IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 1, 2020, accepted November 11, 2020, date of publication November 30, 2020, date of current version December 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041365

An Improved Matrix Generation Framework for Thermal Aware Placement in VLSI

SOMNATH ROY CHOUDHURY AND SAMBHU NATH PRADHAN

Department of Electronics and Communication Engineering, National Institute of Technology Agartala, Agartala 799046, India

Corresponding author: Somnath Roy Choudhury (som.vlsi@gmail.com)

This work has been carried out in the research facility established by the Ministry of Electronics & Information Technology (MeitY), Government of India under the Special Manpower Development Program for Chip to System Design (SMDP-C2SD) Project at the Department of ECE, National Institute of Technology Agartala, India.

ABSTRACT Since hotspots and temperature gradients are reliability and performance-critical issues in processors, thermal awareness finds a vital place in the processor design cycle. Incorporating thermal awareness at the level of physical design, this work proposes a new, fast, and efficient thermal aware placement algorithm called the Thermal Aware Matrix Placement Optimizer (TAMPO) for gate arrays. The algorithm TAMPO is composed of the following components: an improved heat diffusion aware cell arrangement technique called the Initial Matrix Generator (IMaGe), a unique stochastic thermal model based on a thermally improved interpretation of the well known Matrix Synthesis Problem (MSP) and a Simulated Annealing (SA) engine for finding the global optimum solution. TAMPO targets to reduce the peak temperature while maintaining improved values of temperature gradients and the standard deviation in cell temperature with respect to the average chip temperature. This work also presents a methodology, the Co-optimized TAMPO, which extends the concept of TAMPO to simultaneously optimize the thermal attributes and the wirelength of a chip. The proposed algorithms realize a placement in matrix arrangement and upon experimentation on the ISCAS89 benchmark circuits encouraging results have been obtained.

INDEX TERMS Hotspots, temperature gradients, simulated annealing, matrix synthesis problem, gate arrays.

I. INTRODUCTION

Thermal management has always been a challenge in VLSI design, since processors containing billions of transistors and pulsating at gigahertz frequencies, generate a huge amount of heat in small areas and often suffer from hotspots and high temperature gradients. Hotspots degrade the reliability of chips by aggravating failure mechanisms like electromigration, stress migration, gate oxide breakdown, and thermal cycling [1]. Increasing junction temperature exponentially increases the stand by leakage power dissipation which may even cause thermal runaway and subsequently permanent damage of IC [2]. Moreover, temperature gradient leads to ailments like clock skew and cross talk induced noise in interconnects [3]. These factors impose substantial cost overhead for implementing cooling solutions. Data centers waste a massive 40% of energy under cooling [4]. Hence it becomes very imperative to implement thermal aware techniques at the different levels of abstraction of the VLSI design cycle. For power-constrained ICs, placement and floorplanning are vital levels where the thermal awareness can be further incorporated, in addition to the optimization of traditional design objectives.

The thermal aware placement of gate array IC has been recognized in [5] as a combinatorial optimization problem referred to as the Matrix Synthesis Problem (MSP), which targets to minimize the temperature of the hottest region and also tries to ensure an even heat distribution throughout the gate array IC. The MSP is about generating an optimal arrangement of a given set of numbers in a matrix such that the maximum sum of submatrices of a particular size (t x t) is minimized. It considers the numbers as the heat dissipated by cells located at the corresponding matrix locations and the highest submatrix sum as the hottest region in the chip. The Simple Approximation algorithm, also termed algorithm A1 has been proposed in [5] as a solution to the MSP. Algorithm A1 sorts the cells in decreasing order of heat and distributes them successively in to four groups viz. $G_0 = \{L_0\}$, $G_1 = \{L_1\}$, $G_2 = \{L_2\}$, $G_3 = \{L_3\}$, such that the order of the groups on the basis of the heat of the constituent

The associate editor coordinating the review of this manuscript and approving it for publication was Yuh-Shyan Hwang.

elements is $G_0 \geq G_1 \geq G_2 \geq G_3$. The placement has been realized as a square matrix in which every (t x t) sub matrix (with t = 2) is constructed by randomly selecting one element from each of the four groups thereby ensuring that every high power cell is placed along with moderate and low power cells. Basically this grouping and sub matrix formation results in uniform heat or power distribution throughout the IC and a reduction in hotspots.

Electronic Design Automation (EDA) tools like Hotspot [6], [7], [28] provide an accurate temperature estimation of functional blocks by solving a lumped thermal RC network of the stacked-layer package scheme of IC. But Hotspot tool takes considerable time overhead in solving the temperature value from the compact thermal model. In this regards the MSP placement methodology proposed in [5] smartly avoids the expensive computation of actual temperature estimation during the optimization process and provides a fast solution to the thermal aware placement problem of gate array ICs. However, according to [9] and [10] the temperature of a functional block is influenced by the length of its shared boundary as well as the power density differences with the neighboring blocks. Moreover, according to [8] the die boundary wall also influences the local thermal characterization. The die boundary wall has an adiabatic influence on the internally generated heat and the temperature of a functional block is characterized by its relative position from the die boundary wall. These thermal considerations have been ignored by the works done till now on MSP [5], [11]–[14] for the thermal aware placement. Moreover, the experimental works done in [5], [11]–[14] have not quantified the thermal improvements in terms of temperature. The experimental works done in [5], [11]–[14] considers only square matrix formation for cell placement thereby incurring more dummy cells to make up a square number of total cells. Dummy cells are proxy blocks having zero power dissipation and the inclusion of more dummy cells increase the chip area. Hence in this work, we have modified the Matrix Synthesis Problem (MSP) by incorporating the missing thermal considerations and developed a fast, new and efficient placement algorithm capable of generating optimal square matrix as well as minimum-cell matrix (having lesser dummy cells) thermal aware placement for gate array ICs.

Contributions: Our work makes the following salient contributions.

1) It presents a transformation algorithm, the Gate Array Packer (GAP) which maps a logic circuit into a gate array architecture composed of basic cells and clusters.

2) It depicts an algorithm, the Initial Matrix Generator (IMaGe), for a heat diffusion aware even power distribution scheme and the construction of initial placement of cells. The IMaGe generates square matrix and minimum-cell matrix placements.

3) It presents an improved thermal model, designed by modifying the thermal metric of the MSP viz. local summative heat of a submatrix region or thermal zone along with its reflective heat component from the adiabatic die boundary wall.

4) It presents the placement algorithm, Thermal Aware Matrix Placement Optimizer (TAMPO) which incorporates the IMaGe, the proposed thermal model, and a Simulated Annealing (SA) engine to give a fast optimization in peak temperature, temperature gradient, and the standard deviation in temperature of cells in matrix placement.

5) It further extends the concept of TAMPO to generate a placement strategy called the Co-optimized TAMPO which optimizes the thermal attributes and the wirelength simultaneously.

6) Finally it reconstructs few reference placement algorithms; one based on the Hotspot tool [6], [7], [28], one based on the Simple Approximation algorithm [5], and the other based on the thermal aware placement algorithm [11] to validate the performance of the proposed placement algorithms.

II. RELATED WORKS

Some of the efforts made towards the development of efficient thermal aware placement and floorplan algorithms are discussed as follows. Paper [5] introduces the Matrix Synthesis Problem (MSP) for thermal aware placement of gate arrays. Thermal aware placement of standard cells and gate arrays has also been presented in [11]–[13] by implementing the MSP where the proposed algorithms assume square matrix placement and try to minimize the peak (t x t) submatrix sum to reduce the hotspots. The algorithms in [11] consider the matrix elements as the power dissipation of cells and also show an approach to simultaneously optimize the wirelength and hotspots. Authors in [12] and [13] assume the matrix elements as the temperatures of cells and every (t x t) submatrix as a window. Work done in [12] implements a multi-objective optimization heuristic based on the game theory to simultaneously minimize the maximum window temperature and the wirelength. The methodology in [13] also employs a game theory based approach to minimize the maximum window temperature and the deviation of maximum temperature. The MSP has been further applied for the thermal aware 3D IC placement of standard cells in [14]. Using a simulated annealing based approach and considering every active layer as a square matrix with matrix elements as the power density of cells, the algorithm mitigates the hotspots by reducing the maximum aggregate of every (t x t) submatrix. The algorithm in [14] also simultaneously optimizes the wire length and the TSV. However, the temperature of functional blocks is placement dependent and cannot be taken as input to the placement problem as has been considered in [12] and [13]. Moreover, the submatrix sum of quantities like heat, power, and power density considered in [5], [11], and [14] respectively alone cannot account for the degree of hotness of a region. Work done in [15] proposes a 3D MSP cube model for the thermal aware mapping of 3D NOC architecture. Also utilizing a genetic algorithm approach, it achieves improvements in temperature deviation, power, and delay. Work done in [16] shows a thermal aware-placer based on thermal force and thermal padding methods for optimizing the peak temperature and the temperature gradient. It also uses the modified nodal analysis to estimate the temperature from the equivalent thermal circuit of the chip.

Authors in [17] present a pre-RTL tool framework based on the simulated annealing heuristic to optimize the peak temperature and chip area of SoC and chip multiprocessor floorplan. A thermal aware floorplan algorithm has also been presented in [18] for optimizing the temperature-dependent wire delay, routing congestion, reliability factors, area, and peak temperature of the chip based on the HotFloorplan tool. Work done in [19] also portrays a thermal aware hybrid PSO-GA based floorplan algorithm for optimizing the area, wirelength, and temperature of the chip. Hotspot tool has been used in [17]–[19] for the temperature estimation. However, the temperature estimation methods employed in [16]–[19] during the optimization process incur a large computational budget and execution time. A relatively faster conjugate gradient method has been proposed in [20] for computing the temperature from the thermal model of the Hotspot tool. The floorplan algorithm in [8] avoids the computational expense of exact temperature estimation and employs a heat diffusion method to give a fast optimization of the peak temperature, area, and wirelength. A fast fixed outline thermal aware multilevel floorplan algorithm has also been presented in [21] which uses a power blurring analytical method to estimate the temperature and simultaneously optimizes the temperature and wire length of the chip.

III. MOTIVATION

As an example let us consider the matrix placement of a test case circuit containing 33 functional cells each of identical height (H = 0.0006 m) and width (W = 0.0008 m) just like gate array cells. The functional cells and their corresponding power dissipation values have been shown in Fig. 1. In order to obtain a thermal aware solution according to the Simple Approximation or algorithm A1, the cells at first have been sorted in descending order of power in a linear cell array and secondly grouped as shown in Fig. 2. Now considering a square matrix placement, the nearest square number which can accommodate 33 functional cells is $6^2 = 36$. Number of dummy cells (of zero power dissipation) to be added to the placement matrix is 36 - 33 = 3. The minimum or the best possible peak submatrix sum with t = 2 in a 6 x 6 placement matrix with the given set of cells (functional $\{C_i\}$ and dummy $\{D_i\}$) is 1.5 W. As shown in Fig. 3 some matrix placements have been constructed where the dotted envelopes in red and blue denote the submatrix regions with the peak sum for t = 2 and t = 3 respectively.

Consider Placement-1 as shown in Fig. 3a where the cells have been randomly distributed. According to the MSP, Placement-1 is a poor solution due to higher peak submatrix sum (3.6 W with t = 2) and uneven distribution in power dissipation. Following algorithm A1, one element (cell) is selected from each cell group $G_k = \{L_k\}$ within the linear cell array (in Fig. 2) and so placed in the (t x t) submatrix regions that the peak submatrix sum is minimized and Placement-2 in Fig. 3b is obtained. The peak submatrix sum in Placement-2 is 1.8 W (with t = 2) and this is the minimum value achievable by algorithm A1 (with t = 2) but it is not the best value (1.5W). Hence Placement-2 is one of the optimal solutions of algorithm A1 according to MSP with t = 2. Two more solutions viz. Placement-3 in Fig. 3c and Placement-4 in Fig. 3d have been constructed in accordance with the placement scheme of algorithm A1. Finally Placement-5 in Fig. 3e has been constructed according to our proposed Updated Placement Scheme - UPS (refer section VC). All the solutions viz. Placement-3, 4 and 5 are inferior solutions according to MSP since they have higher peak submatrix sum with respect to Placement-2 for both t = 2 and t = 3. The peak submatrix sum of the placements is available in Table 1.

