

## Partition-Based Global Placement Considering Wire-Density Uniformity for CMP Variations\*

DONG Changdao (董昌道), ZHOU Qiang (周强)\*\*, CAI Yici (蔡懿慈), LIU Dawei (刘大为)

Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology,  
Tsinghua University, Beijing 100084, China

**Abstract:** This paper presents a multilevel hypergraph partitioning method that balances constraints on not only the cell area but also the wire weight with a partition-based global placement algorithm that maximizes the wire density uniformity to control chemical-mechanical polishing (CMP) variations. The multilevel partitioning alternately uses two FM variants in the refinement stage to give a more uniform wire distribution. The global placement is based on a top-down recursive bisection framework. The partitioning algorithm is used in the bisectioning to impact the wire density uniformity. Tests show that, with a 10% constraint, the partitioning produces solutions with more balanced edge weights that are 837% better than from hMetis, 1039.1% better than MLPart, and 762.9% better than FM in terms of imbalance proportion and that this global placement algorithm improves ROOSTER with a more uniform wire distribution by 3.1% on average with an increased wire length of only 3.0%.

**Key words:** partitioning; placement; chemical-mechanical polishing (CMP); design for manufacturing (DFM); wire-density

### Introduction

As nanometer technologies advance, the chemical-mechanical polishing (CMP) step in the copper metallization (Damascene) process may cause some loss of interconnections. A non-uniform feature density distribution in one interconnect layer causes the CMP to over or under polish, so thickness variations may accumulate in each layer and finally be very significant<sup>[1,2]</sup>. Dummy features are usually filled into layouts to restrict the variations in each layer<sup>[3]</sup>. However, dummy features may degrade interconnect performance and cause mask data explosion<sup>[4]</sup>. Therefore, to

minimize the side effects of aggressive post-layout dummy filling, the wire density uniformity must be considered in the early design stages such as for the global placement.

Several CMP-aware methods have been developed for the early design stages<sup>[5]</sup>. Cho et al.<sup>[6]</sup> proposed a predictive copper (Cu) CMP model to evaluate topography variations to guide CMP-aware global routing. Chen et al.<sup>[7]</sup> proposed a wire density driven full-chip routing which considers not only the wire density inside a routing gcell but also its gradient between routing gcells. Chen et al.<sup>[8]</sup> proposed a metal-density driven placement algorithm based on an analytical framework which improves the metal-density uniformity by 3% while increasing the wire length by 4% compared with a cell-density driven placement. Chen et al.<sup>[8]</sup> described a probabilistic routing model to estimate the wire density and the metal density and thickness using by the CMP model proposed in Cho et al.<sup>[6]</sup>

---

Received: 2009-04-27; revised: 2010-12-08

\* Supported by the National Natural Science Foundation of China (Nos. 60876026 and 60833004)

\*\* To whom correspondence should be addressed.

E-mail: zhouqiang@tsinghua.edu.cn; Tel: 86-10-62785564

The wire density distribution will determine the layout pattern density<sup>[6]</sup>, which closely correlates to the post-CMP dielectric thickness<sup>[4]</sup>, so it is desirable to minimize the global variation of the wire density<sup>[7]</sup>. Since placement is a crucial design stage which strongly impacts the subsequent routing results, a uniformly distributed wire density during placement becomes very important for CMP variation control. The wire density is closely related to routing congestion, with many congestion driven placement algorithms proposed to improve routability. However, the wire density is not the same as routing congestion, so routability driven placement may only contribute modestly to the wire distribution uniformity.

ROOSTER<sup>[9]</sup>, a state-of-the-art congestion driven placement algorithm, recursively applies multilevel hypergraph partitioning, such as MLPart<sup>[10]</sup> and hMetis<sup>[11]</sup>, to bisection large bins containing hundreds or more standard cells during the global placement stage. hMetis<sup>[11]</sup> is one of the most well-known multilevel partitioning algorithms, which produces solutions with nearly minimized cut sizes. Of the three clustering methods evaluated by Karypis et al.<sup>[11]</sup>, the FM variant using random vertex selection was used in the refinement stage of the multilevel paradigm. MLPart<sup>[10]</sup> is another leading-edge multilevel partitioning algorithm comparable with hMetis. MLPart employs the PinEC scheme to speed up pin removal during vertice clustering and CLIP<sup>[12]</sup> to improve the refinement during the initial partitioning and uncoarsening stage. Although the wire distribution between the two sub-bins is largely determined after each bisection, its uniformity is not a significant issue in traditional partitioning methods. MLPart and hMetis often produce solutions with highly unbalanced numbers of edges between the two blocks.

The motivation of this work is to design a multilevel hypergraph partitioning algorithm that balances the wire distribution and then apply this algorithm to the recursive bisection in global placement to generate results with more uniformly distributed wire densities for better CMP variation control. The multilevel partitioning method alternates between an FM variant capable of balancing edge weights between the two blocks and another FM variant capable of minimizing the cut size for tight edge weight constraints in the refinement stage. The first FM variant generates solutions

satisfying the edge weight constraint and minimizes the imbalance with a time cost almost linearly related to the number of pins in the circuit. The second FM variant minimizes the cut size for the given edge weight constraint. The inheritance of edge weights by corresponding clusters is also considered in the coarsening stage. The global placement framework uses the wire weight balancing constraint in the partitioning. A multilevel hypergraph partitioning with dual balancing constraints on both the cell area and the wire weight is recursively performed in the global placement. The wire weight is assigned such that it estimates the wire distribution in each placement bin. Tests show that the algorithm is capable of generating solutions with more uniformly distributed wire densities and outperforms ROOSTER in terms of the standard deviation of the horizontal and vertical wire density of routing gcells, with a small increment in the total wire length.

## 1 Preliminaries

### 1.1 Hypergraph partitioning with constraints on the wire density balancing weight

Formally, a hypergraph  $H = (V, E)$  is defined as a set of vertices  $V$  and a set of edges  $E$ , where each edge is also called a hyperedge and is a subset of the vertex set  $V$ . A vertex is said to be incident to an edge, if  $v \in e$ . Each vertex and edge has one or more weights associated with it. The hypergraph partitioning is a decomposition of  $V$  into two disjoint subsets  $V_1$  and  $V_2$ , such that  $\cup V_i = V$ . Each subset is called one partitioning block. An edge  $e$  is said to be cut if  $\{v | v \in V_1 \cap v \in e\} \neq \emptyset \cap \{v | v \in V_2 \cap v \in e\} \neq \emptyset$ .

This analysis associates each vertex with a weight  $w(v)$  and each edge with two weights, the cut weight,  $w_c(e)$ , and the wire density balancing weight,  $w_b(e)$ . Define the cut set  $C = \{e | e \text{ is cut}\}$  and the cut size of a partitioning  $S(C) = \sum_{e \in C} w_c(e)$ . Given the set of internal edges of block  $V_i$  defined as  $I(V_i) = \{e | e \notin C \cap (\exists v \text{ s.t. } v \in e \cap v \in V_i)\}$ ,  $i = 1, 2$ , employ the concept of total wire density balancing weight of block  $V_i$  defined as

$$W_E(V_i) = \sum_{e_i \in I(V_i)} w_b(e_i) + \sum_{e_c \in C} w_b(e_c) / 2 \quad (1)$$

Similarly, the total vertex weight of block  $V_i$  is defined as  $W_V(V_i) = \sum_{v \in V_i} w(v)$ . Given the target vertex

Download English Version:

<https://daneshyari.com/en/article/865299>

Download Persian Version:

<https://daneshyari.com/article/865299>

[Daneshyari.com](https://daneshyari.com)