Wiring Space Estimation in Two-Dimensional Arrays

Jun Dong Cho
Sungkyunkwan University
Dept. of Electrical and Computer Engineering
Suwon, Korea 440-746
jdcho@yurim.skku.ac.kr

Abstract

We propose a new global routing area estimation approach for high-performance VLSI and MCMs. The objective is to route nets with minimum density of global cells, producing a four-bend routing for each two-terminal net. A solution to this problem can also be used for quick estimation of necessary wiring space (for standard cell array designs) and difficulty of routing (for gate array designs) in the early design planning stage.

1 Introduction

As physical feature sizes decreases, the time delay of electrical signals traveling in the interconnect between active devices and gates is approaching the delay through the devices and gates. Thus, physical interconnections delay will overtake gate delays as a design concern by the year 2000, mandating a shift in the physical design flow for deep-submicron. Therefore, Iterations between synthesis and layout increase dramatically due to timing and routability problems. The key to solving this problem is knowing more about the physical design, i.e., placement and estimated interconnect, early in the design cycle. Motivated by the above, in this paper, we propose a *wiring space estimation* scheme that has an important application in quick estimation of necessary wiring space and difficulty of routing in the early design planning stage (e.g., high-level synthesis step).

Global routing determines, given terminal locations for each net, a path of global cells through which the nets will be routed. The global analysis of the routing region leads to "uniform" density distribution. Global routing is known to be NP-complete even in the case of one-bend routing of two-terminal nets [4]. For the routing problem in two dimensional arrays, there have been various approaches based on hierarchical wiring [1, 3], sequential methods [7, 2], simulated annealing [11], linear programming [6], multicommodity flow [10] and flat approaches [4, 9].

This paper is organized as follows. In Section 2, we define and formulate the problem. In Section 3, we obtain an approximate upper-bounds for the worst-case density of global cells needed to route nets using at most four bends per net. The bound is computed based on a top-down recursion using 4-way partitioning (i.e., structured as a quad tree). Then, in Section 4, we introduce a global routing strategy based on the upper-bound achieved in Section 3. The conclusion of this paper is given in Section 5.

2 Problem Formulation

We adopt the global routing environment involving n two-terminal nets in two-dimensional arrays (refer to Figure 1). A net $i \in N$ consists of two terminals. We assume that a path goes from cell to cell rather than from grid-point to grid-point. Each plane consists of a two-dimensional $m \times m$ grid, being a square tessellation of the plane, with 1×1 being the basic cell-grid size. Therefore, each cell contains at most one pin.

We define the *global density* of the problem as follows:

Definition 2.1 (Global Density d_R) *(refer to Figure 1) Let $cell(i, j)$ denote a cell at the i-th row and the j-th column of the routing region. In general, more than 1 net may cross the border of a cell. Let $d_h(i, j)$ denote the number of nets crossing the border of cells (i, j) and $(i, j + 1)$, $1 \leq i \leq m$ and $1 \leq j \leq m - 1$. Similarly, let $d_v(i, j)$ denote the number of nets crossing the border of cells (i, j) and $(i + 1, j)$, $1 \leq i \leq m - 1$ and $1 \leq j \leq m$. $d_{hmax} = max_{i,j} d_h(i, j)$ is the* horizontal density *of the problem and $d_{vmax} = max_{i,j} d_v(i, j)$ is the* vertical density *of the problem; $d_R = max(d_{vmax}, d_{hmax})$ is the* global density *of the problem. That is, the global density is the maximum number of wires crossing a border between two cells in a routing solution R. A* vertical channel*, denoted by $V(j)$, consists of cells $(i, j), i = 1, \ldots, m$, and a* horizontal channel $H(i)$ *consists of cells* $(i, j), j = 1, \ldots, m$, *respectively.*

We aim to minimize d_R. The minimum d_R is referred to as an *optimal density*, denoted by d_{opt}.