Through experiments based on the placement matrices described in Fig. 3, the corresponding thermal maps and thermal attributes have been obtained and presented in Fig. 4 and Table 1 respectively. The description of the terms used for the analysis of the placement solutions are as follows: 'Max. Zonal Sum' represents the peak (t x t) submatrix sum, 'Avg. Temp.' is the average temperature of cells, 'Peak Temp.' is the maximum on-chip temperature, 'Min. Temp.' is the minimum temperature on the chip, 'Temp. Grad.' is the temperature gradient on the chip and 'Std. Dev. Temp.' is the standard deviation in cell temperature denoting the extent of the temperature variation of all cells from the average chip temperature. A lesser value of standard deviation in cell temperature implies an improved even temperature distribution throughout the chip. On the basis of the thermal maps and the experimental data in Table 1, the following observations have been made. (a) Placement-1 has the highest peak temperature, temperature gradient, and standard deviation in cell temperature. It is the worst placement and hence experimental data agree well with the MSP interpretation. (b) Placement-2 gives an improvement of about 9.22% in peak temperature, 53.4% in temperature gradient, and 59.2% in the standard deviation in cell temperature over Placement-1. It is a much better solution than Placement-1 and hence again experimental data agree well with the MSP interpretation. (c) Placement-3 gives an improvement of about 15.52% in peak temperature, 76.82% in temperature gradient, and 75.37% in the standard deviation in cell temperature over Placement-1. This makes Placement-3 even a better solution than Placement-2. (d) Placement-4 gives an improvement of about 15.29 % in peak temperature, 75.08% in temperature gradient, and 75.3% in the standard deviation in cell temperature over Placement-1. Hence Placement-4 is also a better solution than Placement-2. (e) Finally, Placement-5 gives an improvement of about 15.85 % in peak temperature, 84.78% in temperature gradient, and 87.48% in standard deviation in cell temperature over Placement-1 which makes Placement-5 the best placement. The experimental observations (c), (d), and (e) are unexplainable by the MSP interpretation.

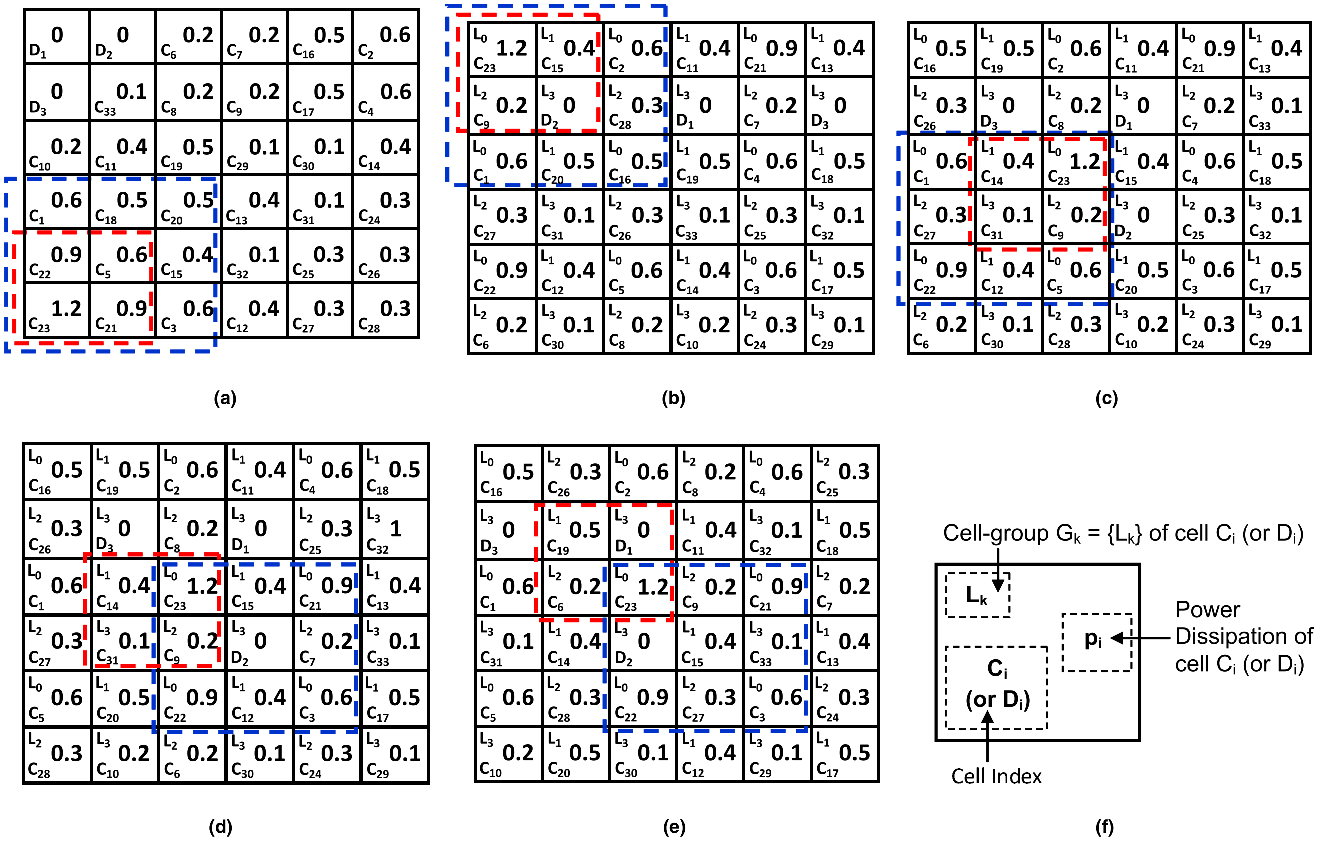
To find an answer to the experimental observations, let us focus on the thermo-resistive IC package model presented in

Functional Cell	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}	C_{27}	C_{28}	C_{29}	C_{30}	C_{31}	C_{32}	C_{33}
Power (watt)	0.6	0.6	0.6	0.6	0.6	0.2	0.2	0.2	0.2	0.2	0.4	0.4	0.4	0.4	0.4	0.5	0.5	0.5	0.5	0.5	0.9	0.9	1.2	0.3	0.3	0.3	0.3	0.3	0.1	0.1	0.1	0.1	0.1

FIGURE 1. Power dissipation values of the functional cells of a test case circuit.

$G_0 = \{L_0\} = \{C_{23},C_{21},C_{22},C_1,C_2,C_3,C_4,C_5,C_{16}\}$									$G_1 = \{L_1\} = \{C_{17},C_{18},C_{19},C_{20},C_{11},C_{12},C_{13},C_{14},C_{15}\}$									$G_2 = \{L_2\} = \{C_{24},C_{25},C_{26},C_{27}, C_{28},C_6,C_7,C_8,C_9\}$									$G_3 = \{L_3\} = \{C_{10},C_{29},C_{30},C_{31},C_{32},C_{33},D_1,D_2,D_3\}$								
C_{23}	C_{21}	C_{22}	C_1	C_2	C_3	C_4	C_5	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{24}	C_{25}	C_{26}	C_{27}	C_{28}	C_6	C_7	C_8	C_9	C_{10}	C_{29}	C_{30}	C_{31}	C_{32}	C_{33}	D_1	D_2	D_3
1.2	0.9	0.9	0.6	0.6	0.6	0.6	0.6	0.5	0.5	0.5	0.5	0.5	0.4	0.4	0.4	0.4	0.4	0.3	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0
Functional Cells																																	Dummy Cells		

FIGURE 2. Linear array of the cell groups in descending order of power dissipation.



(a)

0 D_1	0 D_2	0.2 C_6	0.2 C_7	0.5 C_{16}	0.6 C_2
0 D_3	0.1 C_{33}	0.2 C_8	0.2 C_9	0.5 C_{17}	0.6 C_4
0.2 C_{10}	0.4 C_{11}	0.5 C_{19}	0.1 C_{29}	0.1 C_{30}	0.4 C_{14}
0.6 C_1	0.5 C_{18}	0.5 C_{20}	0.4 C_{13}	0.1 C_{31}	0.3 C_{24}
0.9 C_{22}	0.6 C_5	0.4 C_{15}	0.1 C_{32}	0.3 C_{25}	0.3 C_{26}
1.2 C_{23}	0.9 C_{21}	0.6 C_3	0.4 C_{12}	0.3 C_{27}	0.3 C_{28}

(b)

L_0 1.2 C_{23}	L_1 0.4 C_{15}	L_0 0.6 C_2	L_1 0.4 C_{11}	L_0 0.9 C_{21}	L_1 0.4 C_{13}
L_2 0.2 C_9	L_3 0 D_2	L_2 0.3 C_{28}	L_3 0 D_1	L_2 0.2 C_7	L_3 0 D_3
L_0 0.6 C_1	L_1 0.5 C_{20}	L_0 0.5 C_{16}	L_1 0.5 C_{19}	L_0 0.6 C_4	L_1 0.5 C_{18}
L_2 0.3 C_{27}	L_3 0.1 C_{31}	L_2 0.3 C_{26}	L_3 0.1 C_{33}	L_2 0.3 C_{25}	L_3 0.1 C_{32}
L_0 0.9 C_{22}	L_1 0.4 C_{12}	L_0 0.6 C_5	L_1 0.4 C_{14}	L_0 0.6 C_3	L_1 0.5 C_{17}
L_2 0.2 C_6	L_3 0.1 C_{30}	L_2 0.2 C_8	L_3 0.2 C_{10}	L_2 0.3 C_{24}	L_3 0.1 C_{29}

(c)

L_0 0.5 C_{16}	L_1 0.5 C_{19}	L_0 0.6 C_2	L_1 0.4 C_{11}	L_0 0.9 C_{21}	L_1 0.4 C_{13}
L_2 0.3 C_{26}	L_3 0 D_3	L_2 0.2 C_8	L_3 0 D_1	L_2 0.2 C_7	L_3 0.1 C_{33}
L_0 0.6 C_1	L_1 0.4 C_{14}	L_0 1.2 C_{23}	L_1 0.4 C_{15}	L_0 0.6 C_4	L_1 0.5 C_{18}
L_2 0.3 C_{27}	L_3 0.1 C_{31}	L_2 0.2 C_9	L_3 0 D_2	L_2 0.3 C_{25}	L_3 0.1 C_{32}
L_0 0.9 C_{22}	L_1 0.4 C_{12}	L_0 0.6 C_5	L_1 0.5 C_{20}	L_0 0.6 C_3	L_1 0.5 C_{17}
L_2 0.2 C_6	L_3 0.1 C_{30}	L_2 0.3 C_{28}	L_3 0.2 C_{10}	L_2 0.3 C_{24}	L_3 0.1 C_{29}

(d)

L_0 0.5 C_{16}	L_1 0.5 C_{19}	L_0 0.6 C_2	L_1 0.4 C_{11}	L_0 0.6 C_4	L_1 0.5 C_{18}
L_2 0.3 C_{26}	L_3 0 D_3	L_2 0.2 C_8	L_3 0 D_1	L_2 0.3 C_{25}	L_3 1 C_{32}
L_0 0.6 C_1	L_1 0.4 C_{14}	L_0 1.2 C_{23}	L_1 0.4 C_{15}	L_0 0.9 C_{21}	L_1 0.4 C_{13}
L_2 0.3 C_{27}	L_3 0.1 C_{31}	L_2 0.2 C_9	L_3 0 D_2	L_2 0.2 C_7	L_3 0.1 C_{33}
L_0 0.6 C_5	L_1 0.5 C_{20}	L_0 0.9 C_{22}	L_1 0.4 C_{12}	L_0 0.6 C_3	L_1 0.5 C_{17}
L_2 0.3 C_{28}	L_3 0.2 C_{10}	L_2 0.2 C_6	L_3 0.1 C_{30}	L_2 0.3 C_{24}	L_3 0.1 C_{29}

(e)

