論文 89-26-1-11

VLSI 게이트 레벨 논리설계 최적화를 위한 Rule-Based 시스템

(A Rule-Based System for VLSI Gate-Level Logic Optimization)

李 省 奉*, 鄭 正 和*

(Seong Bong Lee and Jong Wha Chong)

要 約

본 논문에서는 게이트 레벨에서 논리 최적화를 하기 위한, 새로운 시스템을 제안한다. 본 시스템은 회로의 일부분을 간략화된 등가회로로 대치하는 local transformation을 rule로 표현한 rule-based 시스템이다.

본 시스템에서는 효율적인 패턴매칭을 위해, 'rule의 일반화'와 '국소최적화'를 제안한다. Rule의 일반화는 패턴매칭시 회로탐색을 줄이기 위해 사용되며, 국소최적화는 불필요한 회로탐색을 배제하기 위해 사용된다. 또한, 불필요한 패턴매칭 시도를 줄이기 위해, 회로패턴의 매칭순서를 rule 기술에 포함시킨다.

또한, 본 시스템을 하드웨어 컴파일러에 의해 생성된 논리회로 최적화에 적용하여, 그 효용성을 보인다.

Abstract

A new system for logic optimization at gate-level is proposed in this paper. This system is rule-based, in which the rules represent the local transformation replacing a portion of circuits with the simplified equivalent circuits.

In this system, 'rule generalization' and 'local optimization' are proposed for effective pattern matching. Rule generalization is used to reduce the circuit-search for pattern matching, and local optimization, to exclude unnecessary circuit-search. In addition, in order to reduce unnecessary trial of pattern matching, the matching order of circuit pattern is included in the rule descriptions.

The effectiveness of this system is shown by its application to the circuits which are generated by a hardware compiler.

I. 서 론

VLSI의 집적도가 증가함에 따라 설계 시간의 감소와 설계의 정확성을 보장할 수 있는 설계 자동화 시스템에 대한 연구가 활발히 진행되고 있다. 이에

*正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
接受日字 : 1988年 1月 6日

따라 하드웨어 컴파일러나 실리콘 컴파일러 등의 여러 설계 자동화 시스템이 제안되고 있으나, 이들에 의해 설계된 논리 회로도의 칩 면적은 전문가에 의한 설계에 비해 매우 크므로 칩 면적 최소화를 위한 연구가 필수 불가결하게 되었다.

기존의 게이트레벨 논리최적화에 대한 연구로는 Boolean 함수최소화[1]가 있지만 계산시간이 많이 걸리고, 실현 technology를 고려하기가 어렵다는 단점이 있다. 최근 이를 고려하여, 회로의 일부분을 간략화된 등가회로로 대치하는 local transformation[2]이 제안되었고, 이를 이용한 설계최적화 시스템으로는 LSS[3]와 SOCRATES[4,5] 등이 있다. 특히 SOCRATES는 local transformation을 rule로 구성한 rule-based 시스템으로, rule의 변경과 추가에 의해 시스템의 확장이 용이하다는 장점이 있다.

일반적으로, local transformation에 의한 회로최적화에서 가장 중요한 사항은 rule의 적용순서와 패턴매칭이다. 전자는 최적화 결과에, 후자는 최적화 시간에 큰 영향을 미친다. 전자를 해결하기 위해, SOCRATES는 lookahead 전략과 metarule의 사용을 제안하여, 그 효용성을 입증하였으나, 후자에 대한 고려가 미흡하여 최적화 시간이 많이 소요되는 문제점이 있다.

본 논문에서는 패턴매칭을 고려한 새로운 rule-based 시스템을 제안한다. 본 시스템에서는 효율적인 패턴매칭을 위해, 'rule의 일반화'와 '국소최적화'를 제안한다. Rule의 일반화는 패턴매칭을 위한 회로탐색을 줄이기 위해, 국소최적화는 불필요한 회로탐색을 배제하기 위해 사용한다. 또한, 불필요한 패턴매칭 시도를 줄이기 위해, 회로패턴의 매칭순서를 rule 기술에 포함시킨다. 또한, 본 시스템을 하드웨어 컴파일러에 의해 생성된 논리회로 최적화에 적용하여, 그 효용성을 보인다.

