

타이밍 최적화 기술 매핑 시스템의 설계

(Design of a Time Optimized Technology Mapping System)

李相遇*, 黃善泳*

(Sang Woo Lee and Sun Young Hwang)

要約

본 논문에서는 조합 논리 회로의 타이밍 최적화와 동기 회로의 동기 시간 최적화 매핑을 효율적으로 수행하는 기술 매핑 시스템 설계 방식을 제안한다. 제안된 시스템은 조합 논리 회로의 동작 시간 최적화 매핑 수행을 위해 재매핑, 버퍼링 및 게이트 병합 방식을 사용하였고, 정확한 동작 시간 계산을 위하여 부하량을 고려하였다. 또한 동기 회로의 동기 시간 최적화 매핑을 위하여 임계 경로상의 특성을 고려한 레지스터의 위치 조정 휴리스틱 알고리즘을 제안하였다. MCNC 벤치마크에 대한 실험 결과, 제안 시스템이 mis-II에 비해서 보다 최적화된 매핑을 함을 확인하였다.

Abstract

This paper presents the design of a technology mapping system for optimizing delays of combinational and synchronous sequential logic circuits. The proposed system performs delay optimization for combinational logic circuits by remapping, buffering, and gate merging methods through the correct delay calculation in which the loading values are considered. To get time optimized synchronous sequential circuits, heuristic algorithms are proposed. The proposed algorithms reallocate registers by considering the critical path characteristics. Experimental results show that the proposed system produces a more optimized technology mapping for MCNC benchmarks compared with mis-II.

1. 서론

반도체 집적 기술의 발달로 작은 면적의 칩 안에 대규모 시스템의 구현이 가능하게 됨으로써 설계자의

기술을 만족시킨 상위 수준 합성 결과로부터 면적 및 동작 시간을 만족하는 논리 합성 과정이 매우 중요해지고 있다. 논리 합성은 회로의 redundancy를 제거하고 공통된 논리를 가진 부분을 찾아내어 하나의 부분으로 대치하는 기술 독립적 논리 최적화 과정^[1]과 회로를 구성하는 각 논리 소자를 주어진 라이브러리에서 제공하는 게이트만으로 변환하는 기술 의존적 논리 최적화 과정으로 구분되어진다. 시스템 설계시 주어진 라이브러리와 매핑 여부가 실제 논리 합성 결과를 크게 좌우함으로 기술 매핑 문제는 많은 관심의 대상이다.^{[13] [17]}

* 正會員, 西江大學校 電子工學科
(Dept. of Elec. Eng., Sogang Univ.)

※ 이 논문은 92년, 93년 교육부 학술 연구 조성비(과제번호 ISRC 92-E-0016, ISRC 93-E-2030)에 의하여 연구되었음.

接受日字 : 1994년 5월 27일

기술 매핑 과정은 논리 합성의 마지막 단계로 기술 독립적 논리 최적화를 거친 후의 논리 회로를 테크놀러지에 따른 라이브러리에서 제공하는 게이트들로 재구성, 레이아웃이 생성 가능한 논리 회로를 얻는 과정으로 최적화의 복잡함과 어려움 때문에 설계 자동화하려는 노력이 계속되어 왔다. [1], [5], [6], [9], [10], [16] 초기에 기술 매핑 문제는 최소의 면적을 갖는 논리 회로의 매칭에 주안점을 두어 왔으나 반도체 집적 기술의 발달로 면적의 최적화보다는 동작 시간 최적화에 초점이 맞춰지고 있다. 초기 기술 의존적 논리 최적화 시스템인 SOCRATES [11], [9] 와 LSS [5] 등은 규칙 기반 시스템으로 일련의 최적화 룰에 의하여 회로의 성능을 향상시킨다. 그러나 룰 적용 범위에 따른 국소 최적화의 한계성과 동작 시간을 고려한 룰 생성의 어려움 및 테크놀러지 변환에 따른 룰의 재조정 문제와 같은 여러 문제들 때문에 최적화의 한계를 가진다. 이러한 규칙 기반 시스템의 한계를 극복하기 위하여 DAGON 시스템 [10], [11] 에서 제안된 그래프 커버링 방식은 주어진 회로를 대상 그래프 (subject graph) 로 구성하고 라이브러리에서 제공되는 게이트의 패턴 그래프와 매칭되는 게이트 집합을 구성하여, 이 게이트 집합 중에서 대상 그래프 전체를 포함하여 최소의 비용을 가지는 게이트 집합을 선택함으로써 기술 매핑을 수행한다. 이러한 그래프 커버링 방식을 이용한 mis-II technology mapper 시스템 [6], [16] 은 다양한 설계 제약을 사용자가 줄 수 있어 면적 및 동작 시간 최적화가 가능하게 되었다. 이 시스템은 면적 최적화시 DAGON 시스템이 stem 노드를 경계로 설정하여 발생하는 면적 최적화의 한계를 극복하였으나 logic cone의 매핑 순서에 따라 면적의 손실이 발생할 수 있으며, 동작 시간 최적화시 부하량을 고려해 주지 못하여 실제 생성 회로의 정확한 동작 시간을 보증할 수 없는 한계성을 가진다. SiLOS-II 시스템 [20] 은 면적 최적화시 stem 노드를 경계로 설정하지 않고 logic cone에서의 매핑을 수행하여 stem 노드에서 발생할 수 있는 면적 손실을 줄이고 부분 최적화를 수행하여 cone ordering의 문제를 해결하였다. 또한 재매핑 및 버퍼링 방식을 도입한 동작 시간 최적화 수행시 부하량을 고려하여 정확한 동작 시간의 계산을 가능하게 하였다. 그러나, 재매핑 수행시 대상 노드가 전체 주기는 감소시키나 주어진 동작 시간을 만족하지 못하는 경우에 재매핑을 수행하지 못하여 재매핑 적용의 한계성을 보여 최소 동작 시간 커버링을 구하지 못하는 단점이 있다. 본 논문에서 제안된 동작 시간 최소화 알고리즘은 기존의 재매핑 방식의 적용 한계성을 극복하기 위하여 설계 제약으로 주어진

동작 시간외에도 전체 동작 시간을 감소하는 경우에도 재매핑이 가능하게 하였고, 버퍼링 방식과 게이트 크기 조정 (gate resizing) 기법을 이용한 게이트 병합 (gate merging) 방식을 추가로 적용하여 부하량을 고려한 논리 회로의 최소 동작 시간 매핑을 가능하게 하였다.

