

# 직사각형을 기반으로 하는 레이아웃 개체추출 알고리즘

최용석, 천익재, 김보관

충남대학교 전자공학과

전화 : 042-821-7707 / 핸드폰 : 011-9802-1602

## Development of a Rectangle-based Layout Object Extraction Algorithm

Yong-Suk Choi, Ik-Jae Chun, Bo-Gwan Kim

Dept. of Electronics Engineering, Chungnam National University

E-mail : sfmania@cnu.ac.kr

### Abstract

In this paper we present a new hierarchical layout object extraction algorithm, which is based on rectangles rather than edges. The original layout data is modeled as instances connected by wires. Each polygon shape is divided into a set of rectangles and the instances and wires are extracted and recognized from those rectangles together with their connection and size information.

We have applied the algorithm to actual layouts. Experiments on several standard cell library demonstrate the effectiveness of the algorithm.

### I. 서론

현재 IC 개발 기술이 진보하면서 공정의 변화가 빨라지고 점점 더 작은 공정으로 변화하고 있다. 이와 더불어 회로의 복잡도가 증가하고 그에 따라 디자인 생산성의 저하가 심해지고 있다. 이러한 상황에서 좋은 성능과 신뢰성을 갖는 검증된 회로를 설계하는 것은 많은 시간과 노력을 필요로 한다.

---

본 연구는 반도체 설계 교육 센터(IDECE)의 지원에 의하여 수행되었습니다.

이러한 문제를 해결하기 위해 기존에 설계된 설계 정보를 정확히 인식하여 회로 요소를 추출하고 이 정보를 이용하여 대상 공정에 맞는 최적화된 레이아웃을 생성하는 회로요소 재사용 시스템이 개발되었다[1].

본 논문에서는 디지털 회로의 회로 요소 재이식 시스템의 기본이 되는 레이아웃 개체 추출 알고리즘을 개발하였다. 이를 위하여 기존의 polygon에 기반한 레이아웃을 rectangle화하고 rectangle화 된 레이아웃 정보를 object화하여 회로요소를 인식하고 추출하였으며 인식된 회로요소를 데이터베이스화하는 "Layout Extraction Engine"을 개발하였다. 개발된 extraction engine은 본 실험실에서 보유한 standard cell library와 계층적 구조를 갖는 macro cell에 대하여 만족할 만한 성능을 보여 주었다.

### II. 본론

현재 기존 회로(IP) 재사용에 있어서 많은 부분을 hard IP가 차지하고 있다. 이는 hard IP가 soft IP에 비하여 공정에 보다 더 최적화 되어 있고 고성능, 고집적의 특성을 갖고 있으며 레이아웃을 인식하여 대상 공정의 특성에 맞게 재이식 함으로써 기존 회로의 성능 및 특성을 유지할 수 있기 때문이다. 반면 soft IP의 경우 대상공정의 변화에 따른 최적화를 위하여 P&R 과정과 검증과정을 반복적으로 수행해야 한다는

단점이 있다. Hard IP의 재사용에 있어서 초기 회로 요소 재이식의 방법은 대부분 선형적인 scale의 변환과 수작업을 통하여 이루어졌다. 그러나 DSM technology로 공정이 변화하면서 선형적인 scale 변환만으로는 대상공정에 맞는 레이아웃을 생성하는 것이 불가능해 지고 있다. 그러므로 이러한 공정변화를 최대한 반영할 수 있는 레이아웃 인식 및 추출 방법이 요구된다고 할 수 있다. 그림1은 레이아웃의 회로요소를 추출하는 과정을 도식화하여 나타낸 것이다.

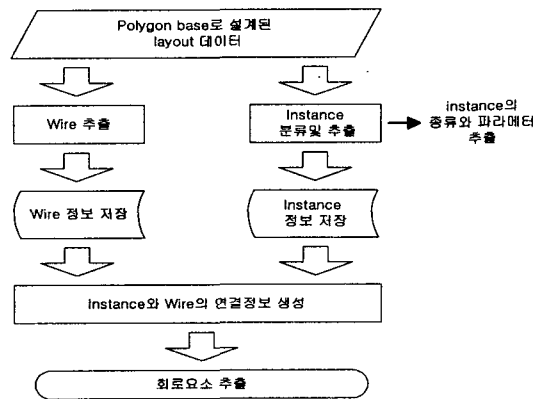


그림1 회로요소 추출 흐름도

### 2.1 회로요소 추출방법

기존의 extraction 방법으로 edge base 방법이 있다. 이 방법은 polygon layer의 edge를 사용하여 instance와 wire를 정의하고 인식한다. 이 방법은 모든 instance에 대해 선형 스케일에 의한 처리가 가능하나, 개개의 instance와 wire에 대해서는 최적화가 어렵고 회로 요소 각각에 대한 parameter 조절이 용이하지 않다.

