

# 순서회로의 Built-In Pseudoexhaustive Test를 위한 테스트 패턴 생성기 및 응답 분석기의 설계

김 연 숙<sup>†</sup>

## 요 약

본 논문에서는 BIST(Built-In Self Test)시 순서회로내의 조합회로를 pseudoexhaustive 시험하는데 필요한 테스트 패턴 생성기와 응답 분석기를 제안한다. 제안하는 테스트 패턴 생성기는 테스트 패턴의 초기값을 스캔 인 할 수 있고, exhaustive test pattern을 생성할 수 있다. 또한, 응답 분석기는 회로의 응답을 분석할 수 있을 뿐만 아니라 응답 결과를 스캔 아웃 할 수 있다. 이러한 테스트 패턴 생성기와 응답 분석기는 SRL과 LFSR을 결합하여 설계하였다.

## Design of Test Pattern Generator and Signature Analyzer for Built-In Pseudoexhaustive Test of Sequential Circuits

Yeoun Sook Kim<sup>†</sup>

## ABSTRACT

The paper proposes a test pattern generator and a signature analyzer required for pseudoexhaustive testing of the combinational circuit part within a sequential circuit when performing built-in self test of the circuit. The test pattern generator can scan in the seed test pattern and generate exhaustive test patterns. The signature analyzer can perform the analysis of the circuit response and scan out the result. Such test pattern generator and signature analyzer have been developed using SRL(shift register latch) and LFSR(linear feedback shift register).

### 1. 서 론

VLSI의 집적도가 증가함에 따라 회로의 최적 설계보다도 생산된 회로의 테스트가 매우 중요한 문제로 대두되었으며, 테스트를 고려한 설계방법에 대한 연구가 활발히 진행되고 있다.[1, 4]

회로의 측정도(testability)를 증진시키는 방법에는 ad hoc 방법, 구조적 방법, BIST(Built-In Self Test) 등이 있다. ad hoc 방법[1]은 설계자의 경험에 따라 테스트가 어려운 부분을 재설계함으로써 측정도를 증가시키는 방법으로 재설계에 걸리는 시간이 길다. 또한, 이 방법은 체계적이지 못하므로 자동화하기는 매우 어려운 실정이다.

이러한 문제점을 극복한 것이 구조적 방법[1]으로 순서회로의 테스트에 조합회로 테스트 방법을 그대로 사용하는 것이다. 이 부류에 속하는 방식으로는 LSSD(Level Sensitive Scan Design) [4], scan path, scan set, random access scan 등이 있다. 스캔 방식의 기본 개념은 회로 내부의 모든 기억 소자를 스캔 가능한 기억소자로 대치하고, 기억소자들 사이의 조합회로 부분의 테스트에 조합회로 테스트 방법을 그대로 사용하는 것이다. 이 방법은 조합회로 부분을 테스트하는데 필요한 테스트 패턴을 생성할 필요가 있으며, 긴 스캔 체인을 통하여 테스트 패턴을 스캔 인하고, 회로의 응답을 스캔 아웃 해야 한다. 따라서, 테스트 시간과 테스트 비용이 증가한다.

BIST 방법[1]은 테스트 패턴 생성과 적용에 필요한 시간을 줄이는 방법이다. 여기서는 테스트

<sup>†</sup> 정회원 : 인천대학교 전자계산학과 강사  
논문접수 : 1994년 1월 25일, 심사완료 : 1994년 6월 25일

트 패턴 및 회로의 응답 분석을 회로의 내부에서 행한다. 이때 테스트 패턴은 시험 패턴 생성기(test pattern generator)로 생성하고, 회로의 응답 분석은 응답 분석기(signature analyzer)로 행한다. 그런데, 지금까지 발표된 BIST 기법은 기억소자들 사이에 있는 조합회로 부분을 테스트 하는데 임의 시험(random test) 방법을 이용했다 [1]. 대부분의 방법들은 순서회로의 입력에 대한 출력의 비선형성을 근거로 조합회로 부분을 임의로 시험 하지만 이는 측정도 측면에서 불합리하다. 왜냐하면, 4입력 앤드 게이트 출력에서의 고정-과오-0(stuck-at-0) 고장을 검출할 확률이 1/16 이기 때문이다. 이러한 문제를 해결하기 위해서는 기억소자들 사이에 있는 조합회로 부분을 pseudoexhaustive 시험할 필요가 있다.

