

배선 밀집도 드리븐 배치

김동현*, 오은경*, 허성우*

*동아대학교 컴퓨터공학과

e-mail: skipio_2000@yahoo.co.kr

(Routing Congestion Driven Placement)

Dong Hyun Kim*, Eun Kyung Oh*, Sung Woo Hur*

*Dept of Computer Engineering, Dong-A University

요 약

VLSI 설계에서 셀 배치단계는 칩의 성능에 영향을 미치는 중요한 단계로서, 셀 배치문제의 주요한 목적비용으로는 배선길이, 타이밍(timing) 그리고 배선밀집도(routing congestion)가 있다. 기존 연구에서 배선길이를 줄이기 위한 많은 기법들이 소개되었으나 배선밀집도를 추정하고 이를 어떻게 줄일 것인가에 대한 연구는 상대적으로 많이 되어있지 않다. 본 논문에서는 셀 배치후에, 주어진 배치를 바탕으로 배선밀집도를 예측하고 배선밀집도가 높은 지역을 국부적으로 해결하는 새로운 기법을 제안한다.

1. 서론

VLSI 기술이 향상됨에 따라 물리설계 단계의 복잡도가 증가되고 특히, 셀 배치는 배선길이(wire length), 성능(performance) 그리고 배선가능성(routability)과 같은 주요한 디자인 변수에 큰 영향을 미친다. 특히 배선가능성을 고려하지 않은 배치는 배선 과정에서 심각한 문제를 야기시키고 결국은 배치 자체를 다시 해야하는 경우도 발생시킨다.

전통적인 배치의 목표는 배선길이를 최소화 시키는 것으로서 이에 대한 연구는 오랫동안 계속되어 왔다. 배선길이만을 최소화하기 위한 배치기법에서는 배선가능성에 대한 고려가 거의 배제되어 있다. 배선길이를 최소화하기 위한 기법은 거시적인 관점에서 볼 때 배선의 밀집도를 줄일 수 있으나 국부적으로 발생하는 높은 배선밀집도를 예방하지는 못하는 문제가 있다.

기존 연구에서 Wang[1-3] 등은 요구/공급 관계를 이용한 배선 모델을 제안하였는데 실험결과에 의하면 배선밀집도만을 고려한 배치는 매우 나쁘다. 그래서 우선 배선길이를 목표로 배치를 구하고, 나중에 배선밀집도를 고려하여 배치를 수정하였다. 배선길이 최소화 작업은 이미 언급한대로 거시적으로는 배선밀집도를 개선시키기 때문이다.

본 논문에서는 [1-3]에서처럼 주어진 상세배치를 바탕으로 배선밀집도를 예측하고 배선밀집도가 높은 지역을 국부적으로 해결하는 새로운 기법을 제안한다. 제안된 기법은 크게 두 가지의 특징을 가지는데 첫째로는 배선밀집도에 기반하여 셀들을 이동시킨 후, Mongrel[4]의 리플 이동(ripple movement) 기법을 사용해서 셀들을 국부적으로 다시 이동시켜 셀의 중첩을 제거한다. 리플 이동 기법을

이용하면 기존 배치를 최대한 유지하면서 셀을 이동시킬 수 있다. 두번째로 셀 이동에 따른 배선밀집도의 변화와 배선길이의 변화를 효과적으로 추적할 수 있도록 증분 자료구조(Incremental data structure)를 사용한다는 점이다. 이것은 모든 넷의 정보를 저장하는 대신 배선밀집도를 개선하는데 효과가 크다고 판단되는 넷들의 정보만을 저장함으로써 실행시간을 효과적으로 단축시켜준다.

2. 용어 정의 및 설명

회로는 하이퍼그래프 $G=(V,E)$ (V 는 셀들의 집합, E 는 넷들의 집합)로 나타낸다. 넷 $e \in E$ 는 2개 이상의 셀들을 연결하며, 연결된 셀들의 부분집합, $e \subseteq V$ 로 나타낸다. 배치(placement)는 직사각형의 칩 영역내에 셀들에게 할당된 위치의 집합으로서, 광역배치(global placement)와 상세배치(detailed placement)가 있다. 광역배치는 $n \times m$ 격자로 나뉘어진 각 영역속에 셀들을 배치하는 것으로 격자의 각 영역을 빈(bin)이라 부른다. 상세배치는 광역배치 이후 각 셀들을 빈내의 구체적인 위치에 배치하는 것이다.