그러나 본 논문에서 제안하는 rectangle base 방법은 polygon을 rectangle로 나누어 object화하여 인식한다. 따라서 각각의 개체간에 연결정보와 위치 이동, 그리고 parameter 조절을 매우 쉽게 처리할 수 있으며 연결정보의 추가와 삭제도 용이하다. 이와 더불어 레이아웃을 polygon형태의 data로 처리하는 것에 비하여 데이터구조 및 그 조작방법이 쉽고 간단하며 더 높은 수준의 추상적 device로 추출되므로 extraction이후의 과정에서 다루기가 편하고 cell 내부의 net-list 묘사가 명확하다. 또한 연결정보에 의한 계층 구조 처리가 용이하다.

### 2.2 Object의 정의

레이아웃을 object형식으로 추출하기 위해서는 먼저 object를 정의해야 한다. Object는 transistor, contact 등과 같은 instance와 이러한 instance들을 연결하는 wire로 이루어진다. Instance는 레이아웃을 구성하는 layer에 "Boolean AND"식을 적용하여 rectangle형태로 추출하며 transistor, diffusion contact, poly contact, bias contact, via, jog등이 이에 해당한다. Wire는 horizon/vertical wire와 instance인 jog로 이루어지며 polygon layer로부터 rectangle형태로 추출된다. Wire는 연결 형태에 따라 좌-우를 연결하는 horizon wire와 상-하를 연결하는 vertical wire로 나뉘며 jog는 vertical wire와 horizon wire를 연결하거나 그 이외의 다양한 용도로 사용된다. Object를 구성하는 instance는 표1에, wire는 표2에서 정의하였다.

표1 Instance 정의

종류	Instance 정의
Transistor	N 또는 P type diffusion layer와 Polysilicon layer의 영역이 겹쳐지는 곳이 transistor의 active영역이다
Diffusion contact	Transistor의 source, drain 역할을 하는 instance이다. diffusion layer와 metall layer가 함께 겹쳐지는 곳에서 추출한다. Contact, metall, diffusion layer로 구성된다.
Poly contact	Polysilicon과 Metall layer를 연결하는 Contact에서 추출한다. Contact, polysilicon, metall layer로 구성된다.
Bias contact	Power line에 존재하여 body effect를 줄이기 위해 사용한다. Power line에 따라 VDD bias contact, GND bias contact으로 구분된다.
Via	서로 층이 다른 두 metal layer와의 연결에 사용
Jog	Metal Jog, poly Jog, Diff jog가 존재한다.

표2 Wire 정의

종류	Wire 정의
Horizon	instance를 수평으로 연결하는 wire
Vertical	instance를 수직으로 연결하는 wire

### 2.3 BOOLEAN AND에 의한 instance 추출

추출할 object의 data structure를 정의한 다음 레이아웃을 이루는 polygon layer들의 조합으로 이루어진 transistor, contact, wire등을 인식하고 추출하는 과정을 수행한다. 예를 들어 N-transistor와 contact을 추출하기 위하여 N-transistor는 Polysilicon과 N-diffusion

## 직사각형을 기반으로 하는 레이아웃 개체추출 알고리즘

의 layer가 겹쳐있는 범위를 active영역으로 정의한다 (식 1). 이와 마찬가지로 poly contact은 metal layer와 polysilicon layer. 그리고 contact layer가 겹쳐있는 영역으로 정의한다(식 2).

$$N\text{-transistor area} =$$

$$\text{Polysilicon AND diffusion (1)}$$

$$\text{Poly contact area} =$$

$$\text{Metal AND Polysilicon AND Contact (2)}$$

이와 같은 방법으로 레이아웃을 Transistor, diffusion contact, poly contact, bias contact, via의 개체로 나누어 추출한다. 이때 추출된 instance는 모두 rectangle 형태로 추출된다. Diffusion contact은 multi-contact을 지원하기 위해 공통된 diffusion layer와 metal layer에 있을 경우 각각의 contact을 하나로 합치고 design rule에 의해 분할한다. 각 instance에 대한 BOOLEAN AND표현은 표3과 같다.

