



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2019년01월14일  
(11) 등록번호 10-1914701  
(24) 등록일자 2018년10월29일

- (51) 국제특허분류(Int. Cl.)  
G06N 3/063 (2006.01) G06N 3/04 (2006.01)  
G06N 3/08 (2006.01)
- (52) CPC특허분류  
G06N 3/063 (2013.01)  
G06N 3/04 (2013.01)
- (21) 출원번호 10-2016-0181976
- (22) 출원일자 2016년12월29일  
심사청구일자 2016년12월29일
- (65) 공개번호 10-2018-0077533
- (43) 공개일자 2018년07월09일
- (56) 선행기술조사문헌

- (73) 특허권자  
한국과학기술원  
대전광역시 유성구 대학로 291(구성동)
- (72) 발명자  
신영수  
대전광역시 유성구 대학로 291 한국과학기술원  
윤기원  
충청남도 공주시 이인면 별문길 47
- (74) 대리인  
특허법인 다해

Ichihara, Hideyuki, et al. "Compact and accurate stochastic circuits with shared random number sources." Computer Design (ICCD), 2014 32nd IEEE International Conference on. IEEE, 2014.\*

(뒷면에 계속)

전체 청구항 수 : 총 5 항

심사관 : 서광훈

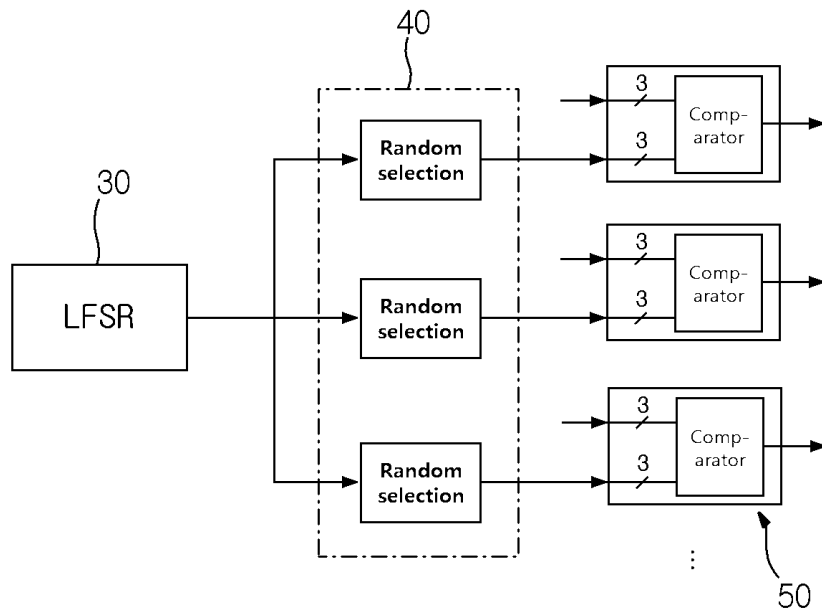
(54) 발명의 명칭 **스토캐스틱 컴퓨팅 기반의 에스엔지 및 이를 이용한 뉴로모픽 회로**

**(57) 요약**

본 발명에 따른 SNG는 유니크 시드(Unique seed)를 이용하여 각 뉴런마다 독립된 난수를 제공하는 기존의 SNG(Stochastic Number Generator) 대비 뉴로모픽 회로의 면적을 대폭 감소시킬 수 있고, 생성된 난수를 로테이션(Rotation)하여 비교기에 공급하는 로테이셔널 방식의 LFSR(Linear Feedback Shift Register) 대비 다량의 난

(뒷면에 계속)

**대표도** - 도9



수를 생성 가능하므로, 뉴로모픽 회로의 면적을 크게 낮추면서도 적은 비트수로 많은 난수를 생성하여 이용할 수 있다. 또한, 상기한 SNG를 적용하여 뉴로모픽 회로를 구현 시, 기존의 뉴로모픽 회로 대비 작은 면적과 높은 난수 효율성을 구현할 수 있다. 이를 위해 본 발명은 난수열 발생기에서 생성되는 m개의 난수 비트열 중 n개의 비트를 랜덤 추출하여 복수의 난수열을 생성하는 난수열 추출기 및 데이터 입력단과 난수 입력단을 구비하며 상기 데이터 입력단으로 인가되는 데이터와 상기 난수 입력단으로 인가되는 난수열을 비트 단위로 비교하는 복수의 비교기를 포함하여 구성될 수 있다.

(52) CPC특허분류

G06N 3/08 (2013.01)

(56) 선행기술조사문헌

Iwagaki, Naoya Kubota Hideyuki Ichihara Tsuyoshi, and Tomoo Inoue. "Stochastic Number Generation with Internal Signals of Logic Circuits.", SASIMI 2016, 2016.10.24.

Št'astný, Jakub. "Stochastic arithmetic complex number operators." Applied Electronics (AE), 2016 International Conference on. IEEE, 2016.

김경훈, 기계학습 시스템 설계를 위한 방법, 서울대학교 공학박사 학위논문, 서울대학교, 2016.2.

US20100241680 A1\*

\*는 심사관에 의하여 인용된 문헌

공지예외적용 : 있음

---

## 명세서

### 청구범위

#### 청구항 1

난수열 발생기에서 생성되는 m개의 난수 비트열 중 n개의 비트를 랜덤 추출하여 복수의 난수열을 생성하는 난수열 추출기; 및

데이터 입력단과 난수 입력단을 구비하며 상기 데이터 입력단으로 인가되는 데이터와 상기 난수 입력단으로 인가되는 난수열을 비트 단위로 비교하는 복수의 비교기;를 구비하며,

상기 난수열 추출기는,

상기 m개의 난수 비트열로  $\mu P_n$ 개의 난수열을 2차 생성하는 것을 특징으로 하는 스토캐스틱 컴퓨팅 기반의 에스엔지(Stochastic Number Generator).

