PC-DRC : PC를 이용한 집적회로 layout 설계 규칙 검사

○ 박 인철, 어 길수, 경 종민

한국 과학 기술원 전기 및 전자공학과

PC-DRC : Design Rule Check for Integrated Circuit Using PC

In Cheol Park, Kil Soo Eo, Chong Min Kyung

Dept. of Electrical Engineering, KAIST

ABSTRACT

This paper describes a new design rule checking system, PC-DRC, for CIF mask layout, which was written in C language on IBM PC/AT under DOS 3.0 environment. H/W devices and S/W utilities for PC-DRC is identical to that for PC-LADY[6], which makes PC-DRC an ideal post-processing routine for CIF file generated by PC-LADY.

Various spurious errors were eliminated by ORing the input CIF data for each layer and the design rule errors were checked by edge based method on rectilinear polygon form. The detected errors are stored in CIF and displayed on CRT simultaneously.

I. 서론

최근 집적회로의 제조기술과 설계기술이 발달하여 거의 자동화 단계에 들어서고 있지만, 설계 특히 layout 의 경우에는 많은 수동적인 조작이 필요하다. 따라서 오류가 없는 Layout을 편집하기는 매우 어려운 일이므로 layout의 오류를 검출할 수 있는 시스템이 요구된다.

PC-DRC(PC Design Rule Checker)는 한국과학기술원 전기 및 전자공학과 CAD 연구실에서 설계 자동화 시스템 개발의 일환으로 만들어진 집적회로 layout 검증 시스템이다.

이 시스템은 크게 RTP 와 DRC 라는 두개의 program으로 이루어져 있으며 CIF data 를 읽어들여 화면에 나타내고 검출된 오류를 화면에 표시하는 기능을 가진다. 또한 검출된 오류를 파일에 저장하여 사용자가 이 파일을 이용할 수 있도록 하였다. 이 시스템은 IBM PC/AT 에서 개발되었으며 직각의 CIF 데이타에 대하여 수행할 수 있다.

II. 시스템의 구성

II.1. Hardware 의 구성

PC-DRC 는 ORing된 집적회로 layout과 검출된 오류를 화면에 나타내며 각 layer 는 color 를 사용하여 구별하였다. 이러한 computer graphics를 위해서 EGA (Enhanced Graphics Adaptor) board 와 MS-mouse를 이용한다. 시스템의 구성은 그림 1.과 같다.

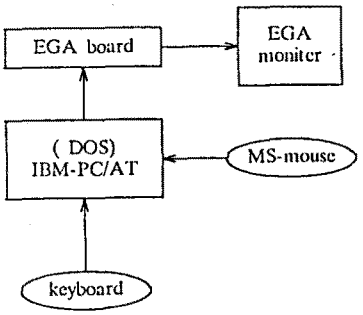


그림 1. 시스템의 구성

II.2. Software 구성

PC-DRC 는 Lattice-C 언어로 프로그래밍 되어졌으며 IBM PC/AT, DOS 3.0 상에서 동작한다.

II.3. 프로그램의 구성

PC-DRC는 RTP 와 DRC라는 두개의 프로그램으로 구성되어 있는데 RTP는 CIF data 를 읽어들여 기본 도형 (Box, Polygon, Wire)을 각 layer에 대하여 ORing 함으로서 merge된 도형의 외곽에 해당하는 Edge들을 출력하는 프로그램으로 spurious 오류를 제거하는 역할을 한다.

DRC는 RTP에 의해 생성된 Edge를 이용하여 설계검증을 하는 프로그램이다. 이 두 프로그램의 관계는 그림 2.와 같다.

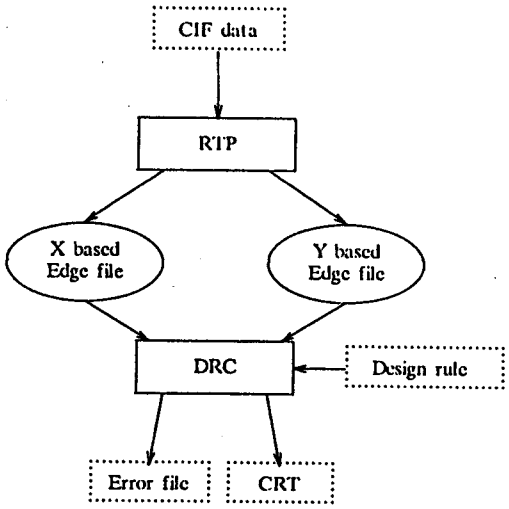


그림 2. 프로그램의 구성

III. RTP program

RTP는 CIF data를 각 layer별로 ORing을 한다. ORing algorithm은 다음과 같다.

정수 배열 A[], B[] 가 있고 0으로 초기화되어 있다고 하자.

1) 각 도형을 시계방향의 polygon으로 변환한다.

2) 이 polygon들을 Edge로 분리하는데 각 Edge에는 그림 3.과 같은 방향성을 부여한다.