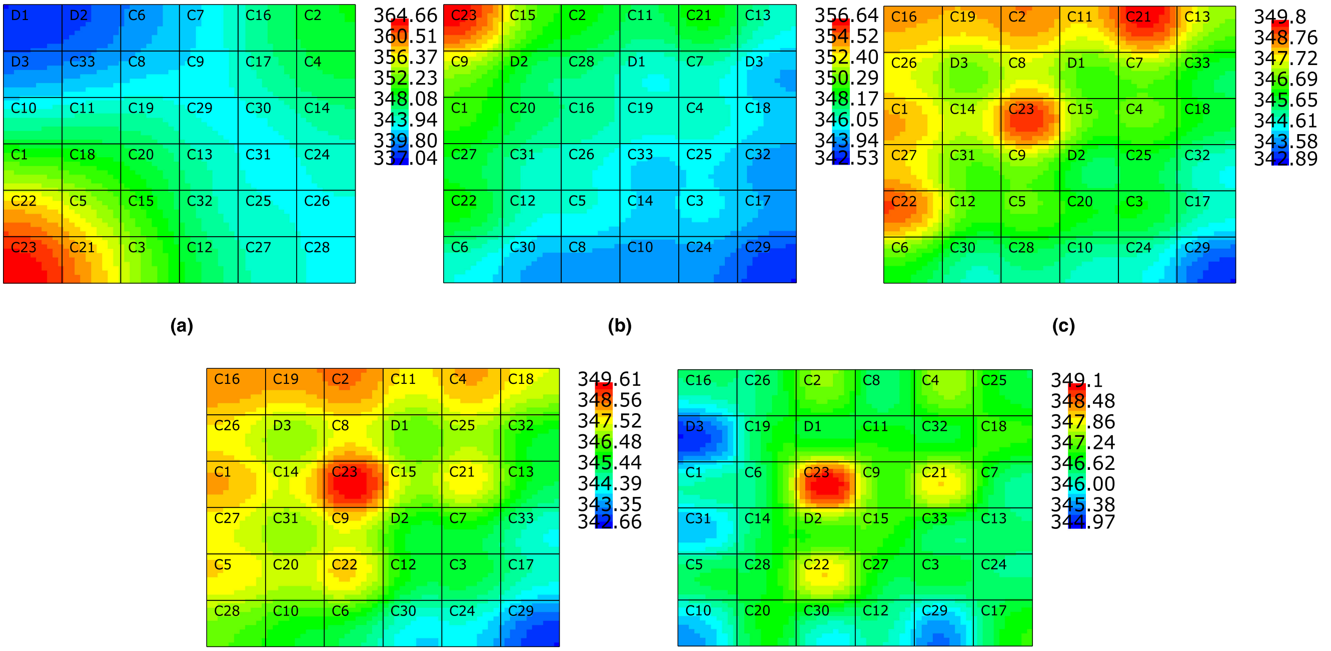
L_0 0.5 C_{16}	L_2 0.3 C_{26}	L_0 0.6 C_2	L_2 0.2 C_8	L_0 0.6 C_4	L_2 0.3 C_{25}
L_3 0 D_3	L_1 0.5 C_{19}	L_3 0 D_1	L_1 0.4 C_{11}	L_3 0.1 C_{32}	L_1 0.5 C_{18}
L_0 0.6 C_1	L_2 0.2 C_6	L_0 1.2 C_{23}	L_2 0.2 C_9	L_0 0.9 C_{21}	L_2 0.2 C_7
L_3 0.1 C_{31}	L_1 0.4 C_{14}	L_3 0 D_2	L_1 0.4 C_{15}	L_3 0.1 C_{33}	L_1 0.4 C_{13}
L_0 0.6 C_5	L_2 0.3 C_{28}	L_0 0.9 C_{22}	L_2 0.3 C_{27}	L_0 0.6 C_3	L_2 0.3 C_{24}
L_3 0.2 C_{10}	L_1 0.5 C_{20}	L_3 0.1 C_{30}	L_1 0.4 C_{12}	L_3 0.1 C_{29}	L_1 0.5 C_{17}

(f)

Cell-group $G_k = \{L_k\}$ of cell C_i (or D_i)
L_k
p_i
Power Dissipation of cell C_i (or D_i)
C_i (or D_i)
Cell Index

FIGURE 3. MSP of test case circuit modules with row = column = 6 and t = 2. (a) Placement-1 is a random bad solution. (b) Placement-2 is an optimal solution, (c) Placement-3 is a nonoptimal solution, (d) Placement-4 is a nonoptimal solution of Simple Approximation algorithm. (e) Placement-5 is an optimal solution of our proposed algorithm. (f) An element of placement matrix and its related notations. The Dotted red and blue envelopes in the placement matrices denote the regions with peak submatrix sum for t = 2 and t = 3 respectively.

[22] and [23] where every heat-generating point on the die is surrounded by a network of resistive heat flow paths which transport the dissipated heat away from the heat-generating points. It can be observed from the model that the resistive heat flow paths decrease as the heat-generating point approaches closer to the die boundary. For instance, in case of a heat-generating point on the die periphery the lateral heat flow paths facing the die boundary, get blocked. The



(d)

(e)

FIGURE 4. **Temperature map of the placements illustrated in Fig. 3. (a) Placement-1, (b) Placement-2, (c) Placement-3, (d) Placement-4, (e) Placement-5. The temperature is in Kelvin.**

TABLE 1. **Thermal attributes of the placements illustrated in Fig. 3 and Fig. 5.**

Solution	Max. Zonal Sum (t = 2) (W)	Max. Zonal Sum (t = 3) (W)	Avg. Temp. (K)	Peak Temp. (K)	Min. Temp. (K)	Temp. Grad. (K)	Std. Dev. Temp. (K)
Placement-1	3.6	6.2	346.73	363.41	337.53	25.88	5.6230
Placement-2	1.8	4.3	346.74	355.09	343.03	12.06	2.2944
Placement-3	1.9	4.7	346.74	349.40	343.4	6.00	1.3851
Placement-4	1.9	4.8	346.75	349.61	343.16	6.45	1.3888
Placement-5	1.9	4.6	346.74	349.10	345.16	3.94	0.7038
Placement-6	1.5	3.7	346.72	352.8	344.29	8.51	1.6445
Placement-7	1.8	3.7	346.74	348.02	345.16	2.86	0.6740

decrease in heat flow paths results in the accumulation of the dissipated heat thereby leading to a rise in temperature of the heat-generating point. Hence the die boundary behaves adiabatically to the heat flux generated inside the chip. The high power cells C_{21}, C_{22}, and C_{23} are the critical players in producing hotspots. In Placement-1, since all the high power cells are sitting together and very close to the adiabatic die wall, the submatrix sum and the peak on-chip temperature is very high. The uneven power distribution on account of the random cell placement resulted in the high gradient and high standard deviation in cell temperature. In placement-2 since the distribution of power is even, the submatrix sum is very less and hence the peak temperature, temperature gradient and the standard deviation in cell temperature are lesser than Placement-1. But in Placement-2, since the critical cell C_{23} is at the corner of the die periphery having its lateral heat flow paths facing the adjacent north and west die boundaries blocked, the peak temperature is much higher than Placement-3, 4, and 5. On the contrary, since the critical cell, C_{23} is relatively far away from the die boundary and also the power distribution is even, the peak temperature, temperature gradient and the standard deviation in cell temperature is better in Placement-3, 4, and 5. Moreover, in Placement-4 and 5, the number of hotspots is lesser than in Placement-3 since the other critical cells C_{21} and C_{22} are relatively far away from the die periphery. On account of the proposed UPS technique, Placement-5 gives an even better power distribution than Placement-3 and 4 (constructed by the algorithm A1 scheme) since UPS allows the lowest-in-order power cell within each submatrix region to share the maximum cell boundary of the highest power cell thereby facilitating improved diffusion of heat.

To further assess the effectiveness of the proposed placement technique, we have manually optimized the placements for peak submatrix sum and also for peak on-chip temperature. Placement-6 in Fig. 5a has been optimized manually for the minimum peak submatrix sum (1.5W) with t = 2, and it is the best placement according to the definition of MSP. However, from the thermal maps (Fig. 4e and Fig. 6a) and the experimental data in Table1, it has been observed that Placement-5 (solution of proposed algorithm TAMPO) despite having a comparatively higher peak submatrix sum of power, gives an improvement of 4.65% in peak temperature, 53.7% in temperature gradient and 57.2% in the standard deviation in cell temperature over Placement-6. The solution of the proposed algorithm TAMPO (Placement-5) is superior to the manually constructed solution (Placement-6) optimized for the peak submatrix sum of power. Hence again it has been observed that minimization of the peak submatrix sum (of power or heat) alone doesn't lead to a thermal-aware solution. Now, Placement-7 in Fig. 5b has been manually optimized for the minimum peak temperature. From the thermal maps (Fig. 4e and Fig. 6b) and the experimental data in Table1, it has been observed that Placement-7 gives an improvement of 1.42% in peak temperature, 27.41% in temperature gradient, and 4.23% in the standard deviation in cell temperature over Placement-5 (solution of proposed algorithm TAMPO). Hence the manually constructed temperature-optimized solution (Placement-7) manages to obtain a little improvement over the solution of the proposed algorithm TAMPO (Placement-5). However, manual optimization is time-consuming and possible for placements with a small number of cells whereas algorithm TAMPO is capable of handling placements with a large number of cells and also gives fast and efficient thermal-aware optimization.

Hence we find that the Matrix Synthesis Problem (MSP) is not adequate to ensure a thermal aware placement since it attributes the degree of hotness only to the peak (t x t) submatrix sum of heat. Therefore we have modified the concept of MSP by addressing the relative position of a sub-matrix (thermal zone) from the adiabatic die boundaries along with the submatrix sum of power for computing the thermal metric of placement and also by enhancing the heat diffusion through the Updated Placement Scheme (UPS).

IV. GATE ARRAY PACKER (GAP)

To emulate the situation of handling a gate array mapped circuit we have constructed a very basic gate array mapping algorithm called the Gate Array Packer (GAP) as shown in Fig. 7. The input to the GAP is the logic circuit composed of basic gates and flipflops and the output is the same circuit composed of a set of rectangular cells of equal number of transistors and identical dimensions and these cells act as input to the proposed thermal aware matrix placement algorithm.

A. ASSUMPTIONS OF GAP

A gate array is made of an array of transistors wherein the fundamental building block is a basic cell enveloping equal 'k' number of nMOS and pMOS transistors. The algorithm further considers 'Z' number of contagious basic cells to form a cluster such that every cluster has equal 'kZ' number of nMOS and pMOS and hence a total transistor count of '2kZ'. The algorithm packs the logic circuit with an objective to maximize the area utilization or minimize the cluster resource without generating any new inter-cluster connection other than the original nets. As a consequence every cluster accommodates only an integral number of functional blocks and the portion of the circuit which gets mapped to a particular cluster remains stationary in it for later stages of the placement process.

B. DEFINITION OF TERMS RELATED TO GAP

"Queue list": It is a set $\{B_g\}$ wherein an element at the g^{th} position is a functional block B_g which has higher or equal number of transistors T_g than its successor sitting at $(g+1)^{th}$ position. The queue list is obtained from the set $\{B_q\}$ of N_0 number of functional blocks of the logic circuit, by arranging the blocks B_q in descending order of total transistor count T_q.

"Queue count": It is a counter 'g' to denote the position of functional block to be selected next from the queue list $\{B_g\}$.

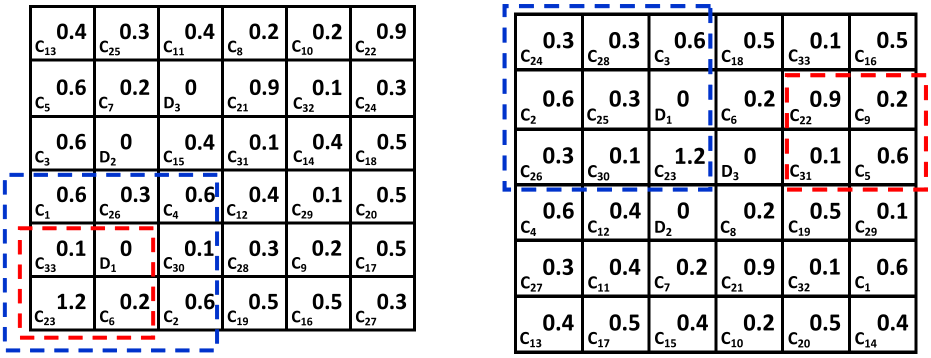
"Cluster count": It is a counter 'i' to indicate the next fresh cluster C_i to be selected for allocating the functional blocks.