배선밀집도 추정을 위해선 특정 영역을 지나는 배선의 수를 예측해야하는데 이를 위해서 빈 경계선을 이용한다. 즉, 빈의 경계선을 통과하는 배선의 개수를 추정하고, 이를 배선요구량(routing demand)이라고 하며, 배선을 위해 사용가능한 자원을 배선자원용량(routing resource capacity)이라고 부른다.

$b(i,j)$ 는 i 행 j 열에 있는 빈을 나타낸다. 빈 $b(i,j)$ 와 $b(i,j+1)$ 의 경계선을 통과하는 배선은 수평선이며 이에 대응하는 배선요구량과 배선자원용량은 각각 $d_h(i,j)$, $c_h(i,j)$ 로 나타낸다. 마찬가지로 빈 $b(i,j)$ 와 $b(i+1,j)$ 의 경계선을 통과

하는 배선은 수직선이며 이에 대응하는 배선요구량과 배선자원용량은 $d_v(i,j)$, $c_v(i,j)$ 가 된다. 특정 넷 e에 의해 발생하는 배선요구량의 수평 및 수직성분은 각각 $d_h(i,j)$, $c_h(i,j)$ 로 나타낸다. 그러면,

$$d_h(i,j) = \sum_{e \in E} d_{h_e}(i,j), \quad d_v(i,j) = \sum_{e \in E} d_{v_e}(i,j) \text{가 된다.}$$

$b(i,j)$ 에서 수직,수평성분의 배선밀집도는 각각 다음과 같이 정의한다.

$$rc_v(i,j) = d_v(i,j)/c_v(i,j)$$

$$rc_h(i,j) = d_h(i,j)/c_h(i,j)$$

그리고 빈 $b(i,j)$ 의 배선밀집도 $rc(i,j)$ 는 $rc_v(i,j)$ 와 $rc_h(i,j)$ 중 큰 값으로 정의한다. 즉,

$$rc(i,j) = \begin{cases} rc_v(i,j), & \text{if } rc_v(i,j) > rc_h(i,j) \\ rc_h(i,j), & \text{otherwise} \end{cases}$$

가 된다.

3. 배치에서 배선밀집도 추정

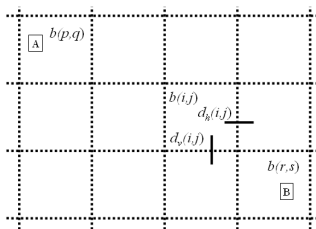
3.1 배선 요구량 계산

그림 1과 같은 배치에서 셀 A,B를 연결하는 넷 $e=\{A,B\}$ 가 있다고 하자. A,B를 연결하는 단조경로(monotone path)는 $\binom{r-p+s-q}{r-p}$ 개이며, 두 빈 $b(p,q)$ 와 $b(r,s)$ 사이에 있는 모든 빈 $b(i,j)$ ($p \leq i \leq r, q \leq j \leq s$)에서 넷 e가 A,B를 연결하기 위해 필요한 수직,수평 배선요구량 $d_v(i,j)$, $d_h(i,j)$ 는 확률에 근거하여 다음과 같이 계산할 수 있다.

e의 경로가 $b(i,j)$ 의 우측 경계선을 통과하기 위해서는 $b(p,q), \dots, b(i,j), b(i,j+1), \dots, b(r,s)$ 를 지나가야 하고, $b(i,j)$ 의 아래쪽 경계선을 통과하려면 $b(p,q), \dots, b(i,j), b(i+1,j), \dots, b(r,s)$ 를 지나가야 한다. 따라서 $d_h(i,j), d_v(i,j)$ 는 다음과 같다.

$$d_h(i,j) = \frac{\binom{i-p+j-q}{i-p} \binom{r-i+s-j-1}{r-i}}{\binom{r-p+s-q}{r-p}}$$

$$d_v(i,j) = \frac{\binom{i-p+j-q}{i-p} \binom{r-i-1+s-j}{r-i-1}}{\binom{r-p+s-q}{r-p}}$$