디지털 회로 설계는 동기 회로와 비동기 회로 설계로 구분된다. 비동기 회로 설계 방식은 회로의 구현 면적을 최소화할 수 있으며 다양한 회로 설계가 가능하다. 복잡하고 불규칙한 회로 패턴을 가지므로 자동화 설계 방식의 도입에 어려움이 있으며 대규모 시스템 설계시 설계 검증이 어려운 단점을 가지고 있다. 동기 회로는 일정한 주기를 가지는 클럭에 의하여 회로의 동작이 정의되며, 동기 회로 설계 방식은 레지스터간에 존재하는 조합회로와 레지스터와의 관계만을 고려하여 회로를 구성함으로써 대규모 시스템 설계의 복잡성이 감소하며 설계 검증이 용이하여 디지털 회로 설계의 주종을 이루어 왔다. 조합회로의 설계는 이단 [3] 혹은 다단 논리 최적화 [2], [4] 및 기술 의존적 최적화를 위한 많은 연구 [1], [5], [6], [9], [10], [11], [18], [20] 가 활발하게 진행되고 있다. 그러나, 동기 회로에 대한 연구는 레지스터와 feedback 연결에 대한 취급의 어려움 때문에 기존의 논리 최적화 방식들은 면적과 동작 시간 제약조건은 만족시키지만 회로의 동기 시간 최적화에 대한 고려가 미흡한 실정이다. 이에 대한 연구로 Leiserson과 Saxe에 의해 제안된 retiming 기법 [12] 이 있으며, 레지스터의 위치 조정에 따른 조합 회로의 변화를 이용하여 논리 최적화를 고려해 주는 연구가 보고되었다. [8], [15]

Leiserson과 Saxe에 의해 제안된 retiming 기법 [12] 은 주어진 회로가 가능한 한 최소의 클럭 주기를 가지도록 레지스터의 위치를 재조정해 주고 있다. 이들이 제안한 알고리즘은 모든 게이트간의 지연 시간을 계산한 후 이진 탐색을 하여 최소의 클럭 주기를 갖도록 레지스터의 위치를 재배열해 주고 있다. 이러한 알고리즘의 수행은 최소의 클럭 주기를 갖게 해주나, 알고리즘 복잡도가 높아 대규모 회로 설계시 탐색 영역이 크다는 단점을 가지고 있다. 레지스터의 위치 조정을 이용하여 새로운 조합 회로를 만들어 낸 후 기존의 논리 최적화 방식을 적용하는 방식 [8], [15] 은 기존의 논리 최적화 방법들이 레지스터의 위치에 제약을 받는다라는 단점을 해결해 주고 있으나, 부분적인 회로를 대상으로 논리 최적화를 수행하므로 부분적인 성능 향상은 기대할 수 있으나 전체 클럭 주기를 최적화해준다는 보장이 없다. 본 논문에서는 Leiserson과 Saxe에 의해 제안된 retiming 기법의

실행 시간 오버헤드를 줄이기 위해 임계 경로상의 특성을 고려한 동기 시간 최적화 적용과 주어진 회로의 클럭 주기를 최적화 시켜주도록 레지스터의 위치를 재조정해주는 휴리스틱 접근으로 보다 빠른 동기 시간 최적화 매핑을 가능하게 하였다.

본 논문에서는 기술 매핑 과정에서 조합 회로에 대한 동작 시간 최적화를 수행하며 동기 회로에 대하여 동기 시간 최적화 매핑을 수행하는 타이밍 최적화 논리 회로 기술 매핑 시스템의 실효성을 보인다. II 장에서는 타이밍 최적화 논리 회로 기술 매핑 시스템의 개관을 설명하였고, III 장에서는 동작 시간 최소화를 위한 알고리즘을 기술하였다. IV 장에서는 동기 시간 최적화 알고리즘을 보였고, V 장에서는 제안된 알고리즘의 효용성을 보이기 위하여 벤치마크 실험 결과를 분석한다. 마지막으로 VI 장에서는 결론 및 추후로 연구되어야 할 사항을 제시한다.

II. 시스템 개관

동작 시간 및 동기 시간 최적화 기술 매핑을 수행하는 전체적인 시스템의 구성을 그림 1에 보였다. SiLOS-II는 다단 논리 최적화 시스템인 SMILE²²을 거쳐 기술 독립 최적화된 equation 형태 기술을

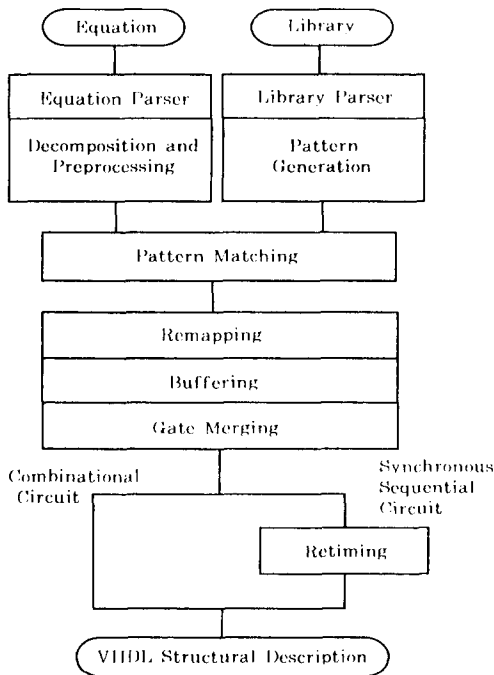


그림 1. 시스템의 구성도
Fig. 1. System flow diagram.

입력으로 취한다. Equation을 파싱하여 얻어진 네트리스트는 2-입력 NAND 게이트와 인버터로 구성된 DAG로 만들어지며, 전처리 과정에서 reconvergence를 검사한다. DAG 형태의 회로는 하나의 출력 노드를 루트 노드로 하는 모든 transitive fanin으로 구성된 logic cone들의 집합으로 인식되며 각 logic cone들은 라이브러리의 게이트를 나타내는 여러 이진 트리의 문자열과 비교하여 라이브러리 게이트로 매칭 가능한 모든 패턴을 찾아낸다. 이 과정은 기존에 구축된 논리 회로 기술 매핑 시스템인 SiLOS-II와 동일하다.¹²⁰ 동작 시간 최적화를 위하여 재매핑, 버퍼링 및 게이트 병합을 통하여 최소 지연 시간 혹은 설계자로부터 주어진 동작 시간 제약을 만족하도록 매핑을 수행한다. 만약 주어진 대상회로가 동기 회로인 경우에는 retiming을 수행하여 클럭 주기를 감소시켜주어 동기 시간 최적화 매핑 결과를 얻는다. 이와 같은 매핑 과정을 통하여 얻어진 최적화 논리 회로 결과는 VHDL 구조 기술¹⁴ 또는 그래픽 출력으로 나타내진다.

