Zero-skew Clock Network Synthesis for Monolithic 3D ICs with Minimum Wirelength

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Wang, W., Pavlidis, V., & Cheng, Y. (Accepted/In press). *Zero-skew Clock Network Synthesis for Monolithic 3D ICs with Minimum Wirelength*.

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

Zero-skew Clock Network Synthesis for Monolithic 3D ICs with Minimum Wirelength

Wei Wang

School of Microelectronics
Beihang University
Beijing, China
wangweiwei@buaa.edu.cn

Vasilis F. Pavlidis

School of Computer Science
University of Manchester
Manchester, UK
vasileios.pavlidis@manchester.ac.uk

Yuanqing Cheng

School of Microelectronics
Beihang University
Beijing, China
yuanqing@ieee.org

ABSTRACT

Clock network synthesis has traditionally been an important step of the physical design process, greatly affecting the performance of ICs. In this paper, we focus on the clock network design process for monolithic 3D (M3D) ICs. Firstly, we investigate the difference between Monolithic Inter-tier Via (MIV) and Through-Silicon Via (TSV) due to the different fabrication process and explore the ramifications of clock network design for monolithic 3D systems. Secondly, we develop a two step clock network synthesis algorithm (M3D-ZST) based on clustering and the deferred-merge embedding algorithm. The proposed algorithm considers the MIV characteristics and constructs a zero-skew clock tree considering wirelength optimization. Furthermore, we apply a look-ahead approach, thereby determining the optimal locations of the merging segments and MIVs such that the wirelength is reduced further (M3D-ZSTLA). Experimental results indicate that M3D-ZST algorithm reduces the total wirelength by 9.7% – 19.7%, and reduces power by 9.4% – 18.6% compared to the 3D-MMM algorithm over IBM benchmarks. The M3D-ZSTLA algorithm further decreases the total wirelength by about 3%, and reduces the power by about 2%.

KEYWORDS

Monolithic 3D ICs, clock network, clock skew, minimum wirelength

1 INTRODUCTION

The continuous advance of technology nodes, the sharply increasing complexity and cost of fabrication process challenges Moore's law [7]. 3D integration was proposed and has been shown to be an effective paradigm to extend Moore's law by stacking chips with abundant short vertical interconnects [19]. The paradigm of 3D integration can mainly be categorized into 2.5D system integration, TSV-based 3D ICs, and monolithic 3D (M3D) ICs [15]. 2.5D integration is an intermediate transition from 2D ICs to 3D ICs. TSV-based 3D ICs can provide higher integration density and communication bandwidth compared with 2.5D integration. However, it also has some limitations. Firstly, the manufacturing process of TSV is not as mature as that of CMOS, and the additional process cost is significant. Secondly, the diameter of TSV is several micrometers, and in order to reduce the influence of mechanical stress on CMOS devices, a Keep-Out-Zone (KOZ) is required to separate the TSV from surrounding devices, which further increases the area overhead. Thirdly, the size of TSV does not scale in the same pace as the CMOS devices due to the limited alignment accuracy.

Recently, monolithic 3D ICs have been proposed to overcome the drawbacks of TSV-based 3D ICs [1]. With the sequential process, the vertical interconnects, i.e., monolithic inter-tier vias (MIVs), can be smaller by 1-2 orders of magnitude compared to TSVs, enabling much higher integration density. Among the unexplored physical design issues, the problem of clock network synthesis for M3D circuits has not been adequately addressed as previous works have emphasized TSV-based 3D ICs [12, 21]. In TSV-based 3D ICs, the clock network can be partitioned on several tiers such that the planar wirelength is reduced and the clock power and frequency can be improved thereafter. However, the large size of TSV induces a large overhead in area, and large *RC* parasitics. Therefore, the number of TSVs is usually strictly constrained during the clock network design [21]. Meanwhile, the MIV is much smaller than the TSV, and the abundance of MIVs allows a more aggressive strategy to provide shorter wirelength, better performance, and lower power. Therefore, it is essential to propose new clock network synthesis algorithms for M3D ICs.