Definition 2.2 *The problem of four-bend global routing (i.e., to estimate the wiring space) can be defined as follows. Given a netlist, find a four-bend global routing with a minimum global density.*

Consider the routing region with uniformly distributed pins. Then, it is often true that global density will be lower if we route the nets with less bends (even though there exists cases where the number of bends is not proportional to the density). Thus, constraining the number of bends produces a "quick" and a "good" global routing. The global routing leads to a "good" wiring space estimation.

3 Computing the Worst Case Approximate Upper-Bound Density

The hierarchical approach has a great attraction such that each level of hierarchy has a manageable-size problem that can be solved exactly. Two top-down partitioning paradigms have been introduced. One is to partition a routing region into four square subregions successively (we refer to it as the *quad-tree*) [1]. The other is to bipartition subregions on the basis of binary cut trees, dividing the routing region vertically or horizontally in a single partition step [5]. The quad-tree approach is more precise than the approach of binary cut trees. The former yields a truly two-dimensional routing paradigm, while the latter results in a one-dimensional partitioning procedure to solve the two-dimensional routing problem. There are two ways of forming the partitioning size: area-based partitioning (slice or rectangular) and point-based partitioning. The former with square bucketing is sufficient since in deep submicron layouts their routing substrates are very dense and pins are usually distributed evenly over the plane. Therefore, we process the top-down recursion by first partitioning the top level representing the whole routing region into four square subregions. That is, a quadrisection having four quadrants is considered at each node of the *quad-tree*. For an $m \times m$ grid (without loss of generality, we assume that m is a power of 2), there are $\log_2 m$ (i.e., 0,1, $\ldots$, $\log_2 m - 1$) levels of two-dimensional arrays. That is, level 0 is the topmost level consisting of a 2 by 2 array, while level $\log_2 m - 1$ is the bottommost level consisting of an $m \times m$ array. We define $\log m$ to be $\log_2 m$ throughout the paper.

The following preliminary definition provides a property of level i of the top-down hierarchy.

Definition 3.1 (QM(i)) *(refer to Figure 1) The quadrisection map* $QM(i)$ *consists of four quadrants* Q_k^i, $k \in (1, 2, 3, 4)$ *(labeled counterclockwise from right-upper corner). Each set of unconnected net in a quadrant is denoted by* U_k^i. *A common boundary of two adjacent quadrants is said to be a* cutline. *Hence, there are two vertical and two horizontal* cutlines *in* $QM(i)$, *denoted by* C_k^i. *The length of each cutline is denoted by* $L(i) = 2^{\log m - i - 1}$. *Each quadrant* Q_k^i *is associated with a set of unconnected terminals* M_k^i.

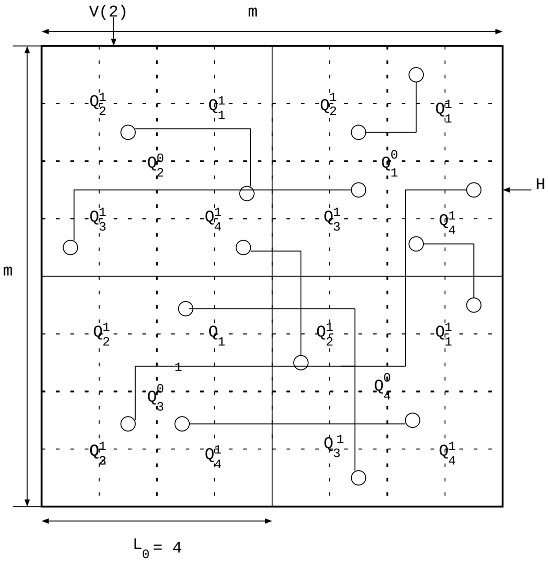

Figure 1. *QM* $(d_R = 1)$

In top-down hierarchical routing, the problem is decomposed into smaller square subregions $QM(i)$ that are solved exactly at level i. For brevity, we denote $Q_k^i, U_k^i, C_k^i, L(i)$, and M_k^i by Q_k, U_k, C_k, L, and M_k, respectively.

