

論文2001-38SD-10-6

반도체 자동 이식 알고리즘에 관한 연구

(Algorithms of the VLSI Layout Migration Software)

李潤埴*, 金塔培**, 申晚澈*, 金俊泳*

(Yun Sik Lee, Yong Bae Kim, Man Chul Sin, and Jun Young Kim)

요 약

인터넷의 확산, 이동 통신기기의 급속한 보급으로 말미암아 가전업체는 소형의 다기능의 시스템을 필요로 하고 있고, 이를 위하여 반도체 업계에 고기능, 다기능, 초소형의 시스템용의 칩을 요구하고 있다. 지수 함수적 증가하는 기능의 요구는 반도체 설계 능력을 넘어선지 이미 오래 전이고 이를 극복하기 위하여서 반도체 업계는 여러 가지 방안을 제시하고 있다. 그러나, 이미 그 차이를 따라 잡기는 포기한 상태이고 이 갭을 줄이고자 하는 방안을 모색 중이다. 그 방안은 SoC(System On a Chip), 설계 재활용(Design Reuse) 등의 개념을 활용하고 있다. 설계 재활용을 위하여서는, 반도체 지적 소유권(Intellectual Property)의 표준화와 더불어 레이아웃 자동이식에 관한 연구와 상품화가 필수적이다. 본 논문은 반도체 설계 형식 중에서 생산 공정과 밀접한 레이아웃 형식의 회로 도면 처리를 자동화하여 설계와 생산 시간을 혁신적으로 단축하기 위한 연구이다. 레이아웃 형식은 특성상 도형(폴리곤)으로 구성되어 있으며, 레이아웃 형태에서 다양한 도형의 중첩이 반도체의 트랜지스터, 저항, 캐패시터를 표현함으로써, 반도체 지적소유권의 하나의 형식으로 자주 활용되고 있다. 본 논문은 반도체 레이아웃 이식 소프트웨어 시스템의 내부 기능에 관한 설명과 처리 능력과 속도를 높이기 위한 알고리즘의 제안과 벤치마킹 결과를 보여 주고 있다. 비교 결과, 자원의 최적 활용(41%)으로 대용량의 처리 가능성을 보여 주고 있으며, 처리 속도는 평균 27배로써 이전의 벤치마킹 회로를 더욱 확장하여 그 결과를 보여 주고 있다. 이러한 비교 우위는 본 논문에 포함된 소자 처리 알고리즘과 그래프를 이용한 컴팩션 알고리즘에 기인한다.

Abstract

Algorithms from the research of the layout migration were proposed in the paper. These are automatic recognition algorithm for the VLSI devices from it, graph based construction algorithm to maintain the constraints, dependencies, and design rule between the devices, and high speed compaction algorithm to reduce size of the VLSI area and reuse the design with compacted size for the new technology. Also, this paper describes that why proposed algorithms are essential for the era of the SoC (System on a Chip), design reuse, and IP DB, which are the big concerns in these days. In addition to introduce our algorithms, the benchmark showed that our performance is superior by 27 times faster than that of the commercial one, and has better efficiency by 3 times in disk usage.

* 正會員, 湖西大學校 벤처專門大學院, (주)에이케드
(School of Venture Hoseo University)

** 正會員, (주)에이케드
(ACAD Corp)

※ 본 연구는 '98,99년 과학기술부/산업자원부에서 시행한 시스템집적반도체기반기술사업기술개발에 의해 수행되었음.

接受日字:2001年5月14日, 수정완료일:2001年7月18日

I. 서 론

레이아웃 이식 시스템 구성은 크게 소자추출 부분, 소자 및 배선의 크기 조정 부분, 컴팩션 부분으로 구성된다. 소자추출 부분은 이식하고자 하는 반도체 도면을 입력으로 그 데이터를 분석하여 소자를 추출하며, 크기 조정 부분은 목적하는 설계 규칙과 테크놀러지를 만족하고 타이밍과 소비전력을 최적화할 수 있도록 소자 및 배선의 크기를 조정한다. 컴팩션 부분은 최적화된 소자와 배선으로 구성된 레이아웃 데이터를 최소의 면적을 얻을 수 있도록 압축기능을 수행한다^[1]. 이렇듯, 반도체 레이아웃 이식은 전반부와 후반부로 구분할 수 있다. 전반부는 소자 추출, 소자와 배선 크기 조정 부분이 포함되며, 후반부는 컴팩션 부분이며 레이아웃 이식의 전체 시스템의 성능을 좌우할 정도로 핵심이 되는 부분이다. 특히, 컴팩션에 대한 연구는 70년대 중반부터 90년대 초반까지 약 20여 년간 활발히 수행되었으며, 일부 상용 툴들이 개발되었으나, 처리하는 용량과 성능의 한계로 인하여 만족스러운 결과를 얻지 못하고 있다. 초기 컴팩션의 주요 관점은 특정 테크놀러지에 종속되지 않고 레이아웃 설계를 효율적으로 하는 것이었으며, 심볼릭 레이아웃이 대상이었다.

그러나 최근 시스템 온 칩(System-on-Chip)의 설계가 보편화되어, 반도체 지적재산권(Intellectual Property)의 중요성이 대두되면서 반도체 레이아웃이 컴팩션의 대상이 되고 있다.

