

# VLSI 회로정보 및 레이아웃의 Viewer 설계 및 제작

배종국\*, 허성우\*\*  
동아대학교 컴퓨터공학과  
e-mail : \* [jkbae@donga.ac.kr](mailto:jkbae@donga.ac.kr), \*\* [swhur@daunet.donga.ac.kr](mailto:swhur@daunet.donga.ac.kr)

## Design and Implementation of the Viewer for VLSI Circuit and Layout

Jong-Kuk Bae\*, Sung-Woo Hur\*\*  
Dept of Computer Engineering, Dong-A University

### 요 약

VLSI 칩 설계는 매우 복잡한 공정이기 때문에 여러 단계, 즉 크게 분류하여 구조 설계, 논리 설계, 물리 설계 등의 과정을 거쳐 완성하게 된다. 그리고 각 단계에서는 그 단계에서 사용될 수 있는 소프트웨어의 도움을 받게 되며, 이런 소프트웨어의 도움 없이는 오늘날의 고밀도 칩 설계는 불가능하다. 각 단계에서 사용되는 소프트웨어의 주요한 기능 중 하나가 시뮬레이션 등을 통한 설계의 적합성을 테스트하는 것이라면 또 다른 주요한 기능은 설계자로 하여금 눈으로 확인하며, 변형된 설계의 일부를 눈으로 볼 수 있도록 보여주는 기능이라고 볼 수 있다.

본 논문에서는 칩 설계에서 가장 복잡한 단계라고 볼 수 있는 물리 설계 과정에 사용될 수 있는 Viewer 를 설계하고 구현하여 제한한 Viewer 를 통하여 회로의 정보를 보여 주며, 또한 상이한 레이아웃을 비교할 수 있도록 도와 준다. 설계된 Viewer 는 비록 초기버전이지만 물리 설계 단계에서 매우 중요한 정보, 예를 들어 critical net, 상이한 배치 등을 눈으로 확인하게 도와줌으로써 물리 설계에 관계된 다른 소프트웨어의 성능 개선을 유도할 수 있으며 또 실제 칩 설계 현장에서 바로 사용될 수 있기 때문에 실용성이 매우 높다.

### 1. 서론

VLSI 칩 설계는 매우 많은 공정과 시간을 요구한다. 특히 여러 과정 중에서 물리 설계(physical design)가 칩의 성능과 면적, 수율 등에 지대한 영향을 미치기 때문에 이 과정은 매우 중요하다. 물리 설계 자체도 매우 복잡한 공정이기 때문에 여러 세부 단계, 즉 회로분할, 배치, 배선 등의 단계로 나누어 수행된다. 각각의 세부 단계를 위한 연구는 오랫동안 계속되어 왔고 지금도 계속되고 있다. 특히 배치 문제는 물리 설계 단계 가운데서도 매우 중요한 문제로써 지난 20여 년 이상 연구되어 왔다.

배치문제를 위한 접근 방법은 크게 세가지로 분류될 수 있는데 첫째는 분할 기법을 이용한 방법으로써 회로와 칩 영역을 하향식으로 계속

분할해 감으로써 각 셀의 위치를 결정하고자 하는 방법이다[1-4]. 두 번째 접근방법은 force-directed 방법으로 알려진 해석적 기법을 사용하는 것이다. 각 셀의 연결도에 대한 정보를 이용하여 셀의 안정된 위치를 찾고, 중복을 해결함으로써 최종 배치를 얻는 방법이다[5-8]. 세 번째 방법으로는 simulated annealing 기법이라고 알려진 것으로써 일단 주어진 배치를 바탕으로 특정한 두 셀의 위치를 맞 바꾸거나, 한 셀의 위치를 옮기는 작업 등을 반복함으로써 배치의 성능을 개선해 가는 방법이다[9,10]. 최근에 와서는 앞의 기법들을 복합적으로 사용한 하이브리드 알고리즘들이 발표되고 있으며, 결과는 단순히 한 방법을 사용한 것보다 우수한 것으로 보여진다[11-13]. 좋은 배치를 얻기 위해 좋은 배치 휴리스틱을 사용해야 한다는 것은 자명한 일이다. 그러나 좋은

배치 휴리스틱을 개발하기 위해선 배치를 눈으로 확인해 볼 수 있도록 도와주는 그래픽 툴의 기능도 매우 중요하다. 얻은 배치를 눈으로 보면서 배치 휴리스틱이 어떤 약점을 가지고 있는지 확인할 수 있고 그러므로 더 좋은 휴리스틱을 개발할 수 있게 되는 것이다. CAD 툴 개발에 있어서 이런 가시적 툴은 오늘날 필수적인 것으로서 VLSI 칩 설계에 많은 도움을 주고 있다[14].

