

# 마스크 레이아웃의 등가 회로 추출을 위한 블록 분할 기법

손영찬, 주리아, 박석홍, 유상대  
경북대학교 전자전기공학부

전화 : (0562) 245-1245 / 팩스 : (0562) 252-5005

## (A Block Disassembly Technique for Equivalent Circuit Extraction of Mask Layouts)

Yeong-Chan Son, Ri-A Ju, Seog-Hong Park, and Sang-Dae Yu  
School of Electronics and Electrical Engineering Kyungpook National University  
E-mail : ycson@pohang.ac.kr

### Abstract

In this paper, we describe an automated extraction program that transforms a mask layout into an approximated equivalent circuit information suitable for circuit simulation, and that extract the geometric parameters of stacked MOSFETs and the distributed RCs of layout blocks. To extract equivalent circuit from mask layout, we propose new block disassembly technique capable of accurate computations of distributed RCs at branch point, using vectorized edges which represent the outline of an individual polygon.

### I. 서론

최근의 급속한 집적회로 공정기술의 발전으로 레이아웃 배선이 가지는 분산 RC (distributed RC)는 회로의 특성을 왜곡시키는 원인으로 작용하고 있다. 레이어 상호간의 연결에서 가능한 최소의 기생성분이 생성되기 위해서 다중접점으로 레이어를 연결하거나 또는 MOSFET의 확산영역의 분산 저항을 최소화하기 위하여 스택구조로 설계하여 자연적으로 생성되는 기생 성분의 영향을 최소화 되도록 설계하고 있다. 자동 레이아웃 생성기를 이용하여 설계하는 경우도 있지만 고성능의 연산증폭기와 같은 회로를 설계하는 경우에는 모든 접점과 소자를 레이아웃하고 라우팅을 직접 해야하는 경우가 많으며, 이러한 경우에 칩 제작에 앞서 설계된 레이아웃으로부터 회로를 추출하고 시뮬레이션을 통하여 회로의 성능을 검증해야 한다. 기생 성분에 대한 영향을 평가하기에 앞서 레이아웃의 논리적 회로가 정확하게

이루어졌는가를 ERC를 수행하여 레이아웃에 대한 전기적 연결관계를 우선적으로 평가해야 한다. 그 다음으로 기하학적인 구조에 따른 소자의 L/W 의 여러 가지 파라미터와 배선간에 생성되는 점점 저항과 분산 RC를 추출하여 보다 근접한 출력특성을 구해야 한다.

레이아웃의 등가 회로를 추출하는 검증도구로서 유한 요소법을 사용하여 각 단자 연결사이의 R과 C를 근사화하는 SPACE[1], 연결 단자간에 위치하는 타일에 의해서 만들어진 경로상의 면적과 길이에 의해 R과 C를 근사화하는 Magic의 회로 추출기[2,3], 그리고 레이아웃의 블록에 대한 패턴 매칭에 의한 라이브러리 룩업 (lookup) 방식을 이용한 EXCL[4] 등의 회로 추출 알고리즘이 널리 사용되고 있다. SPACE는 각 단자간의 연결 성분에 대한 3차원 구조의 저항과 커패시턴스 값을 정확하게 모델링할 수 있지만 CMOS SRAM과 같은 기본 셀에 대한 회로 추출 시간이 10분 이상이 소요되므로 회로의 구조가 복잡할수록 추출에서 검증까지 많은 시간이 소요된다. Magic의 회로추출기의 경우는 한 절점과 다른 인접 절점사이에 놓인 타일의 병합된 면적에 의해서 C를 계산하며, 만약 연결 단자 사이에 분기된 가지가 생성된 경우에 저항 성분을 추출할 때 연결가지에서 분기된 다른 가지를 무시하고 저항 값을 추출하기 때문에 분기가 형성된 경우에 두 절점사이의 추출된 저항 값에는 큰 오차가 발생한다. EXCL에서는 모든 블록에 대해 생성 가능한 블록에 대해서 라이브러리를 만들어 놓고 추출 과정에서 라이브러리를 참조하면서 추출기능을 수행하는데, 그 특징은 레이아웃에서 생성된 패턴을 라이브러리에서 그 값을 찾아 대응시키고 라이브러리에 해당하는 패턴이 없으면 그 패턴에 대한 근사값을 계산하여 라이브러리에 포함시키는 방식을 사용하고 있다. 그러므로 이 방법은 일정한 패턴과 일치하는 영역에 대해서 빠르게 필요한 등가회로를 접목할 수 있다. 그러나 라이브

러리에 포함되지 않은 패턴에 대해서는 여러 가지의 패턴을 추출과정에서 정형화하여 라이브러리화 해야하는 부가적인 과정을 필요로 하기 때문에 부차적인 작업 공정이 필요하다.

