세그먼테이션 기법을 이용한 의사 난수 발생기★

전민정* · 김상춘* · 이제훈**

요 약

최근 스마트폰 및 태블릿 PC를 이용한 무선통신 사용자가 점차 늘면서 암호 알고리즘, 특히 스트림 암호 연구가 활발히 진행되고 있다. 스트림 암호 방식에서 필요한 난수발생기는 하드웨어 구현이 쉬운 LFSR 구조가 주로 사용된다. 그러나 기존의 다중 비트 출력의 LFSR 기반 난수 발생기는 회로가 복잡해지고 출력간의 상관관계가 크다. Leap-ahead 구조를 갖는 LFSR은 이를 해결하기 위해 제안되었으나, 레지스터의 수와 출력비트에 따라 생성되는 난수의 수가 급격히 적어지는 단점을 갖는다. 본 논문은 기존 Leap-ahead 구조에 세그먼테이션 기법을 적용하여 회로 크기의 증가 없이 생성되는 난수의 수를 높일 수 있는 새로운 구조를 제안한다. 제안된 구조는 VHDL을 통하여 회로로 합성된 후, Xilinx사의 Xilinx ISE 10.1의 Virtex 4, XC4VLX15에서 동작을 검증하였다. 실험 결과 제안된 구조는 기존 Multi-LFSR 구조에 비해 20%이내의 회로 크기로 Leap-Ahead 구조에 비해 최소 40% 생성되는 난수의 수를 증가시켰다.

A Pseudo-Random Number Generator based on Segmentation Technique

Min-Jung Jeon* · Sang-Choon Kim* · and Je-Hoon Lee**

ABSTRACT

Recently, the research for cryptographic algorithm, in particular, a stream cipher has been actively conducted for wireless devices as growing use of wireless devices such as smartphone and tablet. LFSR based random number generator is widely used in stream cipher since it has simple architecture and it operates very fast. However, the conventional multi-LFSR RNG (random number generator) suffers from its hardware complexity as well as very closed correlation between the numbers generated. A leap-ahead LFSR was presented to solve these problems. However, it has another disadvantage that the maximum period of the generated random numbers are significantly decreased according to the relationship between the number of the stages of the LFSR and the number of the output bits of the RNG. This paper presents new leap-ahead LFSR architecture to prevent this decrease in the maximum period by applying segmentation technique to the conventional leap-ahead LFSR. The proposed architecture is implemented using VHDL and it is simulated in FPGA using Xilinx ISE 10.1, with a device Virtex 4, XC4VLX15. From the simulation results, the proposed architecture has only 20% hardware complexity but it can increases the maximum period of the generated random numbers by 40% compared to the conventional Leap-ahead archtecture.

Key words : LFSR, Galois, Pseudo-random number generator, Segment

접수일(2012년 8월 30일), 수정일(1차: 2012년 9월 6일), 게재확정일(2012년 9월 7일)

★ 본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임 (2012H1B8A2026055).

* 강원대학교 전자정보통신공학과

** 강원대학교 전자정보통신공학과 (교신저자)

1. 서 론

최근 스마트폰과 태블릿과 같은 개인용 무선 통신 기기들이 급격히 확산되면서, 개인 정보 보안에 대한 관심이 크게 증가되고 있다 [1]. 정보보호를 위해 기존 대칭키, 비대칭키, 그리고 스트림 암호와 같은 다양한 암호 알고리즘 연구가 활발히 진행되고 있다. 특히, 스트림 암호는 선형 쉬프트 레지스터 (LFSR: linear feedback shift register)를 이용한 난수 발생기를 이용하여 평문을 한번에 1-비트 혹은 다중 비트로 암호화하는 방식으로, 블록 암호보다 빠르게 고속으로 암호화를 처리할 수 있다 [2-6]. 하드웨어 구조가 간단하고 저전력 소비를 요구하는 휴대 정보 기기와 USN용 어플리케이션에 적합하다 [2].

