

표준 셀 배치를 위한 하이브리드 기법

(Hybrid Techniques for Standard Cell Placement)

허성우^{*} 오은경^{**}
(Hur Sung Woo) (Oh Eun Kyung)

요약 본 논문은 표준 셀을 효율적으로 배치하는 하이브리드 기법을 제시한다. 제시한 기법에서는 레이아웃 영역을 $n \times m$ 그리드로 분할하고, 분할된 그리드의 각 빈에 셀을 배치함으로써 광역 배치를 얻는다. 광역배치를 얻은 후 Relaxation-Based Local Search (RBLS)라 불리는 해석적 최적 기법[12]을 사용하여 배치를 개선시킨다. 이 기법을 사용하면 배치 전체를 개괄적으로 볼 수 있게 되어 배치가 국부 최적에 빠지지 않도록 할 수 있다. 이 기법의 핵심은 중복을 해결하기 위한 매우 효율적인 적법화 알고리즘이라고 볼 수 있는데, 이에 대해 본 논문은 자세히 설명한다. 광역배치가 더 이상 개선되지 않으면 이를 상세 배치로 변환시킨다. 상세 배치를 얻은 후 “최적 인터리빙”이라 불리는 기법[13]을 사용하여 이를 더욱 개선시킨다. MCNC 벤치마킹 회로에 대한 실험 결과는 [14]에서 보여준 Feng Shui의 결과와 필적할 만하다.
키워드 : VLSI 레이아웃, 표준 셀, 배치

Abstract This paper presents an efficient hybrid techniques for a standard cell placement. The prototype tool adopts a middle-down methodology in which an $n \times m$ grid is imposed over the layout area and cells are assigned to bins forming a global placement. The optimization technique applied in this phase is based on the Relaxation-Based Local Search (RBLS) framework [12] in which a combinatorial search mechanism is driven by an analytical engine. This enables a more global view of the problem and results in complex modifications of the placement in a single search “move.” Details of this approach including a novel placement legalization procedure are presented. When a global placement converges, a detailed placement is formed and further optimized by the optimal interleaving technique[13]. Experimental results on MCNC benchmarking circuits are presented and compared with the Feng Shui’s results in [14]. Solution qualities are almost the same as the Feng Shui’s results.

Key words : VLSI layout, standard cell, placement

1. 서론

VLSI 회로의 기술 발달로 인해 오늘날에는 한 칩 내에 수십 또는 수백 만개의 트랜지스터가 있기 때문에 이를 설계하는 과정은 여러 단계를 거치게 되고, 각 과정에 적합한 CAD 툴 들을 이용하여 설계하게 된다. 여러과정 가운데 물리 설계는 매우 복잡하고 시간이 많이 걸릴 뿐만 아니라 이 설계 결과는 칩의 면적, 성능 등에 절대적인 영향을 미치기 때문에 물리설계 과정 자체도 여러 세부 단계를 거치게 된다. 물리설계 과정의 여러 세부 단계 중 하나가 배치문제인데, 이 문제는 물리설계

과정에서 가장 핵심적이며 중요한 문제라고 볼 수 있다. 물리 설계의 중요한 목적인 면적 최소화, 배선 가능성, 타이밍 등에 배치가 미치는 영향이 매우 중요하기 때문에 이에 관한 연구는 지난 20여년 동안 매우 활발하게 되어 왔다. 레이아웃 스타일도 여러 가지로 분류가 가능 한데 본 논문에서는 표준 셀을 이용한 레이아웃에서 셀 배치를 위한 매우 효과적인 기법을 제안하고 이에 대한 실험 결과를 통해 제안한 기법의 효율성을 입증한다. 표준 셀 배치는 행-기반 배치라고도 불리는데 기능과 내부 구조가 미리 잘 설계된 높이가 같은 표준 셀을 이미 정의된 행에 적절히 배치함으로써 추구하는 목적함수 값을 최소화 (또는 최대화) 시키는 문제이다. 물론 셀들이 서로 중첩되지 않게 배치해야 한다. 전통적인 표준 셀 배치의 목적함수는 추정 배선 길이의 합을 최소화시키는 것인데, 배선길이의 최소화는 칩 면적을 줄이는데 직접적인 영향을 미치며 따라서 칩 수율을 증대시킬 수 있다. 배선길이를 최소화하기 위해서는 각 넷(net)의 배

· 본 논문은 2001학년도 정보통신부 IT학과 장비지원사업의 동아대학교
대용자급에 의해 연구되었음

^{*} 중신회원 : 동아대학교 전기전자컴퓨터공학부 교수
swhur@daunet.donga.ac.kr

^{**} 비 회원 : 동아대학교 컴퓨터공학과
ekoh@donga.ac.kr

논문접수 : 2002년 12월 11일

심사완료 : 2003년 8월 6일

선길이를 추정할 모델이 필요한데 가장 흔히 사용되는 배선길이 추정모델은 Half-Perimeter(HP)를 이용하는 것이다. HP 모델이란 각 넷에 포함된 모든 핀을 포함하는 최소의 직사각형을 구한 후, 그 직사각형 둘레 길이의 반을 해당 넷의 길이로 추정하는 모델이다. HP 모델이 넷 길이를 위해 널리 사용되는 이유는 우선 넷 길이 계산이 매우 간단하며, 핀 수가 2 또는 3인 경우의 넷에 대해선 직교 Steiner 트리 모델의 길이와 정확히 일치하기 때문이다. Wang과 Sarrafzadeh는 HP 모델에 근거한 배선길이의 최소화가 광역적으로 볼 때 배선 밀집도를 최소화시키는데도 도움이 됨을 보였다[1].