III. 최단 동작 시간을 위한 커버링

기술 매핑은 매칭된 모든 패턴들을 가지고 대상 회로를 커버할 수 있는 조합을 구성함으로써 완료된다. 이 때 대상 회로에서 각 노드마다 매칭되는 게이트 패턴은 유일하지 않으며 그 노드에 어느 게이트를 매핑시키는가에 의해 그 노드를 루트 노드로 하는 부분 트리 내의 노드들에 대한 매핑의 면적 및 지연 시간에 영향을 주므로, 한 노드에서의 매핑 문제는 부분 트리의 다른 노드들에서의 매핑을 함께 고려해 주어야 한다. 전체 최적화를 이루기 위해서는 회로를 커버할 수 있는 모든 패턴의 조합들을 조사하여 면적과 시간 제약 조건에 맞는 조합들 중에서 최적의 비용을 가지는 커버를 구해야 한다. 이 문제는 게이트의 내부 지연 시간 (gate intrinsic delay) 만을 고려할 경우 다이내믹 프로그래밍 기법으로 쉽게 해결할 수 있다.¹⁰¹ 그러나 기술 매핑 시스템에서는 동작 시간에 대한 정확한 예측이 가능해야 하며, 게이트 내부 지연 시간만을 고려하여 회로의 동작 시간을 계산할 경우 출력단의 부하로 인해 시간 제약 조건을 만족시키지 못하는 커버를 선택할 수도 있다. 따라서 사용되는 지연 시간 모델은 내부 지연 시간과 부하량을 동시에 고려해 주어야 한다.

1. 지연 시간 모델링

한 게이트에 신호가 인가되어 다음단의 게이트로

출력이 전달되기까지의 시간을 논리 소자의 지연 시간으로 정의하며 다음과 같은 3가지로 구성된다.

- 1) 게이트의 내부 지연 시간
- 2) 게이트의 출력단 fanout load에 의한 부하 지연 시간
- 3) 게이트간의 연결선 지연 시간 (wire delay)

본 논문에서는 게이트 g 의 지연 시간 계산에 논리 소자의 지연 시간을 결정하는 세가지 요소 중 게이트간의 연결선에 의한 지연 시간을 제외한 두가지 요소로 결정되며 이 식을 아래에 보였다. 게이트간의 연결선에 의한 지연 시간의 계산은 실제 배치 배선이 이루어지기 이전까지는 예측이 어려우므로 추후 post-layout 추출과 시뮬레이션을 통해 동작 시간 제약 조건의 만족 여부를 확인한다.

$$\begin{aligned} \text{게이트 지연 시간}(g) = & \text{내부 지연 시간}(g) \\ & + (\sum \text{입력 용량}(i) * \text{부하 저항}(g)) \\ & i \in \text{fanout}(g) \end{aligned}$$

2. 최단 동작 시간을 위한 커버링

대상 회로는 초기 매핑시 수행되는 decomposition 과정에서 임계 경로의 길이는 가능한 한 작게 줄인 형태이며²⁰⁾ 최단 동작 시간을 가지는 커버를 찾기 위하여 신호 대기 시간을 계산한다. 라이브러리를 구성하는 게이트와 지연 시간 모델을 이용하여 보다 정확한 신호 대기 시간을 구하고 최단의 신호 대기 시간을 가지는 커버를 선택한다. 내부 지연 시간만을 고려하여 커버링을 수행할 경우 단순히 대상 노드에서 최단의 신호 동작 시간을 가지는 커버를 선택하면 되지만 부하량을 고려하여 최단의 신호 대기 시간을 가지는 커버를 구하기 위해서는 출력단의 부하가 결정되기 전까지 선택할 수 없다. 기술 매핑 과정에서 부하량을 고려해 줄 때의 문제점은 대상 노드를 입력으로 하는 게이트의 입력 캐패시턴스에 따라 최소의 동작 시간을 가지는 커버가 결정되며 어느 게이트가 출력단에 매핑되는지는 주출력까지 기술 매핑이 수행되어야 결정되므로 stem 노드에서 매핑될 게이트의 출력이 구동시키는 게이트들의 입력 캐패시턴스 예측이 불가능하다는 점이다. 매핑을 끝낸 후의 결과 회로가 동작 시간 제약 조건을 만족하는가의 여부는 매핑이 이루어지는 과정에서 알 수 없다. 따라서 동작 시간 제약 조건 하에서의 매핑은 최소 면적의 커버나 최단 동작 시간의 커버를 구하여, 회로의 동작 시간에 대한 정보를 유지한 상태에서 이루어져야 한다. 최단 동작 시간의 커버를 먼저 구한 후 시간 제약 조

건을 만족시키는 범위 내에서 면적을 최소화시키는 방법은 임계 경로에 대해서 효과적인 방법일 수 있으나, 회로 전체의 면적을 고려하면 비임계 경로에서 낭비되는 면적의 양을 무시할 수 없으므로 비임계 경로에서의 최소 면적 커버링을 재수행하여야 하는 문제점을 가진다. 최소 면적 커버를 출발점으로 잡을 경우 비임계 경로는 이미 면적 최소화가 이루어진 상태이므로 임계 경로에 대한 동작 시간 최적화를 위한 재설계만이 필요하다. 제안된 시스템은 먼저 최소 면적 최적화를 수행한 후 시간 제약을 고려한 재설계를 수행한다.