#### 청구항 2

제1항에 있어서,

상기 n은,

비트 수가 상기 m보다 작고, 1 보다 큰 것을 특징으로 하는 스토캐스틱 컴퓨팅 기반의 에스엔지(Stochastic Number Generator).

#### 청구항 3

제1항에 있어서,

상기 난수열 발생기는,

유니크 시드(Seed)를 초기값으로 하여 상기 m개의 비트열을 생성하는 것을 특징으로 하는 스토캐스틱 컴퓨팅 기반의 에스엔지(Stochastic Number Generator).

#### 청구항 4

삭제

#### 청구항 5

제1항에 있어서,

상기 난수열 발생기는,

LFSR(Linear Feedback Shift Register) 난수열 발생기인 것을 특징으로 하는 스토캐스틱 컴퓨팅 기반의 에스엔지(Stochastic Number Generator).

#### 청구항 6

제1항 내지 제5항 및 제5항 중 어느 한 항의 에스엔지를 구비하는 스토캐스틱 컴퓨팅 기반의 에스엔지를 구비하는 뉴로모픽 회로.

## 발명의 설명

### 기술 분야

본 발명은 SNG(Stochastic Number Generator) 및 뉴로모픽 회로에 관한 것으로, 뉴로모픽 회로를 구현 시, 가장 많은 면적을 차지하는 난수 발생기의 면적을 최소화하여 작고 효율적으로 구현되는 SNG 및 이를 구비하는 뉴로모픽 회로에 관한 것이다.

[0001]

**배경 기술**

- [0002] 뉴로모픽(Neuromorphic) 회로는 수많은 뉴런들이 시냅스에 의해 연결되는 인간의 뇌 구조를 모델링하는데 주안점을 두어 설계된다. 뉴로모픽 회로는 인간의 신경망과 마찬가지로, 입력층(Input layer), 출력층(Output layer) 및 은닉층(Hidden layer)으로 모델링되며,
- [0003] 입력층은 데이터를 입력받고, 데이터의 연산은 은닉층 및 출력층이 담당하도록 구성된다. 또한, 뉴로모픽 회로를 구성하는 뉴런은 인간의 뉴런과 마찬가지로, 각 뉴런은 이웃하는 타 뉴런과 연결되고, 타 뉴런의 결과 값에 영향을 주도록 구성될 수 있다.
- [0004] 각 뉴런은 도 1에 도시된 바와 같이 입력 데이터(x1, x2)와 가중치 값(w1, w2)을 각각 승산한 후, 가산기에서 더해지고, 더해진 값이 쓰레스홀드 평선(threshold function)을 거쳐 최종 출력이 산출되도록 구성된다. 이에 따라, 뉴로모픽 회로를 구성하는 각 뉴런은 기본적으로, 곱셈기, 가산기 및 쓰레스홀드 평선을 위한 SRAM을 필요로 한다. 단일 뉴런이 하나의 곱셈기, 하나의 가산기 및 쓰레스홀드 평선을 위한 SRAM을 필요로 하는데, 1) SRAM은 셀 하나의 크기 자체가 타 회로의 소자보다 큰 문제가 있고, 2) 곱셈기는 하드웨어적으로 연산 대상 데이터의 비트 수에 해당하는 가산기의 집합에 해당하므로 뉴로모픽(Neuromorphic) 회로의 면적을 증가시키는 문제를 유발할 수 있다.
- [0005] 이에 대해, 확률적 컴퓨팅(Stochastic Computing) 방법은 확률을 이용하여 확률적 근사치를 산출하고, 이를 통해 1)과 2)의 문제를 해결하고 있다.
- [0006] 확률적 컴퓨팅 방법에서는 확률을 비트 스트림(Bit stream)에서 논리 "1"이 몇 개가 도출되는가로 표현될 수 있다. 예컨대, 6/8은 8개의 비트 중 논리 "1"이 무작위로 6개가 존재하는 것으로 표현될 수 있고, 4/8은 8개의 비트 중 논리 "1"이 4개 존재하는 것으로 표현될 수 있다. 이를 이용하면, 곱셈기를 AND 게이트 하나로 대체할 수 있으며, 이는 도 2를 함께 참조하여 설명하도록 한다.
- [0007] 도 2는 확률적 컴퓨팅 방법에서 AND 게이트를 곱셈기로 적용하는 일 예를 도시한다.
- [0008] 도 2를 참조하면, 비트 스트림으로 표현되는 확률 6/8과 4/8을 곱셈하는 곱셈기를 단일 AND 게이트로 처리하는 것을 나타낸다.
- [0009] 곱셈기를 통해 6/8과 4/8을 곱할 경우, 그 결과 값은 3/8에 해당한다. 확률을 비트 스트림으로 표현하였을 때, 6/8은 비트 스트림 "10110111"으로 표현되고, 4/8은 비트 스트림 "10101001"로 표현된다.
- [0010] 비트 스트림 "10110111"과 비트 스트림 "10101001"을 각각 Pa, Pb로 하여 AND 게이트의 두 입력단에 인가할 때, AND 게이트는 그 연산 값(Sy)으로서 비트 스트림 "10100001"로 표현되며, 이 값은 3/8에 해당한다.
- [0011] 상기한 바와 같이, 확률적 컴퓨팅 방법을 적용하여 곱셈기를 AND 게이트로 단순화하면 인공신경망의 뉴런에 해당하는 곱셈기의 개수를 대폭 경감시킬 수 있다. 그러나, 뉴로모픽 회로를 구성하는 뉴런의 크기를 대폭 감소시킨다 하여도 전체 뉴로모픽 회로를 구성하는 회로의 면적은 크게 감소되지 않는다. 이는 뉴로모픽 회로를 구성하는 전체 로직의 70% 내지 80%의 논리소자는 각 뉴런에 각기 다른 난수를 생성하여 제공하기 위한 SNG(Stochastic Number Generator)와 관련된 회로가 차지하는데 따른 것으로 뉴로모픽 회로의 면적을 감소시키기 위해서는 SNG의 최적화가 요구된다.