"Utilized transistor count": It denotes two counters (e_i, h_i), where e_i is the number of nMOS and h_i the number of pMOS within a cluster C_i, which has already been utilized for mapping a functional block. Initially when no functional block has been mapped to a cluster, $e_i = h_i = 0$ and it is called a fresh cluster. If $0 < e_i < kZ$ or $0 < h_i < kZ$ or both, the cluster is partially utilized. If $e_i = kZ$ and $h_i = kZ$, the cluster is fully utilized.

C. PROCESS

The algorithm first constructs the Queue list and selects the functional blocks according to the serial number (i.e. Queue count) in the list and packs them into the clusters. The basic idea of GAP is to pack the "largest block first", so that the smaller blocks can, later on, be fitted within the leftover spaces of the partially utilized clusters. During the packing process, the algorithm first scans for the partially utilized clusters and maps a functional block to such a cluster only if the unutilized transistors in it are sufficient to accommodate the total transistors of the functional block. Otherwise, a fresh cluster is allocated to the functional block. The clusters finally utilized in realizing the complete logic circuit are termed as "functional clusters" and they play the role of the set of rectangular cells $\{C_i\}$ to be placed by the proposed thermal aware placement algorithm. The output of GAP also includes a set of inter-cluster nets connecting the functional clusters.

In our experimental work we have considered: (a) Every cluster to be composed of Z = 10 basic cells and every basic cell to be made up of equal k = 4 number of pMOS and nMOS transistors, (b) Cluster height $H_{cell} = 0.0002$ m and cluster width $W_{cell} = 0.0004$ m, (c) Every D-flipflop in the circuit is composed of 3 NOT gates and 2 nMOS transistors (as per the description in ISCAS89 circuits [29]). Further realizing the NOT gates in CMOS logic, a D-flipflop has been



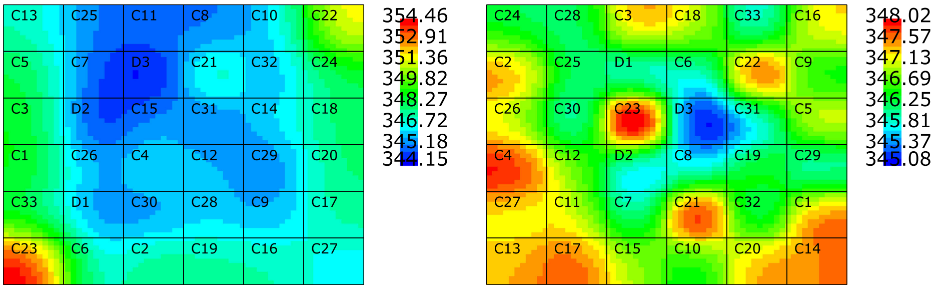
C_{13} 0.4	C_{25} 0.3	C_{11} 0.4	C_8 0.2	C_{10} 0.2	C_{22} 0.9
C_5 0.6	C_7 0.2	D_3 0	C_{21} 0.9	C_{32} 0.1	C_{24} 0.3
C_3 0.6	D_2 0	C_{15} 0.4	C_{31} 0.1	C_{14} 0.4	C_{18} 0.5
C_1 0.6	C_{26} 0.3	C_4 0.6	C_{12} 0.4	C_{29} 0.1	C_{20} 0.5
C_{33} 0.1	D_1 0	C_{30} 0.1	C_{28} 0.3	C_9 0.2	C_{17} 0.5
C_{23} 1.2	C_6 0.2	C_2 0.6	C_{19} 0.5	C_{16} 0.5	C_{27} 0.3

(a)

C_{24} 0.3	C_{28} 0.3	C_3 0.6	C_{18} 0.5	C_{33} 0.1	C_{16} 0.5
C_2 0.6	C_{25} 0.3	D_1 0	C_6 0.2	C_{22} 0.9	C_9 0.2
C_{26} 0.3	C_{30} 0.1	C_{23} 1.2	D_3 0	C_{31} 0.1	C_5 0.6
C_4 0.6	C_{12} 0.4	D_2 0	C_8 0.2	C_{19} 0.5	C_{29} 0.1
C_{27} 0.3	C_{11} 0.4	C_7 0.2	C_{21} 0.9	C_{32} 0.1	C_1 0.6
C_{13} 0.4	C_{17} 0.5	C_{15} 0.4	C_{10} 0.2	C_{20} 0.5	C_{14} 0.4

(b)

FIGURE 5. Manual synthesis of placement matrix for the test case circuit modules with row = column = 6 and t = 2. (a) Placement-6 is the best placement according to the MSP philosophy and obtained by manual design. (b) Placement-7 is an optimal solution for peak temperature obtained by manual design. The Dotted red and blue envelopes denote the regions with peak submatrix sum for t = 2 and t = 3 respectively.



(a)

(b)

FIGURE 6. Temperature map of the manual placements illustrated in Fig. 5. (a) Placement-6, (b) Placement-7. The temperature is in Kelvin.

finally realized with 5 nMOS and 3 pMOS transistors. The rest of the gates in the circuit have been realized in CMOS logic. An illustration of the mapping process has been shown in Fig. 8 where a test circuit containing 9 functional blocks is mapped to clusters C_1 and C_2 each composed of $Z = 4$ basic cells and each basic cell is composed of equal $k = 4$ number of pMOS and nMOS transistors. The functional blocks in the test circuit in Fig. 8a have been numbered according to the Queue list and mapped to the clusters in Fig. 8b according to the serial number 'SL' equals to the Queue count 'g'.

V. INITIAL MATRIX GENERATOR (IMaGe)

We propose a technique called the Initial Matrix Generator (IMaGe) for constructing improved initial placement solutions by modifying the placement scheme of the Simple Approximation algorithm [5]. The functional clusters C_i generated by the Gate Array Packer (GAP) are treated by IMaGe as the rectangular cells to be placed in the initial matrix placement. Hereafter the functional clusters have been termed as functional cells and the unused (dummy) clusters as dummy cells. The inputs to the algorithm IMaGe are the set of N_0 functional cells (clusters) $\{C_i\}$, set of power dissipation $\{p_i\}$ of the corresponding cells (clusters), height H_{cell} and width W_{cell} of each cell (cluster), maximum bound of aspect ratio 'r' of the die. The initial placement solution generated by IMaGe is further optimized by the Simulated Annealing (SA) engine within the proposed thermal aware placement algorithm to obtain the final solution.

A. ORDER OF PLACEMENT MATRIX

The number of rows and columns of the placement matrix are determined according to Step2 and Step3 of the algorithm shown in Fig. 9. The algorithm IMaGe generates "square matrix" as well as "minimum-cell matrix" initial placement solution as follows. A square matrix realizes the placement with the minimum equal number of rows and columns. In square matrix placement the aspect ratio of the die takes a default value as shown,

$$r = \frac{height\ of\ die}{width\ of\ die} = \frac{number\ of\ rows \times H_{cell}}{number\ of\ columns \times W_{cell}} = \frac{H_{cell}}{W_{cell}} \tag{1}$$

A minimum-cell matrix placement realizes the placement matrix with the minimum number of cells while maintaining an aspect ratio within the defined upper bound r and the lower bound 1/r. In this case the minimum integral values of

```
Algorithm GAP (Bench_ckt, k, Z);
(*Bench_ckt is the benchmark circuit under experiment*)
(* k is the number of nMOS and pMOS transistors in a basic cell *)
(* Z is the number of basic cells in a clusrter *)
Begin
Step1: File parsing of Bench_ckt to identify the functional block set {$B_q$} and interconnections;
       Assign appropriate nMOS count $n_q$ and pMOS count $p_q$ to every $B_q$;
       Compute total transistor count $T_q$ of every $B_q$: $T_q = n_q + p_q$;
Step2: Arrange set {$B_q$} in descending order of total transistor count $T_q$ to create a queue list {$B_g$};
Step3: /*Initialize*/
         Queue count g = 1 to select the first functional block from queue list, $B_g = B_1$;
         Cluster count i = 1 to select first cluster $C_i = C_1$;
         Utilized transistor count of $C_i$ cluster: nMOS count $e_i$ = pMOS count $h_i = 0$;
Step4: Select the unmapped functional block $B_g$ from the $g^{th}$ place in the queue list;
         /* For loop for scanning partially utilized clusters if any to map $B_g$ */
         For each cluster $C_f$ ( f = 1 to i) Do    /* Utilized transistor count of $C_f$ cluster: ($e_f$, $h_f$) */
             IF ($n_g \le (kZ - e_f)$ and $p_g \le (kZ - h_f)$)    /* nMOS & pMOS count of block $B_g$: ($n_g$, $p_g$) */
                 {Then assign functional block $B_g$ to $C_f$ cluster;
                  Update utilized transistor count of $C_f$ cluster: $e_f = e_f + n_g$ , $h_f = h_f + p_g$;
                  Break For loop ;}
         End For
         IF ($B_g$ is not assigned to a partially utilized cluster)
              {Then Update cluster count: i = i + 1 to select a fresh cluster;
               Assign functional block $B_g$ to the fresh $C_i$ cluster;
               Update utilized transistor count of $C_i$ cluster: $e_i = n_g$ , $h_i = p_g$;}
         IF (all functional blocks of Bench_ckt are mapped) {Then Goto End ;}
         Else {Update queue count: g = g + 1 to select the next functional block in queue list;
              GoTo Step 4 ;}
End
```

FIGURE 7. Description of Algorithm Gate Array Packer (GAP) for mapping logic circuit to gate array clusters.

rows and columns are evaluated on the basis of the following relationship.

$$\frac{1}{r} \le \frac{number\ of\ rows \times H_{cell}}{number\ of\ columns \times W_{cell}} \le r \tag{2}$$

For the case of minimum-cell matrix placement, in our work we have considered r = 2, thus allowing the aspect ratio of the die to vary between 0.5 and 2.

B. CELL GROUPING

A set of total cells $\Psi = \{E_i\}$ composed of the functional cells $\{C_i\}$ and dummy cells $\{D_i\}$ has been constructed and further arranged in the descending order of power dissipation as shown in Step5 and Step6 of Fig. 9. Further the cells have been divided among the four cell-groups $G_k = \{L_k\}$, $0 \le k \le 3$ as shown in Step7 of Fig. 9. An example of cell grouping has also been shown in Fig. 2 of section III. The variables Q_0, Q_1, Q_2, Q_3 shown in Fig. 9 denote the number of cells in the corresponding cell-groups G_0, G_1, G_2, G_3. The variables Q_k have been computed with the consideration that height of cell H_{cell} is lesser than or equal to its width W_{cell} and the L_1, L_2, L_3, L_4 cells being placed in every (t x t) submatrices in accordance with the Updated Placement Scheme (UPS) as illustrated in Fig. 3e of section III. Referring to the algorithm in Fig. 9, in case when the height of cell is greater than its width, the number of cells Q_2(of G_2 cell group) and Q_3 (of G_3 cell group) have to be exchanged and also the positions of L_2 and L_3 cells in every (t x t) submatrix have to be swapped. The proposed UPS technique for cell placement has been discussed as follows.