본 논문에서는 타일에 의해서 표현되는 Magic의 레이아웃으로 생성되는 도형에 대한 기하학적인 정보를 입력으로 각 레이어 별로 병합하여 독립된 블록을 생성하고 이 블록을 방향성 변으로 기술한다. 한 블록의 방향성 변과 인접한 다른 독립 블록과의 전기적 연결을 갖는 변을 사용하여 소자와 분산 RC를 추출할 수 있는 블록 분할 알고리즘을 제안한다. 제안된 알고리즘 기법을 사용하여 회로 추출 시스템을 구성하였다. 이 추출 시스템은 부차적인 작업이 없이 도형 정보가 형성하는 블록에 대한 등가 저항을 바로 추출하게 된다.

## II. 회로 추출 시스템의 구성

블록 분할 기법을 이용한 회로 추출기의 구성도는 그림 1과 같다. 기하학적인 정보로 표현된 CIF 또는 Magic의 데이터 형식인 타일 정보를 입력으로 받아들인다. 그리고 각 레이어에 대한 공정파라미터의 값과 레이아웃상의 입력력 단자와 전원 단자를 식별하기 위하여 레이아웃과 외부적으로 연결되는 접속 단자에 대한 정보를 도형 정보와는 별개의 입력으로 기술한다. 또한 입력력 데이터 중에는 출력형식을 결정하는 정보가 기술되어야 한다. 사용될 시뮬레이터의 종류를 기술해야 하고 분산 RC의 출력 모델을 지정해야 한다. 분산 RC의 출력 모델에는 lumped, T 및  $\pi$  등가 모델 등이 있으며, 또한 분산 RC와 접점 저항을 무시하고 스택 구조로 설계된 MOSFET을 단일의 소자로 인식하여 논리적 회로를 출력할 수 있는 모델이 포함된다. 이것은 기생성분의 영향을 고려하지 않고 논리적 연결관계만을 판단할 때

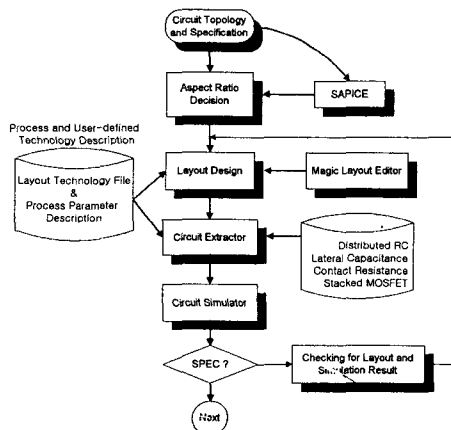


그림 1 회로 추출기의 구성도  
Fig. 1 Diagram of circuit extractor.

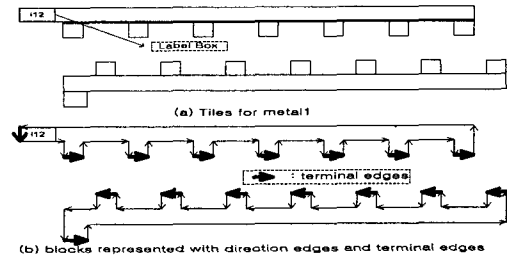


그림 2 방향성 변으로 표현한 블록  
Fig. 2. Blocks represented with directional edges.

사용할 수 있는 모델이다.

본 추출기에서 추출하는 성분은 접점 저항 성분, 단일 또는 스택 구조로 설계된 MOSFET 소자와 소자의 기하학적인 파라미터, 소자와 소자를 연결하는 선에 대한 분산 RC, 그리고 레이어의 측면에 위치하는 커패시턴스(peripheral capacitance) 등이다. 기하학적인 도형 처리에서는 레이어별로 그룹을 만들어 데이터 구조에 저장하고 분류한다.