본 논문에서는 물리 설계 단계에서 사용될 수 있는 그래픽 툴, 즉 레이아웃을 보여주는 그래픽 뷰어(Graphic Viewer)를 설계하고 개발하였다.

제안한 뷰어가 초기 버전이므로 모든 계획된 기능을 다 구현하여 보여주지는 않지만 응용 예가 높은 기능을 우선적으로 구현함으로써 뷰어가 계속 업그레이드 될 수 있음을 보였다.

## 2. 뷰어의 기능

### 2.1. 배치 보여주기

뷰어의 가장 기본적인 기능 중 하나는 얻은 배치를 시각적으로 보여주는 것이다. 회로와 배치 정보를 읽은 다음 각 셀이 어떻게 배치되어 있는지를 보여주는 기능으로써 부분적으로 확대/축소가 가능하다. 확대 및 축소 기능을 통하여 거시적인 배치를 눈으로 확인할 수 있을 뿐만 아니라 특정 지역의 미시적인 배치 상황도 확인할 수 있다.

특정 지역의 미시적인 배치를 확인할 수 있기 때문에 배치 휴리스틱의 문제점을 유추할 수 있도록 도와 준다. 커서가 각 셀 위를 움직임에 따라 셀 이름과 그 셀의 위치정보를 같이 보여준다.

그림 1 에서는 자체적으로 개발 보유한 배치 툴로서 얻은 배치 결과에 대한 전체적인 그림을 보여준다. 이 회로는 반도체 생산으로 유명한 I 사의 회로로서 2056 개의 표준 셀로 구성되었으며, 회사의 기밀상 회로의 이름은 가명을 사용한다.

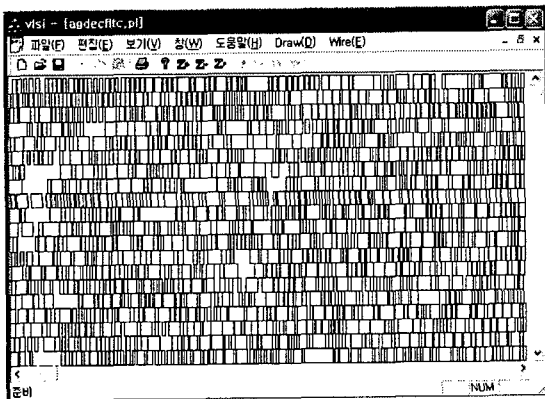


그림 1. 회로 Ckt1 에 대한 전체적인 배치

그림 2 에서는 회로의 특정 부분을 확대하여 보인

것으로써 커서가 놓인 셀의 이름과 위치정보를 보여준다. 이런 확대 및 축소 기능은 사용자의 요구에 따라 어디에서나, 임의의 비율로 적용될 수 있다.

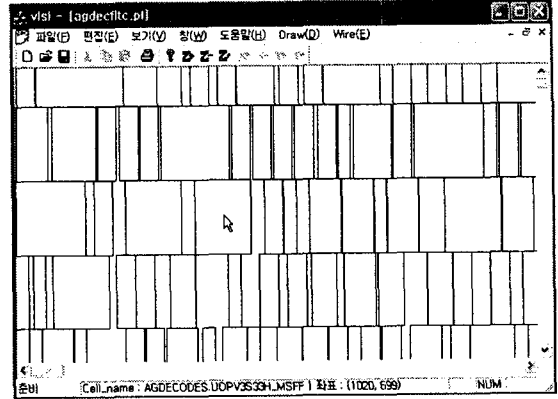


그림 2. 회로 Ckt1 을 위한 배치에서 의 특정 부분을 확대한 그림

### 2.2. 두 개의 상이한 배치 비교하기

배치를 위한 툴을 개발하는 과정 동안 여러 가지 휴리스틱을 사용하여 배치의 질을 높이고 노력하게 된다. 제안한 휴리스틱이 배치에 어떤 영향을 미치는지 알기 위해서 여러 방법이 있겠지만 그 중 한 방법으로는 기존의 방법으로 얻은 배치와 새로운 휴리스틱을 이용하여 얻은 새로운 배치가 어떻게 다른지를 비교함으로써 제안한 휴리스틱의 장점과 문제점을 찾을 수 있다. 즉, 서로 다른 휴리스틱을 이용하여 얻은 두 배치를 비교하는 것은 VLSI 설계 과정 동안 필수이다.

