

방향성 변을 이용한 집적회로 설계검증용 회로 추출기

손 영 찬[†] · 박 석 홍^{††} · 유 상 대^{†††}

요 약

집적회로가 고집적화 그리고 고성능화 되어지면서 설계된 레이아웃의 전기적 연결 관계와 회로 성능을 검증할 필요성이 더욱 강조되고 있다. 본 논문에서는 레이아웃으로부터 소자의 기하학적인 모델 파라미터와 기생 저항과 커패시턴스 등을 포함하는 회로 정보를 추출하기 위하여 레이아웃 내의 모든 배선에 대한 도형을 방향성 변을 사용하여 독립된 다각형 블록의 집합으로 기술하고, 이 블록과 소자의 인접여부에 의해 MOS 트랜지스터와 전기적 연결 관계를 찾아서 회로를 추출할 수 있는 새로운 알고리즘을 제안한다.

A Circuit Extractor Using Directional Edges for Design Verification of Integrated Circuit

Yeong-Chan Son[†] · Seog-Hong Park^{††} · Sang-Dae Yu^{†††}

ABSTRACT

Due to the high density of integration and the high performance in current integrated circuit, it is more emphasized to verify the electrical connectivity and performance of the circuit extracted from the layout. In this paper, we propose a new algorithm to extract the netlist including the geometric parameters of MOSFETs and the parasitic resistance and capacitance values. Where a set of polygon blocks for routing wires in the layout are described with directional edges and the layout extraction is carried out by checking over the abutment of these blocks on MOSFETs.

1. 서 론

집적회로의 공정 기술 발전에 따라 단위 소자의 크기가 작아지고 상대적으로 칩의 집적도는 매우 급속하게 증가하고 있다. 이와 함께 수반되는 기생성분 등에 의한 문제점들도 더욱 증가하게 되었다. 특히 고속 신호 처리에 필요한 ADC(analog-to-digital converter)와 같은 고성능의 회로 설계시 제작된 칩의 성능은 회로

의 레이아웃 기법에 의해서 많은 차이를 가질 수 있다. 레이아웃 기법은 가능한 최소의 기생 성분이 포함되도록 설계되어야 한다. 그래서 MOSFET과 같은 소자의 경우에는 드레인과 소스의 확산 영역에서 생기는 저항을 감소시키기 위하여 이들 단자를 다중 금속 접점으로 만들고 있고, 또한 스택(stack) 구조로 소자를 설계하여 인접한 소자들의 드레인과 소스의 단자를 공유시켜서 기생 저항과 기생 커패시턴스의 크기가 최소가 되도록 설계하고 있다. 회로의 집적도가 기하급수적으로 증가하고 있고 기생 성분의 영향이 커짐에 따라 이러한 문제를 고려하여 회로의 자동 레이아웃 생

[†] 정 회 원 : 포항 1 대학 컴퓨터응용과 교수
^{††} 정 회 원 : 경북대학교 대학원 전자공학과
^{†††} 정 회 원 : 경북대학교 전자전기공학부 교수
논문접수 : 1998년 7월 31일, 심사완료 : 1998년 10월 8일

장, 설계 검증, 배치(placement) 및 라우팅(routing)에 대한 연구[1-6]와 CADENCE의 같은 상용 설계 도구의 보급도 일반화되고 있다.

레이아웃하기 전에 소자의 외형비(W.L)만으로 SPICE 시뮬레이션을 수행했을 경우에 설계 사양을 만족시키지 못하는 회로가 레이아웃한 후에는 원하는 성능을 가지지 못하는 경우가 있다. 그 이유 중의 하나는 연산 증폭기에서 차동단의 MOS 소자를 레이아웃할 때 전기적 배선의 대칭구조가 어긋나거나 민감한 부분에서 배선의 분산 저항과 커패시턴스의 영향이 큰 경우이다. 그리고 레이아웃과정에서 설계자의 오류가 원인이 될 수가 있다. 일반적으로 소자의 외형비가 설계치와 같도록 레이아웃되고 소자의 대칭성과 성합을 고려해서 설계되어져도, 소자를 연결하는 금속선 및 폴리실리덴과 같은 연결선에 대한 기생 저항과 커패시턴스가 무시되어 회로의 성능이 저하될 수 있다. 그리고 다른 레이어 사이를 전기적으로 연결하는 접점의 저항은 시트 저항의 수십 배에 이를 수 있기 때문에 고성능의 아날로그 회로에서 이러한 기생성분의 존재는 회로의 성능을 저하시킬 뿐만 아니라 그 회로의 고유기능을 수행할 수 없게 할 가능성도 있다. 더구나 CAD를 이용한 자동 레이아웃이 아닌 수작업으로 스택 구조의 형태, 소자의 배치, 라우팅을 하는 경우에는 자동으로 설계한 경우보다 설계자의 능력에 따라 설계 면적을 감소시킬 수 있고 성능을 어느 정도 증가시킬 수 있지만, 설계자의 실수에 의해 레이아웃에 오류가 개입될 수 있다.

이러한 문제점들을 칩을 제작하기 전에 설계된 레이아웃으로부터 스택 구조의 소자 인식, 이들 소자들을 연결하는 도선에 대한 기생 저항과 커패시턴스, 그리고 이들의 전기적 연결관계를 포함하는 회로 추출을 수행하고 추출된 회로의 시뮬레이션을 통한 검증이 이루어진다면 잘못된 연결된 소자를 찾아낼 수 있고 또한 설계된 회로의 성능을 보다 정확히 검증함으로써 칩 제작비용을 최소화할 수 있다.

기존의 설계 검증 도구로는 SPACE[7], SPECS[8], 그리고 Magic[9]의 회로추출기[10] 등이 있다. 유한 요소법을 사용하여 만들어진 SPACE는 3차원의 커패시턴스와 저항을 추출하고 단일 소자에 대한 보다 정확한 특성을 확인할 수 있으나 한 개의 CMOS SRAM과 같은 기본 셀의 레이아웃에 대해 회로를 추출하는 경우에 10분 이상이 시간이 소요되므로, 다수의 소자로 이루어진 실제 회로에 대한 회로 검증을 위한 추출에

는 많은 시간이 요구된다. SPECS는 연결 성분들에 대한 레이어를 사각형의 타일로 분할하여 이들의 인접 사각형과의 접촉형태에 따라 등가의 RC 성분을 추출하고 있기 때문에 한 개의 사각형이 두 개 이상의 사각형과 인접할 경우 한 개의 사각형에 대한 등가회로가 인접한 사각형의 개수만큼 생성되기 때문에 지나치게 많은 기생 RC 회로가 포함된다. 그리고 Magic의 회로 추출기는 타일의 인접에 의해서 연결되어진 절점 사이의 연결관계를 찾아내고 절점 사이에 나타나는 기생 성분을 계산한다. 그러나 Magic 회로 추출기의 저항 추출 과정에서 문제점은 절점 사이에 가치가 없을 경우에 군사적으로 사용되어질 수 있으나 절점 사이에 가치가 형성되어 있을 경우에는 추출 저항에 포함되지 않아야 할 가치에 대한 저항 값이 추출하고자 하는 저항 추출에 포함되어 있다. 이것은 단순히 추출하고자 하는 두 절점 사이에 연결된 모든 타일에 대한 길이와 면적에 의해서 분산 RC를 추출하기 때문에 추출된 저항 값은 많은 오차를 수반할 수 있다.

