

LSI의 심볼릭 레이아웃 CAD 技術에 關한 研究 動向

李建培*, 鄭正和**

漢陽大學校 工科學 電子工學科
大學院*, 教授(工博)**

I. 序 論

半導體 集積回路가 출현한 이후 최근 급속히 集積度가 향상되고 있으며 高集積回路에 대한 연구가 활발히 전개되고 있다. LSI/VLSI의 칩 사이즈 최소화에 의한 高集積化로 集積回路의 전체 생산 경비를 절감할 수 있으며 이는 시스템의 경제성과 직접 연관되므로 集積回路는 여러 시스템에 널리 사용되어 왔다.

LSI 설계중 CAD 기술은 論理設計 및 그의 시뮬레이션, 레이아웃 設計 등에 응용되고 있으며 특히 LSI의 集積度가 증가함에 따라 설계 사이클에 있어서의 레이아웃 設計는 점점 많은 시간을 필요로 하는 부분으로 되어 가고 있어 LSI 설계에 필수 불가결한 것이 되었다.

本文에서는 LSI/VLSI 설계방식중 심볼(symbol)을 이용한 심볼릭 레이아웃(symbolic layout)技術의 최근 동향에 대하여 解説한다.

II. 심볼릭 레이아웃 方式이란?

심볼릭 레이아웃은 주어진 回路로부터 回路要素(element)를 나타내는 심볼을 정의하고 이 심볼로 심볼릭 다이어그램을 구성한 후 심볼 레벨에서 설계법칙(design rule)을 만족하는 레이아웃을 얻고 면적을 최소화하기 위한 콤팩션(compaction)을 수행한다. 이렇게 얻어진 레이아웃으로부터 각 要素에 해당하는 마스크 패턴(mask pattern)을 割當하여 최종의 마스크 패턴 레이아웃(mask pattern layout)을 얻는다.

이 方式은 IC 마스크 패턴 설계의 세부적이고 복잡한 과정을 간략화하며 이것은 주어진 공정의 설계법칙의 복잡도를 줄이거나 제거함으로써 이루어진다. 설계법칙은 IC 技術에서 사용되는 마스크 층(mask layer)간의 허용간격이나 폭을 의미하며 층간을 연결하는 전기적인 법칙이나 능동소자의 형성등을 포함한다. 이렇게 간략화된 설계법칙을 사용함으로써 신속한 설계와

설계상의 에러를 줄일 수 있다.

이 方式은 레이아웃의 密度(集積度, density)면에서 손에 의한 레이아웃(hand-packed layout)에 필적할만 하며 設計時間면에서는 매뉴얼 레이아웃(manual layout)보다 월등하다. 결과적으로 심볼을 사용한 심볼릭 레벨에서 回路를 다루므로 IC 설계공정의 복잡도를 감소시킴으로써 전문가는 좀더 복잡한 回路를 이해할 수 있으며 초보자에게도 높은 신뢰도의 레이아웃의 설계를 가능하게 한다.

III. 심볼릭 레이아웃 方式

1. 코오스-격자 레이아웃(Coarse-Grid Layout)

코오스-격자 레이아웃은 칩 표면을 x, y 방향의 등간격 격자(grid)로 分割하였으며 격자크기는 각 격자상에 위치할 심볼의 크기나 配置時의 설계법칙을 만족하는 범위등에 의해 결정된다. 바닥에 타일을 붙여 완성하는 것과 유사하게 주어진 回路設計에서 심볼을 격자점상에 위치시켜 回路를 구성한다. 심볼의 집합은 사용되는 종류에 따라 文字, 數字 또는 그래픽 심볼로 정의한다.

AMI⁽¹⁾와 Rockwell Int.에서는 文字를 이용한 심볼릭 레이아웃을 구성하였으며 Hewlett Packard에서는 고정된 격자상에 심볼을 入力로 받아 들일 수 있는 interactive graphic system(IGS)를 개발하였다.

Symbolic interactive design system(SIDS)⁽²⁾는 갈라 文字 터미널을 사용했으며 설계상의 위반을 점검하기 위하여 회로의 전기적 연결상태를 시각적으로 추적하기 위한 추적기능을 갖고 있다.