두 배치를 비교하는 방법이 여러 가지가 있겠지만 본 논문에서는 눈으로 두 배치를 비교 할 수 있도록 뷰어의 기능을 추가하였다.

서로 다른 배치를 각각 P1, P2 라 할 때, P1, P2 를 동시에 중첩하여 보여주는 것은 혼란만 야기할 뿐 전혀 도움이 되지 않는다. 본 논문에서는 동일한 표준 셀이 P1, P2 에서 어떻게 서로 다른 위치에 놓여 있는지 그 위치 차이를 쉽게 볼 수 있도록 하였다. 예를 들어 표준 셀 A 의 위치가 (여기서 셀의 위치란 셀의 중심의 위치를 말한다) P1 에서 (x1, y1)이라 하고, P2 에서는 (x2, y2)라고 하면 그림에서는 P1 이나 P2 중 사용자가 원하는 요구에 따라 하나만 보여주고 (Δx, Δy)를 선(벡터)으로 보여 줌으로써 셀 A 가 다른 배치에서는 어디에 있는지를 알 수 있도록 하였다. 벡터 (Δx, Δy)를 선으로 보여 줄 때 선의 시작점과 끝점의 색을 다르게 하고 중간에는 색깔이 점진적으로 바뀌게 함으로써 많은 선들이 보일지라도 혼동스럽지 않도록 하였다. 그림 3 에서는 회로 Ckt1 을 위한 상이한 두 배치에 대해 앞에서 설명한 것처럼 보여준다. 물론 이 그림도 특정 영역에서 확대/축소가 가능하며, 커서 위치한 셀의 이름과

좌표를 아래 상태바에서 보여줌으로써 현재 보여주는 영역이 어디인지를 사용자로 하여금 쉽게 알 수 있도록 하였다.

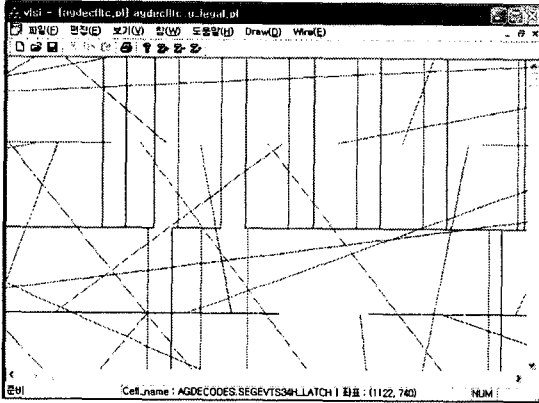


그림 3. 두 개의 상이한 배치 비교하기

### 2.3. 넷 정보 보여주기

좋은 배치를 얻기 위해 CAD 툴을 개선하다 보면 어떤 넷으로 인해 배치의 질이 나빠지는지를 알 필요가 생긴다. 특히 타이밍을 고려한 배치에서는 critical 넷이 무엇인지, 그리고 그 넷에 관계된 셀이 어떻게 배치되어 있는지에 대해 관심이 높아진다. 이런 정보를 눈으로 쉽게 보여주는 기능이 특정 넷에 대한 정보를 눈으로 보여 주는 기능이다. 그림 4 에서 보듯이 넷 degree 가 3 이상인 경우 넷을 둘러싸는 직사각형이 형성되고, 그 직사각형의 중심에 한 심벌(그림에서는 별표)을 표시한 후 그 심벌로부터 그 넷에 관계된 셀의 중심까지를 선으로 연결함으로써 특정 넷에 관한 정보를 눈으로 확인할 수 있도록 하였다. 물론 이 그림 역시 확대/축소가 임의로 가능하다.

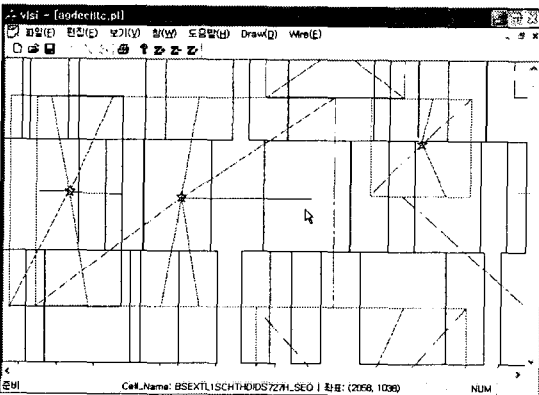


그림 4. 넷 정보 보여주기

### 3. 구현 및 토의

제안한 뷰어는 Compaq Evo W4000 워크스테이션/Windows XP professional 에서 Visual C++ 6.0 을 사용하여 MFC 에서 제공하는 클래스 외 4 개의 클래스를 만들어 구현하였다. 입력용 파일은 모두 bookshelf 형식의 파일을 사용하였다. 그래픽 처리를 위해선 Open GL 을 사용하였다[15,16].