표3 Boolean method table

이름	BOOLEAN Method
n-transistor	polysilicon AND N-diffusin AND NOT(N-well)
p-transistor	polysilicon AND P-diffusin AND N-well
n-diffusion contact	metal 1 AND N-diffusin AND NOT(N-well) and contact
p-diffusion contact	metal 1 AND P-diffusin AND N-well AND contact
poly contact	polysilicon AND metal 1 AND contact
via	metal_bottom AND via AND metal_upper
VDD bias contact	metal 1 AND p-diffusin AND contact AND N-well
GND bias contact	metal 1 AND n-diffusin AND contact AND not(N-well)

### 2.4 Polygon layer에서 wire와 jog의 추출

레이아웃을 구성하는 layer는 인접하는 같은 성분의 layer에 대해 merge되어 있고, polygon layer의 edge point(X,Y)는 반 시계 방향으로 정렬되어 있다고 가정한다. Wire인식에 있어 polygon layer는 rectangle 형태로 나누어진 vertical/Horizon wire와 이들을 연결하는 jog로 인식된다.

Polygon layer를 wire와 jog로 나누기 위해 먼저 polygon layer의 모든 edge 좌표를 정렬한다. 그 다음 반 시계 방향의 polygon edge pass를 따라 돌면서 edge point를 고려한 가상 point를 삽입한다. 삽입된 가상 point는 이후 polygon을 rectangle로 분할 할 때 사용된다.

Polygon edge pass를 따라 추가된 가상 좌표와 반 시계방향의 pass를 참고로 ractangle형태로 나누어진 레이어는 좌우로 다른 레이어와 연결된 Horizon, 상하가 다른 레이어와 연결된 Vertical, 이 외의 경우는 jog로 인식한다. 이후 같은 종류의 wire끼리 인접한 경우 하나로 merge하는 과정을 거치게 된다.

그림2에서 (a)는 추출하기 전의 polygon 레이어, (b)는 추출하기 위해 polygon edge에 가상 point를 추가한 예, (c)는 가상 point와 edge point에 의해 polygon이 분할된 예이다. 마지막으로 (d)는 위의 과정을 통해 rectangle로 분할되어 wire로 추출된 결과를 보여주고 있다.

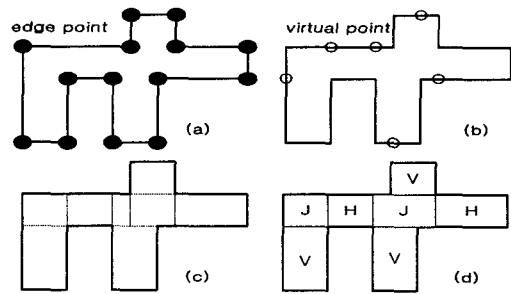


그림2 Wire 분할과정

### 2.5 Wire와 instance간의 연결정보 추출

추출된 instance와 wire간의 연결정보는 instance의 종류와 ID를 wire의 data structure에 저장함으로써 구현할 수 있다. Wire data structure는 horizon의 경우 좌우의 instance 연결정보를, vertical의 경우 상하의 instance 연결정보를 알 수 있도록 구성되어 있다.

먼저 instance와 같은 위치에 있는 wire 또는 jog를 찾는다. 찾은 object가 wire이면, 아직 연결정보가 정의되지 않은 쪽에 instance와의 연결을 정의한다. 만약 양쪽 모두 다른 instance와 연결되어 있다면 wire를 둘로 나눈다. 나누어진 두 wire의 끝은 아직 연결정보가 정의되어 있지 않으므로 이곳에 instance와 연결정보를 정의한다. 찾은 object가 jog일 경우 horizon 또는 vertical wire를 삽입하여 instance와 jog의 연결정보를 생성한다.

## III. 계층 구조를 갖는 레이아웃의 추출

계층구조를 갖는 레이아웃은 두개 이상의 cell 레이아웃, 혹은 cell 레이아웃과 cell 레이아웃간의 신호를 연결하는 wire로 구성된다.

계층적 구조를 유지한 회로요소를 추출하기 위해서는 먼저 계층구조의 레이아웃 디자인에 사용된 cell의 종류와 위치, wire로 사용된 metal layer를 파악한다. 그 다음 cell 각각에 대해 레이아웃을 object화하여 회로요소를 추출한다. cell간의 신호를 연결하는 wire는 rectangle화하여 추출한다.

