

효과적인 배선밀집도 드리븐 배치

오은경^o 허성우

동아대학교 컴퓨터공학과

ekoh^o@donga.ac.kr, swhur@dau.ac.kr

Effective Congestion Driven Placement

Eunkyung Oh^o Sungwoo Hur

Department of Computer Engineering Dong-A University

요 약

본 논문에서는 기존의 CDP(Congestion Driven Placement) 기법[5]을 개선하여 좀 더 빠른 시간에 배선 밀집도를 효과적으로 해결할 수 있는 기법, 즉 ECDP(Effective Congestion Driven Placement) 기법을 제안한다. ECDP는 force-directed 기법을 이용한 배선밀집도 해결 기법으로써 기존의 국부적인 셀 이동 방식을 개선한다. 제시된 기법은 배치 영역의 빈(bin) 상에 셀 밀집도를 해결하는 방향으로 분산력을 구한 후 밀집된 셀을 분산시킴으로써 배선밀집도를 개선시킨다. 실험을 통한 결과에서 CDP와 비교 해 볼 때 배선밀집도와 수행속도 모두 제안기법이 우수한 결과를 보인다.

1. 서 론

VLSI 기술의 물리 설계는 칩의 성능에 결정적인 영향을 미치는 중요한 단계로서 일반적으로 배선길이(wire length), 성능(performance) 그리고 배선가능성(routability)과 같은 목적함수를 개선하는 목적으로 수행된다.

전통적인 배치의 목표는 컷 사이즈를 줄이거나 배선 길이를 최소화 하는 것이다. 최소-컷을 찾는 기법을 사용하는 배치 툴로는 Cap[1], Quad[2], FengShui[3] 등[4]이 있다. 배치 길이를 최소화하기 위해 simulated annealing 기법을 사용하는 툴로는 TimeberWolf[5]가 잘 알려져 있으며, 해석적 기법 또는 FD(Forced - Directed) 법을 사용하는 툴들이 있는데, FD 방법을 사용하는 기법은 그 변화가 매우 다양하다.

배선 밀집도를 줄이는 것을 목표로 하는 연구는 배선 길이를 줄이고자 하는 연구에 비해 상대적으로 많이 되어 있지 않다. Meixner[6]이 제안한 다단계 분할을 이용하는 방법은 미리 구성된 Steiner 트리로부터 실제로 계산된 배선 밀집도를 사용한다. Wang[7,8,9] 등이 요구/공급 관계를 이용한 배선 모델을 제안 했으며, [11]는 Mongrel[10]의 리플 이동(ripple movement) 기법과 증분 자료구조(incremental data structure) 이용하여 기존의 배치를 최대한 유지하면서 배선밀집 문제를 효율적으로 해결하였다.

본 논문에서는 이들 목적함수 중 배선가능성에 초점을 둔다. 배선가능성을 고려한 배치는 배선밀집도를 줄이는데 목적이 있으며 배치단계 후 배선 가능한 배치를 얻기 위함이다.

본 논문에서는 [11]를 수정, 보완한 새로운 배선밀집도 기반 배치기법을 제안한다. 제안 기법의 특징은 기존의 [11]에서 사용한 리플 이동 기법과 증분 자료구조를 그대로 유지하므로 주어진 상세배치의 배선길이를 크게 희생하지 않으면서 효율적으로 배선밀집도를 해결한다. 또한 배선밀집도를 줄이기 위한 셀의 선택 및 이동시 Kraftwerk[12]에서 제안한 force-directed 기법을 통해 기존의 방법보다 우수한 배치결과를 얻는다. 이 배치결과는 셀 밀집도에 근거해 분산력 벡터를 계산하고 셀 밀집문제를 해결함으로써 배선밀집도를 개선한 것이다.

2. 본 론

2.1 용어 정의

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 영역 내에 셀들에게 할당된 위치의 집합으로서, 광역배치(global placement)와 상세배치(detailed placement)가 있다. 광역배치는 $n \times m$ 격자로 나누어진 각 영역 속에 셀들을 배치하는 것으로 격자의 각 영역을 빈(bin)이라 부른다. 상세배치는 광역배치 이후 각 셀들을 빈 내의 구체적인 위치에 배치하는 것이다.

