A fast force-directed simulated annealing for 3D IC partitioning

Aysa Fakheri Tabrizi [a,*], Laleh Behjat [a], William Swartz [b], Logan Rakai [a]

[a] Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, Calgary, Alberta, Canada
[b] University of Texas at Dallas, Dallas, TX, USA

ARTICLE INFO

ABSTRACT

With the advent of three dimensional (3D) IC designs, new partitioning techniques that can take into account the 3D nature of designs are required. In this paper, a new force-directed simulated annealing (FSA) is introduced and used for 3D partitioning. The proposed force-directed simulated annealing introduces force as a new factor during the annealing process and replaces the random moves by probabilistic force-directed moves. Experimental results show that the force-directed move strategy speeds up the convergence and significantly improves the execution time of SA maintaining the quality of solution. FSA algorithm is effective for 3D IC partitioning and can be applied in other optimization problems.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Three dimensional (3D) integration has recently become popular since it reduces the wirelength and consequently affects delay, manufacturability, and power consumption [1], and enhances the integration [2] of the very large integrated circuits (VLSI). However, as any new technology, 3D integration introduces its own issues. Among these issues are the penalties imposed by the communication elements between tiers, called through-silicon vias (TSVs). TSVs are expensive and they consume chip area [1], hence, besides the other objectives of IC design such as wirelength, routability, and temperature, minimizing the total number of TSVs is of utmost importance. Moreover, minimizing TSVs in 3D ICs can further improve the quality of placement and routing. The reason is that TSVs are bulky and they occupy significant area in the chip. Hence, they act as obstacles during the placement and routing and can result in routability congestion which is one of the main concerns of physical design in recent years as the transistor sizes are getting smaller and smaller.

One of the most effective ways to reduce the number of TSVs is to use effective partitioning techniques during the IC physical design stage [3]. However, most of the current techniques for 3D partitioning are designed for 2D placement and not adapted to three dimensions. 2D partitioning is a well developed field and is usually performed by multi-level algorithms such as hMetis [4]

which are based on iterative improvement techniques such as Fiduccia–Mattheyses (FM) [5]. 2D iterative improvement techniques are efficient in reducing the number of connections that pass between partitions. However, they do not consider the number of times a net is cut or how many partitions exist between the end points of a net when the partitions are stacked.

In a 3D circuit, partitions are stacked on top of each other and any connection between two non-adjacent partitions results in using more than one TSV. For example, considering a 3D IC with 5 tiers, if one terminal of a net is placed in the top tier and the other terminal in the bottom tier, the net is effectively cut three times and four TSVs will be required to complete one single connection. Therefore, in this paper, the goal of 3D partitioning is defined as *to minimize the number of cuts on nets*, defined by the sum of partial nets between pairs of adjacent tiers.

Simulated Annealing (SA) is one of the methods that is widely used in physical design [3] and has been implemented in several tools [6]. SA is known as a successful method for small designs; however, it is a slow method and requires a high CPU time which is not suitable for very large scaled designs.

In this paper we introduce a new variation of SA that uses the mass–spring model, as an annealing guide, and improves the performance of simulated annealing by speeding up the annealing process. The inspiration of the technique comes from the force-directed model that has been used in many placement tools [7,8].

The main contributions of this paper are as follows: (i) Development of a fast force-directed simulated annealing which can be adapted to different optimization problems. (ii) Development of a 3D IC partitioning technique.

The rest of this paper is organized as follows. In Section 2, a

* Corresponding author.
E-mail addresses: afakheri@ucalgary.ca (A. Fakheri Tabrizi),
laleh@ucalgary.ca (L. Behjat), bill-swartz@utdallas.edu (W. Swartz),
lmrakai@ucalgary.ca (L. Rakai).

literature review on 3D IC and simulated annealing is given. In Section 3 the 3D partitioning problem is stated. In Section 4, the proposed Force-directed Simulated Annealing algorithm is described in detail. Experimental results are reported in Section 5. Finally, conclusions and future work are presented in Section 6.

2. Preliminaries

2.1. 3D IC

There are two approaches for partitioning and floorplanning of 3D IC design. The first approach performs partitioning and floorplanning simultaneously, where in the second approach these stages are performed successively. It is shown in [9] and discussed in [10] that the first method is inefficient due to its very large solution space and the second method has a better performance.

In this paper we perform the partitioning stage of the successive partitioning and placement approach that aims to reduce the TSV number. Further, the proposed partitioning can be followed by a thermal-aware placement for each tier and the wirelength and temperature can be optimized during this process [11–16].