이러한 도형(폴리곤) 형태의 레이아웃을 직접 컴팩션에 적용하려면 레이아웃을 심볼들로 구성된 레이아웃 형태로 변환하여 기존의 심볼릭 컴팩션 방법을 적용하거나, 또는 도형의 각각 엣지들을 직접 처리하는 엣지 처리 방식의 컴팩션 방법을 사용한다. 심볼릭 처리 방법은 처리하기가 비교적 간단하다는 장점이 있는 반면, 마스크 레이아웃으로 부터 데이터의 손실없이 모든 정보(트랜지스터, 컨택, 메탈, ...)를 심볼 형태로 변환할 수 있어야 된다. 엣지 방식은 도형을 구성하는 각각의 엣지들을 객체로 처리해야 하므로 데이터가 기하급수적으로 증가하여 대부분 처리 용량의 한계에 직면하게 된다.

컴팩션 방법은 대상이 되는 객체의 형태에 따라 심볼릭과 엣지 방식으로 분류되는 것 이외에도 처리하는 방식에 따라 다양하게 분류된다.

첫째로, 컴팩션하는 객체의 움직임은 방향에 따라 1차원 (1D)방법과 2차원 (2D)방법으로 분류할 수 있다. 1 차원 컴팩션 방법에서는 객체가 X 또는 Y방향으로만 움직일 수 있으며, X, Y방향 컴팩션을 반복적으로 수행한다. 2차원 컴팩션 방법에서는 객체가 동시에 X, Y방향으로 움직인다. 또한 1차원 컴팩션을 수행하면서 부분적으로 다른 방향으로 움직이게 하는 1.5차원(1.5D) 컴팩션 방법이 있다.

둘째로는 컴팩션하는 객체사이의 최소 거리를 계산하는 방법에 따라 제약조건 그래프방식과 가상그리드 방식으로 분류된다. 제약조건 그래프 방식에서는 객체들 사이에 적용되는 설계 규칙과 같은 제약 조건들을 선형 부등식으로 기술하며, 이것을 방향성과 가중치가 있는 그래프로 모델링하고 해석함으로써 컴팩션을 수행한다. 가상 그리드 방식은 모든 객체들이 가상적인 그리드 선상에 있다고 가정하고, 이 그리드 선상에 있는 객체들을 고려하여 인접한 그리드 선 사이의 최소 거리를 계산하여 컴팩션을 수행하는 방법이며, 제약조건 그래프방법보다 알고리즘이 비교적 간단하여 수행속도가 빠른 반면 컴팩트한 결과를 얻지 못한다.

본 연구에서는 이식의 가장 기술적인 핵심인 입력도면의 100% 소자 인식, 칩의 면적 축소를 위한 컴팩션, 그리고 대형 도면을 처리하기 위한 고속의 알고리즘과 동반되는 내부 구조를 설명하고자 한다.

II. 레이아웃 이식 알고리즘

레이아웃 이식 시스템 구성은 그림 1에서 보는 바와 같이 크게 소자추출 부분, 소자 및 배선의 크기 조정 부분, 컴팩션 부분으로 구성된다. 소자 추출 부분은 이식하고자 하는 회로를 입력으로 받아 데이터를 분석하여 소자를 추출하며, 크기 조정 부분은 목적하는 설계 규칙과 테크놀러지를 만족하고 타이밍과 소비전력을 최적화할 수 있도록 소자 및 배선의 크기를 조정한다. 컴팩션 부분은 최적화된 소자와 배선으로 구성된 레이아웃 데이터를 최소의 면적을 얻을 수 있도록 압축, 축소 기능을 수행한다. 구성 부분 중 처리 용량과 속도에 가장 밀접한 알고리즘을 위주로 설명하도록 하겠다.

1. 전 처리 과정의 알고리즘

소자 추출을 준비하는 과정으로써, 회로 도면과 더불어 이식 룰을 정의하는 이식 룰 형식을 준비한다. 이를 설계 룰을 기준으로 하여 작성하며, 레이아웃의 각 레

이어에 관한 정의, 소자 정의, 소자 크기의 정의, 연결 정보의 정의, 설계 규칙의 정의, 컨택, 컴팩션의 정의들로 구성되어 있다. 각각의 정의는 일반적으로 현재 도면의 기술 특성에서 적용하고자 하는 기술 특성의 맵핑을 기본으로 구성되어 있다.

2. 소자 추출 과정의 알고리즘

소자추출은 후반부 설계, 즉, 마스크 레이아웃설계 검증에서 가장 핵심이 되는 영역중의 하나다. 특히, 검증 부분은 반도체 설계 전반에 걸쳐 각 단계마다 이루어지고 있으며, 반도체 설계 방법론이 아무리 바뀌더라도 반드시 필요한 핵심 부분이다.

반도체 레이아웃은 여러 개의 레이어로 이루어지며 이들 각각은 도형(폴리곤)으로 구성된다. 이들과 사용자가 지정한 레이어들 간의 관계를 이용하여 소자를 추출하게 되는데, 일반적인 소자의 종류로는 모스나 바이폴라 트랜지스터, 캐패시터, 저항, 다이오드 등이 있다. 특히 트랜지스터는 여러 개의 레이어로 구성되어 있으며, 이들 레이어들 간의 패턴 연산(AND, OR, NOT, XOR..)을 수행하여 추출하게 된다.

본 연구에서는 이식 기술의 핵심인 컴팩션에서 이와 같이 추출된 소자를 활용하기 위하여 소자의 크기정보를 추출하여 적용할 테크놀러지에 적합한 소자크기를 계산하고, 그 결과를 컴팩션 부분에서 각 도형들 간의 제약조건으로 사용될 수 있도록 한다.