**발명의 내용**

**해결하려는 과제**

- [0012] 본 발명은 면적을 최소화하면서 정확도를 크게 낮추지 않는 SNG 및 이를 포함하는 뉴로모픽 회로를 제공함에 있다.

**과제의 해결 수단**

- [0013] 상기한 목적은 본 발명에 따라, 난수열 발생기에서 생성되는 m개의 난수 비트열 중 n개의 비트를 랜덤 추출하여 복수의 난수열을 생성하는 난수열 추출부; 및 데이터 입력단과 난수 입력단을 구비하며 상기 데이터 입력단으로 인가되는 데이터와 상기 난수 입력단으로 인가되는 난수열을 비트 단위로 비교하는 복수의 비교기;를 통해 구현될 수 있다.

**발명의 효과**

[0014] 본 발명에 따른 SNG는 유니크 시드(Unique seed)를 이용하여 각 뉴런마다 독립된 난수를 제공하는 기존의 유니크 시드 방식의 SNG(Stochastic Number Generator) 대비 뉴로모픽 회로의 면적을 대폭 감소시킬 수 있고, 생성된 난수를 로테이션(Rotation)하여 비교기에 공급하는 로테이셔널 방식의 SNG 대비 다량의 난수를 생성 가능하므로, 뉴로모픽 회로의 면적을 크게 낮추면서도 적은 비트수로 많은 난수를 생성하여 이용할 수 있다. 또한, 상기한 SNG를 적용하여 뉴로모픽 회로를 구현 시, 기존의 뉴로모픽 회로 대비 작은 면적과 높은 난수 효율성을 구현할 수 있다.

**도면의 간단한 설명**

[0015] 도 1은 종래의 뉴로모픽 회로의 개념도를 도시한다.  
 도 2는 종래의 뉴로모픽 회로에서 곱셈기를 AND 게이트로 단순화하는 방법에 대한 개념도를 도시한다.  
 도 3은 본 발명의 일 실시예에 따른 SNG가 적용되는 뉴로모픽 회로의 개념도를 도시한다.  
 도 4는 실시예에 따른 SNG 회로의 개념도를 도시한다.  
 도 5는 도 4에 도시된 난수 발생기의 개념도를 도시한다.  
 도 6은 SNG에서 생성되는 난수의 확률적 정확성에 대한 참조도면을 도시한다.  
 도 7은 유니크 시드를 할당하여 정확도를 높이는 일 예에 대한 참조도면을 도시한다.  
 도 8은 각 비교기에 제공되는 난수가 중첩되지 않도록 하여 난수의 정확도를 높이는 방법을 나타낸다.  
 도 9는 실시예에 따른 SNG의 블록 개념도를 도시한다.  
 도 10은 실시예에 따른 뉴로모픽 회로를 적용하였을 경우, 뉴로모픽 회로의 전체 면적을 비교 설명하기 위한 참조도면을 도시한다.  
 도 11은 실시예에 따른 뉴로모픽 회로를 적용하였을 때, 뉴로모픽 회로의 정확도 향상을 설명하기 위한 참조도면을 도시한다.

**발명을 실시하기 위한 구체적인 내용**

[0016] 이하, 도면을 참조하여 본 발명을 상세히 설명하도록 한다.  
 [0017] 도 3은 본 발명의 일 실시예에 따른 SNG가 적용되는 뉴로모픽 회로의 개념도를 도시한다.  
 [0018] 도 3을 참조하면, 뉴로모픽 회로는 입력층(Input), 은닉층(Hidden) 및 출력층(Output)으로 구성되고, 은닉층(Hidden)과 출력층(Output)이 복수의 뉴런(Neuron)을 구비하는 신경망 구조에 따라 구현된다. 입력층(Input)에는 데이터(x1, x2) 외에, SNG가 입력 데이터와 가중치(w1 내지 w8)값을 이진수(binary)에서 비트 스트림(bit stream)으로 변환한다. 이후, 은닉층(Hidden)과 출력층(Output)에서는 뉴런 연산을 수행한다. 마지막으로 카운터(Counter)가 출력층(Output)의 출력을 비트 스트림에서 이진수로 다시 변환하도록 구성될 수 있다.  
 [0019] 도 3에서, 은닉층(Hidden)이나 출력층(Output)에 속하는 뉴런들은 입력 데이터와 가중치 값(w1 내지 w8)을 비교 연산하여 출력하는 구조를 가지며, 도 2를 통해 전술한 바와 같이 단일 게이트로도 구현될 수 있으므로 전체 뉴로모픽 회로에서 차지하는 면적은 극히 적다. 반면, SNG는 각 뉴런에게 독립된 가중치나 난수를 제공하여야 하므로 뉴런의 수가 증가하는 만큼 더욱 복잡하고 많은 소자로 구성되어야 하므로 뉴로모픽 회로에서 차지하는 면적이 매우 크다.  
 [0020] 도 4는 실시예에 따른 SNG 회로의 개념도를 도시한다.  
 [0021] 도 4를 참조하면, SNG는 난수 발생기(10)와 비교기(20)로 구성되며, 비교기(20)는 입력 데이터(Binary number)와 난수 발생기(10)에서 제공되는 데이터를 비트 단위로 비교하여 출력하도록 구성된다.  
 [0022] 예컨대, 입력 데이터로서, 5/8을 의미하는 "5"가 입력 데이터로 들어오면, 랜덤 넘버와 비교하여 클 경우, 논리 "1"을 출력하고, 반대의 경우, 논리 "0"을 출력할 수 있다. 8비트 베이스인 경우, 난수는 0부터 7까지의 값이 무작위로 나오기 때문에 확률에 대응되는 비트 스트림이 만들어질 수 있으며, 비교기(20)에서 출력되는 출력값