Here, based on the top-down 4-way partitioning, we analyze the worst case performance of an approximation algorithm.

Definition 3.2 (Worst Case Lower-Bound Density d_0) *When we recursively partition the square region into 2 by 2 square subregions, there are square subregions of sides* $L_i = 2^{\log m - i - 1}$, *where* $0 \leq i \leq \log m - 1$ *in an* $m \times m$ *two-dimensional array. Consider all square subregions* Q_k^i *with side* L_i, *where* $0 \leq i \leq \log m - 1$ *and let us denote* $max_k |M_k^i|$ *as* U_i, *where* $1 \leq k \leq 4^{i+1}$. *Consider the worst case where all unconnected nets must leave one side of the square containing* U_i. *Then, we define the loosely estimated density as* $d_0 = max_i(\lceil U_i/(L_i) \rceil)$, *where* $0 \leq i \leq \log m - 1$. *Note that* $1 \leq d_0 \leq m/2$.

Lemma 3.1 *The loosely estimated density* d_0 *is less than or equal to* d_{opt}, *where* d_{opt} *is the optimal density in the worst case.*

Proof: The loosely estimated density d_0 is only associated with the nets having its terminals inside a quadrant of top-down *quad-tree*. To ensure the lower-bound density d_0 being below the optimal density, let us consider all square subregions R_j with side L_j, where $L_j = 1, 2, \ldots, m$ in an $m \times m$ two-dimensional array. We define the tightly estimated lower-bound density in the worst case as $d_* = max(\lceil U_j/(L_j)\rceil)$, where U_j is the number of nets with one terminal inside and one terminal outside R_j. Since besides the nets leaving the square region, there might exist nets passing through the region across its borders at the lower level of top-down hierarchy. Thus, the following inequality holds: $d_* \le d_{opt}$. Clearly, $d_0 \le d_*$. Therefore, $d_0 \le d_{opt}$. ■

The approximate density lower-bound d_0 can be calculated in $O(nlogm)$ time.

There are two types of nets in our four-bend routing: **Type 1)** one terminal is in Q_k and the other terminal is in $Q_{(k+1)mod4}$ **Type 2)** one terminal is in Q_k and the other terminal is in $Q_{(k+2)mod4}$. Let us denote by f_{pq} the number of nets whose one terminal is in Q_p and other terminal in Q_q. Let us denote by f_{max_1} the maximum f_{pq} of $\{f_{12}, f_{23}, f_{34}, f_{14}\}$ and by f_{max_2} the second maximum f_{pq} of $\{f_{12}, f_{23}, f_{34}, f_{14}\}$.

Lemma 3.2 (Worst Case Lower-Bound Density d_0) *The tightly estimated upper bound on the number tracks required for connecting a set of nets is*

$$\lceil (f_{13} + f_{24} + f_{max_1} + f_{max_2})/2\rceil.$$

Proof: To distribute nets evenly over cur lines, for Type 1 nets, we need at most $\lceil (f_{max1} + f_{max2})/2\rceil$ tracks to be routed by crossing a single cutline. Among the Type 1 nets, we need at most $\lceil (f_{max1} - f_{max2})/2\rceil$ nets to be routed by crossing the three cutlines if $f_{max1} - f_{max2} \ge 2$. For Type 2 nets, we need at most $\lceil (f_{13} + f_{24})/2\rceil$ tracks when the nets are distributed evenly over two cutlines. In all, we need at most $\lceil (f_{13} + f_{24} + f_{max_1} + f_{max_2})/2\rceil$ tracks. ■

Now, we obtain an upper-bound on the worst case *global density* for 2-bend routing as follows.

