

# Short Papers

## Wiring Space and Length Estimation in Two-Dimensional Arrays

Jun Dong Cho

**Abstract**—We propose a new global routing area estimation approach for high-performance very large scale integration and multichip modules (MCM's). The objective is to route nets with minimum density of global cells, producing a four-bend routing for each two-terminal net. We propose an approximate upper bound on global cell  $d_R \leq 2d_0 \log(m/(2d_0))$ , in an  $m \times m$  two-dimensional array, where  $d_0$  is the estimated lower-bound density. The total wirelength is  $(2\alpha + \beta)4m^2d_0/3$ , where  $\alpha + \beta = 1$  and  $\alpha$  is the percentage of diagonal combinations and  $\beta$  is the percentage of adjacent combinations of nets. If  $\alpha \leq \beta$  (this assumption holds since a good placement minimizes the longer wires), then the total wirelength is at most  $2m^2d_0$ . By counting on the adjacent and diagonal combinations separately in the cost function,  $d_R \leq \lceil 4d_0/3 \rceil \times \log(\lceil m/(4d_0/3) \rceil)$ . We verified that the bound obtained are realistic in the worst case. A solution to this problem can 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.

**Index Terms**—Global routing, routing estimation, VLSI layout.

### I. INTRODUCTION

As physical feature sizes decrease, 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-submicrometer. 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). We also propose an upper bound on total wire length. Therefore, in the early design stage, for example, during high-level synthesis step, we perform a quick placement followed by our quick global wire estimation algorithm introduced in this paper. By doing so, we can effectively estimate the cost of the chip using the quick estimation of wiring space, and also estimate the power consumption due to the total wire capacitance by computing the upper bound on the total wirelength. This paper is organized as follows. In Section II, we define and formulate the problem. In Section III, we describe a global routing estimation scheme in top-down hierarchy. In Section IV, we obtain a lower-bound density of global cells. In Section V, we obtain an approximate upper-bound 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 four-way partitioning (i.e., structured as a quad tree). Also an upper bound on total wirelength is derived. The verification and conclusion of this paper are given in Sections VI and Section VII, respectively.

Manuscript received June 14, 1999; revised November 9, 1999. This paper was recommended by Associate Editor T. Yoshimura.

The author is with Sungkyunkwan University, Department of Electrical and Computer Engineering, Suwon 440-746, Korea (e-mail: jdcho@skku.ac.kr).

Publisher Item Identifier S 0278-0070(00)04730-8.

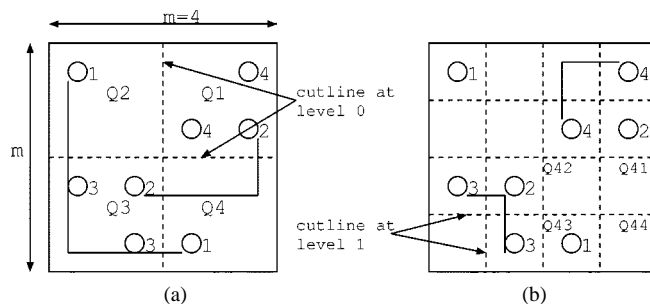


Fig. 1. Global routing in a top-down hierarchy,  $d_R = 1$ . (a) Top-level routing (nets 1 and 2). (b) Second-level routing (nets 3 and 4).

### II. PROBLEM FORMULATION

We adopt the global routing environment involving  $n$  two-terminal nets in two-dimensional (2-D) arrays (Fig. 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 2-D  $m \times m$  grid, being a square tessellation of the plane, with  $1 \times 1$  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 II.1:** (global density  $d_R$ ) (Fig. 1) Let  $\text{cell}(i, j)$  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  $1 \leq j \leq m$ .  $d_{h \max} = \max_{i,j} d_h(i, j)$  is the horizontal density of the problem and  $d_{v \max} = \max_{i,j} d_v(i, j)$  is the vertical 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  $(i, j)$ ,  $i = 1, \dots, m$ , and a horizontal channel  $H(i)$  consists of cells  $(i, j)$ ,  $j = 1, \dots, m$ , respectively.

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

**Definition II.2:** (global routing) The problem of global routing (i.e., to estimate the wiring space) can be defined as follows. Given a netlist with a placement information, find a global routing with a minimum global density of a channel.

