Chiplet Placement for 2.5D IC with Sequence Pair Based Tree and Thermal Consideration

Hong-Wen Chiou, Jia-Hao Jiang, Yu-Teng Chang, Yu-Min Lee, Chi-Wen Pan
{simon.chiou.ee05,justin1023.ee10,lugu7000013.ee09,yumin}@nycu.edu.tw,wayne1234.cm97g@nctu.edu.tw
National Yang Ming Chiao Tung University, Hsinchu, Taiwan

ABSTRACT

This work develops an efficient chiplet placer with thermal consideration for 2.5D ICs. Combining the sequence-pair based tree, branch-and-bound method, and advanced placement/pruning techniques, the developed placer can find the solution fast with the optimized total wirelength (TWL) on half-perimeter wirelength (HPWL). Additionally, with the post placement procedure, the placer reduces maximum temperatures with slight increase of wirelength. Experimental results show that the placer can not only find better optimized TWL (reducing 1.035% HPWL) but also speed up at most two orders of magnitude than the prior art. With thermal consideration, the placer can reduce the maximum temperature up to 8.214 ℃ with an average 5.376% increase of TWL.

KEYWORDS

2.5D IC, chiplet placement, sequence pair, thermal

ACM Reference Format:
Hong-Wen Chiou, Jia-Hao Jiang, Yu-Teng Chang, Yu-Min Lee, Chi-Wen Pan. 2023. Chiplet Placement for 2.5D IC with Sequence Pair Based Tree and Thermal Consideration. In *28th Asia and South Pacific Design Automation Conference (ASPDAC '23), January 16–19, 2023, Tokyo, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3566097.3567911

1 INTRODUCTION

As technology nodes scale down, chips can achieve higher performance. As the transistor size has approached to its physical limit, recently, 2.5D ICs (also called interposer-based 3D ICs) have been developed by the IC foundry [1] to alleviate the issues of technology node scaling. 2.5D ICs can provide better performance and yield than 2D ICs [2] and have been realized as commercial products [3, 4]. Fig. 1 illustrates the structure of a 2.5D IC [5]. It contains a silicon interposer as an interface between chiplets and the package substrate. The nets between chiplets are routed in the redistribution layer (RDL) of the silicon interposer, and μ-bumps and C4 bumps are for the connections between chiplets, interposer, and package substrate.

Several approaches have been proposed to minimize wirelength of interconnects between chiplets. Ho and Chang [6] utilized the simulated annealing (SA) method to place multiple chiplets, macros, and I/O buffers. Liu *et al.* [7] introduced an enumeration-based algorithm with acceleration techniques for solving the multi-die floorplanning problems, which can obtain the optimal placement with minimizing total wirelength (TWL) on HPWL. Recently, Osmolovskyi *et al.* [8] modeled the chiplet placement as the constraint-satisfaction problem (CSP) and applied advanced pruning techniques to an advanced branch-and-bound (B&B) method for saving runtimes and obtaining

ASPDAC '23, January 16–19, 2023, Tokyo, Japan

ACM ISBN 978-1-4503-9783-4/23/01…$15.00
https://doi.org/10.1145/3566097.3567911

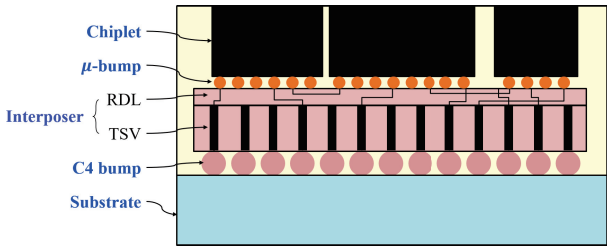

Figure 1: The architecture of 2.5D IC.

the optimal placement while minimizing TWL. For thermal consideration on chiplet placement, [9, 10] presented a thermal-aware chiplet placement by using the SA technique with thermal-aware cost function and inserting spaces between chiplets while finding a routing solution simultaneously. However, the SA based chiplet placement methods might obtain a sub-optimal solution.

To our best knowledge, there is no chiplet placer considering thermal effects while seeking the optimal placement. This work develops an efficient chiplet placer via the sequence pair based tree and we name it SP-CP that can obtain the optimal TWL. Moreover, to alleviate thermal issues, we build a post placement procedure to find a better thermal solution from the optimal and near-optimal placements of SP-CP. The main contributions are

- We develop an efficient chiplet placer, SP-CP. We build a sequence pair (SP) based tree (SP-Tree), which contains both rotation and partial/complete SP representation, and apply the B&B method on SP-Tree to find the optimal solution. Moreover, we advance placement/pruning techniques for the optimal solution and faster runtime in SP-CP.
- For thermal consideration, we build a post placement procedure to reduce the operating temperatures of 2.5D ICs on the optimal and near-optimal placements of SP-CP. It seeks for placement solution considering thermal effects but slightly increasing TWL.

We organize this paper as follows. First, Section 2 introduces the floorplanning and chiplet placement techniques of the prior art and Section 3 states the problem formulation. Next, Section 4 overviews the proposed framework for chiplet placement with SP-Tree and thermal consideration. Section 5 details SP-CP containing SP-Tree and advanced placement/pruning techniques for optimizing TWL. Then, Section 6 presents the post placement procedure of SP-CP to reduce the maximum temperature. Finally, Section 7 shows experimental results and Section 8 concludes this work.