LFSR을 이용한 URNG(uniform random number generator)는 의사 난수 (pseudo-random number)를 생성한다. 이상적인 난수는 생성 방법이 결정되어 있지 않고, 다음에 생성될 난수값이 전혀 예측할 수 없다. 그러나, LFSR을 이용한 URNG는 균일한 출력 분포를 갖는 의사난수열을 생성한다. 따라서, 난수발생기 회로 구조와 LFSR에 입력되는 seed 값을 알면 난수열을 예측할 수 있다. LFSR의 스테이지 수를 증가시켜 생성된 난수열이 반복되는 전체 주기를 증가시키고, 출력되는 난수들간의 상관관계를 줄여 정보를 효율적으로 보호할 수 있다 [7].

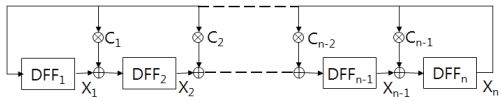
근래에는 무선 인터넷 서비스에서 스트림 기법이 많이 이용되면서 블록 암호알고리즘보다 고속 동작이 가능한 소프트웨어기반의 스트림 암호 개발이 많아지고 있다. LFSR 자체만으로는 안전성을 제공하지 못하므로, 높은 주기성과 좋은 통계적 성질을 LFSR에 결합하여 우수한 암호 알고리즘이 설계된다. 또한 LFSR에 의해 발생된 수열은 큰 주기 및 좋은 통계성을 갖지만, 출력 수열로부터 쉽게 예측이 가능하다. 일반적인 LFSR은 단지 한 사이클에 하나의 난수만을 생성하나 최근 대부분의 어플리케이션이 복잡해지면서 다중 비트의 난수가 요구되다. 이를 위해 다중 LFSR 구조가 제안되었으나, 회로가 복잡하다는 단점을 갖는다. 일례로, 32 비트 출력을 위해서는 32개의 다른 LFSR이 요구된다.

이와 같은 단점을 보완하기 위해 Leap-ahead 구조를 갖는 LFSR이 제안되었다.[8] 단지 하나의 LFSR을 이용하여 다중 비트 출력을 얻을 수 있어 회로 크기는 감소되나 LFSR의 크기와 출력 단계 수의 관계에 따라 생성되는 난수들의 주기가 크게 감소한다. Leap-ahead 구조는 2^n-1이 m으로 나누어지지 않을 때, 생성되는 난수 주기가 2^n-1로 가장 크게 되지만, 그 외의 경우에는 최대 주기가 크게 감소되어, 쉽게 다음 난수를 예측할 수 있다는 단점을 갖는다.

본 논문은 2장에서 기존 Leap-ahead 구조에 대해서 설명하고, 3장에서 본 논문에서 제안한 세그먼테이션 기법을 적용한 새로운 Leap-ahead 구조를 갖는 LFSR에 대해 설명한다. 4장에서는 제안된 LFSR 기반 URNG를 VHDL을 이용하여 합성하고, FPGA에서 구현한 후 성능 평가를 수행한다. 마지막으로, 5장에서 결론을 맺는다.

2. Leap-ahead 구조

Leap-ahead 구조는 그림 1에 나타낸 것처럼 피드백 구조를 갖는 LFSR과 XOR 연산기로 구성되어 있다. Leap-ahead 구조를 갖는 갈로이스(Galois)혹은 피보나치(Fibonacci) 형태의 LFSR이 주로 사용되며, 본 논문에서는 갈로이스 타입의 LFSR을 이용한 URNG를 제안하였다. 그림 1에 나타낸 것처럼, 시드(seed)값이 LFSR 레지스터의 초기값으로 입력된 후 매 클럭마다 난수값을 생성한다. DFF_i는 쉬프트 레지스터이며 이의 출력은 X_i가 된다. C_1에서 C_{n-1}은 Tap이라 부른다.



