

## 표준 셀 배치를 위한 가시화 도구

김동현<sup>o</sup> 허성우

동아대학교 컴퓨터공학과

skipio\_2000@yahoo.co.kr<sup>o</sup>, swhur@daunet.donga.ac.kr

### Visualization Tool for Standard Cell Placement

Dong Hyun Kim<sup>o</sup> Sung Woo Hur

Dept. of Computer Engineering, Dong-A University

#### 요 약

VLSI 셀 배치문제는 셀(노드)과 넷으로 구성된 회로를 최소의 비용과 제약조건을 만족하면서 칩 위에 배치하는 문제로 지금까지 여러가지 다양한 배치 알고리즘들이 제시되어왔다. 배치 알고리즘은 입력으로 회로정보를 받아서 결과를 숫자 데이터로 출력하기 때문에 알고리즘 개발자는 배치결과를 숫자 데이터로 분석할 수 밖에 없다. 이런 점에서 실험 결과에 대한 신뢰성 문제가 발생될 수 있는데 2003년 발표된 벤치마킹에 관한 논문[1]에 따르면 뛰어난 성능을 가진 대표적인 알고리즘들을 비교 분석한 결과 실제 최종 배치된 결과에서 셀들의 오버랩 현상과 균일하지 못한 배치 등 몇 가지 문제점들이 지적되었다.

본 논문에서는 이러한 문제점들을 해결하기 위해 알고리즘 개발자가 실험 결과를 가시화해서 직접 확인하고 분석할 수 있는 가시화 도구를 제안한다. 또한 가시화 도구는 결과분석을 통해 보다 향상된 알고리즘 개발에 도움을 주는 것을 목적으로 한다.

#### 1. 서 론

VLSI 칩 설계는 많은 공정 과정을 거치는데 그 중 표준 셀 배치 과정은 칩에 들어갈 표준 셀들을 특정 알고리즘을 이용하여 최소의 비용으로 회로의 제약조건을 만족하면서 배치하는 최적화 문제의 하나이다. 배치문제의 목적이 되는 비용으로는 배선 길이(wire length), 배선 밀집도(routing congestion), 공간 사용률(space utilization) 등이 있는데 이 중 가장 중요한 비용이 배선길이(혹은 넷 길이)이다. 배치 알고리즘은 크게 광역배치(global placement)와 상세배치(detailed placement) 두 단계를 거치게 된다. 광역배치는 일종의 추상적인 배치로서 그래프 분할(partition) 문제의 한 종류이다. 그래프 정보를 담은 빈(bin)이라는 집합을 2개 혹은 그 이상의 부분 집합인 빈들로 나누어 가는데 이 때 두 빈 사이에 걸쳐서 연결된 넷을 넷 컷(net cut)이라고 하며 넷 컷의 합을 컷 사이즈라고 한다. 광역 배치에서는 컷 사이즈를 최소화하면서 빈들을 일정 크기로 쪼개게 되는데 컷 사이즈가 적을수록 최종 넷 길이도 줄어들게 된다. 상세배치는 광역배치 이후 일정 개수로 나뉘어져 빈에 담긴 셀들을 목적 비용이 최소가 되게 하면서 실제 2차원 평면 좌표계 위에 배치하는 것으로 표준 셀 배치의 경우 셀 높이의 일정함을 이용해서 각 행(row)별로 균일하게 배치되어야 한다. 이 때 각 행의 높이를 최대한 서로 비슷하게 배치해야 전체 배치가 균일해지며 공간 사용률도 높일 수 있다.

지금까지 많은 다양한 배치 알고리즘들이 제시되어 왔는데 그 중 뛰어난 성능을 발휘하는 것으로 Dragon, Mongrel, FengShui, Capo 가 있다. 이러한 알고리즘들을 이용하여 벤치마킹 회로들을 최종 배치한 결과, 몇 가지 문제점들이 지적되

었다[1]. 최종 배치된 셀들을 가시화한 결과 셀들의 오버랩(겹침) 현상이 나타나고 그러한 오버랩들을 제거하는 적법화(legalization)과정을 거쳤을 경우 보고된 결과보다 넷 길이가 더 길어지고 전체 배치가 균일하지 못한 결과가 확인되었다. 또한 벤치마킹 회로에 따라 알고리즘 성능의 차이가 발생되었다. 이러한 문제들을 해결하기 위해서는 실험자가 실험결과를 가시화해서 배치결과를 눈으로 확인하고 오버랩 등을 검출해 줄 수 있는 가시화 도구가 필요하다. 또한 알고리즘의 중간결과 혹은 최종결과를 가시화해서 확인할 경우 알고리즘의 특징과 장단점 등을 분석할 수 있다.