Global routing determines, given terminal locations for each net (i.e., a given a placement of cells), 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 2-D arrays, there have been various approaches based on hierarchical wiring [1], [3], [5], sequential methods [2], [4], [8], simulated annealing [10], linear programming [6], and multicommodity flow [9]. A quick heuristic is necessary for estimation of necessary wiring space and difficulty of routing in the early design planning stage. 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.

Karp *et al.* [4] proposed an approximate upper bound on global cells which is  $m/2$  in an  $m \times m$  arrays. Sarrafzadeh *et al.* [8] proposed a bottom-up hierarchical global routing with a density of global cells  $O(d_{opt} \log(s/d_{opt}))$ , where  $s$  is the span of nets and  $d_{opt}$  is the optimal channel density. In this paper, the bound derived by [8] is improved and the new wire length upper-bound is derived.

### III. QUADRISECTION MAP TOP-DOWN GLOBAL ROUTING

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) [3]. 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 2-D routing paradigm, while the latter results in a one-dimensional partitioning procedure to solve the 2-D 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-submicrometer 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 as in Fig. 1. 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.,  $0, 1, \dots, \log_2 m - 1$ ) levels of 2-D arrays. That is, level 0 is the top-most level, consisting of a  $2 \times 2$  array, while level  $\log_2 m - 1$  is the bottom-most 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 III.1:** [QM( $i$ )] (Fig. 1) The quadrisection map  $QM(i)$  consists of four quadrants  $Q_k^i$ ,  $k \in (1, 2, 3, 4)$  (labeled counterclockwise from upper-right 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  $iL(i) = 2^{\log m - i - 1}$ ,  $0 \leq i \leq \log m - 1$ . Each quadrant  $Q_k^i$  is associated with a set of unconnected terminals  $M_k^i$ .

In top-down hierarchical routing, the problem is decomposed into smaller square subregions  $QM(i)$  that are solved exactly at level  $i$ .

### IV. COMPUTING THE WORST CASE APPROXIMATE LOWER-BOUND DENSITY

Here, based on the top-down four-way partitioning, we analyze the worst case lower bound on the global cells.

**Definition IV.1:** (worst case lower-bound density  $d_0$ ) When we recursively partition the square region into  $2 \times 2$  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  $m \times m$  2-D 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 best case where all unconnected nets should leave two cutlines of the square containing  $U^i$ . Then, we define the estimated density as  $d_0 = \max_i (\lceil U^i / (2L^i) \rceil)$ , where  $0 \leq i \leq \log m - 1$ . Note that  $1 \leq d_0 \leq m/2$ .

The approximate density lower-bound  $d_0$  can be calculated in  $O(n \log m)$  time.

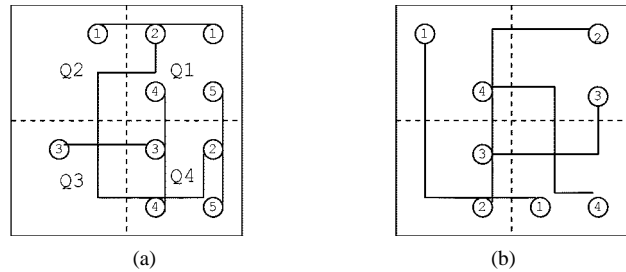


Fig. 2. Four-bend routing. (a) Patterns in adjacent combination. (b) Patterns in diagonal combination.

### V. COMPUTING THE WORST CASE APPROXIMATE UPPER-BOUND DENSITY

**Definition V.1:** (net patterns) As shown in Fig. 3, there are two types of nets in our four-bend routing.

- Type 1 (adjacent combination): one terminal is in  $Q_k$  and the other terminal is in  $Q_{(k+1) \bmod 4}$ .
- Type 2 (diagonal combination): one terminal is in  $Q_k$  and the other terminal is in  $Q_{(k+2) \bmod 4}$ .