2 PRELIMINARY

2.1 The B&B method for chiplet placement [8]

2.1.1 Chiplet ordering. [8] considers the interconnect of chiplets and constructs a complete graph with weights $e_{ij} = n_{ij} \cdot (w_i + h_i + w_j + h_j)/2$. w_i/w_j and h_i/h_j are the width and the height of chiplet i/j (c_i/c_j), respectively. n_{ij} is the number of nets connected by c_i and c_j. Then, it determines a node that has the maximum cut from sorted nodes iteratively.

2.1.2 Placement optimization with whitespace. The default packed chiplets in [8] are set in the left-bottom corner of interposer. As a chiplet has terminal nets, this packing may not be optimal. Hence, first, it treats all chiplets as a "virtual chiplet" and moves this virtual chiplet to minimize terminal connections. Then, it moves one chiplet for minimizing HPWL. It iteratively performs these two steps until there is no further improvement of TWL.

2.1.3 Terminal handling (TH). When a chiplet is added to a partial placement, [8] looks forward to predicting WL augmentation from its terminal nets. For a chiplet, [8] calculates the optimized HPWL by using the method in Section 2.1.2 with its terminal nets. Since a chiplet can have four rotations, each chiplet has four values in TH.

2.1.4 Forward wirelength checking (FC). [8] calculates the minimum WL for each chiplet pair with the nets only connected between them by using Section 2.1.2 in 4^3 possible cases, which includes rotations (4×4, rotations of c_i and c_j) and topologies (4 topologies between c_i and c_j).

2.2 Sequence pair (SP) representation [11]

The sequence pair uses two sequences to express the topological relations between blocks (chiplets) as

$$(\cdots i \cdots j \cdots, \cdots i \cdots j \cdots) \Rightarrow c_i \text{ is on the left of } c_j, \tag{1a}$$

$$(\cdots i \cdots j \cdots, \cdots j \cdots i \cdots) \Rightarrow c_j \text{ is below } c_i. \tag{1b}$$

The placement of SP can be packed with horizontal/vertical constraint graph and its time complexity is $O(n^2)$, where n is the number of chiplets.

3 PROBLEM FORMULATIONS

The problem for chiplet placement with thermal consideration is to find a suitable placement while reducing the maximum temperature of chiplets. Firstly, we optimize the chiplet placement while minimizing TWL and it is formulated as

$$\min \sum_{k}^{\#nets} HPWL_k \tag{2}$$

subject to

$$0 \le x_{l,i},\ x_{r,i} \le W;\ 0 \le y_{b,i},\ y_{t,i} \le H. \tag{3a}$$

$$w_s \le \min\{|x_{l,i} - x_{r,j}|, |x_{l,j} - x_{r,i}|, |y_{b,i} - y_{t,j}|, |y_{b,j} - y_{t,i}|\}, \tag{3b}$$

where W/H is the width/height of the interposer, $x_{l,i}/x_{r,i}$ is the left/right position of c_i along the $x-$axis, and $y_{b,i}/y_{t,i}$ is the bottom/top position of c_i along the $y-$axis.

Here, we utilize the metric of HPWL to estimate the wirelength of a net k and our goal is to minimize TWL. The placement constraints include the fixed-outline constraint in (3a), and the minimum distance between chiplets [5] in (3b).

Then, with the optimal and near-optimal placements by solving (2), the problem is to reduce the maximum temperature from these placements. And we find a near-optimal placement satisfying thermal constraint T_{cstr}, with an acceptable TWL increase in percentage $\eta\%$ compared to the optimized TWL (TWL^{opt}). If the maximum temperature T_{max} of chiplets is still larger than T_{cstr}, we choose the placement with the minimal cost from the placements with increased TWL $< \eta\%$ and the cost function is formulated as

$$cost = \phi \times \frac{\text{TWL} - \text{TWL}^{opt}}{\text{TWL}^{max} - \text{TWL}^{opt}} + (1-\phi) \times \frac{T_{max} - T_{max}^{min}}{T_{max}^{max} - T_{max}^{min}}, \tag{4}$$

where TWL^{max} is the maximum TWL, and $T_{max}^{max}/T_{max}^{min}$ is the maximum/minimum of maximum temperatures of the placements with increased TWL $< \eta\%$.

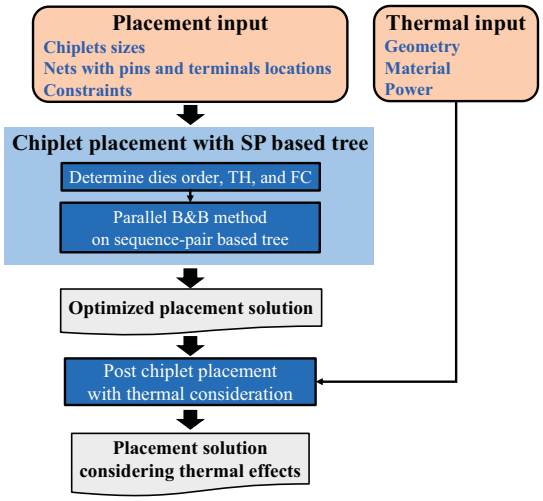

Figure 2: The proposed framework.

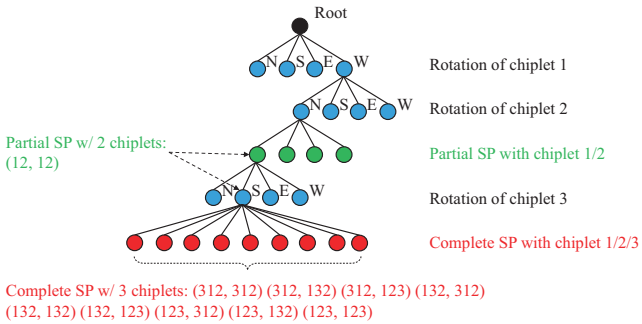

Figure 3: Sequence pair based tree.