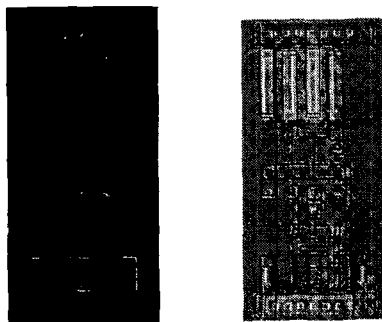
추출된 wire의 type에 따라 Horizon wire는 좌/우, Vertical wire는 상/하의 연결정보를 입력한다. 연결정보는 wire와 연결되어 있는 cell의 이름과 함께 셀 내부의 instance의 고유 ID를 줄 수 있다. Wire에 포함된 연결정보와 wire자체의 위치 정보를 이용하여 계층구조로 디자인된 레이아웃을 계층구조를 유지한 채 추출할 수 있다.

#### IV. 추출된 회로요소의 검증

본 논문에서 제안한 인식 알고리즘을 검증하기 위하여 자체 설계한 0.6um standard cell library, 0.6um datapath cell library, 0.25um standard cell library, 그리고 이를 이용하여 만든 macro cell을 이용하였다. 그리고 인식된 레이아웃을 확인하기 위해 자체 개발한 "l-view(Layout viewer)"를 사용하였다.

0.6um공정의 cell은 78종 94점, 0.25um공정의 cell은 133종 logic gate와 56종 storage element. 그리고 34종의 special function으로 구성되어 있으며 모든 cell에 대해 100% 인식하였다. 계층적 구조에 대한 검증은 0.25um공정의 standard cell을 이용한 4X4 mux, 8X8 mux, 16X16 mux와 0.6um공정 standard cell과 datapath cell을 사용한 ALU, Bit counter, Shift resistor에 대해 테스트하였으며 계층적 구조의 연결정보 추출과 함께 회로요소를 100% 인식하였다.

추출된 회로요소는 l-view를 통해 상용 tool인 cadence의 virtuoso와 비교하였고, 자체 compactor tool을 사용하여 net-list와 DRC를 검사하였다. 그림3은 original layout과 추출된 layout을 보인다.



(a) original

(b) extract

그림3 회로요소 인식 예

#### V. 결론

점점 더 미세화, 고집적화 되어가는 공정의 발전에 따라 늘어나는 time to market을 줄이기 위한 해결책으로 이전에 설계된 IP를 재사용하는 방법이 사용되고 있다. 이를 위해서는 먼저 설계된 Hard IP에서 필요한 회로요소를 정확히 추출하는 과정과 현재 공정에 맞추어진 Hard IP를 새 공정에 맞추어 주는 과정이 필요하다.

본 논문에서 개발된 extraction engine은 현재 본 실험실에서 보유하고 있는 0.25um standard cell library와 0.6um datapath cell library, 그리고 계층구조를 갖는 macro cell에 대해 회로요소를 100% 인식하였다. 이 결과는 회로요소를 object형태로 추출함으로써 device의 parameter analysis 및 instance와 wire간의 net-list 생성, 각 회로요소의 handling을 쉽게 할 수 있음을 보여준다.

따라서 본 논문에서 제안한 rectangle에 기반한 개체 인식 방법은 추후 회로요소 재이식 과정에서 계층적 처리 및 회로요소 resizing을 용이하게 할 수 있고 45도 처리를 추가하여 좀 더 유연성 있는 layout 재구성을 가능하게 할 수 있으며, 회로요소를 추출하는 extractor의 특성상 회로 성능 분석기[2]로 사용할 수 있을 것이다.

#### 참고문헌

- [1] 천익재, Standard Cell Library를 위한 Layout Retargeting 알고리즘 개발, 석사학위 논문, 충남대학교, 2000.
- [2] Alcantara, "A Novel Circuit Extraction Tool Based on X-Spans and Y-Spans," IEEE Proceedings of EUROMICRO-22, 1996.
- [3] Mohan Guruswamy, "CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries," Design Automaton Conference, pp.327-332, 1997.
- [4] Jean-Claude Dufourd, "The STICKIZER : A Layout To Symbolic Converter," IEEE International Conference on Computer Aided Design, Page(s): 534 -537, 1989
- [5] John Lakos, "TECHNOLOGY RETARGETING FOR IC LAYOUT," ACM, Design Automation Conference, Proceedings of the 34th, Page(s): 460 -465, 1997.