표준 셀 레이아웃을 위한 회로에는 표준 셀들과 패드 셀들로 구성되어 있는데, 배치를 보여주거나, 두 배치를 비교할 때, 특정 넷을 보여줄 때는 패드 셀에 대한 것은 그림에서 제외시켰다. 물론 사용자가 요구하면 패드 셀도 같이 보여 줄 수 있지만 일반적으로 패드셀은 이미 고정된 셀로 보기 때문에 실제 VLSI 물리 설계 특히 배치를 위한 툴 개발에 관련되어서는 패드셀을 따로 취급하므로 우리도 그렇게 처리하였다.

### 4. 결론

본 논문에서 제안하고 구현한 뷰어가 현재 가지고 있는 기능은 가장 기본적이고 필수적인 것이다. 제안한 CAD 툴이 제대로 사용되기 위해선 앞으로 더욱 많은 기능이 보완되어야 할 것이라고 여겨진다.

또한 앞으로 계속 추가하길 원하는 기능 중 하나는 배치를 개선해 가는 과정 즉, 각 셀이 위치를 옮기는 과정 자체를 하나의 동영상처럼 보여 주도록 하는 것이다. 배치의 개선 과정을 동영상처럼 볼 수 있다면 우리는 배치를 위한 휴리스틱의 장점과 문제점을 좀 더 거시적인 관점에서 찾을 수 있을 것으로 생각된다. 또한 배치의 개선을 위한 알고리즘이 어떻게 실제로 적용되고 있는지를 확인할 수 있기 때문에 좀 더 나은 휴리스틱을 찾아 내도록 도와 줄 수 있을 것으로 기대한다.

### 참고문헌

- [1] M. Breuer, "Min-cut Placement," Design Automation and Fault-Tolerant Computing, pp. 343-382, 1977.
- [2] U. Lauther, "A Min-cut Placement Algorithm for General Cell Assemblies Based on a Graph Representation," J. of Digital Systems, Vol. IV, Issue 1, pp.21-34, 1980.
- [3] M. Terai, K. Takahashi and K. Sato, "A New Min-Cut Placement Algorithm for Timing Assurance Layout Design Meeting Net Length Constraint," Proc. of DAC, 1990.
- [4] A. E. Caldwell, A. B. Kahng, and Igor L. Markov, "Can Recursive Bisection Alone Produce Routable Placements?," Proc. of DAC, pp. 477-482, 2000.
- [5] N. R. Quinn and M. A. Breuer, "A Force Directed Component Placement Procedure for Printed Circuit Boards," IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 377-388, 1979.
- [6] J. M. Kleinhans and G. Sigl and F. M. Johannes, "GORDIAN: A New Global Optimization / Rectangle Dissection Method for Cell Placement," Proc. of International Conference on CAD, pp.506-509, 1988.
- [7] Jens Vygen, "Algorithms for Large-Scale Flat

- Placement," Proc. of DAC, pp. 746-751, 1997.
- [8] H. Eisenmann and F. M. Johannes, "Generic Global Placement and Floorplanning," Proc. of DAC, pp. 269-274, 1998.
  - [9] W. Swartz and C. Sechen, "New Algorithms for the Placement and Routing of Macro Cells," In Proc. of DAC, pp. 336-339, 1988.
  - [10] Wern-Jieh Sun and Carl Sechen, "Efficient and Effective Placement for Very Large Circuits," IEEE Trans. on CAD, pp.349-359, 1995.
  - [11] M. Sarrafzadeh and M. Wang, "NRG: Global and Detailed Placement," Proc. of ICCAD, pp. 532-537, 1997.
  - [12] X. Yang, M. Wang, K. Egur, and M. Sarrafzadeh, "A Snap-on Placement Tool," Proc. of Intl. Symposium on Physical Design, pp. 153-158, 2000.
  - [13] Sung-Woo Hur and John Lillis, "Mongrel: Hybrid Techniques for Standard Cell Placement", Proc. of ICCAD, pp. 165-170. 2000.
  - [14] Phillip J. Restle, "Technical Visualizations in VLSI Design," Proc. of DAC, pp. 494-499, 2001.
  - [15] Richard S. Wright, Jr. Michael Sweet "OpenGL Super Bible," 2nd. January. 3.
  - [16] Kevin Hawkins, Dabe Astle. "Open GL Game Programming," 2001. October. 8.