

# MBDD를 이용한 저전력 VLSI설계기법

## A Method of Low Power VLSI Design using Modified Binary Decision Diagram

尹京鏞\* · 鄭德鎭\*\*

(Kyeong-Yong Yoon · Duck-Jin Chung)

**Abstract** - In this paper, we proposed MBDD(Modified Binary Decision Diagram) as a multi-level logic synthesis method and a vertex of MBDD to NMOS transistors matching. A vertex in MBDD is matched to a set of NMOS transistors. MBDD structure can be achieved through transformation steps from BDD structure. MBDD can represent the same function with less vertices- less number of NMOS transistors, consequently capacitance of the circuit can be reduced. Thus the power dissipation can be reduced. we applied MBDD to a full adder and a 4-2compressor. Comparing the 4-2compressor block with other synthesis logic, 31.2% reduction and 19.9% reduction was achieved in numbers of transistors and power dissipation respectively. In this simulation we used 0.8  $\mu\text{m}$  fabrication parameters.

**Key Words** : MBDD, Logic synthesis, VLSI Design

### 1. 서 론

노트북, 휴대전화 따위의 Battery powered system의 사용이 증가하고 보편화됨에 따라 제한된 용량에 대해서 시스템의 안정된 동작을 보장하며 사용시간을 최대로 확장할 수 있는 집적회로에서의 저전력 설계 기법이 요구되고 있다. 이에 따라 상위수준에서 하위수준까지의 각 레벨에서의 저전력 설계기법이 개발되어야 하며 궁극적으로는 그 모든 기법이 최적화 되어야 한다. 이 논문에서는 circuit level에서의 시도로, 저전력 구현을 위한 로직 합성방법으로서, BDD(Binary Decision Diagram)를 발전시킨 방법인 MBDD(Modified Binary Decision Diagram)를 제시하였다. 멀티레벨 로직합성의 방법인 BDD[1]는 Lee에 의해 제창되어 Bryant에 의해서 ROBDD[2](Reduced Ordered BDD)가 되면서부터 Canonicity를 가지게 되었다. (이하에서 BDD는 ROBDD를 의미한다) BDD에서부터 발전하여 ROBDD, TBDD 등 그 방법에 따라 여러 종류의 BDD 기법들이 보고되고 있으며, 이 논문에서의 MBDD도 기존의 BDD와는 다른 새로운 접근이라 할 수 있다. 기존의 logic을 표현하는 방법인 Sum of Product(SOP), 진리표, K-map 등은 방법은 n 개의 변수를 가지는 함수를 구현하기 위해서는 지수적 크기의 표현이 요구되고, canonical form을 갖지 않는다. 따라서 하나의 function이 서로 다른 표현을 가질 수 있게되므로

함수간의 등가성(equivalency)점검과 중복(tautology)점검이 어렵다는 단점을 가지고 있다. 이에 대해 BDD는  $2^n-1$ 의 표현이 요구되며[3] 주어진 함수에 대해서 최소화된 표현을 가지므로 등가성이나 점검이 용이하다. 주어진 함수에 대응하는 BDD로부터 변환을 거쳐 만들어진 MBDD 구조가 완성되며, 이 과정에서 보다 적은 개수의 정점(vertex)으로 함수를 표현할 수 있다. 이 MBDD의 각 정점은 두 개의 NMOS와 대응하게 된다. 결과적으로 보다 적은 개수의 트랜지스터를 사용하게 되므로 회로의 Capacitance가 줄어들어 이에 따른 전력을 감소시키게 된다. 함수를 구현하는 모든 회로가 NMOS로 구현되어 full-swing이 되지 않는 문제는 최종단에 보상단 (Level restoration block)을 사용함으로써 해결된다. 예시로 MBDD를 full-adder와 4-2 compressor에 적용하여 합성하였으며, NMOS를 사용한 여타 합성방법으로 합성된 회로와 비교하였다.

