Programmable Delay Logic을 이용한 링 오실레이터 기반의 실 난수 생성기 구현

양희훈,박지호,이상원,*유호영
충남대학교 전자공학과
e-mail : *hhyang.cas@gmail.com, jhpark.cas@gmail.com*
swlee.cas@gmail.com, hyyoo@cnu.ac.kr

Implementation of Ring-Oscillator based TRNG using Programmable Delay Logic

Heehun Yang, Jiho Park, Sangwon Lee, *Hoyoung Yoo
Department of Electronics Engineering
Chungnam National University

Abstract

True random number generator (TRNG) is an important component widely used not only in cryptographic system but also in various other industries such as game, machine learning, and statistical analysis. This paper propose the ring-oscillator based TRNG that uses random jitter as an entropy source in an FPGA environment. By using programmable delay logic (PDL), different jitters are generated for each ring-oscillator and combined using XOR gates. After passing through the first XOR gate, the output is synchronized with the clock. The proposed structure is implemented using an AMD Xilinx Artix-7 chip and the performance is evaluated using the NIST SP 800-22 test suite. The results show that the proposed TRNG structure passed all tests and achieved high performance, particularly in the entropy category.

I. 서론

RNG(Random Number Generator)는 보안 산업 뿐만아니라 여러 분야에서 폭넓게 사용되는 난수 생성기이다. 암호화 key 생성, PRNG(Pseudo Random Number Generator)의 시드(seed) 값 생성 등에서 활용될 뿐 아니라, 게임, 통계분석, 머신러닝의 무작위 학습 등 여러 분야에서 필수적인 요소이다. 난수 생성기는 PRNG와 TRNG(True Random number Generator)로 구분된다. PRNG는 수학적 알고리즘을 기반으로 난수를 생성하기 때문에 길이가 긴 난수 비트를 생성할 수 없다. 요구되는 비트 수가 길어질수록 많은 양의 하드웨어 리소스를 필요로 하며, 반복되지 않는 많은 수의 난수를 생성하는 데 한계가 존재한다. 산업의 발전과 함께 각 시스템에서 요구하는 난수의 비트 길이가 증가함에 따라, 하드웨어 리소스가 제한된 환경에서 PRNG로는 충분한 난수를 생성하는 데 어려움이 있다. 반면, TRNG는 entropy source를 사용하여 난수를 생성하기 때문에, 일정한 양의 하드웨어 리소스로 비트 길이에 제한이 없는 난수를 생성할 수 있다.

FPGA환경에서의 TRNG 하드웨어 개발은 ASIC(Application Specific Integrated Circuit)에 비해 개발시간의 단축하고 개발비를 줄일 수 있는 이점이 있기 때문에 활발하게 진행되고 있다. FPGA에서 적용되는 TRNG는 metastability[1], 클럭 jitter [2-4], chaos[5] 등을 entropy source로 사용하여 연구되고 있다.

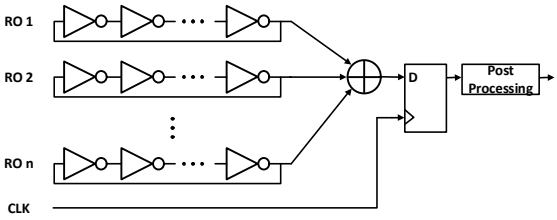

그림 1. original Ring-Oscillator TRNG[2]