(그림 1) 갈로이스 LFSR

갈로이스 형태의 LFSR 구조에서 모든 쉬프트 레지스터의 수식은 (1)에 설명되어있다.

$$X(t+1) = AX(t) \quad (1)$$

식 (1)에서 X(t)는 현재 시간, t의 모든 레지스터들

의 출력값이고, X(t+1)는 다음 클럭 사이클에서의 레지스터 출력값을 의미한다. A는 변환행렬로 현재 출력과 다음 출력간의 관계식으로 얻어진다. 식 (1)에서 생성된 임의의 수열의 값이 다음 클럭 사이클에 쉬프트되어 나오기 때문에 매우 가까운 상관관계를 가지고 있다. 따라서, 수식 (1)에서 m 클럭 사이클 지연을 가질 때의 레지스터 출력 X(t+m)을 식 (2)와 같이 구할 수 있다.

$$\begin{aligned} X(t+m) &= AX(t+m-1) \\ &= A(AX(t+m-2) \\ &= A^m X(t) \end{aligned} \tag{2}$$

식 (2)를 이용하여 변환 행렬 A^m을 회로로 구현함으로써, 갈로이스 형태 LFSR에서 생성되는 난수열을 한 클럭마다 주기적으로 생성시키는 대신 m-번째 출력을 하나의 클럭 사이클에 출력할 수 있다. 식 (3)은 식 (1)의 변환 행렬, A에 해당하며 이를 연속적으로 m번 곱하여 식 (4)과 같이 A^m 행렬의 일반화된 식을 얻을 수 있다.

$$A_{Galois} = \begin{pmatrix} 0_{1\times(n-1)} & C_{n\times 1} \\ 1_{(n-1)\times(n-1)} & \end{pmatrix}_{n\times n} \tag{3}$$

$$A^m = \begin{pmatrix} 0_{m(n-m)} & C_{n\times 1}\, AC_{n\times 1} \cdots A^{m-1}C_{n\times 1} \\ I_{(n-m)(n-m)} & \end{pmatrix}_{n\times n} \tag{4}$$

Galois 구조에서 Tap은 식 (5) 및 식(6)의 관계식으로 구현되며, 이를 이용하여 그림 1의 탭을 결정한다. Tap은 피드백 계수(feedback coefficient)로서 0 또는 1의 값을 가진다. 0인 경우는 스위치처럼 열린 경우이고, 1인 경우 스위치처럼 닫힌 경우에 해당된다.

$$C_{n-1} = C_{n-2} = \cdots = C_{n-(m-1)} = 0 \tag{5}$$

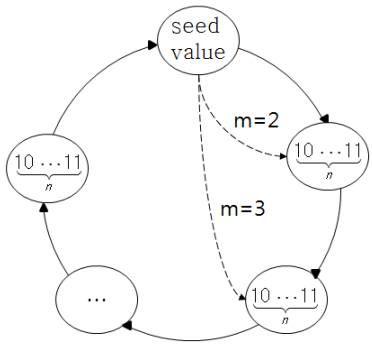
$$C_{n\times 1} = (C_1\, C_2 C_3 \bullet \bullet \bullet\, C_{n-1}\, 1) \tag{6}$$

난수 출력값 X(t)와 X(t+1)은 높은 상관관계를 갖기 때문에 쉽게 예측할 수 있으나, 현재 출력 X(t)와 m-번째 출력 X(t+m)간에는 상관관계가 낮다. 또한 식 (2)를 이용하여 하나의 클럭에 m-번째 출력, X(t+m)을 계산할 수 있어 암호 회로에 적합하다. 특히, 다중 비트 X_1부터 X_n까지 생성된 단일 비트 스트림(one-bit stream)은 LFSR의 초기 입력값, seed에 의해 변화하며 최대 2^n-1의 주기를 갖는다. 따라서, 생성된 난수는 의사 난수(p-seudo random numbers)이며 단일 비트 스트림이 출력 가능한 난수이기 때문에, 생성 가능한 난수열, 단일 비트 스트림이 2^n-1의 주기를 갖도록 seed값을 주의깊게 선택해야 한다. 또한 갈로이스 타입의 LFSR에 의해 생성되는 난수들의 주기, T는 식 (7)과 같이 구할 수 있다.