Magic의 타일을 각 레이어별로 분류하고 분류된 타일을 edge-based scanline 알고리즘을 사용하여 병합한다. 하나의 레이어에 대하여 다수의 블록이 구성되며 각각의 블록은 반시계 방향의 변에 의해 연계리스트(linked-list)로 표현한다. 추출할 블록의 변과 인접하게 되는 변을 terminal edge (TE)로 정의하고, TE는 회로 추출과정에서 전기적으로 또 다른 레이어와 연결을 가지므로 독립된 하나의 절점으로 표현하기 위해서 terminal node (TN)로 정의한다.

그림 2(a)는 타일로 표현된 스택구조의 MOSFET에서 공유된 드레인과 소스 단자와 연결된 metal1에 대한 레이아웃이다. 그림 2(b)는 각 타일을 병합하여 생성된 독립 블록의 외형을 방향성 변으로 표현하였으며, metal1 블록의 변 중에서 접점과 연결을 가지는 TE를 나타낸 것이다. 입력단자 또는 외부 전원 연결을 나타내기 위하여 레이블 블록의 변과 접치는 블록의 변을 TE로 표현한다. 그림 2(b)에서 'i12'는 외부 전류 전원이 연결되는 것을 나타내기 위한 레이블인데 TE로 분류된다.

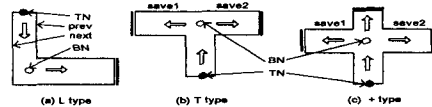


그림 3 회로 추출에 사용되는 기본 구조  
Fig. 3 Basic structures for extraction.

연계리스트에서 TE로 표현된 블록을 사용하여 각 블록에 대한 등가의 분산된 RC 추출을 시작한다. 그림 3(a)는 선택된 TE와 연결된 앞뒤의 두 변에서 추출 진행 방향과 같은 변을 next, 반대 방향의 변을 prev로 표

시하고, 이들 쌍을 EP (edge pair)라고 정의한다.

EP는 수직으로 또는 수평으로 나란히 위치하고 진행 변의 등가 분산 RC 추출의 기초가 된다. EP가 진행하면서 다음 진행이 생성될 수 있는 형태는 L, T, 또는 +자 형태로 그림 3과 같으며, 진행 중에 진로가 변경되는 위치를 branch node (BN)로 표기하였다. L자형으로 형성되는 곳에서는 새로운 절점 생성은 되지 않는다.

next->next와 쌍을 이루는 변을 save1, 이들 쌍을 NEP(next edge pair)라 하고, prev->prev와 쌍을 이루는 변을 save2, 이들 쌍을 PEP (prev edge pair)라고 정의할 때, NEP의 진행 방향은 next->next에 의해서, PEP의 진행 방향은 save2에 의해서 결정된다. NEP와 PEP가 모두 형성되고 save1과 save2가 동일한 변이거나 save1->prev와 save2->next가 동일한 변이면 T자형이 형성된다. 그림 3(b)의 NEP는 (next->next, save1)이고 PEP는 (save2, prev->prev)인데 이 경우에는 save1과 save2가 동일한 변이다. 그림 3(c)는 (save1->next, save2->prev)을 쌍으로 하는 새로운 진행 방향이 생긴다. Manhattan 기하에서는 +자 형의 경우처럼 최대 3개의 가지가 생성될 수 있다. T자와 +자 형이 형성된 곳에서는 새로운 절점 (BN)이 생성된다.

TE에서 추출 과정을 시작하여 또 다른 TE를 만나면 한 개의 가지에 대한 가지자르기가 완성되고 한 개의 부분 블록에 대한 분산 RC 추출을 마친다. Edge-based 블록 분할 알고리즘에 대한 가상 코드는 그림 3과 같다. L, T, 또는 +자로 형성되는 절점에서 저항 계산은 근사화된 값[4]을 참조하여 대응시킨다.

```
void recursiveCall( EdgeList *next, EdgeList *prev )
{
    if a new branch created in only prev side
        save1 = findCounterpartEdge( prev->prev );
        if save1 == next->next // L type
            recursiveCall( save1, prev->prev );
        else // T type
            recursiveCall( save1, prev->prev );
            recursiveCall( next, save1->prev );
    else if a new branch created in only next side
        save2 = findCounterpartEdge( next->next );
        if save2 == prev->prev // L type
            recursiveCall( next->next, save2 );
        else // T type
            recursiveCall( next->next, save2 );
            recursiveCall( save2->next, prev );
    else if branches are created in both directions // T type
        save1 = findCounterpartEdge( prev->prev );
        save2 = findCounterpartEdge( next->next );
        recursiveCall( save1, prev->prev );
        recursiveCall( next->next, save2 );
        if the third branch is created // + type
            recursiveCall( save2->next, save1->prev );
}
```

그림 4 블록 분할에 대한 가상 코드  
Fig. 4 Pseudo code for block disassembly.