임계 경로는 slack 계산에 의해 결정되며²¹⁾, 이의 제거를 위한 재설계의 방식으로 재매핑, 버퍼링 및 게이트 병합을 수행한다. 임계 경로상의 재설계 대상의 선택은 면적 및 최단 동작 시간에 큰 영향을 주므로 최소 면적 최적화 수행후 최소의 면적 증가만으로 최단 동작 시간을 가지는 회로를 구성하기 위해서 먼저 임계 경로에 대한 재매핑을 수행한다. 재매핑은 재설계 방안인 decomposition, compounding, inversion을 모두 포함하여 수행된다. 대상 노드에서 매핑 가능한 모든 패턴에 대해 부하량을 고려한 신호 대기 시간을 재계산하여 최단 동작 시간을 가지는 패턴으로 바꾸어 준다. 재매핑 과정은 주어진 라이브러리에서 제공되는 회로 패턴중 최소의 동작 시간을 갖는 논리 회로 설계를 가능하게 해준다. 재매핑 과정을 통하여 재설계 회로에는 부하량에 의해 발생하는 임계 경로가 있다. 이를 해결해 주기 위해선 버퍼링과 게이트 병합 과정을 거친다. 이 두가지 재설계 방식은 회로의 면적은 증가시키나 부하량에 의한 지연 시간을 감소시켜 준다. 먼저 재매핑을 통하여 재설계된 회로에 대하여 bottleneck analysis를 수행하여 버퍼링이 적용 가능한 대상 노드를 선택하고 버퍼링을 적용하여 동작 시간을 감소한다. 또다른 재설계 방식으로 임계 경로상에 인접한 동일 게이트를 하나의 게이트로 병합하여 게이트의 크기를 키워주는 게이트 병합 과정을 거쳐 최단 동작 시간을 갖는 논리 회로를 얻어낸다.

1) 재매핑

기존의 SiLOS-II 시스템에서 제안된 재매핑 방식은 대상 노드의 매핑 가능한 여러 패턴 중에 설계 제약으로 주어진 동작 시간을 만족하는 패턴이 존재하는 경우에만 재매핑을 수행하였다.²²⁾ 이러한 적용 방식은 여러 노드에서의 재매핑으로 발생하는 동작 시간 감소에 의한 설계 제약을 만족시킬 수 없으므로 동작 시간 향상에 한계를 보인다. 이러한 한계성을 극복하기 위하여, 제안된 재매핑 방식은 slack 계산

에 의해 선택된 임계 경로상의 모든 대상 노드에 대하여 매핑 가능한 여러 패턴들 중 최소의 지연 시간을 가지는 패턴을 선택함으로써 동작 시간을 감소시킨다. 한 노드에서의 매칭되는 패턴의 변화는 그 노드를 루트 노드로 하는 부분 트리내의 다른 노드들에 영향을 주므로 재매핑으로 발생하는 전체 회로 변화를 저장해 주어야 한다.

2) 버퍼링

버퍼링은 버퍼 자체의 내부 지연 시간이 회로의 동작 시간에 첨가되므로 대상 게이트가 구동시키는 비임계 경로 게이트의 slack 값이 인버터 쌍이나 버퍼의 첨가에도 (-)로 바뀌지 않아야 한다는 조건이 필요하다. 임계 경로를 구성하는 게이트들 중 부하로 인한 게이트의 지연 시간이 게이트의 내부 지연 시간을 초과할 경우나 게이트 출력단의 구동 게이트 중 일부만이 임계 경로에 속하여 게이트가 구동하는 출력 게이트들의 slack 편차가 큰 경우 게이트의 출력단에 구동 능력이 큰 버퍼나 인버터 쌍을 삽입, 비임계 경로를 버퍼의 출력에 할당하여 임계 경로의 부하량을 줄임으로써 임계 경로의 제거가 가능하다. 버퍼링 과정에서 사용되는 인버터는 cascade된 쌍으로 삽입되며, 구동 능력의 향상을 위해 출력단의 인버터 크기를 크게 하는 게이트 크기 조정 기법을 이용한다.

3) 게이트 병합

재매핑과 버퍼링을 수행한 이후의 회로에 대하여 동작 시간을 감소시키는 또다른 방법으로 게이트 병합이 있다. 임계 경로상에 인접한 두 노드가 동일 게이트인 경우에는 두 노드를 하나의 동일 게이트로 병합하여 게이트의 크기를 키워주어 부하량에 의한 지연 시간을 감소시켜준다. 그림 2에서 볼 수 있듯이 동일 게이트는 게이트의 크기 및 부하량에 따라 서로 다른 지연 시간을 가지게 된다. 이 때 내부 지연 시간은 X 혹은 Y점에 해당되는 지연 시간이며, 부하 지연 시간은 출력에 영향을 주는 신호가 어느 입력 단자와 연결되었는가에 따라 각기 다른 값을 가진다. 부하 용량이 큰 경로에서 인접한 두개의 동일 게이트를 하나의 게이트로 병합하여 부하 저항이 작은 게이트로 대체하여 부하 지연 시간을 줄일 수 있으나 게이트의 크기가 커지므로 내부 지연 시간이 증가하게 되므로 임계 경로상의 게이트를 병합할때는 이러한 요소를 고려하여 게이트 병합 여부를 결정해야 한다. 그림 2에서 Z점은 게이트 병합 여부의 결정점이 되며, Z점의 선택은 라이브러리에서 제공되는 rising time 및 falling time의 정확한 지연 시간 정보에 의해 결정된다.

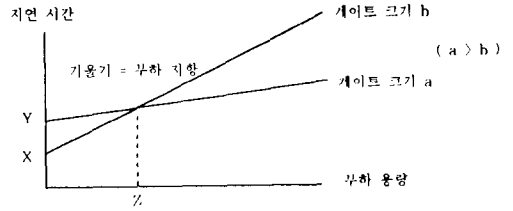


그림 2. 부하 용량에 따른 게이트의 지연 시간
Fig. 2. Delay vs. loading capacitance.

IV. 동기 시간 최적화

기존의 논리 최적화 방법들은 동기회로의 클럭 주기에 대한 고려를 해 주지 못하고 있다. 그림 3 (a)와 같은 회로에서 전체 주기는 지연시간이 가장 긴 조합회로에 의해 50ns로 결정된다. 상위 수준에서 주어진 시스템의 클럭 주기가 40ns이내로 주어진 경우에는 이와같은 회로 설계는 비록 면적 제약 조건을 만족하여도 동작 시간에 대한 재설계가 필요하다. 이 회로에 대하여 동기 시간 최적화를 고려해 주어 그림 3 (b)와 같은 결과를 얻는다면 상위 수준에서 주어진 면적과 시간 제약 조건을 모두 만족시킨 최적화 결과를 얻게된다.