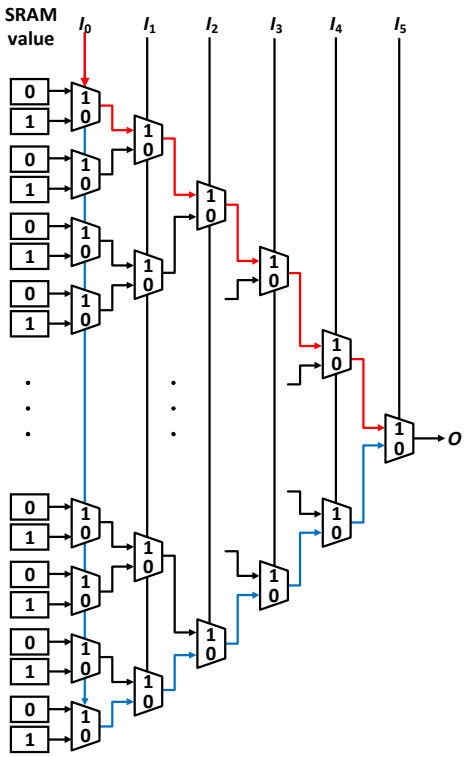

그림2. 6입력 LUT에서의 인버터 구조

본 논문에서는 clock jitter를 entropy source로 사용하는 TRNG로 널리 사용되는 Ring-Oscillator(RO) 기반의 TRNG를 이용하며, RO에서 생성된 jitter 영역을 PDL로 조절한 후 XOR게이트로 중첩하는 구조를 제안한다.

II. Ring-Oscillator based TRNG

2.1 Ring-Oscillator based TRNG

RO기반의 TRNG는 홀 수 개의 인버터를 사용한 RO에서 발생하는 jitter를 entropy source로 사용한다. 그림 1은 일반적인 구조의 RO 기반의 TRNG를 나타낸다[2]. 독립적으로 작동하는 RO에서 생성된 jitter는 가우시안 분포를 따라 각 clock

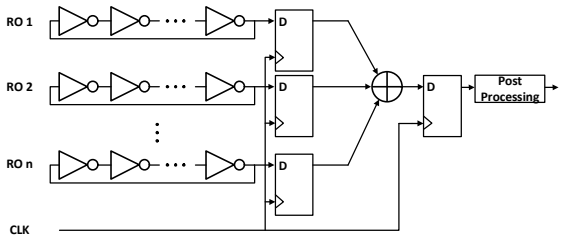

그림 3. 클럭동기화 된 Ring-Oscillator TRNG[3]

transition에서 논리적 0 또는 1상태를 가지게 된다. 이러한 jitter는 샘플링 주기에서 clock transition이 서로 다른 시간에 나타나도록 만든다. XOR gate는 이러한 transition 영역을 중첩하여 출력하며, 최종 D flip-flop에서 샘플링 되어 난수를 생성한다

2.2 Programmable Delay Logic(PDL)

FPGA에서 논리 소자는 Look-Up Table(LUT)를 이용하여 구성된다. 본 논문에서 사용한 Artix-7 chip의 FPGA는 6-입력 LUT로 구성되어 있으며, LUT의 내부는 MUX가 트리 구조로 연결되어 있어 입력 신호에 따라 SRAM에 저장된 값을 선택하여 출력한다.

그림 2는 하나의 6-입력 LUT에서 {I_5, I_4, I_3, I_2, I_1, I_0}이 6'b111111인 경우와 6'b000000인 경우 SRAM의 값이 출력되는 경로를 각각 빨간색과 파란색으로 표현한 인버터의 구조이다. 두 경우 입력 신호에 따라 활성화되는 MUX가 달라져 내부 딜레이 또한 달라진다. 이때, 입력에 따라 달라지는 내부 딜레이를 조정하는 신호를 PDL이라고 하며, 이 PDL을 조정해 RO의 출력 딜레이를 변화시킬 수 있다. [3]의 연구에서는 PDL을 이용한 TRNG를 구성 후, 매 클럭 마다 PDL을 변화시켜 restart 테스트를 통해 각 시행에서 출력들의 상관계수가 0에 가까움을 보여준다.

Ⅲ. Proposed Design

여러 개의 RO의 동작으로 발생하는 출력은 XOR을 거쳐 D flip-flop으로 샘플링 한다. 출력에서 발생하는 transition의 주기는 RO가 병렬적으로 늘어나면서 거치는 XOR의 수가 증가할수록 짧아진다. transition의 주기가 FPGA의 동작주파수보다 빨라질 경우 출력이 변화하는 상태를 샘플링 할 수 없게 되고 이는 랜덤성의 저하를 야기한다. 그림 3에서 나타낸 [4]에서 제안하는 구조는 이러한 문제점을 해결하기 위해 XOR에 입력되는 각 RO의 출력을 D flip-