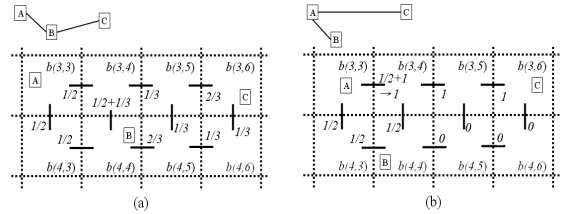
(그림 1) 배선요구량 계산 방법 예

3.2 배선 요구량 추정을 위한 넷 모델링

본 논문에서는 [5]에서처럼 배선을 위해 RMST(Rectilinear Minimum Spanning Tree) 모델을 사용하여 셀들을 연결한다. RMST는 넷에 연결된 셀들의 맨해튼 거리의 합이 최소가 되도록 셀들을 연결한 스패닝트리이다.

그림 2에서 A,B,C 세 개의 셀과 이를 연결하는 넷 $e=\{A,B,C\}$ 를 가진 배치에서 RMST와 그에 따라 계산된 배선요구량의 예를 보였다. 그림 2(a)의 $b(3,4)$ 에서 수직성분의 배선요구량은 두 에지 (A,B), (A,C)에 의해 과생되는 배선요구량의 합을 나타낸다. 그림 2(b)의 $b(3,3)$ 에서 에지 (A,C)에 의해 요구되는 수평배선요구량은 1이고, 에

지 (A,B)에 의해 요구되는 수평배선요구량은 0.5이다. 이 둘을 합하면 $d_h(3,3)=1.5$ 가 되는데 실제 배선에서 한 넷에 의해 과생되는 배선요구량은 1을 넘지 않으므로 1을 초과하는 배선요구량은 그 값을 1로 보정해준다.



(그림 2) 세 개의 셀로 구성된 넷에서 RMST를 구하고, 배선요구량을 계산한 예

4. 배선밀집도 개선을 위한 기법

제안하는 기법은 주어진 배치와 배선자원용량 정보를 바탕으로 배선요구량을 분석한 후, 각 빈의 셀 밀집도와 배선요구량의 변화를 고려하면서 셀을 효과적으로 옮겨 궁극적으로 배선밀집도가 높은 지역의 배선밀집도를 낮추어 준다.

4.1 배치의 평가

일반적으로 배치기(placer)는 배선길이를 배치결과의 평가기준으로 삼지만 본 연구에서는 배선밀집도를 낮추는 것이 목표이기 때문에 배선밀집도와 관련되어 배치를 평가한다. 배선밀집도 평가를 위해 임계값을 정하여 이를 초과하는 경우에 대해서만 고려하는 다음과 같은 평가함수를 이용한다.

$$\delta = rc(i,j) - th \quad (th \text{ 는 임계값})$$

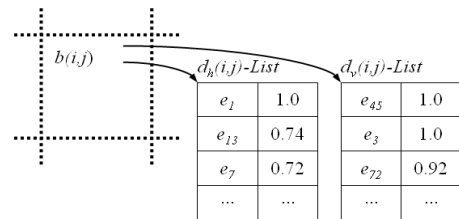
$$cost_{cong} = (1 + \delta)^\alpha \quad (\alpha > 1, \delta > 0 \text{ 인 경우})$$

위에서 α 를 제공하는 이유는 배선밀집도가 임계값보다 클수록 더 많은 가중치를 두기 위함이다. 여기에 배선길어도 함께 고려한 최종 평가함수는 다음과 같다.

$$COST = \omega \times COST_{cong} + (1 - \omega) \times len \quad (len \text{ 은 배선길이, } \omega \text{ 는 외부변수})$$

4.2 제안하는 기법을 위한 자료구조

제안하는 기법의 구현을 위해서는 각 빈에서의 수직/수평 배선요구량을 계산해야한다. 이런 계산을 위해 각 빈은 그림 3과 같은 배선요구량 리스트를 가진다.