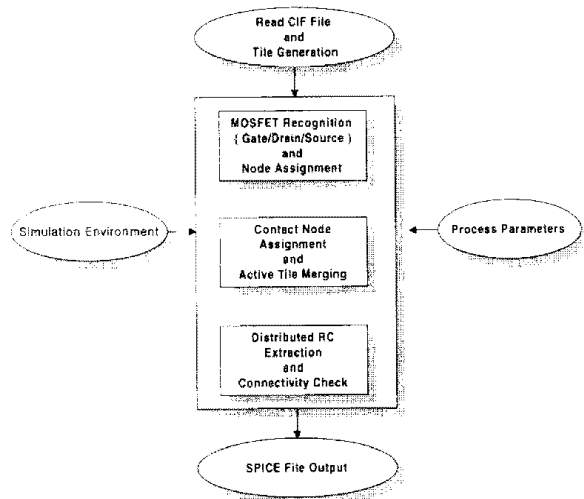
본 논문에서는 레이아웃을 트랜지스터와 배선 레이어로 구분하여 처리하며 스택 구조의 MOSFET 소자들을 인식하여 소자의 파라미터를 계산하고, 이들 소자들의 단자사이에 위치하는 연결선에 의해 구성되는 다각형의 변을 사용하여 소자 사이의 전기적 연결관계를 찾아내고, 소자 사이에 분포하는 분산 저항과 커패시턴스 등의 기생성분을 추출하기 위하여 변의 방향성을 이용한 새로운 알고리즘을 제안한다. 제안된 알고리즘의 사용으로 분기가 형성된 곳에서 Magic 회로 추출기가 저항 계산에서 가지는 문제점을 해결하였다. Magic에서 레이아웃 도형을 기술하는 타일을 병합하여 다각형을 생성하기 위하여 변을 기초로 한 스캔라인(edge-based scanline)[11,12] 연산을 수행한다. 이들 다각형의 외형선을 이루는 변에 방향성을 부여하고, 변과 소자들의 인접여부를 조사하여 소자들의 연결관계를 설정하고 그리고 다각형에서 가치가 형성되는 곳에 새로운 절점을 생성시키고 절점과 절점 사이의 분산된 RC의 값을 계산하게 된다.

회로 추출기의 입력은 Magic에서 레이아웃한 기하학적인 도형 정보와 CIF[13]이다. CIF일 경우에는 이것을 Magic에서 사용 가능한 타일로 변환하여 입력으로 사용할 수 있다. 레이아웃된 레이어들의 시트 저항과 커패시턴스에 대한 정보를 가지고 있는 공정 파라미터를 입력으로 읽어들이 분산 RC의 계산에 사용하

있고, 또 하나의 입력정보는 시뮬레이션 환경을 기술한 파일이다. 이 파일은 레이아웃에 표현된 전원 단자와 입출력 단자에 대한 레이블과 일치하는 이름을 직접적으로 사용한 신호원과 신호원 등을 포함하고 있다. 그리고 회로 추출 과정에서 출력은 SPICE에서 사용 가능한 회로 기술 파일이다. 본 논문의 2장에서는 회로 추출기의 전체 구성도를 나타내었고, 3장에서는 MOSFET 소자를 인식하고 소자의 단자에 직접 번호를 부여하는 과정을 기술한다. 4장에서는 레이아웃 도형을 병합하여 다각형을 형성하고 이들 다각형의 변을 이용하여 분산된 RC를 출력하기 위해 제안한 새로운 알고리즘을 기술하고 있다. 5장에서는 제안한 알고리즘에 의해 구현된 회로 추출기를 폴다드 케이스코드 CMOS 연산증폭기의 레이아웃에 적용하여 출력 파일을 만들고, 이 파일을 SPICE 시뮬레이션을 통하여 레이아웃 기전의 시뮬레이션 결과와 비교하여 레이아웃을 검증하였다. 끝으로 6장은 결론으로 구성하였다.

2. 회로 추출기의 구성도

회로 추출 과정의 구성도를 그림 1과 같이 나타낼 수 있다. 입력으로 받아들이는 기하학적인 도형 정보는 CIF와 Magic에서의 레이아웃 도형 mag를 입력으로 받아들인다. 방향성 변을 사용한 제안 알고리즘은 이들 입력 정보를 분석(parsing)하는 것으로 시작한다. 입력파일은 타일과 레이블로 구분된다. 먼저 MOSFET



(그림 1) 회로 추출 흐름도
(Fig. 1) Flowchart for circuit extraction.

소자 영역을 나타내는 타일을 구분하고 각 레이어 별로 타일을 분류하여 저장한다. 레이블은 입출력과 신호원에 대한 단자를 나타내는 것과 MOSFET 소자를 식별하기 위한 것이 있다. 이들 레이블은 출력파일 생성시에 사용되므로 타일과는 별도로 저장한다. 각 분류된 레이어에 대해서 이들을 병합하여 다각형을 구성하고 다각형을 구성하는 연결된 변들에 대해서 방향성을 가지게 한다. 한 개의 독립된 다각형은 방향성 변으로 이루어진 한 개의 순환 고리(loop)를 형성시킨다. MOSFET 소자와 다각형 변의 인접여부에 의해서 소자의 각 단자에 대한 절점 번호가 결정되고 소자의 외형비, 드레인과 소스 영역의 길이와 폭과 같은 MOSFET의 기하학적인 파라미터 값이 결정된다.

금속 접점에 대해서는 독립 절점 번호를 부여하고 이들 접점과 병합된 각 다각형의 인접한 변에 대한 절점 번호를 일치시켜 접점과 다각형의 연결성을 표현한다. 기하학적인 형태가 다양한 다각형에 대해서 알고리즘을 적용하여 다각형으로 표현된 레이어에 대한 능가의 RC모델을 추출한다. 그림 1의 회로 추출기는 시뮬레이션 환경과 공정 파라미터를 입력으로 읽어들이는 이들 추출되어진 결과와 레이아웃의 레이블로 표현된 입출력단자와 전원단자 등을 연결시키고 수행된 시뮬레이션 환경을 기술하게 했으며 공정 변화에 따른 모델 파라미터를 받아들여 공정변화에 대한 적응을 추출과정에 반영하게 했다. 일련의 과정에서 얻어지는 출력은 SPICE에서 받아들여지는 파일 형태이다. 레이블과 일치하는 이름을 사용하여 전원과 신호원 등에 대한 정보를 포함한 시뮬레이션 환경을 입력으로 받아들여 출력 파일을 생성하므로 이것을 바로 시뮬레이션할 수 있다. 또한 시뮬레이션 환경 기술에 포함된 출력 파일 형식에 대한 선택에 의해서 PSPICE, HSPICE, 또는 SPICE3에서 사용 가능한 형태로 출력 파일이 생성된다.

3. MOSFET 인식 및 단자 식별

3.1 입력 데이터 분류

회로 추출 순서는 MOS 트랜지스터를 먼저 추출하고 그리고 이 소자들을 연결하는 선의 길이와 넓이에 따라 분산된 저항과 커패시턴스의 값을 추출한다. 회로추출에 이용될 도형들은 Magic에서 형성된 타일들이다. 만약에 CIF의 정보를 읽어들이는 경우에는 이들의 도형 정보를 모두 타일로 바꾸어 사용된다. 타일은 모

두 직각으로 이루어진 Manhattan 도형[13]이며, Magic에서는 이러한 타일을 코너 스티칭(corner stitching) [15]시켜서 이웃한 모든 타일을 검색할 수 있게 한다. 그리고 이들 타일의 형성은 임의로 형성되는 것이 아니라 한 개의 타일의 추가되어질 때 형성된 위치에서 그 타일이 만드는 수평선에 의해 기존의 타일들을 잘라 놓기 때문에 한 개의 타일이 삽입될 때 여러 개의 타일이 형성되는 것이 일반적이다.