은 뉴런으로 제공되어 뉴런의 입력 데이터로 이용된다.

- [0024] 도 5는 도 4에 도시된 난수 발생기의 개념도를 도시한다.
- [0025] 도 5를 참조하면, 난수 발생기(10)는 통상 LFSR(Linear Feedback Shift Register)로 구현되며, 1비트를 저장하는 n개의 플립플롭과 이들의 출력값을 논리 연산하는 XOR 게이트로 구성될 수 있다.
- [0026] 예컨대, 초기값(1, 0, 0)이 플립플롭에 저장되어 있다고 가정하면, 다음 사이클(cycle)에는 초기값 "1, 0, 0"이 한 자리씩 자리 이동을 하여 "0, 1, 0"이 되고, 그 다음 사이클에는 "0, 0, 1"로 그 값이 변환된다. XOR 게이트는 입력 데이터의 입력값이 서로 상이할 때, 출력값이 논리 "1"이므로, 이것을 매 사이클마다 반복하면 아래와 같이 랜덤 넘버를 생성하게 된다.
- [0027] LFSR에서 중요한 점은, 0을 제외한 1부터 2의 n승 마이너스 1까지의 숫자가 동등하게 나온다는 점과 시드(Seed)가 달라지면 난수 스트림이 달라진다는 점이다. 즉, 도 5에서는, 시드로 주어진 초기값이 논리 "1, 0, 0"이므로, "0, 1, 0", "0, 0, 1"의 순서로 난수가 출력되지만, 초기값 (1, 0, 0)이 바뀌면, 출력되는 난수의 값이 달라지는 것을 의미한다.
- [0028] 또한, LFSR에서 출력되는 난수의 정확도는 비트 스트림의 길이나 난수의 독립성에 따라 달라질 수 있다. 이는 도 6을 함께 참조하여 설명하도록 한다.
- [0029] 도 6은 SNG에서 생성되는 난수의 확률적 정확성에 대한 참조도면을 도시한다.
- [0030] 난수의 정확도는 발생하는 난수가 서로 중복되지 않게 뉴런에 제공되는 정도를 의미할 수 있다.
- [0031] 따라서, 난수의 정확도는 난수를 구성하는 비트 스트림의 길이가 길어질수록 증가하고, 난수를 생성하는 난수 생성기가 독립적일수록 증가하며, 반대의 경우 감소한다고 볼 수 있다.
- [0032] 난수가 상호 독립적이지 않고 연관되어 있으면, 뉴런에 제공되는 비트 스트림이 비슷해질 가능성이 있다. 예컨대, 도 5에 도시된 것처럼, LFSR의 시드(Seed)가 동일한 경우, 상호 독립적인 난수 발생기라 하더라도 동일한 난수를 생성할 우려가 있다. 이에 대해, 시드를 서로 상이한 값으로 할당하면 비트 스트림 간 독립성이 높아져 정확도를 높일 수 있으나, 독립된 난수 발생기와 상호 연관성이 적은 시드를 뉴런에 제공하기 위해서, 뉴로모픽 회로는 더욱 복잡해질 우려가 있다.
- [0033] 이에 대해, 유니크 시드(Unique seed)를 이용하여 동일한 난수가 발생하지 않도록 할 수 있으며, 이는 도 7과 도 8을 참조하여 설명하도록 한다.
- [0034] 도 7은 유니크 시드를 할당하여 정확도를 높이는 일 예에 대한 참조도면을 도시한다.
- [0035] 도 6을 통해 설명한 진술한 문제의 해결을 위한 첫번째 기존 방법은 유니크 시드 방법(unique seed method)으로, 이 방법은 SNG 마다 LFSR 시드를 다른 값으로 할당한다. 가장 쉽게 난수의 중복을 방지할 수 있으나,
- [0036] - 각 SNG 마다 고유의 시드를 할당하기 위한 로직이 요구되고,
- [0037] - 그에 따라 뉴로모픽 회로의 면적이 증가하는 단점이 있다.
- [0039] 다음으로, 도 8은 각 비교기에 제공되는 난수가 중첩되지 않도록 하여 난수의 정확도를 높이는 방법을 나타낸다.
- [0040] 도 8을 참조하면, 단일 LFSR(10)을 이용하되, LFSR(10)에서 출력되는 출력값을 1비트씩 로테이션(Rotation)하여 각 비교기에 제공함으로써, 각 비교기(21, 22, 23)에 제공되는 난수가 상호 중첩되지 않도록 하는 SNG를 도시한다.
- [0041] 도 8에 도시된 SNG는 단일 LFSR(10)에서 출력되는 난수를 공유하는 방법으로서, SNG에는 비교기(21, 22, 23)만 존재하고, 난수는 외부의 LFSR(10)에서 제공 받도록 구성될 수 있다.
- [0042] 단일 LFSR(10)이 생성하는 단일 난수를 도 8에 도시된 것처럼 비교기(21)에는 1비트 로테이션하여 제공하고, 비교기(22)에는 2비트 로테이션하여 제공하며, 비교기(23)에는 3비트 로테이션하여 제공할 수 있다.
- [0043] 1비트 로테이션된 난수, 2비트 로테이션된 난수 및 3비트 로테이션된 난수가 각각 비교기(21), 비교기(22), 및 비교기(23)에 난수로서 제공된다.