Theorem 3.1 (Worst Case Upper-Bound Density) *There is an approximation algorithm achieving the density of global cells, $d_R \le 1.5 d_0 \log(m/d_0)$, in an $m \times m$ two dimensional array, where d_0 is the loosely estimated lower-bound density in the worst case. The total wiring area is at most* $(m \times 1.5 d_0 \log(m/d_0))^2$.

Proof: The proof of the upper-bound *global density* in the worst case is done by induction on i, the number of times the top-down recursion is executed based on a 2×2 subdivisions.

Let $\lfloor\lfloor x\rfloor\rfloor$ be defined such that $\lfloor\lfloor x\rfloor\rfloor = 2^i$ if and only if $2^i \le x < 2^{i+1}$. We define a *basis cell* as a cell with one side $\lfloor\lfloor d_0\rfloor\rfloor \times \lfloor\lfloor d_0\rfloor\rfloor$ at the $\log(m/\lfloor\lfloor d_0\rfloor\rfloor)$-th level of recursion.

Inductively, we consider one square subregion of side $2^{\log m - i}$ that is partitioned into four adjacent square subregions, $Q_k^i, 1 \le k \le 4$, each with side $2^{\log m-i-1}$, for level i, where $0 \le i \le \log(m/\lfloor\lfloor d_0\rfloor\rfloor)$.

Let us assume that at each level we maintain the following invariants.

- **Invariant 1:** if $0 < i < \log(m/\lfloor\lfloor d_0\rfloor\rfloor)$, then at most $\lceil 1.5 d_0\rceil$ tracks are required at each channel of each square subregion.

- **Invariant 2:** if $i = 0$ or $i = \log(m/\lfloor\lfloor d_0\rfloor\rfloor)$, then at most $\lceil d_0\rceil$ tracks are required at each channel of each square subregion.

We want to show that at level i, the invariant is maintained so that at the bottom of the inductive recursion the density of each global cell is, $d_R \le 1.5 d_0 \log(m/d_0)$.

Now, let us consider the set of unconnected nets in four square subregions with each side $2^{\log m-i-1}$ at level i. There are at most $U_i = 2^{\log m-i-1} d_0$ *unconnected* terminals (by definition of d_0 such that $U_i/(2^{\log m-i-1}) < d_0$) leaving the half perimeter of each $Q_k^i, 1 \le k \le 4$. That is, there are at most $U_i = 2^{\log m-i} d_0$ *unconnected* nets.

The lower-bound density of $QM(i)$ can be derived based on Lemma III.2 as follows. There are at most $2^{\log m-i-1} d_0 \times 2$ nets (at most $2^{\log m-i-1} d_0 \times 4$ terminals) at level i. We distribute all $2^{\log m-i} d_0$ nets arbitrarily over the cutlines with at most four bends, using at most $\lceil \frac{(f_{13}+f_{24}+f_{max_1}+f_{max_2})/2}{2^{\log m-i-1}}\rceil \le \lceil \frac{(2^{\log m-i} d_0)/2}{2^{\log m-i-1}}\rceil$ tracks (i.e., d_0 tracks) per channel on each cutline with side $2^{\log m-i-1}$. Thus Invariant 2 is maintained for both top (when $i = 0$) and bottom (when $i = \log(m/d_0)$) levels.

Next, consider routing the set of yet unconnected nets inside Q_1^i, Q_2^i, Q_3^i, and Q_4^i, respectively. Consider the nets, each of whose terminals leaving "different" cutlines (with side $2^{\log m-i-1}$), introducing a vertical or horizontal jog so as to maintain the invariant. The jogs are necessary because the channel on the cutline the net leaves is in a different row or column from the channel in which the terminal of the net lies. Therefore, at most $\frac{2^{\log m-i-1} d_0}{2}$ jogs are introduced at each subregion with side $2^{\log m-i-1}$.

