令和元（２０１９）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： B183239，氏名： 高橋 諒太
工学分野名： 情報通信システム	指導教員名： 市川 周一

題　　目：和

MIPS プロセッサにおける Count レジスタを用いた URNG の検討

（英　　Examination of URNG using Count-register in MIPS processor　　）

Abstract

Unpredictable Random Number Generators (URNG) use internal state of a computer to generate random numbers. Marton et al. proposed a URNG which utilizes performance counter as an entropy source in Intel processor. Juna investigated the URNG with RISC-V processor for embedded device. The purpose of this study is to implement a URNG in MIPS processor which has been provided as open source. An entropy source of this URNG was the Count-register. When the Count-register values are simply harvested, the derived sequence passed only 1 test out of 18 for DIEHARD test. To improve randomness, the Count-register value were post-processed with MD5 and SHA3. After the post-process, the derived sequence passed almost all test when the lower 32 bits of Count-register value were used.

概　　要

暗号化には秘密鍵の生成が必要であり，その際には予測不可能な乱数が求められる．Unpredictable Random Number Generators(URNG)はコンピューターの内部状態に基づいて乱数を生成する手法であり，その値は実質的に予測不可能であるとされている．Marton ら(2017)は LinuxOS が動作する Intel プロセッサ上で，プロセッサ内部のパフォーマンスカウンタをエントロピー源とする URNG を提案した．いくつかのイベントセットについて検討し，単純な後処理を施すことで乱数評価テストにおいて高い合格率を示した．重名ら(2019)は組み込みデバイス向けである RISC-V プロセッサのパフォーマンスカウンタを用いた URNG を提案し，乱数生成命令として FPGA に実装した．

本研究ではプロセッサとして MIPS アーキテクチャの microAptivC コアを使用した．microAptivC は WaveComputing によって 2019 年にオープンソース化され注目されている．プロセッサの内部状態として Count レジスタを用いた．Count レジスタは CPU クロック毎に加算し起動からの合計サイクル数を確認することできるため，値を取得するタイミングによってランダム性が生じる．

まず合計サイクル数をエントロピー源とした乱数生成が可能であるか確認のため，LinuxOS が動作する PC 環境において Count レジスタ相当のパフォーマンスカウンタを取得し，DIEHARD テストを用いて評価を行った．合計サイクル数の取得プログラムを単独で実行する場合，また姫野ベンチ，IOzone，STREAM を並列実行する場合についてそれぞれ測定を行った．合計サイクル数を単純に並べた場合，18 項目あるテストで 1 項目のみ PASS となった．後処理として暗号学的ハッシュ関数である MD5 と SHA3 について検討した.合計サイクル数の下位 16 ビットを入力とした場合，MD5 では平均 8.75 項目，SHA3 では平均 9.75 項目で PASS および WEAK となった．一方で下位 32 ビットを入力とした場合は 18 項目全てに PASS した．また,バックグラウンドで並列実行するプログラムによる差は見られなかった．

次に microAptivC による実験をシミュレーター上で行った．Count レジスタの値を後処理無しで単純に並べた場合，PC 環境による実験と同様に 1 項目のみ PASS した．後処理を加えた結果，MD5 および SHA3 による後処理では 32 ビットを入力とした場合にほぼ全ての項目で PASS した．

本研究の結果から，プロセッサの合計サイクル数の下位 32 ビットを暗号学的ハッシュ関数の入力として後処理することで質の良い乱数を生成できると言える．今後の課題として，microAptivC 環境で測定出来なかった IOzone を並列実行した場合の乱数評価を行うこと，また FPGA に実装した場合の実機による乱数評価を行うことなどがある．

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　49

（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）