$$T = \frac{[2^n - 1, m]}{m} \tag{7}$$

식 (7)에서 n은 LFSR의 스테이지 수이며, m은 URNG (uniform random number generator)의 출력 단계의 수를 나타낸다. [2^n-1,m]은 2^n-1과 m의 최소공배수를 나타내며, 따라서 2^n-1과 m이 나누어지지 않을 때 T는 2^n-1의 최대주기를 얻을 수 있다. 2^n-1과 m이 나누어지면 생성된 난수들의 주기가 크게 감소된다.

(그림 2)는 Leap-ahead LFSR의 출력된 단일 비트 스트림 출력을 나타낸다. Leap-ahead 구조는 n개의 레지스터를 가지며, 한 사이클마다 m-번째 출력값을 난수로 출력한다. LFSR의 초기화된 후 레지스터에 seed 값이 입력된 후, seed값에 따라 각 클럭 사이클별로 n bit의 출력값이 레지스터들에게 피드백되어 입력된다. 식 (7)에 보여지듯이 Leap-ahead 구조가 m이 1의 값을 가질 경우 단일 비트 스트림 출력을 갖게 되고, 생성되는 난수의 주기는 2^n-1이 된다. 또한, m이 2인 경우, 각 클럭 사이클마다 단일 비트 스트림에서 2-번째 값을 난수로 출력한다. 이 경우, 2^n-1이 m으로 나누어지 않을 경우 생성되는 난수의 반복 주기는 2^n-1이 된다. 반면에 2^n-1이 m으로 나누어지는 경우, 주기가 2^n-1을 2^n-1과 m의 최대공약수로 나눈 수만큼의 주기로 짧아진다. 이 경우 출력되는 난수들 간의 상관관계가 적어져 높은 암호 효율을 얻을 수 있지만, 최대 주기가 짧아져 의사 난수열의 예측이 쉽게 가능해지는 단점을 갖는다.



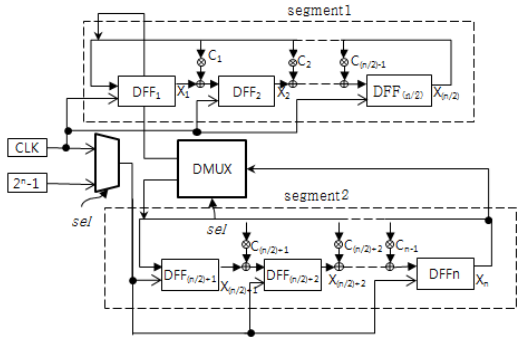
(그림 2) Leap-ahead 구조의 주기

3. 세그먼테이션 기법을 이용한 Leap-ahead URNG

Leap-ahead LFSR 구조는 일반적인 다중 LFSR 구조에 비해 작은 크기의 회로로 여러 비트의 난수를 발생시킬 수 있는 장점을 가진 반면 LFSR 스테이지 수와 출력 단계 수에 따라 생성되는 난수들의 주기가 크게 감소한다는 단점을 갖는다. 본 논문에서는 단점을 해결하기 위하여 Leap-ahead LFSR에 세그먼테이션 기법을 적용한 새로운 구조를 제안한다.