4 THE PROPOSED FRAMEWORK

Fig. 2 shows the framework for minimizing TWL in (2) and reducing the maximum temperature under placement constraints. Given sizes of chiplets, nets with pins, net weights, terminal locations, placement constraints, geometries of 2.5D IC components, material, and power consumption of each chiplet, firstly, we perform the placer preprocessing for die ordering and estimated HPWL (*TH* and *FC*) as Sections 2.1.1, 2.1.3, and 2.1.4. Then, we apply the parallel B&B method on sequence-pair based tree for obtaining a set of optimal and sub-optimal placements with minimizing TWL. Finally, we perform post placement by considering the thermal effect to reduce T_{max} and find a near-optimal placement with lower T_{max}.

5 CHIPLET PLACEMENT WITH SP BASED TREE

5.1 SP based tree (SP-Tree)

SP-CP builds SP-Tree to represent all possible solutions. SP-Tree consists of three kinds of nodes: (i) rotation node with the direction of chiplet, north (0°), west (90°), south (180°) and east (270°) (blue nodes in Fig. 3), (ii) partial SP for topological relation between chiplets (green nodes in Fig. 3), and (iii) complete SP (red leaf nodes in Fig. 3). We add one chiplet while traversing to a rotation node. And a new index of chiplet is inserted into the SP with partial SP of the parent node. For example, when we insert chiplet 3 into the parent node with SP (12, 12), we can get nine child nodes (312, 312), (312,132), (312,123), (132, 312), (132,132), (132,123), (123, 312), (123,132), and (123,123).

CSP-Tree (a tree with a constraint-satisfaction problem) in [8] represents the rotation of chiplets and topology (left, right, bottom, and top) of each chiplet pair. Compared with CSP-Tree, all the nodes of

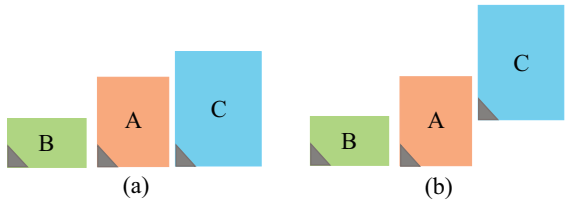

Figure 4: An example of two topology nodes in CSP-Tree having the same SP representation (BAC, BAC).

partial SP and complete SP in SP-Tree can be packed into a placement, but some of topology nodes in CSP-Tree might not be packed into a placement. For example, if chiplet B (c_B) is at the left of chiplet A (c_A), chiplet C (c_C) is at the right of c_A, and c_C is at the left of c_B. It forms a cyclic horizontal constraint graph. Thus, it cannot be packed into a placement.

In addition, one SP representation might cover multiple topology nodes in CSP-Tree. For example, Fig. 4.(a) shows one node in CSP-Tree (c_B is at the left of c_A, c_C is at the right of c_A, and c_C is at the right of c_B). And Fig. 4.(b) shows the other node in CSP-Tree (c_B is at the left of c_A, c_C is at the right of c_A, c_C is above of c_B). Both nodes in Fig. 4 can be represented as a SP (BAC, BAC).

The solution spaces of CSP-Tree and SP-Tree for n chiplets are $4^n \times 4^{n(n-1)/2}$ and $4^n \times (n!)^2$, respectively, and $\forall n > 2, 4^{n(n-1)/2} > (n!)^2$. For instance, as the chiplet number n is 11, the solution spaces of CSP-Tree are about 5.44×10^{39} which is much larger than the solution spaces of SP-Tree which are 6.68×10^{21}.

5.2 Advanced placement/pruning techniques

5.2.1 Analytical whitespace optimization. Besides the two steps presented in Section 2.1.2 [8] for optimizing TWL within whitespace, SP-CP introduces two extra moves to optimize TWL within whitespace. One move is that SP-CP fixes one chiplet and moves other chiplets as a "virtual chiplet" while minimizing TWL with terminal nets. Then, it iteratively fixes each chiplet and optimizes the "virtual chiplet" until no further improvement on TWL. The other move is that SP-CP fixes two chiplets and moves other chiplets as a "virtual chiplet" while minimizing TWL with terminal nets. Then, it fixes two chiplets among all chiplets iteratively and optimizes the "virtual chiplet" until no further improvement on TWL.

5.2.2 Remaining distance for SP-Tree. [8] uses "remaining distances (RD)" to estimate the HPWL of chiplets that are unplaced in the partial placement. In SP-Tree, this methodology can be applied as well. SP-CP utilizes the TH and the FC from Section 2.1.3 and 2.1.4 to calculate the RD for a partial SP node. If there are k placed chiplets $(c_k, \cdots, c_1)$ at a partial SP node, the RD of this node can be calculated by $(n-k)$ rotations for the TH and $(n \times (n-1)/2 - k \times (k-1)/2)$ topologies for the FC. Moreover, if there are k placed chiplets $(c_k, \cdots, c_1)$ at a rotation node, the RD of this node can be calculated by $(n-k-1)$ rotations for the TH and $(n \times (n-1)/2 - (k-1) \times (k-2)/2)$ topologies for the FC. SP-CP chooses the minimum value of TH (with four values) and the minimum value of FC (with sixty-four values) while calculating the RD at a rotation node and a partial SP node in SP-Tree, respectively. Note that sixty-four values are needed for calculating the minimum FC in a topology for the RD. The number of values for calculating the minimum FC can be reduced to sixteen values (if the rotation of one chiplet in FC is known) or four values (if the rotation of both chiplets in FC are known).