배선밀집도 추정을 위해선 특정 영역을 지나는 배선의 수를 예측해야 하는데 이를 위해서 빈 경계선을 이용한다. 즉, 빈의 경계선을 통과하는 배선의 개수를 추정하

고, 이를 배선요구량(routing demand)이라고 하며, 배선을 위해 사용가능한 자원을 배선자원용량(routing resource capacity)이라고 부른다.

그림 1에서 배선자원요구량과 배선자원용량에 관한 개요를 보인다. $b(i, j)$ 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 $dh(i, j)$, $ch(i, j)$ 로 나타낸다. 마찬가지로 빈 $b(i, j)$ 와 $b(i+1, j)$ 의 경계선을 통과하는 배선은 수직선이며 이에 대응하는 배선요구량과 배선자원용량은 $dv(i, j)$, $cv(i, j)$ 가 된다. 특정 넷 e 에 의해 발생하는 배선요구량의 수평 및 수직성분은 각각 $d_h(i, j)$, $c_h(i, j)$ 로 나타낸다. 이 때,

$$d_h(i, j) = \sum_{e \in E} d_{h_e}(i, j) \tag{1}$$

$$d_v(i, j) = \sum_{e \in E} d_{v_e}(i, j) \tag{2}$$

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

$$rc_v(i, j) = d_v(i, j) / c_v(i, j) \tag{3}$$

$$rc_h(i, j) = d_h(i, j) / c_h(i, j) \tag{4}$$

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,

$$rc(i, j) = \begin{cases} rc_v(i, j), & \text{if } rc_v(i, j) > rc_h(i, j) \\ rc_h(i, j), & \text{otherwise} \end{cases} \tag{5}$$

가 된다.

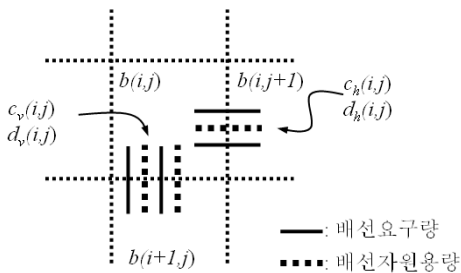


그림 1. $b(i, j)$ 의 배선요구량 및 배선자원용량

2.2 배선밀집도 추정과 배선요구량 계산

그림 2와 같은 배치에서 셀 A, B 를 연결하는 넷 $e = \{A, B\}$ 가 있다고 하자. A, B 를 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 e 가 A, B 를 연결하기 위해 필요한 수직, 수평 배선요구량 $d_{v_e}(i, j)$, $d_{h_e}(i, j)$ 는 확률에 근거하여 다음과 같이 계산할 수 있다.

e 의 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 지나야 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 $d_{h_e}(i, j)$, $d_{v_e}(i, j)$ 는 다음과 같다.

$$d_{h_e}(i, j) = \frac{\binom{i-p+j-q}{i-p} \binom{r-i+s-j-1}{r-i}}{\binom{r-p+s-q}{r-p}} \tag{6}$$

$$d_{v_e}(i, j) = \frac{\binom{i-p+j-q}{i-p} \binom{r-i-1+s-j}{r-i-1}}{\binom{r-p+s-q}{r-p}} \tag{7}$$

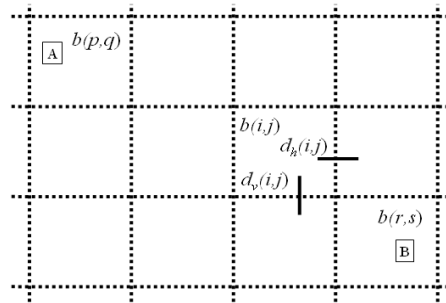


그림 2 배선요구량 계산 방법을 보이는 예

2.3 배선 요구량 추정을 위한 넷 모델

본 논문에서는 [11]에서처럼 배선을 위해 RMST (Rectilinear Minimum Spanning Tree) 모델을 사용하여 셀들을 연결한다. RMST는 넷에 연결된 셀들의 맨해튼 거리의 합이 최소가 되도록 셀들을 연결한 신장 트리(spanning tree)이다.