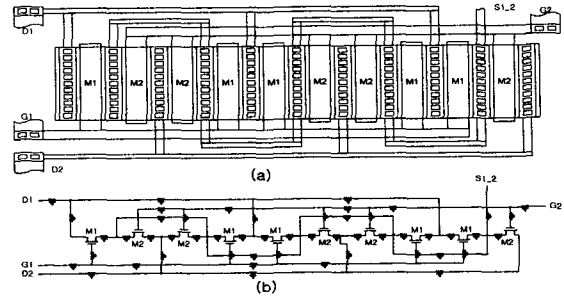


그림 5 스택 구조의 MOSFET에 대한 등가회로  
Fig. 5 Equivalent circuit for stacked MOSFETs.

평행판에 대한 상수 값과 측면 모서리에 대한 상수 값을

$$K_{plate} = \frac{\epsilon_{oxide}}{depth} \text{ 와 } \alpha_{fringe} \text{로 나타내면 커패시턴스는}$$

$$C_{ab} = K_{plate} \cdot (W \times L) + \alpha_{fringe} \cdot (2W + 2L_{prev}) \quad (1)$$

로 근사화할 수 있다. 두 절점 a에서 b까지 가상 절점이 m개 포함되어 있을 경우 RC 계산은

$$R_{ab} = \sum_{i=1}^m \rho_{sh} \cdot \frac{L_i}{W_i} \quad (2)$$

$$C_{ab} = \sum_{i=1}^m (K_{plate} \cdot (W_i \times L_i) + \alpha_{fringe} \cdot (2W_i + L_{i,prev} + L_{i,next})) \quad (3)$$

로 근사화 한다. 여기서 가상 절점은 절점 a에서 b까지의 진행 중 경로가 90°로 변하거나 폭이 변하는 곳의 위치에서 생성되는 절점이다. 레이어의 측면 성분의 C를 추출하는 경우에 추출의 진행 성분이 next와 prev 등의 변으로 이루어졌기 때문에 이들 변의 길이를 포함하면 위의 식 (3)에서 (L<sub>i,prev</sub> + L<sub>i,next</sub>)와 같이 EP 변의 길이로 계산할 수 있다.

그림 5(a)는 스택구조의 MOSFET을 나타내고 있으며, 스택구조를 구성하는 배선 성분을 추출 결과에 포함하였다. 금속과 확산 영역의 접점 또는 인접한 다른 레이어와의 연결에서 다중접점이 형성된 경우에는 접점 저항의 등가 모델은 공정 파라미터에서 참조한 접점 저항의 값에서 다중 접점의 수로 나눈 저항 값을 인접한 TN사이에 직렬 연결하였다. 그림 5(b)의 기생성분을 포함하여 추출된 회로망은 등가 RC에 대한 추출 모델의 등가회로에 대한 값으로 출력된 것이다.

### III. 회로 추출 및 실험 결과

제안한 블록 분할 기법을 사용하여 레이어아웃으로부터 소자와 상호 연결관계 정보를 추출하기 위해 그림 6(a)의 폴드드 캐스캐드 CMOS 연산증폭기를 사용하였다. 레이어아웃은 그림 6(b)에, 각 레이어 별로 병합한 결과를 그림 6(c)에 나타내었고, RC 추출 모델은 lumped 모델