In this paper, we develop a two step clock network synthesis algorithm for M3D ICs to construct a zero-skew clock tree with minimal total wirelength. The contributions of this paper are as follows:

- We investigate the differences between MIVs and TSVs when utilized for 3D clock network design. We demonstrate that the direct application of TSV-3D IC clock network synthesis to M3D ICs may not exploit the full benefits of M3D ICs and results in sub-optimal solutions.
- To synthesize clock networks based on M3D IC technology, we propose a two step clock network synthesis algorithm called M3D-ZST. In the first step, we use the extended Greedy-DME algorithm [5] to generate a clock tree topology without a constraint on the number of MIVs while determining the merging segments at the same time. In the second step, we embed the whole clock tree in a top-down manner. The proposed algorithm provides shorter detour and total wirelength. Power is also reduced effectively.
- During the clock topology generation, we observe that in some cases, locating the merging point at the top or the bottom tier has no impact on the current branch wirelength, but may have a substantial impact on the total wirelength. Therefore, we extend the algorithm to look one step ahead deferring the decision of the optimal locations of merging segments and MIVs. With this look-ahead step the total wirelength can be further reduced.
- Experimental results indicate that the proposed M3D-ZST algorithm reduces the total wirelength by 9.7% – 19.7%, and

Figure 1: The structure of TSV-based 3D IC and M3D IC.

power by 9.4% – 18.6%, respectively, compared with the 3D-MMM algorithm [21] over IBM benchmarks. The M3D-ZSTLA algorithm further reduces the total wirelength by about 3%, and reduces the power by about 2% compared to M3D-ZST algorithm.

The remainder of this paper is organized as follows. Section 2 presents the background and related work. Section 3 formulates the M3D clock synthesis problem, the motivation of this work, and describes the M3D-ZST algorithm. Section 4 adds the look-ahead capability to the algorithm further improving the wirelength. Section 5 presents experimental results, and Section 6 concludes the paper.

Table 1: Specifications of TSV and MIV.

	TSV [10, 21]	MIV [14]
Diameter	5 μm	100 nm
Length	50 μm	500 nm
Pitch	10 μm	200 nm
Resistance	35 m Ω	2 Ω
Capacitance	100 fF	0.1 fF

2 BACKGROUND AND RELATED WORK

2.1 Introduction to Monolithic 3D IC Technology

The basic structure of monolithic 3D integration is shown in Fig. 1. As shown in the figure, differently from TSV-based 3D ICs, in M3D ICs, the tiers of devices are fabricated sequentially, one on the top of another, and the interconnects between tiers are called monolithic inter-layer vias (MIVs). Compared with TSV-based counterparts, M3D ICs have their unique advantages. As shown in Table 1, the size of MIV is much smaller than TSV, and can support ultra high integration density, which can reduce on-chip wirelength, and yield better performance and lower power. Moreover, MIV has much higher alignment accuracy than TSV, which improves the stacking yield significantly compared to TSV-based 3D ICs.

2.2 Brief Introduction to Clock Network Design of 3D ICs

The clock tree synthesis (CTS) can be divided into two phases, topology generation and embedding internal nodes of the topology. A clock tree topology is initially constructed under specific constraints, such as clock skew. After that, the detailed routing is

fixed aiming either minimal wirelength or power depending on the specific optimization objective. Different from 2D clock network design, TSV-based 3D clock network synthesis should consider the insertion of TSVs to reduce wirelength. Furthermore, TSV RC parasitics should also be considered during the clock routing. Finally, the number of TSVs is typically another important constraint. Zero-skew clock tree design is an important research topic both in 2D and 3D ICs, and the Elmore delay model [6] is commonly used to determine the clock skew, which is defined as the difference of arrival time of the clock signal from the clock source to different clock sinks. In our work, we focus on the zero-skew clock network design with minimal wirelength for M3D ICs.

2.3 Related Work

Many algorithms have been proposed for building 2D clock trees. Jackson *et al.* proposed a clock tree construction algorithm for cell-based layout named "method of means and medians (MMM)" [9]. Chao *et al.* first proposed the deferred-merge-embedding (DME) algorithm embedding any given topology to create a clock tree with zero-skew while minimizing the total wirelength [3]. Masato *et al.* proposed a clock tree synthesis algorithm named Greedy-DME, which achieved shorter wirelength than MMM by selecting several nearest-neighbor pairs to merge simultaneously [5]. Charikar *et al.* presented a bounded-skew routing tree generation algorithm with minimum wirelength while satisfying the skew bound [4].