Ⅱ. 논리최적화 시스템

전문가에 의한 논리설계 최적화 작업은 전문가의 경험과 지식을 바탕으로 하여 redundant한 회로를 찾아내고 이를 간략화된 등가회로로 변경함으로서 이루어진다. 이와 같이 redundant한 부분을 간략화된 등가회로로 변경하는 것을 local transformation[2]이라 부르며 이것의 장점은 회로 전체를 간략화의 대상으로 하지 않기 때문에 계산시간이 짧어 VLSI에 쉽게 확장시켜 사용할 수 있다.

위의 설계전문가에 의한 설계최적화 과정은 설계자의 지식을 내포하는 local transformation을 IF(redundant한 회로 패턴) THEN(간략화) 형태의 rule로 표현하고 인간에 의한 redundant한 회로탐색을 패턴매칭으로 모델링함으로써 간단히 rule-based 시스템으로 모델화할 수 있다.

본 시스템은 그림 1과 같이 크게 4개로 module로 구성된다.

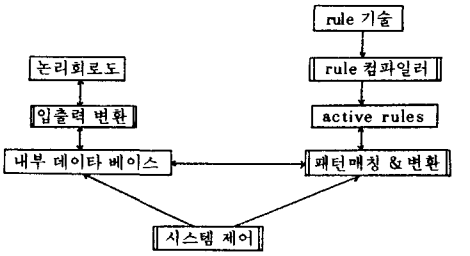


그림 1. 시스템 구성도
Fig. 1. System block diagram.

1) NDL (network description language)[6]로 기술한 논리 회로도를 내부 데이타 구조(internal data structure)로 변환하는 module

2) Rule 기술로부터 내부 active rule 형태로 변환하는 module

3) Rule에 주어진 패턴과 data base에 저장된 내부 데이타 구조와 매칭을 시도하고, 내부 데이타 구조를 변경하는 module

4) 적용할 rule의 선택 및 rule의 특성에 따른 회로 탐색을 제어하는 시스템 control module

이와 같이 rule-based 시스템을 구성하면 설계전문가의 경험과 지식을 rule에 포함시켜 설계에 반영할 수 있으며, 새로운 rule의 추가 및 변경을 행할 수 있어 시스템의 확장 및 변경이 용이하다는 장점이 있다.

한편, rule-based 시스템에 의한 회로의 최적화 과정을 간략하게 나타내면, 그림 2와 같다.

```
회로최적화 ( )
{
  for(모든 rule)
    for(회로 내의 모든 게이트)/*회로탐색*/
      if(패턴 매칭)
        rule 적용;
}
```

그림 2. 최적화 과정
Fig. 2. Optimization flow.

여기서, 패턴매칭은 최적화 실행시간의 대부분을 차지하기 때문에, rule-based 시스템의 최대 단점인 과다한 실행시간의 단축을 위해서는 패턴매칭 시도의 최소화가 필요하다. 이를 위해서는 최적화 결과에 영향을 주지 않는 범위내에서, i) rule 수의 최소화, ii) 패턴매칭시 불필요한 회로탐색의 배제, iii) 효율적인 패턴매칭이 요구된다.

