2019 年 01 月 08 日

電気・電子情報工学専攻	学籍番号	M153230
申請者氏名	重名 英史	

指導教員氏名	市川 周一 藤枝 直輝

論　文　要　旨（修士）

論文題目	プロセッサの内部状態を用いた乱数生成命令

IoT (Internet of Things) への関心の高まりとともに, 近年, 暗号通信における秘密鍵の生成など乱数の重要性が高まっている. Unpredictable Random Number Generator (URNG) は, 実質的に特定不可能なコンピュータの内部状態をエントロピ源とした乱数生成器であり, 単純な実装で高品質な乱数の取得が可能である. Suciu ら (2011) は, Linux OS が動作する一般的なパソコンの環境において, プロセッサ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，IoT の分野では Micro Controller Unit (MCU) などの小型プロセッサがしばしば用いられる.

本研究では, MCU の内部状態から乱数を生成し, 乱数生成命令として実装することを目的とした. 利用可能な内部状態として PFC の値のほかにもパイプライン中の信号の値が考えられるが, 本研究では初期検討として PFC の値のみを用いて乱数の生成を行った. 評価基準として, DIEHARD 乱数テストの結果, 生成ビットレート ([bit/s]) , 命令実装に伴うハードウェア使用量の増加量を用いた. MCU としてパフォーマンスカウンタを搭載したソフトプロセッサである PULPino を用い, OS として freeRTOS を動作させた. freeRTOS 上で動作させるアプリケーションとして CHStone に含まれる AES, DFADD, DFMUL, DFDIV, GSM, MIPS, MOTION の 7 種を用いた. 自身同士を除くペア計 21 種を実行し, 実行中の PFC の値を乱数として抽出し評価した. 乱数の抽出は ZedBoard 上の PULPino の実機で行った.

Suciu ら (2011) の手法と同様に PFC 値をそのまま連結して乱数とした場合, 全てのテストにおいて不合格となった. これは一般的なパソコンの環境ほど MCU の環境が複雑性をもたないことが原因と考えられる. そのため, 後処理として PFC 値同士の排他的論理和 (XOR) 演算を freeRTOS 上でソフトフェア処理として行い, 生成された数列をそれぞれ連結し 44bit の数列として取得することを提案した. また，同等の処理をハードウェアの乱数生成命令として実装する時には，44bit の乱数列を 32bit の乱数列として取得する命令とした.

提案手法のソフトウェア処理において, アプリケーションの組み合わせ 21 種のうち 8 種で全てのテストで合格した. ハードウェア処理では, 21 種中 4 種で合格した. また, AES を含む組合せでは合格となるテストの数が増加した. どのようなアプリケーションの組み合わせでも合格するためのさらなる後処理の検討が今後の課題である. 全てのテストで合格した組み合わせの生成レートは, ソフトウェア処理で 5.2 ～22.6 [kbit/s] , ハードウェア処理で 4.5～7.9 [kbit/s] であった. この生成レートの値は先行研究の目標値 150 [kByte/s] に対して最大約 400 分の 1 である. PULPino の動作周波数が先行研究で使用しているものの約 40 分の 1 であることを考慮しても, 改善が必要であり今後の課題である.. 命令実装に際し増加した FPGA のハードウェアリソース量は, LUT で 200 個, FlipFlop で 168 個, DSP48 は増加せず, BUFG は 3 個であった. これは命令実装をしない場合の使用リソース量に対して LUT と FlipFlop で約 2%, BUFG で 200% の増加率であった. BUFG の削減は今後の課題である.