(그림 3) 수직/수평 성분의 배선요구량 리스트

그림 3의 $b(i,j)$ 에서 넷 e_1 에 의한 수평배선요구량이 1이고, 넷 e_{45} 에 의한 수직배선요구량이 1임을 나타낸다. 리스트에서 배선요구량은 내림차순으로 정렬되어 배선밀집도에 큰 영향을 미치는 넷들을 빨리 파악할 수 있다. 이 리스트의 내용은 셀이 이동되어 배치가 변형되는 동안에도 동적으로 바뀌면서 유지된다. 또 다른 중요한 자료구조는 그림 4와 같은 빈의 배선밀집도 $rc(i,j)$ 의 값에 따라 내림차순으로 정렬되는 배선밀집도 리스트이다. 이 리스트는

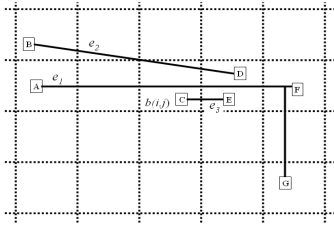
배선밀집도가 가장 큰 빈부터 배선밀집도를 낮추도록 처리하기 위함이다. 배선밀집도 리스트가 구성되면, 리스트의 첫번째 빈부터 배선밀집도를 낮추도록 셀을 옮긴다.

rc	row	col	rc	row	col	rc	row	col	rc	row	col	rc	row	col	...
1.56	12	4	1.54	12	3	1.41	12	7	1.33	5	16	1.32	13	5	

(그림 4) 빈의 배선밀집도 rc(i,j)에 따라 정렬된 리스트

4.3 이동할 셀 선택

배선밀집도 개선을 위해서는 셀을 이동해야하는데 이때 고려해야할 것은 기존 배치를 최대한 유지해야 한다는 점이다. 기존의 주어진 배치는 배선길이에 따라 최적으로 수립된 배치이기 때문이다. 셀 이동은 현재 빈에서 이웃한 8개의 빈 가운데 하나로 이동한다.



(그림 5) b(i,j)의 수평 배선밀집도에 영향을 미치는 넷을 보인 예

그림 5의 b(i,j)의 배선밀집도 rc(i,j)가 임계값보다 크고 $rc_h(i,j) > rc_v(i,j)$ 라고 가정하자. 그러면 b(i,j)의 수평 배선밀집도가 수직 배선밀집도보다 크므로 rc(i,j)를 낮추기 위해서는 수평 배선밀집도를 개선해야한다. 먼저 수평 배선밀집도에 큰 영향을 미치는 넷을 찾아야하는데 4.2절에서 설명한 수평 배선요구량 리스트를 이용하면 쉽게 찾을 수 있다. 그림 5의 경우, 셀 G를 제외한 나머지 셀들을 위나 아래 방향으로 옮기면 수평 배선밀집도가 개선되는 것을 알 수 있다. 만약 셀 B를 우측 빈 b(i-1,j-1)로 옮기면 넷 e_2 에 의해 b(i,j)에서 생기는 수평 배선요구량은 0.75에서 0.66으로 줄어든다. 또한 셀 D를 왼쪽 빈으로 옮기는 것도 밀집도를 줄일 수 있다.

하지만 실제 배치는 그림 5처럼 단순하지 않다. 셀 B가 다른 넷에 연결되어 있을 경우에 B를 우측으로 옮기면 그 넷의 길이가 증가하고, 그 넷으로 인해 다른 빈의 배선밀집도가 높아질 수 있다. 또한 셀을 움직일 때 목적지 빈의 셀 밀집도를 고려해야한다. 빈 b(i-1,j-1)에 이미 많은 셀이 있어 셀 B가 이동되어 올 경우 셀 밀집도가 1을 초과한다면 셀 B는 해당 빈으로 옮길 수 없다. 빈의 셀 밀집도가 1을 초과한다는 말은 그 빈 내에서 셀들이 중첩된 상황임을 의미하고 이를 해결하기 위해선 어떤 셀을 다시 다른 빈으로 옮겨야 하는데 이런 이동들이 수립된 기존 배치의 변경을 가져오기 때문이다.

지금까지 설명한 것을 아래 CellSelection함수에서 의사 코드로 보였다.