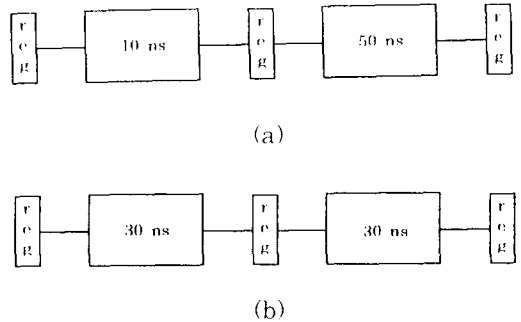


그림 3. 동기 시간 최적화 고려에 따른 주기 감축
(a) 동기 시간 최적화를 고려하지 않은 논리 합성 결과 (주기 50ns), (b) (a)에 동기 시간 최적화를 고려해 준 결과 (주기 30ns).

Fig. 3. Clock period reduction by synchronous time optimization.
(a) Synthesis result without time optimization (clock period 50ns). (b) Synthesis result after time optimization (clock period 30ns).

그림 4에 레지스터의 위치를 재조정해 주어 동기 시간 최적화를 이루는 타이밍 최적화 동기회로의 자동 생성시스템의 흐름도를 보였다. 본 시스템에서는 사용되는 레지스터는 edge-triggered F/F으로 가정하였다.^[12]

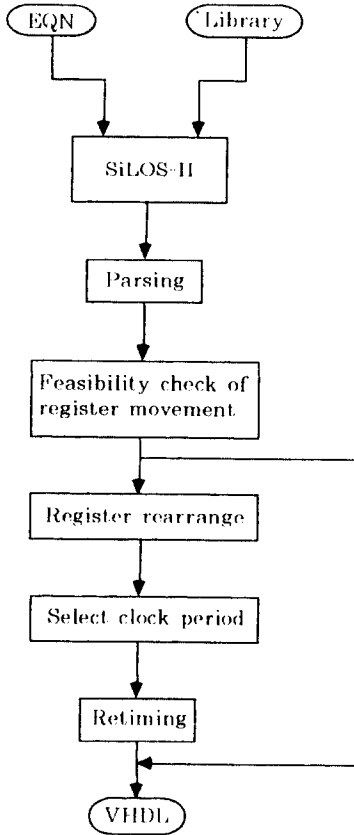


그림 4. 타이밍 최적화 동기 회로 자동 생성 시스템의 흐름도

Fig 4. System flow chart for automatic generation of time optimized synchronous logic.

기술 매핑을 통하여 얻어진 내부 자료 구조는 동기 시간 최적화에는 부적합하므로 새로운 자료 구조를 생성한 후, 동기 시간 최적화 적용 가능 여부를 검사하여 적용 불가능한 경우에는 초기 회로를 출력으로 내보낸다. 동기 시간 최적화가 적용가능한 경우에는 레지스터의 위치를 재조정해 주어 주어진 주기를 만족하도록 동기 시간 최적화를 적용하여 그 결과를 VHDL 구조 기술^[14] 또는 그래픽 출력으로 사용자에게 제공한다.

1. 동기 시간 최적화를 위한 자료 구조
기술 매핑을 통하여 얻어진 자료 구조를 토대로 하여 동기 시간 최적화에 필요한 W와 D를 계산하여 새로운 자료 구조를 갖춘다. W와 D는 Leiserson의 제안과 같이 다음과 같이 정의된다.^[12]

○ 주어진 경로 $p = V_0^{e_0} \rightarrow V_1^{e_1} \rightarrow \dots \rightarrow V_k^{e_k} V_k$

$$w(p) = \sum_{i=0}^{k-1} w(e_i) : \text{경로상의 레지스터 갯수}$$

$$d(p) = \sum_{i=0}^{k-1} d(v_i) : \text{경로의 지연 시간}$$

○ $W(u, v) = \min \{w(p) : u \rightarrow v\}$

노드 u에서 노드 v로의 경로중 최소의 레지스터 갯수

○ $D(u, v) = \max \{d(p) : u \rightarrow v \text{ and } w(p) = W(u, v)\}$

노드 u에서 노드 v로의 경로중 최소의 레지스터를 가지는 경로의 최대 지연 시간

2. 레지스터의 동기시간 최적화 적용 가능성 검사
주어진 회로에 따라서는 레지스터의 위치가 동기 시간 최적화 적용이 불가능한 경우가 있으며, 이런 경우는 초기 회로가 최소화된 클럭 주기를 가지고 있다. 주어진 회로에서 레지스터의 위치를 알아내고 동기 시간

```

check_reg_move ( C, num_gate )
/* C : 주어진 회로를 나타내는 자료 구조
   num_gate : 주어진 회로의 게이트 갯수 */
{
    ( reg, num_reg ) = find_reg_allocation ( C, num_gate );
    /* 주어진 회로에서 레지스터의 위치와 갯수를 검출하여
       reg와 num_reg로 return */
    if ( check_each_reg_move ( C, reg, num_reg ) == FAIL )
        return FAIL;
    /* 동기 시간 최적화 적용이 불가능한 경우 */
    else
        return SUCCESS;
}

check_each_reg_move ( C, reg, num_reg )
{
    for ( i=0 ; i<num_reg ; i++ )
    {
        if ( check_move_forward ( C, reg[i] ) == FAIL )
            /* 앞방향으로의 이동 여부를 검사 */
            if ( check_move_backward ( C, reg[i] ) == FAIL )
                /* 뒷방향으로의 이동 여부를 검사 */
                not_moved ++;
    }
    if ( not_moved == num_reg )
        return FAIL;
    /* 모든 레지스터가 이동 불가능한 경우 */
    else
        return SUCCESS;
}
  
```

그림 5. 레지스터의 동기시간 최적화 적용 가능성 확인 알고리즘

Fig. 5. Algorithm for feasibility check of register movement.

최적화 적용의 가능 여부를 검사하여 불필요한 수행을 방지하기 위한 알고리즘을 그림 5에 나타내었다.