2. 게이트 매트릭스 레이아웃(Gate Matrix Layout)^(3,4)

최근 대규모 CMOS 回路設計를 위해 Bell Lab.에서 연구되고 있는 방식으로 "Gate Matrix" 방식이 있다. 이 레이아웃에서는 폴리실리콘(polysilicon) 층은

고정된 폭과 높이로 수직방향의 레로만 지날수 있으며 확산(diffusion) 층은 수평방향의 행이나 폴리실리콘층이 없는 수직레로 지날 수 있다. 폴리실리콘의 수직레과 확산영역의 수평행이 교차하는 부분에는 트랜지스터가 생성되며 메탈(meta) 층은 트랜지스터간의 연결을 위해 수직, 수평방향으로 지날 수 있다.

設計方法은 게이트 신호선을 정하여 각 신호선마다 폴리실리콘층의 레를 割當하고 확산층의 행에 의해 트랜지스터를 위치시키고 확산층의 수직레이나 메탈층으로 트랜지스터간을 연결한다.

이 方法은 높은 密度의 레이아웃을 얻을 수 있으며 大型回路 設計에 있어서 손에 의한 레이아웃의 密度에 필적할만 하다.

3. 스틱 레이아웃(Stick Layout)

이 方式은 격자에 제약되지 않고 자유로운 位相 機荷學的 표현법을 사용하므로 그래픽 심볼을 절대좌표 값에 의해 격자상에 위치시키는 것이 아니라 각 심볼간의 상대적인 위치에 의해 각 심볼의 위치가 결정된다. 초기의 루스(loose)한 형태의 레이아웃으로부터 각 要素간의 최소 허용간격을 나타내는 설계법칙을 만족하는 레이아웃을 얻은후에 과다하거나 불필요한 면적을 제거하기 위한 콤팩션을 수행한다. 이렇게 얻어진 심볼릭 레이아웃을 각 要素에 해당하는 마스크 패턴을 割當하여 얻고자하는 마스크 패턴 레이아웃을 얻는다.

이 레이아웃 방식을 이용한 시스템은 Williams¹⁵⁾, Dunlop^{6, 7)}, Hseuh⁸⁾ 등에 의해 연구되어 왔다.

4. 버추얼 격자 레이아웃(Virtual Grid Layout)¹⁶⁾

Bell Lab.에서 고안한 심볼릭 레이아웃 시스템(MU-LGA)¹⁶⁾은 코오스-격자 레이아웃과 유사하게 격자를 이용하여 설계하는 방식으로 격자간의 간격은 이웃하는 격자에 위치하는 要素들의 密度와 상호관계에 의해 결정된다.

그림 1과 같이 격자간의 간격이 位相學에 의해 변하므로 같은 레이아웃을 고정격자표현(그림 1 (b))과 비교하여 그림 1 (c)와 같이 표현할 수 있다.

이 시스템은 데이터 베이스로 IC DL(intermediate circuit description language)¹⁶⁾를 사용하며 이 언어로 각 要素들을 표현할 수 있다.

격자를 이용한 설계방식은 要素를 격자상에 위치시킴으로써 신속히 설계할 수 있으며 세부적인 사항을 포괄적으로 파악할 수 있다.

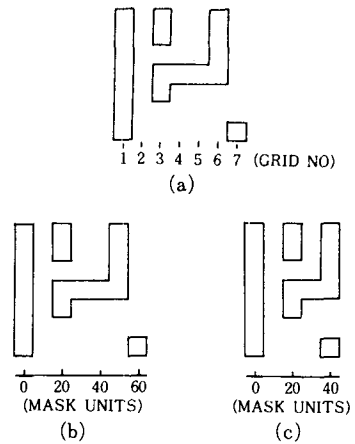


그림 1. (a) 격자상의 회로 표현
(b) 고정 격자 표현
(c) 버추얼 격자 표현

IV. 面積을 줄이기 위한 콤팩션 方法

칩을 설계하는 경우 높은 集積度를 갖는 칩을 설계하는 것이 중요하며 심볼로 표현된 레이아웃에서 면적을 최소화하기 위한 콤팩션 방법에는 여러 가지가 있다.