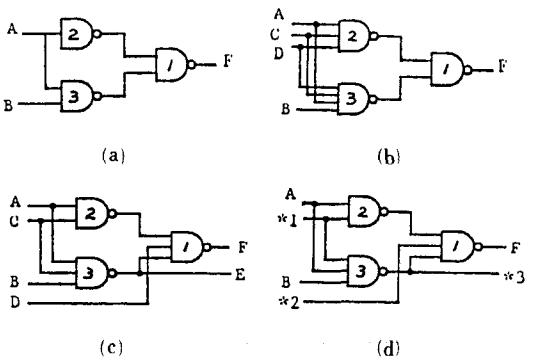
Ⅲ. Rule과 패턴매칭

1. Rule의 일반화(generalization)

Rule의 일반화란, 회로패턴이 서로 유사하여, 동일한 최적화 과정을 갖는 여러 개의 rule들을 통합하여, 1개의 rule로 구성하는 것을 말한다. 예를 들어, 그림 3의 (a), (b), (c)는 Boolean식 a+ab=a를 적용하여 회로를 간략화 할 수 있는 회로패턴의 일부를 보인 것이다. 이때, 이 3개의 회로패턴은 fanin, fanout 수에서의 차이점 외는 매우 유사한 회로패턴이며, 또한 동일한 최적화 과정, 즉 게이트 1과 게이트 3을 연결하는 신호선을 제거하는 과정을 거쳐, 간략화된다. 이러한 회로패턴들을 1개의 기본 회로패턴(그림 3의 (d))으로 통합할 수 있다. 여기서 *가 붙은 신호선은 0개 이상의 신호선 그룹을 나타낸다. 그림 3의 (d)에서 신호선 그룹 *1, *2, *3를 0개의 신호선으로 보면, 즉 이들 신호선을 제거하면, 그림 3의 (a)가 된다. 또 신호선그룹 *1을 b, c의 2개의 신호선으로 보고, 신호선그룹 *2, *3을 제거하면 그림 3의 (b)가 유도된다(이때, 신호선 b는 d로 대치). 같은 방법으로 그림 3의 (c) 도 유도할 수 있다. 즉 그림 3의 (d)으로부터 이 3개의 회로패턴을 유도해 낼 수 있다. 역으로, 그림3의 (d)는 그림 3의 (a), (b), (c)를 하나로 통합시킨 회로패턴임을 알 수 있다.

본 시스템에서는 그림 3의 (d)와 같은 일반화된 회로패턴에 대해 rule을 기술할 수 있다. 기존의 시스템[5]으로는, fanin 수를 4이하로 제한한 경우, 그림 3의 (d)의 회로패턴에 대해 모두 9(=3×3) 개의 rule을 구성하게 된다. 따라서 각 rule에 대해 최소 1회 이상의 회로탐색이 필요하므로, 모두 9회 이상의 전체 회로탐색을 행하게 된다. 그러나, 본 시스템에서는 rule의 일반화를 통해, 1번의 회로탐색으로 동일한 회로 개선이 가능하다.

Rule의 일반화는 최적화 결과에 영향을 주지 않고, rule의 수를 줄일 수 있게 한다. 이러한 rule수의 감소는 패턴매칭시 전체회로의 탐색을 줄이게 되어, 결과적으로 최적화실행시간을 대폭 단축시킨다. 그림 4는 일반화된 회로패턴의 일부를 보인 것이다. 그



(a) (b) (c) : non-generalized
(d) : generalized

그림 3. 회로 패턴의 예
Fig. 3. Examples of circuits patterns.

러나, rule의 일반화는 rule의 우선순위를 결정하기 어렵게 하는 문제점이 있다. 이러한 단점을 보완하기 위해, 본 시스템에서는 회로의 개선도가 비슷한 rule들을 1개의 rule로 일반화시키는 방법을 이용하고 있으며[6], lookahead 전략(Ⅲ-2절 참조)을 사용하고 있다.

2. Rule의 적용 및 제어

Rule-based 시스템에서 rule의 적용순서는 회로의 개선정도에 많은 영향을 미치지만, 일반적으로 이것은 회로에 따라 다르기 때문에 그 순서를 명확히 결정하는 것은 매우 어렵다. 그러나, 일반적으로 회로 최적화는 어떤 한계조건(최대 허용 칩면적 등)을 만족하는 회로를 구하는 것이므로, 이 경우의 최적화된 회로는 다수가 존재할 수 있다. 따라서 이들 중 어느 하나를 빠른 시간내에 찾는 것이 중요하다.

본 시스템에서는 기본적으로 rule의 우선 순위에 따라 rule를 적용한다. 이러한 rule의 우선 순위는 rule이 적용되었을 경우 기대되는 회로의 개선도 감소하는 게이트 수와 신호선수를 기준으로 하여 설정한다. 또한, 이를 지원하기 위해, lookahead 전략과 metarule을 사용한다.[4,5]

Lookahead 전략이란 주어진 rule에 대해 2개 이상의 회로부분에서 패턴매칭이 이루어지는 경우가 발생하면 각각의 경우에 대해 rule을 적용한 다음, 그 이후의 rule의 적용사항을 검토하여 최적이라고 평가한 부분을 선택하는 방법이다. 이는 rule의 적용순서를 명확히 결정할 수 없는 경우에 매우 유용한 방법으로, rule의 일반화에 따른 우선순위 결정의 어려움을 해소할 수 있다.

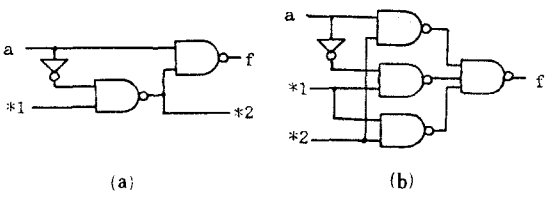


그림 4. 일반화된 rule의 회로패턴의 예
Fig. 4. Circuit pattern examples of generalized rule.

