

X-Architecture Placement Based on Effective Wire Models

Tung-Chieh Chen[†], Yi-Lin Chuang[†], and Yao-Wen Chang^{†‡}

Graduate Institute of Electronics Engineering[†]

Department of Electrical Engineering[‡]

National Taiwan University, Taipei 106, Taiwan

{donnie, nicky}@eda.ee.ntu.edu.tw; ywchang@cc.ee.ntu.edu.tw

ABSTRACT

In this paper, we derive the X-half-perimeter wirelength (XHPWL) model for X-architecture placement and explore the effects of three different wire models on X-architecture placement, including the Manhattan-half-perimeter wirelength (MHPWL) model, the XHPWL model, and the X-Steiner wirelength (XStWL) model. For min-cut partitioning placement, we propose a generalized net-weighting method that can exactly model the wirelength after partitioning by the net weight. The net-weighting method is general and can be incorporated into any wire models such as the XHPWL and XStWL models. For analytical placement, we smooth the XHPWL function using log-sum-exp functions to facilitate analytical placement. Our study shows that both the XHPWL model and the XStWL model can reduce the X wirelength. In particular, our results reveal the effectiveness of the X architecture on wirelength reduction during placement and thus the importance of the study on the X-placement algorithms, which is different from the results given in the previous work that the X-architecture placement might not improve the X-routing wirelength over the Manhattan-architecture placement.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids [Placement and Routing]

General Terms

Algorithms, Performance, Design

Keywords

Physical design, placement, X architecture, min cut, partitioning, net weighting, Steiner tree

1. INTRODUCTION

1.1 The X Architecture

As integrated circuit (IC) geometries keep shrinking, interconnect delay has become the dominant factor in determining circuit performance. To minimize interconnect delay, the *X architecture* [26] has been introduced as a new

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD'07, March 18–21, 2007, Austin, Texas, USA.

Copyright 2007 ACM 978-1-59593-613-4/07/0003 ...\$5.00.

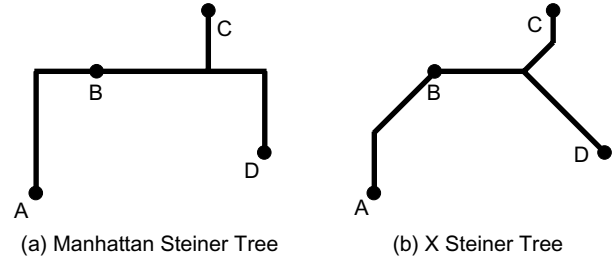


Figure 1: Example Steiner trees using (a) Manhattan routing and (b) X routing.

interconnect architecture for the IC's to reduce interconnect length and thus improve circuit performance. The X architecture allows 45- and 135-degree routes, leading to smaller wirelength and thus smaller delay and power consumption. Figures 1(a) and (b) show the Steiner trees based on the traditional Manhattan and X architectures, respectively. It is obvious that the wirelength of the X-Steiner tree is smaller than that of the Manhattan-Steiner tree due to the diagonal routes.

The traditional Manhattan architecture has its obvious advantages of easier design, but it incurs significant and needless wirelength over the Euclidean optimum. As reported in [26], the X architecture results in significantly shorter average wirelength than the Manhattan architecture. The X architecture's pervasive uses of diagonal routing can reduce wirelength. Further, the wirelength reduction make the circuit design problem easier to solve, resulting in faster timing closure.

1.2 Previous work

To fully utilize the X architecture, we need to consider both X-placement and X-routing algorithms. Some X-routing algorithms have been proposed in the literature [4, 13], and their results show that the wirelength can be reduced effectively using the X architecture. In contrast, not much work on X-placement is studied in the literature. In [7], both 45- and 60-degree wiring metrics were explored, using a simulated annealing based placer with a Steiner wirelength optimization objective. The work was based on some simplified assumption that all cells are of unit size, and only five pins and higher-degree nets are considered. Further, the simulated annealing method does not scale well and can handle the problem sizes of only up to 1500 nets.

Based on the partitioning placement framework, Teig and Ganley [27] patented 45-/135-degree diagonal cutlines (or *X cutlines*) to partition the placement region to favor diagonal wiring. Very recently, Ono and Madden in [20] conducted a complete study on the patent and proposed a pioneering min-cut partitioning based placer for the X architecture.

They found that the X cutlines does not lead to better placement results for the X architecture; the resulting wirelength by using the X cutlines is even worse than that by using traditional Manhattan cutlines.

1.3 Our Contribution

In this paper, we derive a new X-half-perimeter wirelength (XHPWL) model for X-architecture placement. We define the X bounding box as the smallest bounding box formed by the 0-/45-/90-/135-degree line segments that enclose all terminals of a net. The XHPWL is the half of the perimeter length of the X bounding box. We then incorporate this new wire model into both min-cut partitioning and analytical placement algorithms.

For the min-cut partitioning placement, we propose a generalized net-weighting method that can exactly model the wirelength after partitioning by the net weight. The net-weighting method is general and can be incorporated into any wire models. We apply the XHPWL model and the X-Steiner wirelength (XStWL) model for X-architecture min-cut placement based on the net-weighting method. Experimental results show that the total X-Steiner wirelength can be reduced by 1% and 5% on average for the XHPWL and the XStWL models, respectively.

For the analytical placement, we first use log-sum-exp functions to smooth the XHPWL function so that analytical solvers can minimize XHPWL effectively and efficiently. Our experimental results show that the analytical placer incorporated with the XHPWL model can reduce the X-Steiner wirelength by 3% on average.

It should be noted that our X-architecture min-cut partitioning and analytical placers are *the first* effective placers to reduce the X-Steiner wirelength. The work in [20] does not obtain smaller X-Steiner wirelength, compared with the traditional Manhattan partitioning placement. As a result, they concluded that the X-architecture placement does not lead to smaller X-Steiner wirelength. Based on the new wire models, in contrast, our min-cut and analytical placers can effectively obtain smaller X-Steiner wirelength and show the promise of the X architecture, as it should be.