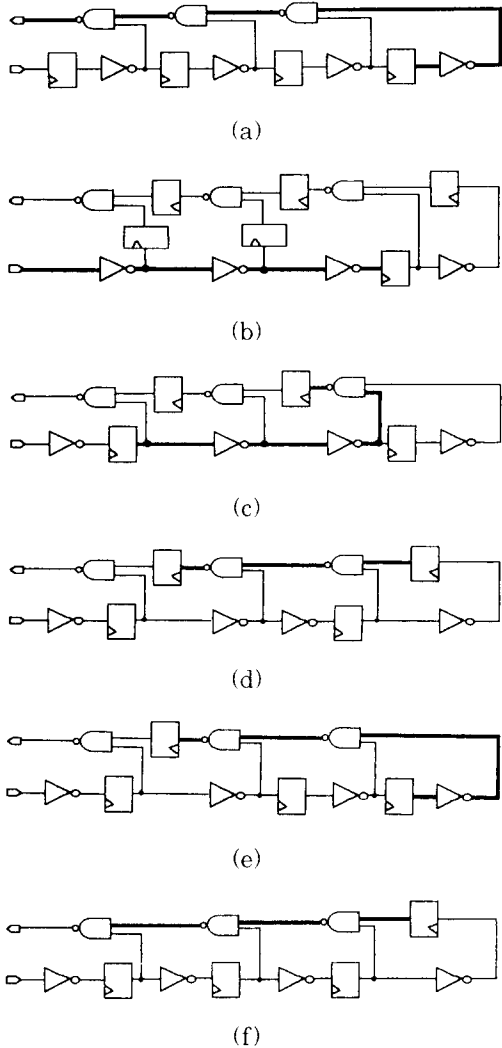


그림 6. 같은 기능을 갖는 회로들
 (a) 초기 회로 (주기 24ns) (b) 주기가 9ns인 경우 (c) 주기가 13ns인 경우 (d) 주기가 14ns인 경우 (e) 주기가 17ns인 경우 (f) 주기가 21ns인 경우

Fig. 6. Circuits with the same functionality.
 (a) Initial logic circuit (clock period 24ns). (b) Circuit with clock period of 9ns. (c) Circuit with clock period of 13ns. (d) Circuit with clock period of 14ns. (e) Circuit with clock period of 17ns. (f) Circuit with clock period of 21ns.

3. 레지스터의 위치 조정

그림 6에 동기 시간 최적화 적용으로 나타날 수 있는 다양한 회로 패턴에 대해 보이고 있다. 이 경우 지연 시간의 계산에는 내부 지연 시간만을 고려하였고, NAND 게이트는 7ns, Inverter는 3ns의 내부 지연 시간을 갖는다고 가정하자. 주기를 결정하는 경로는 굵은 실선으로 나타내었으며, 초기 회로 그림 6 (a)에 대한 동기 시간 최적화의 적용으로 그림 6 (b)의 9ns에서 그림 6 (f)의 21ns까지 여러가지의 주기로 표현될 수 있다. 이처럼 retiming의 적용으로 대상 회로는 여러가지의 주기를 갖는 회로로 표현될 수 있으므로 초기 회로에 대하여 직접 retiming 기법을 적용하여 이 중에서 동기 시간 최적화된 맵핑 결과를 얻는다는 많은 시간이 소모된다.¹² 본 논문에서는 이러한 오버헤드를 줄이기 위하여 레지스터의 위치를 가능한 주출력쪽으로 재조정하는 휴리스틱 방법을 쓰고 있다. 임계 경로에 대한 실험적 분석을 통하면 최대 지연 시간에 영향을 주는 임계 경로가 주로 주출력쪽에 위치하는 경우가 많으며 retiming 적용시 주출력쪽에서 입력단으로의 레지스터 위치 재조정이 보다 효율적이다. 그림 6 (a)의 경우 레지스터의 위치를 주출력쪽으로 재조정하면 오직 한 개의 레지스터에 대해 retiming 기법을 적용하여 그림 6 (b)와 같이 최소의 클럭 주기를 갖는 설계가 가능하다. 이와 같이 레지스터의 위치를 가능한 주출력단으로 재조정해줌으로써 그림 6 (b)와 같이 최소의 클럭 주기를 갖는 설계가 보다 빨리 실행될 뿐 아니라 그림 6 (c)에서 그림 6 (f)로 표현되는 모든 패턴들을 설계자의 설계 제약에 따라 빠른 시간내에 얻을 수 있다.

4. 동기 시간 최적화

동기 시간 최적화 과정에서는 비용 함수에 의해 결정된 주기를 만족시키도록 레지스터의 위치를 지정해준다. 먼저 자료 구조 구축시 얻은 정렬된 최대 지연 시간을 나타내는 D중에서 주어진 주기보다 큰 지연 시간을 갖는 임계 경로들을 추출한다. 이렇게 선택된 임계 경로는 지연 시간이 큰 순서로 정렬되어 있으므로 순서대로 레지스터의 위치를 지정해준다. 이때 추출된 임계 경로들을 비교하면 다음과 같은 특성을 관찰할 수 있다. 아래에 주어진 경로 1,2,3이 주어진 주기를 초과하는 임계 경로들로 추출되었다고 가정하자. 이러한 경로들이 주어진 주기를 만족하기 위해서는 반드시 경로 3의 gate1 과 gate2 사이에 레지스터가 존재하도록 동기 시간 최적화가 적용되어야 한다. 본 논문에서는 이와 같이 임계 경로 상에 존재하는 여러 경로중 반드시 동기 시간 최적화가 적용되어

야 하는 경로를 frequently used path (FUP)로 정의한다.

- 경로 1 : gate0 → gate1 → gate2 → gate3
- 경로 2 : gate1 → gate2 → gate3
- 경로 3 : gate1 → gate2

FUP는 임계 경로 상의 모든 경로들을 비교하여 경로간의 포함 관계를 비교하여 얻어낼 수 있으나 이에는 많은 시간이 소모되므로, FUP를 직접 추출하는 대신에 이러한 특성을 이용하여 보다 빠른 동기 시간 최적화를 수행한다. 추출된 임계 경로들에 대한 동기 시간 최적화의 적용 이전에 전체 주기를 계산하여 주어진 동기 시간 설계 제약의 만족 여부를 확인한다. 이 때 주어진 동기 시간 설계 제약을 만족하는 경우는 나머지 고려되어야 할 임계 경로들의 FUP에 대한 동기 시간 최적화 고려가 이전의 임계 경로에 대한 동기 시간 최적화 적용으로 만족되었음을 의미하므로 더 이상의 동기 시간 최적화가 필요없다.