At level $i+1$, there were originally at most $2^{\log m-i-1} d_0$ nets in $QM(i+1)$. At level i, $\frac{2^{\log m-i-1} d_0}{2}$ nets are newly introduced for $QM(i+1)$. Thus in total, there are at most $1.5 \times 2^{\log m-i-1} d_0$ nets in $QM(i+1)$.

The lower-bound density of $QM(i+1)$ can be derived based on Lemma III.2 as follows. There are at most $1.5 \times 2^{\log m-i-1} d_0$ nets at $QM(i+1)$. We distribute all $1.5 \times 2^{\log m-i-1} d_0$ nets arbitrarily over the cutlines with at most four bends, using at most $\lceil \frac{(1.5 \times 2^{\log m-i-1} d_0)/2}{2^{\log m-i-2}}\rceil$ tracks (i.e., $1.5 d_0$ tracks) per channel on each cutline with side $2^{\log m-i-2}$. Thus, the invariant is maintained at level $i+1$.

Therefore, we obtain a tight upper-bound *global density*, $d_R \leq 1.5 d_0 \log(m/d_0)$ at the bottom of the recursion. Therefore, the total wiring area is at most $(m \times 1.5 d_0 \log(m/d_0))^2$. ■

The time complexity of computing the upper-bound only depends on computing time of the lower-bound density d_0, thus we have:

Corollary 3.1 *The upper-bound developed in Theorem III.1 can be achieved in time $O(n \log m)$.*

4 Global Routing with the Worst Case Approximate Upper-Bound Density

Next, let us consider the global routing algorithm complying with the bound developed in the above theorem.

Lemma 4.1 *A global routing to obtain the Invariants developed in Theorem III.1 at each level i can be done in time $O(r \log p)$, where r is the number of unconnected nets in $QM(i)$, and p be the number of channels in the cutline of $QM(i)$.*

Proof: We perform the *net-channel assignment* by employing a so-called 1.5-Dimensional binary search tree. we will construct the tree as follows. The 1.5-D binary search tree consists of a balanced binary search tree and a set of *segment tree* [8] at each node u of the binary search tree. We build two such trees: H-BST and V-BST. The H-BST (resp. V-BST) is employed for locating the horizontal (resp. vertical) channel containing the current event terminal. The leaves of the *segment tree* at each node of H-BST (resp. V-BST) are associated with $p-1$ elementary intervals, $[i, i+1]$, for $i = 1, 2, \ldots, p-1$, where p is the number of vertical (resp. horizontal) channels in this case. The *segment tree* T(1,p), for example, of H-BST, is employed for maintaining the vertical density associated with each interval (i.e, each vertical channel). We can check the maximum value of the number of horizontal segments crossing the current event net's horizontal segment, using the segment tree of the current horizontal channel. Let us consider the case where both horizontal channels containing two terminals (located in different vertical channels) belonging to the same net have already been *saturated* (i.e., exceeded the worst case upper-bound developed in Theorem III.1) within the net's horizontal interval. Then, the net is enqueued, declaring it as a "extending net". If the queue is not empty in the next horizontal channel, then we assigned the net to the channel dequeuing the queue. The time taken to handle each "extending net" is constant. We need $O(\log m)$ time to locate each terminal and another $O(\log m)$ time to check the current horizontal density. After assigning a net to a horizontal channel, we need to update the current horizontal density associated with the net's horizontal segment, spending another $O(\log m)$ time. In this way, we check both H-BST and V-BST whether an interval associated with a horizontal or vertical segment is saturated or not. Therefore, in all, the algorithm runs in $O(n \log m)$ time. ■

Theorem 4.1 *A global routing with the upper-bound density developed in Theorem III.1 can be achieved in time $O(m^2 d_0)$.*