그림 3에서 A, B, C 세 개의 셀과 이를 연결하는 넷 $e = \{A, B, C\}$ 를 가진 배치에서 RMST와 그에 따라 계산된 배선요구량의 예를 보인다. 그림 3(a)의 $b(3,4)$ 에서 수직성분의 배선요구량은 두 간선 (A, B) , (A, C) 에 의해 파생되는 배선요구량의 합을 나타낸다. 그림 3(b)의 $b(3,3)$ 에서 간선 (A, C) 에 의해 요구되는 수평배선요구량은 1이고, 간선 (A, B) 에 의해 요구되는 수평배선요구량은 0.5이다. 이 둘을 합하면 $d_{h_e}(3,3) = 1.5$ 가 되는데 실제 배선에서 한 넷에 의해 파생되는 배선요구량은 1을 넘지

않으므로 1을 초과하는 배선요구량은 그 값을 1로 보정한다.

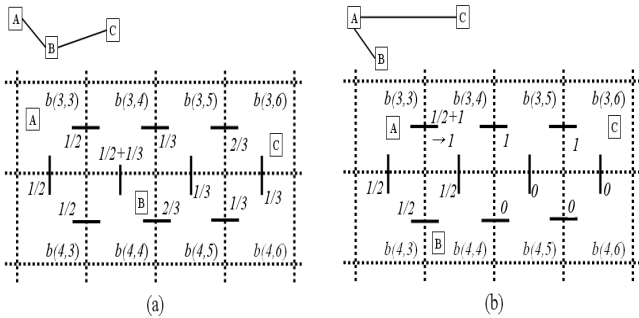


그림 3. 세 개의 셀로 구성된 넷에서 RMST를 구하고, 배선요구량을 계산한 예

3. 제안 기법

배선밀집도를 줄이기 위해 [11]에서 제안한 기법은 배선밀집도에 영향을 미치는 넷을 결정하고 이 넷에 속한 셀들 각각을 주변의 8개 빈으로 하나씩 이동하면서 배선밀집도의 변화를 관찰한 후 배선밀집도를 낮추는 빈으로 셀들을 이동한다.

본 논문에서는 셀들의 이동 위치를 결정하기 위해 배선밀집도에 영향을 미치는 셀을 모두 선택한 후 Kraftwerk[12]에서 제안한 force-directed 기법을 이용하여 셀의 최적 위치를 결정한다.

3.1 Force-directed 기법

Kraftwerk[12]는 셀을 정점(vertex)으로, 넷은 간선(edge)들의 집합으로 표현하여 회로를 그래프로 모델링한다. 이 때 비용(cost)은 넷에 관계된 모든 간선들의 비용 합으로 표현되고, 한 간선의 비용은 그 간선에 있는 두 정점 사이 거리제곱으로 모델링되며, 목적함수(objective function)는 모든 넷의 비용 합으로 표시된다. 고정된 패드(pad)를 제외한 셀 즉, 움직일 수 있는 셀이 n 개 일 때, 셀 $i(1 \leq i \leq n)$ 의 좌표는 셀의 중심의 좌표를 의미하며, (x_i, y_i) 로 나타낼 수 있다. 두 셀 i 와 j 사이에 있는 간선의 x -성분에 대한 비용은 $(x_i - x_j)^2 = x_i^2 + 2x_i x_j + x_j^2$ 이고, y -성분에 대한 비용도 유사하게 표현된다. 모든 넷을 고려한 비용의 합 즉, 목적함수는 [12]에서처럼 다음과 같이 식 (8)로 나타낼 수 있다.

$$\Phi = \sum_{i,j} a_{ij} (x_i - x_j)^2 + \sum_{i,j} a_{ij} (y_i - y_j)^2 \quad (8)$$

여기서 $a_{ij} \in \{0,1\}$ 로서 셀 i 와 셀 j 사이에 간선이 존재하면 1이고 그렇지 않으면 0이 된다. 식 (8)은 행렬과 벡터의 곱으로 표현된 선형방정식으로 나타낼 수 있고, 이를 식 (9)에서 보인다.