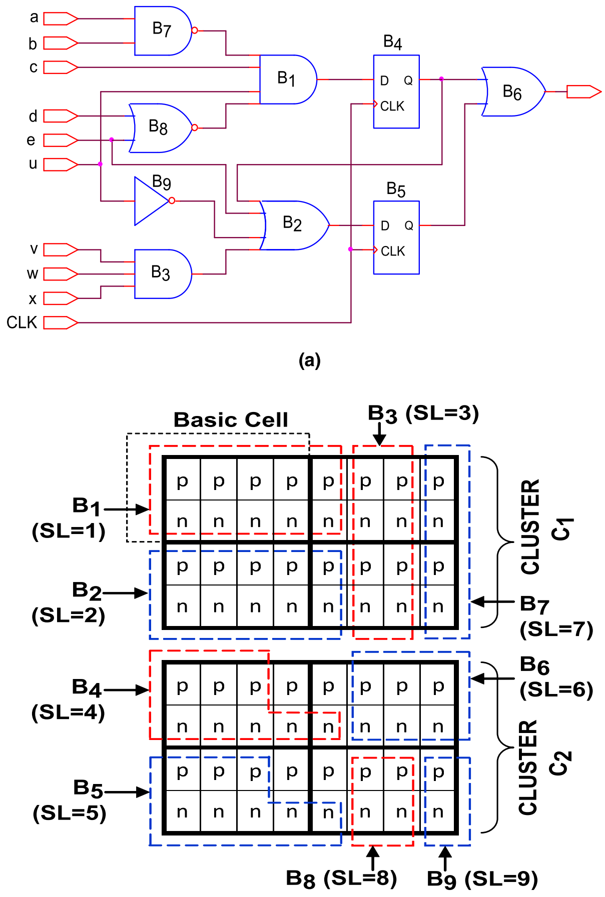
C. UPDATED PLACEMENT SCHEME (UPS)

For two adjoining cells in a chip floor sharing a boundary of length L, having power densities d_i and d_j respectively, the heat diffusion H between them according to [9] and [10] is given by,

$$H(d_i, d_j) = (d_i - d_j) \times L \tag{3}$$

Since in our work all the cells have equal area, the power densities d_i and d_j has been replaced by the power of cells p_i and p_j respectively and (3) has been modified as,

$$H(p_i, p_j) = (p_i - p_j) \times L \tag{4}$$



(a)

(b)

FIGURE 8. Illustration of circuit mapping to gate array. (a) Test case logic circuit. (b) Allocation of clusters and transistors to functional blocks of the test circuit. Allocated transistors for respective functional blocks have been represented by the dotted red and blue envelopes.

Similarly the total heat diffusion H_T of a cell with all its neighbors given in [10] can be modified as,

$$H_T(p) = \sum_j H(p, p_j) \tag{5}$$

The increase in the total heat diffusion H_T from a high power cell will help in lowering its temperature. Now consider the placement scheme of the Simple Approximation algorithm or algorithm A1 as shown in Fig. 3b, Fig. 3c and Fig. 3d. Here inside a (t x t) submatrix region the highest power cell L_0 shares its smaller edge (height) with L_1 (the second highest power cell), its larger edge (width) with L_2 (the third highest power cell) and no boundary with the lowest power cell L_3. Hence according to (4) and (5) there will be poor diffusion of heat from L_0 in this placement scheme. Hence we modify the placement scheme of algorithm A1 and propose the Updated Placement Scheme (UPS) where inside a (t x t) submatrix region, L_0 shares its larger edge with L_3, its smaller edge with L_2 and no boundary with L_1 as shown in Fig. 3e of section III. This placement scheme facilitates an improved diffusion of the heat from the critically high power cells belonging to the $G_0 = \{L_0\}$ cell group and also gives an improved balance in power distribution inside every (t x t) submatrix regions.

VI. PROPOSED THERMAL MODEL

We propose a thermal model that avoids the expensive computational budget for the exact temperature estimation and adopts a stochastic method for finding an assumption of the degree of hotness of a partial placement solution generated during the optimization process. The design assumptions held by the proposed thermal model are as follows.

A. CRITICAL THERMAL ZONE

The model identifies the cells (clusters) which have a power value greater than or equal to the aggregate of the average power and the standard deviation in power dissipation of the cells and labels them as critically hot cells. The critically hot cells belong to the cell group $G_0 = \{L_0\}$. The model then segregates an entire placement matrix in terms of thermal zones composed of the overlapping (t x t) submatrices of cells (clusters). The model then finds the (t x t) submatrices containing the critically hot cells and designates them as critical thermal zones responsible for the creation of hotspots.

B. ADIABATIC NATURE OF DIE WALL

The model perceives the die boundary wall to be adiabatic and reflective towards the heat generated within the chip. For computational simplicity, the model further assumes every heat-reflecting point on the die boundary wall to have zero power dissipation.

C. CENTER OF HEAT MAP

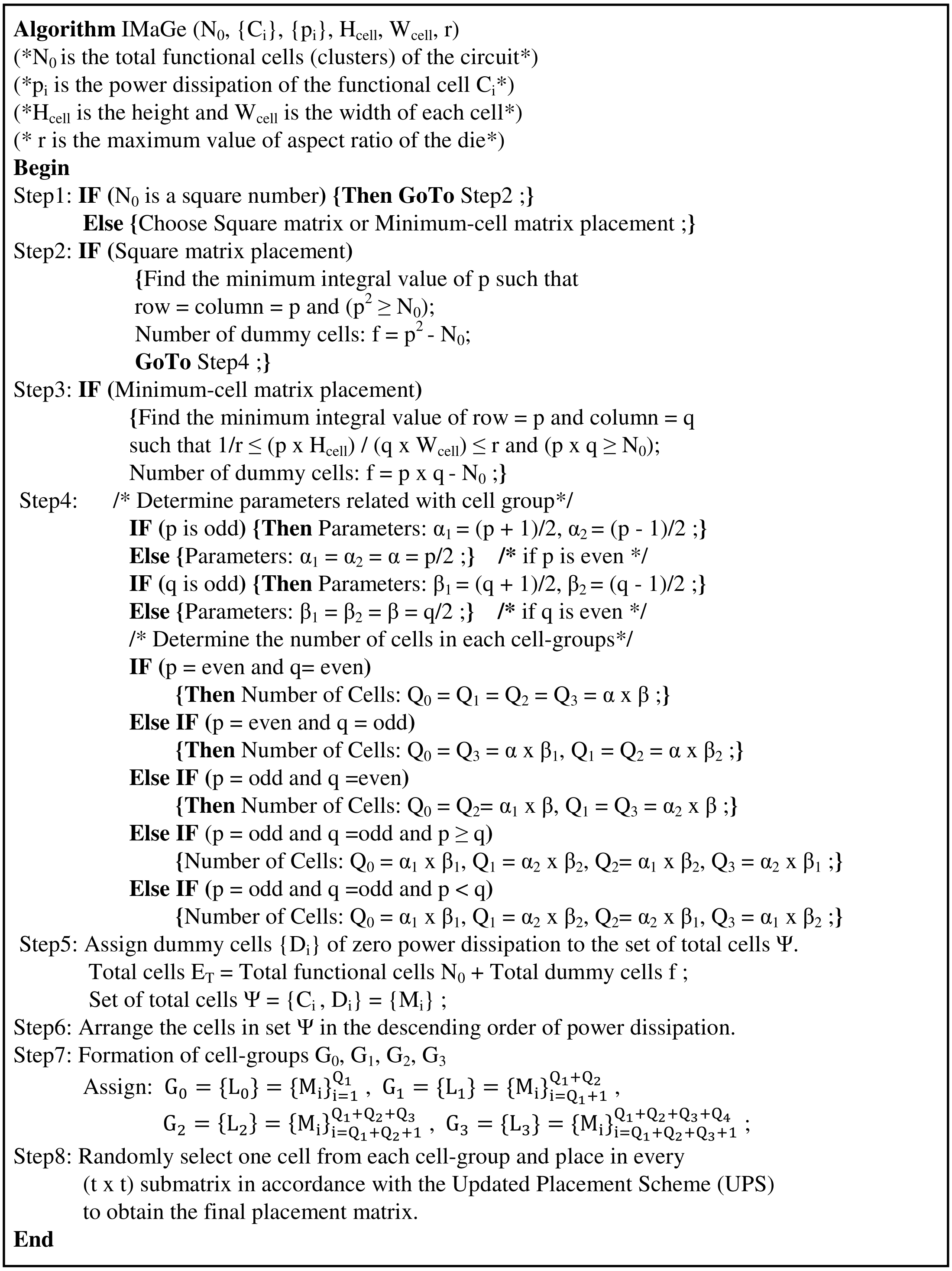
The model translates the placement matrix into a center of heat map composed of only the critical thermal zones with their centers of heat positioned at the respective center of critical cells. The thermal characterization of placement occurs based on the center of heat of the critical thermal zones. The center of heat map for the matrix placement in Fig. 3e has been shown in Fig. 10 below. The designated critically hot cells in the placement (Fig. 3e) are C_{21}, C_{22}, C_{23} and any thermal zone encompassing a critically hot cell is a critical thermal zone. In the center of heat map (Fig. 10), few critical thermal zones Z_1, Z_2 and Z_3 with $t = 2$ have been shown containing their centers of heat positioned at the geometrical centers of the respective critically hot cells. For every center of heat in the map, ξ denotes it summative power and (δ_x, δ_y) denote its nearest distance from the adiabatic die boundary.

D. HEAT COMPONENTS

The thermal model characterizes every critical thermal zone by affixing with it two thermal constituents, viz. the summative heat component and the reflective heat component described as follows.

1) SUMMATIVE HEAT COMPONENT

It accounts for the heating effect in a thermal zone due to the heat collectively generated by all the cells present within it. The model quantifies the summative heat component ξ_k for the k^{th} critical thermal zone as the aggregate power (like the



```
Algorithm IMaGe ($N_0$, {$C_i$}, {$p_i$}, $H_{cell}$, $W_{cell}$, r)
(*$N_0$ is the total functional cells (clusters) of the circuit*)
(*$p_i$ is the power dissipation of the functional cell $C_i$*)
(*$H_{cell}$ is the height and $W_{cell}$ is the width of each cell*)
(* r is the maximum value of aspect ratio of the die*)
Begin
Step1: IF ($N_0$ is a square number) {Then GoTo Step2 ;}
          Else {Choose Square matrix or Minimum-cell matrix placement ;}
Step2: IF (Square matrix placement)
                 {Find the minimum integral value of p such that
                 row = column = p and ($p^2 \geq N_0$);
                 Number of dummy cells: f = $p^2 - N_0$;
                 GoTo Step4 ;}
Step3: IF (Minimum-cell matrix placement)
                {Find the minimum integral value of row = p and column = q
                such that $1/r \leq (p \text{ x } H_{cell}) / (q \text{ x } W_{cell}) \leq r$ and ($p \text{ x } q \geq N_0$);
                Number of dummy cells: $f = p \text{ x } q - N_0$ ;}
 Step4:     /* Determine parameters related with cell group*/
                IF (p is odd) {Then Parameters: $\alpha_1 = (p + 1)/2$, $\alpha_2 = (p - 1)/2$ ;}
                Else {Parameters: $\alpha_1 = \alpha_2 = \alpha = p/2$ ;}   /* if p is even */
                IF (q is odd) {Then Parameters: $\beta_1 = (q + 1)/2$, $\beta_2 = (q - 1)/2$ ;}
                Else {Parameters: $\beta_1 = \beta_2 = \beta = q/2$ ;}   /* if q is even */
                /* Determine the number of cells in each cell-groups*/
                IF (p = even and q= even)
                     {Then Number of Cells: $Q_0 = Q_1 = Q_2 = Q_3 = \alpha \text{ x } \beta$ ;}
                Else IF (p = even and q = odd)
                     {Then Number of Cells: $Q_0 = Q_3 = \alpha \text{ x } \beta_1$, $Q_1 = Q_2 = \alpha \text{ x } \beta_2$ ;}
                Else IF (p = odd and q =even)
                     {Then Number of Cells: $Q_0 = Q_2 = \alpha_1 \text{ x } \beta$, $Q_1 = Q_3 = \alpha_2 \text{ x } \beta$ ;}
                Else IF (p = odd and q =odd and $p \geq q$)
                     {Number of Cells: $Q_0 = \alpha_1 \text{ x } \beta_1$, $Q_1 = \alpha_2 \text{ x } \beta_2$, $Q_2 = \alpha_1 \text{ x } \beta_2$, $Q_3 = \alpha_2 \text{ x } \beta_1$ ;}
                Else IF (p = odd and q =odd and $p < q$)
                     {Number of Cells: $Q_0 = \alpha_1 \text{ x } \beta_1$, $Q_1 = \alpha_2 \text{ x } \beta_2$, $Q_2 = \alpha_2 \text{ x } \beta_1$, $Q_3 = \alpha_1 \text{ x } \beta_2$ ;}
 Step5: Assign dummy cells {$D_i$} of zero power dissipation to the set of total cells $\Psi$.
          Total cells $E_T$ = Total functional cells $N_0$ + Total dummy cells f ;
          Set of total cells $\Psi = \{C_i , D_i\} = \{M_i\}$ ;
Step6: Arrange the cells in set $\Psi$ in the descending order of power dissipation.
Step7: Formation of cell-groups $G_0$, $G_1$, $G_2$, $G_3$
          Assign:  $G_0 = \{L_0\} = \{M_i\}_{i=1}^{Q_1}$ ,  $G_1 = \{L_1\} = \{M_i\}_{i=Q_1+1}^{Q_1+Q_2}$ ,
                   $G_2 = \{L_2\} = \{M_i\}_{i=Q_1+Q_2+1}^{Q_1+Q_2+Q_3}$ ,  $G_3 = \{L_3\} = \{M_i\}_{i=Q_1+Q_2+Q_3+1}^{Q_1+Q_2+Q_3+Q_4}$ ;
Step8: Randomly select one cell from each cell-group and place in every
          (t x t) submatrix in accordance with the Updated Placement Scheme (UPS)
          to obtain the final placement matrix.
End
```

FIGURE 9. Description of the Initial Matrrix Generator (IMaGe) algorithm.

aggregate heat in [5]) as follows.

$$\xi_k = \sum_{i=1}^{(t\times t)} p_i \tag{6}$$

Here p_i indicates the power dissipated by each cell C_i present in the (t x t) submatrix region. The model further attributes the aggregate power ξ_k as the power dissipation of the center of heat of the k^{th} critical thermal zone. With the increase in the summative heat component of a thermal zone, its degree of hotness also increases.