We also perform X-routing by constructing X-Steiner trees to explore the effect of our X-placement on the final routing solutions. Without X-placement, our experimental results show that X-routing reduces only 8% wirelength, compared with the traditional Manhattan routing. With X-placement, the X-routing can reduce the wirelength by 11% and 12% for analytical and min-cut partitioning placement, respectively. The results reveal the effectiveness of the X architecture on wirelength reduction during placement and thus the importance of the study on the X-placement algorithms, which is different from the results given in the previous work [20] that the X-architecture placement might not improve the X-routing wirelength over the traditional Manhattan-architecture placement.

This paper is organized as follows. Section 2 introduces our new XHPWL model. The XHPWL model is applied to min-cut partitioning and analytical placement algorithms in Section 3 and Section 4, respectively. The experimental results are given in Section 5, and Section 6 gives the conclusion.

2. X-HALF-PERIMETER WIRELENGTH (XHPWL) MODEL

Traditional placement for the Manhattan architecture is based on the minimization of the Manhattan-half-perimeter wirelength (MHPWL for short, or traditionally called HPWL). An example Manhattan bounding box of a four-terminal net is shown in Figure 2(a). The MHPWL is the half of the perimeter length of the Manhattan bounding box, and the

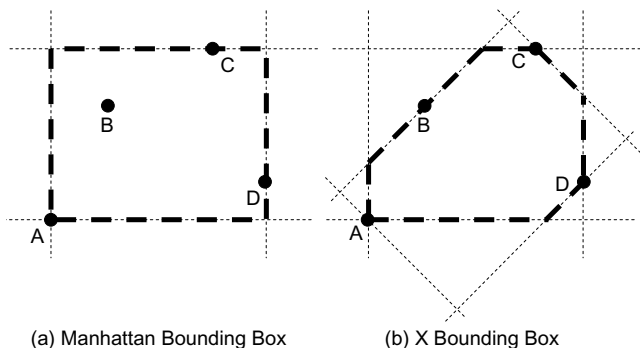


Figure 2: The Manhattan bounding box and the X bounding box.

MHPWL of a net e can be computed by the following equation:

$$\text{MHPWL}(e) = \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j|, \quad (1)$$

where v_i is a terminal of the net, and (x_i, y_i) is the coordinate of v_i .

The MHPWL does not consider the 45-/135-degree routes of the X architecture. We thus propose a new X-half-perimeter wirelength (XHPWL) model for the X architecture. We define the X bounding box as the smallest bounding box formed by 0-/45-/90-/135-degree line segments. Figure 2(b) gives an example X bounding box of the four terminals. The X bounding box has the following properties:

PROPERTY 1. *The size of the X bounding box is always smaller than or equal to that of the Manhattan bounding box.*

PROPERTY 2. *Every optimal X-Steiner tree (with the minimum wirelength) must be within its X bounding box.*

To compute the perimeter length of the X bounding box, we can use the procedure shown in Figure 3. We first derive the Manhattan bounding box, then remove the dotted line segments as shown in Figure 3(b), introduce the oblique line segments (see Figure 3(c)), and finally form the resulting X bounding box (see Figure 3(d)). As a result, the XHPWL of a net e can be computed by the following equation:

$$\begin{aligned} \text{XHPWL}(e) &= \text{MHPWL} - \frac{\text{dotted line length in Figure 3(b)}}{2} + \frac{\text{oblique line length in Figure 3(c)}}{2} \\ &= (\sqrt{2} - 1) \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) - \left(\frac{\sqrt{2}}{2} - 1 \right) \left(\max_{v_i, v_j \in e} |(x_i + y_i) - (x_j + y_j)| + \max_{v_i, v_j \in e} |(x_i - y_i) - (x_j - y_j)| \right), \end{aligned} \quad (2)$$

where v_i is a terminal of the net and (x_i, y_i) is the coordinate of v_i . Based on the triangle inequality, the total length of the added oblique line segments in Figure 3(c) is always smaller than that of the removed line segments in Figure 3(b). Thus we have

$$\text{XHPWL}(e) \leq \text{MHPWL}(e). \quad (3)$$

In the following sections, we will show how to apply the XHPWL model to min-cut partitioning and analytical placement.

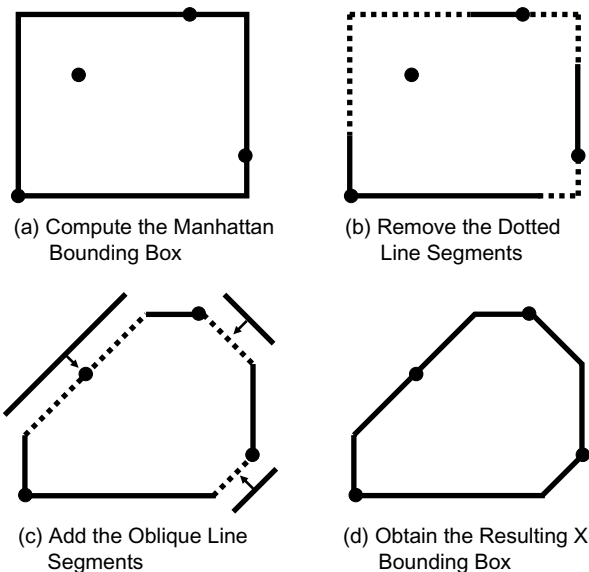


Figure 3: The procedure of computing the perimeter length of the X bounding box.

3. X-ARCHITECTURE MIN-CUT PARTITIONING PLACEMENT

Min-cut partitioning placement is usually based on recursive bisection [2,9,21] or quadrisection [25]. In the following, we first introduce the min-cut partitioning placement framework, and then propose a generalized net-weighting method for min-cut partitioning to incorporate the XHPWL model and the X-Steiner wirelength (XStWL) model into the min-cut partitioning placement algorithm.

3.1 Min-Cut Partitioning Placement Framework

Partitioning placement recursively divides a placement region into several subregions, cut a netlist into sub-netlists, and assign the sub-netlists into regions. Min-cut placers partition the netlist based on the KL [17] or FM [12] heuristic, or some other extensions [3,16].