Let us denote by  $F_{pq}$  (respectively,  $f_{pq}$ ) the set of nets (respectively, its cardinality) whose one terminal is in  $Q_p$  and other terminal in  $Q_q$ . That is,  $F_{12}, F_{23}, F_{34},$  and  $F_{4,1}$  are in Type 1, and  $F_{13}$  and  $F_{24}$  are in Type 2. Then, the total number of nets  $f = f_{12} + f_{23} + f_{34} + f_{4,1} + f_{13} + f_{24}$ . Let us denote by  $f_{\max 1} = \max\{f_{12}, f_{23}, f_{34}, f_{4,1}\}$  and by  $f_{\max 2} = \max\{f_{13}, f_{24}\} - f_{\max 1}$ , and also  $f_{\max 3} = \max\{f_{12}, f_{23}, f_{34}, f_{4,1}\} - f_{\max 1} - f_{\max 2}$ ,  $f_{\max 4} = \min\{f_{12}, f_{23}, f_{34}, f_{4,1}\}$ .

**Lemma V.1:** (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 cutlines, for Type 1 nets, we need at most  $\lceil (f_{\max 1} + f_{\max 2}) / 2 \rceil$  tracks to be routed by crossing a single cutline. Type 1 nets can be routed using two patterns, *detour* (routed by crossing the three cutlines) and *straight* (routed by crossing the single cutline) connections. We need at most  $\lceil (f_{\max 1} - f_{\max 2}) / 2 \rceil$  nets to be routed with *detour connection* if  $f_{\max 1} - f_{\max 2} \geq 2$ . For Type 2 nets, we need at most  $\lceil (f_{13} + f_{24}) / 2 \rceil$  tracks since 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.

For example, in Fig. 2(a),  $d_R = (f_{\max 1} + f_{\max 2}) / 2 = (f_{14} + f_{12}) / 2 = (3 + 1) / 2 = 2$ , and in Fig. 2(b),  $d_R = (f_{13} + f_{24}) / 2 = 2$ . Now, we obtain an upper-bound on the worst case global density as follows.

**Theorem 1:** (worst case upper-bound density) There is an approximation algorithm achieving the density of global cells,  $d_R \leq 2d_0 \log(m / (2d_0))$ , in an  $m \times m$  2-D array, where  $d_0$  is the estimated lower-bound density. The total wiring area is at most  $(m \times 2d_0 \log(m / (2d_0)))^2$ .

**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  $2 \times 2$  subdivisions.

Inductively, we consider one square subregions,  $Q_k^i$ ,  $1 \leq k \leq 4$ , each with side  $2^{\log m - i - 1}$ , for level  $i$ , where  $0 \leq i \leq \log(m / (2d_0))$ .

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

- **Invariant:** at most  $2(i+1)d_0$  tracks are required at each channel on the cutline of each square subregion at level  $i$ .

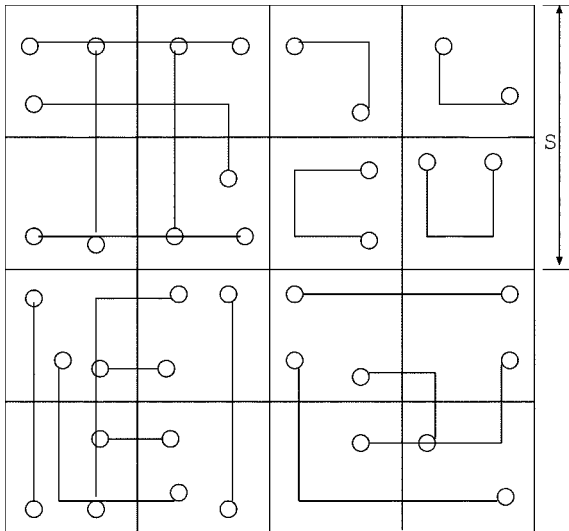


Fig. 3. Well-distributed instance.

We want to show that at level  $i$ , the invariant is maintained so that the bottom of the density of each global cell is,  $d_R \leq \lceil 2d_0 \times \log(m/(2d_0)) \rceil$ .

Now, let us consider the set of unconnected nets in four square subregion with each side  $2^{\log m-i-1}$  at level  $i$ . There are at most  $U^i = 2^{\log m-i} d_0$  unconnected terminals (by definition of  $d_0$  such that  $U^i/(2 \times 2^{\log m-i-1}) < d_0$ ) leaving the half perimeter of each  $Q_k^i, 1 \leq k \leq 4$ . That is, there are at most  $U^i = 2^{\log m-i+1} d_0$  unconnected nets. We distribute all  $2^{\log m-i+1} d_0$  nets arbitrarily over the cutlines with at most four bends, using at most  $\lceil [(f_{13} + f_{24} + f_{\max 1} + f_{\max 2})/2]/[2^{\log m-i-1}] \rceil \leq \lceil [(2^{\log m-i+1} d_0)/2]/[2^{\log m-i-1}] \rceil$  tracks (i.e.,  $2d_0$  tracks) per channel on each cutline with side  $2^{\log m-i-1}$ . Thus, invariant is maintained for level  $i$ .