- [0044] 로테이션은 도 8에 도시된 것처럼, k bit 만큼 LFSR(10)에서 출력되는 난수를 회전시키는 것이다. 이러면 1bit 로테이션과 2bit 로테이션된 값이 달라지고, 결과적으로 각 비교기(21, 22, 23)를 위한 서로 다른 시드가 만들어진다.
- [0045] 한편, 도 8에 도시된 SNG는 LFSR(10)의 개수가 도 7에 도시된 것 대비 대폭 감소되어 뉴로모픽 회로의 면적을 감소시키는 잇점이 있으나, 하나의 LFSR(10)이 생성할 수 있는 난수의 수가 제한되는 단점이 있다.
- [0046] 본 발명은 상기한 문제점을 해결하기 위해 안출되었으며, 하나의 LFSR이 비교기에 공급하는데 충분한 만큼의 독립적인 난수를 생성할 수 있도록 한다. 이하, 도 9를 참조하여 본 발명의 SNG를 상술하도록 한다.
- [0047] 도 9는 실시예에 따른 SNG의 블록 개념도를 도시한다.
- [0048] 실시예에 따른 SNG는 단일 LFSR(30), 난수열 추출기(40) 및 비교기(50)를 포함하여 구성될 수 있다. LFSR(30)은 도 7과 도 8을 통해 설명된 것과 동일한 것으로, 유니크 시드(Seed)를 초기값으로 하여 순차로 비트열을 생성할 수 있다.
- [0049] 난수열 추출기(40)는 연결되는 비교기(50)의 수에 따라 LFSR(30)에서 생성된 비트열의 일 영역을 추출할 수 있다. 이때, 난수열 추출기(40)는 LFSR(30)에서 생성된 비트열을 구성하는 비트 수 보다는 작아야 하고, 1 비트 보다는 큰 비트열로 구성되어야 한다.
- [0050] 예컨대, LFSR(30)에서 출력되는 비트열이 5비트라면, 난수열 추출기(40)는 2비트 내지 4비트로 구성되는 비트열을 난수열로 추출할 수 있다.
- [0051] 이때, 난수열 추출기(40)는 LFSR(30)에서 출력되는 비트열을 랜덤하게 추출하여야 하며, 사전에 어떤 규칙을 가지고 추출하지 않는 것이 바람직하다.
- [0052] 전술한 도 5를 통해 설명한 바와 같이, 본 발명에서 확률은, 특정 비트열에 포함되는 논리 "1"(또는 논리 "0")의 개수에 따라 정해진다. 즉 8개의 비트 중 3개의 비트가 논리 "1"이면, 확률은 3/8에 해당한다. 따라서, 난수열 추출기(40)가 LFSR(30)에서 출력되는 비트열을 가지고 생성할 수 있는 경우의 수는  ${}_m P_n$ 개에 해당하며, m은 난수열 발생기(30)에서 출력되는 비트열의 수에 해당하고, n은 난수열 추출기(40)에서 추출되는 난수열의 구성 비트 수를 나타낸다.
- [0053] 난수열 발생기(30)에서 생성되는 비트열의 일 영역을 난수열로서 획득하는 경우의 수는  ${}_m P_n$ 개에 해당하므로, 난수열 추출기(40)는 이 수에 해당하는 난수를 생성할 수 있으며, 도 7을 통해 설명된 유니크 시드 방식의 SNG 대비 N(LFSR의 개수) : 1로 SNG 수가 최소화되고,
- [0054] 도 8을 통해 설명된 로테이셔널 방식의 SNG 대비 n :  ${}_m P_n$ 의 비율로 비교기(50)에 제공 가능한 난수의 수를 증가시킬 수 있다.
- [0055] 즉, 단일 LFSR(30)을 이용한다는 측면에서는 로테이셔널 방식의 SNG와 유사점이 있으나,
- [0056] 도 8을 통해 설명된 로테이셔널 방식의 SNG는 LFSR(20) 출력 비트수에 따라 비교기(21, 22, 23)에 제공할 수 있는 난수의 수가 제한되는 반면, 실시예에 따른 SNG는 출력 비트수(m)와 비교기(50)의 수(n)의 경우의 수로서 표현되어( ${}_m P_n$ ) 그 수가 대폭 증가하게 된다.
- [0057] 또한, 유니크 시드(Seed)를 이용한다는 측면에서는 도 7을 통해 설명된 유니크 시드 방식의 SNG와 유사점이 있으나,
- [0058] 도 7을 통해 설명된 유니크 시드 방식의 SNG는 각 SNG 마다 독립적인 LFSR을 구비하여야 하는 반면, 실시예에 따른 SNG는 단일 LFSR(30)을 필요로 할 뿐이다.
- [0059] 즉, 실시예에 따른 SNG는 도 7의 유니크 시드 방식의 SNG와, 도 8의 로테이셔널 방식의 SNG 양자의 장점을 모두 가지면서도, 유니크 시드 방식의 SNG와 로테이셔널 방식의 SNG 대비 더 적은 면적으로 뉴로모픽 회로를 구현할 수 있도록 하는데 그 특징이 있다.
- [0060] 도 10은 실시예에 따른 뉴로모픽 회로를 적용하였을 경우, 뉴로모픽 회로의 전체 면적을 비교 설명하기 위한 참조도면을 도시한다.
- [0061] 도 10을 참조하면, 도 7의 유니크 시드 방식이 황색으로 표현되고, 도 8의 로테이셔널 방식이 청색으로 표현되