Traditional Manhattan partitioning placers partition the placement region either vertically or horizontally to minimize the total wirelength. Through the min-cut partitioning, the partitioning placer minimizes the cut size of each cutline, and the total wirelength is minimized indirectly. (See Figure 4(a) for the traditional Manhattan cutlines.)

For the X-architecture placement, Teig and Ganley proposed X cutlines, 45-degree and 135-degree cutlines, to minimize the total X-Steiner wirelength [27]. (See Figure 4(b) for the X cutlines.) However, the study in [20] shows that using only X cutlines or combining X cutlines with Manhattan cutlines for min-cut partitioning placement cannot obtain shorter X-Steiner wirelengths.

3.2 Terminal Propagation and Weighted Net-Cut

In addition to the cutline selection, another important technique used in partitioning placers is “terminal propagation.” When a certain placement region is divided into multiple subregions, some of the cells may be strongly connected to other cells (terminals) outside the region. These terminals might significantly affect the wirelength. To consider the connection to the cells in other subregions, a possible remedy is to propagate the terminal into the nearest

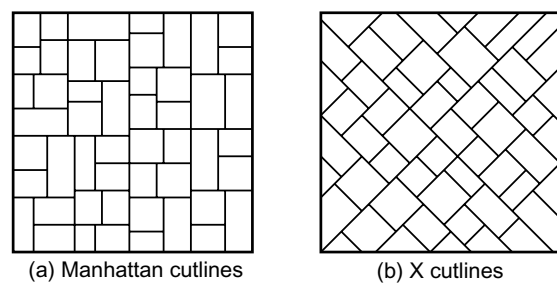


Figure 4: The Manhattan and X cutlines.

subregion and create a fixed node with a zero area in the partitioning graph [18].

Selvakkumaran and Karypis first observed the inaccuracy of the traditional terminal propagation by using the same strength for all terminals to guide the partitioning [23,24]. To fix the problem, they classified several cases according to the terminal positions and correlate the net weights to the wirelengths after partitioning. The weighted net-cut objective for min-cut partitioning is more accurate for MHPWL minimization. Recently, Chen et al. proposed a unified method to assign net weights to minimize MHPWL [8], and Roy et al. extended the method in [8] to minimize the Manhattan-Steiner wirelength for min-cut partitioning placement [22]. All the aforementioned net-weighting methods focus on minimizing the Manhattan wirelength. To minimize the wirelength for the X architecture, we propose a generalized net-weighting method, which can be incorporated into any wire models.

3.3 Generalized Net Weighting

We give our generalized net weighting as follows. A circuit is modeled as a hypergraph. Each node in the hypergraph corresponds to a cell inside the region, with the node weight being set to the area of the corresponding cell. A two- or multi-terminal net corresponds to one or two hyperedges. The hyperedge weight is set to the value of the wirelength contribution if the hyperedge is cut.

We consider a region to be divided into subregions 1 and 2. Let c_1 and c_2 be the centers of the two subregions. A net has multiple terminals $\{v_1, v_2, \dots, v_m, t_1, t_2, \dots, t_n\}$, where v_1, v_2, \dots, v_m are connected to the movable cells inside the region and t_1, t_2, \dots, t_n are fixed terminals outside the region. Let w_1 be the wirelength when all cells are in subregion 1, w_2 be the wirelength when all cells are in subregion 2, and w_{12} be the wirelength when cells are in both subregions. We assume that all cells are placed at the center of the assigned region. See Figure 5 for an illustration of a net with three terminals. We have

$$w_1 = \text{wirelength}(\{c_1, t_1, t_2, \dots, t_n\}), \quad (4)$$

$$w_2 = \text{wirelength}(\{c_2, t_1, t_2, \dots, t_n\}), \quad \text{and} \quad (5)$$

$$w_{12} = \text{wirelength}(\{c_1, c_2, t_1, t_2, \dots, t_n\}), \quad (6)$$

where $\text{wirelength}(\{p_1, p_2, \dots, p_n\})$ is the wirelength of the point set $\{p_1, p_2, \dots, p_n\}$ based on the given wire model.

We create a hypergraph G which has two *fixed pseudo nodes* to represent the two subregions and movable nodes to represent the movable cells. For a net, we introduce two hyperedges e_1 and e_2 : e_1 connects all movable nodes and the fixed pseudo node corresponding to the subregion that results in a smaller wirelength; e_2 connects between all movable nodes. We then assign the weight of the hyperedges as

$$\text{weight}(e_1) = |w_2 - w_1|, \quad (7)$$

$$\text{weight}(e_2) = w_{12} - \max(w_1, w_2). \quad (8)$$

If the net has only one movable cell, we do not need to add e_2 since it is impossible to obtain the case with cells in both regions. w_{12} is usually larger or equal to $\max(w_1, w_2)$ since separating cells into both regions often results in a larger wirelength. However, if $w_{12} < \max(w_1, w_2)$, we may make $\text{weight}(e_2) = 0$ to avoid negative edge weights for which some hypergraph partitioners cannot handle.

Partitioning the resulting hypergraph gives the partition to which the cell belongs. Let n_{cut} be the cut size of the corresponding hyperedge. We have the following theorem:

THEOREM 1. *With the generalized net weighting, the wirelength is given by $\min(w_1, w_2) + n_{cut}$ for a single net.*

PROOF. There are three possible partitioning results for a single net: (1) all nodes connected to the net are in the partition corresponding to the subregion resulting in the smallest wirelength (i.e. $\min(w_1, w_2)$), (2) all nodes connected to the net are in the other partition, and (3) nodes connected to the net are in the two different partitions. Without loss of generality, we use Figures 5(d), (e), and (f) to represent the respective cases (1), (2), and (3), and the three partitioning results correspond to the configurations shown in Figures 5(a), (b), and (c), respectively.