배치문제는 NP-hard 문제이며, 이를 위한 효과적인 휴리스틱이 그 동안 많이 발표되었다. 발표된 기법들은 다음과 같이 크게 분류해 볼 수 있다. 첫째는 넷 리스트 분할에 기초한 방법이다[2-4]. 이는 넷 리스트를 둘 또는 넷으로 재귀적으로 분할하고, 또한 레이아웃 영역도 수평 또는 수직으로 반복하며 재귀적으로 분할한다. 각 넷 리스트의 크기가 지정한 크기 이하가 될 때까지 분할을 계속한 후, 각각의 작은 넷 리스트를 작게 분할된 레이아웃 영역에 배치한다. 이는 분할정복법의 개념을 이용한 것으로서 넷 리스트를 분할할 때의 목적함수는 컷 사이즈를 최소화시키는 것이 일반적이다. 넷 리스트 분할에 기초한 배치 기법에 관해 수 많은 논문들이 발표되었는데 대부분이 기본적으로 Fiduccia와 Mattheyses가 발표한 알고리즘(FM 알고리즘) [5]의 변형을 이용한 것이라고 볼 수 있다. 배치를 위한 두 번째 기법은 simulated annealing에 기초한 방법이다[6,7]. 이는 주어진 초기 배치로부터 다양한 셀-이동을 통하여 변형된 배치를 얻는 방법으로써 좋은 배치를 얻기 위해서는 시간이 많이 걸리는 단점이 있으나 시간이 충분한 경우에는 매우 성공적으로 좋은 배치를 얻을 수 있는 방법으로 알려져 있다. 시간이 많이 걸리는 단점을 극복하기 위해 어떤 논문에서는 simulated annealing과 다른 방법을 같이 적용하기도 하였다[8,9]. 배치를 위한 세 번째 기법은 해석적 방법이다. 각 넷에 연결된 셀들은 상호간에 인력이 작용하는 것으로 보고 이런 인력을 이용해 각 셀의 위치를 수학적 방법으로 구한다[10,11]. 이 방법의 문제는 많은 셀들이 중첩된다는 점인데 이를 위해서 대부분 분할기법에 기초한 하향식 기법을 이용하여 중첩문제를 해결한다.

본 논문에서는 표준 셀 배치문제를 위한 하이브리드 기법을 제안한다. 제안하는 기법은 크게 두 단계로 작동한다. 첫째 단계에서는 광역배치(global placement)를 구하는 단계로서 칩을 $n \times m$ 격자(grid)로 나눈 후 이 격자 상에 표준셀을 우선 배치한다. 다른 논문에서 광역 배치를 위해 주로 사용하는 격자는 가변적인데 반해 본

논문에서 사용하는 격자는 일단 크기가 결정되면 고정된다는 점이다. 광역배치를 얻은 후 이를 상세배치(detailed placement)로 변환하고 변환된 상세배치를 최적화시키기 위한 알고리즘을 이용해 결과를 구한다. 제안된 배치 기법은 구현과 실험을 [14]에서 보여준 Feng Shui의 결과와 비교하였다. 비교의 공정성을 기하기 위해 동일한 행수를 사용하였으며, GSRC bookshelf 양식을 이용한 회로를 사용하였다[15].

2. 용어 설명 및 문제 정의

물리 설계에 사용되는 회로는 넷 리스트에 의해 정의된다. 넷이란 각 셀이 공유하는 신호를 연결하는 회선으로 볼 수 있고, 넷 리스트란 회로에 포함된 넷 정보를 나열한 것을 의미한다. 넷 리스트는 하이퍼그래프 $G=(V, E)$ 로 나타내며 여기서 V 는 셀 집합을 의미하며 E 는 넷 집합을 의미한다. 하이퍼에지 $e \in E$ 는 V 에 속한 셀의 부분집합 즉, $e \subset V$ 를 의미한다. $|e_i|$ 는 넷 e_i 에 속한 셀의 개수를 의미하며 $s(v)$ 는 셀 v 의 크기(size)를 나타낸다. A 는 모든 셀의 크기 합 즉, $A = \sum_{v \in V} s(v)$ 를 나타낸다. 넷 리스트는 하이퍼그래프로 표현되고, 그래프에서 V 는 노드의 집합으로 볼 수 있으므로 앞으로 셀과 노드는 같은 의미를 갖는 용어로 사용하기로 한다.

광역배치 단계에서는 모든 셀이 놓여지는 영역 즉, 코어 영역(core area)을 $R \times C$ (여기서, R 은 행의 수를 말하는데 이는 칩 설계자가 정해진 값을 주로 사용한다. C 의 값은 시스템 사용자가 정할 수도 있고, 시스템이 자동으로 정할 수도 있다) 격자(grid)로 나누고, 나누어진 각각의 격자를 빈(bin)이라고 부르고 i 번째 행과 j 번째 열에 있는 빈을 $b_{i,j}$ 로 나타낸다. 광역배치에서는 모든 셀을 각 빈에 배치한다. 이때 동일한 빈에 놓인 셀은 중첩이 허용되며, 각 빈의 기본 크기는 코어 영역을 빈 수로 나눈 것 즉, $\frac{A}{(R \times C)}$ 가 된다. 셀의 크기가 서로 다른 점과 광역 배치에서는 셀의 대략적인 위치를 결정하기 때문에 실제로 각 빈의 크기는 고정되기 보다 ϵ 만큼 여유를 가지고 있다. 또 각 행에 배치된 셀의 총면적은 가능하면 비슷해야 전체적으로 코어 영역의 낭비가 줄어든다. 광역배치에서는 다음과 같은 제약 조건을 만족하면서 총 배선길이를 최소화시키는 문제로 볼 수 있다.

$$(1 - \epsilon) \cdot \frac{A}{(R \times C)} \leq \sum_{v \in b_{i,j}} s(v) \leq (1 + \epsilon) \cdot \frac{A}{(R \times C)} \quad \text{조건 (1)}$$

$$(1 - \zeta) \cdot \frac{A}{R} \leq \sum_{j=1}^C \sum_{v \in b_{i,j}} s(v) \leq (1 + \zeta) \cdot \frac{A}{R} \quad \text{조건 (2)}$$