As for the 3D IC clock design, Minz *et al.* first proposed a clock network synthesis algorithm for TSV-based 3D ICs [12]. Zhao *et al.* extended the MMM algorithm to 3D-MMM for TSV based 3D ICs [21]. 3D-MMM algorithm can generate a topology with various TSV bounds. The MMM algorithm is appropriate for adding the TSV count constraint to alleviate for the large TSV area and parasitics. Liu *et al.* presented a TSV-aware clock tree topology generation method while taking the TSV density as a constraint, TSV parasitics and coupling effects are also taken into account [11]. Kumar *et al.* presented a minimal buffer insertion algorithm for TSV-based 3D ICs [17]. Chandrakar *et al.* presented an obstacle aware clock network synthesis algorithm for TSV-based 3D ICs [2]. These algorithms are based on the 3D-MMM. Oh *et al.* presented a thermal-aware symmetrical buffered clock network synthesis algorithm for TSV-based 3D ICs [13]. There are few clock network synthesis algorithms specifically for the monolithic 3D ICs. Samadi *et al.* presented a clock distribution network for the monolithic 3D ICs, which slightly modified the clock tree construction algorithm of 2D ICs [16]. During the construction of the clock network, all the sinks are projected into one tier and then 2D clock tree construction algorithm is applied to all sinks. After the clock network is constructed, the sinks on different tiers are directly connected by MIVs, which means the delay of MIV is not been taken into account and the skew of the clock network cannot be balanced properly.

3 MONOLITHIC 3D CLOCK NETWORK SYNTHESIS

3.1 Motivation of the Technique

In TSV-based 3D ICs, multiple-TSV insertion can reduce the total wirelength of the clock network effectively compared to 2D circuits [21]. However, the large number of TSVs increases the power

and area overhead. Therefore, there is trade-off between TSV count and clock wirelength. The area and capacitance of MIV are both much smaller than the TSV. Therefore, the number of MIVs may not be a constraint for M3D clock network design.

Table 2: Wirelength (μm) comparison when directly applying 3D-MMM algorithm to M3D CTS.

crt	3D-MMM		
	TSV	MIV	WL increased (%)
r1	1578216	1580293	-0.132
r2	3359608	3364578	-0.148
r3	4466893	4469183	-0.051
r4	9529387	9529899	-0.005
r5	14276541	14276973	-0.003

The 3D-MMM has been widely used in clock tree synthesis. To evaluate the effectiveness of the existing 3D-MMM algorithm [21] for M3D ICs, we replaced the TSV with MIV, and compared the clock network wirelength for the IBM benchmark suite listed in Table 2. Note that the total wirelength slightly worsens or is comparable to the TSV-based 3D ICs. The reason is that the resistance of MIV ($2\ \Omega$) is much larger than TSV ($35\ m\Omega$). In other words, the state-of-the-art techniques cannot usefully exploit the much smaller size of MIVs and the related lower overhead in area. Another observation is that in 2D ICs, the Greedy-DME algorithm performs better than the MMM algorithm in terms of wirelength [5, 20]. However, for the TSV-based ICs, the Greedy-DME algorithm is not suitable because it cannot predict the TSV usage in advance and may violate the TSV count constraint. Alternatively, in M3D ICs, the MIV count is rather not a constraint. Therefore, a novel clock network synthesis algorithm is essential to fully utilize the gains of M3D integration.

3.2 Problem Formulation

The sinks of the clock network are the clock pins of synchronous elements, the location of which is determined in the placement phase. Let $S = \{s_1, s_2, \dots, s_n\}$ denote a set of sinks. In addition, let K denote a set of sinks or merging segments considered in the clock network construction. Initially, $K=S$. Each s_i is denoted by a three-dimensional coordinate (x_i, y_i, z_i) , where x_i and y_i denote the planar coordinates of the sink within a tier, and z_i is the tier index of s_i .

The M3D clock network construction problem can be described as follows: given a set of sinks in all tiers, a pre-determined source location s_0 , the capacitive load c_i of sink i , and the resistance and capacitance of wire per unit length and MIV (r, c, r_m, c_m), the objective is to construct a M3D clock network where 1) the clock sinks in all tiers are connected by one single tree; 2) the clock skew is zero under the Elmore delay model; 3) the total wirelength of clock network is minimized.

3.3 Overview of the Proposed M3D-ZST Algorithm

This section presents our algorithm to synthesize a M3D Zero Skew Tree (M3D-ZST) over a given set of sinks. It can be divided into two steps: 1) tree topology generation; 2) embedding of the topology.