Next, let us consider the set of unconnected nets in four square subregion with each side  $2^{\log m-i-2}$  at level  $(i+1)$ . Consider routing the set of yet unconnected nets inside  $Q_1^i, Q_2^i, Q_3^i$ , and  $Q_4^i$ , respectively. Consider for example  $Q_1^i$ . The nets inside  $Q_1^i$ , each of whose terminals leaving cutlines with side  $2^{\log m-i-1}$  at level  $i$  also contribute to density increase on the cutline at level  $(i+1)$ . Thus,  $2^{\log m-i} d_0$  terminals at level  $i$  are propagated to level  $(i+1)$  to be routed at level  $(i+1)$ .

At level  $(i+1)$ , there were originally at most  $2^{\log m-i-1} d_0$  unconnected terminals leaving the half perimeter of each  $Q_k^i, 1 \leq k \leq 4, \in QM(i+1)$  (by definition of  $d_0$  such that  $U^{i+1}/(2 \times 2^{\log m-i-2}) < d_0$ ). That is, there are at most  $U^{i+1} = (2^{\log m-i-1} d_0 \times 4)/2 = 2^{\log m-i} d_0$  unconnected nets at  $Q_1^i (= QM(i+1))$ .

Thus, in total there are at most  $2^{\log m-i+1} d_0$  nets to be routed in  $QM(i+1)$ . We distribute all  $2^{\log m-i+1} d_0$  nets arbitrarily over the cutlines with at most four bends, using at most  $\lceil [(f_{13} + f_{24} + f_{\max 1} + f_{\max 2})/2]/[2^{\log m-i-2}] \rceil \leq \lceil [(2^{\log m-i+1} d_0)/2]/[2^{\log m-i-2}] \rceil$  tracks (i.e.,  $4d_0$  tracks) per channel on each cutline with side  $2^{\log m-i-2}$ . Thus, invariant is maintained for level  $i+1$ . Invariant for level  $i+2$  can also be obtained from the relation between levels  $i+1$  and  $i+2$  which has the same structure between levels  $i$  and  $i+1$ .

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

The time complexity of computing the upper-bound only depends on computing time of the lower-bound density  $d_0$ , thus, the upper-bound can be achieved in time  $O(n \log m)$ .

TABLE I  
A VERIFICATION RESULT ON SINGLE  
LEVEL INSTANCES

$Ex(m/n/d_0)$	2-b	4-b	The 5.1	Coro. 5.2
Ex1 (4/8/1)	2	2	2	2
Ex2 (8/32/2)	5	4	4	4
Ex3 (16/128/4)	9	8	8	8

*Corollary V.1:* The total wirelength required to achieve the bound obtained in Theorem 5.1 is  $(2\alpha + \beta)4m^2 d_0/3$ , where  $\alpha + \beta = 1$  and  $\alpha$  is the percentage of diagonal combinations and  $\beta$  is the percentage of adjacent combinations.

*Proof:* As shown in Theorem 5.1 and Fig. 2, Type 1 nets can be routed using two patterns, *detour* (routed by crossing three cutlines) and *straight* (routed by crossing single cutline) connections. Type 2 nets can be routed with *one-bend* (routed by crossing two cutlines). That is, we need at most  $f_3 = \lceil (f_{\max 1} - f_{\max 2})/2 \rceil$  detour connections, at most  $f_2 = (f_{13} + f_{24})$  one-bend connections, and at most  $f_1 = f - f_3 - f_2$  straight connections.