2) REFLECTIVE HEAT COMPONENT

It accounts for the influence on the heating effect in a critical thermal zone due to the proximity of the adiabatic die walls. The model assumes that the heat dissipated by a center of

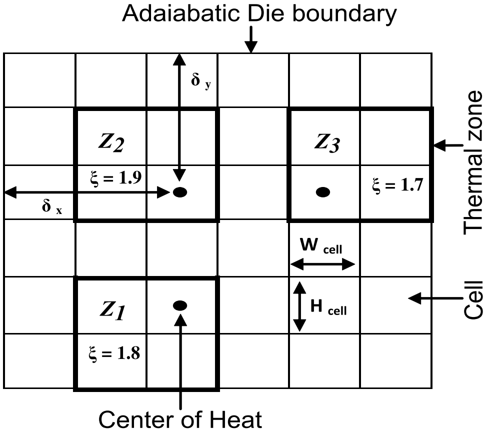


FIGURE 10. Illustration of the center of heat map of the matrix placement shown in Fig. 3e.

heat travels up to the nearest die walls along the orthogonal x and y directions and the die walls being adiabatic again reflect the incident heat back to the source. The reflected heat influx results in heat accumulation at the center of heat and subsequently increases its temperature. For simplicity, the model considers that rate of heat generation or power dissipation from a center of heat = rate of heat incidence on the die wall = rate of heat reflection from the die wall = rate of heat accumulation at the center of heat. Now, according to the Fourier heat flow equation the rate of heat flow Q between two points δ distance apart, normal to the cross-sectional area A, having thermal conductivity K and temperature difference $\Delta\theta$ is given by,

$$Q = KA(\Delta\theta/\delta) \tag{7}$$

According to [10], the temperature difference $\Delta\theta$ between two heat exchanging points is directly proportional to their power density difference Δd. Since the thermal zones have identical dimensions, the associated surface area on the die and cross sectional area A perpendicular to the die are constant. Since power density = power / surface area, and area being a constant parameter, the power density difference Δd between thermal zones can be substituted with their power difference Δp. As $\Delta\theta \propto \Delta$d and K is constant, it follows from (7) that,

$$\begin{aligned} Q &= constant \times A\,(\Delta d/\delta) \\ or, \quad Q &= constant \times (\Delta p/\delta), \end{aligned} \tag{8}$$

since Δd ∝ Δp and A is a constant. Now considering one of the heat exchanging points to be the center of heat (having power ξ_k) of the kth critical thermal zone and the other a heat reflecting point (having power $p_j = 0$) on the die boundary wall δ distance apart, the power difference $\Delta P = \xi_k - p_j = \xi_k - 0 = \xi_k$. Hence the rate of heat flow Q_k from the center of heat of the kth critical zone to the die wall according to (8) is given by,

$$Q = Q_k = constant \times (\xi_k/\delta) \tag{9}$$

The heat gets reflected from the die wall and accumulates at the center of heat at a rate of Q_k. The model computes the total reflective heat component $\hat{o}_k$ as the aggregate rate of heat accumulation at a center of heat along the x and y directions as,

$$\Phi_k = (Q_k)_{along\ x} + (Q_k)_{along\ y} \tag{10}$$

E. SATURATION THERMAL ZONE

The model defines an imaginary thermal zone (submatrix) having the maximum possible heat ξ_{sat} by considering all of its t^2 cells (clusters) to have the maximum power, max $\{p_i\}$. Hence the power dissipation of the center of heat of the saturation zone is given according to (6) by,

$$\xi_{sat} = (t \times t) \times max\{p_i\} \tag{11}$$

Similarly, the maximum reflective heat accumulation for the saturation thermal zone occurs when its center of heat is positioned at a minimum distance δ_{min} from the die walls and the associated maximum reflective heat accumulation rate Q_{sat} according to (9) is given by,

$$Q_{sat} = constant \times (\xi_{sat}/\delta_{min}) \tag{12}$$

Along x direction, $\delta_{min} = W_{cell}/2$ and along y direction, $\delta_{min} = H_{cell}/2$. Hence the maximum possible reflective heat component in the center of heat of the saturation thermal zone upon reflections along the x and y directions according to (10) and (12) is given by,

$$\begin{aligned} \Phi_{sat} = constant \times [\{\xi_{sat}/(W_{cell}/2)\} \\ + \{\xi_{sat}/(H_{cell}/2)\}] \end{aligned} \tag{13}$$

F. CRITICAL THERMAL METRIC

The model redefines the thermal metric μ_t described in [5] by characterizing every critical thermal zone of the placement matrix with the critical thermal metric function μ_{CTM} given by,

$$\mu_{CTM} = \lambda_1(\xi_k/\xi_{sat}) + \lambda_2(\Phi_k/\Phi_{sat}) \tag{14}$$

Parameters λ_1 and λ_2 are the weights specifying the relative importance of the normalized heat components in defining the thermal metric. In our experimental work we have obtained good results by configuring $\lambda_1 = 1$ and $\lambda_2 = 0.2$. The target of the proposed thermal aware placement algorithm is to minimize the critical thermal metric μ_{CTM} such that the peak on-chip temperature is minimized as well as the power dissipation and temperature is evenly distributed in the entire placement.

VII. PROPOSED THERMAL AWARE MATRIX PLACEMENT OPTIMIZER (TAMPO) ALGO RITHM

The proposed Thermal Aware Matrix Placement Optimizer (TAMPO) algorithm as described in Fig. 11 integrates the Initial Matrix Generator (IMaGe), the proposed thermal model, and the Simulated Annealing (SA) heuristics [24], [25] to optimize and finally obtain the thermal aware placement solution from the global optimum in the solution space. The important design aspects of the proposed algorithm TAMPO are as follows.

A. PROBLEM DEFINITION

Given a set of rectangular cells $\{C_i\}$ with an associated set of power dissipation $\{p_i\}$ and identical cell dimensions, the objective of the proposed algorithm is to distribute the cells in a matrix arrangement such that the temperature is evenly distributed and the peak on-chip temperature of the arrangement is minimized, subjected to the satisfaction of certain shape constraints. Shape constraints: For square matrix placement the aspect ratio of the chip equals to the aspect ratio of a rectangular cell C_i. For a minimum-cell matrix placement with a defined upper bound r the aspect ratio of the chip varies between 1/r and r.

B. PLACEMENT ENCODING

The algorithm TAMPO reads a geometrical placement as shown in Fig. 3e of section III containing $N_0 = 33$ functional cells and $f = 3$ dummy cells as an encoded string format as follows - $C_{10}C_{20}C_{30}C_{12}C_{29}C_{17}H$ $C_5C_{28}C_{22}C_{27}C_3C_{24}H$ $C_{31}C_{14}D_2C_{15}C_{33}C_{13}H$ $C_1C_6C_{23}C_9C_{21}C_7H$ $D_3C_{19}D_1C_{11}$ $C_{32}C_{18}H$ C_{16} $C_{26}C_2C_8C_4C_{25}$. Here term H denotes the operator to indicate the starting of a new row. The encoded expression is further simplified by replacing a cell C_i by index 'i' and a dummy cell D_j by index '$N_0 + j$' to form the encoded matrix expression as follows - 10 20 30 12 29 17 H 5 28 22 27 3 24 H 31 14 35 15 33 13 H 1 6 23 9 21 7 H 36 19 34 11 32 18 H 16 26 2 8 4 25.

C. PERTURBATION FUNCTION

It is the function Perturb () defined by algorithm TAMPO which randomly chooses any one of the following mechanisms to operate on an existing placement solution and generate a new partial placement solution. (a) Swapping cells between identical cell-groups G_k in two randomly selected (t x t) submatrices. (b) Swapping of two randomly selected rows containing cells belonging to the identical cell-groups G_k. (c) Swapping of two randomly selected columns containing cells belonging to the identical cell-groups G_k. The existing or the present solution is denoted by Present_matrix and the Perturb () generated new solution by New_matrix.

D. COST FUNCTION

The algorithm TAMPO defines a function Cost () to evaluate the fitness of a placement solution as follows.

$$Cost = \frac{Thermal\ metric\ of\ new\ placement}{Thermal\ metric\ of\ initial\ placement} \times 1000 \quad (15)$$

The thermal metric is the μ_{CTM} defined in (14). Fitness of a solution is the reciprocal of its cost and hence more the cost value, lesser is its fitness. The thermal metrics of the Simple Approximation based, the Hotspot tool based and the proposed TAMPO placement algorithms differ from each other in order of magnitude and dimension. The thermal metric has been normalized in (15) so that the cost metric bears the same order of value in all the three placement algorithms and the same simulated annealing engine may be implemented with identical optimization parameters. Moreover, since the normalized cost is a small fractional number, it has been up scaled 1000 times to help set the other simulation parameters effectively and obtain a better optimization. The difference in cost of the new solution New_matrix and the present solution Present_matrix is given by,

$$\Delta h = Cost\ (New\_matrix) - Cost(Present\_matrix) \quad (16)$$

If $\Delta h < 0$, the new solution is superior and if $\Delta h > 0$ the new solution is inferior than the present solution.

E. METROPOLIS ACCEPTANCE CRITERIA

This is a probability based criteria [24], [25] for accepting a new placement solution described as follows. (a) Incase $\Delta h < 0$ i.e. when the new solution is superior, the probability of acceptance is $P = 1$. (b) Incase $\Delta h > 0$ i.e. when the new solution is inferior, the probability of acceptance is,

$$P = exp(-\Delta h/T) \quad (17)$$

The inferior solution is accepted if the condition $\gamma < P$ is satisfied, where γ is a randomly generated number varying between 0 and 1. Also the corresponding annealing temperature T can be derived from (17) as,

$$T = -\Delta h/\ln P \quad (18)$$

F. SIMULATION PARAMETERS

The list of simulation parameters and their configured values related with the Simulated Annealing based proposed thermal aware placement algorithm TAMPO has been shown in Table 2.