For easier explanation, we consider a 3-terminal net with one fixed terminal and two movable cells. (Note that the following claims still hold for other cases.) We compute the three wirelength values, w_1 , w_2 , and w_{12} , according to the aforementioned equations. In this case, we assume $w_1 < w_2$, so e_1 connects the fixed node corresponding to subregion 1. In Figure 5(a), the two cells are at the left side, and the resulting wirelength is w_1 . w_2 gives the wirelength when the two cells are both at the other side (i.e., the right side for the example shown in Figure 5(b)). Similarly, w_{12} gives the wirelength when the two cells are at different sides (see Figure 5(c)). For the case of Figure 5(d), no hyperedge in the resulting hypergraph is cut. Therefore, its cut size $n_{cut} = 0$. In Figure 5(e), e_1 is cut, and the cut size is given by $n_{cut} = \text{weight}(e_1) = |w_2 - w_1| = w_2 - w_1$. In Figure 5(f), both e_1 and e_2 are cut, and thus the cut size $n_{cut} = \text{weight}(e_1) + \text{weight}(e_2) = (w_{12} - w_2) + (w_2 - w_1) = w_{12} - w_1$. For all the three cases, we conclude that the corresponding wirelength is given by $\min(w_1, w_2) + n_{cut}$ ($w_1 + 0$, $w_1 + (w_2 - w_1)$, and $w_1 + (w_{12} - w_1)$ for the three cases, respectively). The claims are similar for the cases with different terminal numbers. \square

Further, we have the following theorem:

THEOREM 2. *The generalized net weight exactly maps the wirelength (based on the given wire model) to the min-cut cost.*

PROOF. Let wirelength_i be the wirelength of net i , $w_{1,i}$ ($w_{2,i}$) be the wirelength of net i when its cells are all at the side closer to subregion 1 (2), and $n_{cut,i}$ be the cut size of net i . By Theorem 1, we have

$$\begin{aligned} & \min \left(\sum \text{wirelength}_i \right) \\ &= \min \left(\sum (\min(w_{1,i}, w_{2,i}) + n_{cut,i}) \right) \\ &= \sum \min(w_{1,i}, w_{2,i}) + \min \left(\sum n_{cut,i} \right). \end{aligned} \quad (9)$$

Since $\sum \min(w_{1,i}, w_{2,i})$ is a constant, minimizing the wirelength of a net is equivalent to minimizing the cut size of the net, as long as the wire model and the external terminals are given. Therefore, the generalized net weight exactly maps the wirelength to the min-cut cost. \square

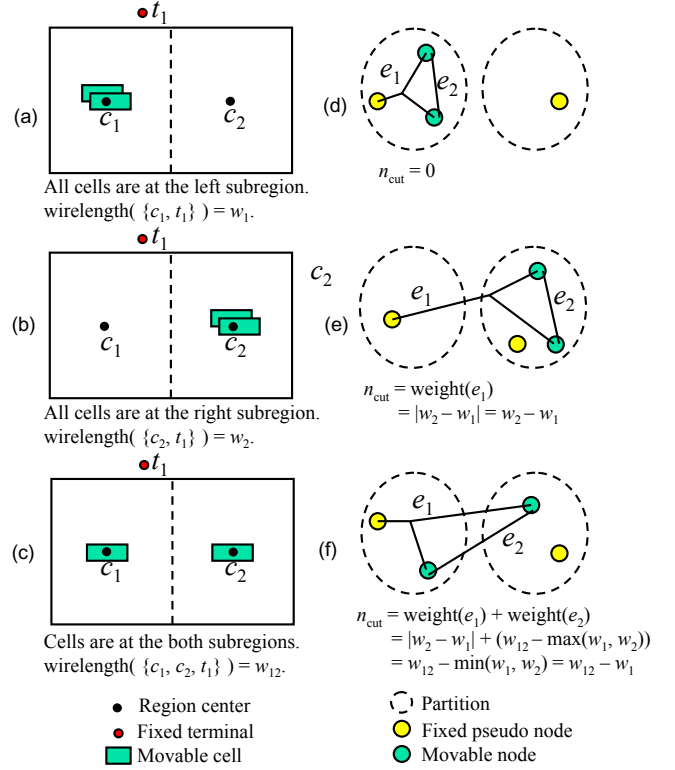


Figure 5: An example of determining a net weight. (a), (b), and (c) are three possible partitioning results. (d), (e), and (f) are corresponding partitioning hypergraphs.

3.4 X-Steiner Wirelength Minimization

Unlike the method of using X cutlines to minimize the total X-Steiner wirelength, we seek for more accurate wire models to propagate terminals and assign net weights. There are two models that can be used in the generalized net-weighting method to minimize the total X-Steiner wirelength for min-cut partitioning placement:

1. the XHPWL model and
2. the X-Steiner wirelength (XStWL) model.

For the XHPWL model, we can use Equation 2 as the wirelength function to evaluate w_1 , w_2 , and w_{12} for each net and assign net weights according to the method described in Section 3.3. For the XStWL model, we need to construct X-Steiner trees to evaluate w_1 , w_2 , and w_{12} for each net. Due to the Steiner-tree construction, using the XStWL model will take more running time than using the XHPWL one. However, the XStWL model is more accurate than the XHPWL model, so we can expect that the placement using XStWL will result in smaller X wirelengths than using XHPWL. The experimental results to be presented in Section 5 confirm this observation.

4. X-ARCHITECTURE ANALYTICAL PLACEMENT

In this section, we introduce the analytical placement framework and explain the method for incorporating the XHPWL model into the analytical placement. Then, we detail the difference between using the XHPWL model and the MHPWL one.

4.1 Analytical Placement Framework

We apply the force-directed technique for the analytical placement. The interconnection between cells provides wire forces to pull cells together and minimize the total wirelength. Considering the wire forces alone, however, cannot always obtain legal placement due to large amounts of overlaps. Consequently, we need to add spreading forces to remove the overlaps between cells.

The analytical placement is usually solved in an iterative fashion. The placement process minimizes the total wirelength and gradually adds more spreading forces until cells evenly spread to the whole chip.