조건 (1)은 모든 빈에 배치된 셀 크기의 합이 기준 빈 크기에 크게 벗어나지 않게 한다. 이렇게 함으로써 광역배치의 결과는 최종적을 구할 상세배치의 결과와 크게 차이가 나지 않게 된다. 조건 (2)는 각 행에 배치된 셀 크기의 합이 ± 5 범위 내에 있게 한다. 이렇게 함으로써 각 행이 대략적으로 같은 크기가 되고 따라서 칩 면적을 최소화시킬 수 있게 된다. 광역배치 P 가 위의 두 조건을 만족시킬 때 우리는 '적법'(legal)하다고 한다. 상세배치에서는 셀의 중첩이 허용되지 않기 때문에 조건 (2)만 만족하면서 셀이 중첩되지 않을 때 상세배치 P 가 적법하다고 말한다. 셀 v 의 중심 좌표를 (x_v, y_v) 로 나타낼 때 넷 e_i 의 길이 $len(e_i)$ 는 다음과 같이 표현될 수 있다.

$$len(e_i) = \max|x_u - x_b| + \max|y_u - y_b| \quad (u, v, a, b \in e_i)$$

그러면, 배치문제는 "모든 적법한 배치 P 에서 $\sum_{i=1}^k len(e_i)$ 를 최소화"시키는 문제가 된다.

3. 광역배치 알고리즘

Hur와 Lillis는 선형배치 알고리즘을 소개하면서 'Relaxation 기반 국부 검색(RBLS: Relaxation Based Local Search)' 기법을 소개한 바 있다[12]. 본 논문에서는 광역배치 단계에서 RBLS 기법의 개념을 그대로 적용하였다. RBLS 알고리즘의 기본 개념은 다음과 같이 요약될 수 있다. 현재의 주어진 배치를 해 공간의 한 지점으로 간주한다. 그리고 해 공간(solution space)에서 잘 정의된 '이동 연산(move operation)'를 이용하여 이웃 해(neighbor solution)를 검색한다. 이웃 해가 더 좋은 배치이면 이웃 해를 현재의 지점으로 간주하고 계속 해 공간 검색을 계속하며, 만약 이웃 해가 더 나쁘면 원래의 지점으로 돌아와서 다른 이웃을 검색한다. 원래의 지점으로 돌아올 수 있는 최대 횟수는 미리 정의된 파라미터에 의해 결정되며 그 최대 횟수를 초과하여 현 지점으로 돌아오면 그 때의 해를 최적해로 간주하고 해 공간 검색을 마친다. RBLS 알고리즘의 핵심은 어떻게 이웃 해를 찾을 것인가 하는 점 즉, 어떻게 이동 연산을 정의하는가 하는 점이다. Simulated annealing 기법에서처럼 단순히 두 셀을 이동하는 식 따위로 이웃 해를 찾을 경우 해 공간 검색에 매우 많은 시간이 요구되고 또 더 중요한 문제는 해가 국부 최적(local optima)에 빠지기 쉽다는 점이다. 이런 문제점을 극복하기 위해 RBLS 알고리즘에서는 매우 효과적인 이동 연산을 제시한다. 이동 연산은 다음과 같이 요약될 수 있으며 각각의 단계에 대해선 나중에 자세히 설명한다.

- 부분회로 추출(Sub-circuit extraction)

- 최적 relaxed 배치¹⁾를 찾음(Optimal relaxed placement)
 - 배치를 적법화 시킴²⁾(Placement legalization)
 - 분할기법을 이용하여 배치를 최적화시킴(Optimization using partitioning technique)
- 광역배치를 위한 알고리즘 RBLS를 의사코드(pseudo code)로 나타내면 다음과 같다.

Algorithm RBLS (Relaxation Based Local Search)	
Input:	m, k , and current placement P
Output:	new placement
1.	Counter $\leftarrow k$
2.	while (Counter > 0) {
3.	Extract a mobile node set M ($ M = m$)
4.	Determine a fixed node set F using M
5.	For $\forall v \in M$, determine optimal relaxed location
6.	$P' \leftarrow$ new placement after legalization
7.	$P' \leftarrow$ optimize P' by local partitioning
8.	if ($WL(P') < WL(P)$) then
9.	Counter $\leftarrow k$
10.	$P \leftarrow P'$
11.	else
12.	Counter \leftarrow Counter - 1
13.	}
14.	return P

그림 1 알고리즘 RBLS

3.1 부분회로 추출(Sub-circuit Extraction)

본 절에서는 알고리즘 RBLS의 라인 3, 4에서 실행하는 부분회로 추출에 대해 설명한다. 라인 3에서는 주어진 파라미터 m 을 이용하여 현재의 배치 P 에서 m 개의 셀(노드)을 선택한다. 우리는 이 m 개의 셀을 '이동 가능한(movable or mobile) 셀 또는 노드'라고 부르고 이 노드의 집합을 M 이라고 부른다. 이 선택된 노드들은 다음 단계에서 구할 최적 relaxed 배치에서 다른 빈으로 이동가능하기 때문에 이렇게 부른다. m 개의 노드를 선택하는 방법은 비교적 단순하다. Degree가 m 을 넘지 않는 임의의 넷 e 를 우선 선택한 후 e 에 속한 모든 노드들이 M 에 포함된다. 그리고 나서 M 에 속한 임의의 노드에 인접한 노드들 중 아직 M 에 속하지 않는 것을 랜덤하게 선택하여 M 에 포함시킨다. M 에 속한 노드의 수가 m 이 될 때까지 이 선택 과정을 반복한다.

알고리즘 RBLS 라인 4에서 M 을 이용하여 '고정된 노드 집합(fixed nodes)' F 를 결정한다. 어떤 노드 v 가 M 에 속하지 않고 즉, $v \notin M$ 이고 M 에 속한 원소에 인

1) 최적 relaxed 배치란 조건 (1)과 조건 (2)를 무시하고 추정 배선 길이만 최소가 되도록 한 배치를 말한다.
2) 배치를 적법화시킨다는 것은 조건 (1)과 조건 (2)가 동시에 만족되도록 셀을 이동하는 과정을 말한다.