At the partial SP node, SP-CP defines the optimized partial TWL of a partial SP as $\text{TWL}_{\text{par},c_k}$ which can be obtained by using the method in Section 5.2.1. Fig. 5 illustrates one example for estimating the HPWL lower bound. There are two cases, the rotation node and the partial SP

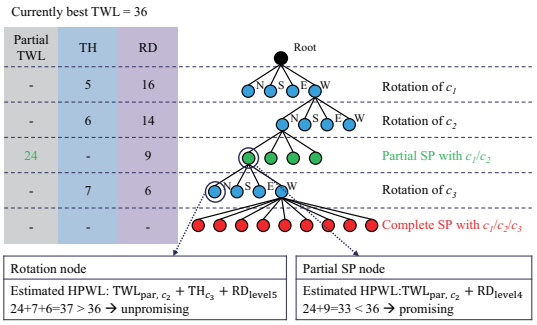

Figure 5: An example for estimating HPWL.

node. For the rotation node, the estimated HPWL is the summation of $\text{TWL}_{\text{par},c_2}$ (from the parent node), TH of the current node, and RD (the minimum value of FC for c_2 and c_3). For the partial SP node, the estimated HPWL is the summation of $\text{TWL}_{\text{par},c_2}$ and RD (the minimum value of TH for c_3 and the minimum value of FC for c_2 and c_3). If the estimated HPWL is larger than the current best TWL, this node is pruned in the B&B method.

5.2.3 Pruning dominated nodes of rotation and partial SP. [8] proposed the pruning dominated configuration to address the non-linear nature of HPWL metric for partial configurations on CSP-Tree. For B&B on SP-Tree, we might similarly discard a rotation/topology node when its estimated TWL is larger than the estimated minimum TWL, $bestTWL$, with the amount of $bestTWL/(n_k+1)$, and we call this procedure the PDSP. Here, n_k is the number of placed chiplets at the level.

However, a topology node in CSP-Tree represents one topology while a partial SP node in SP-Tree represents multiple topology relationships. Hence, we calculate a modification coefficient $\gamma[i]$ by Algorithm 1 (A1) for each node i to reduce the amount to be $\gamma[i] \times bestTWL/(n_k+1)$ for pruning more topology nodes. We call this procedure the PDSP & A1. First, we define $\alpha[i]$ to be the number of topology differences between the partial SP at node i and the partial SP having the best estimated HPWL at the same level. For example, after inserting c_3 into partial SP (12, 12), we analyze the contribution to $\alpha[i]$. Here, we assume (312, 312) have the minimum estimated HPWL at the same level. For (312, 312) and (312, 132) at node 1 at the same level, we compare the topology of c_3 with c_1, and the topology of c_3 with c_2.

1) $(\cdots 3 \cdots 1 \cdots, \cdots 3 \cdots 1 \cdots)(\cdots 3 \cdots 1 \cdots, \cdots 1 \cdots 3 \cdots)$ the topology of c_3 and c_1 is different. ⇒ this case contributes 1 to $\alpha[1]$.
2) $(\cdots 3 \cdots 2 \cdots, \cdots 3 \cdots 2 \cdots)(\cdots 3 \cdots 2 \cdots, \cdots 3 \cdots 2 \cdots)$ the topology of c_3 and c_2 is the same. ⇒ this case contributes 0 to $\alpha[1]$.

Here, $\alpha[1] = 1 + 0 = 1$.

Algorithm 1 (A1) calculates $\gamma[i]$ with $\alpha[i]$. First, we record the positions of the chiplet inserted by two sequences, called insertion position 1 (IP_1) and insertion position 2 (IP_2), and record the gap between the insertion positions of these two sequences (IP_{gap}). Then, using the insertion position of the node with the minimum estimated HPWL ($IP_{1\,min}$, $IP_{2\,min}$, $IP_{gap\,min}$), we calculate the difference between IP and IP_{min} to get ΔIP_1, ΔIP_2, and ΔIP_{gap}. Finally, α is half of the sum of ΔIP_1, ΔIP_2, and ΔIP_{gap}. The time complexity for $\gamma[i]$ is $O(1)$ for each node i at the same level.

5.3 Parallel B&B on SP-Tree

The B&B method consists of two parts. In the branching part, we traverse SP-Tree using depth-first search. First, we insert a chiplet into SP-Tree at a time. For each node, we generate four new nodes of the partial SP node and n_k^2 nodes for the rotation node. In the bounding (pruning) part, we prune the nodes which do not meet

Algorithm 1 Modified coefficient γ on the same level of partial SP

Input: integer array $[1, 2, ..., n^2]$ IP_1, IP_2
float array $[1, 2, ..., n^2]$ TWL
Output: float array $[1, 2, ..., n^2]$ γ

integer array $[1, 2, ..., n^2]$ α
$IP_{1min} = |\min(IP_1)|$;
$IP_{2min} = |\min(IP_2)|$;
$n = n_k$;
for each $i \in [1, n^2]$ **do**
  $IP_{gap}[i] = IP_1[i] - IP_2[i]$
  $IP_{gap\ min} = IP_{1\ min} - IP_{2\ min}$
  $\alpha[i] = (|IP_1[i] - IP_{1\ min}| + |IP_2[i] - IP_{2\ min}| + |IP_{gap}[i] - IP_{gap\ min}|)/2$;
end for
for each $i \in [1, n^2]$ **do**
  $\gamma[i] = 1 - \frac{\alpha[i]}{n_k}$;
end for

placement constraints, the nodes with estimated HPWL larger than the current best TWL, and the nodes which do not satisfy the method in Section 5.2.3.