2.1 패턴 연산과 소자 추출

소자추출에서 사용되는 도형 연산은 반도체 레이아웃 데이터를 분석 및 검증하는데 가장 기본적이며 핵심기능이라고 할 수 있다.

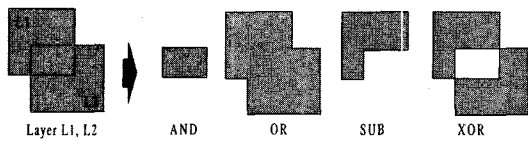


그림 1. 기본적인 도형의 연산
Fig. 1. Polygon Operation.

소자 추출의 기능시에, MOS 트랜지스터의 경우, Substrate를 제외하고 전기적 특성을 갖는 포트는 3개 (게이트/소스/드레인)이며, 레이아웃 상에서 각 포트에 맵핑 될 수 있는 레이아웃은 그림 2와 같이 표현 할 수 있다. 여기서, MOS 소자는 레이어 D가 될 수 있

며, 게이트는 레이어 B, 소스와 드레인은 각각 레이어 C로 맵핑 될 수 있다. 소자추출 프로그램 상에서는 이렇게 만들어진 레이어 자료에 대해 맵핑이 가능한 정보를 입력하면 조건에 맞는 데이터들을 추출하여, 정의된 형태의 소자로 출력하게 되는 것이다.

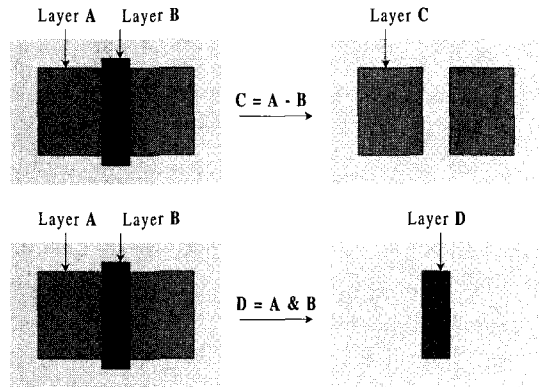


그림 2. MOS 트랜지스터 소자 추출의 예
Fig. 2. Example of the Extraction of MOS Tr.

소자 추출의 전체 흐름도는 그림 3과 같으며, 사용자가 정의한 소자 규칙과 소자에 해당하는 도형의 좌표 데이터를 가지고, 부울 연산을 수행하여 가공을 한 뒤 각 소자의 포트 맵핑 규칙에 대해 추출하도록 구성된다.

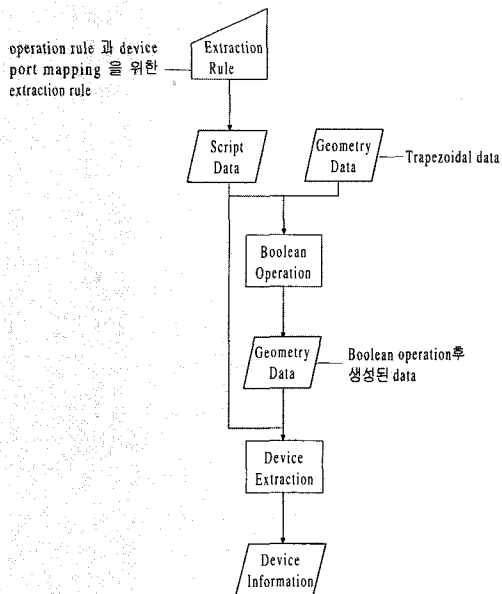


그림 3. 소자 추출 흐름도
Fig. 3. Diagram of the Extraction Process.

따라서, 각 소자 형태에 대해 그림 3.의 과정을 거치게 되며, 여러 종류의 소자들이 정의되는 경우에는 그 종류만큼 위의 과정을 거치면서 추출하게 된다.

2.2 소자의 크기 조정 알고리즘

소자에 대한 크기조정은 전 단계에서 추출된 소자와 연결정보를 이용하여 적용 테크놀러지와 설계규칙에 맞도록 소자의 크기를 계산하는 기능이다^[1].

2.2.1. 트랜지스터의 크기 조정 방법

그림 4의 위 부분은 기본형태의 트랜지스터의 크기 조정을 나타내며, 그림 4의 아래는 트랜지스터 소자의 일반적인 모양과 컴팩션 단계로 넘기기 위한 제한 조건 정보를 나타낸다. 위의 그림처럼 제한 조건 정보는 배선 조건과 각 평행한 엣지들에 대한 각각의 엣지 제한 조건으로 구성되며, 45to45나 45to90에 대한 정보는 포함되어 있지 않다.

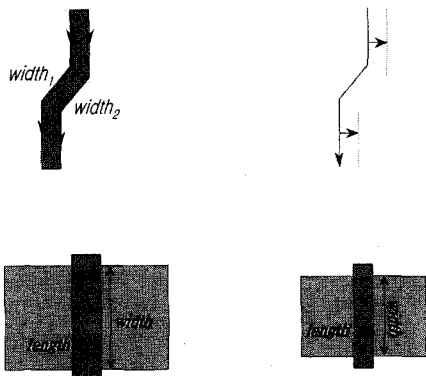
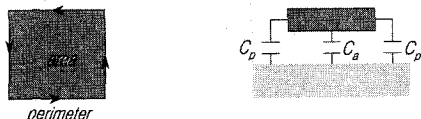


그림 4. 트랜지스터의 크기 조정
Fig. 4. Transistors Resizing.