본 논문에서는 VLSI 표준 셀 배치 결과를 가시화하는 도구를 제안하고 구현하였다. 제안된 가시화 도구는 표준화된 포맷을 가진 벤치마킹 회로정보에 따라 셀과 넷 정보를 2차원 그래픽으로 보여주며 셀 오버랩을 검출, 제거하고 서로 다른 배치 결과를 비교하는 기능 등을 가짐으로써 알고리즘 개발자가 실험 결과를 분석하여 더 나은 알고리즘을 개발할 수 있도록 도움을 준다.

#### 2. 가시화 도구의 목적

가시화 도구의 목적은 기본적으로 배치 알고리즘의 결과를 보여주고 그것을 분석하는 것이다. 실제 배치된 셀과 넷들을 확인함으로써 실험결과들의 특징을 알아낼 수 있다. 그리고 좋은 결과와 나쁜 결과를 서로 비교하면서 그 원인을 찾는 데도 도움을 준다. 배치 결과에는 광역배치와 상세배치 두 가지가 있는데 광역배치 결과는 알고리즘의 중간 결과로서 회로가 어떻게 분할되어 가는지 알아보는 것이 목적이다. 상세배치 결과는 광

역배치 이후 최종적으로 모든 셀들의 배치가 끝난 상태의 결과이며 전체 배치 결과를 모두 확인할 수 있다. 상세배치에서 오버랩같은 경우, 최종 배치 결과를 눈으로 확인하지 않으면 찾아내기가 어려운데 본 프로그램에서는 자동으로 오버랩을 검출하는 기능을 가지고 있어서 완전한 배치 알고리즘을 설계하는데 도움을 준다.

3. 가시화 도구의 기능

3.1 광역배치 가시화 기능

광역배치는 서론에서 언급한 바와같이 전체 회로를 2개 이상의 부분 회로로 반복해서 나누어가며 셀들의 대략적인 배치 위치를 결정하는 과정이다. 이때 나뉘어진 각 셀들이 속한 집합을 빈이라고 한다. 광역배치는 표준 셀 배치 알고리즘의 결과에 실질적으로 가장 큰 영향을 미치는 과정으로 가시화 도구로 분석해보면 그래프 분할(partition) 알고리즘의 성능을 알 수 있다. 그래프 분할 알고리즘의 목적은 분할된 부분 그래프들 사이에 연결되는 넷 혹은 엣지(edge)의 수를 최소로 하는 것이며 광역배치에서는 이러한 넷들을 넷 컷이라고 하고 넷 컷의 합을 컷 사이즈라고한다. 광역배치는 보통 초기에 전체 영역을 2x2, 4x4 와 같이 동일한 개수의 행과 열로 나누는데 이렇게 나뉘어진 빈들을 다음 반복(iteration)에서 다시 각 빈별로 나누어 간다. 각 빈이 담고있는 셀의 개수와 셀들의 크기의 합이 일정 크기만큼 적어지게 되면 광역배치는 종료되고 상세배치 단계로 넘어가게 된다.

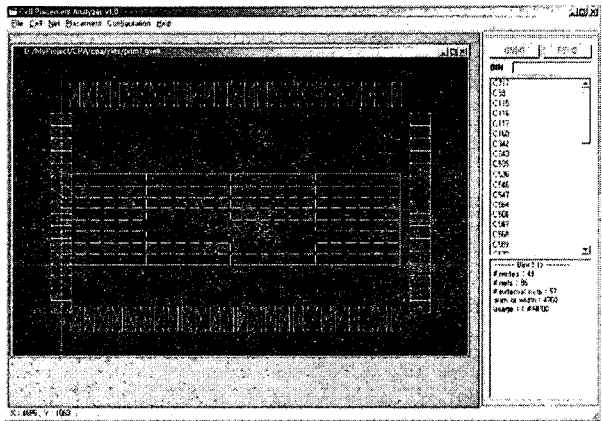


그림 1. 광역배치 결과와 빈 사이의 컷 사이즈 측정

그림 1.은 mongrel[2] 알고리즘을 이용해서 표준벤치마킹 회로의 하나인 prim1 회로를 광역배치 단계에서 분할한 결과를 가시화한 모습이다. 광역배치 가시화에서 빈들은 사각형으로 표현되며 전체 모양은 그림과 같이 격자 모양이 된다. 가시화 도구는 정해진 포맷을 따르는 광역배치 결과파일을 읽어서 보여주며 사용자의 입력에 따라 빈의 크기와 상태, 빈에 속한 셀과 넷 등을 보여주고 빈과 빈 사이의 컷 사이즈를 측정한다.