디지털 집적회로에서의 전력소모는 개략적으로 standby current, leakage current에 의한 static power dissipation과 short-circuit current, capacitance에 의한 dynamic power dissipation으로 구분할 수 있다.

$$P_{total} = P_{static} + P_{dynamic} = I_{standby}V_{dd} + I_{lkg}V_{dd} + I_{sc}V_{dd} + \alpha C f V_{dd}^2 \quad (1)$$

전체 소비전력 중에서 가장 큰 영역을 차지하고 있는 것이 dynamic power dissipation이므로 저전력을 위한 설계에 있어서는 dynamic power dissipation을 줄이는 것이 전체 전력소모를 최소화하는데 효과적인 방법이라 할 수 있다.

$$Pd = \alpha C f V_{dd}^2 \quad (2)$$

여기서  $\alpha$ 는 switching activity 등에 관련된 성분으로서 보다 상위수준의 저전력 설계 기법에서 다루어져야하고,  $f$

\* 準 會 員 : 仁荷大 電子電氣컴퓨터工學部 碩士課程  
 \*\* 正 會 員 : 仁荷大 電子電氣컴퓨터工學部 教授 · 工博  
 接受日字 : 1999年 12月 22日  
 最終完了 : 2000年 4月 29日

는 동작속도에 관련하여 주어진다.  $V_{dd}$ 가 제곱의 항이므로 전력을 줄이는 데 가장 효과적인 방법이나  $V_{dd}$ 의 감소는 trade off 관계에 있는 Delay 증가를 수반하게 되고 이 또한 보다 상위수준인 구조수준(architecture level)에서 다루어져야 한다. 마지막으로, Capacitance성분의 감소는 회로설계시 보다 뛰어난 알고리즘을 사용하여 같은 함수를 구현함에 있어서 그 함수를 나타내는 transistor의 수를 줄임으로써 이루어질 수 있다.

## 2. MBDD의 과정

이러한 시도로써 이 논문에서 제시한 MBDD의 첫 번째 단계에서는 임의의 함수를 BDD로 나타낸다. 그리고, MBDD의 한 정점을 출력을 공유하는 두 개의 NMOS로 이루어진 회로로 치환한다. 이 치환과정에서 BDD를 PMOS와 NMOS를 동시에 사용하면 full swing 등에 대한 문제가 해결되나 PMOS의 크기가 NMOS에 비해 두배 이상이 되어야 하므로 회로전체의 capacitance가 커진다는 단점이 있다. 따라서, NMOS만으로 모든 로직연산부분을 구성하고 출력부분에 보상단을 사용하는 것이 회로의 capacitance를 줄이는데 보다 효과적이다. 즉 함수의 연산부분을 MBDD를 적용하여 합성하는 것이다.

MBDD는 진리표로부터 주어진 함수의 output Bitarrays를 top vertex에 할당하고 전개한다. 그리고 같은 level에 있는 vertex에 대해서, 서로 다른 두 vertex가 같은 bits-Array를 가지고 있다면 서로 연결하고 하나의 vertex를 top vertex로 하는 이하의 tree는 제거한다. 이 과정에서 Canonicity가 보장되며 최소의 vertices 만으로 구성된 MBDD가 완성된다. 여기서 하나의 function에 대해 유일하게 하나의 MBDD 구조가 존재하는 (canonicity) 것은 중요하며 이는 이 방법이 자동화설계 응용에 있어서 매우 중요하다. 마지막 단계로 대해서 하나의 vertex를 NMOS를 사용하여 대치하고 보상단을 연결하여 회로를 구성한다. BDD로부터 MBDD로 변환을 하려면 다음과 같은 연산의 정의가 필요하다.

### 【정의】 • (dot) operation

두 개의 Bits array  $V1$ 과  $V2$ 가 주어지면 (각각의 bitarray가  $n$  개 bits로 되어있을 때)

$V1 \cdot V2 = V$ 의 연산은  $V$ 를 이루고  $V$ 는  $2n$  bits의 array를 가진다.

ex)  $[V1] = 0110$ ,  $[V2] = 1001$  이면  
 $V = V1 \cdot V2 = 01101001$

MBDD는 다음의 3단계를 따라 BDD의 vertex tree를 변형시킴에 따라 이루어진다.

### 2. 1 MBDD 변환

#### 1단계> 확장

주어진 함수에 대한 최종 출력의 bits array가 주어지면 top vertex에 bits array를 할당하고 • operation을 통해 bottom vertex가 될 때까지 확장한다. 여기서 Bottom

vertex란 2bit의 bits-array를 가지는 vertex를 말한다. 또한 top vertex는 보상단으로 연결되기 직전의 가장 상위에 위치한 vertex를 말한다. 예를 들어 01101000이 top vertex에 인가되었다면 0110과 1001은 다음 level의 두 vertex에 인가될 것이다. 이 단계가 끝나면 기본적 BDD의 형태에 bits array를 할당한 구조가 된다.