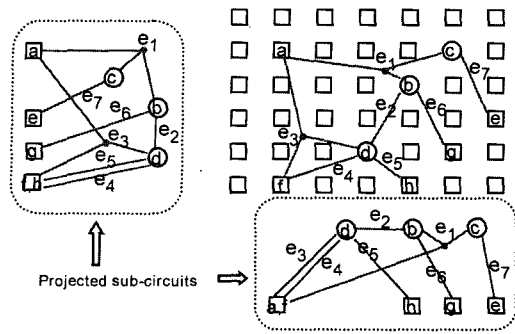


그림 2 x-축과 y-축에 각각 투영된 부분 회로

접할 때 v 는 F 에 포함된다. 부분회로를 $G=(V, E)$ 로 나타내면 $V=M \cup F$ 가 되고 E 는 V 으로부터 유도된 하이퍼에지의 집합이다. 그림 2에서 부분회로를 예로 보았다. 여기서 $M=\{c, b, d\}$ 이고 $F=\{a, e, f, g, h\}$ 이다.

3.2 최적 relaxed 배치(Optimal Relaxed Placement)

부분회로를 추출한 다음 라인 5에서는 M 에 속한 각 원소 v 에 대해 최적의 위치를 결정한다. v 의 최적의 위치 (x, y) 를 결정하기 위해서 우선 추출된 부분회로를 그림 2에서 보듯이 x 축과 y 축에 투영시킨다. 이때 F 에 속한 노드들은 각 축에 그대로 투영되나 M 의 원소들은 임의의 곳으로 움직일 수 있기 때문에 그림 2에서 보듯이 축에서 떨어진 위치에 투영시켰다. 그림 2의 예에서는 노드 b, c, d 가 이동가능 노드로 선택되었고 노드 a, e, f, g, h 는 b, c, d 에 인접해 있으므로 고정된 노드가 된다. 그리고 이렇게 결정된 V 에 의해 유도된 넷 집합 E' 은 $\{e_1, e_2, \dots, e_7\}$ 이 된다. E' 에 속한 각 넷의 길이(HP)를 최소화시키도록 각 이동가능 노드의 위치를 결정하기 위해 우리는 x, y 축에 각각 투영된 부분회로를 이용한다. 예를 들어 각 이동가능 노드의 x 좌표 값을 결정하기 위해서 x 축에 투영된 부분회로를 이용한다. 고정된 노드의 x 좌표는 고정되어 있다고 가정하고 이동가능 노드의 위치를 결정하기 위해 LP(linear programming) 방법을 사용할 수 있다. 이동가능 노드의 위치를 결정하기 위한 LP 식은 다음과 같다.

$$\min \sum_{e \in E'} (r_e - l_e) \text{ subject to}$$

$$l_e \leq x_v \leq r_e, \quad \forall v \in e,$$

$$x_v = X_v, \quad \forall v \in F$$

여기서 X_v 는 고정된 노드의 x 좌표이다. 위 식에 의해 l_e 와 r_e 가 결정되면 그 값을 이용해 x_v 를 쉽게 얻을 수 있다. 노드 v 의 y 좌표도 유사하게 구할 수 있다. Hur와 Lillis[12]는 이 LP 식을 network flow 방법을 이용하여 매우 효과적으로 풀 수 있음을 보였다. 본 논

문에서도 [12]에서 제시한 network flow 기법을 이용하여 이동가능 노드 v 의 (x, y) 좌표 값을 구한다.

3.3 배치 적법화(Placement Legalization)

Relaxation 기반 기법에서 가장 중요한 문제는 어떻게 셀(노드)의 중첩을 해결하는가 하는 점이다. 중첩 해결을 위해 우리는 두 가지 독특한 방법을 제시하였고, 이 방법이 매우 유용함을 실험적으로 증명하였다. 배치 적법화를 위한 전통적인 방법은 relaxed 배치에서 중첩된 위치에 있는 각 셀을 주위에 이동시킴으로 최종적으로 적법한 배치를 얻는 것이다. 이런 방법에 의하면 적법한 배치는 최종적으로 하나만 얻을 수 있다. 우리가 본 논문에서 제안한 방법의 첫 번째 특징은 적법화 과정에서 다수의 적법한 배치를 얻게 되며 그 중 가장 좋은 것을 택하는 것이다. 이 과정은 다음과 같이 요약될 수 있다.

1. 현재의 적법한 배치를 최상의 것으로 간주하여 기억시킴, 즉 P_{best} 를 저장.
2. 알고리즘 RBLIS의 3-5 단계를 실행하여 이동가능한 노드(셀)의 최적 위치를 구함
3. 각 이동가능한 노드 v 에 대해
 - (1) v 를 최적의 relaxed 위치에 배치하여 relaxed 배치 P 를 얻음.
 - (2) 만약 P 가 적법하면 단계 (3)으로 이동. 그렇지 않으면 다음에 설명할 “적법화 과정”을 실행하여 적법한 새로운 배치를 얻음.
 - (3) 만약 새로운 적법 배치가 P_{best} 보다 더 좋으면 (즉, 추정 배선길이의 합이 더 적으면) P_{best} 에 새로운 적법 배치를 기억시킴.

단계 3-(2)에서 언급된 “적법화 과정”의 구체적인 기법이 본 논문에서 제안하는 적법화 기법의 두 번째 특징이다. 중첩해결을 위해 셀을 이동하다 보면 2절에서 설명한 제약조건 두 가지를 위배하는 경우가 발생하게 된다. 제약조건을 위배하지 않기 위해서는 가능하면 셀을 적게 움직여야 한다. 이런 점을 고려하면서 배선길이 또한 최적화시키기 위해서 우리는 다음과 같은 “적법화 과정”을 적용한다.