2.2.2. 캐패시터의 크기 조정

그림 5는 캐패시터 소자의 일반적인 모양과 소자값의 계산 방법을 나타내며, 여기서, fperimeter와 farea는



캐패시턴스, $C = perimeter * f_{perimeter} + area * f_{area}$.

그림 5. 캐패시터의 크기 조정
Fig. 5. Capacitor Resizing.

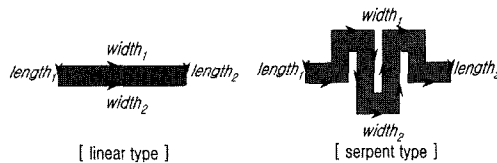
공정 변수이며, 전체 캐패시턴스에서 면적 항이 차지하는 비율이 거의 대부분이므로 주변 길이 항은 무시 가능하다.

2.2.3 저항의 크기 조정

그림 6은 저항 소자의 일반적인 모양과 소자 값의 계산방법을 나타내며, 여기서, R₀은 공정 변수이며, n₀은 다음과 같은 근사식으로 계산될 수 있다.

$$n_0 = (width_1 + width_2)/(length_1 + length_2)$$

물론 우측의 형식(serpent type)과 같은 형태에서 정확도를 높이려면 각 세그먼트별로 계산을 해야 하며, 각 굴곡점에서의 전기장을 고려해야 하지만, 일반적인 설계의 경우 세그먼트에 따라 형태가 차이 나는 경우는 거의 없기 때문에 속도문제를 고려하여 근사값으로 계산한다.



레지스턴스, $R = R_0 * n_0$

그림 6. 저항의 크기조정
Fig. 6. Resistor Resizing.

2.3 연결 정보의 추출

각종 타이밍과 파워소모를 분석하기 위해서는 레이

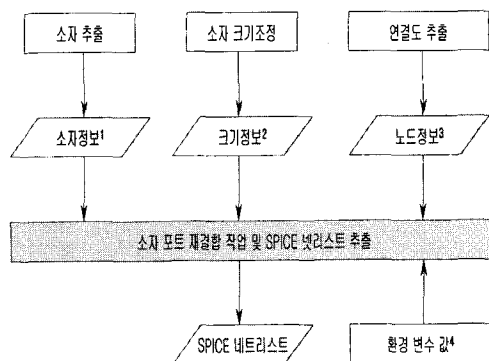


그림 7. SPICE 넷 리스트를 출력하기 위한 흐름도
Fig. 7. Flow Diagram of the Extraction of Netlist.

아웃 데이터에서 회로의 연결도(네트리스트) 추출은 불가피한 작업이다. 레이아웃 데이터에서 회로의 연결도를 추출하기 위해서는 미리 각 소자들에 대한 정보와 연결 정보들을 추출해야 한다. 트랜지스터와 레지스터, 캐패시터 소자의 추출이 가능하도록 하였고, 이를 바탕으로 SPICE용 네트리스트를 출력할 수 있도록 작업하였다. 그림 7의 흐름도는 SPICE 네트리스트를 추출하기 위한 작업들이다.

그림 7의 전체 구성도상에서 각 단계는 아래와 같다.

- 단계 1 : 지정된 소자별로 생성되는 출력파일
(소자별 포트에 대한 정보를 담고 있다.)
- 단계 2 : 지정된 소자별로 생성되는 출력파일
(소자의 크기에 대한 정보를 담고 있다.)
- 단계 3 : 지정된 레이어별로 생성되는 출력파일
(연결도에 대한 맵핑 테이블정보를 보임)
- 단계 4 : 환경 변수들에 대한 입력
(형식은 아래와 같다.)

```

DEVICE(dev1_type, prefix) = dev1_file <port->
    [siz1_file]
DEVICE(dev1_type, prefix) = dev1_file <port->
    [siz1_file]
.
.
TOPCELL = cell1 cell2 ..
NETLAYER = layer1 layer2 ..

```

위 형식에서 크기 관련 파일에 대한 지정이 없으면 네트리스트 출력 시 크기 관련 파라미터들은 추출되지 않으며, 네트레이어 항목은 네트를 구성하는 레이어들에 대한 정의를 나타낸다. 스파이스 네트리스트를 추출하는 목적 외에도 특정 네트에 대한 레이아웃 데이터의 출력도 필요한 경우가 있다. 이는 컴팩션 규칙상의 특정 네트에 대한 레이아웃 규칙을 달리 적용할 때에 필요하며, 이를 지원하기 위하여 두 가지의 진행 흐름도를 가지고 있다.

III. 고안된 컴팩션 알고리즘

반도체 도면의 이식기능의 핵심은 컴팩션 알고리즘의 구현에 있다. 컴팩션 알고리즘에 따라 이식용 자동화 기능이 속도, 성능에 직접적인 영향을 미친다. 본 연

구에서 사용한 컴팩션 알고리즘은 평탄한 설계 도면(flattened GDS circuit)을 이용하여, 제약 조건 그래프를 이용한 1차원 엣지 컴팩션 방식을 사용하여 구현하였다. 연구에서 고안한 컴팩터가 지원하는 설계규칙에 대해서 설명하고, 각 설계 규칙을 방향성과 가중치가 있는 그래프인 제약 조건 그래프로 변환하는 방법에 대하여 기술하며, 그래프 해석 방법인 가장 긴 경로 탐색, 배선길이 최소화, 조그 삽입에 대하여 기술한다.