$$\Phi = \frac{1}{2} \vec{p}^T C \vec{p} + \vec{d}^T \vec{p} + const \quad (9)$$

이때, C 는 $2n \times 2n$ 인 Laplacian 행렬이고, $\vec{p} = (x_1, \dots, x_n, y_1, \dots, y_n)^T$, $\vec{d} = (d_1, \dots, d_{2n})^T$ 이다. 벡터 \vec{d} 의 각 원소 d_i 는 $1 \leq i \leq n$ 인 경우 셀 i 에 직접 또는 간접적으로 연결된 패드의 x -좌표에 의해 결정된 상수이고, $n+1 \leq i \leq 2n$ 인 경우 패드의 y -좌표에 의해 결정된 상수이다. $const$ 는 패드의 위치로부터 결정된 상수이다. 목적함수 Φ 를 최소로 만들기 위해 $\nabla \Phi = 0$ 를 풀면 되는데 이는 곧 식 (10)으로 표현된다.

$$C \vec{p} + \vec{d} = 0 \quad (10)$$

식 (10)을 풀어 벡터 \vec{p} 의 각 원소의 값을 결정한다는 것은 Φ 값을 최소로 하는 셀의 위치를 구하는 것을 의미한다. 식 (10)을 이용하여 셀의 위치를 구할 때 문제점은 많은 셀들이 중첩된다는 점이다. 이를 해결하기 위해 Kraftwerk에서는 중첩의 정도 즉, 배치 영역에서 셀 밀집도 분포로부터 유도된 벡터 \vec{e} 를 추가한 식 (11)을 이용한다.

$$C \vec{p} + \vec{d}^T + \vec{e} = 0 \quad (11)$$

벡터 \vec{e} 를 분산력 벡터라 하며, 이는 높은 셀 밀집도를 가진 영역에서 낮은 셀 밀집도를 가진 영역으로 셀을 옮기기 위해 사용한다.

3.2 배선밀집도 반영을 위한 분산력 벡터

식(11) 기술한 분산력 벡터 \vec{e} 는 다음과 같다.

$$\vec{f}(x,y) = \frac{k}{2\pi} \cdot \frac{H}{n} \cdot \frac{W}{m} \cdot \sum_{y'=0}^{n-1} \sum_{x'=0}^{m-1} D(x',y') \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^2} \quad (12)$$

이때, $n \times m$ 은 배치영역의 격자 수이고, W 와 H 는 배치 영역의 너비와 높이이다. 이때 $D(x',y')$ 은 셀 밀집도를 반영한다.

3.3 평가함수

셀 밀집도에 기반한 분산력 벡터를 통해 셀의 목적지가 결정되면 셀의 이동에 대한 배선밀집도 이득을 다음과 같은 식으로 평가한다.

$$\delta = rc(i,j) - th \quad (th: \text{임계값}) \quad (13)$$

$$\text{cost}_{con} = (1 + \delta)^a \quad (a > 1, \delta > 0 \text{ 인 경우}) \quad (14)$$

$$\text{cost} = w * \text{cost}_{con} + (1 - w) * \text{length} \quad (15)$$

식(13)은 빈 $b(i,j)$ 의 배선밀집에서 임계값을 뺀 값이다. 식 (14)에서 배선밀집도가 임계값 보다 클수록 더 많은 가중치를 부여하기 위해 a 를 제곱해준다. 식 (15)에서 $length$ 는 배선길이, w 는 외부변수이며 배선밀집도와 배선길이를 모두 고려한 평가함수이다.

3.4 배선밀집도 리스트

본 논문에서는 각 빈에서의 수직/수평 배선요구량을 효율적으로 관리하게 위해 그림 4와 같이 각 빈의 배선밀집도 정보를 가지는 배선밀집도 리스트를 사용한다. 이 자료구조는 [11]에서와 같이 배선밀집이 가장 높은 빈으로부터 밀집문제를 해결하기 위함이다.

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

3.5 배선밀집도 개선

[11]에서 제안한 기법은 배선밀집도 리스트에서 배선밀집도에 가장 큰 영향을 미치는 넷을 선택한 후, 이 넷에 속한 셀을 이웃 8개 빈으로 하나씩 이동하면서 배선밀집도 이득을 고려한다. 이 중 가장 이득이 큰 빈으로 셀을 하나씩 이동시킨다.

본 논문에서는 배선밀집에 영향을 미치는 셀을 이웃한 8개 빈으로 하나씩 이동하는 방식을 사용하지 않고 영향력을 미치는 넷에 속한 셀을 모두 이동 가능한 셀 (mobile-cell)로 간주하고 force-directed 기법을 통해 최적 위치를 한 번에 계산한다. 배선밀집도 개선 기법은 다음과 같다.