Firstly, we generate a 3D tree topology based on the M3D-Greedy-DME algorithm without constraining the number of MIVs. The algorithm is an extension of the Greedy-DME algorithm in 2D clock network design [5]. The algorithm allocates the clock sinks into different clusters based on the nearest neighbor graph and then determines which pairs should be merged together and whether a MIV needs to be inserted or not. At the same time, the algorithm also calculates the minimum merging cost using the zero-skew equations of the Elmore delay model and determines the minimum detour wirelength if a detour is needed. The result of this algorithm is a 3D tree topology with merging segments, which represent possible locations of the internal merging nodes and MIVs. After the 3D tree topology is generated, we determine the optimal positions for all internal nodes and MIVs in a top-down manner.

3.4 M3D Clock Tree Topology Generation

The first step of our 3D clock network construction is the 3D tree topology generation. The basic idea of the M3D-Greedy-DME is to recursively merge several nearest pairs of sinks (or merging segments) each time in a bottom-up manner.

Let K be a set of sinks or merging segments. Initially, $K = S$. Firstly, M3D-Greedy-DME algorithm takes the clock sink set S as input. Then, the algorithm constructs a nearest-neighbor graph (NNG) for K . The NNG indicates the nearest-neighbor sink (or segment) to each sink (or segment) in K . $|K|/k$ nearest-neighbor pairs are selected from the NNG in a non-decreasing order of merging cost, where k is a constant generally ranging between $2 \leq k \leq 4$ [5]. For each pair a and b , if neither a nor b has already been merged, the nodes are merged based on the minimum cost principle, and the merging segment for parent node p from a and b is calculated with the Elmore delay model under the zero-skew constraint. An MIV is used, if a and b are located in different tiers. Then, delete a and b from K and add merging segment of p to K . The zero-skew merge calculates the minimum merging cost and determines the minimum detour wirelength if a detour is needed. Then, reconstruct the NNG for K until $|K| = 1$. Let a and b be the children of p in the tree topology. $d(a, b)$ denotes the Manhattan distance between a and b . $t_{ED}(p, a)$ denotes the Elmore delay from p to a , and $t_{ED}(p)$ denotes the Elmore delay from p downstream to the sinks of the subtree rooted at p . Directed edges from p to a and b are denoted as e_a, e_b , respectively. Their wirelengths are $|e_a|$ and $|e_b|$, which must be greater than or equal to the Manhattan distance $d(p, a)$ and $d(p, b)$, respectively. Let α and β denote the resistance and capacitance per unit length of the interconnect, and the resistance r_a and capacitance c_a of edge e_a are $\alpha \cdot |e_a|$ and $\beta \cdot |e_a|$, respectively. The resistance and capacitance of MIV is denoted by r_m and c_m . The capacitance of the subtree rooted at p is denoted by C_p . According to the definition of Elmore delay, we have that $t_{ED}(p, a) = r_a((1/2)c_a + C_a)$.

In the zero-skew merge, we calculate the merging segment for two sinks (or segments) that are to be merged with minimum merging cost. For example, we calculate the merging segment $MS(p)$ needed to merge a and b . The subtrees of a and b , respectively, have capacitance C_a and C_b and delay $t_a = t_{ED}(a)$ and $t_b = t_{ED}(b)$. There are a few possible scenarios to be considered when we calculate the minimum merging cost, as follows.

Scenario 1: the merge pair a and b is located in the same tier.

Figure 2: The scenario of a and b in the same tier.

For the case shown in Fig. 2, the merging segment of p satisfies

$$r_a(\frac{1}{2}c_a + C_a) + t_a = r_b(\frac{1}{2}c_b + C_b) + t_b. \quad (1)$$

Let $d(a, b) = d$, suppose a and b can be merged with the merging cost d , which means $|e_a| + |e_b| = d$. We also assume $|e_a| = x$ and $|e_b| = d - x$. Then (1) becomes,

$$\alpha x(\frac{1}{2}\beta x + C_a) + t_a = \alpha(d - x)(\frac{1}{2}\beta(d - x) + C_b) + t_b. \quad (2)$$

Case 1: If $0 \leq x \leq d$, a and b can be merged with merging cost d , and $|e_a| = x$ and $|e_b| = d - x$.