Magic의 레이아웃 기술 파일인 mag파일은 레이어의 종류와 타일의 좌표를 기억하고 있기 때문에 회로 추출의 예비 단계에서는 각각의 레이어의 종류별로 분류하여 저장한다. 동일한 종류의 레이어를 스캔라인 알고리즘을 이용하여 인접한 타일을 병합하면 여러 개의 블록이 형성된다. 생성된 블록을 구성 변의 다각형으로 기술할 수 있다. 이들 다각형은 방향성 변의 연결로 표현된다. 변의 진행 방향은 반시계 방향이고 진행 방향의 좌측은 다각형 내부로 정의한다. 이들 방향성 변과 회로 구성 소자 사이의 인접 여부를 조사하여 레이아웃에 대한 연결정보를 추출할 수 있는 알고리즘에 다각형을 기술하는 연결된 방향성 변을 사용한다. 입력 파일에 포함된 정보는 위에서 기술한 레이어를 나타내는 타일과 절점의 이름, 소자의 명칭, 또는 셀(cell)의 단자이름을 나타내는 레이블(label)로 구분되며 표 1과 같이 분류된다.

표 1에서 타일은 트랜지스터, 소자를 연결하는 선, 그리고 접점에 대한 레이아웃을 기술하는데 사용된다. 타일이 트랜지스터인 경우는 이것이 소자의 채널 영역을 나타내며 PMOS와 NMOS의 두 가지가 있다. 접점 타일은 인접한 레이어를 서로 연결하기 위해서 사용되며 이들 접점 저항 값이 크기 때문에 다중 접점으로 연결한다. 그래서 한 개의 접점 타일은 두 레이어 사이를 직각 연결하는 저항으로 주증된다.

3.2 MOS 소자의 추출

MOS 소자의 채널 형성은 확산영역과 폴리실리콘의 교차에 의해서 형성된다. 채널에 인접해 있는 두 개의 확산 영역은 MOS 소자의 드레인과 소스로 구분되며 그 면적과 주변길이는 MOS 소자의 파라미터로 추출된다. 채널은 게이트로 식별되고 두 개의 확산영역은 드레인과 소스단자가 된다. 그리고 이 채널을 포함하고 있는 웰(well)이 있다면 웰의 절점 번호를 그 소자의 기판 단자의 절점 번호로 부여하고 웰의 형(type)에 따라 NMOS인지 PMOS인지를 판단한다. 그리고 다른 트랜지스터에 대해서도 같은 방법으로 모든 단자에 대해서 독립된 절점 번호를 부여한다.

아날로그 회로의 경우에 소자의 외형비는 상당히 커지는 경우가 있게 되며, 이 경우 드레인과 소스 확산 영역의 시트 저항은 $1 \mu\text{m}$ CMOS 공정에서 약 20 Ω 정도로 무시할 수 없는 값이다[16]. 이런 경우에는 이들 단자들과 금속선을 연결하는 접점을 다중 접점(multi-contact)으로 만들어 생성되는 저항의 크기를 감소시킨다. 또한 외형비가 큰 MOS 소자의 구조는 칩 점유 면적을 줄이고 기생성분의 크기를 줄이기 위하여 드레인과 소스 단자를 공유시켜 놓은 스택 구조로 레이아웃 한다. 예를 들면 폭 W인 단일의 소자로 구성된 소자가 드레인과 소스에서 기판에 대해 가지는 커패시턴스가 각각 C_{cb} , C_{cb} 이라고 할 경우에 소자를 N개로 분할하여 스택 구조로 설계한 경우 이들의 값을 C'_{cb} , C'_{cb} 로 표현하면 식 (1)과 같은 커패시턴스 값을 가지게 된다. 단 N개의 소자가 가지는 폭의 합은 W와 같다.

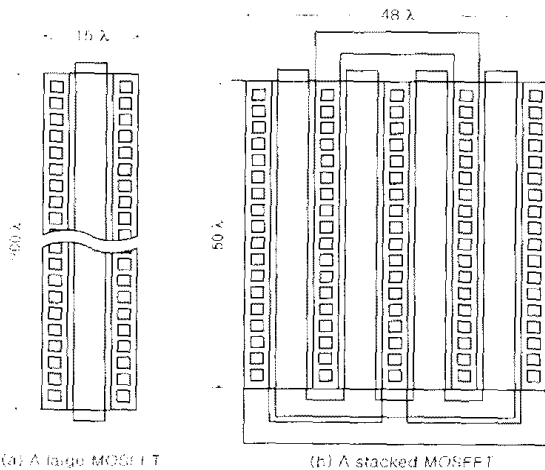
〈표 1〉 회로 추출에 사용되는 입력 데이터 분류
 〈Table 1〉 Input data classification used in circuit extraction

입력 데이터 분류			설 명
타일의 종류	채널 라우팅 접점	PMOS, NMOS Metal1,2, 확산영역, 폴리실리콘 Well, 확산영역, metal2, 폴리실리콘	폴리실리콘과 확산영역의 교차 영역 소자와 소자 연결 및 라우팅 영역 서로 다른 레이어 연결 영역
레이블	입출력 단자, MOSFET 소자의 명칭, 바이이스 진압 단자		소자의 명칭, 셀(cell)의 명칭, 전원 단자의 명칭, ...

$$C_{in} = \frac{N+1}{2N} C_{in}, C_{out} = \frac{N+1}{2N} C_{out}, \text{ for } N \text{ is odd} \quad (1.1)$$

$$C_{in} = \frac{1}{2} C_{in}, C_{out} = \frac{N+2}{2N} C_{out}, \text{ for } N \text{ is even.} \quad (1.2)$$

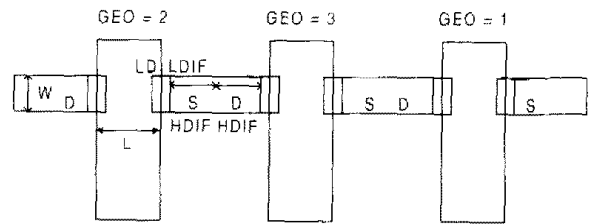
이러한 커패시턴스의 값을 감소시키는 것은 특히 고속을 요하는 회로의 경우에 더욱 중요한 문제이다. 일반적으로 레이아웃하기 전에 한 개의 소자에 대해 단일의 외형비를 갖는 소자로 고려하여 시뮬레이션을 하게된다. 그러나 레이아웃을 자동으로 생성하였거나 직접 설계했을 경우일지라도 레이아웃은 단지 주변에서 생성되는 이러한 커패시턴스를 최소로 하기 위한 스택 구조로 설계한다. 그러므로 실제 레이아웃 구조로부터 MOS 소자 추출이 필요하다. 본 회로 추출기에서 인식하는 스택구조의 소자를 그림 2에 표현하였다. 그림 2(a)에는 단일 소자인 경우이고 이것을 스택구조로 나타낸 것이 그림 2(b)이다



(그림 2) 외형비가 큰 MOS 소자의 스택구조로 레이아웃 (Fig. 2) Layout of a stacked MOSFET with large aspect ratio.

HSPICE에서 사용되어질 출력파일을 추출할 경우에 본 회로 추출기에서는 입력 파일 중 시뮬레이션을 위한 환경 설정 파일에서 기술되어진 ACM(Area Calculation Method) 모드[17]에 따라서 추출할 모델 파라미터의 값을 다르게 계산한다. ACM은 0에서 3까지 4개의 모드로 구분되는데 ACM=3인 경우는 그림 3에서 드레인과 소스의 공유를 고려하여 이들의 기하학적인 형태를 나타내는 GEO 값을 결정하게 된다. 드레인과 소스의 공유 형태에 따라서 GEO는 0에서 3까지의 값을 가질 수 있는데, 한 개의 소자가 공유가 없으면 0,

드레인만 공유되어 있다면 1, 소스만 공유되어 있다면 2, 그리고 드레인과 소스 모두 공유되어 있다면 3의 값을 가진다. ACM=3인 경우에 GEO의 값에 의해서 소자의 외형비, AS:AD, PS:PD 등의 값을 다르게 추출한다. 이것은 HSPICE에서 ACM과 GEO에 의해서 MOS 소자의 파라미터의 값을 다르게 계산하기 때문이다.