배선밀집도 리스트로부터 배선밀집도에 영향을 미치는 셀이 속한 넷을 선택 한다. 해당 넷에 속한 셀들을 이동 가능한 셀 (mobile-cell)로 간주하고 셀들의 밀집도를 고려하여 밀집도를 분산시키는 방향으로 셀들이 이동할 수 있도록 분산력 벡터를 계산한다. 이때 [13]에서와 같이 배치영역을 성근격자(coarse grid)와 미세격자(fine grid)로 구분하여 성근격자 상에서 셀의 밀집도를 고려한 분산력을 구한다. 성근격자와 미세격자의 관계를 그림 5에

보인다. 그림 5에서 굵은 선은 성근격자를 나타내고 점선은 미세격자를 나타낸다.

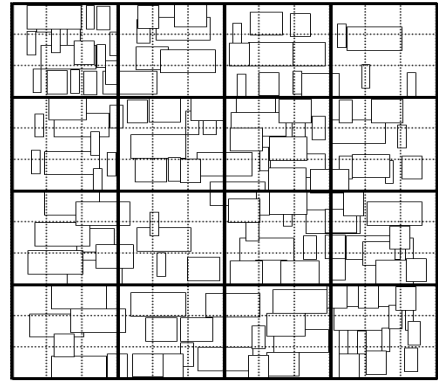
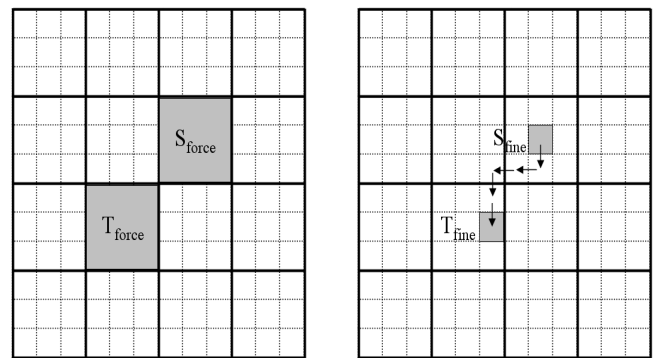


그림 5. 성근격자와 미세격자

분산력은 성근격자의 밀집된 셀을 분산시키는 작용을 하며 배선밀집도에 영향을 미치는 넷에 속한 셀들은 분산력에 의해 다른 성근격자로 분산된다. 이때, 성근격자에서 출발지 성근격자와 목적지 성근격자를 각각 S_{force} 와 T_{force} 라 한다. 또한 S_{force} 내에 미세격자 중 가장 밀집도가 높은 빈이 S_{fine} 으로, T_{force} 내에 미세격자 중 가장 밀집도가 낮은 빈이 T_{fine} 으로 결정된다. S_{fine} 과 T_{fine} 이 결정되면 미세격자 상에서 최대 이득 경로를 구하여 가장 이득이 큰 경로의 위치로 셀을 이동시킨다. 이것을 그림 6에 나타내었다.



(a) 힘-격자에서 결정된 S_{force} 와 T_{force} (b) 미세격자에서 결정된 S_{fine} 과 T_{fine} 그리고 최대 이득 단조경로

그림 6. 성근격자와 미세격자를 이용한 셀의 이동

배선밀집도를 개선하기 위해 [13]에서 제안한 셀 밀집도에 기반한 분산력 벡터를 사용한 이유는 궁극적으로 셀 밀집문제를 해결하는 것이 배선밀집도를 개선하는 것임을 보이기 위함이다.

3.6 개선된 기법

본 논문에서 제안한 기법의 구현을 ECDP(Effective

Congestion Driven Placer)라 명명하고 다음과 같이 기술할 수 있다.