Moreover, we implement the B&B algorithm with parallel processing by using OpenMP [12]. We assign stacks to different threads for processing. When updating the currently best TWL, we suspend other threads (by setting the barrier in OpenMP) to ensure efficiency on the B&B method.

6 POST CHIPLET PLACEMENT (POST-CP) WITH THERMAL CONSIDERATION

6.1 Chiplet thermal simulation

To calculate the thermal profile of 2.5D IC efficiently and accurately, based on the duality between thermal and electrical quantities [13] and the finite difference method [14], we solve temperatures by building the equation

$$\mathbf{G}T = P, \tag{5}$$

where $\mathbf{G}$ is the thermal conductance matrix, T is the temperature vector, and P is the power source vector.

To model the heat transfer, as shown in Fig. 1, five layers, substrate, C4 bump, interposer, μ-bump and chiplet, are included. The mesh is $64 \times 64 \times 5$. The heat transfer coefficients for the top and bottom sides of the 2.5D IC are 8700 and 2017 $\mathrm{W/(m^2K)}$ for modeling the primary heat flow to the heat sink and secondary heat flow to PCB [15], and it is adiabatic for lateral sides of the 2.5D IC. We solve T of chiplets by using a sparse matrix solver, SuperLU 5.3.0 [16]. The errors of maximum temperature for the experiments are less than 1% compared to the commercial tool Icepak [17]. Since it is time-consuming (taking hours) while the proposed framework integrates with Icepak, we solve the thermal profile of 2.5D IC as mentioned above.

6.2 Placement refinement with thermal effects

After completing SP-CP, some placements might have shorter wirelength but violate T_{cstr}. To trade off their wirelength and operating temperatures, we perform the refinement procedure to satisfy T_{cstr} by slightly modifying the placements with increasing TWL $< \eta\%$ compared to TWL_{opt}. Then, we select the placement that has the minimum wirelength and also satisfies T_{cstr}. If none of them satisfies T_{cstr}, we choose the minimum cost from (4) of the placements with increasing TWL $< \eta\%$.

The refinement of a placement consists of two types of movement. One is moving one chiplet at a time without alternating other chiplets as illustrated in Fig. 6(a). The other is moving all chiplets together

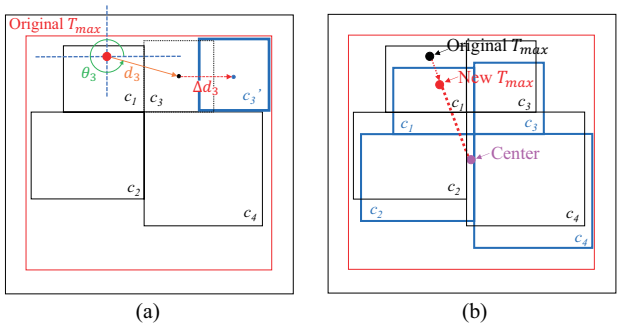

Figure 6: Placement refinement. (a) Move one chiplet (c_3); (b) Move all chiplets (c_1–c_4).

toward the center but sticking to their same relative positions shown in Fig. 6(b). Generally, the rising temperature is highly related to the distance of power sources and their values. Hence, we plan to move away some chiplets which influence temperatures and have less impact on wirelength.

Given a placement result violating the thermal constraint and having its maximum temperature value ($T_{\max}$) and position, we calculate the contribution to $T_{\max}$ by each c_i ($T_{\max,i}$) via superposition and calculate the wirelength increase because of moving c_i by a certain distance (Δd_i). We also simply assume that the temperature influence between two chiplets is proportional to the reciprocal of their distance. Then, the value ($\delta T_i/\delta W_i$) is calculated by dividing the temperature variations and the increasing wirelength per unit moving length of each c_i to measure its effectiveness. And we choose the c_m that has the lowest value ($\delta T_m/\delta W_m$) to be the moving candidate because it is the most effective chiplet for trading off temperatures and wirelength. Then, we choose a suitable moving length that is less than the estimated required length for satisfying the thermal constraint.

After moving one chiplet, we renew the value and position of $T_{\max}$. Finally, to increase the heat dissipation region of $T_{\max}$, we move all chiplets together in the direction from the position of $T_{\max}$ to the center of the interposer by a certain distance. We iteratively perform the above steps until the maximum temperature meets the threshold temperature or the chiplets cannot be moved such as encountering the boundaries or other chiplets.

We summarize the steps of the refinement procedure as follows.

1) Calculate $T_{\max}$ and its position of a given placement by using the simulation in Section 6.1.
2) Define and calculate the thermal gain of each c_i, $g_i = T_{\max,i}/P_i$, i=1~n. P_i is the power of c_i and $T_{\max} = \sum_{i=1}^{n} g_i P_i$.
3) Partially differentiate $T_{\max,d} = \sum_{i=1}^{n} d_i g_i P_i/(d_i + \Delta d_i)$ on each Δd_i and have $\delta T_i = -g_i P_i/d_i$. Here, d_i is the distance between c_i and the position of $T_{\max}$.
4) Calculate the increasing HPWL per unit moving length of c_i to be $\delta W_i = (|\cos\theta_i| + |\sin\theta_i|) * (\#net_i)$.
5) Calculate $\delta T_i/\delta W_i$ for each c_i and choose c_m having the lowest value.
6) Calculate $\Delta d_{m,\max}$ to be $(T_{\max} - T_{cstr}) \div (g_i P_i/d_i)$.
7) Move c_m away from the point of $T_{\max}$ with a suitable distance $\Delta d_m \le \Delta d_{m,\max}$.
8) Renew the position and value of $T_{\max}$.
9) Move all chiplets simultaneously along the direction of interposer center from the position of $T_{\max}$, and the displacement is $r\%$ of the distance between the position of $T_{\max}$ and the interposer center (the default value of r is 1).
10) Repeat the above steps iteratively until the temperature meets the thermal threshold, or the chiplet cannot be moved.

