士)

論文題目	マイクロプロセッサの内部状態とタイミング揺らぎを利用した乱数生成手法

多くのセキュリティ技術では安全性の根拠として乱数が使用される．Unpredictable Random Number Generator (URNG) は実質的に予測できない乱数生成器である．URNG は主に CPU (Central Processing Unit) の内部状態を用いて乱数生成を行う．Suciu ら (2011) は CPU のパフォーマンスカウンタ (PFC) をエントロピ源とした URNG を提案した．一方, Suciu ら (2011) を含む URNG の先行研究では, CPU, および OS (Operating System) によるタイミング揺らぎによる乱数について定量的な評価を行っていない．

本研究ではタイミング揺らぎに基づいた乱数取得を目的とする．まず, Marton ら (2017) の研究を元に, PFC 値をエントロピ源としてソフトウェアで乱数取得した．実験には Intel 社の Core-i5, Ubuntu18, および PAPI (Performance Application Programming Interface) を用いた．タイミング揺らぎについて, Linux OS のシステムコール nanosleep 関数, および乱数取得プログラムとの並列実行アプリケーションを検討した．評価基準として, DIEHARD 試験による乱数品質の評価, 乱数生成レート, 割り込み数, およびコンテキストスイッチ回数を用いた．

他のアプリケーションを実行しないとき, nanosleep 関数を for ループにて実行後, 5 つの PFC (PAPI_BR_PRC, PAPI_BR_UCN, PAPI_LD_INS, PAPI_LST_INS, PAPI_TOT_CYC) の下位 8 bit 同士の排他的論理和 (Exclusive OR : XOR) を取って乱数とした．nanosleep 関数における停止時間を 1 ns ～ 10 ms の間で変化させた．この時, 実験の組合せ 8 通りですべて DIEHARD 試験に不合格だった．一方で 1 ms 以上の停止時間で乱数品質が向上した．この時, 乱数生成レートは減少するが, 割り込み数およびコンテキストスイッチ回数は増加した．上記の nanosleep 関数を用いた乱数取得プログラムに, HimenoBMT, IOzone, および STREAM それぞれを並列実行した．並列実行した場合では, nanosleep 関数による遅延時間, および並列アプリケーションに関係なく DIEHARD 試験に合格した．また, STREAM との並列実行について, XOR を用いずに, nanosleep 関数実行後に 1 つの PFC 下位 8 bit を直接取得した．PAPI_TOT_CYC について, 停止時間 10 ns 以上で DIEHARD 試験に合格した．PAPI_TOT_CYC はサイクル数が対象のため常にカウントされている．そのため, タイミング揺らぎを用いた PAPI_TOT_CYC からの乱数取得では, 単にサンプリングのタイミングをずらしたことで乱雑性が生じたと考えられる．

以上の検討結果より, LFSR (Linear Feedback Shift Register) の回路を追加した MCU (Micro Controller Unit) の PULPino を検討した．OS の FreeRTOS を用いて 128 bit LFSR 回路の下位 32 bit を乱数として取得する．この場合, STREAM を並列に実行した場合もしない場合も DIEHARD 試験に合格した．