Case 2: If $x < 0$, the merging operation with merging cost d leads to a negative $|e_a|$. In this case, $t_a > t_b$, the merging point of p is the location of a . In this case, detour is needed to balance the delays between two trees, we set $|e_a| = 0$ and $|e_b| = d'$. The detour wirelength is $(d' - d)$. Then (1) becomes,

$$t_a = \alpha d'(\frac{1}{2}\beta d' + C_b) + t_b. \quad (3)$$

Case 3: If $x > d$, the merging operation with merging cost d leads to a negative $|e_b|$. In this case, $t_a < t_b$. The merging point of p is the location of b . Detour is also needed to balance the delays between two trees, we set $|e_a| = d'$ and $|e_b| = 0$. The detour wirelength is $(d' - d)$, and (1) is written as,

$$t_b = \alpha d'(\frac{1}{2}\beta d' + C_a) + t_a. \quad (4)$$

Scenario 2: the merge pair a and b is not located in the same tier.

Figure 3: Scenarios of a and b in different tiers. (a) the merging segment of p is at the bottom terminal of MIV, and (b) the merging segment of p is at the top terminal of MIV.

In this situation, an MIV is needed, there are two options for the merging segment of p as shown in Fig. 3. Then, a question raised is which option to choose to yield the minimum merging cost. We calculate the merging cost for the cases in Fig. 3 (a) and Fig. 3 (b), respectively. Then, we choose the shorter one. For the case shown in Fig. 3 (a), the merging segment of p is at the bottom terminal of MIV and satisfies

$$r_a(\frac{1}{2}c_a + C_a) + t_a = r_b(\frac{1}{2}c_b + C_b) + r_m(\frac{1}{2}c_m + c_b + C_b) + t_b. \quad (5)$$

Similar to *Scenario 1*, let $d(a, b) = d$, assume $|e_a| = x$ and $|e_b| = d - x$. Then (5) becomes,

$$\alpha x(\frac{1}{2}\beta x + C_a) + t_a = \alpha(d - x)(\frac{1}{2}\beta(d - x) + C_b) + r_m(\frac{1}{2}c_m + \beta(d - x) + C_b) + t_b. \quad (6)$$

Case 1: If $0 \leq x \leq d$, which means a and b can be merged with merging cost d , and $|e_a| = x$ and $|e_b| = d - x$.

Case 2: If $x < 0$, detour is needed to balance the delays between two sub-trees. We set $|e_a| = 0$ and $|e_b| = d'$. The detour wirelength is $(d' - d)$.

There are two situations for detouring, and we calculate both situations and choose the shorter one. If the detour wirelength is between the top of MIV and b , (5) becomes,

$$t_a = \alpha d'(\frac{1}{2}\beta d' + C_b) + r_m(\frac{1}{2}c_m + \beta d' + C_b) + t_b. \quad (7)$$

If the detour wirelength is at the bottom of MIV, (5) becomes,

$$t_a = \alpha d(\frac{1}{2}\beta d + C_b) + r_m(\frac{1}{2}c_m + \beta d + C_b) + \alpha(d' - d)(\frac{1}{2}\beta(d' - d) + C_b + c_m + \beta d) + t_b. \quad (8)$$

Case 3: If $x > d$, detour is also needed to balance the delays between two merged subtrees. We set $|e_a| = d'$ and $|e_b| = 0$. The detour wirelength is $(d' - d)$. The detour wirelength is between a and the bottom of MIV, then (5) is written as,

$$\alpha d'(\frac{1}{2}\beta d' + C_a) + t_a = r_m(\frac{1}{2}c_m + C_b) + t_b. \quad (9)$$

For the case shown in Fig. 3 (b), the merging segment of p is at the top terminal of MIV, and a similar process can be followed to determine the merging segment.

3.5 The Embedding phase of M3D-ZST algorithm

The second step of our algorithm is embedding the 3D tree topology, i.e., given a 3D tree topology, determine the exact locations of all internal nodes and MIVs in a top-down manner as the merging segments are determined in the 3D tree topology generation. Assume node v is embedded, if v is the root node, we can select any point along the merging segment of v to be the root location. If v is an internal node, we select the location to be a point along the merging segment of v satisfying the zero-skew merge equation from v to its parent.