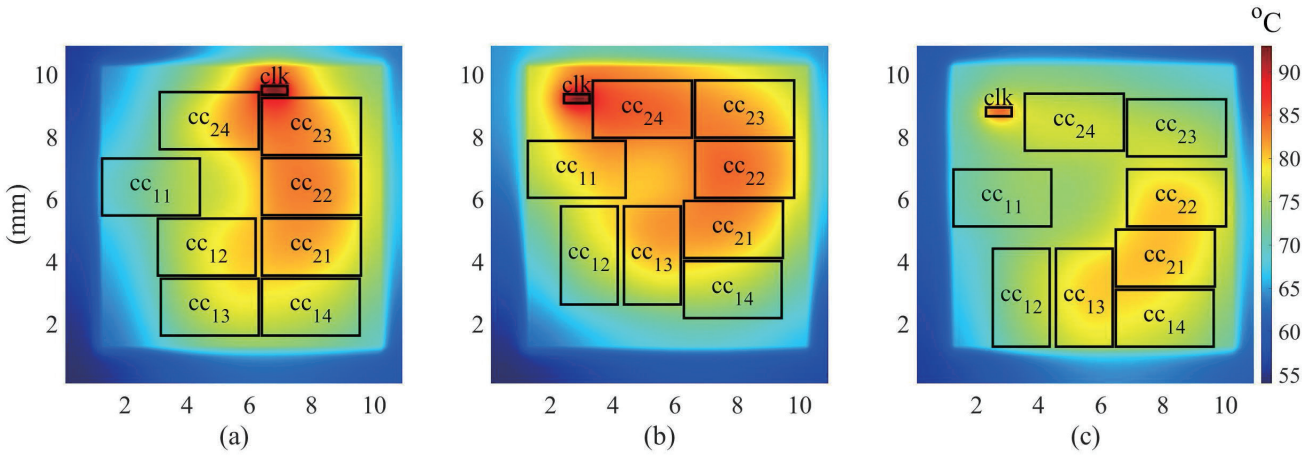

Figure 7: Thermal maps on apte_scaled30. (a) The optimal placement before Post-CP with TWL = 0.40872 m (+0.00%) and maxT = 92.669 ℃; (b) The sub-optimal placement before Post-CP with TWL = 0.41076 m (+0.49%) and maxT = 90.308 ℃; (c) The sub-optimal placement after Post-CP with TWL = 0.42707 m (+4.48%) and maxT = 84.455 ℃.

11) If none of placements with increasing TWL $< \eta\%$ is satisfied T_{cstr}, then choose the minimum cost from (4) of those placements.

7 EXPERIMENTAL RESULTS

We implement the proposed framework in C++ language with compiler gcc 8.3.1, and execute it on a Linux workstation with Intel CPU Xeon E5-2620 v4 with eight cores at 2.10 GHz (disabled hyper-threading). Three benchmarks are used from [8, 18] for the experiments. One is interposer-based chiplets cases [7] with four, six, and eight chiplets (t4/t6/t8). Another is from the MCNC benchmark [19] with nine, ten, and eleven chiplets (apte/xerox/hp). The other is the modified MCNC benchmark scaled by [8] for practical 2.5D IC design with scaled interposer size and terminals within whitespace from 5% to 20%.

7.1 Wirelength-driven placement compared to [8]

To demonstrate the performance of SP-CP, we set the minimum gap w_{space} = 0 mm which is the same setting as [7, 8]. [7, 8] performed chiplet placement for the optimal TWL, and its B&B with the advanced pruning technique [8] is $10^4\times$ faster than the enumeration-based placer [7] at most. To compare SP-CP with [8], we run its released tool from [18]. In Table 1, for SP-CP with PDSP, the optimized TWL of SP-CP is at most 1.035% better than that of [8]. The better optimized TWL in SP-CP with PDSP results from applying the method in Section 5.2.1. Compared to [8], the speed-up of SP-CP with PDSP can be up to 103.824×. Moreover, in the xerox case, we stop searching for the optimal solution after 12 hours for [8], and SP-CP with PDSP obtains the optimal solution by taking 9.72 hours and finds a better placement (−0.430 % in reducing TWL). One reason why we can achieve the speed-up is that partial SP nodes in SP-Tree can be placed while some topology nodes in CSP-Tree [8] can not be packed into a placement. The other reason is that one partial/complete SP in SP-Tree might cover multiple topology nodes in CSP-Tree [8] described in Section 5.1.

For SP-CP with PDSP & A1 presented in Section 5.2.3, it obtains the same optimized TWL of SP-CP with PDSP. Compared with [8], the runtime is accelerated up to 156× (in the xerox case, 6.16 hours to obtain the best TWL). It shows that PDSP & A1 can help prune more nodes while obtaining the optimized TWL for all cases.