Proof: Let r be the number of unconnected nets in QM(i). Let p be the number of horizontal channels in QM(i). Since the entire routing region contains at most $2^{\log m - i - 1} d_0 \times 2 \times 4^i$ nets at level i, it takes at most $r \log p$ (by Lemma IV.1) $= 2^{\log m + i} d_0 \times \log(m/2^i)$ time to connect the unconnected terminals in all quadrisections at level i. That is, the time complexity at level i, $T(i)$, is $O(d_0 m 2^i \log(m/2^i))$. There are $\log(m/d_0)$ levels of top-down hierarchy, the time complexity required for this algorithm is

$$
\begin{aligned}
T(m, d_0) &= m d_0 \sum_{i=0}^{\log(m/d_0)} (2^i \log(m/2^i)) && (1)\\
&= m^2 d_0 \sum_{i=0}^{\log(m/d_0)} (i/2^i) && (2)\\
&< m^2 d_0 \sum_{i=0}^{\log m - 1} (i/2^i) && (3)\\
&= O(m^2 d_0) && (4)
\end{aligned}
$$

To obtain bounds on the summation in Equation 4, we partition the range of the index to express the series as the sum of two series and then to bound each of the resulting series. To obtain an asymptotic upper-bound on the summation, we observe that the ratio of consecutive terms is $((i+1)/2^{i+1})/(i/2^i) = (i+1)/(2i) \leq 3/4$, if $i \geq 2$. Thus, the summation can be split into $\sum_{i=0}^{\log m - 1}(i/2^i) = \sum_{i=0}^{1}(i/2^i) + \sum_{i=2}^{\log m - 1}(i/2^i) = O(1)$, since the second summation is a decreasing geometric series. ■

Note that the number of nets n is $O(m^2)$ and d_0 is constant in most of practical instances. Then, the algorithm takes $O(n)$ computational time.

5 Conclusion

In this paper, by incorporating the flat approximation into routing of each level of top-down recursion, we obtain in $O(n \log m)$ time a new and tight approximate upper-bound on the worst case density of global cells. Also, a fast 4-bend global routing algorithm was presented, complying with a worst case upper-bound density.

Acknowledgments

The author wishes to acknowledge the financial support of the Korea Research Foundation under Grant 981-0909-037-2.

References

[1] M. Burstein and R. Pelavin. "Hierarchical Wire Routing". *IEEE Transactions on Computer Aided Design*, CAD-2(4):223–234, October 1983.

[2] C. Chiang, M. Sarrafzadeh, and C. K. Wong. "Global Routing based on Steiner Min-Max Trees". *IEEE Transactions on Computer Aided Design*, 9(12):1315–1325, 1990.

[3] J. D. Cho and M. Sarrafzadeh. "Four-Bend Top-Down Global Routing". *IEEE Trans. on Computer Aided Design, vol 17, no. 9*, pages 793–802, Sept. 1998.

[4] R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. V. Vazirani, and V. V. Vazirani. "Global Wire Routing in Two-Dimensional Arrays". *Algorithmica*, 2(1):113–129, 1987.

[5] M. Marek-Sadowska. "Route Planner for Custom Chip Design". In *International Conference on Computer-Aided Design*, pages 246–249. IEEE, November 1986.

[6] G. Meixner and U. Lauther. "A New Global Router Based on a Flow Model and Linear Assignment". In *International Conference on Computer-Aided Design*, pages 44–47. IEEE, November 1990.

[7] R. Nair. "A Simple Yet Effective Technique for Global Wiring". *IEEE Transactions on Computer Aided Design*, CAD-6(2):165–172, March 1987.

[8] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[9] M. Sarrafzadeh and D. Zhou. "Global Routing of Short Nets in Two-Dimensional Arrays". *The International Journal of Computer Aided VLSI Design*, 2(2):197–211, 1990.

[10] E. Shargowitz and J. Keel. "A Global Router Based on Multicommodity Flow Model". *INTEGRATION: The VLSI Journal*, 5:3–16, 1987.

[11] M. P. Vecchi and S. Kirkpatrick. "Global Wiring by Simulated Annealing". *IEEE Transactions on Computer Aided Design*, CAD-2(4):215–222, 1983.