따라서, 순서회로내의 조합회로부를 보다 철저히 테스트할 수 있는 BIST 방법이 요구된다. 이에 본 논문에서는 순서회로내의 조합회로를 pseudoexhaustive 시험 [6, 7]하는데 필요한 시험 패턴 생성기와 응답 분석기를 제안한다. 제안하는 시험 패턴 생성기는 테스트 패턴의 초기값을 스캔 인 할 수 있고, exhaustive 시험 패턴을 생성한다. 또, 응답 분석기는 회로의 응답을 분석할 수 있을 뿐만 아니라 응답 분석 결과를 스캔 아웃 할 수 있다.

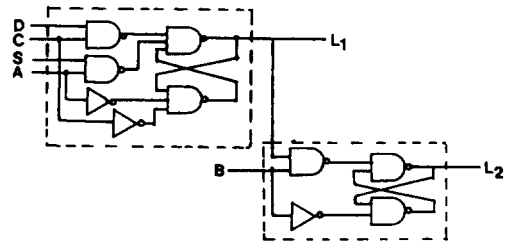
본 논문은 다음과같이 구성되어 있다. 제 2절에서는 SRL(Shift Register Latch)[4]과 LFSR(Linear Feedback Shift Register)[4]에 대하여 고찰한다. 제 3절에서는 SRL과 LFSR을 결합한 시험 패턴 생성기를 기술하고, 4절에서는 응답 분석기를 기술한다. 제 5절에서는 설계한 회로를 평가하고 결론을 맺는다.

## 2. SRL 과 LFSR

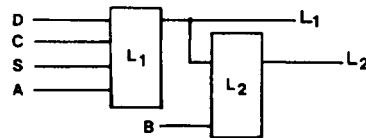
여기서는 순서회로내의 조합회로부의 pseudoexhaustive 시험에 사용할 수 있는 시험 패턴 생성기와 응답 분석기를 구성하는데 필요한 SRL과 LFSR에 관하여 기술한다.

### 2.1 SRL(Shift Register Latch)

(그림 1)은 LSSD[4]에서 사용하던 하자드-프리(hazard-free), 레이스없는(raceless) SRL이다.



(a) 논리 구조  
(a) Logic Diagram



(b) 기호  
(b) Symbol

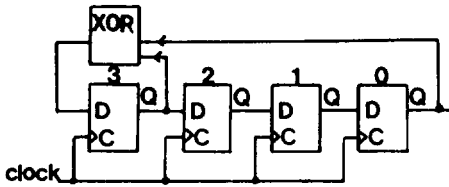
(그림 1) 쉬프트 레지스터 래치  
(Fig. 1) Shift Register Latch

(그림 1)에서 클럭 C는 정상동작시 D 입력을 L1 래치에 로드하는 시스템 클럭이고, 클럭 A는 테스트시 S 입력을 L1 래치에 로드하는 시스템 클럭이다. L1의 출력은 항상 두번째 래치 L2의 데이터 입력에 연결되어 있어서, 클럭 B는 L1의 출력을 L2에 로드한다. L1에있는 데이터 입력 S는 테스트를 하기 위한 쉬프트 기능을 수행하기 위한 것이고, 다른 입력 D는 정상동작시 데이터 입력으로 사용된다. 쉬프트 기능의 수행을 위해서 각 단의 L2의 출력은 다음 단의 입력 S에 연결된다. 또, 맨 첫단의 SRL의 출력은 주출력에 연결된다. SRL의 정상적 동작을 수행하기 위해서는 클럭 (A, B), (C, B), (C, A)가 중첩되어서는 안된다. 또, 시스템의 정상동작을 위해 필요한 것은 클럭신호가 래치를 안정상태로 세트시키도록 충분히 길어야 하며 클럭신호 사이의 간격은

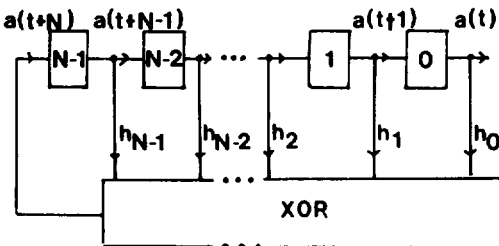
모든 래치가 변화되고 전송되어 다른 래치의 입력에서 안정상태가 되도록 충분히 길어야 한다. 이러한 구조는 시스템의 동작이 회로의 ac특성에 거의 의존하지 않게 된다[4].