제안된 회로는 (그림3)과 같이 하나의 Leap-ahead LFSR 구조를 두 개의 세그먼트로 구분하였다. 제안된 세그먼테이션 Leap-ahead LFSR은 2^n-1이 m으로 나누어지는 경우와 그렇지 않은 경우에 따라 두 개의 MUX의 제어 신호를 입력한다. 첫 번째로 2^n-1이 m으로 나누어지지 않을 경우 두 개의 세그먼트들에게 동일한 클럭 입력이 들어가고, 오른쪽 세그먼트, Segment#2의 마지막 레지스터, DFF_n의 출력은 왼쪽 세그먼트, Segment#1의 첫 번째 레지스터, DFF_1의 입력으로 피드백 되도록 MUX를 선택한다. 따라서, 기존 Leap-ahead 구조와 동일하게 구동된다. 두 번째로 2^n-1이 m으로 나누어지는 경우 두 개의 세그먼트들에게는 분리된 클럭 입력이 인가된다. 왼쪽 세그먼트에는 원래 클럭 입력이 인가되고, 오른쪽 세그먼트에는 에게 2^n-1이 m으로 나누어진 수로 분주된 클럭 입력을 인가한다. 또한 Segment#2의 마지막 레지스터, DFF_n의 출력은 Segment#1의 DFF_1으로 피드백 되는 대신 Segment#2의 첫 번째 레지스터인 $DFF_{(n/2)+1}$로 입력되도록 MUX 출력을 선택한다.

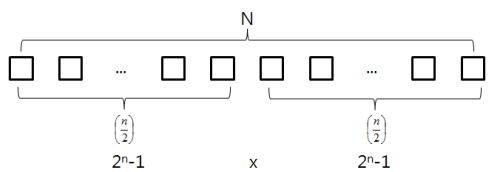
따라서, 제안된 세그먼테이션 Leap-ahead LFSR 구조는 두 개의 나누어진 세그먼트들은 2^n-1이 m으로 나누어지지 않는 경우 기존 Leap-ahead 구조와 동일하게 구동되며, 그렇지 않은 경우 두 개의 세그먼트로 나누어 구동된다. 이와 같은 방식으로 구동할 경우, 따라서, 기존 Leap-ahead 구조의 URNG의 생성된 난수의 최대 주기가 2^n-1을 2^n-1과 m의 최대공약수로 나눈 수만큼의 주기로 짧아지는 대신에, 두 개의 세그먼트의 최대 주기들의 곱으로 결정된다.



(그림 3) 세그먼테이션 기법을 적용한 Leap-ahead URNG

(그림 3)는 제안된 세그먼테이션 기법을 적용한 Leap-ahead 구조의 최대 난수 생성 주기를 나타낸다. 2^n-1이 m으로 나누어 질 때 식 (7)에 나타낸 것처럼 기존 Leap-ahead 구조와 동일한 주기를 갖는다. 그러나 2^n-1이 m으로 나누어 질 때 두 개의 세그먼트로 분할되어 구동되며, 최대 주기 역시 식 (8)과 같이 첫 번째 세그먼트의 최대 주기, T_1과 두 번째 세그먼트의 최대 주기, T_2의 곱으로 구할 수 있다.

$$T = T_1 \times T_2 \tag{8}$$



(그림 4) 세그먼테이션 Leap-ahead를 이용한 의사 난수 발생기의 생성되는 주기

제안된 세그먼테이션 Leap-ahead 구조는 두 개의 세그먼트로 나뉜다. 이 때, LFSR의 스테이지 수와 출력 단계의 수가 짝수 혹은 홀수인지에 따라 다음과 같이 4가지 경우로 나뉜다.

1st case : n과 m이 짝수일 때

$$\Rightarrow\left(\frac{n}{2},\frac{m}{2}\right),\left(\frac{n}{2},\frac{m}{2}\right)$$

2nd case : n이 짝수이고, m이 홀수일 때

$$\Rightarrow\left(\frac{n}{2},\left\lfloor\frac{m}{2}+1\right\rfloor\right),\left(\frac{n}{2},\left\lfloor\frac{m}{2}\right\rfloor\right)$$

3rd case : n이 홀수이고, m이 짝수일 때

$$\Rightarrow\left(\left\lfloor\frac{n}{2}+1\right\rfloor,\frac{m}{2}\right),\left(\left\lfloor\frac{n}{2}\right\rfloor,\frac{m}{2}\right)$$