1. 만약 어떤 빈 S 에 있는 셀이 너무 많아 빈의 용량을 초과했다고 하면 (즉 제약조건 (1)을 위배함), S 주위의 빈 가운데 임의의 한 셀을 더 허용할 수 있는 빈 T 를 찾는다. 그런 빈이 여럿 있으면 S 에서 가장 가까운 빈을 T 로 지정한다. 그리고 S 로부터 T 까지 모노톤 경로(monotone path)를 따라 셀을 ripple move 방법으로 옮긴다.
2. 만약 어떤 빈 T 에 있는 셀이 너무 적어 제약조건 (1)을 위배했다면, T 주위의 빈 가운데 임의의 한 셀을

옮겨도 좋은 빈 S 를 찾는다. 그런 빈이 여럿 있으면 T 에서 가장 가까운 빈을 S 로 지정한다. 그리고 S 로부터 T 까지 모노톤 경로(monotone path)를 따라 셀을 ripple move 방법으로 옮긴다.

3. 만약 제약조건 (2)가 만족되지 않는다면 셀이 초과한 행에서 가장 용량이 초과한 빈 S 를 찾고, 셀이 부족한 행에서 가장 용량이 부족한 빈 T 를 찾아 S 로부터 T 까지 모노톤 경로(monotone path)를 따라 셀을 ripple move 방법으로 옮긴다.

4. 위의 1-3 과정을 제약조건이 만족될 때까지 반복한다.

빈 S 에서 빈 T 로 셀을 이동시킬 때 모노톤 경로를 따라 옮기는데 다양한 모노톤 경로의 후보들 중에서 이득을 고려하여 이득이 최대가 되는 모노톤 경로를 따라 셀을 옮긴다. Ripple move 방법으로 셀을 옮긴다는 것은 다음과 같이 셀을 옮기는 것을 의미한다. 한 셀이 S 로부터 모노톤 경로를 따라 이동하여 이웃 빈에 들어가면 그 이웃 빈에 있는 모든 셀을 고려하여 이득이 최대가 되는 셀을 다시 선택하여 모노톤 경로를 따라 옮긴다. 이런 방식으로 셀을 S 로부터 T 로 옮기면 한 셀만 움직이게 되는 것이 아니라 매번 이웃 빈에 옮길 때마다 다른 셀이 (물론 이득이 가장 큰 셀) 옮겨질 수 있게 된다.

3.4 분할기법을 이용한 배치 최적화(Further Optimization using Partitioning Technique)

본 절에서는 알고리즘 *RBLs*의 단계 7에 대해 설명한다. 광역배치 단계에서 배선 길이를 줄이는 것은 이웃한 두 빈사이의 컷 사이즈를 줄이는 것과 밀접한 관계가 있다. Relaxation과 배치 적법화 과정을 거쳐 얻은 배치는 좀 더 거시적인 관점에서 배치를 개선하게 하지만 이웃한 빈 사이의 컷 사이즈를 줄이는 것은 미시적인 관점에서 배치를 개선하게 한다. 배치를 개선하기 위해서 회로 분할을 위해 사용되는 FM-알고리즘[5]을 이웃한 빈 사이에 적용하여 컷 사이즈를 줄인다.

광역 배치 전체를 위 행부터 아래로, 각 행 내에서는 왼쪽 빈부터 오른쪽으로 스캔하면서 이웃한 두 빈의 현재 컷 사이즈를 초기 분할로 보고 FM-알고리즘을 적용하여 컷 사이즈를 개선한다. 이 최적화 과정은 더 이상 컷 사이즈가 줄어들지 않을 때까지 반복하여 실행된다.

4. 상세 배치(Detailed Placement)

본 절에서는 광역배치를 상세배치로 바꾸는 기법을 우선 설명하고, 상세배치를 최적화시키기 위해 동적프로그래밍 기법을 이용한 최적 인터리빙 기법[13]의 동작 원리에 대해 설명한다. 광역배치에서 각 행에 배치된 셀

은 상세배치에서도 같은 행에 배치하도록 되기 때문에 광역배치에서 제약조건 (2)를 만족하면 상세배치에서도 반드시 그 조건이 만족되게 된다.

4.1 광역배치를 상세배치로 변환

알고리즘 *RBLs*의 실행 후 얻은 최적화된 광역배치는 상세배치로 우선 바뀌게 된다. 이 과정은 단순한 탐욕 알고리즘을 사용하여 이루어진다. 우선 광역배치의 각 행에 위치한 셀은 상세배치에서도 같은 행에 위치하게 된다. 그리고 상세배치에서는 모든 셀이 서로 공간없이 인접해 있다고 가정한다. 광역배치를 상세배치로 바꿀 때, 광역배치에서 빈 $b_{i,j}$ 에 있는 셀들은 $b_{i,j+1}$ 에 있는 셀들보다 상세배치에서 상대적으로 왼쪽에 위치하게 한다. 즉, $b_{i,1}$ 에 있는 셀들을 우선 상세배치의 i 번째 행 왼쪽부터 위치시킨다. 이때 $b_{i,1}$ 에 있는 셀의 연결도를 고려하여 배선 길이가 적게 되도록 각 셀을 배치한다. 그런 다음 빈 $b_{i,2}$ 에 있는 셀들을 상세배치의 i 번째 행에 이어서 배치시킨다. 물론 이 때도 $b_{i,2}$ 에 있는 셀의 연결도를 고려하여 배선 길이가 적게 되도록 각 셀을 배치한다. 빈 $b_{i,j}$ 에 있는 셀들은 각 셀의 연결도에 근거하여 탐욕방법을 이용하여 상대적인 위치를 결정함으로써 상세배치에 위치된다. 이런 과정을 모든 빈에 대해, 각 행에 대해 적용하면 광역배치로부터 유도된 상세배치를 얻게 된다.

4.2 최적 인터리빙

일단 상세배치를 얻은 후 '최적 인터리빙[13]'이라 불리는 기법을 이용하여 각 행 내에 있는 셀들의 위치를 재조정함으로써 상세배치를 개선시킨다. 최적 인터리빙 과정은 다음과 같이 요약될 수 있다.