1. 코오스-격자 콤팩션(Coarse-Grid Compaction)

1970년에 Aker¹¹⁾에 의해 제안된 방법으로 2차원 레이아웃으로부터 과다한 면적을 제거하기 위한 코오스-격자 기법이다.

레이아웃 내부의 어느 곳이나 존재할 수 있는 압축 영역에 해당하는 면적은 줄여 들 수 있으며(그림 2(a)), 압축 영역이 분리되어 있는 경우(그림 2 (b))에는 sheer 라인(shear-line)으로 연결될 수 있는 압축 영역을 제거할 수 있다.

x, y 방향을 별도의 과정으로 번갈아 수행하며 제거될 면적이 없을 때까지 반복한다.

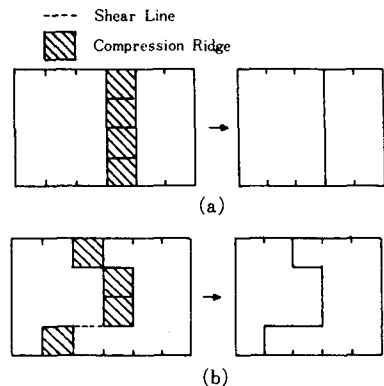


그림 2. 코오스-격자 콤팩션

이때 압축 영역과 웨어 라인에는 약간의 제약 조건이 따른다.

2. 마스크 레이아웃 콤팩션⁽⁶⁾(Mask Layout Compaction)

1.의 방법과 유사하나 격자에 局限되지 않고 마스크 레벨에서 직접 줄어들 수 있는 부분을 찾아 콤팩션을 수행한다. 압축 영역은 레이아웃의 하단으로부터 상단으로 찾아가며 압축 영역이 露인 경우 웨어 라인을 이용하여 가장 가깝게 인접한 압축 영역에 연결하는 방식으로 압축 영역을 찾는다. 이렇게도 인접한 압축 영역에 연결되지 않는 경우 역추적(back tracking)으로 찾는다. 이렇게 하여 얻어진 압축 영역중 실제적으로 줄어들 수 있는 최종 압축 폭(final compression width)만큼을 이동시켜 콤팩션을 수행한다.

그림 3에서 빗금친 부분이 압축 영역을 나타낸다.

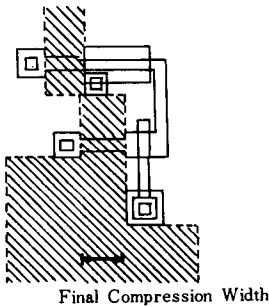


그림 3. 마스크 레이아웃 콤팩션

3. 全体的인 그리고 局部的 콤팩션(Grobal & Local Compaction)

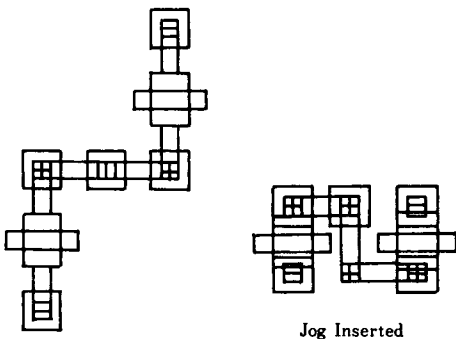


그림 4. 죠그(jog) 삽입

국부 콤팩션은 레이아웃을 몇 개의 부분 영역(sub region)으로 나눠 밀집해 있는 要素간에 죠그(jog)를 삽입하여 몇 개의 국부 영역(local region)으로 분리시킨다.

이렇게 얻어진 국부 콤팩션 레이아웃으로부터 앞의 1. 혹은 2.의 방식으로 전체적인 콤팩션을 수행하여 要素간의 과도한 면적을 제거한다.

그림 4는 죠그가 첨가된 경우 要素들이 밀집될 수 있는 경우를 나타낸다.

SLIM⁽⁷⁾에서는 이 방식으로 콤팩션을 수행한다.

4. 그래프(Graph) 理論을 利用한 콤팩션(Compaction)

1) 최장 경로 탐색(Longest Path Search)⁽⁸⁾

초기 레이아웃중 같은 행이나 열에 묶여 있는 要素群을 절점(vertex)로 표시하고 이 要素群間的 연결상태를 邊(edge)으로 나타내어 그래프를 형성한다. 이때 邊의 웨이트(weight)는 간 要素群간의 설계 법칙(design rule)을 만족하는데 필요한 최소의 간격중 최대(maximum)값을 나타낸다.