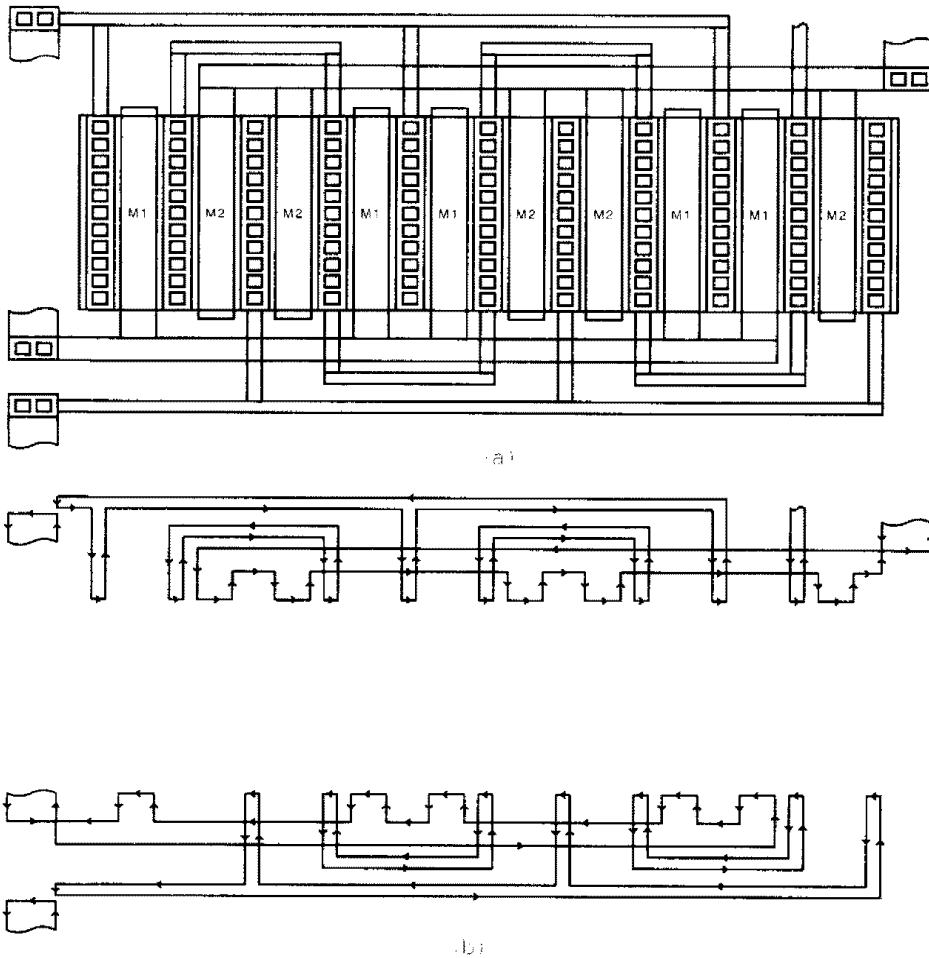
(그림 3) MOS 소자의 인식과 ACM=3에서의 GEO 값의 결정 (Fig. 3) MOSFET recognition and GEO value decision in ACM=3.

4. 방향성 변을 이용한 등가 RC 추출 알고리즘

4.1 소자의 연결관계 및 분산된 RC 추출

동일한 레이어의 모든 타일을 스캔라인 알고리즘으로 병합하여 여러 개의 다각형을 만든다. 한 개의 다각형은 인접해 있는 타일들이 병합되어 만들어지며, 다각형의 구성 변을 반시계 방향(CCW)으로 연결한 폐곡선으로 표현된다. 그래서 방향성 변의 진행 방향의 좌측은 다각형의 내부로 정의된다. 그림 4(a)는 소스 단자가 연결된 자동단의 M1과 M2에 대한 스택 구조의 레이아웃이다. 확산 영역은 MOSFET의 드레인과 소스 단자에 대한 기하학적인 파라미터 계산에 사용된다. 그림 4(b)는 MOSFET과 접점을 제외한 타일에 대하여 스캔라인 알고리즘을 적용하여 병합한 후의 다각형을 나타낸 것이다. 이렇게 만들어진 다각형의 변이 MOS 소자나 접점 타일 등과 맞닿은 변이 있는지 조사한다. 만약에 다각형 변에서 이들과 맞닿은 변이 있다면 그 변을 **terminal edge**로 정의하고 이를 **terminal edge**의 절점 번호는 인접해 있는 MOS 소자나 접점 타일의 절점 번호와 전기적인 연결관계가 있음을 표시한다. 여기서 변에 대한 데이터 구조는 다음과 같다.

```
typedef struct edge {
    int a0; // (a0 a1 b)는 변의 좌표가
    int a1; // 수평선이면 (a0 a1 b) (left right y)
    int b; // 수직선이면 (a0 a1 b) (bottom top x)
} Edge;
```



(그림 4) (a) 소스 단자가 함께 연결된 차동단에 대한 레이아웃.
 (b) (a)에서 트랜지스터와 접점을 제외한 타일을 병합한 후의 다각형
 (Fig. 4) (a) A layout for a differential source-coupled pair.
 (b) Polygons from merging tiles except those for MOSFETs and contacts in (a).

```

typedef struct edgelist {
    int direction; // 변의 방향 기수, left right up down
    int node_number; // 변의 절점 번호
    Edge coordinate; // 변의 좌표, (x0 xl y) 또는 (y0 yl x)
    edgelist *next; // 변의 CCW에 위치한 변을 가리키는 포인터
    edgelist *prev; // 변의 CW에 위치한 변을 가리키는 포인터
}
    
```

1. Edgelist:

다각형을 구성하는 일련의 변들은 포인터 **next**와 **prev**에 의한 양방향 연결리스트(bidirectional linked list) 구조로 표현된다. 현재의 변과 CCW에서 연결된 변을 포인터 **next**가, CW에서 연결된 변을 포인터 **prev**가 가리키고 있다. 이 데이터 구조는: 다각형의 변을 표현할 때 한 변은 **next**와 **prev**에 의해서 인접한

두 변을 모두 참조하도록 하였기 때문에 어느 방향으로 진행하더라도 특정한 변을 찾을 수 있다. 이러한 데이터 구조로 기술된 다각형으로부터 기생 RC를 추출하기 위하여 분기 가지를 생성하고, 가지를 진행하는 재귀적 함수에 대한 알고리즘 기술 과정은 그림 5와 같다. 그림 5의 가상 코드에 부연하여 알고리즘 기술을 다음과 같이 설명할 수 있다.

1. 각 다각형을 기술하는 순환하는 edgelist에서 최초의 **terminal edge**를 찾아낸다.