4th case : n과 m이 홀수일 때

$$\Rightarrow\left(\left\lfloor\frac{n}{2}+1\right\rfloor,\left\lfloor\frac{m}{2}+1\right\rfloor\right),\left(\left\lfloor\frac{n}{2}\right\rfloor,\left\lfloor\frac{m}{2}\right\rfloor\right)$$

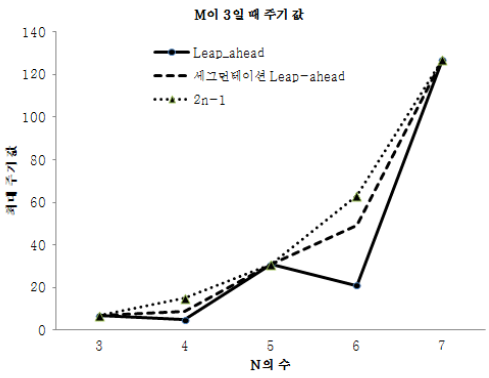
첫 번째 경우와 같이 n과 m이 모두 짝수일 경우 두 개의 세그먼트들은 n/2 그리고 m/2로 각각 동일한 수의 LFSR과 출력 비트수를 갖는다. 그러나, 두 번째 경우와 같이 출력 비트수 m이 홀수 이거나 세 번째 경우와 같이 레지스터의 개수가 홀수 일 경우 두 세그먼트는 LFSR의 스테이지 수와 출력 비트수가 서로 다른 길이를 갖는다. 그러나, 생성되는 난수의 최대 주기 T는 식 (8)에 나타낸 것처럼 구할 수 있다.

일례로 LFSR의 스테이지 수가 6이고 출력 비트수가 6인 경우 2^n-1은 63이 되고, m은 6으로 3의 최대 공약수가 발생한다. 따라서, 기존 Leap- ahead 구조의 생성되는 난수의 최대 주기는 21로 감소된다. 그러나, 제안된 구조에서는 두 개의 세그먼트들로 나누어지고 각각의 세그먼트들은 3개의 LFSR 스테이지를 갖고 3비트를 출력한다. 따라서, 첫 번째 세그먼트의 주기 T_1은 식 (7)에 나타낸 것처럼 $T_1=[2^3-1, 3]/3=[7, 3]/3$이 된다. 7과 3은 서로 나누어지지 않기 때문에 주기 T_1은 7이 된다. 또한 두 번째 세그먼트도 동일한 LFSR 스테이지 수와 출력 비트 수를 갖기 때문에 T_2는 7이 되고, 전체 주기 T는 T_1과 T_2의 곱 49가 된다. 따라서, 기존 Leap-ahead 구조에서 발생되는 난수의 최대 주기인 21보다 크게 증가한다.

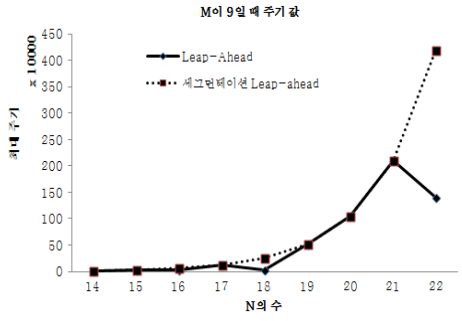
4. 실험 결과

본 논문에서 제안된 세그먼테이션 Leap-ahead LFSR 구조는 VHDL을 이용하여 회로로 합성하였고, Xilinx사의 Virtex 4, XC4VLX15 디바이스를 이용하여 동작을 검증하였다.