2.2 LFSR(Linear Feedback Shift Register)

테스트 패턴은 LFSR[1]로 생성할 수 있다. LFSR이란 기억소자 (D F/F)를 외부입력없이 직렬로 연결한 후 XOR 게이트를 이용하여 피드백을 건 회로이다. (그림 2(a))는 4단 LFSR을 보여주며, (그림 2(b))는 LFSR의 일반적인 표준형을 보여준다. (그림 2(b))의 hi는 각 단의 출력으로부터 피드백이 있는지 없는지를 나타낸다. hi=1은 i단으로부터 피드백이 있음을 나타내고, hi=0은 i단이 XOR 피드백 네트워크에 연결되지 않음을 나타낸다. LFSR은 hi값들을 지정하거나, (그림 2(a))처럼 생성 함수(generating function)을 지정함으로써 일의적으로 결정된다.



(a) 4단 LFSR  
(a) 4-stage LFSR



(b) N단 LFSR  
(b) N-stage LFSR

(그림 2) LFSR의 표준형  
(Fig. 2) Standard form of LFSR

(표 1)그림 2의 상태 시퀀스  
(Table 1) State sequence for Fig. 2

State	Q1	Q2	Q3	Q4	State	Q1	Q2	Q3	Q4
0	1	0	0	0	8	1	1	0	1
1	1	1	0	0	9	0	1	1	0
2	1	1	1	0	10	0	0	1	1
3	1	1	1	1	11	1	0	0	1
4	0	1	1	1	12	0	1	0	0
5	1	0	1	1	13	0	0	1	0
6	0	1	0	1	14	0	0	0	1
7	1	0	1	0	15=0	1	0	0	0

(표 2) 1에서 32까지의 n 값에 대한 primitive polynomial  
(Table 2) Primitive polynomial for each n from 1 to 32

n	f(X)
1, 2, 3, 4, 6, 7, 15, 22	$1+X+X^n$
5, 11, 21, 29	$1+X^2+X^n$
10, 17, 20, 25, 28, 31	$1+X^3+X^n$
9	$1+X^4+X^n$
23	$1+X^5+X^n$
18	$1+X^7+X^n$
8	$1+X^2+X^3+X^4+X^n$
12	$1+X+X^4+X^6+X^n$
13	$1+X+X^6+X^4+X^n$
14, 16	$1+X^2+X^4+X^6+X^n$
19, 27	$1+X+X^8+X^6+X^n$
24	$1+X+X^8+X^7+X^n$
26	$1+X+X^8+X^6+X^n$
30	$1+X+X^2+X^{26}+X^n$
32	$1+X+X^4+X^{32}+X^n$

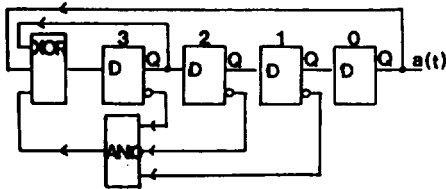
(그림 2(a))에 나타난 LFSR의 상태 시퀀스를 (표 1)에 나타냈다. 상태 시퀀스는 15 (2<sup>4</sup>-1) 클럭 후에 반복된다. 이것이 4단 LFSR의 최대 주기이다. 모든 기억소자가 0으로 되는 경우는 발생하지 않는다. 왜냐하면 모든 기억소자가 0이 되는 경우가 발생하면 피드백을 형성하는 XOR 게이트로 인하여 다음 패턴이 변하지 않게 되어 모든 기억소자가 0을 갖게 되기 때문이다. 일반적으로 n단 LFSR의 최대주기는 2<sup>n</sup>-1이다. 특정

n에 대한 최대주기의 LFSR이 항상 존재하며, 최대주기 LFSR에 대응하는 생성 함수를 primitive polynomial이라 부른다. primitive polynomial에 대한 자료는 Abramovici, Breuer, Friedman[1] 과 Peterson, Weldon[8]에 의해 명확하게 증명되었다. <표 2>는 1에서 32까지의 n값에 대한 primitive polynomial을 나타낸다.