2. 이 **terminal edge**의 **prev**와 **next** 변을 결정하고, **terminal edge**를 기준 절점으로 정한다. 그리고 **terminal edge**의 **prev**와 **next**를 매개변수로 recursiveCall() 함수를 호출하여 다각형에 대한 기생 RC 추출을 시작한다.

```

void recursiveCall( Edgelist *next, Edgelist *prev )
{
    if a new branch created in only prev side
    save1 = findCounterpartEdge( prev->prev );
    if save1 == next->next
        recursiveCall( save1, prev->prev );
    else
        recursiveCall( save1, prev->prev ); // 1st branch
        recursiveCall( next, save1->prev ); // 2nd branch
    else if a new branch created in only next side
    save2 = findCounterpartEdge( next->next );
    if save2 == prev->prev
        recursiveCall( next->next, save2 );
    else
        recursiveCall( next->next, save2 ); // 1st branch
        recursiveCall( save2->next, prev ); // 2nd branch
    else if branches are created in both prev and next
    save1 = findCounterpartEdge( prev->prev );
    save2 = findCounterpartEdge( next->next );
    recursiveCall( save1, prev->prev ); // 1st branch
    recursiveCall( next->next, save2 ); // 2nd branch
    if save1 != save2 && save1->prev != save2->next
        recursiveCall( save2->next, save1->prev ); // 3rd branch
}
    
```

(그림 5) 가지의 진행
(Fig. 5) Progresses of branches.

④ next 변의 방향으로 진행하면서 기준 절점에서 다음 절점까지 분산 RC를 계산한다. 진행 중에 terminal edge를 만나거나 가지가 형성된 곳을 만나면 다음 절점으로 설정된다. 기준 절점에서 다음 절점까지 가상 절점의 수가 $i (\geq 1)$ 개가 있었다면 식 (2)와 같이 두 절점 사이의 분산 RC가 계산된다.

$$R_{lumped} = R_{sh} \times \sum_i \text{Length}(i) / \text{Width}(i) \quad (2.1)$$

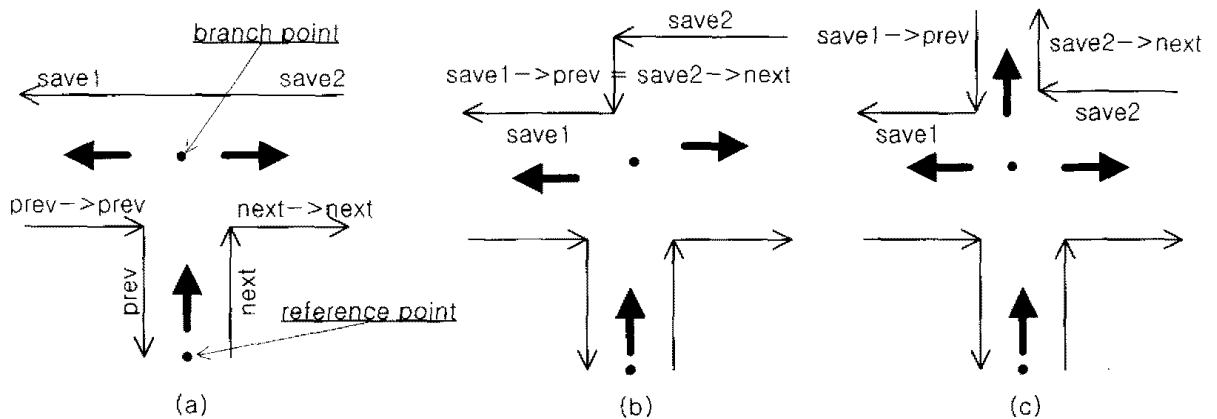
$$C_{lumped} = C_{sh} \times \sum_i \text{Length}(i) \times \text{Width}(i) \quad (2.2)$$

여기서 R_{sh} 와 C_{sh} 는 입력파일인 공정 파라미터에 기술된 값이며, 현재 추출하고 있는 레이어의 시트 저항

과 관계된 값이다. 가상 절점이란 변의 진행 과정에서 진행 방향은 바뀌었으나 새로운 값이 생성되지 않았을 경우에 생성되는 절점을 말하며 유효한 절점이 아니다. 가상 절점에서는 변의 진행방향이 180도의 방향으로 바뀌므로 직선 구간의 시트 저항 값보다 작은 값을 가진다.

⑤ 진행 중에서 가지가 생성되는지를 조사한다. prev->prev가 next와, next->next가 prev와 떨어져 있을 때, 떨어져 있는 변의 진행 방향으로 새로운 가지가 생성된다. 여기서 가상 절점이 생기거나 두 가지로 진행되거나 또는 3개의 진행 가지가 형성될 수 있다. 가상 절점은 prev->prev의 대응 변이 next->next이거나 next->next의 대응 변이 prev->prev이면 진행 방향이 굽어지는 그 위치가 가상 절점이다. prev->prev의 대응변이 next->next가 아니면 진행 방향의 왼쪽으로 가지가 형성된다. 그리고 next->next의 대응변이 prev->prev가 아니면 진행 방향의 오른쪽 반향에서 가지가 형성된다. 3가지가 형성되는 경우는 진행 방향의 좌우 모두에 가지가 형성되었을 경우 next->next의 대응변과 prev->prev의 대응변이 동일한 변이 아니고 이 두 대응변이 또 다른 한 개의 변에 의해서 직접 연결될 경우가 아니면 3번째 가지가 형성된다.

⑥ 진행과정에서 가지가 형성되면 다음 진행될 가지의 수를 확인하고, 기준 절점에서 가지 점까지 분산 RC를 출력한다. 그리고 이 가지의 분기점은 다음 진행의 기준 절점이 되고 가지가 형성된 경우에 따라 재귀적으로 생성된 가지의 방향으로 함수 recursiveCall()이



(그림 6) next와 prev에서 모두 가지가 생성
(Fig. 6) Branching in both next and prev.

호출된다. Manhattan 모양만 취급하기 때문에 생성 가능한 가짓수는 3개 이하이며 가지의 형성 기준과 각 가지의 진행 과정은 그림 5와 같이 기술되어진다.

한 만약에 진행 과정에서 **terminal edge**를 만나면 기준 점점에서 **terminal edge**까지의 RC 값을 출력하고 진행을 멈춘다. 더 이상 **terminal edge**가 없으면 분산 RC의 추출 과정이 마무리된 것이다.

그림 6은 **prev**와 **next**에서 모두 가지가 형성되는 경우의 예이다. **prev >prev**의 대응 변 **save1**과 **next >next**의 대응 변 **save2**를 찾으면, 그림 6(a)는 **save1 - save2**, 그림 6(b)는 **save1->prev = save2 >next** 인 경우이므로 두 개의 가지가 형성되고, 그림 6(c)는 **save1 ≠ save2**이고 **save1->prev ≠ save2 >next**이므로 **(save2->next, save1 >prev)**의 한 쌍의 변이 recursiveCall() 함수를 호출하므로 세 가지가 형성된 경우이다.

4.2 점점저항의 등가 모델

인접해 있는 두 레이어를 연결하는 점점은 일급속의 시트 저항 값보다도 상당히 큰 값을 가진다. 이들의 값을 연결 정보에 포함하여야 레이아웃에 대한 보다 정확한 RC를 추출할 수 있다. 그림 2의 MOSFET 스택구조에 있어서 확산 영역의 저항 값을 감소시키기 위하여 다중 점점으로 연결하고 있고, 또한 인접한 다른 레이어와의 연결에서도 그림 7과 같이 다중 점점으로 이들을 연결하는 것이 일반적이다. 다중 점점인 경우의 점점 저항을 다중 점점의 수가 m개 일 경우에 식 (3)과 같이 근사화할 수 있다.