단계 6에서 배선 밀집도에 따른 배선 요구량 리스트를 구한 뒤 빈 배선밀집도 리스트를 내림차순으로 정렬한다. 배선밀집도가 높은 빈들을 순차적으로 찾아 배선 밀집도를 낮추기 위해 단계10 ~ 12에서 빈에 속한 넷 e에 대해 eval_FD(e)를 수행한다. eval_FD(e)은 3.1, 3.2, 3.3에서 설명한 계산식을 이용하여 넷에 속한 모든 셀들의 목적 빈 즉 밀집도가 낮은 빈을 동시에 찾아준다. 목적지 빈으로 셀들을 이동 시켜 새로운 배치 P'를 구한다. P'의 오버랩을 해결하기 위해 단계 17~ 19가 수행된다. 단계 22~27 이 새로운 배치 p'를 선택할 것인가를 결정한다.

```

procedure ECDP (Effective 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. for every e $\in$ b(i,j){
11. eval_FD(e)
12. //eval_FD(e) evaluates the target bin of cell $\in$ e at one time
13. mv_cell, mv_direction
14. Move 'mv_cell' along 'mv_direction' to a target bin
15. P'  $\leftarrow$  New placement after moving a cell
    } // for every e $\in$ b(i,j)
16. } // for each bin b  $\in$  S
17. for each bin b(i,j) {
18. Move cells in b(i,j) within the bin to resolve overlap
19. }
20. P'  $\leftarrow$  New placement after finishing step 14~16
21. // Now compare the new placement with the so-far best placement
22. if( IsImproved(P,P',th,w) > 0) {
23. P = P'
24. counter = k
25. } else {
26. counter = counter - 1;
27. }
28.} // while (counter > 0)
29.return P
    
```

4. 실험 및 고찰

ECDP는 C언어로 구현되었으며 팬티엄4 2G/리눅스 상에서 실험을 수행하였다. 실험은 A사에서 개발한 5개의 회로에 대하여 수행하였으며, 표 1에 회로의 사양을 보인다.

표 1. 회로 정보

회로 이름	셀의 개수	넷의 개수
ckt1	5,592	6,016
ckt2	8,151	8,720
ckt3	9,023	10,757
ckt4	10,117	12,015
ckt5	22,406	23,557

표 2는 [11]에서 제안한 CDP(Congestion Driven Placer)와 본 논문에서 제안한 ECDP를 각 회로에 대해 비교 실험한 결과이다. th는 임계값으로써 예를 들어 th = 0.7인 경우, 특정 빈의 밀집도가 임계값 th = 0.7이상 일 때 그 빈의 밀집도를 낮추기 위해 셀을 이동시킴을 의미한다. 또한 빈의 수는 배선밀집도가 특정 값 사이에 있는 빈의 개수가 몇 개인지 보여준다. th = 0.7인 빈의 수 열(column)은 배선밀집도가 0.7이상 0.8미만인 빈의 개수가 몇 개인지 나타낸다.

표 2의 각 회로에 대하여 CDP에서 해결하지 못한 높은 배선밀집도의 빈을 ECDP에서 해결하고 있음을 보여준다. 예를 들어 ckt1의 경우, CDP에서 th = 0.9인 빈이 2개였으나 ECDP에서 이것을 해결하여 th = 0.9인 빈을 모두 제거하였고 th = 0.8인 빈은 CDP가 13인데 비해 ECDP는 3개로 상당한 개선을 보였다. ckt5의 경우, th = 1.0인 빈이 CDP는 21개, ECDP는 7개으로써 배선밀집도가 높은 빈의 수가 기존의 CDP보다 월등히 줄었음을 알 수 있다.

또한 수행속도 측면에서 ckt1을 제외한 나머지 4개 회로에 대하여 기존의 CDP보다 제안한 ECDP가 빠른 시간에 효율적으로 배선밀집문제를 해결함을 알 수 있다.

5. 결론 및 향후과제

VLSI 물리적 설계 단계의 배치(placement) 과정 후 회로의 라우팅(routing)이 불가능하다면 모든 배치 과정을 다시 수행해야 한다. 그런 면에서 배선 밀집도를 배치 단계에서 고려하는 이 기법은 매우 실용적이며 핵심적인 기술이라고 볼 수 있다. 본 논문에서는 기존의 배선밀집도 배치 기법을 개선하여 실행시간 면이나 배선밀집도의 질 면에서 우수한 기법을 제시하였다. 제시한 기법은 배선 밀집도 뿐만 아니라 배선길이를 같이 고려할 수 있도록 설계되어 잠재적으로 타이밍 문제까지 고려 대상에 포함되어 있는 장점이 있다.