이렇게 측정된 정보들은 알고리즘 개발자에게 배치기(placer)

의 광역배치 혹은 그래프 분할 루틴의 성능과 특징을 분석할 수 있는 정보를 제공해준다.

3.2 상세배치 가시화 기능

상세배치 가시화에서는 기본적으로 최종 배치가 끝난 셀과 넷 정보를 보여주는데 넷의 경우 3개 이상의 셀들과 연결된 하이퍼 넷(hypernet)이 대부분이기 때문에 서로 다른 표현 방식이 있을 수 있다. 본 프로그램에서는 넷 표현방식으로 그림 2.와 같이 바운딩 박스와 최소 신장 트리 (Minimum Spanning Tree) 두 가지를 사용한다. 넷 길이 측정도 표현방식과 같은 방법을 사용하는데 실제 배선길이와 가장 근접한 방법인 Half-Perimeter 방식의 경우 그림 2.(a)와 같이 한 넷에 연결된

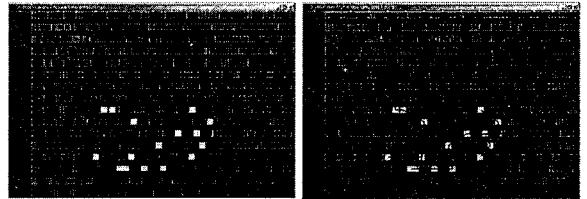


그림 2. 셀과 넷의 표현

셀들을 모두 포함하는 바운딩 박스 둘레의 반을 넷 길이로 계산한다. 그림 2.(b)는 최소 신장 트리 방식의 표현으로 실제 배선길이와는 다소 차이가 있지만 서로 근접한 셀들을 쉽게 구분할 수 있다.

3.3 셀 오버랩 검출 기능

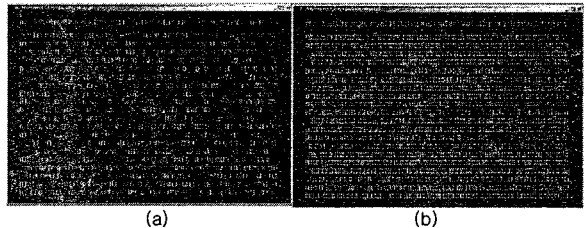


그림 3. 오버랩검출(a)과 오버랩제거(b) 후의 배치상황

셀 오버랩(겹침)을 검사하기 위해 각 행별로 일렬로 배열되는 표준 셀 배치의 특징을 이용한다. 먼저 각 셀들을 y축 좌표에 따라 정렬하게 되는데 이때 같은 행에 있는 셀들끼리 많은 동률(tie)이 발생하면서 행과 행이 서로 구분된다. 이렇게 정렬된 셀들을 다시 각 행별로 x축 좌표에 따라 정렬하게 되면 가장 왼쪽아래에 있는 셀부터 가장 오른쪽위에 있는 셀까지 정렬된 정보를 얻을 수 있다. 이렇게 정렬된 정보를 가지면 한 셀에 대해서 다음 셀과의 x좌표값과 넓이값을 비교하면 오버랩 여부를 판단할 수 있다. 따라서 대략 O(n) 시간에 모든 셀을 검사해서 오버랩을 검출할 수 있다. 오버랩된 셀이 검출되면 같은 행에 있는 셀들을 재정렬해서 오버랩을 제거하고 결과를 볼 수

있다. 이때 재정렬 방법은 오버랩된 셀과 그 위에 있는 모든 셀들을 오른쪽으로 이동시키는 방식이다. 오버랩이 많을수록 셀들이 오른쪽으로 밀려서 행의 넓이가 넓어지므로 전체 배치의 불균형을 발견할 수 있다. 그림 3은 오버랩된 셀들과 오버랩을 제거한후 전체 셀들을 재정렬했을때의 모습이다.