첫 번째 실험은 본 논문에서 제안된 Leap-ahead LFSR 기반 URNG에서 생성되는 난수열의 최대 주기를 기존 다중-LFSR 및 Leap-ahead LFSR 구조와 비교하였다. (그림 5)와 (그림 6)에 나타낸 것처럼, 다중 출력 비트, m을 3과 9로 설정하였을 때, LFSR의 스테이지 수에 따른 생성되는 난수의 최대 주기를 구하였다. 다중 LFSR 구조는 출력 비트수만큼의 LFSR들로 구성되어 회로 크기가 매우 커지는 반면 이상적인 난수열의 최대 주기를 얻을 수 있다. 기존 Leap-ahead LFSR은 출력 비트수에 상관없이 하나의 LFSR만을 갖고, (그림 5)와 (그림 6)에 보여진 것처럼 출력 비트수와 LFSR의 스테이지 수의 관계에 의해 난수열의 최대주기가 급격한 감소하는 구간을 갖는다. 그러나 제안된 세그먼테이션 Leap-ahead LFSR 구조의 경우 난수열의 최대 주기 감소를 기존 Leap-ahead 구조에 비해 크게 낮출 수 있다. 또한 Leap-ahead 구조를 기반으로 하기 때문에 하나의 LFSR로 회로를 구성할 수 있어 회로 크기면에서 효율적이다. 또한, 그림에서 보여지듯이 기존 Leap-ahead 구조의 경우 LFSR 스테이지의 수, n과 상관없이 최대 주기 감소율이 크다. 그러나, 제안된 세그먼테이션 Leap-ahead 구조의 경우 LFSR 스테이지의 수가 증가함에 따라 최대 주기 감소율이 크게 감소함을 알 수 있다. 따라서, LFSR 스테이지의 수와 출력 비트수가 많을수록 다중 LFSR 구조와 비슷한 최대 주기를 가짐을 알 수 있다.



(그림 5) LFSR 스테이지 수, n에 따른 생성 난수의 최대 주기 변화 (m=3)



(그림 6) LFSR 스테이지 수, n에 따른 생성 난수의 최대 주기 변화 (m=9)

표 1은 기존 VHDL을 이용하여 RTL 레벨로 합성한 후 Leap-ahead 구조와 본 논문에서 제안한 세그먼테이션 Leap-ahead 구조의 회로 크기 및 성능을 비교하였다. 우선 회로 크기의 경우 기존 Leap-ahead 구조는 26개의 slices가 사용되었고, 제안된 세그먼테이션 Leap-ahead 구조의 경우 35개의 slices로 회로크기는 35% 증가하였다. 그리고 동작 주파수도 제안된 Leap-ahead 구조의 경우 임계경로에 Mux가 하나 추가되어 기존 Leap-ahead 구조가 최대 651 MHz로 동작하는 반면 가 나왔지만 제안된 Leap-ahead 구조의 경우 22% 감소된 535 MHz로 동작한다. 그러나, 다중 LFSR 구조에 비해서는 회로 크기가 크게 감소되었다. 또한, LFSR의 스테이지 수와 출력 비트수와의 관계에 따라 생성되는 난수열의 최대 주기가 기존 Leap-ahead 구조는 크게 감소하나, 제안된 구조는 이와 같은 문제를 해결할 수 있다. 또한, URNG의 출력 비트수가 크게 증가할수록, 난수열의 최대 주기가 다중 LFSR 구조에 가깝게 됨으로써 상관관계가 적어져 암호 효율이 높아짐을 확인하였다.

<표 1> Leap-ahead 성능 및 회로 크기 비교
(LFSR 스테이지 수, n=18, 출력 비트 수, m=18)

구조	주기	LFSR 수	Slices	주파수 (MHz)
기존 Leap-ahead	29,127	1	26	651
제안된 Leap-ahead	261,121	1	35	535

5. 결 론

Leap-ahead 구조는 다중 LFSR 구조에 비해 작은 크기로 멀티 비트의 출력을 얻어낼 수 있기 때문에 최근 복잡한 암호 시스템에 적합하다. 그러나, LFSR 스테이지 수와 출력 단계수의 관계에 따라 급격히 생성된 난수열의 주기가 감소된다는 단점을 갖는다. 본 논문에서 제안된 세그먼테이션 Leap-ahead 구조는 전체 구조를 두 개의 LFSR 세그먼트로 구분하여 난수열의 최대 주기의 감소폭을 크게 줄였다. 이와 같은 방법으로 LFSR 스테이지 수와 출력 단계수와 관계없이 생성된 난수열의 최대 주기가 높은 암호 효율을 얻을 수 있다. 또한 다중 비트 출력에도 LFSR을 하나만 사용함으로써 회로 크기를 크게 줄여 휴대 정보기기 혹은 소형 USN 시스템에 적합하다.