1. 주어진 윈도우 크기 W 에 대해, 하나의 행 내에 있는 셀들로부터 W 범위 안에서 부분열 A 를 찾아낸다.
2. A 셀들에 포함되지 않은 셀들은 B 에 보관한다. 이때 A 와 B 배열 순서는 현 배치의 상대적 순서를 유지한다.
3. 두 부분 배열 A 와 B 내에 있는 셀들의 상대적 위치를 유지하면서 A 와 B 를 인터리브(interleave) 시킨다.
4. 단계 1~3을 첫째 행부터 마지막 행까지, 그리고 각 행내에서는 윈도우를 우측으로 이동하면서 반복 적용한다.
5. 위의 모든 단계를 배치가 개선되지 않을 때까지 반복한다.

주어진 부분열 A 와 B 가 있을 때 이 두 부분열을 인터리브시켜서 얻을 수 있는 모든 가능한 열 중에서 제안한 방법은 최적의 열을 찾기 때문에 최적 인터리빙 기법이라 불린다. 그림 3에서는 최적 인터리빙을 통해 어떻게 셀의 순서를 바꿀 수 있는지 예를 보여 준다. 최

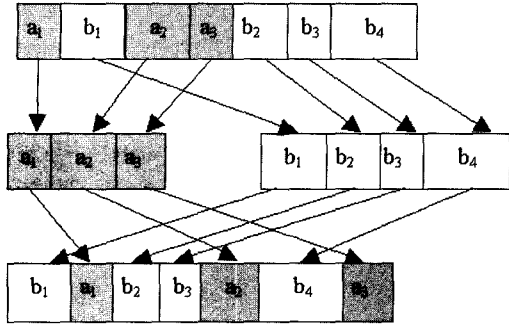


그림 3 최적 인터리빙의 동작 예

적 인터리빙은 동적 프로그래밍 방법을 통해 구현될 수 있음을 [13]에서 보였다.

4.3 동적 클러스터링

최적 인터리빙의 반복 적용 후 상세배치가 더 이상 개선되지 않을 경우 우리는 동적 클러스터링이라 불리는 기법을 사용하여 상세배치를 더 개선시킬 수 있다. 선형 배치의 개선을 위해 [12]에서 동적 클러스터링을 사용하였다. 이 기법을 사용할 경우 지역최적(local optima)을 벗어나는 효과가 있음을 [12]에서 보였고 그 기법을 본 논문에서 그대로 사용하였다. 그 기법의 동작원리는 다음과 같이 요약될 수 있다. 파라미터 L 과 U (여러 개의 셀을 클러스터하여 하나의 슈퍼 셀로 만들 때 L, U 는 한 슈퍼 셀에 포함되는 셀의 상한과 하한값이다. 즉, L 개 이상의 셀과 U 개 이하는 셀로 하나의 슈퍼 셀이 만들어 진다)가 주어질 때, 각 행에 있는 셀들을 왼쪽부터 검사하여 인접한 셀들을 클러스터하여 슈퍼 셀로 만든다. 슈퍼 셀을 구별하는 위치는 [12]에서 처럼 배선밀도를 계산하여 밀도가 가장 낮은 지점에서 분리한다. 클러스터시킨 배치에 또 다시 최적 인터리빙 기법을 적용한다. 클러스터된 배치가 개선된 후 이를 풀어 정상배치로 환원하고, L, U 의 값을 변경시켜 다시 위의 과정을 반복함으로써 배치를 전체적으로 개선시킨다.

5. 하이브리드 기법

그림 4에서는 앞 절에서 설명한 각종 기법을 종합적으로 사용하여 배치를 개선시키는 하이브리드 기법을 보여준다. 우선, 임의의 초기 광역 배치를 얻는다. 그런 다음 알고리즘 $RBLS$ 를 수행시켜 광역배치를 개선시킨다. 알고리즘 $RBLS$ 의 수행을 제어하기 위해 즉, 추출하는 부분회로의 크기와 개선되지 않을 경우 최대 반복 회수를 제어하기 위해 파라미터 m 과 k 를 사용한다. m 의 초기값은 그림 4의 라인 2에서 보듯이 회로의 사이즈에 비례하여 결정되며 그 비례 값은 알고리즘 수행시 파라미터 a 에 의해 결정된다. 광역배치의 개선을 위해

```

Algorithm Hybrid Algorithm for standard cell placement

Input: n, m, a, beta, gamma, i0, kmin, netlist, etc
Output: an optimized detailed placement

/***** Codes for a global placement *****/
1. P ← Obtain an initial placement
2. m ← a · |V|
3. k ← i0
4. while (TRUE) {
5.     Call RBLS(P, m, k)
6.     m ← beta · m
7.     k ← gamma · k
8.     if (m < kmin) break
9. }

/***** Codes for a detailed placement *****/
10. Pd ← Transform P to a detailed placement
11. W ← 2 · (avg. #cells in a bin)
12. while (∃ Improvement) {
13.     Pd ← Opt-interleave(Pd, W)
14.     Determine L and U
15.     Pc ← Cluster(Pd, L, U)
16.     Pc ← Opt-interleave(Pc, W)
17.     Pd ← Flatten(Pc)
18. }

19. return Pd
    
```

그림 4 표준 셀 배치를 위한 하이브리드 기법

알고리즘 $RBLS$ 를 반복 수행하는데 $RBLS$ 에 제공되는 파라미터의 값들은 실행 시간이 경과함에 따라 변경된다.

광역배치가 어느 정도 최적화되면 이를 우선 상세배치로 변환한다(라인 10). 4.2절에서 설명한 최적 인터리빙 기법과 4.3절에서 설명한 동적 클러스터링 기법을 이용하여 상세배치를 개선시킨다. 최적 인터리빙을 적용할 때 윈도우의 크기는 광역배치에서 두 번에 들어 있는 셀 개수의 평균을 계산하여 그 수만큼의 셀이 한 윈도우에 들어 올 수 있도록 결정한다. 클러스터링 기법을 적용할 때 사용될 파라미터 L 과 U 는 다음과 같이 결정된다. L 의 값을 2~5 사이에 있는 임의의 값으로 정하고, $U = 2 \cdot L$ 이 되도록 결정된다.