#### 2단계> 연결

vertex 확장 과정에서 같은 bits array를 갖는 서로 다른 vertex가 있으면 근접성에 상관없이 모두 연결하고, 오직 하나의 vertex만을 남기고 모두 제거한다. 이 vertices 간의 연결은 최종적 회로구현 관점에서는 Drain을 서로 연결하는 것과 같은 효과를 가져온다.

이 step은 "Index ( $w$ ) = index ( $v$ ) 인 두 vertex  $v, w$ 에 대해서  $id(low(v)) = id(low(w))$ 이고,  $id(high(w)) = id(high(v))$ 이면, 두 vertex 각각이 top vertex가 되는 sub-graph tree는 isomorphic"[3]이므로 타당성을 가진다. 여기서  $id(low(w))$ 는 vertex 14의 control signal이 low일 때의 값을 의미한다.

이 두 번째 과정에서 같은 bits array를 갖는 서로 다른 vertex가 있으면 근접성에 상관없이 모두 연결하고, 오직 하나의 vertex를 남기고 모두 없애는 과정을 통해 주어진 함수에 대한 MBDD를 이용한 구조의 유일성과 canonicity가 보장되고 이 과정을 통해 [4]와 구분된다. 이 논문에서 canonicity는 어떤 함수에 대해서 그에 대응하는 MBDD가 유일함을 의미한다.

#### 3단계> Transform

2단계를 거쳐 canonicity가 확립된 tree를 구성하는 각각의 vertex를 transistor pair로 치환한다. [그림1] 여기서  $x'$  등의 상보적 입력은 inverter를 이용하여 얻어진다.

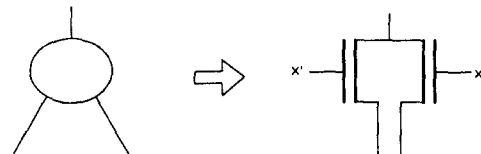


그림 1 정점과 트랜지스터의 대응

Fig. 1 Vertex-transistor mapping

( $x$ 는 주어진 함수의 입력변수)

‘확장’과 ‘연결’을 거친 후 tree의 bottom vertex가 가질 수 있는 bits-array는 (00), (01), (10), (11)의 4가지가 있을 수 있다. 그 각각의 경우에 대해서

→ (00)이면 해당 vertex를 없애고 GND에 연결한다.

(01)이면 해당 vertex를 없애고 complementary input을 연결한다.

(10)이면 해당 vertex를 삭제하고 해당 input을 연결한다.

(11)이면 해당 vertex를 삭제하고 output 부분에  $V_{dd}$ 를 연결한다

주어진 함수의 Complementary output이 나오도록 입력변수를 인가한다.

### 2. 2 MBDD transistor matching

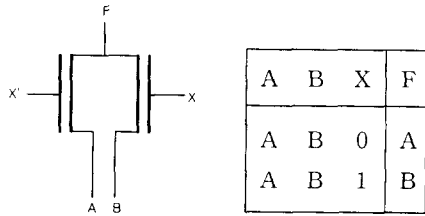


그림 2 변수에 따른 회로의 출력  
Fig. 2 Output for variables

이상에서 Complementary output은 control signal의 위치를 바꿈으로서 ( $X \leftrightarrow X'$ ) 구현될 수 있다. 따라서 임의의 function을 나타내는 BDD의 vertices의 개수를 MBDD를 이용하여 줄이는 것은 그 숫자의 2 배에 해당하는 Transistor의 개수를 줄이는 효과를 낸다.

검증을 위해 MBDD를 Full adder와 4-2 compressor에 적용하였고 다른 logic 합성법을 이용한 경우와 비교하였다. 이는 여타 보고된 NMOS 트랜지스터를 기반으로 설계 및 합성방법 [4],[5] 등에서 제시한 것과 같은 회로로 비교하기 위해서이다.

## 3. 적용

### 3. 1 전가산기 구현에의 적용

input A, B, C; output : Sum, Carry;  
진리표로부터, [Sum] = [01101001], [Carry]=[00010111]의 bit array를 얻을 수 있다.

#### 3.1.1 F<sub>sum</sub>의 구현

1단계> Sum의 bits-array를 top vertex에 할당하고 bottom vertex가 될 때까지 확장한다. 3 input이므로 3단의 vertex tree가 된다.