3.1 설계 규칙

컴팩션 과정에서 만족되어야 하는 제약 조건으로는 객체들간의 최소거리를 지정하는 거리에 대한 제약 조건과 연결 상태를 유지할 수 있는 연결도에 대한 제약 조건으로 대표할 수 있고, 이 제약 조건을 기술하기 위한 설계규칙으로 Width, Distance, Overhang, Overlap Rule과 같은 4가지를 지원하며, 이에 대한 기술 형식은 다음과 같다.

```

DesignRuleType LayerDefinition[Layer-Definition]>=
    MinimumValue

```

DesignRuleType은 위의 4가지 Rule Type을 지정하는 것이며, LayerDefinition으로는 단순한 layer 이름을 지정하거나 아래와 같이 부울(AND, OR, NOT) 연산과 레이어 지정으로 표현할 수 있다.

```
(LayerName Op LayerName). (LayerSelection)
```

예를 들어, (M1&CO).CO 는 레이어 Metal1(M1) 안에 있는 컨택(CO)를 지정하는 것이다.

위 4가지 설계규칙은 부울 연산을 통하여 하나의 Width Rule로 표현될 수 있으며, Width Rule에 대한 제약조건 그래프 생성 기능만으로 다른 설계 규칙들을 처리할 수 있다. 또한, 다른 레이어간의 설계규칙을 부울 연산을 통하여 한 레이어로 통합함으로써 처리하는 엣지 간에는 서로 교차하지 않는 특성이 있으며 이는 제약조건 그래프 생성을 간단하게 한다.

3.2 제약조건 그래프 생성

컴팩션에서 일반적으로 사용되는 방법인 제약조건 그래프 방식은 그래프 생성이 가장 많은 시간을 소모하는 부분으로 많은 연구가 수행되었다. 그래프생성 방법 중 간단한 방법은 설계규칙에 의한 제약 조건이 있는지를 모든 객체 쌍에 대하여 비교하는 것으로 비교 횟수는 $O(N^2)$ 가 된다. 이 비교 횟수를 줄이기 위한

방법으로 SP(Shadow Propagation)과 PS(Plane Sweep)이 있다. SP방법은 객체의 그림자를 컴팩션 방향으로 전달시키고 객체들 간의 관계를 찾아내어 제약 조건을 생성하는 방법이다. 그림 8의 예에서와 같이, 즉

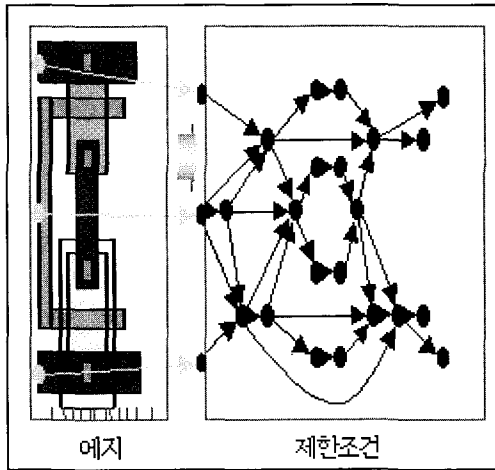


그림 8. 제약 조건 그래프의 생성 예
Fig. 8. Example of constructing Constraint base Graph.

하나의 객체에 가상의 빛을 비추었을 때 그로 인하여 생기는 그림자를 차단하는 객체들과 제약조건을 생성하면 되며 비교 횟수는 $O(N^{1.5})$ 이다. 직각 PS방법은 컴팩션하고자 하는 직각 방향으로 PS 방법을 적용하여 객체가 스캔 라인에 들어올 때와 나갈 때 제약조건을 검색하면 되며 비교 횟수는 $O(N \log N)$ 이다. SP와 PS방법을 접목시킨 PPSS(Parallel Plane Sweep Shadowing)방법은 컴팩션하고자 하는 방향으로 PS를 적용하는 방법으로 각 객체를 한번만 처리하면 되며, 가장 효율적인 방법이다. 본 연구에서는 PPSS 방법을 엣지 컴팩션에 적용할 수 있도록 구현하였으며, 현재는 45도 엣지까지만 지원한다.

그림 9의 알고리즘은 Y방향 컴팩션을 기본으로 작성되었으며, (h,j,k)는 45도 엣지를 처리하기 위하여 추가된 부분이다.

- (a) Yqueue = Sort_edges_in Y_direction ()
- (b) Sfront = 0 Dfront = 0
- (c) while (Yqueue is not empty)
- (d) Nedges = get_scan_in_edge(Yo, NextYo, Yqueue)

- (e) foreach (E in Nedges)
- (f) update Sfront list using E
- (g) and generate constraints among the interacting ones
- (h) if(E is not horizontal edge) push E into Dfront list
- (i)
- (j) if (Dfront list is not empty)
- (k) do_oblique_shadowing
- (l) (Yo, NextYo, D front, Sfront)

그림 9. PPSS 컴팩션 알고리즘
Fig. 9. PPSS Compaction Algorithm.

위 알고리즘에서 Yqueue는 엣지의 정렬된 리스트를 저장하는 변수이다.

Sfront는 Shadow Front List를 저장하기 위한 변수이며, Dfront는 Shadow Front List에 있는 45도 엣지 중 현재 스캔 라인에 걸쳐있는 엣지 리스트를 가지고 있는 변수이다. Yo와 NextYo는 현재와 다음 스캔 라인의 Y좌표를 가지고 있는 변수이다. (a)는 엣지의 양 끝점 중 작은 Y좌표 (Ymin)를 기준으로 정렬을 하여 리스트 Yqueue에 저장한다.