4 ZERO-SKEW CLOCK TREE SYNTHESIS WITH LOOK-AHEAD CAPABILITY

As shown in Fig. 4, the merging segment of p has two choices, i.e., at the top or the bottom terminals of the MIV. In M3D-ZST algorithm, we calculate the merging cost for both choices, and choose the shorter one. Often, the merging costs of these two choices are the same. For these situations, M3D-ZST chooses randomly the merging segment at either the top or bottom terminal of the MIV.

However, in these situations, choosing a merging segment at the top or bottom of MIV affects the total wirelength. As shown in Fig. 4, choosing a merging segment at the top or bottom of the MIV has no impact on the current wirelength, but this choice leads to the

Figure 4: Examples of M3D-ZSTLA algorithm. (a) the merge pair p and c are located in the same tier, and (b) the merging pair of p and c are not located in the same tier.

next merge pair p and c in the same (Fig. 4 (a)) or different (Fig. 4 (b)) tier, affecting the merging cost of this pair and the total wirelength thereafter. Thus, we extend M3D-ZST algorithm to look-ahead one step to determine the optimal location of the merging segment. The goal is to predict the effect of choosing a merging segment on the total wirelength. Then, the appropriate one is chosen to further reduce the total wirelength.

The key idea of M3D-ZSTLA algorithm is that each time the merging segment for two sinks (or segments) is calculated, if choosing a merging segment at the top or bottom terminal of the MIV exhibits the same merging cost, we do not determine the location of the merging segment in the current step, but instead, we determine the location in the next merging step which contains this merging segment. For example, in the cases shown in Fig. 4, when merging a and b , if choosing the merging segment of p at either the top or bottom terminal of the MIV leads to the same merging cost, we defer the allocation of the merging segment of p until the next merging of p and c . Consequently, the M3D-ZSTLA algorithm, respectively, calculates the merging cost for situations similar to Fig. 4 (a) and Fig. 4 (b). The case with the shorter merging cost determines the location of the merging segment of p .

The time complexities of both the M3D-ZST and the M3D-ZSTLA are $O(n \cdot \log n)$, where n is the number of sinks.

5 EXPERIMENTAL RESULTS

5.1 Experiment setup

We implement zero-skew 3D clock networks for M3D ICs with the M3D-ZST and M3D-ZSTLA algorithms detailed in the previous sections with C++ and STL library, and we run the algorithms on a server with a 3.0 GHz quad-core Xeon processor with 32 GB memory. The technology parameters of interconnects are based on the 45 nm Predictive Technology Model [8]: per unit-length wire resistance is $0.1 \Omega/\mu\text{m}$, per unit-length wire capacitance is $0.2 \text{ fF}/\mu\text{m}$. The assumed TSV resistance and capacitance is $35 \text{ m}\Omega$ and 100 fF , respectively. The MIV resistance and capacitance is 2Ω and 0.1 fF , respectively. The supply voltage is set to 1.2 V. The clock frequency is 1 GHz.

5.2 Experimental results

We present experimental results based on benchmark circuits IBM r1 to r5 [18]. Since r1 to r5 are clock network benchmarks for 2D ICs, we randomly distribute the sinks on two tiers.

Table 3 and Table 4 compare the MIV count (#MIVs), wirelength (WL) and the clock power between the proposed algorithms and existing 3D-MMM algorithm [21] for the different MIV bounds.

Table 3: Comparison of Wirelength (μm), Power (W), MIV count between the 3D-MMM and the M3D-ZST algorithm for a two tier stack. The MIV bound in 3D-MMM is set to 100.

crt	3D-MMM			M3D-ZST			WL dec(%)	Power dec(%)
	#MIVs	WL	Power	#MIVs	WL	Power		
r1	75	1580293	0.476	134	1426238	0.431	9.7	9.4
r2	97	3364578	1.016	314	2936523	0.892	12.7	12.2
r3	100	4469183	1.356	392	3777314	1.156	15.5	14.7
r4	100	9529899	2.896	945	7980939	2.450	16.2	15.4
r5	100	14276541	4.357	1510	11461347	3.546	19.7	18.6

Table 4: Comparison of Wirelength (μm), Power (W), MIV count between the 3D-MMM and the M3D-ZST algorithm for a two tier stack. The MIV bound in 3D-MMM is set to 1,510.