A smoothed wirelength function is necessary to effectively optimize the wirelength using analytical solvers. Based on the traditional Manhattan placement, several smooth MHPWL approximation functions are proposed, such as the quadratic wirelength [11,18], the L_p -norm wirelength [6,15], and the log-sum-exp (LSE) wirelength [5, 10, 14, 19]. The LSE wirelength function for MHPWL,

$$\begin{aligned} \text{MHPWL-LSE}(e) = & \quad (10) \\ & \gamma \left(\log \sum_{v_i \in e} \exp \left(\frac{x_i}{\gamma} \right) + \log \sum_{v_i \in e} \exp \left(\frac{-x_i}{\gamma} \right) + \right. \\ & \left. \log \sum_{v_i \in e} \exp \left(\frac{y_i}{\gamma} \right) + \log \sum_{v_i \in e} \exp \left(\frac{-y_i}{\gamma} \right) \right), \end{aligned}$$

proposed in [19], achieves the best results among the three smooth MHPWL functions [6]. When γ is small, the MHPWL-LSE wirelength is close to the MHPWL [19].

For X-architecture analytical placement, we need to minimize the total X wirelength, instead of the total Manhattan wirelength. Thus we shall change the wire model from MHPWL to XHPWL.

4.2 Smoothing the XHPWL Function

To facilitate XHPWL optimization, we use log-sum-exp (LSE) functions to smooth the XHPWL function in Equation 2. The smoothed XHPWL function is shown in the following:

$$\begin{aligned} \text{XHPWL-LSE}(e) = & \quad (11) \\ & \gamma (\sqrt{2} - 1) \left(\log \sum_{v_i \in e} \exp \left(\frac{x_i}{\gamma} \right) + \log \sum_{v_i \in e} \exp \left(\frac{-x_i}{\gamma} \right) + \right. \\ & \left. \log \sum_{v_i \in e} \exp \left(\frac{y_i}{\gamma} \right) + \log \sum_{v_i \in e} \exp \left(\frac{-y_i}{\gamma} \right) \right) - \\ & \gamma \left(\frac{\sqrt{2}}{2} - 1 \right) \left(\log \sum_{v_i \in e} \exp \left(\frac{x_i + y_i}{\gamma} \right) + \right. \\ & \left. \log \sum_{v_i \in e} \exp \left(\frac{-x_i - y_i}{\gamma} \right) + \log \sum_{v_i \in e} \exp \left(\frac{x_i - y_i}{\gamma} \right) + \right. \\ & \left. \log \sum_{v_i \in e} \exp \left(\frac{-x_i + y_i}{\gamma} \right) \right). \end{aligned}$$

This function has similar property to the MHPWL-LSE function: when γ is sufficiently small, the XHPWL-LSE wirelength is close to the XHPWL.

Comparing the HPWL-LSE function and the XHPWL-LSE function, the XHPWL-LSE function needs to compute four more terms, $\log \sum \exp \left(\frac{x_i + y_i}{\gamma} \right)$, $\log \sum \exp \left(\frac{-x_i - y_i}{\gamma} \right)$, $\log \sum \exp \left(\frac{x_i - y_i}{\gamma} \right)$, and $\log \sum \exp \left(\frac{-x_i + y_i}{\gamma} \right)$. This is the main reason why the placement using the XHPWL-LSE

function takes more running time than that using the MHPWL-LSE one. However, since the wire-force evaluation is only a small part of the analytical placement process, the total runtime overhead is typically not too much. In Section 5.3, we will show that XHPWL-LSE incurs only 12% more CPU time on average.

4.3 Wire Forces of the XHPWL Model

The wire-force direction is given by the gradient direction of the wirelength function. Thus, for a terminal v_j of a net e at the coordinate (x_j, y_j) , its gradient along the x - and y -directions can be computed by the following equations:

$$\begin{aligned} \frac{\partial \text{XHPWL-LSE}(e)}{\partial x_j} = & \quad (12) \\ & (\sqrt{2} - 1) \left(\frac{\exp \left(\frac{x_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{x_i}{\gamma} \right)} + \frac{\exp \left(\frac{-x_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{-x_i}{\gamma} \right)} \right) + \\ & \left(\frac{\sqrt{2}}{2} - 1 \right) \left(\frac{\exp \left(\frac{x_j + y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{x_i + y_i}{\gamma} \right)} - \right. \\ & \left. \frac{\exp \left(\frac{-x_j - y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{-x_i - y_i}{\gamma} \right)} + \frac{\exp \left(\frac{x_j - y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{x_i - y_i}{\gamma} \right)} - \right. \\ & \left. \frac{\exp \left(\frac{-x_j + y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{-x_i + y_i}{\gamma} \right)} \right), \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \text{XHPWL-LSE}(e)}{\partial y_j} = & \quad (13) \\ & (\sqrt{2} - 1) \left(\frac{\exp \left(\frac{y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{y_i}{\gamma} \right)} + \frac{\exp \left(\frac{-y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{-y_i}{\gamma} \right)} \right) + \\ & \left(\frac{\sqrt{2}}{2} - 1 \right) \left(\frac{\exp \left(\frac{x_j + y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{x_i + y_i}{\gamma} \right)} - \right. \\ & \left. \frac{\exp \left(\frac{-x_j - y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{-x_i - y_i}{\gamma} \right)} - \frac{\exp \left(\frac{x_j - y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{x_i - y_i}{\gamma} \right)} + \right. \\ & \left. \frac{\exp \left(\frac{-x_j + y_j}{\gamma} \right)}{\sum_{v_i \in e} \exp \left(\frac{-x_i + y_i}{\gamma} \right)} \right). \end{aligned}$$

The wire forces are along the gradient directions toward the interior of the bounding box.

To illustrate the different effects of using the MHPWL and the XHPWL models, we use the following simple example. Considering four terminals, A , B , C , and D , of a net in Figure 6(a), the Manhattan bounding box is shown in the dashed lines. The wire forces using the MHPWL-LSE function are shown using arrows. It should be noted that the terminal B does not have any wire force since moving the terminal B cannot reduce the size of the Manhattan bounding box.

Using the XHPWL-LSE function, we have the bounding box and the wire forces shown in Figure 6(b). Compared with Figure 6(a), the terminal B does have a non-zero wire force, and the wire-force directions of the terminals C and D also change due to the XHPWL-LSE function. This is the reason why the XHPWL-LSE function can effectively

reduce the size of the X bounding box and obtain smaller total X wirelengths for the X-architecture placement.