(d)는 Yqueue에 있는 엣지를 참조하여 현재와 다음 스캔라인의 Y값을 결정하고, Yo에 걸쳐있는 엣지들을 Nedges에 저장한다. (f,g)는 설계규칙을 고려하여 제약조건 그래프에 제약조건을 등록한다. (h)는 현재의 Scan-line에 들어오는 45도 Edge를 Dfront 리스트에 등록하는 부분이다. (k)는 현재 스캔라인에 걸쳐있는 45도 엣지, 즉 Dfront 리스트에 등록되어 있는 엣지에 대하여 Shadowing을 수행하는 과정이며, 때 다음 스캔라인 (NextYo)에서 나가게 되면 Dfront 리스트에서 삭제한다.

알고리즘을 그림 10을 적용하면, 스캔라인이 y0에서 모든 과정을 마치고 나면 각 변수들은 다음과 같은 값을 가지게 된다.

$$Yo = y0, NextYo = y1, Sfront = (E2, E3), Yqueue = (E4, E5, E6)$$

알고리즘의 (d)연산에 의하여 변수값들은 다음과 같이 변하게 된다.

$$Yo = y1, NextYo = y2, Sfront = (E2, E3),$$

Nedgs = (E4, E5), Yqueue = (E6)

다음 알고리즘 (e-i) 연산에 의하여 Sfront=(E4, E5, E3)로 변하며, 제약조건 그래프에 설계규칙 Width 와 Distance에 해당하는 Arc A2, A3이 추가된다. 이와 같은 과정을 스캔 라인을 증가시키면서 반복하면 제약조건 그래프가 생성된다.

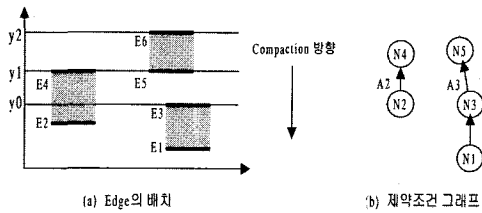


그림 10. 엣지의 배치와 제약조건 그래프
Fig. 10. Example of the Edge and Constraint based Graph.

3.3 가장 긴 경로 탐색

가장 긴 경로 탐색은 제약조건을 위반하지 않는 각 객체들의 최소 위치를 결정하는 과정으로 가장 긴 경로를 임계경로(Critical Path)라고 한다. 이때 사용자 제약조건(User Constraint) 등으로 인하여 가장 긴 경로를 찾을 수 없는 경우가 발생하며, 이는 그래프에 순방향 사이클이 존재하기 때문이고 이를 찾아내어 해결(조그 삽입 또는 리랙싱)하는 것이 중요한 관건이 된다. 이에 대한 방법으로는 그래프에 순방향 사이클이 없을 때 적용 가능한 Bellmann & Ford 방법^[2], 순방향 사이클이 있을 경우 이 벤트 기법을 사용하여 수렴할때까지 반복적으로 수행하는 Liao & Wong방법^[3], 점진적인 방법을 이용하여 순방향 사이클이 발견되는 즉시 해석 과정을 수행하는 확장된 DBR방법^[4] 등이 있다. 본 연구에서는 초기 가장 긴 경로 탐색 시에는 확장된 DBR 방법을 이용하였으며, 조그 삽입시에는 순방향 사이클이 존재하지 않으므로 Bellmann & Ford 방법^[2]을 이용하여 구현하였다.

가장 긴 경로 탐색 시 고려해야 할 사항으로 특정 객체를 특정한 그리드 상에 놓이도록 하는 그리드 제약 조건이 있다. 이는 그리드 라우터가 연결을 할 수 있도록 셀의 핀 위치를 배선상의 그리드 상에 놓이게 하는 것이다.

3.4 배선 길이 최소화

배선길이 최소화는 경로 탐색 과정에 의하여 컴팩션

방향으로 과도하게 이동한 객체들을 제약조건을 위반하지 않는 범위 내에서 배선길이가 최소화 되도록 컴팩션 반대 방향으로 이동시키는 과정이며, 이 과정을 통하여 기생효과를 줄여 고성능 회로를 생성할 수 있고 다른 방향으로의 컴팩션 효과를 높일 수 있다.

이 배선길이 최소화 문제는 리니어 프로그래밍으로 공식화 할 수 있으며, 이를 그래프 상에서 보면 가장 긴 경로 탐색에 의해서 구해진 스패닝트리를 초기 해로 설정하고 반복적인 트리 변환 과정을 통하여 최적의 해를 구하는 것이다. 그러므로 배선길이 최소화는 스패닝트리를 반복적으로 변환하는 과정으로 볼 수 있다. 스패닝의 각 아크에서의 코스트는 트리에 있는 각각의 코스트의 합으로 표현된다.

3.5 조그 삽입

조그 삽입은 그래프 해석 시 순방향 사이클의 해석 과정과 면적 최소화 과정에 이용될 수 있다. 조그를 이용한 면적 최소화는 임계경로 상에 있는 배선에 조그를 삽입하고 새로운 임계경로 해석을 수행하며 이 과정을 조그를 삽입할 공간이 없을 때까지 반복하는 것이다. 조그를 삽입하는 방법으로는 배선 상에 토크 위치를 지정하여 양쪽방향에서 힘을 가하는 Force-directed 조그 기법과 채널 배선에서 주로 사용되는 Contour 컴팩션 조그 기법이 있다

그림 11은 본 연구에서 구현한 조그 삽입에 따른 그래프의 변화를 보여 주고 있다.