G. TERMINATION CRITERIA OF OPTIMIZATION PROCESS

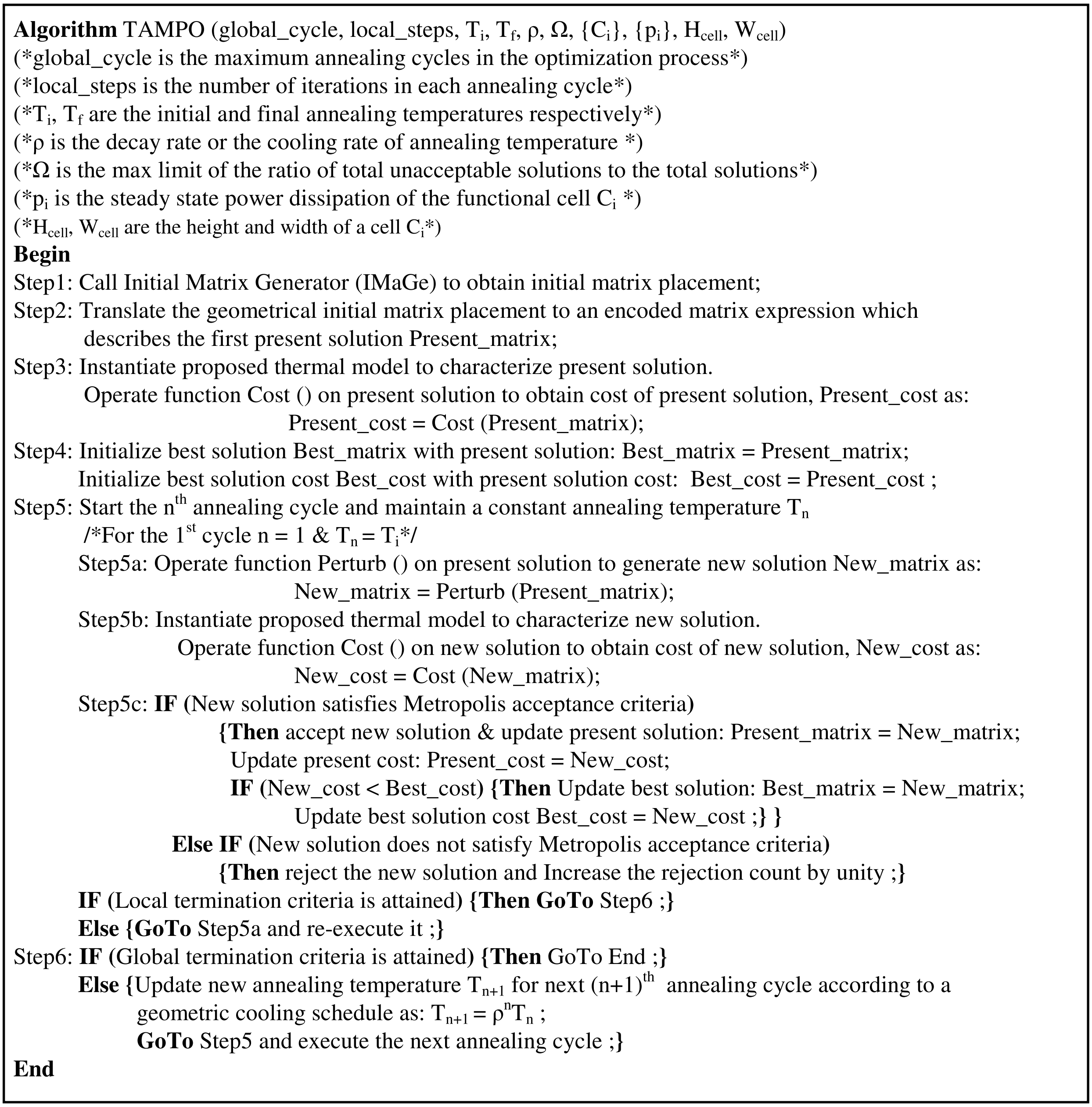
The overall optimization process consists of several annealing cycles wherein every annealing cycle again comprises of a number of iterations. For terminating the process, following criteria have been adopted from [25].

1) LOCAL TERMINATION CRITERIA

It is the criteria for terminating an annealing cycle running at a constant annealing temperature. For a placement having E_T number of total cells (including functional and dummy), an annealing cycle stops: (a) if the number of iterations in an annealing cycle exceeds the limiting value, local_steps $= 2E_T\Gamma$, or (b) if the number of inferior solutions accepted probabilistically exceeds the limiting value, 1/2 (local_steps) $= E_T\Gamma$, where Γ is an integral valued simulation parameter.

2) GLOBAL TERMINATION CRITERIA

It is the criteria for terminating the overall optimization process. The overall optimization process terminates: (a) if the total annealing cycles surpass the defined limiting value



```
Algorithm TAMPO (global_cycle, local_steps, $T_i$, $T_f$, $\rho$, $\Omega$, {$C_i$}, {$p_i$}, $H_{cell}$, $W_{cell}$)
(*global_cycle is the maximum annealing cycles in the optimization process*)
(*local_steps is the number of iterations in each annealing cycle*)
(*$T_i$, $T_f$ are the initial and final annealing temperatures respectively*)
(*$\rho$ is the decay rate or the cooling rate of annealing temperature *)
(*$\Omega$ is the max limit of the ratio of total unacceptable solutions to the total solutions*)
(*$p_i$ is the steady state power dissipation of the functional cell $C_i$ *)
(*$H_{cell}$, $W_{cell}$ are the height and width of a cell $C_i$*)
Begin
Step1: Call Initial Matrix Generator (IMaGe) to obtain initial matrix placement;
Step2: Translate the geometrical initial matrix placement to an encoded matrix expression which
         describes the first present solution Present_matrix;
Step3: Instantiate proposed thermal model to characterize present solution.
         Operate function Cost () on present solution to obtain cost of present solution, Present_cost as:
                              Present_cost = Cost (Present_matrix);
Step4: Initialize best solution Best_matrix with present solution: Best_matrix = Present_matrix;
        Initialize best solution cost Best_cost with present solution cost:  Best_cost = Present_cost ;
Step5: Start the $n^{th}$ annealing cycle and maintain a constant annealing temperature $T_n$
        /*For the $1^{st}$ cycle n = 1 & $T_n = T_i$*/
        Step5a: Operate function Perturb () on present solution to generate new solution New_matrix as:
                              New_matrix = Perturb (Present_matrix);
        Step5b: Instantiate proposed thermal model to characterize new solution.
                     Operate function Cost () on new solution to obtain cost of new solution, New_cost as:
                              New_cost = Cost (New_matrix);
        Step5c: IF (New solution satisfies Metropolis acceptance criteria)
                          {Then accept new solution & update present solution: Present_matrix = New_matrix;
                           Update present cost: Present_cost = New_cost;
                           IF (New_cost < Best_cost) {Then Update best solution: Best_matrix = New_matrix;
                                 Update best solution cost Best_cost = New_cost ;} }
                     Else IF (New solution does not satisfy Metropolis acceptance criteria)
                          {Then reject the new solution and Increase the rejection count by unity ;}
        IF (Local termination criteria is attained) {Then GoTo Step6 ;}
        Else {GoTo Step5a and re-execute it ;}
Step6: IF (Global termination criteria is attained) {Then GoTo End ;}
        Else {Update new annealing temperature $T_{n+1}$ for next $(n+1)^{th}$ annealing cycle according to a
              geometric cooling schedule as: $T_{n+1} = \rho^n T_n$ ;
              GoTo Step5 and execute the next annealing cycle ;}
End
```

FIGURE 11. Description of the proposed Thermal aware placement algorithm TAMPO based on Simulated Annealing.

global_cycle, or (b) if the ratio of the number of the rejected solutions (i.e. the rejection count) to the total number of solutions (accepted and rejected), exceeds the defined limiting value Ω, or (c) if the annealing temperature falls below the final minimum value T_f.

H. WIRELENGTH AND AREA

The set of inter-cluster nets associated with the circuit is generated by the Gate Array Packer (GAP) algorithm. The total inter-cluster wirelength associated with a placement solution of the circuit is determined according to the Half Perimeter Wirelength (HPWL) model given in [25] and [26]. Also the area of the placement is computed as,

$$Area\ of\ chip = No.of\ rows \times No.of\ columns \times Area\ of\ a\ cell \tag{19}$$

Experimental data in Table 3 shows the impact of the control parameters (P_i, P_f, ρ) variation on the performance of the SA based proposed algorithm TAMPO, denoted by the optimal cost and annealing cycles (global cycles) required for convergence. Here the results have been obtained for the s5378 circuit and it has been observed that combination COM-8 gives the best results. Other benchmark circuits show the similar trend as well. The convergence graph of

TABLE 2. Parameters of the simulated annealing based proposed placement algorithm TAMPO.

Parameter	Description	Value
global_cycle	Maximum number of annealing cycles in the optimization process	1000
Γ	Parameter related with the no. of iterations (local_steps) in each annealing cycle	5
Δh_avg	Average cost difference of an acceptable inferior solution	1
P_i	Initial probability of accepting an inferior solution having a cost difference of Δh_avg with the previous good solution.	0.99
P_f	Final probability of accepting an inferior solution having a cost difference of Δh_avg with the previous good solution.	0.01
T_i	Initial annealing temperature. As derived from (18), $T_i = -\Delta h\_avg/\ln(P_i)$	99.5
T_f	Final annealing temperature. As derived from (18), $T_f = -\Delta h\_avg/\ln(P_f)$	0.22
ρ	The decay rate or the cooling rate of annealing temperature.	0.99
Ω	Maximum limit of the ratio of total unacceptable solutions to the number of total solutions.	0.99

TABLE 3. Control parameter variation and their impact related with the proposed placement algorithm TAMPO experimented on the s5378 benchmark circuit in square matrix placement.

Combin -ation	Control parameters			T_i	T_f	Global Cycles for Conver -gence	Optimal Cost
	P_i	P_f	ρ				
COM-1	0.8	0.2	0.8	4.48	0.62	9	849.99
COM-2	0.8	0.2	0.9	4.48	0.62	19	842.61
COM-3	0.8	0.2	0.99	4.48	0.62	197	836.58
COM-4	0.8	0.2	1	4.48	0.62	999	856.69
COM-5	0.9	0.2	0.99	9.49	0.62	272	834.90
COM-6	0.99	0.2	0.99	99.5	0.62	506	833.98
COM-7	0.99	0.1	0.99	99.5	0.43	541	831.71
COM-8	0.99	0.01	0.99	99.5	0.22	610	830.65
COM-9	0.99	0.001	0.99	99.5	0.14	651	830.65

optimization with the proposed algorithm TAMPO for the s5378 benchmark circuit has been shown in Fig 12.

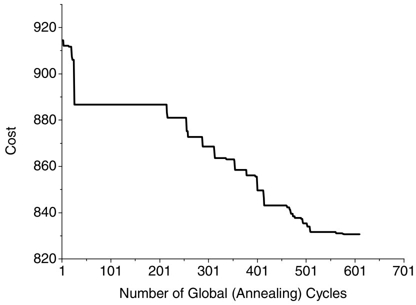


FIGURE 12. Convergence curve of the proposed placement algorithm TAMPO experimented on the s5378 benchmark circuit.

The cost function here is the normalized value of the critical thermal metric μ_{CTM} defined in (14) and (15). The convergence of the cost function has been shown with respect to iterations defined by the number of annealing cycles (global cycles). Here the initial cost is 1000 and after the first annealing cycle the cost is 914.535. The minimum cost achieved upon optimization is 830.65. Here the convergence occurs at the 578th and continues up to 610th annealing cycle.

I. CO-OPTIMIZATION OF THERMAL METRIC AND WIRELENGTH

The algorithm TAMPO has been further extended to simultaneously optimize the critical thermal metric μ_{CTM} in (14) and the half perimeter wirelength (HPWL) of placement solutions by redefining its cost function as follows.

$$\begin{aligned} Cost = W_1 \times \frac{\textit{Thermal metric of new placement}}{\textit{Thermal metric of initial placement}} \\ + W_2 \times \frac{\textit{Wire length of new placement}}{\textit{Wire length of initial placement}} \end{aligned} \tag{20}$$

Here W_1 and W_2 are the weights related with the relative refinement of the thermal metric μ_{CTM} and the wirelength respectively. The weights W_1 and W_2 are defined such that $0 \leq (W_1, W_2) \leq 1$ and $W_1 + W_2 = 1$. This extended form of the TAMPO which uses (20) as the objective function for optimization is the 'Co-optimized TAMPO'.

VIII. REFERENCE PLACEMENT ALGORITHMS

The dimension and power specification of the cells used and the temperature of the final optimized placement solutions are not defined in previous works [5], [11]–[13]. Hence we have constructed three more placement algorithms; (i) one based on the Hotspot tool [6], [7], [28], (ii) the second based on the Simple Approximation algorithm [5] and (iii) the third based on the thermal-aware placement algorithm [11] for the purpose of performance validation of the proposed algorithms TAMPO and Co-optimized TAMPO.