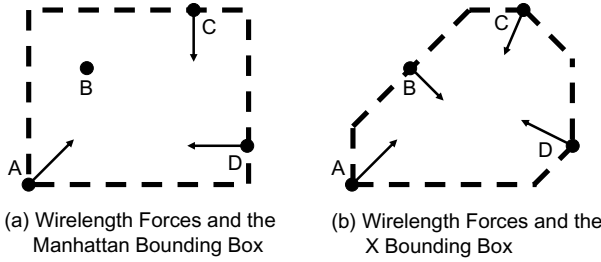


Figure 6: The wire-force directions of different bounding boxes.

5. EXPERIMENTAL RESULTS

We applied different wire models for both min-cut partitioning [9] and analytical placers [10]. For the min-cut partitioning placer, we have three wire models, MHPWL, XHPWL, and XStWL. For the analytical placer, we have two wire models, MHPWL and XHPWL. We do not use the XStWL model for analytical placement since there exists no publicly available method to minimize the X-Steiner wirelength currently. All experiments were performed on an AMD Opteron 2.6GHz machine. For both the min-cut partitioning and the analytical placers, they consist of three major stages: (1) global placement, (2) legalization, and (3) detailed placement. Our main purpose is to compare the effects of using the wire models. Therefore, we keep the global placement and the legalization stages, and disable the detailed placement for fair comparisons since the detailed placement is based on the MHPWL model.

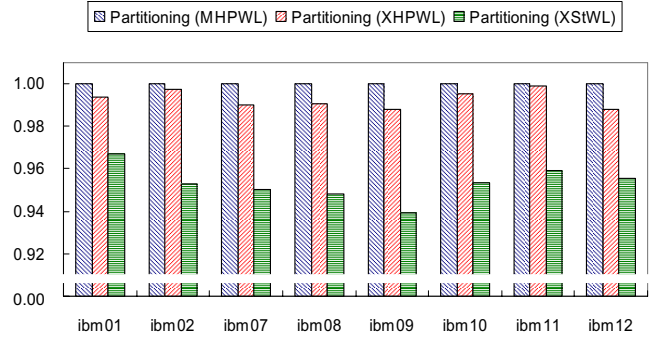
The benchmarks we used, “IBM version 2.0”, are the same as those used in [20]. There are totally 8 circuits in this benchmark suite which is widely used in academia [1].

5.1 Comparisons with Traditional Placers

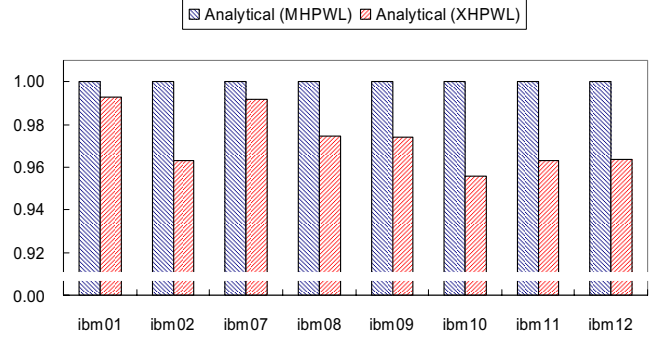
We first show that our placers are comparable to other recent academic placement tools, including Feng Shui 5.1 [2], Capo 10.2 [21], APlace 2.0 [15], and mPL6 [5]. Table 1 shows the Manhattan-half-perimeter wirelengths (MHPWLs) for all placers. In this table, we also report the MHPWLs of our placers using different placement algorithms and wire models. The results show that our placers are comparable to other recent placers. Note that it is reasonable that the MHPWL’s for our placers increase by using the XHPWL and XStWL models since the objective functions of XHPWL and XStWL do not optimize the MHPWL’s directly.

5.2 Total Steiner Wirelength Comparisons

We use the total Steiner wirelength to evaluate the quality of the placement. Compared with the half-perimeter wirelength, the Steiner wirelength is much closer to the routed wirelength. The results are shown in Table 2. The left part reports the total Manhattan-Steiner wirelengths while the right part reports the total X-Steiner ones. The average values are normalized to the respective placement algorithms using the MHPWL model. We observe that the average total Manhattan-Steiner wirelengths do not change much when different wire models are used. For the total X-Steiner wirelength, to show the effect of different wire models, we depict the normalized X-Steiner wirelengths for the min-cut partitioning placers and analytical placers in Figure 7. Compared with the MHPWL model, the X-Steiner



(a) Partitioning Placement



(b) Analytical Placement

Figure 7: Normalized Total X-Steiner Wirelengths

wirelengths are reduced by 1% and 5% for min-cut partitioning placement using the XHPWL model and the XStWL model, respectively. For analytical placement, the X-Steiner wirelength is reduced by 3% on average, compared with the MHPWL model. Figure 8 shows the resulting placement of ibm01 using our analytical placer with the XHPWL model.

5.3 CPU Time Comparison

Table 3 gives the total CPU times for different algorithms and wire models. Since it needs more computation efforts for the XHPWL model than the MHPWL model, the XHPWL model incurs average runtime overheads of about 8% and 15% for min-cut partitioning and analytical algorithms, respectively. The average runtime overhead for the XStWL model is the highest, 22%, due to the Steiner-tree construction.

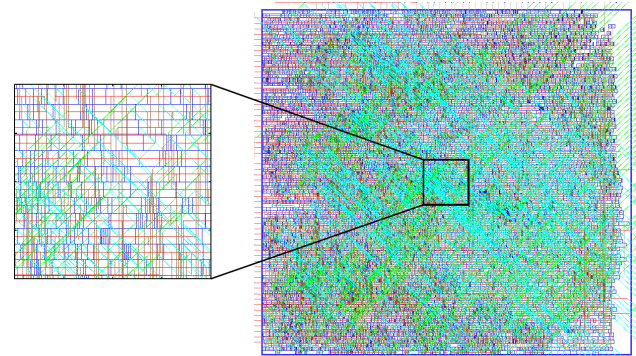


Figure 8: The resulting placement of ibm01 using our analytical placer with the XHPWL model.