조그 삽입이 가능한 노드 N3, N5를 찾아 분할하는 방법으로 다음 임계경로 해석에 의하여 Node N2, N3와 N4, N5가 자유롭게 위치하게 되며, 이로 인하여 조그 삽입이 가능하게 된다.

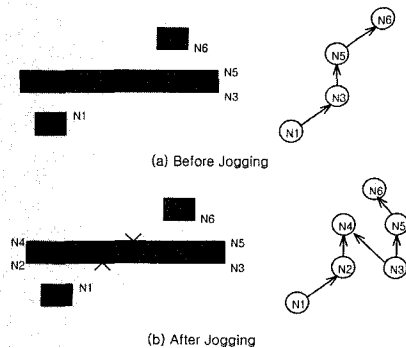


그림 11. 조그 삽입에 따른 그래프의 변화
Fig. 11. Graph with Jog Insertion.

3.6 컨택의 합병

설계 레이아웃 상에서 가장 많은 객체를 포함하는 것은 컨택과 관련한 부분이다. 컨택은 전체 셀의 50% 가까이 차지하기 때문에 컴팩션의 기능을 수행 할 때 이러한 컨택들을 가능한 줄이는 것이 성능의 향상에 크게 기여 할 수가 있게 된다. 따라서, 컴팩션 기능 이전에 컨택의 합병이 필요하며, 컴팩션이 끝난 후에는 다시 새로운 설계규칙에 따라 컨택 재 설계를 해주는 작업이 함께 들어가게 된다.

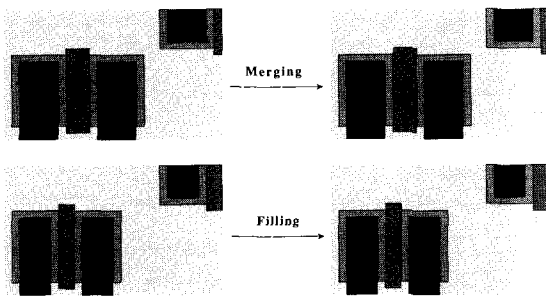


그림 12. 컨택의 이식
Fig. 12. Migration of the Contact.

IV. 실험 결과

본 연구에서 구현한 이식 기능의 효율성을 검증하기 위하여 상용 툴과 비교하여 실험하였다. 사용 시스템은 선 스팩 20, 솔라리스 2.5.1이며, 트랜지스터 300~50,000여 개의 1.0 μ m 테크놀러지로 설계된 회로에 0.6 μ m 테크놀러지의 설계규칙을 적용하여 Y방향, X방향의 컴팩션을 순차적으로 수행하였으며, 또한 조그 삽입과 배선길이 최소화 기능을 적용하여 수행하였다^[1]. 이식된 면적, 수행 시간, 사용된 메모리를 비교한 것이며,

면적 차이는 1%미만으로 거의 동일한 결과를 얻을 수 있는 반면^[그림 13], 수행 속도에는 평균 27.4배의 차이가 나며, 메모리는 상용 소프트웨어에 비해 평균 40%의 메모리를 사용하였다^[1]. 제약 조건그래프 생성 시 각 설계규칙에 대하여 부울 연산과 섀도우 방법을 조합하여 구현한 것과 배선길이 최소화 시 점진적 기법을 이용한 것이 성능 차이의 주요 원인으로 추측된다.

그림 13의 TRIPS는 Technology Re-Targeting for IP System 의 약자로서, 본 연구의 연구로 응용된 설계용 소프트웨어의 이름이다. DREAM은 현재 상용화된 이식용 소프트웨어의 이름이다. 약 1,792개의 트랜지스터로 구성된 회로를 자동이식기를 사용하여 초기 면적인 550 × 616.5마이크론에서 305.3 × 291.2 (306.9 × 290.9)로 축소된 결과를 얻었다. 또한, 조그를 사용하면 각각 302.6×320.5(301.7×320.5)마이크론의 결과를 얻었으며, 거의 면적에서는 차이가 없음을 알 수 있다. 그러나 수행결과는 27배 속도로 월등 우월할 것임을 알 수 있다.

본 연구에서 구현한 이식 기능의 효율성을 검증하기 위하여 상용 소프트웨어인 미 Sagantec사와 비교하여 실험하였다. 사용 시스템은 선 스팩20, 솔라리스 2.5.1이다. 수행 속도에는 평균 27.4배의 차이가 나며, 메모리는 경쟁 제품에 비해 평균 41%의 메모리를 사용하였다. 트랜지스터의 크기에 따라 수행시간이 영향을 받는데, 회로 5의 경우 25,000개의 트랜지스터 회로부터는 기하급수적으로 수행시간이 길어지며, 그 시간도 15시간 10분으로써, 회로 4와 대비하여 트랜지스터 수는 5.8 배 증가한 반면 수행시간은 9.75배로 증가하는 현상을

표 1. 실험 결과

Table 1. Benchmark(Speed and Area).