$$R_{cm} \approx \frac{R_c}{m} \tag{3}$$

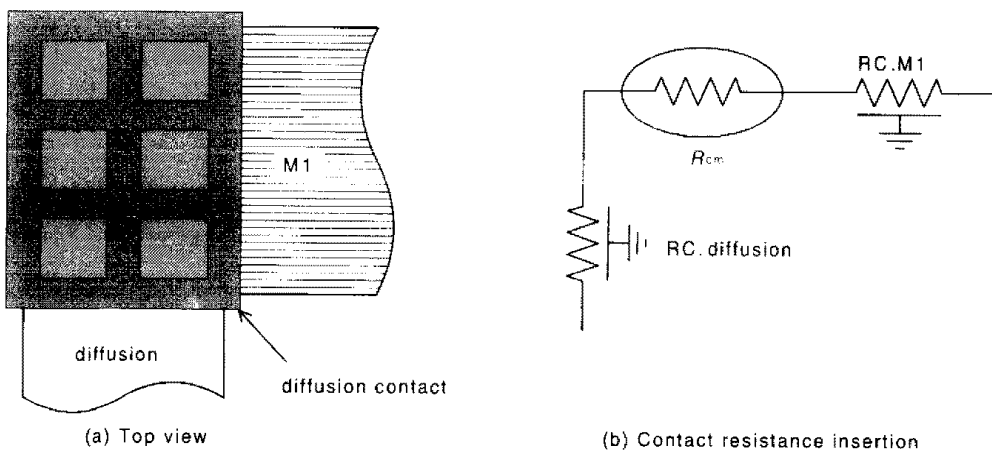
여기서 R_{cm} 은 두 레이어 사이의 점점에 의해서 형성되는 직렬 저항이고 R_c 는 공정 파라미터에 의해서 주어지는 점점 저항이다. 그림 7(a)에서 다중 점점의 수가 6개이고 M1과 확산영역의 점점 저항 $R_c = 10 \Omega$ 이라면, 그림 7(b)와 같이 M1과 확산영역 사이에 $10/6 \Omega$ 의 저항이 직렬로 연결되어 있는 등가 모델로 근사화할 수 있다. 그림 7(b)에서 RC.M1과 RC.diffusion은 Metal과 확산영역에 대한 분산 RC모델을 나타낸다.

4.3 점점 번호 할당 및 전기적 연결 설정

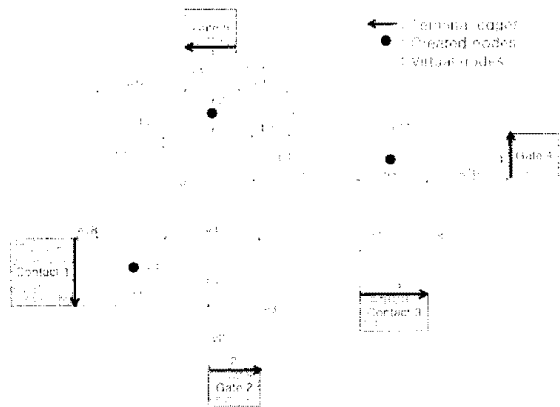
트랜지스터의 각 단자에 대한 점점 번호는 MOS소자 인식과정에서 스택구조의 트랜지스터를 추출하면서 할당된다. 또한 연결 선에 대한 등가 RC를 추출하기 전에 레이아웃의 모든 점점에 대한 점점 번호를 유일하게 할당한다. 점점의 점점 번호는 점점이 다른 두 레이어 사이를 연결하기 때문에 두 레이어 사이를 잇는 직렬 연결 저항의 양단의 점점 번호를 연결하는 목적으로 사용된다. 연결선에 대한 점점 번호는 연결선의 기생 RC를 추출하는 과정에서 추출될 다각형의 변이 트랜지스터의 단자 또는 점점과 인접하게 되면 이들에 대해 할당된 유일한 점점 번호를 참조하여 이들 점점과 전기적 연결관계를 설정한다.

4.4 적용 예

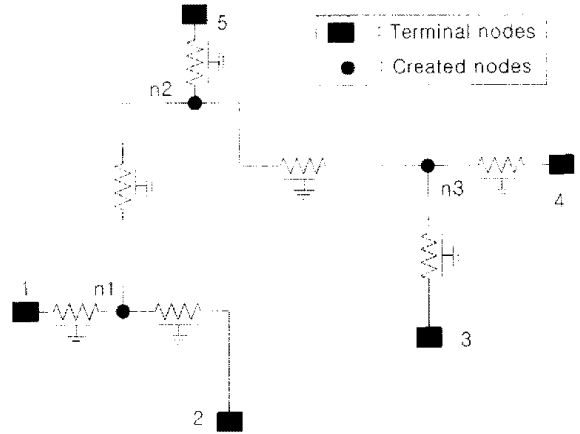
위에서 기술한 알고리즘을 그림 8의 다각형 블록에



(그림 7) 다중 점점 저항의 등가 모델
(Fig. 7) Equivalent model for multi-contact resistance.



(그림 8) Scanline 알고리즘에 의해 병합된 다각형과 인접한 MOS 소자와 접점에 의해 terminal edge로 표현된 변 (Fig. 8) A merged polygon by scanline algorithm and edges marked as terminal edge adjacent to MOSFETs or contacts.

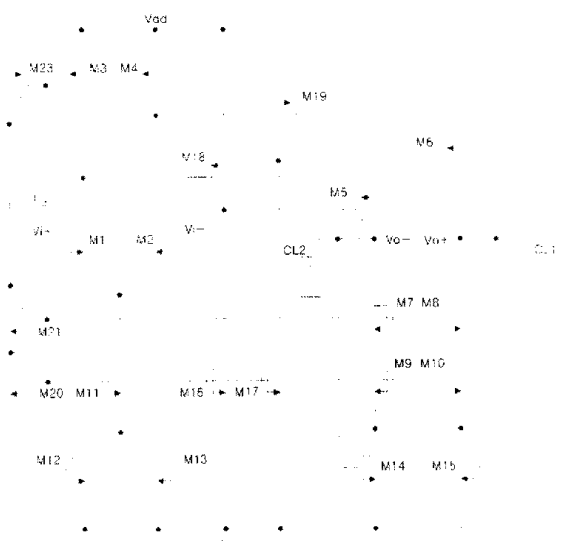


(그림 9) 그림 8의 레이아웃과 등가인 분산된 RC 모델 (Fig. 9) Distributed RC model equivalent of layout in Fig. 8.

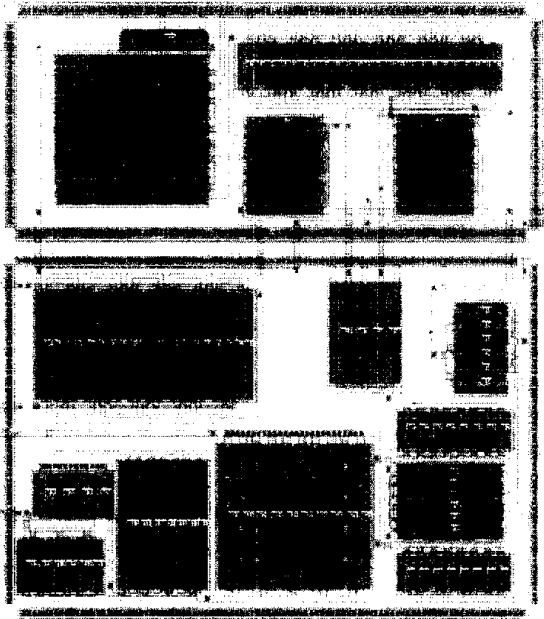
적용할 수 있다. 먼저 다각형을 구성하는 변에서 terminal edge를 식별해야한다. 접점과 연결된 변 1과 3 그리고 게이트와 연결된 변 2, 4, 5는 모두 terminal edge로 표현된다. 그리고 나머지 변, e1~e18은 terminal edge를 잇는 변으로 다각형을 구성하는 변이다. 다각형을 기술하는 이들 변에 의해 연결된 변의 데이터로부터 최초의 terminal edge 1이 선택될 경우, 이 위치를 기준절점 1로 정하고 이 절점에서 변 e1의 진행방향으로 recursiveCall(e1, e18)이 호출되어 분산 RC 추출을 시작한다. 변 e17이 e1과 거리가 멀어지고 있으므로 진행 방향에서 왼쪽으로 가지가 형성된다. 진행과정에서 가지가 형성되었으므로 절점 n1을 설정하고 기준 절점 1에서 절점 n1까지의 분산 RC가 계산된다. 그리고 절점 n1을 새롭게 생성된 두 개의 가지에 대한 기준 절점으로 설정하고 (e1, e4)와 (e5, e17)을 매개 변수로 하여 재귀적 함수 recursiveCall()을 각각 호출한다. 두 가지 중의 한 가지 (e1, e4)의 진행에서는 가상 절점 b1이 있고 (e2, e3)로 진행하여 terminal edge 2를 만났으므로 진행을 멈춘다. 식 (2)에 의해 절점 n1에서 terminal edge 2까지의 분산 RC가 계산된다. 그리고 다음 진행 (e5, e17)에 의해 terminal edge 2, 3, 4 절점까지 진행을 끝내면 그림 8의 다각형에 대한 RC 추출이 끝난다. 이러한 과정을 통하여 추출된 그림 8의 다각형에 대한 분산 RC 모델을 그림 9와 같이 표현할 수 있다. 이 모델을 π 3과 같은 직당한 등가의 시뮬레이션 모델[18]로 바꾸어 출력한다.