Metarule은 rule 적용을 제어하는 rule을 말한다. 여기에서는 lookahead 전략에 따른 최적화 실행시간의 단축을 위해, lookahead의 정도(미리 적용시켜 보는 rule의 수 등)를 조절하기 위해 사용된다.

3. 국소최적화[6]

우선순위에 의해 rule을 적용할 경우, n번째 우선순위의 rule이 적용될 때의 전체 회로는 크게 3부분으로 나눌 수 있다(그림 5). 영역 a는 1부터 (n−1)번째의 rule들이 이미 적용되어, 이들의 회로패턴이 매칭될 수 없는 부분이다. 또 영역 b는 n번째의 rule이 적용된 부분이다. 이때, 영역 c는 이 n번째의 rule이 적용됨에 따라, 회로가 변경되어 1부터 n번째의 rule들의 회로패턴이 매칭될 가능성이 있는 부분으로, 영역 b를 포함한 점선부분이다.

국소최적화란, 우선순위가 낮은 rule에 의해 회로가 변경되면, 변경된 부분의 주위(영역 b)에 대해서만 우선순위가 높은 rule을 재적용하는 것을 말한다. 이는 회로의 변경에 의해 우선순위가 높은 rule의 회로패턴이 발생하는 경우를 고려하기 위해서이다. 이 국소최적화는 변경된 이외의 부분(영역 a)에 대한 불필요한 회로 탐색을 피할 수 있기 때문에 실행시간을 대폭 단축할 수 있다.

또한, 기존의 시스템에서는 최적화 과정(그림 2)을 여러 회 반복하여 회로를 최적화하지만, 본 시스템에서 제안하는 국소최적화를 사용하면, 그림 2의 최적화 과정을 1회 적용으로 동일한 최적화 결과를 얻을 수 있다는 장점이 있다. 이에 따른 최적화 실행시간의 대폭적인 감소를 기대할 수 있다.

4. 패턴 매칭

전체 회로내에서 redundant한 회로패턴을 찾아 내는 패턴매칭은 실행시간의 대부분을 차지한다. 이는 subgraph isomorphism 문제로서 NP-complete문제이다.[7] 따라서 각 회로패턴에 따른 휴리스틱한 매칭방법이 요구된다. 그러나, 이러한 매칭방법은 회로패턴에 따라 서로 다르기 때문에, 각 회로패턴에 따른 최적의 매칭방법을 rule에 포함시켜 rule을 기술하는 것이 최적이다.

그러나, 실제 rule의 회로패턴이 그리 복잡하지 않기 때문에, rule 기술의 난해성을 피하기 위해, 본 시스템에서는 각 회로패턴에서 패턴매칭을 시도하는 게이트의 순서를 rule에 포함시켜 rule을 기술하도록 한다. 그 순서는 게이트에 대한 제약조건(fanin, fanout수 등)이 가장 매칭되기 어려운 순서이다. 이것은 실제 회로에서 패턴매칭이 이루어지는 부분이 매우 작기 때문이다. 따라서, 이 방법은 매칭되지 않는 부분에 대한 매칭시도를 감소시키기 위해서이다. 이것은 특히, rule의 일반화에 따른 패턴매칭의 복잡도를 줄일 수 있다. 예를 들어 그림 6과 같은 일반화된 rule의 회로패턴에 대해, 게이트 2부터 매칭을 시도하는 경우에는 게이트 1과 게이트 3을 매칭시키기 위해, 최대 fi2×(fi2−1)번(fi2 : 게이트 2의 fanin수)의 경우에 대해 게이트 1과 게이트 3의 연결관계를 조사하게 된다. 그러나, "게이트 1의 fanout수가 1이어야 한다"는 정보를 이용하여 게이트 1를 먼저 찾는 경우에는, 게이트 2를 바로 찾을 수 있기 때문에, 게이트 3과 게이트 1의 연결관계를 최대 fi2-1번 조사함으로써, 패턴매칭이 가능하다.

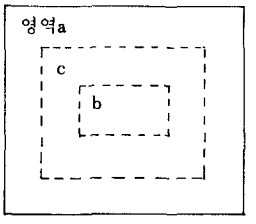


그림 5. Rule의 적용시 회로의 3영역
Fig. 5. 3 regions of circuits at the rule applicaton.