Function <i>CellSelection</i>
Input: i,j // bin index of a congested bin
Output: cell name and direction
// The bin is assumed being congested horizontally
max_gain = 0
for every $e \in d_h(i,j) - List$ {
for every cell $C \in e$ {
r,c ← index of a bin which cell C belongs

```

to
// if r is far from row i, ignore this case
if (r > i+1 || r < i-1) continue
for every 8 neighbor bins of b(r,c) {
// Each neighbor bin can be a candidate
for target bin
if(cell density of target bin after
moving > 1) {
continue
}
// Assume the target bin's index is
(t_r, t_c)
gain = ComputeGain(C,r,c,t_r,t_c)
if (gain > max_gain) {
max_gain = gain
save all necessary information
regarding max_gain
}
} // neighbor bins
} // every cell c ∈ e
} // every e ∈ d_h(i,j) - List
return information of cell name, target bin
regarding max_gain

```

4.4 배선밀집도 개선을 위해 제안하는 기법

지금까지 설명한 내용들을 종합하여 본 절에서는 배치에서 배선밀집도를 개선하기 위한 기법을 제안한다. 제안하는 기법은 CDP(Congestion Driven Placer)라 부르며, 다음과 같이 기술될 수 있다.

<i>procedure</i> CDP (Congestion Driven Placer)
Input: P: a detailed placement
k: // max num of iteration
w: // weight for congestion gain
th: // threshold value of congestion
Output: new placement P
1. counter = k
2. $s = \Phi$ // S: a set of congested bins
3. // Consider input placement the best one
4. while (counter > 0) {
5. // s contains a set of congested bins
6. EvaluateCongestion(P, s, th)
7. Sort s in non-increasing order according to congestion value
8. for each bin $b \in s$ {
9. // Assume b(i,j) is congested horizontally
10. mv_cell, mv_direction ← CellSelection(i,j)
11. Move 'mv_cell' along 'mv_direction' to a target bin
12. $P' \leftarrow$ New placement after moving a cell
13. } // for each bin $b \in s$
14. for each bin b(i,j) {
15. Move cells in b(i,j) within the bin to resolve overlap
16. }
17. $P' \leftarrow$ New placement after finishing step 14~16
18. // Now compare the new placement with the so-far best placement
19. if (IsImproved(P,P',th,w) > 0) {
20. $P = P'$
21. counter = k
22. } else {
23. counter = counter - 1;
24. }
25. } // while (counter > 0)
26. return P

단계 4-25에 걸친 while 문은 국부 탐색 기법을 적용한 것이다. 먼저 단계 6에서 현재 배치 P를 평가해서 각 bin에서의 배선요구량 리스트를 구하고 단계 7에서 앞에서 설명한 bin 배선밀집도 리스트를 내림차순으로 정렬한다. 단계 8-13은 배선밀집도가 높은 bin들을 순차적으로 찾아서 배선밀집도를 낮추기 위해 셀들을 이동시켜 새로운 배치 P'을 구하는 과정이다. 단계 14-16은 이동된 셀들에 의해 오버랩이 생긴 경우, 해당 bin 내에서 앞서 설명한 Mongrel의 리플 이동 기법을 이용해서 오버랩을 제거하는 과정이다. 마지막으로 단계 19에서 새로 구해진 배치 P'와 기존의 배치 P를 비교하여 P를 P'으로 대체할 것인지 새로운 배치를 계속 얻어갈 것인지를 결정한다.

5. 실험 결과 및 분석

CDP는 C로 구현되었고, 실험은 IntelXeon 3.6G 8G/linux 상에서 하였다. 실험에 사용된 회로는 반도체 제작회사로 잘 알려진 I사에서 최근에 개발된 마이크로 프로세서로부터 얻은 회로 중 8개를 사용하였다. 각 회로의 특징은 표 1과 같다.

표 2는 실험을 통해 가장 적절하다고 판단된 파라미터의 값을 이용하여 얻은 결과를 보여준다. 이 표에서 $cost_{cong}$ 값을 계산하기 위해 지수값으로 사용되는 α 는 3으로 고정시켰다. $w=0.5$ 로 둔다는 말은 배선길이 개선과 배선밀집도 개선에 동일한 가중치를 둔다는 것을 의미하며, $th=0.85$ 로 둔다는 말은 bin의 배선밀집도가 임계값 0.85이상이면 그 bin의 배선밀집도를 낮추기 위해 셀을 옮긴다는 것을 의미한다. number of bins 열은 배선밀집도가 특정 값 사이에 있는 bin의 개수가 어떻게 변화되는지 보여준다. 예를 들어, 표에서 number of bins의 0.8열에 있는 것은 bin의 배선밀집도가 0.8이상 0.9미만인 bin의 개수를 보여준다.