crt	3D-MMM			M3D-ZST			WL dec(%)	Power dec(%)
	#MIVs	WL	Power	#MIVs	WL	Power		
r1	101	1500249	0.453	134	1426238	0.431	4.9	4.8
r2	213	3005634	0.913	314	2936523	0.892	2.3	2.3
r3	294	3897632	1.191	392	3777314	1.156	3.1	2.9
r4	650	8015327	2.459	945	7980939	2.450	0.4	0.4
r5	959	11751086	3.630	1510	11461347	3.546	2.5	2.3

We use two tier stacks. We also report the wirelength and power reduction of the proposed algorithms with respect to the 3D-MMM algorithm. In Table 3, the MIV bound in 3D-MMM algorithm is set to 100. Comparing with the 3D-MMM algorithm, M3D-ZST algorithm decreases, respectively, the wirelength from 9.7% to 19.7%, and the power from 9.4% to 18.6%. From r1 to r5, the decrease in wirelength increases because the number of MIVs in M3D-ZST increases rapidly but the maximum number of MIVs in 3D-MMM is set as 100. A large MIV number provides more vertical interconnects, which reduces the total wirelength and power.

In Table 4, the MIV bound in 3D-MMM algorithm is set to 1,510, which is the same as the maximum number of MIV in M3D-ZST. In this case, the M3D-ZST algorithm reduces, respectively, the wirelength from 0.4% to 4.9%, and the power from 0.4% to 4.8%. Although the reductions are much smaller than the results for MIV bound equal to 100, our proposed algorithm still performs better than 3D-MMM. The total wirelength and power are reduced as more MIVs are used.

Table 5: Comparison of Wirelength (μm), Power (W), MIV count between the M3D-ZST and the Look-ahead M3D-ZSTLA algorithm for a two tier stack.

crt	M3D-ZST			M3D-ZSTLA			WL dec(%)	Power dec(%)
	#MIVs	WL	Power	#MIVs	WL	Power		
r1	134	1426238	0.4315	141	1425360	0.4312	0.06	0.07
r2	314	2936523	0.8927	306	2935067	0.8923	0.05	0.04
r3	392	3777314	1.1564	392	3762195	1.1521	0.40	0.37
r4	945	7980939	2.4498	978	7794581	2.3961	2.34	2.19
r5	1510	11461347	3.5466	1553	11416403	3.5336	0.39	0.37

Table 5 compares the MIV count (#MIVs), wirelength (WL) and the clock power between the M3D-ZST algorithm and M3D-ZSTLA

Figure 5: 3D clock networks produced by 3D-MMM, M3D-ZST and M3D-ZSTLA algorithms for the benchmark r4 implemented with two tiers.

algorithm. We also use two tier stacks. Comparing with the M3D-ZST algorithm, M3D-ZSTLA algorithm decreases, respectively, the wirelength from 0.05% to 2.34%, and the power from 0.04% to 2.19%.

Fig. 5 illustrates the 3D clock networks for benchmark r4 produced by 3D-MMM, M3D-ZST, and M3D-ZSTLA algorithms, respectively. The clock network on the bottom tier is depicted in red color, and the clock network on the top tier is depicted in blue color. The pink dots denote the inserted MIVs. As shown in this figure, the wirelength of M3D-ZST is reduced compared to 3D-MMM, and the wirelength of M3D-ZSTLA is further reduced compared to M3D-ZST.

6 CONCLUSION

A clock network synthesis algorithm has been developed for monolithic 3D ICs, which constructs a clock tree with zero skew while minimizing the total wirelength. We demonstrate the difference between MIV and TSV in the clock tree synthesis. We find that network construction for TSV based 3D ICs is not suitable for M3D ICs due to the area and parasitics of MIV that are both much smaller than the TSV. We developed the M3D-ZST algorithm to synthesize clock networks for M3D ICs without constraining the number of MIV. Furthermore, we extend the M3D-ZST algorithm to M3D-ZSTLA algorithm, which determines the optimal location of the merging segment by looking ahead one step to further reduce the total wirelength. Experimental results show that the proposed M3D-ZST algorithm reduces the total wirelength from 9.7% to 19.7%, and decreases power from 9.4% to 18.6% compared with the 3D-MMM algorithm. Additionally, the M3D-ZSTLA algorithm further reduces the total wirelength from 0.05% to 2.34%, and reduces power from 0.04% to 2.19% compared to the M3D-ZST algorithm.