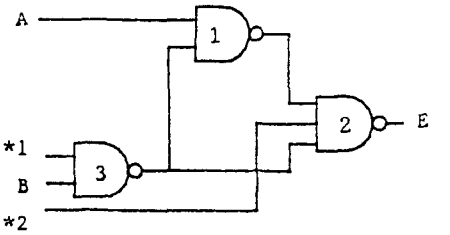


그림 6. 패턴 정보를 이용한 패턴매칭
Fig. 6. Pattern matching using the pattern information.

본 시스템에서는 이러한 패턴정보를 게이트의 제약조건으로 기술하며, rule의 패턴기술부에 기술한 게이트 순서대로 패턴매칭을 시도하도록 하여, 매칭 순서를 암시적으로 표현한다. 실제로 패턴매칭 시도는 전체 회로내의 모든 게이트에 대해 이루어지기 때문에, rule에 기술된 게이트 순서에 따른 패턴매칭은 이를 고려하지 않은 경우에 비해, 최적화 실행시간을 대폭 단축할 수 있다.

5. Rule의 기술

본 시스템에서는 그림 7과 같은 형태로 rule을 기술한다. 먼저 rule 특성 기술부에, rule의 우선순위 등의 rule 특성을 기술한다. 주어진 회로 전체내에서 찾고자 하는 부분의 회로 패턴은 그 기술과 이해가 용이한 변형된 NDL로 기술하며, 이 회로 패턴이 매칭되었을 경우 회로 변경 과정을 본 시스템에서 제공하는 기본 명령어로 기술한다.

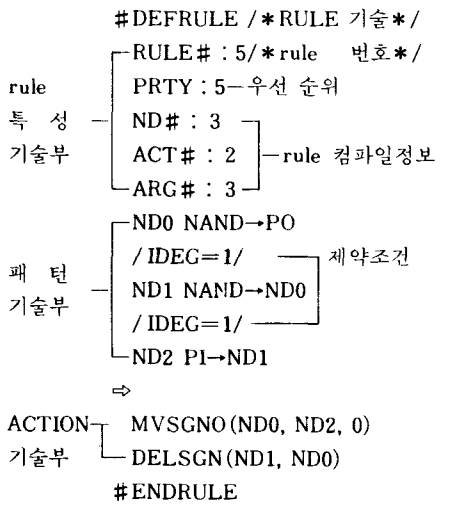


그림 7. Rule 기술의 예
Fig. 7. An example of rule description.

Ⅳ. 실험 및 고찰

본 방법의 유효성을 입증하기 위해 5개의 예제회로에 대하여 프로그램 실행을 행하였다. 여기서는 CMOS gate array의 macrocell[8]을 이용하여 회로를 최적화하였다. 본 시스템은 C언어로 프로그램되어 SSM-16의 UNIX v7하에서 실현하였다.

표 1은 여러 회로에 대한 실행결과를 보인 것이다. 표 1에서 회로의 크기는 게이트 수 및 신호선 수를 나타냈으며, 괄호안의 수치가 신호선 수를 나타낸다. 여기서 신호선의 수를 계산할 때 동일 포텐셜을 갖는 신호선일 경우라도 분기가 있으면 서로 다른 신호선으로 간주하여 구했다. 여기서, 게이트 수는 입출력 단자를 포함한 수치이다. 표 1에서 경우 1은 rule의 일반화와 국소최적화를 모두 사용한 경우이고, 경우 2는 rule의 일반화만 사용한 경우이며, 경우 3은 둘 모두를 사용하지 않은 경우로써 기존의 방법[5]만을 사용한 경우이다. 회로 #1, 회로 #3과 회로 #4는 제어회로이며, 회로 #2는 1bit 전가산기이고, 회로 #5는 4bit ALU이다. 경우 1과 2에 사용된 rule의 수는 18개이고, 경우 3에서는 41개이다. 그림 8은 예제회로 #2에 대한 결과를 보인 것이다.

표 1. 최적화 결과
Table 1. Optimization results.