광역 배치를 위한 그리드 차수(dimension)를 결정하기 위해 입력 파라미터 R 과 C 를 사용한다. R 은 행수를 나타내는데 [14]에 보인 Feng Shui의 결과와 비교하기 위하여 같은 수의 행을 사용한다. 그리고 행간에 공백이 있는 경우와 없는 경우 두 가지 경우에 대해 동일한 행수를 이용하여 실험한다. 열 수를 조정하기 위한 파라미터 C 값은 각 번에 평균적으로 셀이 10~20개 정도 들어 갈 수 있도록 결정된다. 이 값들은 실험에 근거하여 설정된 값이다. 하이브리드 기법에서 광역배치를 위한 while loop (라인 4-8) 내에서 부분회로의 크기는 점진적으로 줄어드는데 그 비율이 $\beta (<1)$ 로 고정되어 있고,

m 의 값이 미리 정한 하한값 k_{min} 보다 적으면 광역배치 개선을 중단한다.

6. 실험 및 고찰

본 논문에서 제안한 표준 셀 배치를 위한 하이브리드 기법은 C 언어로 구현되어 MCNC 벤치마크 회로를 대상으로 Sun Ultra 10/Solaris 상에서 실험하였다. 입력 파라미터 α 는 0.7~0.8로 주어지고, 알고리즘 RBLS의 수행을 제어하기 위해 즉, 추출하는 부분회로의 크기와 개선되지 않을 경우 최대 반복 회수를 제어하기 위한 초기 k 값을 위한 i_0 는 2~3으로 주어진다. β 는 0.1로, ν 는 2, k_{min} 은 회로의 크기에 따라 5~10으로 주어진다. 알고리즘에는 명시되어 있지 않지만 총 CPU 사용 시간의 상한을 제어하기 위한 파라미터 등도 사용될 수 있다. 2절에서 설명한 조건 (2)를 만족시키기 위해 최대 행과 최소 행의 비율, 즉 $\frac{\text{최대행}}{\text{최소행}} < 1.03$ 이 되도록 적법화 과정에서 셀을 이동시킨다. 이렇게 최대행과 최소행의 비율이 일정 범위 내에 있게 함으로써 white space의 비율을 매우 낮출 수 있었다. 다만 primary 회로에 대한 이 조건을 다소 완화시켜 최대행과 최소행의 비율이 1.05까지 허용되었다.

표 1에서는 실험에 사용된 MCNC 벤치마크 회로의 사양을 간략히 보였다. 표 2와 표 3에서는 [14]에서 보인 Feng Shui의 결과와 비교하기 위해 동일한 행수

표 1 회로의 사양

Circuit	# nets	# cells
Prim 1	904	833
Struct	1920	1888
Prim 2	3019	3014
Biomed	5742	6417
Industry 2	13419	12142
Industry 3	21940	15059
Avq small	22124	21854
Avq large	25384	25114

그리고 행간 공백을 사용한 조건하에서 얻은 결과를 보였다.

7. 결론

본 논문에서는 표준 셀 배치를 위한 하이브리드 기법을 제안하였다. 제안한 알고리즘은 두 단계 즉, 광역배치와 상세배치를 거쳐서 배치를 최적화 시킨다. 광역배치에서는 칩의 코어 영역을 $n \times m$ 그리드로 분할한 뒤 각 빈에 셀을 배치하여 초기 배치를 얻은 다음, RBLS 알고리즘을 이용하여 배치를 최적화 시킨다. RBLS 기법은 이미 선형 배치에서 그 효율성이 증명되었고, 이 알고리즘의 동작 원리가 표준 셀 배치에서 응용되었는데 역시 좋은 결과를 보여 주었다. RBLS 알고리즘의 핵심은 추출된 부분 셀들의 최적 위치 구하는 것과 중첩해결 방법이라고 볼 수 있는데, 추출된 부분 셀들의 최적 위치는 네트워크 플로우 기법을 이용하여 해결될

표 2 Feng Shui의 결과와 비교 (With Row Spacing)

Circuit	Feng Shui's Results in [14]				Ours				
	#rows	Row Len	White space	Wire Len	#rows	Row Len	White space	Wire Len	CPU time (sec)
Prim 1	17	5,141	9.0%	1,044,291	17	4,810	1.9%	965,615	34
Struct	21	4,769	1.9%	755,176	21	4,752	1.5%	706,980	63
Prim 2	22	10,501	1.3%	3,781,567	22	10,500	1.3%	3,592,968	200
Biomed	44	10,513	2.7%	3,403,408	44	10,368	1.2%	3,485,918	423
Industry 2	69	15,449	6.5%	15,627,343	69	14,696	1.3%	16,160,831	1,140
Industry 3	52	28,153	3.8%	45,960,343	52	27,528	1.5%	44,687,804	2,328
Avq small	79	9,497	3.7%	5,653,141	79	9,296	1.5%	5,722,467	4,869
Avq large	83	9,777	1.3%	6,210,026	83	9,800	1.5%	6,161,967	5,626

표 3 Feng Shui의 결과와 비교 (Without Row Spacing)

Circuit	Feng Shui's Results in [14]				Ours				
	#rows	Row Len	White space	Wire Len	#rows	Row Len	White space	Wire Len	CPU time (sec)
Prim 1	23	3,610	3.6%	846,822	23	3,580	2.6%	849,603	60
Struct	29	3,472	2.4%	516,236	29	3,440	1.5%	569,646	72
Prim 2	39	6,150	5.2%	3,007,493	39	5,940	1.6%	2,916,407	214
Biomed	62	7,536	3.7%	2,735,042	62	7,376	1.5%	2,725,123	444
Industry 2	98	10,712	4.9%	10,454,680	98	10,352	1.4%	12,364,077	771
Industry 3	74	20,344	6.8%	33,832,792	74	19,304	1.3%	34,726,862	2,952
Avq small	112	6,960	7.7%	4,347,417	112	6,552	1.4%	4,381,825	7,070
Avq large	118	7,648	2.7%	4,921,154	118	6,888	1.4%	4,802,575	8,464

수 있음을 보였다. 중첩해결을 위해선 최대 이득 모노톤 경로를 구한 다음 그 경로를 따라 셀을 ripple move 시킴으로 배선길이의 최적화를 할 수 있었다.