The total wirelength required to route nets at level  $i$  of the top-down hierarchy,  $w_i = 2^{\log m-i-1} \times (3f_3 + 2f_2 + f_1)$ . Thus, the total wirelength required to route in an  $m \times m$  2-D array is  $w = \sum_{i=0}^{\log m-1} 2^{\log m-i-1} \times (3f_3 + 2f_2 + f_1)$ . As shown in Theorem 5.1, the total number of nets to be routed in an  $m \times m$  2-D array is  $\sum_{i=0}^{\log m-1} 2^{\log m-i+1} d_0$ . Thus, we have  $w = \sum_{i=0}^{\log m-1} 2^{\log m-i-1} \times (2\alpha + \beta) \times 2^{\log m-i+1} d_0 = (2\alpha + \beta) \times 4m^2 d_0/3$ , provided that we are given a good placement, i.e.,  $f_{\max 1}$  is similar to  $f_{\max 2}$  (i.e.,  $f_3$  term is removed) (this assumption holds since a good placement distributes nets evenly over the cutlines). If  $\alpha \leq \beta$  (this assumption holds since a good placement minimizes the longer wires), then  $w \leq 2m^2 d_0$ .

Now we further improve the bound derived in Theorem 5.1 by counting on the adjacent and diagonal combinations separately in the cost function, in the most of cases, it is true that  $f_{13} + f_{24} + f_{\max 1} + f_{\max 2} \leq 2f/3$ . This is true based on the assumption that  $f_{13}, f_{24}, f_{\max 1}, f_{\max 2}, f_{\max 3}$  and  $f_{\max 4}$  are of approximately same value, since a good placement attempts to distribute nets evenly over the cutlines. Thus, we have the following.

*Corollary V.2:* If  $f_{13} + f_{24} + f_{\max 1} + f_{\max 2} \leq 2f/3$ , then we obtain a global routing with  $d_R \leq \lceil 4d_0/3 \rceil \times \log(\lceil m/(4d_0/3) \rceil)$ .

## VI. VERIFICATION

To verify the approximation scheme, we tested with two types of instances: single-level instances and multiple-level instances as follows.

For single-level instances, as we see in Table I, the approximation bound derived in this paper was same as the optimal case in most of instances. Here, 2-b (respectively, 4-b) represents an optimal solution with allowing at most two bends (respectively, four bends) per net for the worst case instances. In this table, "The 5.1" (respectively, "Corollary 5.2") represents the results of applying the bound derived in Theorem 5.1 (respectively, Corollary 5.2).

Next we observe the multiple-level instances. For the well-distributed and localized two-level instances, e.g., as in Fig. 3, where  $d_R \leq 2d_0 \log s/(2d_0)$ , where  $s$  is the span of nets. In Fig. 3, the wires depicts the global routes for two-terminal nets.

The more general multiple-level instances such that density-centric spots exists are shown in Fig. 4. The filled region in Fig. 4 represents the wire congestion spots. The darker the region filled, the denser the routing congestion in the region.

In the density-centric spots, the routing density is increased by  $2d_0$  in the worst case at each level of the top-down hierarchy. Therefore,

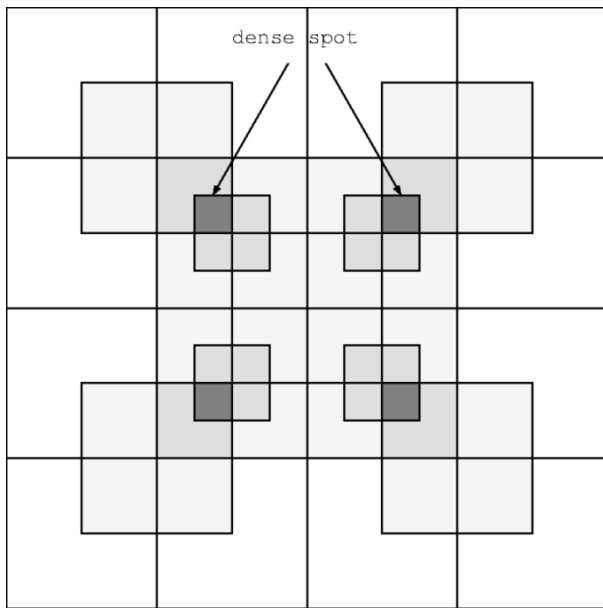


Fig. 4. An instance where dense spots exist.

due to the density-centric spots, i.e., the darkest four spots,  $d_R$  is close to  $2d_0 \log(m/(2d_0))$ , as shown in Theorem 5.1.