그래프의 기준 절점으로부터 각 절점의 위치를 그래프 이론의 최장 경로 탐색 알고리즘을 적용하여 구한다. 이렇게 결정된 절점의 위치가 최종 레이아웃에서 要素群의 위치가 된다.

이 방법은 x, y 방향을 별개의 과정으로 행하고 要素를 要素群으로 묶어 이동시키므로 要素간의 불필요한 면적이 제거되지 않는 경우가 발생하는 단점이 있고 서로 연관되지 않는 要素간에 설계 법칙 위반이 생기는 경우가 있으므로 대각선 방향 설계 법칙⁽¹²⁾을 적용할 필요가 있다. 그러나, 설계 시간이 적게 든다는 것이 잇점이다.

2) 사용자(User)에 의한 제약조건이 첨가되는 경우⁽¹³⁾

要素간의 허용간격을 나타내는 설계법칙이외에도 사용자에게 의한 제약조건이 첨가되는 경우, 예를 들어 특정 2要素간의 간격이 지연시간(delay time) 문제때문에 어느 간격 이상 떨어지지 못하는 경우, 이 사용자에게 의한 제약 조건을 (-)값의 웨이트를 갖는 邊으로 그래프에 삽입하여 이 2가지 조건을 모두 만족하면서 면적이 최소가 되는 레이아웃을 구하는 것이다.

2가지 제약조건이 서로 상반되지 않는 경우에는 반복적으로 해를 구할 수 있다.

이 방법도 x, y 방향 콤팩션을 별개의 과정으로 행하고 모든 조건을 만족하면서 콤팩트(compact)한 레이아웃을 얻을 때까지 반복 시행한다.

5. 기타 方法의 콤팩션(Compaction)

x, y 방향을 별개의 과정으로 취급하여 반복 수행하는 것보다 x, y 방향을 함께 고려함으로써, 좀더 최적 값에 접근할 수 있도록 한다. 전체 레이아웃을 서로 연관이 없는 부분으로 분할하여 연결이 있는 부분간의 관계만을 고려하여 연관이 있는 부분에만 콤팩션을 적용한다.

종전에 정해진 要素의 모양과 연결방식을 필요에 의해 설계법칙이 만족하는 범위내에서 변경(그림 5) 함으로써 겹치게 되어도 전기적으로 영향이 없는 부분을 증가시킴으로써 좀더 콤팩트한 레이아웃을 얻을 수 있다.^[14]

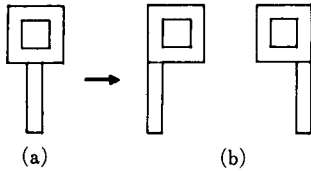


그림 5. (a) 회로要素의 연결
(b) 변경된 연결 방식

V. 今後의 課題

심볼릭 레이아웃 방식은 다른 設計方式에 비해 높은 集積度를 얻을 수 있다(그림 6).^[15]

마스터 슬라이스(master slice) LSI의 경우 이미 정해진 셀(cell)을 사용하므로 設計時間이 단축되나 集積度面에서는 떨어짐을 알 수 있다.

심볼릭 레이아웃 방식은 필요한 회로 내부를 심볼(symbol)을 사용하여 최종 마스크 패턴(final mask pattern)을 얻으므로 집적도 면에서 우수한 칩을 설계할 수 있다.

반면에 앞으로의 과제는 설계시간면에서 회로내부의

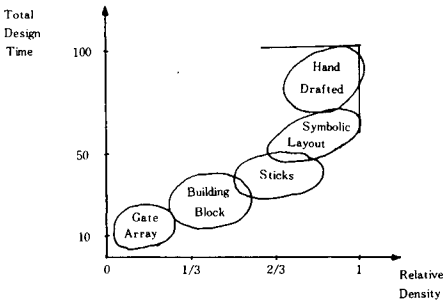


그림 6. 레이아웃 방식의 비교

세부적인 것을 다루어야 하므로 다른 레이아웃방식보다 다소 설계시간이 더 걸린다.