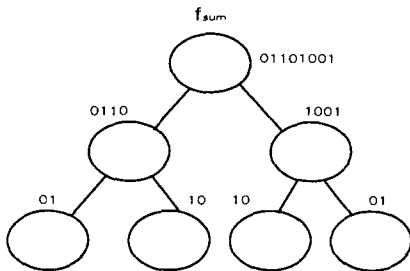


그림 3 MBDD 1단계 - BDD합성  
Fig. 3 The 1st step of MBDD - BDD synthesis

2단계>마지막 단의 첫 번째와 네 번째, 두 번째와 세 번째가 같은 bits array를 가지므로 각각 연결하고 하나의 vertex를 삭제한다.

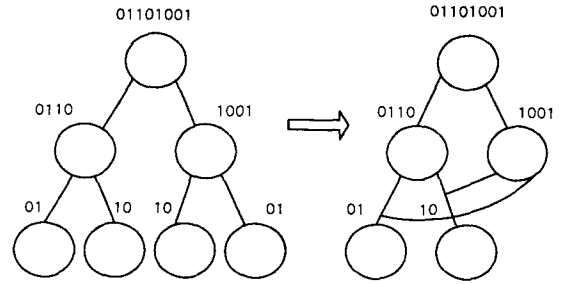


그림 4 MBDD 2단계 -연결  
Fig. 4 The 2nd step of MBDD - Connection

2단계가 끝난 상태의 이 형태는 Full adder의 sum에 대해서 유일한 형태를 가지는 것이다. 이는 full adder의 구현 방법에는 이 방법 외에도 여러 방법이 있겠으나, MBDD를 이용한 구현은 이 형태 하나만을 갖게된다.

#### 3단계> 치환

2 단계를 거친 이 정점 (vertex)과 연결상태를 앞에서 보인 두 개의 NMOS 트랜지스터로 치환하고 3단계에 따라서 연결한다.

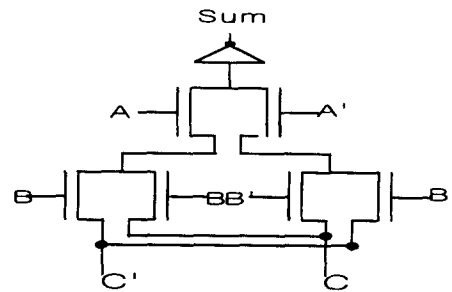


그림 5 정점-트랜지스터 변환  
Fig. 5 Vertex-transistor transformation

#### 3.1.2 F<sub>carry</sub>

Carry에 대해서도 sum 블록을 합성할 때와 같은 단계를 갖는다.

#### 1단계>

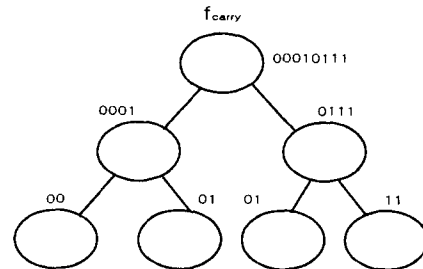


그림 6 Carry를 합성하기 위한 MBDD 1단계  
Fig. 6 The 1st step of MBDD for Carry synthesis

2단계> 세 번째 단의 가운데 두 vertices 가 서로 같은 array를 가지므로 이 둘을 연결하고 삭제한다.

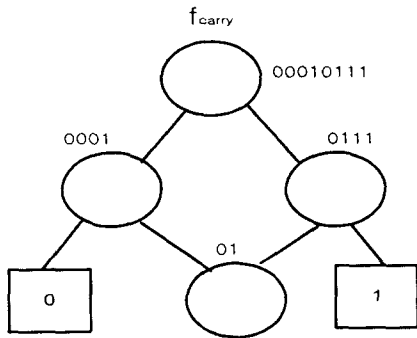


그림 7 Carry를 합성하기 위한 MBDD 2단계  
 Fig. 7 The 2nd step of MBDD for Carry synthesis

3단계> vertex를 트랜지스터로 바꾸면 다음과 같다.

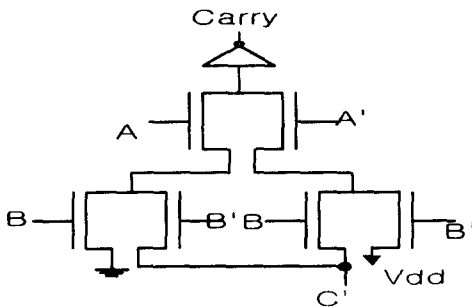


그림 8 Carry를 합성하기 위한 MBDD 3단계  
 Fig. 8 The 3rd step of MBDD for Carry synthesis

3. 1. 3 Sum 과 Carry를 동시에 구현

공통인 (같은 bits array를 갖는) 정점을 최대한 연결하기 위해서, Sum에 대한 정점들과 Carry에 대한 정점 중 같은 Bits array를 갖는 것을 연결하고 하나를 생략한다.