## VII. 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 lower and upper-bound on the worst case density of global cells. We observed that  $2d_0 \leq d_R \leq 2d_0 \log(m/(2d_0))$ , in an  $m \times m$  2-D array, where  $d_0$  is the worst case lower-bound density. We showed the total wirelength required to route nets in an  $m \times m$  2-D array is at most  $2m^2 d_0$ . By counting on the adjacent and diagonal combinations separately in the cost function,  $d_R \leq \lceil 4d_0/3 \rceil \times \log(\lceil m/(4d_0/3) \rceil)$ . We verified that the bound obtained are realistic in the worst case. We noted that placement congestion is necessary to find a more tight bound since  $s$  depends on the net distribution and net span, where  $d_R \leq 2d_0 \log(s/(2d_0))$ .

## REFERENCES

- [1] M. Burstein and R. Pelavin, "Hierarchical wire routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 223–234, Oct. 1983.
- [2] C. Chiang, M. Sarrafzadeh, and C. K. Wong, "Global routing based on Steiner min-max trees," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 1315–1325, Dec. 1990.
- [3] J. D. Cho and M. Sarrafzadeh, "Four-bend top-down global routing," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 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*, vol. 2, no. 1, pp. 113–129, 1987.
- [5] M. Marek-Sadowska, "Route planner for custom chip design," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1986, pp. 246–249.
- [6] G. Meixner and U. Lauther, "A new global router based on a flow model and linear assignment," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1990, pp. 44–47.
- [7] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Berlin, Germany: Springer-Verlag, 1985.
- [8] M. Sarrafzadeh and D. Zhou, "Global routing of short nets in two-dimensional arrays," *The Int. J. Computer-Aided VLSI Design*, vol. 2, no. 2, pp. 197–211, 1990.
- [9] E. Shargowitz and J. Keel, "A global router based on multicommodity flow model," *Integration: The VLSI Journal*, vol. 5, pp. 3–16, 1987.
- [10] M. P. Vecchi and S. Kirkpatrick, "Global wiring by simulated annealing," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 215–222, 1983.

## A Practical Approach to the Synthesis of Arithmetic Circuits Using Carry-Save-Adders

Taewhan Kim and Junhyung Um

**Abstract**—Carry-save-adder (CSA) is one of the most widely used types of operation in implementing a fast computation of arithmetics. An inherent limitation of the conventional CSA applications is that the applications are confined to the sections of arithmetic circuit that can be directly translated into addition expressions. To overcome this limitation, from the analysis of the structures of arithmetic circuits found in industry, we derive a set of simple, but effective CSA transformation techniques other than the existing ones. Those are 1) *optimization across multiplexors*, 2) *optimization across design boundaries*, and 3) *optimization across multiplications*. Based on the techniques, we develop a new timing-driven CSA transformation algorithm that is able to utilize CSA's extensively throughout the whole circuits. Experimental data for practical testcases are provided to show the effectiveness of our algorithm.

**Index Terms**—Circuit optimization, timing, arithmetic circuit, RTL synthesis.

## I. INTRODUCTION

Timing of circuit is one of the most important design criteria to be optimized in several phases of synthesis process. In behavioral synthesis phase [1], timing refers to two factors, namely, *number of control steps* (i.e., *latency*) and *cycle time*. Scheduling step optimizes the timing factors under resource constraint or optimizes the amount of resources required under the timing constraint. Hardware allocation/binding step then allocates and binds implementations to realize a circuit for the scheduled control and data flow graph (CDFG) of design while satisfying the required timing constraint. Ideally, to derive not only fully optimized but also "feasible" timing it is necessary to take into account the problem of implementation selection for operations during scheduling because mapping an operation to different implementations leads to different timings in CDFG, thereby resulting in schedules with different latencies. Practically, however, due to a huge computational complexity it is assumed that each operation was bound to a particular implementation during the scheduling step and the possibility of (re)implementations is considered in the allocation/binding step. Consequently, the timing obtained from the behavioral synthesis is either "underestimated" or "overestimated." By overestimated, we mean the situation that the scheduler assumed too slow implementations, thereby wasting performance unnecessarily. By underestimated, we mean the

Manuscript received July 1, 1999; revised December 16, 1999. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc). This paper was recommended by Associate Editor R. Gupta.

The authors are with the Department of Electrical Engineering, Computer Science and Advanced Information Technology Research Center (AITrc), Korea Advanced Institute of Science and Technology, Taejon 305-701, Korea (e-mail: tkim@cs.kaist.ac.kr).

Publisher Item Identifier S 0278-0070(00)04731-X.