Table 1: The resulting Manhattan-half-perimeter wirelengths (MHPWLs) from different placement algorithms and wire models. The results show that our placer is comparable to other recent works. Our placer has both min-cut partitioning and analytical modes. Three wire models (MHPWL, XHPWL, and XStWL) for our min-cut partitioning placer and two wire models (MHPWL and XHPWL) for our analytical placers are available. The average values are normalized to our results using the same placement algorithm with the MHPWL model. Note that it is reasonable that the MHPWLs for our placers increase by using the XHPWL and XStWL models since the objective functions of XHPWL and XStWL do not optimize the MHPWLs directly.

Total Manhattan Half-Perimeter Wirelength (MHPWL) ($\times e8$)									
Algorithm	Min-Cut Partitioning					Analytical			
Placer	Ours			Feng Shui 5.1	Capo 10.2	Ours		APlace 2.0	mPL6
Wire Model	MHPWL	XHPWL	XStWL	MHPWL	MHPWL	MHPWL	XHPWL	MHPWL	MHPWL
ibm01	0.52	0.55	0.57	0.54	0.55	0.50	0.52	0.48	0.49
ibm02	1.54	1.58	1.65	1.54	1.50	1.38	1.39	1.34	1.44
ibm07	3.48	3.56	3.66	3.28	3.48	3.11	3.32	3.08	3.04
ibm08	3.68	3.78	3.94	3.75	3.78	3.36	3.42	3.26	3.31
ibm09	3.18	3.21	3.30	3.14	3.13	2.79	2.86	2.79	2.80
ibm10	6.06	6.20	6.34	5.77	5.93	5.34	5.58	5.18	5.28
ibm11	4.67	4.83	4.96	4.73	4.64	4.18	4.33	4.25	4.22
ibm12	8.07	8.24	8.56	7.78	7.99	7.14	7.32	7.15	7.05
Average	1.00	1.03	1.06	0.99	1.00	1.00	1.03	0.99	1.00

Table 2: Comparison of the resulting total Manhattan-Steiner wirelengths and total X-Steiner wirelengths based on different placement algorithms and different wire models. The average values are normalized to the respective placement algorithms using the MHPWL model.

Total Manhattan-Steiner Wirelength ($\times e8$)						Total X-Steiner Wirelength ($\times e8$)				
Algorithm	Min-Cut Partitioning			Analytical		Min-Cut Partitioning			Analytical	
Wire Model	MHPWL	XHPWL	XStWL	MHPWL	XHPWL	MHPWL	XHPWL	XStWL	MHPWL	XHPWL
ibm01	0.62	0.62	0.62	0.57	0.58	0.57	0.57	0.55	0.53	0.52
ibm02	1.81	1.84	1.80	1.61	1.59	1.68	1.68	1.60	1.50	1.45
ibm07	3.89	3.95	3.88	3.66	3.76	3.60	3.56	3.42	3.35	3.32
ibm08	4.35	4.41	4.31	3.99	3.99	4.02	3.98	3.81	3.66	3.57
ibm09	3.61	3.63	3.53	3.25	3.29	3.32	3.28	3.12	2.98	2.90
ibm10	6.80	6.92	6.77	6.29	6.23	6.27	6.24	5.97	5.81	5.55
ibm11	5.12	5.26	5.18	4.74	4.74	4.72	4.71	4.52	4.32	4.16
ibm12	9.06	9.19	9.09	8.48	8.49	8.35	8.25	7.98	7.78	7.50
Average	1.00	1.01	1.00	1.00	1.01	1.00	0.99	0.95	1.00	0.97

Table 3: Comparison of the CPU times for different placement algorithms and wire models. The average values are normalized to the respective placement algorithms using the MHPWL model.

Total CPU Time (sec)					
Algorithm	Min-Cut Partitioning			Analytical	
Wire Model	MHPWL	XHPWL	XStWL	MHPWL	XHPWL
ibm01	33	36	41	29	46
ibm02	65	81	98	81	84
ibm07	200	206	244	350	380
ibm08	239	254	299	350	367
ibm09	209	213	227	398	386
ibm10	380	382	406	538	596
ibm11	304	321	347	634	865
ibm12	366	402	452	644	648
Average	1.00	1.08	1.22	1.00	1.15

Table 4: The normalized wirelength under different placement and routing architectures. The average values are normalized to the respective placement algorithms.

Normalized Wirelength						
Algorithm	Min-Cut Partitioning			Analytical		
Placement	M-Arch	M-Arch	X-Arch	M-Arch	M-Arch	X-Arch
Routing	M-Arch	X-Arch	X-Arch	M-Arch	X-Arch	X-Arch
ibm01	1.00	0.92	0.89	1.00	0.93	0.90
ibm02	1.00	0.93	0.89	1.00	0.93	0.91
ibm07	1.00	0.92	0.88	1.00	0.91	0.88
ibm08	1.00	0.92	0.88	1.00	0.92	0.89
ibm09	1.00	0.92	0.88	1.00	0.92	0.88
ibm10	1.00	0.92	0.88	1.00	0.92	0.89
ibm11	1.00	0.92	0.87	1.00	0.91	0.88
ibm12	1.00	0.92	0.88	1.00	0.92	0.88
Average	1.00	0.92	0.88	1.00	0.92	0.89

5.4 Wirelength Using Different Architectures

We summarize in Table 4 the wirelength reductions using the Manhattan architecture and the X architecture for placement and routing. All results are normalized to those using the “traditional” Manhattan-architecture placement and routing. Without our X-architecture placement, the X-architecture routing alone reduces the wirelength by only 8% on average. With our X-architecture placement, the X-architecture routing can reduce the wirelength by 12% and 11% on average for min-cut partitioning and analytical placement algorithms, respectively.

The results reveal the effectiveness of the X architecture on wirelength reduction during placement, which is different from the results given in the previous work [20] that the X-architecture placement does not improve the X-routing wirelength over the Manhattan-architecture placement.

6. CONCLUSIONS

We have proposed the XHPWL model that can be used in both min-cut partitioning and analytical placement for the X architecture. We have also studied the XHPWL and the XStWL models for min-cut partitioning placement and the XHPWL model for analytical placement. Experimental results have shown that using the XHPWL or the XStWL model in placement can lead to shorter X-Steiner wirelengths than traditional Manhattan placement. Without X-placement, X-routing alone would reduce less wirelength than X-routing with X-placement. The results reveal the effectiveness of the X architecture on wirelength reduction during placement and thus the importance of the study on the X-placement algorithms.