### 3.4 부분배치(Subplacement)

부분배치(Subplacement)는 사용자가 지정한 특정 영역에서의 배치결과를 분석하기 위한 것으로 해당 영역에서의 공간 사용률(space utilization), 셀과 넷의 개수, 넷 밀집도(net congestion) 측정과 임의의 두 부분배치 사이의 컷 사이즈 등을 보여줌으로써 특정영역에서의 배치상태를 볼 수 있다.



그림 4. 부분배치와 컷 사이즈 측정

부분배치는 그림 4.과 같이 사용자가 지정한 임의의 영역에 관한 정보를 보여주는데 넷 밀집도 계산의 경우 부분배치 영역 안에 있는 셀과 영역 밖에 있는 셀을 연결하는 넷의 개수를 계산함으로써 구해질 수 있다. 광역배치 단계에서 셀들이 어떻게 분할 되었는지 유추하기위해 사용자는 부분배치 영역 내에 임의의 선을 긋고 그 선에 따라 다시 분할되는 두 영역 사이의 컷 사이즈를 측정함으로써 해당영역에서 그래프분할이 어떻게 되었는지를 알아낼 수 있다.

### 3.5 셀 위치이동

배치 알고리즘의 중간 혹은 최종 결과로 광역배치나 세부배치 결과를 가시화 하였을때 사용자가 임의로 셀들의 위치를 서로 이동시켜 보면서 변화된 배치결과를 미리 볼 수 있다. 이렇게 셀들을 임의로 이동시켜 봄으로써 알고리즘의 동작 과정을 유추하거나 알고리즘의 새로운 동작을 유도할 수 있는 아이디어를 얻는데 도움을 준다.

### 3.6 배치결과 비교

배치 알고리즘은 한 회로에 대해서도 매번 다양한 배치결과를 내며 변경된 알고리즘을 적용했을때도 상이한 결과가 나온다.

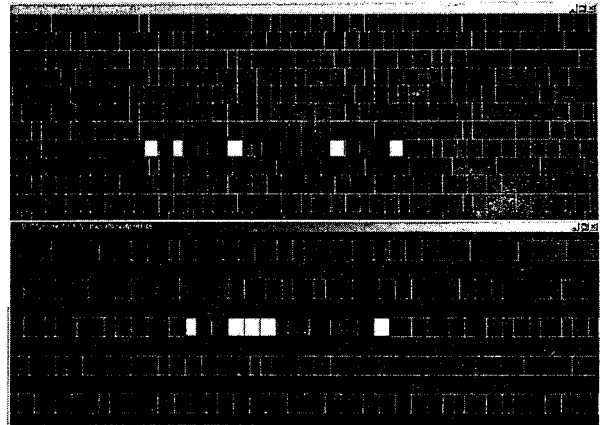


그림 5. 배치결과 비교

이때 서로 다른 배치 결과를 비교해보면 각 알고리즘의 상대적인 특징과 장단점을 파악하는데 도움을 준다. 그림 5.는 5개의 셀이 연결된 한 넷에 대해서 서로 다른 배치결과를 비교한 그림이다. 아래쪽의 결과에서 셀들이 좀 더 서로 가깝게 배치가 되었으며 넷 길이 역시 더 짧아졌음을 알 수 있다.

## 4. 결론

본 논문에서 제안한 가시화 도구는 qt 라이브러리 3.0으로 구현하였으며 컴파일은 MS-Visual C++와 리눅스의 gcc를 사용하였다. 본 툴을 사용해서 배치 알고리즘의 결과를 가시화하였을때 알고리즘 개발자가 각 셀과 넷에 대해서 직관적으로 배치 상황을 알아볼 수 있다는 점과 임의의 셀이나 넷에 대해서 해당 배치상황이 일어난 원인을 유추하는데 도움을 주었다.

앞으로 알고리즘 개발자가 툴을 사용하면서 필요로하는 다양한 기능들을 추가해서 계속 업그레이드를 할 예정이다.

## 참 고 문 헌

- [1] Saurabh N. Adya, Mehmet C. Yildiz, Igor L. Markov, Paul G. Villarrubia, Phiroze N. Parakh, Patrick H. Madden, "Benchmarking For Large-scale Placement and beyond", Proc. of the 2003 International Symposium on Physical Design, April, 2003
- [2] Sung-Woo Hur and John Lillis, "Mongrel: Hybrid Techniques for Standard Cell Placement", Proc. of International Conference on CAD, pp. 165-170, Nov. 2000, San Jose, USA.