7.2 Placement with thermal consideration

Since [7, 8] did not handle the constraint of minimum gap w_{space}, we compare placement results with thermal consideration between "SP-CP" (the framework in Fig. 2 without thermal consideration) and "SP-CP & Post-CP" (the framework in Fig. 2). w_{space}, η, and ϕ are set to 0.1 mm [5], 10, and 0.5, respectively. Because SP-CP cannot obtain the solution for cases (apte_scaled10, apte_scaled5, xerox_scaled10, xeror_scaled5, and all scaled hp cases in Table 1 with w_{space} = 0.1 mm) due to the constraint of w_{space} with less whitespace, we build scaled cases for the MCNC benchmark within whitespace 25% and 30%. The power density is assigned randomly from 10^5 to 10^7 W/m^2 [20] for each chiplet, and T_{cstr} is set to 85 ℃ [10]. As illustrated in Table 2, with Post-CP in Section 6, the maximum temperature reduction can be up to 8.214 ℃. Even two cases (apte_scaled20 and apte_scaled15) cannot meet T_{cstr}, with the minimum cost in (4) of the searched placements in Post-CP, our framework can reduce 3 to 5 ℃ with increasing wirelength less than 6.7%. Moreover, the average runtime overhead is about 50 seconds, and most of the runtime overhead is to solve the temperatures of chiplets while applying thermal simulation in Section 6.1. Fig. 7 plots the case apte_scaled30 which has the maximum reduced temperature among all the cases. Fig. 7.(a) shows its optimal wirelength solution. Fig. 7.(b) and Fig. 7.(c) are its sub-optimal wirelength placements before and after Post-CP, respectively. It shows different topologies between the optimal placement and sub-optimal placements. And the maximum temperatures of sub-optimal placements are lower. During Post-CP, the chiplet "clk" is moved away from other chiplets, and the whole chiplets move together toward the middle slightly for better thermal performance.

8 CONCLUSION

We have developed an efficient chiplet placer, SP-CP, with sequence pair based tree, advanced placement/pruning techniques, and parallel B&B method for the optimal placement. Moreover, with thermal consideration, we have built Post-CP to reduce the maximum temperatures with maximum allowable increasing wirelength. The results show that SP-CP can speed up at most two orders of magnitude with better TWL (reducing 1.035% HPWL) compared to the prior-art. With Post-CP, the maximum temperature can be reduced to 8.214 ℃ with an average increase of 5.376% TWL.

ACKNOWLEDGEMENT

This work was partially supported by the ministry of science and technology (MOST) in Taiwan, 108-2221-E-009-093-MY2 and 110-2221-E-A49-152. And this work was partially supported by MediaTek Inc. and collaborated with the PTMFO department from MediaTek in Taiwan.

Table 1: Results on wirelength driven placement with w_{space} = 0 mm

Case	Chiplets	Pins	Nets	Terminals	[8]		SP-CP			Comparison		
					TWL (m)	Time (s)	TWL (m)	w/ PDSP Time (s)	w/ PDSP & A1 Time (s)	TWL Diff. (%)	w/ PDSP Speedup (×)	w/ PDSP & A1 Speedup (×)
t4_s	4	15611	1808	789	10.87000	0.263	10.87000	0.127	0.123	0.000	2.071	2.138
t4_m	4	91005	5326	1174	38.14000	0.577	38.14000	0.226	0.214	0.000	2.553	2.696
t4_b	4	223781	12265	1033	58.92000	1.180	58.92000	0.411	0.396	0.000	2.871	2.980
t6_s	6	20138	1720	639	9.01000	0.366	9.01000	0.122	0.098	0.000	3.000	4.572
t6_m	6	121935	7123	1162	33.77000	1.791	33.77000	0.439	0.392	0.000	4.080	4.572
t6_b	6	229228	14264	1192	62.71000	2.470	62.71000	0.945	0.886	0.000	2.614	2.788
t8_s	8	18689	1918	882	23.51000	1.341	23.51000	0.192	0.165	0.000	6.984	8.127
t8_m	8	159149	8391	1391	36.39000	2.058	36.39000	0.711	0.683	0.000	2.895	3.013
t8_b	8	306057	12593	1049	66.61000	12.116	66.61000	1.094	0.933	0.000	11.075	12.986
apte_scaled20	9	287	97	73	0.37701	17.620	0.37704	9.782	8.501	0.008	1.801	2.073
apte_scaled15	9	287	97	73	0.37320	14.087	0.37265	9.559	8.457	-0.147	1.474	1.666
apte_scaled10	9	287	97	73	0.36630	12.192	0.36551	9.430	6.963	-0.216	1.293	1.751
apte_scaled5	9	287	97	73	0.37526	31.774	0.37526	6.299	5.270	0.000	5.044	6.029
xerox_scaled20	10	698	203	2	0.36399	22881.822	0.36398	220.391	146.051	-0.003	103.824	156.670
xerox_scaled15	10	698	203	2	0.37876	4634.995	0.37861	111.165	88.330	-0.040	41.695	52.474
xerox_scaled10	10	698	203	2	0.41998	3685.743	0.41830	101.963	65.245	-0.400	36.148	56.491
xerox_scaled5	10	698	203	2	0.43747	1853.318	0.43747	64.173	50.623	0.000	28.880	36.610
hp_scaled20	11	309	83	45	0.14002	17.315	0.13992	10.044	9.524	-0.071	1.724	1.818
hp_scaled15	11	309	83	45	0.14342	9.649	0.14194	2.851	2.580	-1.035	3.384	3.740
hp_scaled10	11	309	83	45	0.14377	5.140	0.14295	1.729	1.641	-0.570	2.974	3.132
hp_scaled5	11	309	83	45	0.16401	4718.180	0.16401	930.729	836.377	0.000	5.069	5.641
apte	9	287	97	73	0.43751	1397.697	0.43751	323.930	186.660	0.000	4.315	7.488
xerox	10	698	203	2	0.36587	>12hr	0.36430	34994.700	22187.000	-0.430	NA	NA
hp	11	309	83	45	0.15026	14281.288	0.15014	438.440	348.699	-0.080	32.573	40.956
Avg.										-0.124	13.406	18.242