회로	TR수	면적(μm^2)		수행속도 (초)		메모리 양(MB)	
		경쟁	연구	경쟁	연구	경쟁	연구
1	374.0	16.98	17.16	23.0	8.4	5.0	2.0
2	1.8K	89.27	88.90	1.5K	55.0	22.0	10.0
3	2.5K	58.59	60.14	3.2K	123.0	32.0	13.0
4	4.3K	235.0	237.5	5.6K	201.0	54.0	20.0
5	25.0K	1431.5	1435.3	54.6K	2.0K	272.0	119.0
6	39.0K	X	2159.7	X	3.3K	X	179.0
7	52.0K	X	2867.0	X	6.1K	X	239.0
평균			1.0		27.4배		41%

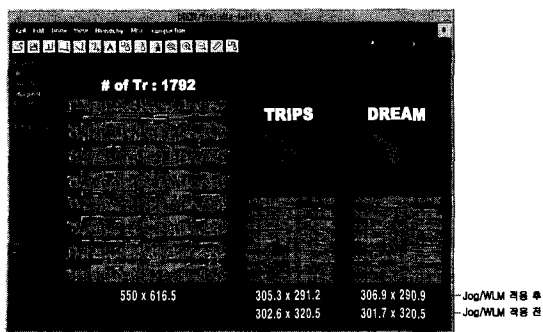


그림 13. 수행 결과 비교(면적)
Fig. 13. Performance Comparison(Area).

보이고 있다. 현재의 VLSI회로의 크기가 수십만 트랜지스터 이상이 되는 연유로 경쟁사의 수행 시간은 경쟁력을 잃어버리고 있다. 본 연구는 트랜지스터의 크기에 산술적인 증가를 보이고 있어 상용의 기하급수적 증가와 비교할 때 경쟁력을 짐작 할 수 있다. 또한, 회로 6,7의 경우에는 경쟁사의 메모리 사용량이 벤치마킹에 사용된 시스템의 메모리 한계를 넘어서고 있기 때문에, 수행결과를 얻을 수 없어 처리 용량에도 문제시되는 실정이다.

V. 결 론

레이아웃 이식은 마스크 레이아웃과 관련된 다양한 기술들로 구성된 복합기술이라고 할 수 있다. 특히, 컴팩션기술은 그 중에서도 이식의 성능과 품질을 좌우하는 핵심 부분이며, 실험결과에서 보여 주듯이 현재 상용 소프트웨어와 벤치마크 테스트를 해본 결과 상당히 좋은 결과를 얻었다. 동일한 조건에서 테크놀러지의 최소 설계규칙을 적용하였으므로 면적은 거의 유사하지만, 수행속도(성능)나 메모리 사용측면에서 월등한 결과를 얻었다. 이와 같은 결과를 얻었다는 사실만으로도 충분한 경쟁 가능성이 있다고 사료된다. 조그 삼입과 배선길이 최소화기술을 적용하여 면적 최적화와 더불어 컴팩션 기능 및 성능을 보장하고, 레이아웃 이식에서 가장 문제시되고 있는 처리용량의 한계를 극복하기

위하여 평면구조의 분할처리 기능을 구현 하였으며, 계층구조를 효율적으로 처리할 수 있는 연구개발이 중점적으로 진행되었다. 이외에 향후 VDSM(Very Deep Submicron : 0.25 μ m이하) 공정기술에서 발생할 수 있는 다양한 문제점들, 즉, 배선간의 상호 간섭현상(Crosstalk), 안테나 효과(Antenna Effect), 제조장비의 광학적 근접효과 보정(OPC : Optical Proximity Correction), 공정에서 수율(yield)에 미치는 다양한 요인들을 레이아웃 이식 시스템에 반영될 수 있도록 지속적인 연구가 되어야할 것이다.

참 고 문 헌

- [1] 이윤식 외, "반도체 레이아웃 자동이식과 수율 향상에 관한 연구," 제28회 정보과학회 춘계 학술 발표회, 제28권 1호, pp 25~27, 2001
- [2] D.G. Boyer, "Symbolic Layout Compaction Review," Proc. of the 25th ACM/IEEE DAC, pp.383~389, 1988.
- [3] J. Fang, et. al. "A New Constraint Graph Generation Algorithm for VLSI Layout Compaction," Proc. ISCAS. pp.2858~2861, 1991.
- [4] John Lakos, "Technology Re-targeting for IC Layout," Proc. of the 30th DAC, 1997.

저 자 소 개



李潤埴(正會員)

1981년 2월 : 서강대학교 전자공학과 학사. 1983년 2월 : 한국과학기술원 전산학 석사. 1994년 12월 : 미 남 플로리다주립대 컴퓨터공학 박사. 1983년 3월~1989년 1월 : LG전자 선임. 1990년 1월~1994년 4월 : 미 Center for Microelectronics Lab. 책임. 1994년 8월~1998년 6월 : LG반도체 책임. 1998년 7월 : (주)파이손테크 대표. 2000년 8월 : 호서대 전문대학원 교수, (주)에이케드 연구소장. <주관심분야> VLSI CAD 알고리즘, Design Automation, Hardware Accelerator

申 晚 澈(正會員)

2000년 2월 : 대전대학교 전자공학과 졸업. 2000년 3월 : 호서대학교 벤처전문대학원 재학. <주관심분야> VLSI CAD 알고리즘, 반도체 Test 분야



金塔培(正會員)

1985년 2월 : 아주대학교 전자계산학과 졸업. 1997년 2월 : 아주대학교 대학원(공학석사, 전산학전공). 1987년 1월~1998년 8월 : LG반도체 책임연구원. 1994년 3월~1995년 3월 : (미)일리노이대 Coordinate Science Lab. 연구원. 1998년 8월~2000년 8월 : (주)에이케드 이사. <주관심분야> VLSI CAD 알고리즘, Physical Design, Silicon Compiler

金 俊 泳(正會員)

2000년 2월 : 대전대학교 전자공학과 졸업. 2000년 3월 : 호서대학교 벤처전문대학원 재학. <주관심분야> VLSI CAD 알고리즘, 반도체 Test 분야