표 2. 실험 결과

회로이름	적용기법	빈(bin)의 개수							배선길이	CPU 시간
		th=0.7	th=0.8	th=0.9	th=1.0	th=1.1	th=1.2	th=1.3		
ckt1	실험전	20	20	15	0	1	0	0	12,978,178	
	CDP	36	13	2	0	0	0	0	14,000,749	8m 44s
	ECDP	35	3	0	0	0	0	0	14,382,597	8m 55s
ckt2	실험전	33	29	21	23	39	19	38	23,800,521	
	CDP	41	37	38	47	33	6	0	23,645,578	36m 01s
	ECDP	43	40	53	72	20	1	0	24,362,015	27m 15s
ckt3	실험전	75	45	25	13	6	3	5	34,666,219	
	CDP	112	30	2	0	0	0	0	34,176,563	19m 36s
	ECDP	125	20	0	0	0	0	0	34,427,403	9m 11s
ckt4	실험전	451	342	288	213	106	88	93	40,344,321	
	CDP	713	476	256	105	34	8	12	39,913,256	1h 21m 20s
	ECDP	730	501	268	56	44	12	8	39,983,399	51m 16s
ckt5	실험전	410	309	243	163	106	46	35	118,710,121	
	CDP	701	390	141	21	0	0	1	115,621,690	3h 59m 32s
	ECDP	718	424	120	7	1	0	1	115,943,380	2h 16m 06s

비록 잠재적으로는 타이밍이 고려되어 있다고 볼 수 있지만 실제적인 타이밍 문제는 단순한 배선길이에 대한 고려만으로는 해결되지 않으므로 앞으로의 과제로는 배선 밀집도 뿐만 아니라 타이밍에 초점을 맞추어 배치를 효과적으로 할 수 있는 기법을 찾아야 할 것이다.

참 고 문 헌

[1] A. E. Caldwell, A. B. Kahng, and Igor L. Markov, "Can Recursive Bisection Alone Produce Routable Placements?," Proc. of DAC, pp. 477-482, 2000.

[2] D. J. -H. Huang and A. B. Kahng, "Partitioning-Based Standard-Cell Global Placement with an Exact Objective," Proc. of ISPD, pp. 18-25, 1997.

[3] M. C. Yildiz and P. H. Madden "Improved Cut Sequences for Partitioning Based Placement," Proc. of DAC, pp. 776-729, 2001.

[4] Ke Zhong and S. Dutt, "Effective Partition-Driven Placement with Simultaneous Level Processing and a Global Net Views," Proc. of ICCAD, pp. 254-259, 2000.

[5] Wern-Jieh and Carl Sechen, "Efficient and Effective Placement for Very Large Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp. 349-359, 1995.

[6] G. Meixner and U. Lauther, "Congestion-Driven Placement Using a New Multi-Partitioning Heuristic," Proc. of ICCAD, pp. 332-335, 1990.

[7] Maogang Wang, Xiaojian Yang and Majid Sarrafzadeh, "Congestion Minimization During Placement", IEEE Trans. CAD of Integrated Circuits and Systems, Vol. 19, No. 10, pp. 1140-1148, 2000

[8] Maogang Wang and Majid Sarrafzadeh, "Modeling and Minimization of Routing Congestion", Proc. of ASP-DAC, pp. 185-190, 2000

[9] Maogang Wang and Majid Sarrafzadeh, "On the Behavior of Congestion Minimization During Placement", Proc. of ISPD, pp. 145-150, 1999

[10] S. Hur and J. Lilis, "Mongrel: Hybrid Techniques for Standard Cell Placement", Proc. of ICCAD, pp. 165-170, 2000

[11] 오은경, 허성우, "배선밀집도 드리븐 배치", 한국정보처리학회 논문지, 제13-A권, 제1호, pp. 57-70, 2006

[12] H. Eisenmann and F.M. Johannes, "Generic Global Placement and Floorplanning", Proc. of DAC, pp. 269-274, 1998

[13] 성영태, 허성우, "포스 디렉티드 방법과 최적 인터리빙 기법을 이용한 타이밍 드리븐 배치", 한국정보과학회 논문지, 시스템 및 이론, 제33권, 제2호, pp. 92-104, 2006