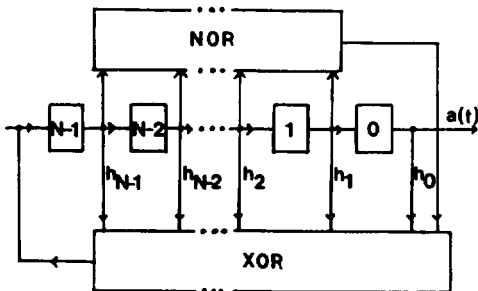
### 3. 테스트 패턴 생성기

여기서는, 먼저 LFSR을 이용한 완전 사이클 시험 패턴 생성기에 대하여 기술한 후 스캔 가능한 완전 사이클 시험 패턴 생성기에 대하여 기술한다.

완전 사이클 시험 패턴 생성기란 모든 상태가 0인 경우 테스트할 수 없는 불안정한 경우도 테스트할 수 있는 시험 패턴 생성기로 이에 (그림 3)의 완전 사이클 LFSR과 (그림 5)의 스캔 가능한 시험 패턴 생성기에 대해 기술한다.



(a) 완전 사이클 4단 LFSR  
(a) Full cycle 4-stage LFSR



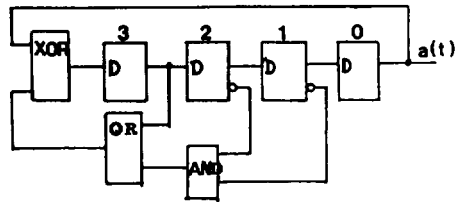
(b) 완전 사이클 N단 LFSR  
(b) Full cycle N-stage LFSR

(그림 3) 완전 사이클 LFSR  
(Fig. 3) Full cycle LFSR

### 3.1 완전 사이클 LFSR

(그림 3(a))와 같이 (그림 2(a))를 변형하면 모든 기억소자가 0값을 가지면서도 다음 상태에서 모든 기억소자가 0이 되지 않게 할 수 있다. Q1' Q2' Q3'를 XOR 게이트의 입력으로 사용한 이유는 상태 0001에 대해 LSB(Least Significant Bit)에 1이 입력되는 것을 막기 위한 것이다. 모든 상태를 시퀀스에 포함하는, 즉 완전 사이클 LFSR의 일반적인 형태는 (그림 3(b))와 같다. 기존의 논리 최소화법[2, 3]을 이용하면 요구되는 회로의 면적을 줄이는 것도 가능하다. (그림 3(a))를 최소화한 것이 (그림 4)이다. (그림 3(b))들은 회로를 사용하면 순서회로내의 조합회로를 pseudoexhaustive 시험하는데 필요한 테스트 패턴을 생성할 수 있다.

이와 같은 테스트 패턴을 일반적인 수학적 표현으로 나타내는 방법이 연구 과제로 남아있다.

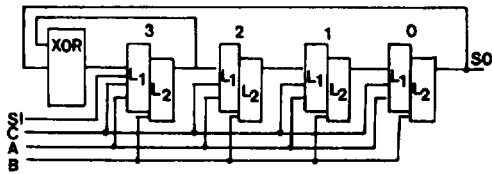


(그림 4) 최소화된 그림 3의 회로  
(Fig. 4) Simplified form of Fig. 3 circuit

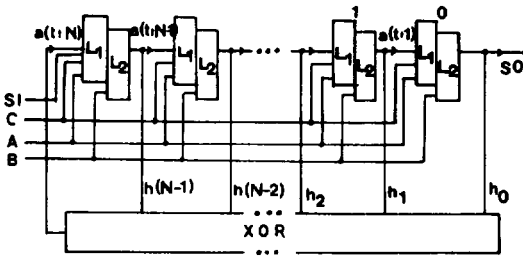
### 3.2 스캔 가능한 테스트 패턴 생성기

(그림 3)과 같은 회로는 순서회로 내부에 배치되어 있을 때 제어도(controllability)가 낮기 때문에 연속된 패턴을 인가하기는 어렵다. 따라서, 테스트 패턴의 초기값을 외부로부터 스캔 인 할 수 있는 기능을 추가할 필요가 있다. 이를 위하여 (그림 3)의 회로에 있는 기억소자에 2절에서 기술한 SRL을 결합하면 (그림 5)와 같은 구조의 스캔 가능한 시험 패턴 생성기에 의해 연속된 패턴을 스캔 할 수 있다. (그림 5(a))는 (그림 3(a))의 기억소자를 SRL로 대체한 것이고, (그림 5(b))는 (그림 3(b))의 기억소자를 SRL로 대체

한 것으로 본 논문에서 제안하는, 순서회로내의 조합회로부를 pseudoexhaustive 시험하는데 필요한 시험 패턴 생성기이다.