5. 회로 추출과 시뮬레이션 결과

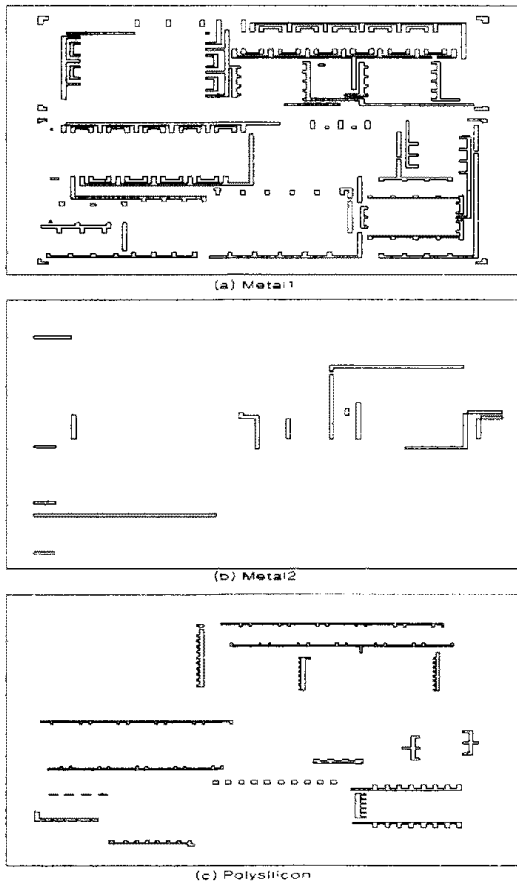
회로 추출기는 리눅스 운영체제의 개인용 컴퓨터 팬티엄 150 MHz에서 C 언어로 구현하였으며 컴파일러는 gcc 2.7.0을 사용하였다. 그림 10은 현대 0.8 μ m CMOS 설계 공정[19]의 HSPICE 레벨 13 BSIM1[20]을 사용하여 SAPICE[21]로 설계한 조정된 캐스코드를 갖는 폴디드 캐스코드 CMOS 연산 증폭기이다. 이 증폭기를 레이아웃과 회로 추출의 예제로 선택하였으며, 그림 11은 그림 10의 회로에 대한 레이아웃으로 MOS



(그림 10) 조정된 캐스코드를 가지는 폴디드 캐스코드 CMOS 연산증폭기 (Fig. 10) A folded-cascode CMOS operational amplifier with regulated cascodes.



(그림 11) 폴디드 캐스코드 CMOS 연산 증폭기의 레이아웃
(Fig. 11) Layout for a folded-cascode CMOS operational amplifier.



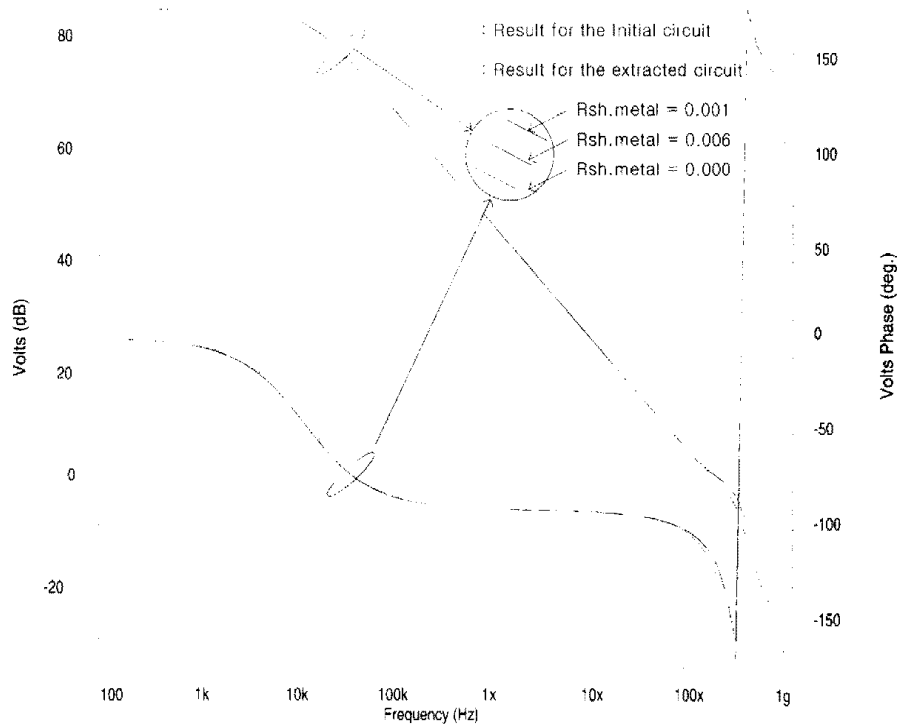
(그림 12) 분산된 RC를 추출하기 위하여 병합된 블록
(Fig. 12) Blocks merged to extract distributed RCs.

소자를 스택 구조로 설계하였다. 레이아웃에서 소자간의 연결과 외부 신호 단자 등의 연결선은 폴리실리온과 제 1금속과 제 2금속을 사용하였다. 그림 12는 스텝라인 알고리즘을 이용하여 폴리실리온, 제 1금속, 그리고 제 2금속 등의 각각의 타일들을 병합하여 생성된 블록이다. 생성된 블록은 제 1금속이 85개, 제 2금속이 16개, 그리고 폴리실리온이 30개가 생성되었다.

레이아웃으로부터 추출한 소자들의 개수를 초기 회로와 비교하기 위하여 표 2에 나타내었다. 그림 11의 레이아웃으로부터 MOS 소자와 이들의 기하학적인 공정 파라미터 그리고 분산 RC의 값과 이들의 연결관계를 포함하는 SPICE 파일을 추출하는데 CPU 경과 시간은 약 1초가 소요되었다. 추출된 MOSFET과 생성된 블록에 대한 등가의 분산 RC를 π 3 시뮬레이션 모델로 출력한 회로를 HSPICE로 시뮬레이션한 주파수 특성을 그림 13에 나타내었다. 시뮬레이션 환경은 HP SWS-J210 CPU 120 MHz의 워크스테이션에서 HSPICE를 사용하여 수행하였다. 그림 13의 결과로부터 회로 추출기에 의해 추출된 netlist는 초기 회로에 대한 레이아웃이 정확함을 확인할 수 있었고 직류 이득과 위상 여유 특성 등이 초기 회로의 것과 거의 일치한다. HSPICE로부터 생성된 초기 회로에 대한 시뮬레이션 결과와 추출된 회로의 시뮬레이션 결과를 비교하여 표 3에 나타내었다. 표 2와 표 3의 결과를 참조해 볼 때 분산 RC 모델을 어떠한 등가 모델로 변환하는가에 의해서 생성되는 절점, 소자의 개수, 그리고 시뮬레이션 실행시간이 모두 상당히 커짐을 볼 수 있었다.