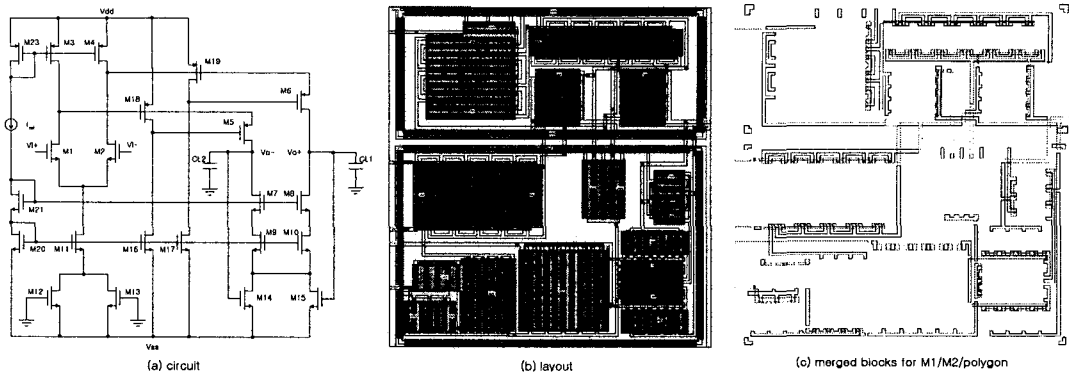


그림 6 조정된 캐스코드를 가지는 폴디드 캐스코드 CMOS 연산증폭기

Fig. 6 A folded-cascode CMOS operational amplifier with regulated cascodes.

및  $\pi 3$  모델로 추출하여 결과를 표 1에 정리하였다. 추출된 회로의 성능을 검증하기 위하여 현대  $0.8\mu\text{m}$  CMOS 설계 공정[5]의 HSPICE 레벨 13 BSIM1을 사용하여 시뮬레이션한 결과를 그림 7에 나타내었다.

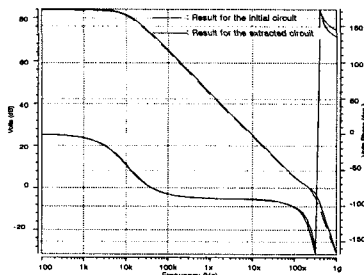


그림 7 시뮬레이션 결과

Fig. 7 Simulation Results.

표 1 그림 6의 레이아웃으로부터 추출한 결과

Table 1 Elements extracted from layout in Fig. 6.

# of elements	Initial circuit	Extracted model		
		lumped	$\pi 3$	
PMOS	7		39	
NMOS	15		79	
Distributed RC	-	m1	m2	polysilicon
		251	16	232
Element	31	1143	3635	
Node	65	777	1771	

#### IV. 결론 및 추후 연구

본 논문에서는 백터화한 변으로 기하학적인 도형을 블록으로 표현하고 제안한 블록 분할 알고리즘을 적용하여 레이아웃을 검증하는 회로 추출 시스템을 구현하였다. 레이아웃 구조가 복잡하더라도 블록 분할에 의해서 구현되고 초기 입력 파일과 연동하여 회로 추출을 행하므로 회로 추출 시간에서 시뮬레이션 시간이 단축된다.

또한 추출 모델을 다양하게 할 수 있도록 하여 레이아웃의 논리적 회로 구성을 확인하거나, 보다 정확성 있는 분산 RC 모델로 추출할 수 있도록 하여 기생 성분이 회로에 미치는 영향을 평가할 수 있도록 하였다. 그러므로 레이아웃의 비대칭적 구조에 의한 회로의 출력 특성 변화를 예측할 수 있으므로 칩 제작 후의 성능 평가에 정확성을 가질 수 있다.

제안된 알고리즘이 보다 효율적으로 사용되기 위해서 절점 감소 방법을 접목하여 추출되는 회로의 절점 수와 소자의 수를 감소시킬 수 있는 기능이 포함되어야 한다. 기생 성분이 일정 수준이하일 때 인접한 절점사이에 추출된 소자를 병합할 수 있는 방법을 연구하고 있다.

#### 참고 문헌

- [1] A. J. van Genderen and N. P. van der Meijs, "SPACE: A finite element based capacitance extraction program for submicron integrated circuits," *NASECODE VI Conference*, 1989.
- [2] J. K. Ousterhout, G. T. Hamachi, R. N. Mayo, W. S. Scott, and G. S. Taylor, "The Magic VLSI layout system," *IEEE DESIGN & TEST*, Feb. 1985.
- [3] W. S. Scout and J. K. Ousterhout, "Magic's circuit extractor," *IEEE DESIGN & TEST*, 1986.
- [4] Steven P. McCormick, "EXCL : A Circuit Extractor for IC Designs," *Proc. of 21st Design Automation Conference*, pp. 616-623, 1984.
- [5] Hyundai Electronics Industries, *0.8 $\mu\text{m}$  Model Parameter*, April 1994