(a) 스캔 가능한 4비트 테스트 패턴 생성기  
(a) Scannable 4-bit test pattern generator



(b) 스캔 가능한 N비트 테스트 패턴 생성기  
(b) Scannable N-bit test pattern generator

(그림 5) 스캔 가능한 테스트 패턴 생성기  
(Fig. 5) Scannable test pattern generator

#### 4. 응답 분석기

여기서는, LFSR을 이용한 응답 분석기를 고찰한 후 순서회로의 응답을 분석할 수 있고 결과를 스캔 아웃 할 수 있는 응답 분석기에 대해서 기술한다.

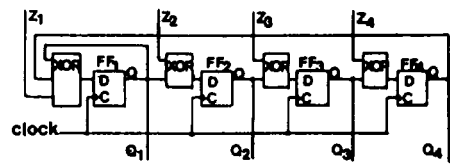
응답 분석기란 회로의 응답 분석을 행하는 것으로, 외부에서 초기 데이터를 설정하고 응답 분석 결과를 관찰하기가 어려우므로, 이에 테스트 데이터를 스캔 인 하고 응답 분석 결과를 스캔 아웃 할 수 있도록 (그림 6)의 병렬 응답 분석기와 (그림 7)의 스캔 가능한 응답 분석기에 대해 기술한다.

##### 4.1 LFSR을 이용한 응답 분석기

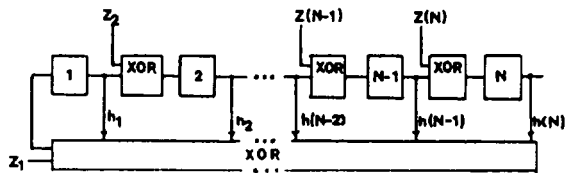
순서회로 내부의 조합회로를 pseudoexhaustive test을 사용하여 테스트할 경우 생성되는 회로의 출력은 방대하므로 이를 압축하는 것이 바람직하

다. 이를 위하여 여러 방법[1]이 제안되었으나 본 논문에서는 LFSR을 이용한 병렬 응답 분석기에 국한하여 고찰한다.

LFSR을 이용한 병렬 응답 분석기는 K개의 회로 출력을 K비트 코드 체커(code checker)를 이용하여 압축한다. 회로의 출력은 LFSR의 첫단의 XOR 게이트와 각 단계에 삽입된 XOR 게이트에 연결된다. (그림 6(a))는 4단 병렬 응답 분석기를 보여준다. (그림 6(b))는 n단 병렬 응답 분석기의 일반형을 보여준다.



(a) 4비트 병렬 응답 분석기  
(a) 4-bit parallel signature analyzer



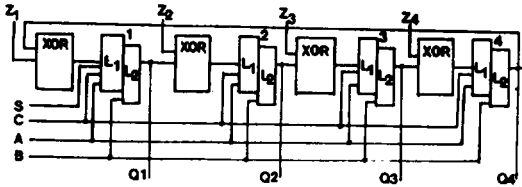
(b) N비트 병렬 응답 분석기  
(b) N-bit parallel signature analyzer

(그림 6) 병렬 응답 분석기  
(Fig. 6) Parallel signature analyzer

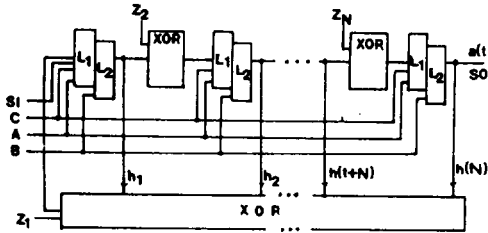
##### 4.2 스캔 가능한 응답 분석기

(그림 6)과 같은 회로는 순서회로 내부에 배치되어 있을 때 제어도와 관찰도(observability)가 낮기 때문에 응답 분석기의 초기 데이터를 외부에서 설정하고 응답 분석 결과를 외부에서 관측하기는 어렵다. 따라서, 초기 데이터를 스캔 아웃하고 응답 분석 결과를 스캔 아웃 할 수 있는 기능을 추가할 필요가 있다. 이를 위하여 (그림 6)의 회로에 있는 기억소자에 2절에서 기술한 SRL을 결합하면 (그림 7)과 같은 구조의 스캔 가능한 응답 분석기를 얻을 수 있다. (그림 7(a))는 (그림 6(a))의 기억소자를 SRL로 대체한

것이고 (그림 7(b))는 (그림 6(b))의 기억소자를 SRL로 대체한 것으로 본 논문에서 제안하는, 순서회로내의 조합회로부를 pseudoexhaustive 시험하는데 필요한 응답 분석기이다.