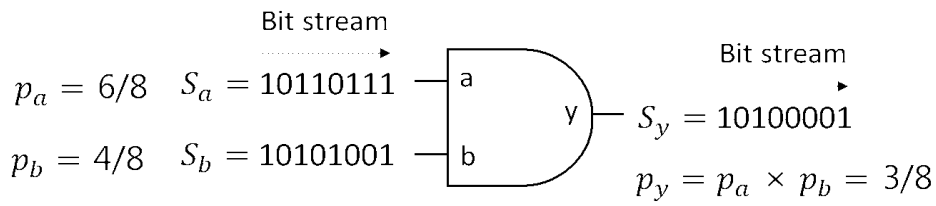




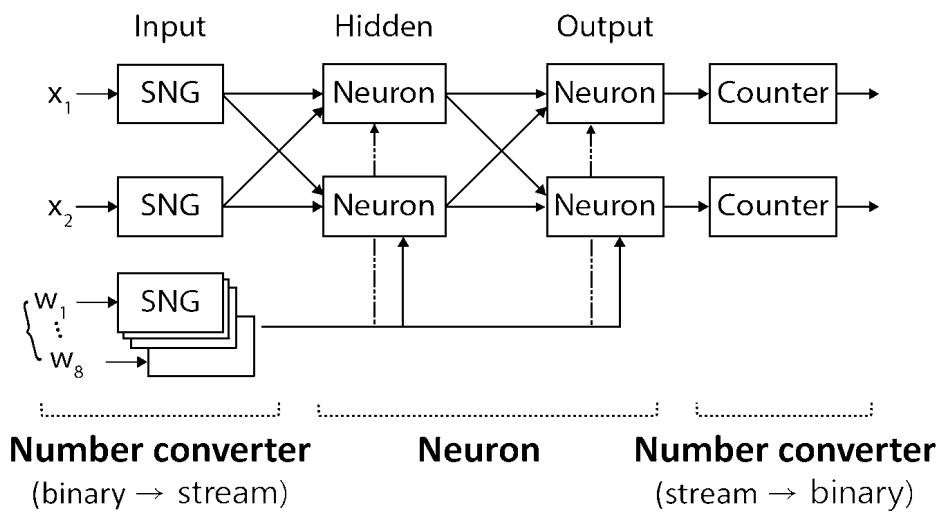


도면2

### Multiplication

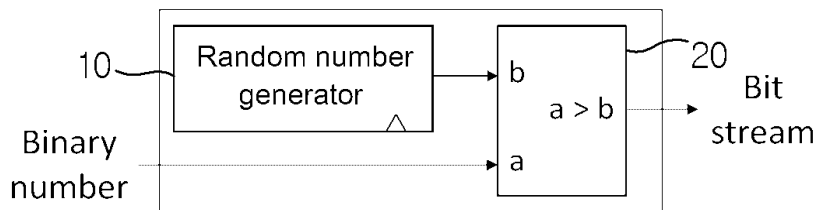


도면3



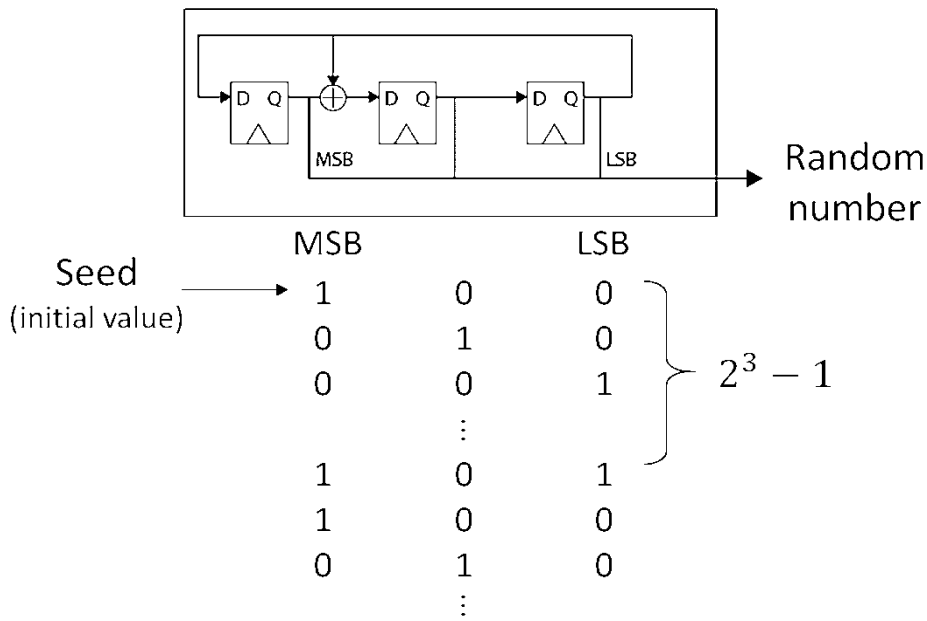
도면4

### Stochastic number generator (SNG)

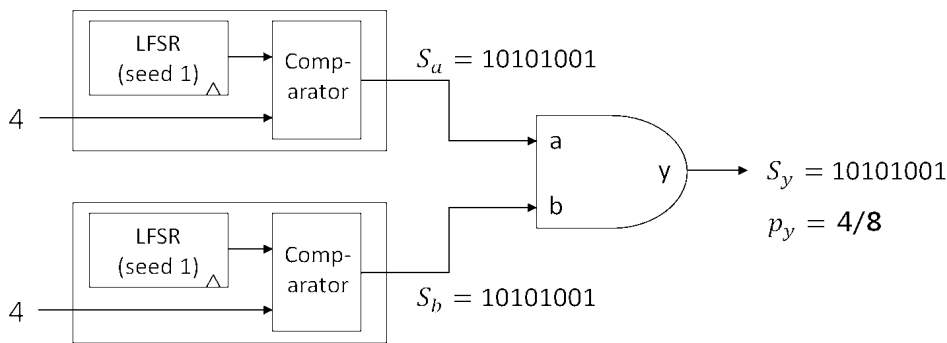


Binary number	5	(corresponds to 5/8)
Random number	4 5 7 3 6 1 2 4	
Bit stream	1 0 0 1 0 1 1 1	