REFERENCES

- [1] Perrine Batude, Thomas Ernst, Julien Arcamone, Gregory Arndt, Perceval Coudrain, and Pierre-Emmanuel Gaillardon. 2012. 3-D sequential integration: A key enabling technology for heterogeneous co-integration of new function with CMOS. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2, 4 (2012), 714–722.
- [2] Leena Chandrakar, S Ravi, and Harish M Kittur. 2017. Power and Obstacle Aware 3D Clock Tree Synthesis. (2017).
- [3] Ting-Hai Chao, Yu-Chin Hsu, Jan-Ming Ho, and AB Kahng. 1992. Zero skew clock routing with minimum wirelength. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 39, 11 (1992), 799–814.
- [4] Moses Charikar, Jon Kleinberg, Ravi Kumar, Sridhar Rajagopalan, Amit Sahai, and Andrew Tomkins. 2004. Minimizing wirelength in zero and bounded skew clock trees. *SIAM Journal on Discrete Mathematics* 17, 4 (2004), 582–595.
- [5] Masato Eda. 1993. A clustering-based optimization algorithm in zero-skew routings. In *30th ACM/IEEE Design Automation Conference*. IEEE, 612–616.
- [6] William C Elmore. 1948. The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of applied physics* 19, 1 (1948), 55–63.
- [7] Nor Zaidi Haron and Said Hamdioui. 2008. Why is CMOS scaling coming to an END?. In *2008 3rd International Design and Test Workshop*. IEEE, 98–103.
- [8] Nanoscale Integration and Modeling (NIMO) Group. [n.d.]. *Predictive technology model*. <http://ptm.asu.edu/>
- [9] Michael AB Jackson, Arvind Srinivasan, and Ernest S Kuh. 1990. Clock routing for high-performance ICs. In *27th ACM/IEEE Design Automation Conference*. IEEE, 573–579.
- [10] Chang Liu and Sung Kyu Lim. 2012. A design tradeoff study with monolithic 3D integration. In *Thirteenth International Symposium on Quality Electronic Design (ISQED)*. IEEE, 529–536.
- [11] Wulong Liu, Haixiao Du, Yu Wang, Yuchun Ma, Yuan Xie, Jinguo Quan, and Huazhong Yang. 2013. TSV-aware topology generation for 3D clock tree synthesis. In *International Symposium on Quality Electronic Design (ISQED)*. IEEE, 300–307.
- [12] Jacob Minz, Xin Zhao, and Sung Kyu Lim. 2008. Buffered clock tree synthesis for 3D ICs under thermal variations. In *2008 Asia and South Pacific Design Automation Conference*. IEEE, 504–509.
- [13] Deok Keun Oh, Mu Jun Choi, and Ju Ho Kim. 2018. Thermal-Aware 3D Symmetrical Buffered Clock Tree Synthesis. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 9–16.
- [14] Shreepad Panth, Kambiz Samadi, Yang Du, and Sung Kyu Lim. 2014. Power-performance study of block-level monolithic 3D-ICs considering inter-tier performance variations. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.
- [15] Vasilis F Pavlidis, Ioannis Savidis, and Eby G Friedman. 2017. *Three-dimensional integrated circuit design*. Newnes.
- [16] Kambiz Samadi, Shreepad A Panth, Jing Xie, and Yang Du. 2015. Clock distribution network for 3D integrated circuit. US Patent 9,098,666.
- [17] Kamineni Sumanth Kumar and John Reuben. 2016. Minimal Buffer Insertion Based Low Power Clock Tree Synthesis for 3D Integrated Circuits. *Journal of Circuits, Systems and Computers* 25, 11 (2016), 1650142.
- [18] R.S. Tsay. [n.d.]. *GSRC Benchmark*. <https://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST/>
- [19] S Simon Wong and Abbas El Gamal. 2009. The prospect of 3D-IC. In *2009 IEEE Custom Integrated Circuits Conference*. IEEE, 445–448.
- [20] Alexander Z Zelikovsky and Ion I Mandoiu. 2001. Practical approximation algorithms for zero- and bounded-skew trees. *SIAM Journal on Discrete Mathematics* 15, 1 (2001), 97–111.
- [21] Xin Zhao, Jacob Minz, and Sung Kyu Lim. 2010. Low-power and reliable clock network design for through-silicon via (TSV) based 3D ICs. *IEEE Transactions on Components, Packaging and Manufacturing Technology* 1, 2 (2010), 247–259.