V. 실험 결과

SiLOS-II는 UNIX 환경하에서 C로 구현되었으며, 구현된 시스템의 성능 평가를 위해 표준 테스트 회로로 채택된 IWLS86 벤치마크 프로그램¹⁾에 대하여 TECHMAP 시스템에서 사용된 MCNC 라이브러리의 수정 라이브러리를 사용하였다.

시스템 간의 성능 비교를 위하여 mis-II technology mapper 시스템과의 최소 동작 시간 매핑 결과를 표 1에 보였다. 최소 동작 시간 커버의 비교를 위하여 mis-II에서 제공하는 최소 지연 시간 매핑 모드 (-m l -F -A)를 이용하였다. 시스템 비교를 위하여 사용된 mis-II의 지연 시간 모델에서는 부하량을 고려하지 못하고 있으므로 정확한 지연 시간 비교를 위하여 mis-II 매핑 결과 회로에 대하여 부하량을 고려하여 동작 시간을 재계산하였다. 제안된 동작 시간 최적화 기술 매핑을 수행한 결과 지연 시간은 13.9% 감소되었고 면적은 13.8% 증가된 결과를 보였다. 벤치마크 비교 결과 최대 37.2%까지 지연 시간이 감소된 이유는 mis-II에서는 동작 시간을 고려한 재설계시 부하량에 의한 지연 시간 변화를 전혀 고려해 주지 못하기 때문이다. 동작 시간 최적화 매핑시 부하량을 직접 고려하여 제안된 동작 시간 최적화 기법들이 mis-II에 비해 효율적으로 적용됨을 볼 수 있다.

표 2에서는 제안된 시스템의 면적 최적화 모드와 동작 시간 최적화 모드의 매핑 결과와 각 모드에 대한 동기 시간 최적화를 적용한 결과를 보이고 있다.

면적 최적화 모드에 대한 동기 시간 최적화 고려로 전체 주기가 10.8% 감소되었고 레지스터의 수가 25% 증가되었으며, 동작 시간 최적화 모드에서 전체 주기는 0.9% 감소되었으며 레지스터의 수는 1% 증가된 결과를 얻었다. 이처럼 매핑 모드에 따른 동기 시간 최적화 결과가 차이를 보이는 이유는 동기 시간 최적화의 적용이 회로의 초기 구조에 의존하므로 동작 시간 최적화 모드에서 생성된 논리 회로가 최소의 주기를 갖는 경우가 대부분이기 때문이다. 동기 회로에 대한 동기 시간 최적화의 고려로 최소 주기를 가지는 매핑 결과를 얻었다.

표 1. SiLOS-II와 mis-II의 성능비교

Table 1. Performance comparison of mis-II and SiLOS-II.

회로	mis-II		SiLOS-II		Δ %	
	지연 시간(ps)	면적	지연 시간(ps)	면적	지연 시간	면적
r2	3417	25056	2657	24128	-22.2	-3.7
con1	4396	20416	3752	21808	-14.6	+6.8
misex2	7605	128064	5664	122960	-25.5	-4.0
misex1	6201	76560	6065	83520	-2.2	+9.1
z4w1	8333	51504	6096	60784	-26.8	+18.0
rd53	6511	61248	6429	55216	-1.3	-9.8
5xpl	10986	131312	6894	142448	-37.2	+8.5
bw	9280	219008	7531	248240	-18.8	+13.3
vg2	9079	102080	7859	147088	-13.4	+44.1
f51w	10351	156368	8390	167504	-18.9	+7.1
sao2	9668	184672	8760	187920	-11.2	+1.8
rd73	10578	144768	9160	153584	-13.4	+6.1
rd84	14673	312736	10974	326656	-25.2	+4.5
duke2	14065	528960	12118	592064	-13.8	+11.9
9sym1	13878	254736	12895	233856	-7.1	-8.2
9sym	12819	257984	13541	244538	+5.6	-5.2
slup1a	18278	146160	17092	250096	-6.5	+71.1
misex3	21755	700640	19535	925680	-10.2	+32.1
계	192073	3503373	165412	3988090	-13.9	+13.8

표 2. 기술 매핑 모드에 따른 동기 시간 최적화 결과

Table 2. Retiming results according to technology mapping mode.

회로	면적 최적화 매핑		Retiming 적용후		동작시간 최적화 매핑		Retiming 적용후	
	주기 (ps)	#FFs	주기(ps)	#FFs	주기 (ps)	#FFs	주기(ps)	#FFs
S27	6732	3	6267	4	4495	3	3858	4
S208	12641	8	12641	8	8864	8	8864	8
S298	15590	14	14727	16	9032	14	9032	14
S344	16804	15	11979	24	11618	15	11618	15
S249	16804	15	11979	24	10941	15	10941	15
S382	15269	21	14294	26	9293	21	9293	21
S386	10669	6	10669	6	8383	6	8383	6
S400	16184	21	16184	21	9934	21	9934	21
계	110693	103	98740	129	72560	103	71923	104
Δ %			-10.8	+25			-0.9	+1

VI. 결론

본 논문에서는 조합 회로의 동작 시간 최적화와 동기 회로의 동기 시간 최적화를 위한 타이밍 최적화

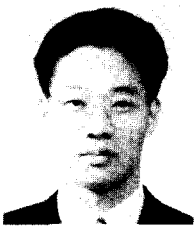
기술 매핑 시스템의 설계에 대하여 설명하였다. 조합 논리 회로가 최소의 동작 시간을 갖기 위한 재설계 방식으로 부하량을 고려한 재매핑, 버퍼링 및 게이트 병합에 의한 최적화 알고리즘을 제안하였다. 각 재설계 방식에 부하량을 직접 고려함으로써 벤치마크 실험 결과 mis-II 시스템에 비해 작은 면적의 증가로 보다 빠른 동작 시간을 갖는 회로 설계가 가능하다. 동기 회로의 동기 시간 최적화를 위하여 레지스터의 위치 재조정 및 임계 경로상의 FUP 특성을 고려하여 알고리즘 복잡도를 줄이는 휴리스틱 접근 방법을 제안하였다. 벤치마크 실험 결과를 통하여 조합 회로에 대하여 부하량을 고려하여 최소의 동작 시간을 갖는 회로 매핑이 가능하였고, 동기 회로에 대하여 최소의 주기를 갖도록 해줌으로써 기존의 기술 매핑 시스템들에 비해 타이밍 측면에서 보다 최적화된 논리 회로 매핑이 가능해졌다. 현재 동작 시간 최적화 및 동기 시간 최적화 결과가 초기 구조의 차이에 많이 의존하는 단점을 보완하기 위하여 초기 매핑시 수행하는 decomposition에서 정확히 부하량을 고려하는 연구가 진행되고 있으며, 앞으로 VLSI 시스템 설계 자동화 틀 환경화에서 개발중인 silicon compiler에 통합을 위한 작업이 계속될 것이다.