광역배치가 더 이상 개선되지 않을 경우 이를 상세배치로 변환하였는데, 변환 알고리즘은 비교적 단순한 그리디 방법을 사용하였다. 그렇게 한 이유는 변환 후 최적 인터리빙이라 불리는 기법을 통해 상세배치를 최적화시킬 수 있기 때문에 변환 시 사용되는 알고리즘은 단순한 것이어도 무방한 것으로 판단되기 때문이다. 상세배치에서 최적 인터리빙 기법과 동적 클러스터링 기법은 매우 효과적으로 배치를 개선시킴을 실험을 통해 알 수 있었다.

제안한 하이브리드 기법을 이용하여 MCNC 벤치마킹 회로를 대상으로 실험하여 [14]에서 보인 Feng Shui의 결과와 비교하였다. Feng Shui의 결과와 본 논문에서 제시한 방법으로 구한 배치의 길이 거의 비슷함을 보여 주었다. 제안한 기법으로 구한 결과가 지금까지 발표된 가장 좋은 결과보다는 좋지 못하지만 표준 셀 배치기의 첫 버전이라는 점에서 아직 개선될 여지가 많이 있다. 사용된 자료구조를 더 최적화시킬 수 있는 여지가 있으며, 최적 인터리빙도 지금은 2-way만 실험하였으나 3-way 최적 인터리빙을 적용할 경우 상세배치가 더 개선될 것으로 예측된다. 또한 지금 최적 인터리빙은 한 행내에서만 적용 가능하나 인접한 두 행을 동시에 고려한 최적 인터리빙이나 열을 중심으로 한 최적 인터리빙도 앞으로 실험해 볼 예정이다. 또한 본 논문에서 제안한 기법에 몇 가지 제약을 추가한다면 타이밍을 고려한 배치를 구할 수 있을 것으로 생각된다.

참 고 문 헌

- [1] M. Wang and M. Sarrafzadeh, "Behavior of Congestion Minimization During Placement," in Proc. of International Symposium on Physical Design, pp. 145-150, 1999.
- [2] M. A. Breuer, "Min-cut Placement," Design Automation and Fault-Tolerant Computing, pp. 343-382, 1977.
- [3] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Optimal End-Case Partitioners and Placers for Standard-Cell Layout," in Proc. of International Symposium on Physical Design, pp. 90-96, 1999.
- [4] D. J. H. Huang and Andrew B. Kahng, "Partitioning-Based Standard-Cell Global Placement with an Exact Objective," in Proc. of International Symposium on Physical Design, pp. 18-25, 1997.
- [5] C. M. Fiduccia and R. M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitions," in Proc. of ACM/IEEE Design Automation Conference, pp. 175-181, 1982.

- [6] C. Sechen and K. W. Lee, "An Improved Simulated Annealing Algorithm for Row-Based Placement," in Proc. of IEEE International Conference on Computer-Aided Design, pp. 478-481, 1987.
- [7] Wern-Jieh Sun and Carl Sechen, "Efficient and Effective Placement for Very Large Circuits," IEEE Transactions on Computer-Aided Design, pp. 349-359, 1995.
- [8] Sivanarayana Mallela and Lov K. Grover, "Clustering Based Simulated Annealing for Standard Cell Placement," in Proc. of Design Automation Conference, pp. 312-317, 1988.
- [9] Majid Sarrafzadeh and Maogang Wang, "NRG: Global and Detailed Placement," in Proc. of IEEE International Conference on Computer-Aided Design, pp. 532-537, 1997.
- [10] G. Sigl, K. Doll and F. Johannes, "Analytical Placement: A Linear or a Quadratic Objective Function?," in Proc. of ACM/IEEE Design Automation Conference, pp. 427-432, 1991.
- [11] H. Eisenmann and F. M. Johannes, "Generic Global Placement and Floorplanning," in Proc. of ACM/IEEE Design Automation Conference, pp. 296-274, 1998.
- [12] Sung-Woo Hur and John Lillis, "Relaxation and Clustering in a Local Search Framework: Application to Linear Placement," VLSI Design, Vol. 14, No. 2, pp. 143-154, 2002.
- [13] 오은경, 허성우, "동적 프로그래밍 기법을 이용한 효율적인 배치 개선 알고리즘," 한국정보처리학회 추계학술발표대회, 2002.
- [14] Patrick H. Madden, "Reporting of Standard Cell Placement Results," IEEE Trans. of CAD, pp. 240-247, 2002.
- [15] GSRC Bookshelf slot [online] "http://www.gagascale.org/bookshelf."



허 성 우

1981년 경북대학교 전자공학과(학사)
1983년 KAIST 전산학과(석사). 2000년
UIC Dept. of EECS(박사). 1986~ 동아
대학교 전기전자컴퓨터 공학부 교수
2001~ 미국 Intel 사 Physical Design
분야 기술자문 위원. 관심분야는 CAD
알고리즘, 계산 기하학, combinatorial optimization



오 은 경

1997년 동아대학교 컴퓨터 공학과 졸업
(학사). 2000~2003년 동아대학교 컴퓨터
공학과 대학원 졸업(공학 석사). 2003~
현재 동아대학교 컴퓨터 공학과 대학원
(박사과정). 관심분야는 컴퓨터 알고리즘,
운영체제