<표 1> 실험에 사용된 회로의 특성

Test circuits	Number of cells	Number of nets
Ckt 1	8,214	9,338
Ckt 2	15,334	16,295
Ckt 3	16,579	20,859
Ckt 4	17,200	19,865
Ckt 5	18,732	21,431
Ckt 6	19,179	24,057
Ckt 7	19,575	19,867
Ckt 8	21,800	24,538

표 2를 보면 각 회로에서 CDP를 적용하기 전 배선밀집도가 0.9이상인 bin의 개수가 CDP 적용후 현저하게 줄어드는 것을 관찰할 수 있다. 반면 배선밀집도가 0.9미만인 bin의 수는 CDP 적용후 늘어나는데 이것은 배선밀집도를 분산시키는 과정에서 일어나는 자연스러운 현상이다.

표 2에서 배선길이의 변화를 보면 CDP 적용후 배선길이가 오히려 줄어드는 것을 볼 수 있는데 이는 배선밀집도를 해결하면서 또한 배선길이를 함께 고려하도록 $w=0.5$ 로 두었기 때문이다. 이처럼 배선밀집도 개선과 더불어 배선길이 또한 개선할 수 있다는 점은 제안한 기법이 매우 우수함을 보여주는 예라고 할 수 있겠다.

<표 2> $k=20, w=0.5, th=0.85$ 로 두었을 때의 결과 (#bins: bin의 배선밀집도가 주어진 값보다 큰 bin의 수)

Test Ckts	CDP	wire length (meter)	number of bins					CPU time
			0.8	0.9	1.0	1.1	1.2	
Ckt1	Before	0.36538	152	91	64	28	22	
	After	0.36300	190	15	3	0	0	2m15s
Ckt2	Before	0.78416	133	56	29	19	8	
	After	0.77064	151	19	7	2	0	4m13s
Ckt3	Before	1.1567	345	184	88	28	8	
	After	1.1467	350	19	4	0	0	9m9s
Ckt4	Before	1.0699	19	13	0	0	0	
	After	1.0699	18	0	0	0	0	32s
Ckt5	Before	1.0417	23	15	9	8	2	
	After	1.0342	39	3	4	0	0	82s
Ckt6	Before	1.2648	85	31	9	2	0	
	After	1.2595	77	0	0	0	0	2m34s
Ckt7	Before	1.0966	260	182	86	32	30	
	After	1.0792	358	88	45	10	4	33m15s
Ckt8	Before	1.1195	284	161	90	48	31	
	After	1.0973	376	7	0	0	0	11m38s

6. 결론

기존의 배치 알고리즘들이 주로 배선길이 감소를 목표로 하였다면, 본 논문에서는 배선길이와 배선밀집도에 기초한 배선가능성을 고려한 알고리즘을 제안하였다. 앞서 언급했듯이 배선길이를 목표로 하는 연구에 비해 배선가능성을 목표로 하는 연구는 상대적으로 많이 되어있지않다. 따라서 앞으로 배선가능성을 목표로 하는 연구가 상대적으로 많이 요구될 것이다.

본 논문에서는 넷 모델로서 RMST 방식을 사용하였는데 향후 다양한 넷 모델을 적용한 실험 결과를 도출할 예정이다.

7. 참고문헌

- [1] Maogang Wang, Xiaojian Yang, Majid Sarrafzadeh, "Congestion Minimization During Placement", IEEE Trans. CAD of Integrated Circuits and Systems, Vol. 19, No.10, pp. 1140-1148, 2000.
- [2] M.Wang and M. Sarrafzadeh, "Modeling and Minimization of Routing Congestion", Proc. of ASP-DAC, pp. 185-190, 2000.
- [3] M. Wang and M. Sarrafzadeh, "On the Behavior of Congestion Minimization During Placement", Proc. of International Symposium on Physical Design, pp. 145-150, 1999.
- [4] S. Hur and J. Lilis, "Mongrel: Hybrid Techniques for Standard Cell Placement", Proc. of ICCAD, pp. 165-170, 2000.
- [5] Andrew B. Kahng and Xu Xu, "Accurate Pseudo-Constructive Wirelength and Congestion Estimation", ACM International Workshop on System-Level Interconnect Prediction, pp. 61-68, 2003.