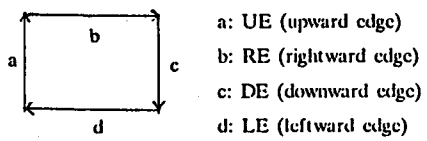


그림 3. Edge의 방향

3) UE 와 DE는 Edge의 두 vertex의 X좌표가 같으므로 한 그룹으로 하여 X좌표를 Keyword로 sorting한다.

4) sorting된 Edge들 중에서 가장 작은 X좌표를 가지는 Edge들을 뽑는다. 골라낸 각 Edge의 vertex의 Y좌표가 y1, y2 라면 그 Edge가 UE일 경우는 A[y1] 부터 A[y2-1] 까지 1을 더하고 DE일 경우에는 1을 뺀다.

5) B[] 와A[]을 비교하여 0에서 다른 수로 변환된 부분은 merging 된 도형의 upward edge에 해당하고 0 아닌 수에서 0으로 변환된 부분은 downward edge에 해당하므로 출력파일에 저장한다.

6) A[] 를 B[]에 copy 한다.

7) 나머지 edge에 대하여 4) 5) 6)을 반복한다.

RE 와 LE에 대하여도 똑같은 algorithm을 적용하면 된다.

IV. DRC program

집적회로 layout에 나타나는 오류는 다음 4가지로 구별할 수 있다.

1. Width rule error
2. Space rule error
3. Margin rule error
4. Enclosure rule error

이러한 오류를 PC 에서 검증하기 위하여 Edge based 설계검증을 하였으며, 큰 layout의 경우에는 layout을 분할하여 각 분할된 영역에 대하여 검증을 하도록 하였다.

IV.1. 데이타 구조

빠른 설계검증을 위하여 scan-line 방법과 work list 방법을 사용하였으며, 이를 위하여 그림 4.와 같은 데이타 구조를 만들었다. 이 데이타 구조로 insert 와 delete 동작을 하지않고도 work list 방법을 이용할 수 있었다.

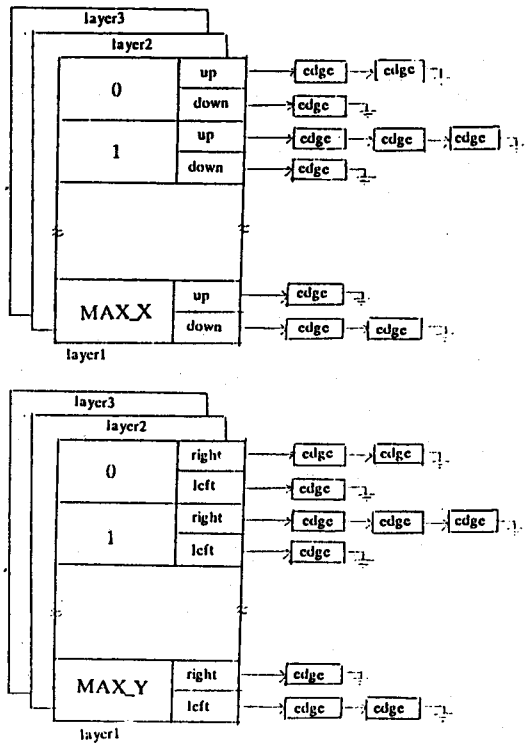


그림 4. 데이타 구조

IV.2. Width rule 검사

Width rule은 도형이 가져야할 최소한의 폭이며, 아래의 procedure를 이용하였다.

```
procedure Width_X (i:layername; rule:rule_value)
  var
    s,r: integer;
    UE,DE: edge;
  begin
    for s := 0 to MAX_X do begin
      DE := X[i,s].down;
      while ( DE is not NULL ) begin
        for r := s-rule+1 to s do begin
          UE := X[i,r].up;
          while(UE is not NULL) begin
            if( UE and DE are overlapped ) then error report;
            else if ( vertex distance < rule) then error report;
            UE := UE->next
          end while
        end for
        DE := DE->next
      end while
    end for
  end.
```

IV.3. Space rule 검사

Space rule은 두 도형이 가져야할 최소한의 거리이다.

```
procedure Space_X (i,j:layername; rule:rule_value)
  var
    s,r: integer;
    UE,DE: edge;
  begin
    for s := 0 to MAX_X do begin
      UE := X[i,s].up;
      while ( UE is not NULL ) begin
        for r := s-rule+1 to s do begin
          DE := X[j,r].down;
          while(DE is not NULL) begin
            if( UE and DE are overlapped ) then error report;
            else if ( vertex distance < rule) then error report;
            DE := DE->next
          end while
        end for
        UE := UE->next
      end while
      UE := X[j,s].up;
      the same code as above except i,j layer index;
    end for
  end.
```