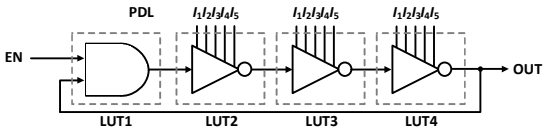

그림 4. PDL을 적용한 Ring-Oscillaotr

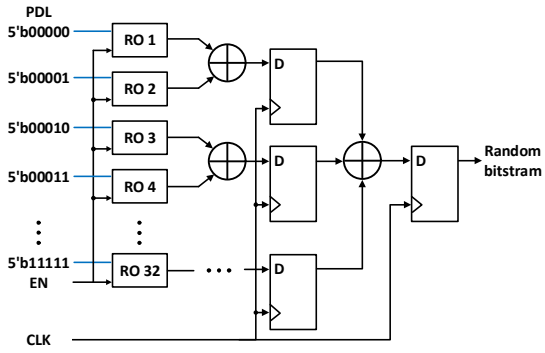

그림 5 제안하는 TRNG의 구조

flop으로 샘플링 하여 출력을 클럭에 동기화 시키는 방법을 사용한다. [4]에서의 구조는 Intel Cyclone II FPGA 환경에서 실험되었으며, 해당 환경에서의 최대 동작 주파수는 402.5MHz이다. 그러나 FPGA 제작 기술의 발전으로 인해 최대동작주파수가 증가하였으며, 이는 transition의 변화를 샘플링 할 수 있는 속도도 증가시켰다. 이에 따라 샘플링 할 수 있는 중첩된 출력의 RO개수 또한 늘어나게 되었다.

본 논문에서는 최대동작주파수의 증가로 가능해진 2개의 RO의 출력을 XOR한 후 샘플링 하는 구조를 제안한다. 그림 5에서는 제안하는 TRNG의 구조를 보여준다. 본 논문에서는 [4]의 구조와는 다르게 각 RO에 서로 다른 값을 입력하여 PDL에 변화를 줌으로써 서로 다른 jitter 영역을 생성하고, XOR 게이트로 중첩 시킨다. 이후 샘플링 하여 출력을 클럭과 동기화 시킨다.

본 논문의 실험에 사용된 Artix-7의 6입력 LUT를 이용한 PDL은 출력 논리값을 변화시키지 않는 조건에서 $\{I_1,I_2,I_3,I_4,I_5\}$=5'b00000부터 $\{I_1,I_2,I_3,I_4,I_5\}$=5'b11111까지 선택될 수 있다. 제안하는 구조에서는 32개의 RO의 PDL에 서로 다른 값을 입력한다. 그림 4에서는 PDL을 적용한 하나의 RO를 보여준다. 본 논문에서는 인버터 3개로 구성된 RO를 사용하였으며, 각 인버터에는 반전시키기 위한 값을 입력 I_0로 인가한다. 나머지 I_1부터 I_5의 입력은 PDL 입력으로 사용한다. 한 RO에 대해서는 각 인버터에 대하여 동일한 PDL값을 인가한다. 서로 다른 PDL은 각각 다른 위치에서 transition 영역을 가지게 되며, 이는

표1. PDL이 모두 0일 경우, NIST SP 800-22 test 결과

Statistical test	P-Value	Proportion
Frequency	FAIL	-
BlockFrequency	FAIL	-
CumulativeSums	FAIL	-
Runs	0.7399	0.94
LongestRun	0.7598	0.99
Rank	0.4559	0.99
FFT	0.4190	1.00
NonOverlappingTemplate	0.5189	0.95
OverlappingTemplate	0.2897	0.98
Universal	0.4012	1.00
ApproximateEntropy	0.2757	0.98
RandomExcursions	0.3859	0.93
RandomExcursionVariant	0.3213	0.93
Serial	0.6900	1.00
LinearComplexity	0.8514	0.99

표2. PDL을 모두 다르게 한 경우(0~31), NIST SP 800-22 test 결과

Statistical test	P-Value	Proportion
Frequency	0.3669	0.99
BlockFrequency	0.4012	0.99
CumulativeSums	0.7681	0.99
Runs	0.6163	1.00
LongestRun	0.2757	0.99
Rank	0.3505	1.00
FFT	0.9643	0.99
NonOverlappingTemplate	0.5017	0.97
OverlappingTemplate	0.9915	0.99
Universal	0.1223	1.00
ApproximateEntropy	0.8677	1.00
RandomExcursions	0.3840	0.98
RandomExcursionVariant	0.5148	0.98
Serial	0.7937	0.97
LinearComplexity	0.0805	0.99

XOR gate를 이용하여 중첩한 후 D flip-flop으로 샘플링 한다. 샘플링 된 값은 후처리과정을 거치지 않고 출력된다.

