DATE: 2019/01/08

Department of Electrical and Electronic Information Engineering		ID	M153230
Name	Hidefumi Juna		

Supervisor	Shuichi Ichikawa Naoki Fujieda

Abstract

Title	Random Number Generation Instruction which utilizes internal state of processor

As the interest in IoT (Internet of Things) has increased in recent years, the importance of random numbers has been increasing, which are used in, for example, secret key generation in cryptographic communication. Unpredictable Random Number Generator (URNG) is a random number generator whose entropysources are internal states of a computer that cannot be substantially identified. It has an advantage of acquiring high-quality random numbers with a simple implementation. Suciu et al. (2011) proposed a URNG with processor performance counters (PFCs) as entropy source in a general PC environment running Linux OS. On the other hand, in the field of IoT, small processors such as Micro Controller Unit (MCU) are often used.

The purposes of this research are to generate random numbers from internal states of an MCU and to implement it as a random number generation instruction. In addition to the values of PFCs, the values of signals in a pipeline can be considered as internal states. In this study, however, random numbers were generated using only the value of PFCs as an preliminary study. Evaluation criteria are the result of the DIEHARD statistical test, the generation rate of random numbers in bit/s, and the increment of the amount of hardware by instruction implementation. PULPino and freeRTOS are selected as an MCU and an OS, respectively, because PULPino has a soft processor with PFCs. Two out of seven applications in the CHStone benchmark, AES, DFADD, DFMUL, DFDIV, GSM, MIPS, and MOTION, are executed on FreeRTOS at the same time. Another task to generate random numbers from PFC values is added to them. Random numbers from each of 21 pairs of applications are obtained in PULPino on a ZedBoard.

Random numbers generated in a similar way to the previous work, which concatenate parts of PFC values, failed in the all tests. This is because an MCU environment is not as complex as a general PC environment. Instead, an exclusive OR (XOR) operation among parts of PFC values is performed as postprocessng to obtain a random number of 44 bits. It is implemented both in software and hardware. In the hardware implementation, an additional buffer is required in order to fit in a 32-bit interface. In the software processing implementation, 8 out of 21 pairs of applications passed all the tests. In the hardware processing, 4 out of 21 of applications passed all the tests. Further examination of post-processing for passing any combination of applications is left as a future work. The generation rate of the combination passed for all the tests was 5.2 – 22.6 [kbit/s] in the software processing and 4.5 – 7.9 [kbit/s] in the hardware processing. The increase of the number of hardware elements was 200 in LUTs, 168 in the Flip-Flops, and 3 in BUFGs. DSP48 was not increased. This mainly comes from the XOR operation and the buffer for implementing instructions. The reduction of BUFG is left as a future work.