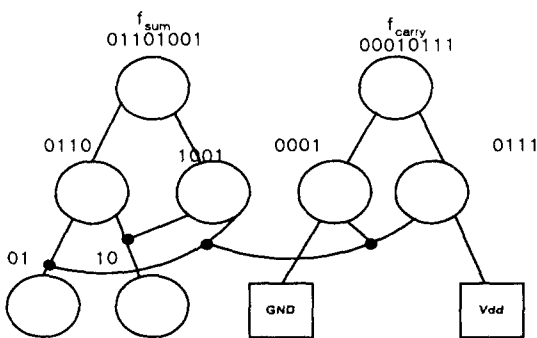


그림 9 MBDD로 표현된 Full Adder  
 Fig. 9 MBDD of Full Adder

최종적으로 sum과 carry를 나타내는 MBDD를 동시에 나타내면 추가적으로 공통인 array를 갖는 verex를 찾을 수 있으며 이 과정에서 같은 bits array를 갖는 vertex가 추가적으로 줄어들어 결과적으로는 2개의 Transistor 감소 효과를 가져온다.

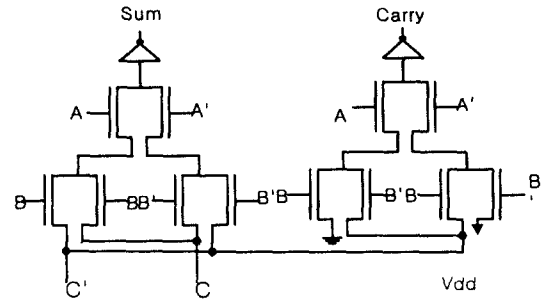


그림 10 MBDD를 이용하여 합성된 전가산기  
 Fig. 10 Synthesized Full adder using MBDD

표 1 합성방법별 트랜지스터 개수의 비교 I  
 Table 1 The comparison of transistor numbers among synthesis methods I

방법	MBDD	CMOS	CPL	LEAP
개수	22	32	28	24

3. 2 Compressor의 합성

- 위와 같은 방법으로 MBDD를 4-2 compressor에 적용하여 합성하였다.