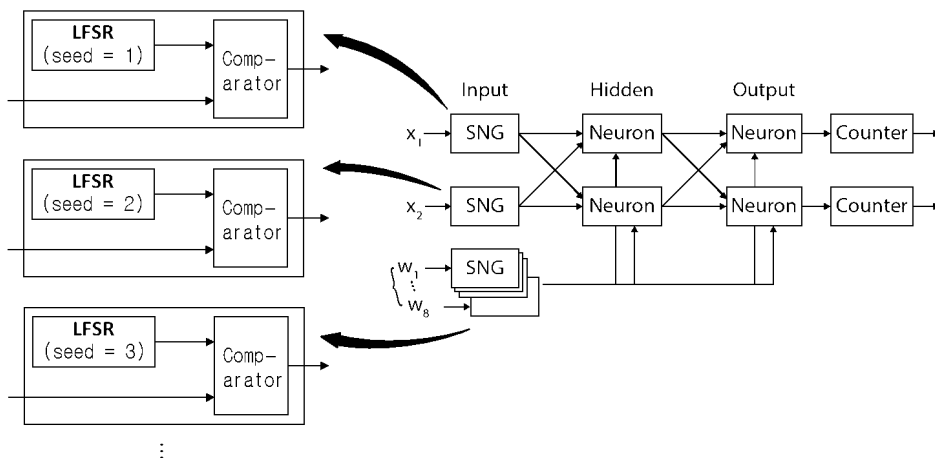
도면5



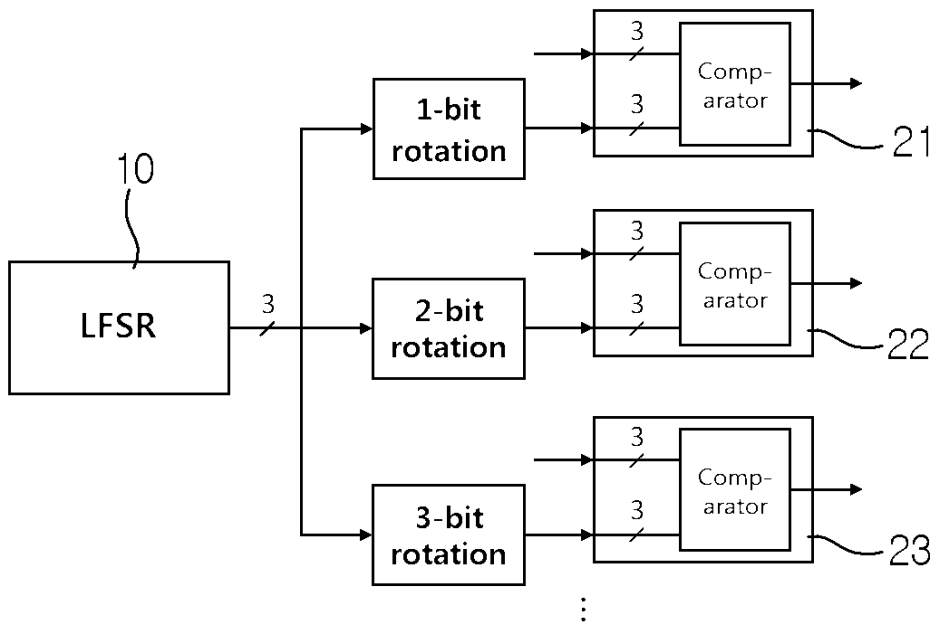
도면6



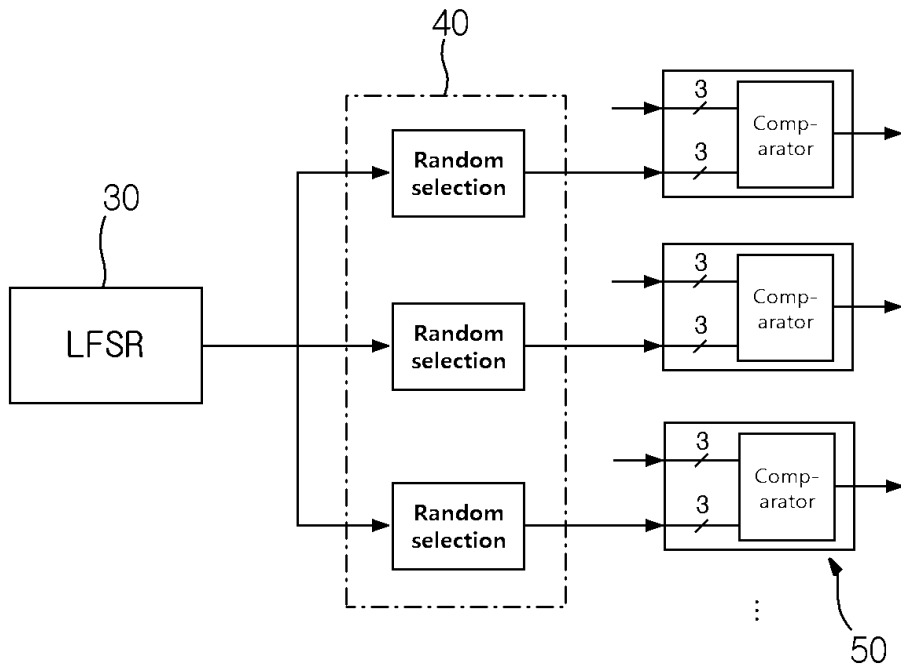
도면7



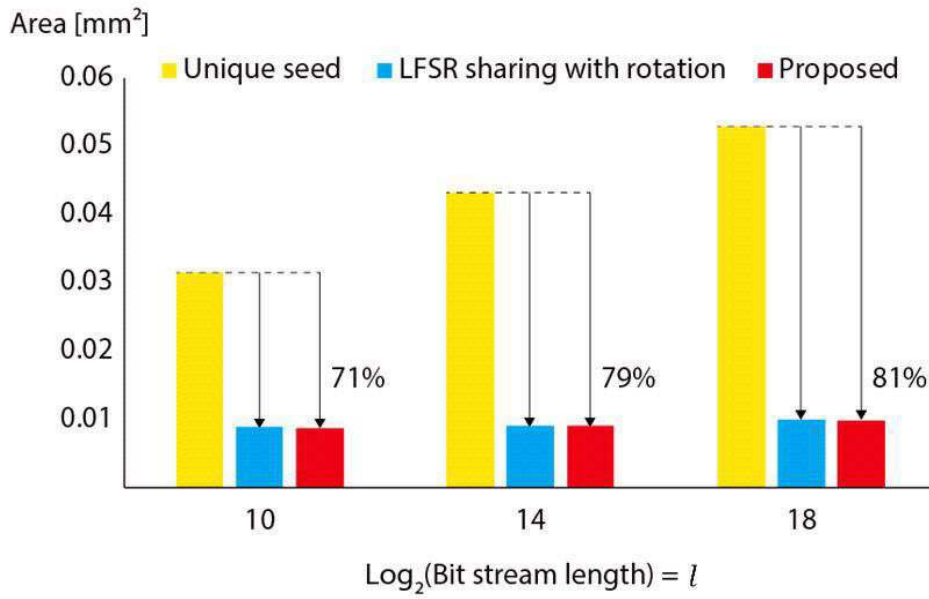
도면8