7. ACKNOWLEDGMENTS

This work was partially supported by MediaTek Inc. and National Science Council of Taiwan under Grant No’s NSC 95-2221-E-002-372, NSC 95-2221-E-002-374, NSC 95-2752-E-002-008-PAE.

8. REFERENCES

- [1] S. N. Adya, M. C. Yildiz, I. L. Markov, P. G. Villarrubia, P. N. Parakh, and P. H. Madden. Benchmarking for large-scale placement and beyond. *IEEE Trans. Computer-Aided Design*, 23(4):472–487, 2004.
- [2] A. R. Agnihotri, S. Ono, and P. H. Madden. Recursive bisection placement: Feng Shui 5.0 implementation details. In *Proceedings of ACM International Symposium on Physical Design*, pages 230–232, San Francisco, CA, Apr. 2005.
- [3] C. J. Alpert, J.-H. Huang, and A. B. Kahng. Multilevel circuit partitioning. *IEEE Trans. Computer-Aided Design*, 17(8):655–667, Aug. 1998.
- [4] Z. Cao, T. Jing, Y. Hu, Y. Shi, X. Hong, X. Hu, and G. Yan. DraXRouter: global routing in X-architecture with dynamic resource assignment. In *Proceedings of IEEE/ACM Asia South Pacific Design Automation Conference*, pages 618–623, Yokohama, Japan, Jan. 2006.
- [5] T. Chan, J. Cong, J. Shinnerl, K. Sze, and M. Xie. mPL6: Enhanced multilevel mixed-size placement. In *Proceedings of ACM International Symposium on Physical Design*, pages 212–214, San Jose, CA, Apr. 2006.
- [6] T. Chan, J. Cong, and K. Sze. Multilevel generalized force-directed method for circuit placement. In *Proceedings of ACM International Symposium on Physical Design*, pages 185–192, San Francisco, CA, Apr. 2005.
- [7] H. Chen, C.-K. Cheng, A. B. Kahng, I. Mandoiu, and Q. Wang. Estimation of wirelength reduction for λ -geometry vs. Manhattan placement and routing. In *Proceedings of System Level Interconnect Prediction Workshop*, pages 71–76, Monterey, CA, Apr. 2003.
- [8] T.-C. Chen, Y.-W. Chang, and S.-C. Lin. IMF: Interconnect-driven multilevel floorplanning for large-scale building-module designs. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pages 159–164, San Jose, CA, Nov. 2005.
- [9] T.-C. Chen, T.-C. Hsu, Z.-W. Jiang, and Y.-W. Chang. NTUplace: a ratio partitioning based placement algorithm for large-scale mixed-size designs. In *Proceedings of ACM International Symposium on Physical Design*, pages 236–238, San Francisco, CA, Apr. 2005.
- [10] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, and Y.-W. Chang. A high-quality mixed-size analytical placer considering preplaced blocks and density constraints. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, Nov. 2006.
- [11] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 269–274, June 1998.
- [12] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 175–181, 1982.
- [13] T.-Y. Ho, C.-F. Chang, Y.-W. Chang, and S.-J. Chen. Multilevel full-chip routing for the X-based architecture. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 597–602, Anaheim, CA, June 2005.
- [14] A. B. Kahng and Q. Wang. Implementation and extensibility of an analytic placer. *IEEE Trans. Computer-Aided Design*, 24(5), May 2005.
- [15] A. B. Kahng and Q. Wang. A faster implementation of APlace. In *Proceedings of ACM International Symposium on Physical Design*, pages 218–220, San Jose, CA, Apr. 2006.
- [16] G. Karypis and V. Kumar. Multilevel k-way hypergraph partitioning. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 343–348, New Orleans, LA, June 1999.
- [17] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, Feb. 1970.
- [18] M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich. Gordian: VLSI placement by quadratic programming and slicing optimization. *IEEE Trans. Computer-Aided Design*, 10(3):356–365, 1991.
- [19] W. C. Naylor, R. Donnelly, and L. Sha. US patent 6,301,693: Non-linear optimization system and method for wire length and dealy optimization for an automatic electric circuit placer. 2001.
- [20] S. Ono, S. Tilak, and P. H. Madden. Bisection based placement for the X architecture. In *Proceedings of IEEE/ACM Asia South Pacific Design Automation Conference*, Yokohama, Japan, Jan. 2007 (to appear).
- [21] J. Roy, D. Papa, A. Ng, and I. Markov. Satisfying whitespace requirements in top-down placement. In *Proceedings of ACM International Symposium on Physical Design*, pages 206–208, San Jose, CA, Apr. 2006.
- [22] J. A. Roy, J. F. Lu, and I. L. Markov. Seeing the forest and the trees: Steiner wirelength optimization in placement. In *Proceedings of ACM International Symposium on Physical Design*, San Francisco, CA, Apr. 2005.
- [23] N. Selvakkumaran and G. Karypis. Theto - a fast and high-quality partitioning driven global placer. Technical Report 03-46, Dept of Computer Science and Engineering, Univeristy of Minnesota, Nov. 2003.
- [24] N. Selvakkumaran and G. Karypis. Theto - a fast and high-quality partitioning driven placement tool. Technical Report 04-40, Dept of Computer Science and Engineering, Univeristy of Minnesota, Oct. 2004.
- [25] T. Taghavi, X. Yang, B.-K. Choi, M. Wang, and M. Sarrafzadeh. Dragon2006: Blockage-aware congestion-controlling mixed-size placer. In *Proceedings of ACM International Symposium on Physical Design*, pages 209–211, San Jose, CA, Apr. 2006.
- [26] S. L. Teig. The X architecture: not your father’s diagonal wiring. In *Proceedings of System Level Interconnect Prediction Workshop*, pages 33–38, San Diego, CA, Apr. 2002.
- [27] S. L. Teig and J. L. Ganley. US patent 6,848,091: Partitioning placement method and apparatus. 2002.