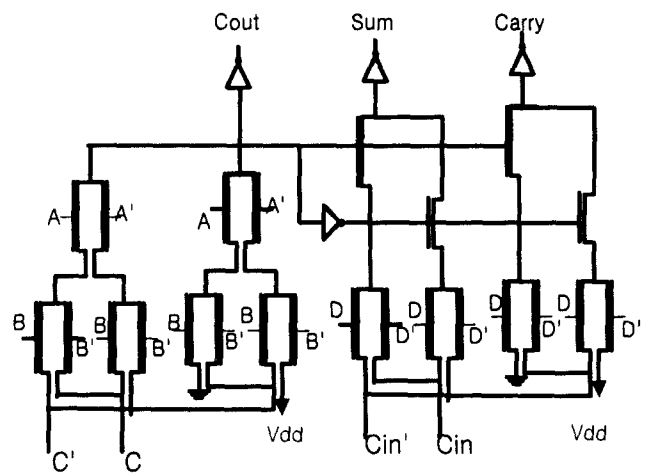


그림 11 MBDD를 이용한 4-2 Compressor  
 Fig. 11 4-2 compressor using MBDD

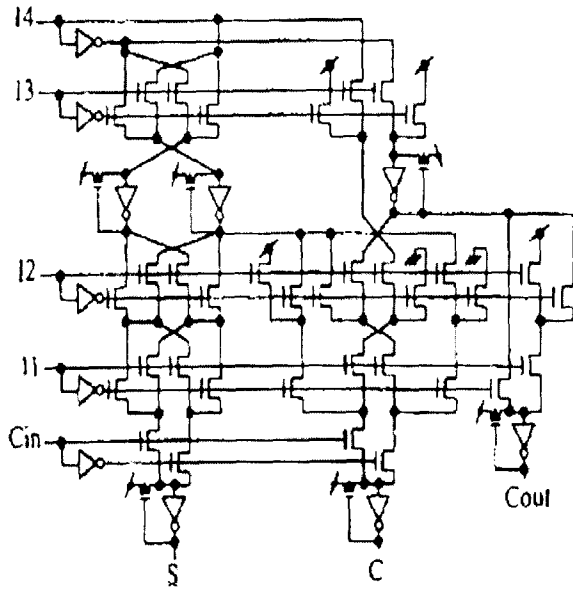


그림 12 LEAP를 이용한 4-2 compressor  
Fig. 12 4-2 compressor with LEAP

4. 비교결과 및 결론

MBDD를 4-2 compressor에 적용하여 합성하였다. 표1와 표2에서 보이는 것처럼 Transistor의 개수 면에서 여타 logic으로 회로를 합성하는 경우 보다 작은 수로 같은 함수가 구현되었고, power consumption의 경우와 Delay에 있어서 우수한 성능을 보였다. Power consumption의 경우 LEAP 대비 19.94%의 감소를 보였고, Transistor 개수에 있어서는 31.2%의 감소를 보였다. 그림13에 각 합성방법별 power-delay product를 나타냈다.

표 2 합성방법에 따른 트랜지스터의 개수비교 II  
Table 2 The comparison of transistor numbers among synthesis methods II

방법	MBDD	CMOS	CPL	LEAP[5]
개수	44	56	64	74

MBDD를 이용하여, 임의의 함수의 출력으로서, 그에 해당하는 Bits array가 주어졌을 때 그 Bits array를 비교하고 알고리즘에 따라 정렬한 후 회로의 연결상태로 나타내는 자동설계가 이루어 질 수 있다. 이를 바탕으로 저전력 라이브리 셀이 만들어 질 수 있을 것이다. 이 논문에서는 보상단으로서 가장 간단한 형태인 단순한 인버터를 사용하였으나 여타 보고서에 언급된 것처럼 여러 형태의 보상단 (Level Restoration Block)이 있으며, NMOS logic이 가지는 취약점인 load capacitance에 대한 비교적 큰 의존성을 해결하기 위한 보상단에 대한 연구가 진행될 수 있을 것이다.

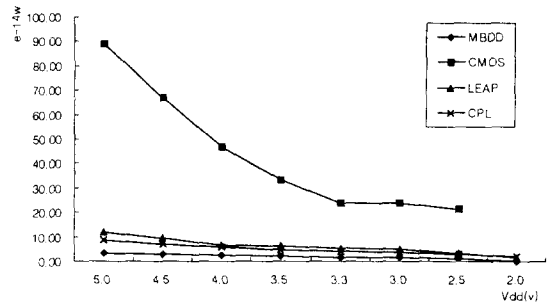


그림 13 전력-지연시간 곱  
Fig. 13 Power-delay product

감사의 글

본 연구는 1998년 인하대학교 교내연구비 지원을 받아 수행되었습니다.

참 고 문 헌

- [1] A. Ghosh, S. Devadas, K. Keutzer and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," ACM/IEEE Design Automation Conf., pp. 253-259, 1992.
- [2] F.N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," IEEE Trans. on VLSI Systems, pp. 446-455, Dec. 1994.
- [3] J. Monteiro, S. Devadas, and B. Lin, "A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits," ACM/IEEE Design Automation Conf., pp. 12-17, 1994.
- [4] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A Monte Carlo Approach for Power Estimation," IEEE Trans. on VLSI systems, vol. 1, No. 1, pp.63-71, March 1993.
- [5] A. Papoulis, Probability, Random Variables, and Stochastic Processes, 3rd Edition, New York: McGraw-Hill, 1991.
- [6] Reto Zimmermann and Wolfgang Fichtner, "Low-power logic styles:CMOS versus Pass-transistor logic", IEEE Journal of Solid-state circuits, VOL. 32, No.7, July 1997
- [7] Kazuo Yano, Yasuhiko Sasaki, Kunihito Rikino and Koichi Seki, "Top-down pass transistor logic design", IEEE Journal of Solid-state circuits, VOL. 31, No.6, June 1996

저 자 소 개



윤경용 (尹京鏞)

1973년 3월 1일생. 1999년 인하대 전자재료공학과 졸업. 1999년~현재 동 대학원 전자전기컴퓨터공학부 미디어시스템전공 석사과정

Tel : 82-32-874-1663, Fax : 82-32-875-5882

E-mail: kyyoon@pluto.emde.inha.ac.kr



정덕진 (鄭德鎭)

1948년 2월 8일생. 1970년 서울대 전기공학부 졸업. 1984 미국 Utah State university(식사) (Electrical Eng.) 1988년 미국 University of Utah (공박) (Electrical Eng.). 1989~ 현재 인하대 전자전기컴퓨터 공학부 반도체전공 교수

Tel : 82-32-874-1663

E-mail: djchung@inha.ac.kr