Table 2: Results on placement with thermal consideration and w_{space} = 0.1 mm

Case	SP-CP			SP-CP & Post-CP					
	TWL (m)	Max. Temp. (℃)	Time (s)	TWL (m)	Max. Temp. (℃)	Time (s)	Increasing TWL (%)	Max. Temp. Reduction (℃)	Runtime Overhead (s)
apte_scaled30	0.40872	92.669	17.504	0.42707	84.455	75.605	4.490	8.214	58.101
apte_scaled25	0.40213	91.889	15.803	0.43193	84.979	87.846	7.411	6.910	72.043
apte_scaled20	0.39267	99.579	9.782	0.41887	94.562	83.610	6.672	5.017	73.828
apte_scaled15	0.41692	95.123	43.177	0.43818	91.556	88.489	5.099	3.567	45.312
xerox_scaled30	0.40664	88.631	149.792	0.42894	83.603	184.882	5.484	5.028	35.090
xerox_scaled25	0.42087	89.632	71.292	0.45233	84.673	187.037	7.475	4.959	115.745
xerox_scaled20	0.48135	87.810	192.405	0.50846	84.091	220.035	5.632	3.719	27.630
xerox_scaled15	0.51508	86.778	334.773	0.56097	84.918	344.944	8.909	1.860	10.171
hp_scaled30	0.16144	86.284	10.252	0.16362	84.773	23.564	1.350	1.511	13.312
hp_scaled25	0.19377	85.050	265.859	0.19617	84.584	320.769	1.239	0.466	54.910
Avg.							5.376	4.125	50.614

REFERENCES

[1] M.-F. Chen, F.-C. Chen, W.-C. Chiou, and C. Doug, "System on integrated chips (SoIC) for 3D heterogeneous integration," in *IEEE ECTC*, 2019.
[2] 2D vs. 2.5D vs. 3D ICs, https://www.eetimes.com/2d-vs-2-5d-vs-3d-ics-101/.
[3] S. Naffziger, K. Lepak, M. Paraschou, and M. Subramony, "2.2 AMD chiplet architecture for high-performance server and desktop products," in *IEEE ISSCC*, 2020.
[4] M.-S. Lin, T.-C. Huang, C.-C. Tsai, K.-H. Tam, C.-H. Hsieh, T. Chen, W.-H. Huang, J. Hu, Y.-C. Chen, S. K. Goel, C.-M. Fu, S. Rusu, C.-C. Li, S.-Y. Yang, M. Wong, S.-C. Yang, and F. Lee, "A 7nm 4GHz Arm®-core-based CoWoS® chiplet design for high performance computing," in *Symposium on VLSI Circuits*, 2019.
[5] R. Chaware, K. Nagarajan, and S. Ramalingam, "Assembly and reliability challenges in 3D integration of 28nm FPGA die on a large high density 65nm passive interposer," in *IEEE ECTC*, 2012.
[6] Y.-K. Ho and Y. W. Chang, "Multiple chip planning for chip-interposer codesign," in *Proc. DAC*, 2013.
[7] W. H. Liu, M. S. Chang, and T. C. Wang, "Floorplanning and signal assignment for silicon interposer-based 3D ICs," in *Proc. DAC*, 2014.
[8] S. Osmolovskyi, J. Knechtel, I. L. Markov, and J. Lienig, "Optimal die placement for interposer-based 3D ICs," in *Proc. ASP-DAC*, 2018.
[9] A. Coskun, F. Eris, A. Joshi, A. B. Kahng, Y. Ma, A. Narayan, and V. Srinivas, "Cross-layer co-optimization of network design and chiplet placement in 2.5-D systems," *IEEE TCAD*, vol. 39, no. 12, pp. 5183–5196, 2020.
[10] Y. Ma, L. Delshadtehrani, C. Demirkiran, J. L. Abellan, and A. Joshi, "TAP-2.5D: A thermally-aware chiplet placement methodology for 2.5D systems," in *Proc. DATE*, 2021.
[11] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," *IEEE TCAD*, vol. 15, no. 12, pp. 1518–1524, 1996.
[12] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan, and J. McDonald, *Parallel programming in OpenMP*. Morgan kaufmann, 2001.
[13] T. L. Bergman, T. L. Bergman, F. P. Incropera, D. P. Dewitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.
[14] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. S. Kang, *Electrothermal analysis of VLSI systems*. Springer Science & Business Media, 2000.
[15] S. S.-Y. Liu, R.-G. Luo, S. Aroonsantidecha, C.-Y. Chin, and H.-M. Chen, "Fast thermal aware placement with accurate thermal analysis based on green function," *IEEE TVLSI*, vol. 22, no. 6, pp. 1404–1415, 2013.
[16] SuperLU 5.3.0, https://portal.nersc.gov/project/sparse/superlu/.
[17] ANSYS Icepak, https://www.ansys.com/products/electronics/ansys-icepak.
[18] S. Osmolovskyi and Jens Lienig, "Placement framework for interposer-based 3D ICs," 2017, https://www.ifte.de/english/research/interposer-design/index.html.
[19] MCNC benchmark, https://s2.smu.edu/~manikas/Benchmarks/MCNC_Benchmark_Netlists.html.
[20] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proc. ICCAD*, 2004.