Ⅳ. 실험결과

제안하는 구조는 최대동작 주파수가 1.2GHz인 AMD Xilinx사의 Artix-7 chip을 이용하여 구현하였으며, 동작주파수는 50MHz로 설정했다. 실험에서 TRNG의 성능평가를 위해 NIST에서 제공하는 800-22 test suit가 사용되었다. NIST SP

800-22는 TRNG의 성능을 평가하기위해 보편적으로 사용되는 test suit이며, 총 15가지의 test를 포함한다[6]. test를 위한 출력 bit는 재시작 하지 않고 100,000,000 bits를 생성한다. test는 1,000,000의 bitstream으로 100회 시행되었으며, 결과는 표1에 나타난다. 제안하는 구조에서 모든 PDL을 조정하지 않고 0으로 고정했을 경우, 몇 가지의 test에서 통과하지 못했다. 하지만 서로 다른 PDL 값을 입력 후 test를 진행한 결과 모든 test를 통과하였다. 각 실험은 후처리과정을 수행하지 않고 진행되었다. 표 1,2에 나타나는 CumulativeSums, NonOverlappingTemplate, RandomExcursions, RandomExcursionVariant test의 경우 여러 번 진행되므로 p-value는 평균값으로 작성하였으며, proportion은 최소값으로 나타냈다.

V. 결론

본 논문에서는 동작주파수의 증가로 가능해진 2개의 RO출력을 XOR한 후 D flip-flop으로 클럭 동기화 하는 회로를 제안한다. RO 32개로 제안하는 회로를 구성하였으며, 각 RO에 사용되는 인버터는 3개로 고정하였다. 같은 RO 내에서의 인버터의 PDL에는 모두 같은 값을 인가하였다. 각 RO끼리는 서로 다른 PDL 입력 값으로 jitter의 딜레이를 변화시킨 후 XOR로 중첩 시켜 랜덤성을 향상시킨다. 제안한 구조는 AMD Xilinx사의 Artix-7 chip을 이용하여 구현되었으며, 최대동작 주파수는 1.2GHz이다. 회로의 동작 주파수는 50MHz로 설정하였으며, 재시작 하지 않고 100,000,000bits를 생성하였다. 평가는 NIST SP 800-22 test suit를 이용하여 진행하였다. 실험결과 각 RO마다 추가적인 D flip-flop을 추가해 XOR의 입력을 클럭 동기화 시키는 [4]와 매 클럭마다 PDL 입력 값을 변화시켜 모든 RO에 동시에 입력시키는 TRNG 구조를 제안한 [3]에 비해 절반의 D flip-flop으로 후처리과정 없이 고성능의 TRNG를 구현하였으며, NIST SP 800-22 test suit의 approximate entropy 항목에서 높은 결과가 나타났다.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A5A8026986), and supported by National R&D Program through the National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT(2020M3H2A1078119)

참고문헌

[1] M.Majzoobi, F.Koushanfar, and S.Devadas, “FPGA-based true random number generation using circuit metastability with adaptive feedback control,” In Cryptographic Hardware and Embedded Systems-CHES pp. 17-32, Sep. 2011

[2] B. Sunar, W. J. Martin, and D. R. Stinson, "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks," in IEEE Transactions on Computers, vol. 56, no. 1, pp. 109-119, Jan. 2007.

[3] N. Nalla Anandakumar, S. K. Sanadhya, and M. S. Hashmi, "FPGA-Based True Random Number Generation Using Programmable Delays in Oscillator-Rings," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 3, pp. 570-574, March 2020.

[4] K.Wold and C.H.Tan, “Analysis and enhancement of random number generator in FPGA based on oscillator rings,” in International Journal of Reconfigurable Computing, pp. 1-8, Feb. 2009.

[5] A. Beirami and H. Nejati, "A Framework for Investigating the Performance of Chaotic-Map Truly Random Number Generators," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no. 7, pp. 446-450, July 2013.

[6] A. L. Rukhin et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-22, Apr. 2010.