<표 2> 그림 11의 레이아웃으로부터 추출한 결과
<Table 2> Elements extracted from layout in Fig. 11

소자의 개수	초기 회로	추출된 회로		
		lumped	π 3	
PMOS	7	39		
NMOS	15	79		
분산 RC		metal1 : 251	metal2 : 16	polysilicon : 232
전체 소자	31	1143	3635	
절점(node)	65	777	1771	



(그림 13) 초기 회로와 레이아웃으로부터 추출된 회로의 시뮬레이션 결과 비교
 (Fig. 13) Comparison of the initial circuit with the extracted circuit by simulation.

<표 3> 초기 회로와 추출한 회로의 시뮬레이션 결과
 <Table 3> Results simulated for the initial circuit and the extracted circuit

성능평가	초기회로	$R_{sh} \leq 0.006$
위상 여유 [degree]	48	41 ~ 43
단위 이득 주파수 [MHz]	215	220
DC 이득 [dB]	83.7	83.9

학적인 모델 파라미터를 추출하고, 이들을 연결하는 도선들의 분산된 RC를 추출할 수 있다. 추출된 회로를 시뮬레이션을 통하여 설계된 레이아웃이 정확한지를 검증할 수 있고 레이아웃에 내재하는 기생성분의 영향이 회로의 성능에 미치는 영향을 평가하고 그 회로가 설계 사양을 만족하는지를 확인할 수 있었다. 제안된 알고리즘은 단일의 셀 뿐만 아니라 전체 칩 단위의 레이아웃에 적용가능하기 때문에 여러 개의 셀이 포함된 회로에 적용하여 그 성능을 검증할 수 있다.

6. 결 론

직접 회로 전체의 레이아웃에 대한 MOSFET 소자와 이들을 연결하는 레이어들의 기생저항과 커패시턴스를 추출하여 시뮬레이션 가능한 파일을 출력하는 CAD 도구를 만들었다. 기존 회로 추출기는 일반적으로 연결 성분들에 대한 레이어를 사각형의 타일로 분할하여 이들의 인접 사각형과의 접촉형태에 따라 능가의 RC 성분을 추출하고 있으나 본 논문에서는 다각형을 나타내는 방향성 변을 이용한 회로 추출 알고리즘을 제안하였다. 레이아웃으로부터 소자와 소자의 기하

참 고 문 헌

[1] U. Choudhury, A. Sangiovanni-Vincentelli, "Automatic generation of analytical models for interconnect capacitances," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol.14, No.4, pp.470-480, April 1995.
 [2] C. Toumazou and C.A. Makris, "Analog IC design automation: Part I - Automated circuit generation: New concepts and methods," *IEEE Trans. on CAD of Integrated Circuits and Sys-*

tems Vol.14, No.2, pp.218-238, Feb. 1995.

[3] C. A. Makris and C. Toumazou, "Analog IC design automation: Part II- Automated circuit correction by qualitative reasoning," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol.14, No.2, pp.239-254, Feb. 1995.

[4] Enrico Malavasi and Davide Pandini, "Optimum CMOS stack generation with analog constraints," *IEEE Trans. on CAD of Integrated Circuits and Systems* Vol.14, No.1, pp.107-122, Jan. 1995.

[5] Young Seok Baek, Byoung Yoon Cheon, Kyung Sik Kim, Hyun Chan Lee and Chul Dong Lee, "Cell designer: An automatic placement and routing tool for the mixed design of macro and standard cells," *IEICE Trans. Fundamentals*, Vol.E75 A, No.2, pp.224-232, Feb. 1992.

[6] Yeong Chan Son, Seok Hong Park and Sang-Dae Yu, "New approach to extract netlist from layout," *Proc. of 5th International Conference on VLSI and CAD*, pp.421-423, 1997.

[7] 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.

[8] J. D. Bastian, M. Ellement, P. J. Fowler, C. E. Huang and L. P. McNamee, "Symbolic Parasitic Extractor for Circuit Simulation (SPECS)," *Proc. of 20th Design Automation Conference*, pp.346-352, 1983.

[9] 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.

[10] W. S. Scout and J. K. Ousterhout, "Magic's circuit extractor," *IEEE DESIGN & TEST*, 1986.

[11] P. Chapman and K Clark, "The scan line approach to design rules checking : Computational expressions," *Proc. of 21st Design Automation Conference*, pp.235-241, 1984.

[12] Erik C. Carlson, "A scanline sata structure processor for VLSI geometry checking," *IEEE Trans. on CAD*, Vol.6, No.5, 1987.

[13] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, pp.115-127, 1980.

[14] Steven M. Rubin, *Computer Aids for VLSI Design*, Addison Wesley, pp.2-111, 1989.

[15] J. K Ousterhout, "Corner Stitching: A data structuring technique for VLSI layout tools," *IEEE Trans. on CAD*, 3:1, pp.87-100, 1984.

[16] Mohammed Ismail and Terri Fiez, *Analog VLSI: Signal and Information Processing*, McGraw-Hill, pp.699-707, 1994.

[17] *HSPICE User's Manual* Vol.2, Elements and Models, HSPICE Version H92, Meta-Software, 1992.

[18] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*, Prentice-Hall, pp.471-476, 1996.

[19] Hyundai Electronics Industries, *0.8 μm Model Parameter*, April 1994.

[20] B. J. Sheu, D. L. Scharfetter, P.K. Ko and M.C. Jeng, "BSIM: Berkeley short channel IGFET model for MOS transistors," *IEEE J. Solid-State Circuits*, Vol.SC 22, No.4, pp.558-566, 1987.

[21] Sang-Dae Yu and Chong-Min Kyung, "SAPICE: A design tool of CMOS operational amplifiers," *IEICE Trans. Fundamentals*, Vol.E80-A, No.9, pp.1667-1675, Sep. 1997.



손영찬

e mail : yeson@pohang.ac.kr

1991년 경북대학교 전자공학과(공학사)

1994년 경북대학교 대학원 전자공학과(공학석사)

1996년 경북대학교 대학원 전자공학과(박사과정 수료)

1998년-현재 포항 1 대학 컴퓨터응용과 전임강사

관심분야 : 집적회로 설계검증, 설계자동화



박 석 흥

e-mail : shpark@palgong.kyungpook.ac.kr
1992년 경북대학교 전자공학과(공학사)
1994년 경북대학교 대학원 전자공학과(공학석사)
1996년 경북대학교 대학원 전자공학과(박사과정 수료)

1997년~현재 (주)한국멘토
관심분야 : CAD/CAM, 시뮬레이션, 시스템 통합



유 상 대

e-mail : sdyu@bh.kyungpook.ac.kr
1980년 경북대학교 전자공학과(공학사)
1982년 한국과학기술원 전기 및 전자공학과(공학석사)
1998년 한국과학기술원 전기 및 전자공학과(공학박사)

1983년~1986년 경북대학교 전자공학과 전임강사
1986년~1991년 경북대학교 전자공학과 조교수
1991년~1997년 경북대학교 전자공학과 부교수
1997년~현재 경북대학교 전자전기공학부 교수
관심분야 : 집적시스템, 설계자동화, 표면음파필터설계