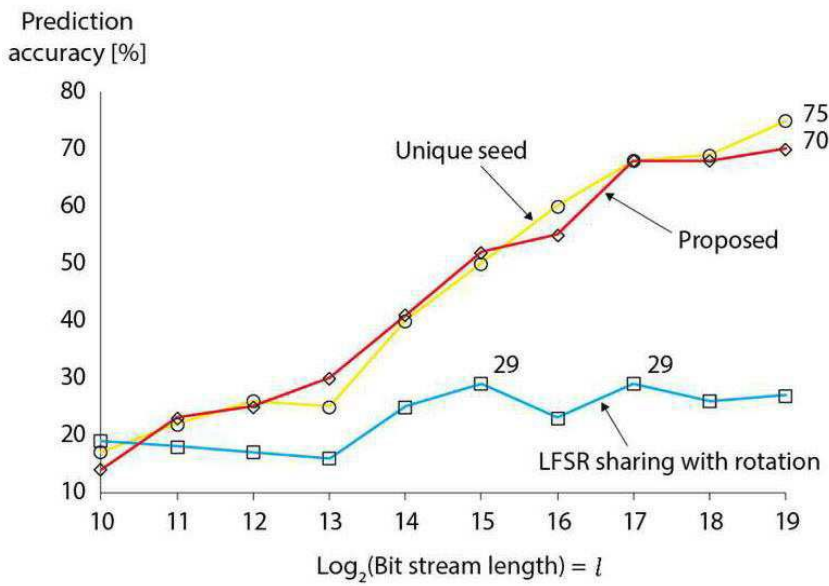
도면9



도면10



도면11



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 6

【변경전】

제1항 내지 제3항 및 제5항 어느 한 항의

【변경후】

제1항 내지 제3항 및 제5항 중 어느 한 항의