IV.4. Margin rule 검사

Margin rule은 서로 다른 layer에 있는 두 도형이 중첩되어 있는 경우에 alignment 오차를 고려하여 충분히 중첩되어야 할 정도이다. 이 검사에는 교차 검사를 하는 procedure를 썼다.

```
procedure Margin_cross_UP(UE:edge;j:layername;rule:rule_value)
  var
    umin,umax,s:integer;
    RE,LE; edge;
  begin
    umin := UE.min;
    umax := UE.max;
    for s := umax to umax-rule+1 do begin
      RE := Y[j,s].right;
      while (RE is not NULL) begin
        if( RE cross UE ) then error report;
        RE := RE->next
      end while
    end for
    for s := umin to umin+rule-1 do begin
      LE := Y[j,s].left;
      while (LE is not NULL) begin
        if(LE cross UE ) then error report;
        LE := LE->next
      end while
    end for
  end.
```

IV.5. Enclosure rule 검사

Enclosure rule은 한 도형이 다른 도형에 완전히 포함되는 경우에 적용된다. 이 검사에는 거리 검사와 교차 검사를 하는 procedure를 썼다.

```
procedure Enclo_cross_UP(UE:edge;j:layername;rule:rule_value)
  var
    umin,umax,s:integer;
    RE,LE; edge;
  begin
    umin := UE.min;
    umax := UE.max;
    for s := umin to umax do begin
      RE := Y[j,s].right;
      while (RE is not NULL) begin
        if( RE cross UE ) then error report;
        RE := RE->next
      end while
      LE := Y[j,s].left;
      while (LE is not NULL) begin
        if(LE cross UE ) then error report;
        LE := LE->next
      end while
    end for
  end.
```

IV.6. Layout 의 분할

Layout이 큰 경우에는 Page 별로 나누어 검증을 하는데 한 Page는 130 X 75 lambda이다.

IV.7. Utility

사용자의 편의를 위하여 거리 측정, 점의 좌표값, 원하는 Page로 이동하여 검증을 할 수 있는 기능이 제공되고 있다.

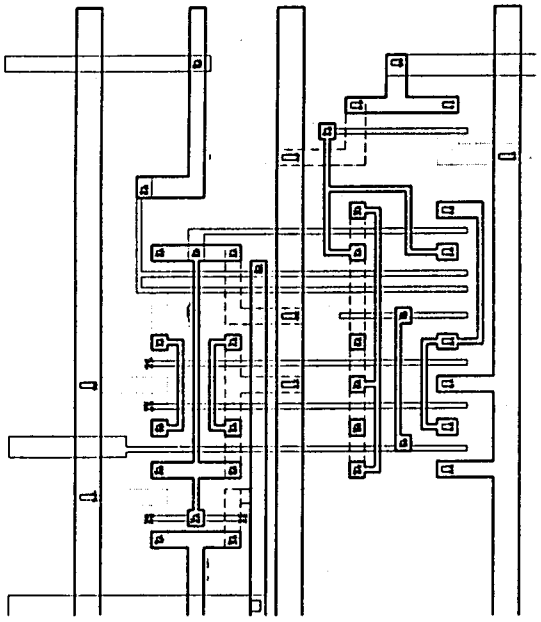
IV.8. 출력

이 시스템의 출력은 화면과 파일 두가지이며, 화면에 오류를 나타낼 때에는 위의 4가지 오류의 표시 색깔을 달리하여 구별할 수 있도록 하였고, 파일에는 오류를 CIF 데이타 형태로 기술하여 PC-LADY에서도 사용 가능하도록 하였다.

V. 결론

IBM PC/AT 상에서 수행되는 집적회로 layout 검증 시스템을 개발함으로서 손쉽게 접할 수 있고 사용할 수 있게 되었다. 특히 이 시스템은 Edge based 검증 방법을 사용하여 빠른 검증을 할 수 있게 하였으며 Boolean operation을 사용하지 않고 검증을 하는 algorithm을 썼다.

이 시스템은 PC-LADY의 보조 시스템으로 사용할 수 있어서, 완전히 PC 상에서 설계하고 오류를 검증할 수 있는 시스템을 구축하였다는 데 그 의의가 있다고 본다. 현재는 직각의 CIF 데이타에 대해서만 오류 검증을 할 수 있으므로 비직각적인 데이타에 대하여 오류를 검증할 수 있도록 연구가 진행되어야 할 것이다.



(Example)

REFERENCES

[1] C.Mead and L.Conway, "Introduction to VLSI systems", Addison-wesley, 1980.

[2] "LATTICE C compiler for MS-DOS", Version 3.00E, Lattice Incorporated, 1985.

[3] "GSS*CGI Programmer's Guide", Graphics Software Systems, Inc.

[4] T.Ohtsuki, "Layout Design and Verification", North-Holland, 1985.

[5] Yoshida, "A Layout Checking System for Large Scale Integrated Circuits", 14th DA Conf., 1977.

[6] 이 평한, "집적회로 layout 편집 시스템(PC-LADY)의 개발", 대한전자공학회, 1987.