예제회로	회로의 크기		실 행 시 간		
	최적화 이전	최적화결과	경우 1	경우 2	경우 3
회로# 1	18(23)	13(18)	4.6	5.0	11.6
회로# 2	19(24)	12(17)	5.0	5.2	12.0
회로# 3	29(42)	27(40)	5.0	6.3	12.0
회로# 4	72(92)	44(64)	13.4	18.1	40.9
회로# 5	128(188)	95(155)	23.6	33.8	73.9

표 1의 결과를 검토해 보면, rule의 일반화와 국소최적화의 사용으로, 실행시간이 대폭 감소함을 확인할 수 있다. 그러나, rule-based 시스템은 입력 회로와 사용된 rule에 따라 그 성능이 크게 차이가 있기 때문에, 입력회로에 대한 연구와 timing을 고려한 rule을 개발한 후, 재평가해야 할 것이다.

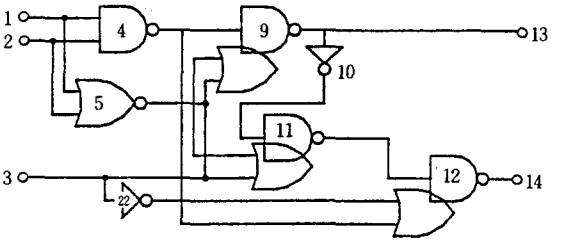


그림 8. 표 1의 회로 #2에 대한 회로도
Fig. 8. Circuit diagram of circuits #2 in table 1.

Ⅴ. 결 론

본 논문에서는 게이트 레벨에서 논리설계 최적화를 하기 위한 새로운 rule-based 시스템을 제안했다.

본 시스템에서는 rule-based 시스템의 최대단점인 실행시간을 단축하기 위해, rule의 일반화와 국소최적화를 제안했다. Rule의 일반화는 패턴매칭시 회로탐색을 줄이기 위해 사용되며, 국소최적화는 불필요한 회로탐색을 배제하기 위해 사용된다. 그리고, rule의 일반화에 따른 패턴매칭의 복잡도를 줄이기 위해, 회로패턴의 매칭순서를 rule에 포함시켰다. 프로그램 실험결과, 기존의 방법보다 실행시간이 단축되어 본 방법의 정당성이 입증되었다.

앞으로의 연구과제로는 timing을 고려한 회로최적화에 대한 연구와, metarule에 대한 보다 세부적인 연구가 계속되어야 할 것이다.

參 考 文 献

[1] R.K. Brayton, et al., "A comparison of logic minimization strategies using EXPRESSO," ISCAS, vol.3, pp.42-48, 1982.

[2] J.A. Darringer, et al., "Logic synthesis through local transformation," IBM J. Res. Develop., vol. 25, no. 4, pp. 272-280, 1981.

[3] J.A. Darringer, et al., "LSS: A system for production logic synthesis," IBM J. Res. Develop., vol. 28, no. 5, pp. 537-544, 1984.

[4] K. Garrison, et al., "Automatic area and performance optimization of combknatorial logic," ICCAD-84, pp. 212-214, 1984.

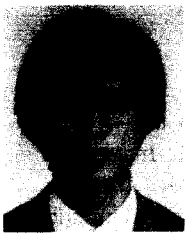
[5] W.W. Cohen, et al., "A rule-based expert system for optimizing combinational logic," *IEEE Design & Test of Computers, pp. 22-32*, Aug. 1985.

[6] 이성봉, "VLSI 논리설계최적화를 위한 Rule-Based System", 한양대학교 석사학위 논문, 12월 1985년.

[7] Garey, Michel R., "Computers and interactibility," Bell Lab., p. 202, 1979.

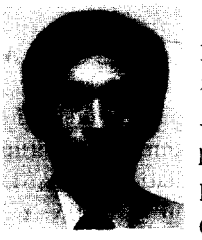
[8] CMOS Macrocell Manual, Goldstar Semiconductor, Ltd. *

著 者 紹 介



李 省 泰(正會員)

1962年生. 1984年 한양대학교 전자공학과 졸업. 1986年 2月 한양대학교 대학원 전자공학과 졸업, 공학석사학위 취득. 1986年 3月～현재 한양대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 VLSI CAD 특히 논리설계 및 검증 등임.



鄭 正 和(正會員)

1950年生. 1975年 한양대학교 전자공학과 졸업. 1981年 3月 일본 와세다대학 박사학위 취득. 일본 NEC(주) 중앙연구소 연구원, KIET 위촉 연구원, University of California, Berkeley 교환교수. 1981年～현재 한양대학교 전자공학과 부교수. 주관심분야는 VLSI CAD 특히 Layout 및 HDL 등임.