참고문헌

[1] 조성진, 최언숙, 황윤희, 권민정, 김진경, 임지미, 허성훈, “LFSR 기반의 효과적인 PRPG의 설계”, 한국전자통신학회 2009 추계종합학술대회지 제3권 제2호, pp.41-46, 2009.

[2] 최병훈, 이종형, 조현숙, “RC4 스트림 암호 알고

리즘을 위한 고속 연산구조의 FPGA 구현 및 성능 분석", 정보보호학회 논문지 제4 권 제4 호, 2004.

[3] 성재철, 문덕재, 임홍수, 지성택, 이상진 "소프트웨어 구현에 적합한 스트림 암호의 대수덕 공격에 대한 안정성", 정보보호학회 논문지 제15 권 제1 호, 2005.

[4] 박창수, 조경언, "갈로이 선형 궤한 레지스터의 일반화", 2006년 1월 전자공학회 논문지 제 43권 CI편 제1호, 2006.

[5] 류희수, "최근 스트림 암호 동향 분석", 통신정보보호학회 제2권 제3호, pp.67-80 ,1992. 9.

[6] 정윤태, 임광철, 최은희, 박병전, "관용키암호알고리즘을 이용한 의사 난수 생성기", KSIAM IT series Vol. 9, No.2, pp.21-29, 2005.

[7] S. Mourad and Y. Zorain, *Principles of Testing Electronic Systems*, John Wiley & Sons, 2000.

[8] X. Gu and M. Zhang, "Uniform random number generator using Leap-Ahead LFSR architecture," 2009 Int'l Conf. on Computers and Communication Security, pp. 150-154, 2009.

[9] M. Goresky and A. M. Klapper, "Fibonacci and Galois representations of feedback-with-carry sh ift registers," IEEE Trans. on Information Theor y, vol. 48, no. 11, pp. 2826-2836, 2002.

[10] Pong P. Chu and Robert E. Jones, "Design Te chniques of FPGA Based Random Number Gene rator"

[11] 이석한, 허언, 이용석, "임베디드 시스템에 적합한 듀얼모드 의사난수 생성 확장 모듈 설계", 2009년 8월 전자공학회 논문지 제 46 권 SD 편 제 8 호, pp.682-688, Aug. 2006.

[저 자 소 개]

전 민 정 (Min-jung Jeon)

2011년 2월: 강원대학교 공학대학 정보통신공학과 학사
2011년~현재: 강원대학교 공학대학 정보통신공학과 석사 재학 중

email: wjsals0331@kangwon.ac.kr

김 상 춘 (Sang-choon Kim)

1986년: 한밭대학교 전자계산학과 학사
1989년: 청주대학교 전자계산학과 석사
1999년: 충북대학교 전자계산학과 박사
1983년~2001년: 한국전자통신연구원 정보보호연구단
2001년~현재: 강원대학교 정보통신공학과 주교수

email: kimsc@kangwon.ac.kr

이 제 훈 (Je-Hoon Lee)

1998년 8월 : 충북대학교 정보통신공학과 학사
2001년 2월 : 충북대학교 정보통신공학과 통신회로 및 시스템공학 석사
2005년 2월 : 충북대학교 정보통신공학과 통신회로 및 시스템공학 박사
2005년 4월 ~2006년 4월 : Univ. of Southern California Viterbi School 박사후연구원
2006년 8월 ~2009년 8월: 충북대학교 BK21 충북정보기술사업단 초빙조교수
2009년 8월 ~ 현재 : 강원대학교 삼척캠퍼스 전자정보통신공학부 조교수

email: jehoon.lee@kangwon.ac.kr