参 考 文 献

- [1] K. Bartlett, W. Cohen, A. de Geus, and G. D. Hachtel, "Synthesis and Optimization of Multilevel Logic under Timing Constraints", *IEEE Trans. CAD of Int. Circ. Syst.*, Vol. CAD-5, No. 4, Oct. 1986, pp. 582-596.
- [2] D. Bostick, G. D. Hachtel, R. Jacoby, M. R. Lightner, P. Moceyunas, C. R. Morrison, D. Ravenscroft, "The Boulder Optimal Logic Design System", in Proc. ICCAD, Nov. 1987, pp. 62-65.
- [3] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, "Logic Minimization Algorithms for VLSI Synthesis", Kluwer Academic Pub., 1984.
- [4] R. K. Brayton, R. Rudell, A. L. Sangiovanni-Vincentelli, and A. Wang, "MIS : A Multiple-Level Logic Optimization System", *IEEE Trans. CAD of Int. Circ. Syst.*, Vol. CAD-6, No. 6, Nov. 1987, pp. 1062-1081.
- [5] J. A. Darringer, D. Brand, J. Gerbi, W. H. Joyner, Jr., and L. Trevillyan, "LSS: A System for Production Logic Synthesis", *IBM J. Research and Development*, Vol. 28, No. 5, Sept. 1984, pp. 537-545.
- [6] E. Detjens, G. Gannot, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, "Technology Mapping in MIS", in Proc. ICCAD, Nov. 1987, pp. 116-119.
- [7] A. de Geus, "Logic Synthesis and Optimization Benchmarks for the 1986 Design Automation Conference", in Proc. 23rd Design Automation Conf., June 1986, p. 78.
- [8] G. De Micheli, "Synchronous Logic Synthesis: Algorithms for Cycle-Time Minimization", *IEEE Trans. CAD of Int. Circ. Syst.*, Vol. CAD-10, Jan. 1991, pp. 63-73.
- [9] D. Gregory, K. Bartlett, A. de Geus, and G. Hachtel, "SOCRATES: A System for Automatically Synthesizing and Optimizing Combinational Logic", in Proc. 23rd Design Automation Conf., June 1986, pp. 79-85.
- [10] K. Keutzer, "DAGON: Technology Binding and Local Optimization by DAG Matching", in Proc. 24th Design Automation Conf., June 1987, pp. 341-347.
- [11] K. Keutzer and M. Vancura, "Timing Optimization in a Logic Synthesis System", in 'Logic and Architecture Synthesis for Silicon Compilers', G. Saucier and P. M. McLellan (ed.), North-Holland, 1989.
- [12] C. Leiserson, F. Rose, and J. Saxe, "Optimizing Synchronous Circuitry by Retiming", in Proc. 3rd Caltech Conference on VLSI, Computer Science Press, 1983, pp. 87-116.
- [13] A. Langmaier, "Technology Mapping", in 'The Synthesis Approach to Digital System Design', P. Michel, U.

- Lauther, and P. Duzy (ed.), Kluwer Academic Pub., 1992.
- [14] R. Lipsett, "VHDL: Hardware Description and Design", Kluwer Academic Pub., 1989.
- [15] S. Malik, E. M. Sentovich, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Retiming and Resynthesis: Optimizing Sequential Networks with Combinational Techniques", *IEEE Trans. CAD of Int. Circ. Syst.*, Vol. CAD-10, Jan. 1991, pp. 74-84.
- [16] C. R. Morrison, R. M. Jacoby, and G. D. Hachtel, "TECHMAP: Technology Mapping with Delay and Area Optimization", in 'Logic and Architecture Synthesis for Silicon Compilers', G. Saucier and P. M. McLellan (ed.), North-Holland, 1989.
- [17] W. Rosentiel, D. Schmid, "Logic Synthesis", in 'Logic Design and Simulation', E. Horbst (ed.), North-Holland, 1986.
- [18] K. J. Singh, A. R. Wang, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Timing Optimization of Combinational Logic", in Proc. ICCAD, Nov. 1988, pp. 282-285.
- [19] H. J. Touati, C. W. Moon, and R. K. Brayton, "Performance-Oriented Technology Mapping", in Proc. of Sixth MIT Conference, MIT Press, 1990, pp. 79-97.
- [20] 김 태선, 황 선영, "논리 회로의 기술 매핑 시스템 설계", 대한전자공학회 논문지, 29-A권 2호, 1992년 2월, pp. 88-99.
- [21] 이 재형, 황 선영, "성능 구동 논리 회로 자동 논리 설계 시스템", 대한전자공학회 논문지, 28-A권 1호, 1991년 1월, pp. 74-84.
- [22] 임 춘석, 황 선영, "Fanin 제약하의 다단 논리 최적화 시스템의 설계", 대한전자공학회 논문지, 29-A권 4호, 1992년 4월, pp. 64-73.

 著 者 紹 介



李相遇(正會員)

1968年 3月 11日生. 1990年 2月 서강대학교 전자공학과 졸업. 1993年 8월 서강대학교 전자공학과 공학 석사 취득. 1993年 11월 ~ 현재 금성 일렉트론 근무. 주관심 분야는 CAD 시스템, Computer

Architecture 및 VLSI 설계 등임.

黃善泳(正會員)

1976年 2월 서울대학교 전자공학과 졸업. 1978年 2월 한국 과학원 전기 및 전자 공학과 공학 석사 취득. 1986年 10월 미국 Stanford대학 공학 박사 학위 취득. 1976年 ~ 1981年 삼성 반도체 주식회사 연구원. 1986年 ~ 1989年 Stanford대학 Center for Integrated Systems 연구소 연구원, Fairchild Semiconductor Palo Alto Research Center 기술 자문. 1989年 3월 ~ 현재 서강 대학교 전자 공학과 교수. 주관심 분야는 CAD 시스템, Computer Architecture 및 Systems Design, VLSI 설계 등임.