콤팩션 과정에서도 시간 복잡도(time complexity)를 줄이는 연구가 계속되어야 할 것이다.

參 考 文 獻

- [1] Gibson, D and Nance, *SLIC-Symbolic Layout of Intergrated Circuit*. Proc. 13th. Design Automation Conference, pp.434-440, June 1976.
- [2] Clary, D. Kirk, R and Sapiro, S, *SIDS-a Symbolic Interactive Design System*. Proc. 17th D.A Conference, pp.292-295, 1980.
- [3] Lopez, A.D and Law, H.F.S, "A dense gate matrix layout method for MOS LSI," *IEEE Trans. on Elec. Dev.*, pp. 1671-1675, Aug. 1980.
- [4] Wing, O. *Automated Gate Matrix Layout* Proc. IEEE ISCAS, pp. 681-685, 1982.
- [5] Williams, J. *Stick-A, Graphical Compiler for High Level LSI Design*. Proc. of the NCC., pp.289-295, May 1978.
- [6] Dunlop, A., "SLIP-symbolic layout of integrated circuit with compaction" *Computer-Aided Design*, vol.10, no.6, pp.387-391, Nov. 1978.
- [7] Dunlop, A., *SLIM-the Translation of Symbolic Layouts into Mask Data*. Proc. 17th Design Automation Conference. pp.595-602, June 1980.
- [8] Hseuh, M. Y and Pederson, D.O., *Computer Aided Layout of LSI Circuit Building Blocks*. Proc. of 1979 ISCAS, pp.474-477, July 1979.
- [9] Weste, N., *Virtual Grid Symbolic Layout*. Proc. of the 18th Design Automation Conference, pp.225-233, June 1981.
- [10] Weste, N., "MULGA-an interactive symbolic layout system for the design of integrated circuits", *Bell System Technical Journal*, vol.60, no.6, pp.823-857, July-Aug. 1981.
- [11] Akers, S.B. Geyer, J.M., *IC Mask Layout with a Single Conductor Layer*. Proc.

7th. DA Conference, pp. 7-16, 1970.
 [12] 李建培, "LSI의 Symbolic Layout CAD 에 關한 研究", 碩士學位論文, 漢陽大學校, 1983.
 [13] Liao, Y. A and Wong C.K "An algorithm to compact a VLSI symbolic layout with mixed constraints," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and System*, vol. CAD-2 no.2, pp.62-69, Apr. 1983.

[14] Kedem, G and Watanabe, H., *Graph-Optimization Technique for IC Layout and Compaction*. Proc. 20th DA conference, pp.113-120, 1983.
 [15] Avenier, J.P "Digitizing, layout, rule checking-The everyday tasks of chip designers", *Proceedings of the IEEE*, vol.71,no.1, pp.49-56, Jan. 1983. *

♣ 用 語 解 說 ♣

Indexed Sequential File

Sequential file에서는 file내의 record들을 file의 첫번째 record로부터 찾아가야 하므로 시간의 낭비가 많다. Indexed sequential file에서는 sequentile file의 이러한 단점을 보완하기 위하여 file내의 record들을 track에 저장할 때 각 track의 마지막에 위치하는 record들이 들어 있는 주소를 갖고 있는 index table을 만든다. 그림에서 index table을 만들 때 track 2의 마지막 record의 key는 track 1의 마지막 record의 key보다 반드시 커야 한다. 그러나 하나의 track내에서는 record들이 크기 순으로 sorting되어 들어 있을 필요는 없다.

여기서 가령 file A의 record 12를 찾는다고 하

면 record 12의 key와 index table의 record 10의 key의 크기를 비교한다. 그리하면 record 12의 key가 더 크므로 index table의 다음번에 있는 record 20의 key와 record 12의 key를 비교한다. 그리하면 record 12의 key가 더 작으므로 이단계에서 record 12는 track 2 내에 저장되어 있는 것으로 판정하여 record 20이 들어 있는 track 2로 찾아가는 것이다. Track 2 내에서 順次的으로 sequential search를 시행하여 record 12를 검색해 낸다. 이와같이 index table을 이용하여 disk track를 찾아가서 거기서 부터 sequential search를 시행하는 방식으로 record를 검색하는 구조를 가진 file이 indexed sequential file이다.