(a) 스캔 가능한 4비트 병렬 응답 분석기  
(a) Scannable 4-bit parallel signature analyzer



(b) 스캔 가능한 N비트 병렬 응답 분석기  
(b) Scannable N-bit parallel signature analyzer

(그림 7) 스캔 가능한 병렬 응답 분석기  
(Fig. 7) Scannable parallel signature analyzer

### 5. 결 론

본 논문의 특징은 제 3절과 제 4절에서 스캔 가능한 시험 패턴 생성기와 응답 분석기를 제안 하므로써, 기존의 BIST에서 순서회로내의 조합회로를 임의의 시험하는 것과는 달리, 순서회로내의 조합회로를 pseudoexhaustive 시험하는데 적합하도록 설계되었다는 것이다.

이 회로는 순서회로 내부에 배치되어 순서회로 내의 조합회로부를 pseudoexhaustive 시험하기 위한 시험 패턴 생성기와 응답 분석기로 사용할 수 있다. 또한, 스캔 가능하기 때문에 시험 패턴 생성기 및 응답 분석기의 초기값을 외부에서 임의로 설정할 수 있고, 응답 분석 결과를 외부로 쉽

게 스캔 아웃 할 수 있다는 장점을 가지고 있다. 그러나, 기존의 시험 패턴 생성기와 응답 분석기에 있는 기억소자를 SRL로 대체한 것이기 때문에 하드웨어 오버헤드가 발생한다. 또, 기존의 병렬 응답 분석기가 방대한 양의 회로응답을 압축하기 때문에 생기는 alias 문제[1]는 본 논문에서 제안하는 응답 분석기에도 그대로 존재한다.

앞으로의 연구과제는 순서회로내부의 조합회로가 pseudoexhaustive 시험이 가능하도록 회로를 분할(partitioning)[5]하고, 구성(configuration)하는 방법이 되어야 할 것이다.

### 참 고 문 헌

- [1] M. Abramovici, M. A. Breuer and A. D. Friedman, 'Digital System Testing and Testable Design', Computer Science Press, 1990.
- [2] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, 'Logic Minimization Algorithms for VLSI Synthesis', Kluwer Academic Publishers, Boston, 1984.
- [3] R. K. Brayton, R. Rudell, A. L. Sangiovanni-Vincentelli, and A. R. Wang, "Mis: A multiple-level logic optimization system," IEEE Trans. on Computer-Aided Design, pp. 1062-1081, Nov. 1987.
- [4] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testing," Proc. 14th Design Automation Conference., pp. 462-468, June. 1977.
- [5] W. B. Jone, and C. A. Papachristou, "A Coordinated Approach to Partitioning and Test Pattern Generation for Pseudoexhaustive Testing," Proc. Design Automation Conference, pp. 525-530, 1989.
- [6] E. J. McCluskey, S. Bozorgui-Nesbat, "Design For Autonomous Test," IEEE Trans. on Computers, pp. 866-875, Nov. 1981.
- [7] E. J. McCluskey, S. Bozorgui-Nesbat,

“Verification Testing—A Pseudoexhaustive Test Technique,” IEEE Trans. on Computers, pp. 561—546, Nov. 1984.

- [8] W. W. Peterson and E. J. Weldon, ‘Error—Correcting Codes’, 2nd Ed., The Colonial Press, Inc., 1972.

김 연 속

1981년 아주대학교 공과대학 전자공학과 졸업 (공학사)

1983년 아주대학교 전자공학과 전자공학전공 (공학석사)

1981년~1982년 아주대학교 전자공학과 조교

1994년~현재 건국대학교 대학원 전자공학과 박사과정

1992년~현재 인천대학교 전자계산학과 강사

관심분야 : VLSI Testing, VLSI Design Automation, 신경회로망, 음성인식, 패턴인식, 퍼지추론