In successive partitioning and placement, there might be tradeoffs between TSVs and wirelength. One way to decide on the limit on number of TSVs is to generate a TSV#-WL trade-off curve and select the favorable design such as discussed in [10].

Several 3D computer-aided design tools and algorithms exist. A 3D Place and Route (TPR) CAD tool that performs 3D physical design is presented in [17]. This tool uses hMetis [4] to perform partitioning and divides a netlist, i.e. circuit elements (cells) and their connections (nets), to the number of partitions equal to the number of desired tiers. After dividing a circuit to tiers, cells are placed and routed in each tier. In [18,19], Alternate Pins method aims to provide an optimal solution by balancing the I/Os; the UnlockedPins method uses hMetis to partition the I/Os as free nodes aiming TSV minimization; and in I/O Pin method, 3D partitioning is performed by first partitioning and placing the Input/Output (I/O) pins and then partitioning the cells based on the I/O locations.

A min-cost maximum flow based algorithm for TSV planning and pin assignment is introduced in [20] that considers total wirelength and thermal optimization and a discussion on the TSV resource sharing possibilities and new algorithms for TSV allocation and optimization are presented in [21].

2.2. Simulated annealing

Simulated Annealing (SA) is an iterative meta-heuristic for solving nonlinear and non-convex optimization problems. It is first introduced in [22], based on the Metropolis algorithm [23] and different variations are introduced later [24,25]. SA is often used in combinatorial problems such as TSP [26], supply chain management [27], packing problem [28], vehicle routing [29], timetabling [30], and machine scheduling [31].

A typical SA algorithm is given in Algorithm 1. The algorithm starts from an initial partitioning solution s_{init} and the temperature parameter T is set to its initial value T_0 (lines 1–3). Then the cost of the initial solution is calculated (line 4) and the following steps are repeated until the temperature T reaches its minimum value T_{min} or a stopping criterion is satisfied (lines 5–6). A neighbor solution s' is generated and its cost is calculated (lines 7–8). The neighbor function is a function that generates a new solution by changing the value of one or more variables. If the cost of the new solution generated by the neighbor function is less than the cost of current solution, the current solution is replaced by the new solution (lines 10–12). Otherwise, this move takes effect with a probability

(lines 13–18). Then the temperature T is reduced by a factor of α (line 21).

Algorithm 1. Simulated annealing.

Input: Initial Solution s_{init}
Output: Optimized Solution
1: **Begin**
2:    $s \leftarrow s_{init}$
3:    $T \leftarrow T_0$
4:    $cost_{current} \leftarrow Cost(s)$
5:    **while** $T > T_{min}$ **do**
6:      **while** stopping criterion is not met yet **do**
7:        $s' \leftarrow$ neighbor(s)
8:        $cost_{new} \leftarrow Cost(s')$
9:        $\Delta cost = cost_{new} - cost_{current}$
10:       **if** $\Delta cost < 0$ **then**
11:         $s \leftarrow s'$
12:         $cost_{current} \leftarrow cost_{new}$
13:       **else**
14:         $r \leftarrow Random(0, 1)$
15:         **if** $r < e^{-\Delta cost/T}$ **then**
16:           $s \leftarrow s'$
17:           $cost_{current} \leftarrow cost_{new}$
18:         **end if**
19:       **end if**
20:     **end while**
21:     $T \leftarrow \alpha.T$
22:   **end while**
23: **End**.

Based on classical SA, new extended simulated annealing algorithms are introduced. In [32], authors introduce smart SA for 3D floorplanning. This algorithm uses a discrete distribution of selection based on objectives of the problem to guide the SA to make more efficient moves and performs random selection in every nth iteration to preserve the randomness of SA. In [33] an SA algorithm with adaptive cooling schedule is used for 3D IC design to effectively explore the solution space and scape local minima.

2.3. Force-directed placement

Force directed placement is a type of placement technique during which cells and nets are modelled as a mass–spring system [34]. The connected cells attract each other and the magnitude of attraction is directly proportional to the distance between the cells or the connection length. The cells, which are free to move, will move in the direction of the forces applied to them and will settle in a configuration with the minimum energy. Therefore, the placement problem is reduced to solving an optimization problem.

The force measure between two cells is inspired from Hooke's law and is given in the following equation:

$$F_{i,j} = \sum_{i=1, j=1}^{n} c(i,j)((x_i - x_j)^2 + (y_i - y_j)^2),$$

(1)

where n represents the total number of cells and $c(i,j)$ is the corresponding connection cost between the cells i and j if they are connected and 0, otherwise.

3. 3D Partitioning problem

The goal of our 3D partitioning approach is to form the pre-