The Hotspot tool based placement algorithm generates a random initial square matrix placement and optimizes the solution using the Simulated Annealing (SA) heuristics similar to that implemented by TAMPO but with the following changes. (a) The perturbation function considers a random swap of cells or a random swap of matrix rows or a random swap of matrix columns. (b) The thermal metric is the peak temperature estimated by the Hotspot tool. (c) On account of the extremely high run time incurred due to exact temperature estimation, the number of iterations in an annealing cycle has been limited to 50 for all the circuits and the total annealing cycles have been limited to 100 for s5378, s9234, s13207, s15850 and to 10 for s38417 and s38584. The parameters of the Hotspot tool [28] and their configured values utilized in this paper have been specified in Table 4.

TABLE 4. Parameters of the Hotspot Tool.

IC Layers	Parameter	Description and unit	Value
Si Die	t_chip	Thickness (m)	0.000525
	k_chip	Thermal conductivity at 85°C (W/(m-K))	100
Cu Heat Spreader	t_spreader	Thickness (m)	0.001
	s_spreadrer	Side (m)	0.03
	k_spreader	Thermal conductivity (W/(m-K))	400
TIM	t_interface	Thickness (m)	0.00003
	k_interface	Thermal conductivity (W/(m-K))	1.33
Cu Heat Sink	t_sink	Thickness (m)	0.0065
	s_sink	Side (m)	0.06
	k_sink	Thermal conductivity (W/(m-K))	400
	c_convec	Convection capacitance (J/K)	140.4
	r_convec	Convection resistance (K/W)	0.7
	ambient	Ambient temperature (°C)	40

The Simple approximation based placement algorithm generates the initial square matrix solution according to algorithm A1 in [5] and optimizes the solution using the same Simulated Annealing (SA) heuristic with identical simulation parameters implemented by the TAMPO. The cost function here is the thermal metric defined as the maximum (t x t) submatrix sum of power (instead of heat) in the placement matrix i.e. μ_t as described by algorithm A1 in [5]. Since this algorithm is built on the same SA heuristics, parameter variations show a similar trend of performance impact as that of the TAMPO. Hence the same parameter combination COM-8 (mentioned in Table 3) has been maintained. The convergence curve of optimization process obtained from the Simple approximation based placement algorithm for the s5378 benchmark circuit has been shown in Fig 13. Here the cost function is the normalized value of the thermal metric μ_t according to (15). The convergence of the cost function has been shown with respect to iterations defined by the number of annealing cycles (global cycles). The initial cost here is 1000 and after the first annealing cycle, the cost is 982.652. The minimum cost achieved by optimization is 905.566. Here the convergence occurs at the 553rd and continues up to the 610th annealing cycle.

The thermal-aware placement algorithm [11] (refer Fig. 2 of [11]) which targets to minimize the hotspots, distributes the functional blocks or modules in square matrix arrangement, and optimizes the maximum aggregate of power dissipation or the critical threshold function in thermal regions denoted by the (t x t) submatrices. It initially generates a random placement and further improves it by two levels of iteration viz. outer loop and inner loop. Every outer loop identifies a local optimum through the different inner loop operations and finally, the global best is obtained from all such local optimum. Perturbations: (a) For every inner loop a new solution is generated by swapping the highest power cell of the maximum aggregate power (t x t) submatrix with the lowest power cell of the minimum aggregate power (t x t) submatrix. (b) Every outer loop starts with a new solution generated by interchanging any one of the cells swapped in the last iteration with any other randomly selected cell. Acceptance criteria: (a) A new solution is accepted if its cost does not exceed the cost of the local best solution plus the parameter max_ascent. (b) The local best solution and its cost are updated with a new solution attribute if the new solution cost is lesser than the local best. Termination criteria: (a) An inner loop terminates if the number of iterations exceeds the maximum limit defined by the by parameter inner_iter. (b) The whole process terminates if the number of outer iterations exceeds the maximum limit defined by the by parameter outer_iter. (c) The whole process may also terminate if the difference between the values of the maximum aggregate power and the minimum aggregate power submatrices of a solution fall below parameter $\in$.

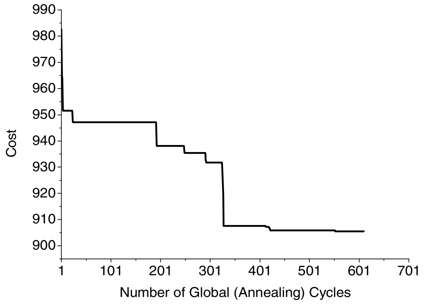


FIGURE 13. Convergence curve of the Simple Approximation based placement algorithm experimented on the s5378 benchmark circuit.

In the case of placement algorithm [11], we have experimented with different combinations of the control parameters ($\in$, max_ascent) to test the impact on the performance denoted by the optimal cost and the outer iterations (global cycles) required to converge. Also in order to ensure equal maximum iteration limits in the outer and inner cycles, similar to TAMPO, here we have configured outer_iter = 1000 (which is equal to global_cycle of TAMPO) and the inner_iter = $2E_T\Gamma$ (which is equal to the local_steps of TAMPO). Table 5 shows the results obtained for the s5378 benchmark circuit with placement algorithm [11] and it has been observed that combination COM-7 gives the best outcome. Similar trend has been observed for other benchmark circuits as well.

The convergence curve for the optimization process of the thermal-aware placement algorithm [11] has been shown in Fig.14. Here the cost is the critical threshold function defined as the maximum (t x t) submatrix sum of power dissipation in a matrix placement. The convergence of the cost function has been shown with respect to the number of outer iterations (global cycles). Here the initial cost is 0.642687 W and after the 1st outer iteration, the cost is 0.420266 W. The minimum cost achieved after the optimization is 0.346629 W. Here the

TABLE 5. Control parameter variation and their impact related with the thermal-aware placement algorithm [11] experimented on the s5378 benchmark circuit in square matrix placement.

Combination	Control parameters		Global Cycles for Convergence	Optimal Cost (W)
	∈	max_ ascent		
COM1	0.00026	0.0015	236	0.36713
COM2	0.00025	0.0015	332	0.35975
COM3	0.00024	0.0015	999	0.35082
COM4	0.00023	0.0015	999	0.35082
COM5	0.00022	0.0015	999	0.35082
COM6	0.00024	0.002	999	0.34997
COM7	0.00024	0.0025	999	0.34663
COM8	0.00024	0.003	999	0.35198
COM9	0.00024	0.0035	999	0.35694

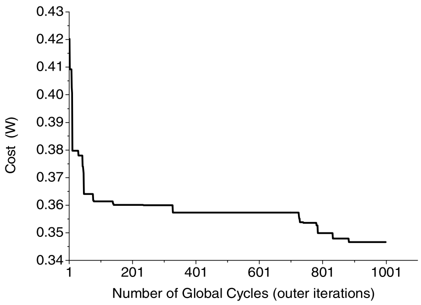


FIGURE 14. Convergence curve of the thermal-aware placement algorithm [11] experimented on the s5378 benchmark circuit.

convergence occurs at the 883rd and continues up to the 999th outer iteration.

IX. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed algorithms TAMPO and the Co-optimized TAMPO have been experimented on the ISCAS89 [29] benchmark circuits listed in Table 6. The reference placement algorithms based on (i) Hotspot tool [28], (ii) Simple Approximation algorithm [5], and (iii) thermal-aware placement algorithm [11], have also been implemented on the benchmark circuits. A power model is required to estimate the power of the clusters (rectangular cells) generated by the Gate Array Packer (GAP) algorithm. However, the heat generated by cells in the placement matrix has been generated randomly in [5]. We have also generated the power of the rectangular cells (clusters) according to [27] varying randomly between 4.06 x10^6 W/m^2 to 0.22 x 10^6 W/m^2 for 90 nm processor to test the effectiveness of the placement algorithms. The placement algorithms have been designed in C language and experiments have been conducted in a Linux system running on a 3GHz Intel Core i5 processor. Hotspot tool has been used to find the exact temperature of the final optimized solutions with the configuration mentioned in Table 4 of section VIII. Table 6 shows the average power dissipation and other particulars of the benchmark circuits in original and post gate array map. Moreover, the functional clusters obtained in Table 6 are the functional cells mentioned in Table 7 and Table 10. Table 7 shows the attributes of the benchmark circuits, common to the square matrix solutions given in Table 8, Table 9 and also in Table 11.

TABLE 6. Attributes of the benchmark circuits before and after the gate array mapping operation by algorithm GAP.

Circuit Name	Original Circuit Attributes		Post Gate Array Map Attributes		
	No. of Functional Blocks	No. of Nets	No. of Functional Clusters	No. of Inter - Cluster Nets	Avg. Power (W)
s5378	2958	2910	134	2163	8.5547
s9234	5808	5784	261	5739	16.6766
s13207	8589	8445	400	8415	25.4342
s15850	10306	10188	467	10147	29.6238
s38417	23815	23710	1115	18090	67.5406
s38584	20679	20376	1201	20353	72.8923

TABLE 7. Common attributes of square matrix placement solutions given in Table 8, Table 9 and Table 11.

Circuit	No. of Functional Cells	No. of Dummy Cells	Matrix Size	Area (10^{-6} m^2)
s5378	134	10	12 x 12	12.0
s9234	261	28	17 x 17	23.0
s13207	400	0	20 x20	32.0
s15850	467	17	22 x 22	39.0
s38417	1115	41	34 x 34	92.0
s38584	1201	24	35 x 35	98.0

TABLE 8. Particulars of the optimized square matrix solutions obtained from the Hotspot tool based placement algorithm.

Circuit	HPWL (m)	Avg. Temp. (K)	Peak Temp. (K)	Temp. Grad. (K)	Std. Dev. Temp. (K)	Run Time (s)
s5378	5.74	341.65	343.26	3.18	0.7340	554
s9234	21.86	347.50	349.62	4.5	0.7809	3351
s13207	36.77	356.47	358.23	3.44	0.7517	8923
s15850	48.48	358.80	361.07	4.74	0.9658	18387
s38417	140.20	386.41	389.50	7.28	1.3040	24531
s38584	166.11	390.81	393.62	6.03	1.0247	31158

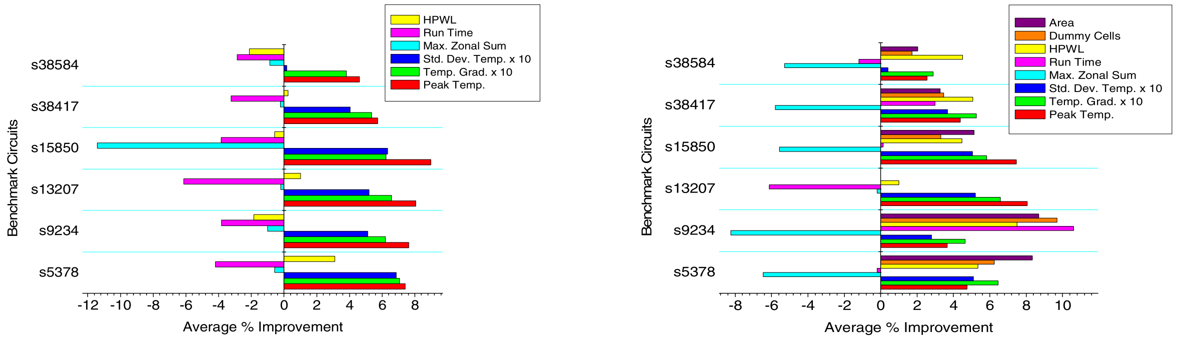
Table 8 presents the attributes of the optimized square matrix placements achieved by the Hotspot tool based placement algorithm. Similar to [5], in our work also the Simple Approximation algorithm has been implemented for generating square matrix placements with t = 2. Table 9 demonstrates the particulars of optimized square matrix solutions achieved by (a) the Simple Approximation based, (b) thermal-aware placement algorithm [11] based, and (c) the proposed TAMPO placement algorithms with t = 2. Table 10 depicts the attributes of the optimized minimum-cell matrix solutions attained by the proposed placement algorithm TAMPO with t = 2. The half perimeter wirelength (HPWL) reported here relates to the inter-cluster nets generated by the algorithm GAP.

TABLE 9. Particulars of the optimized square matrix placement solutions obtained from the Simple Approximation based placement algorithm, placement based on thermal-aware algorithm [11], and the proposed TAMPO algorithm.

	Circuit	HPWL (m)	Max. Zonal Sum (W)	Avg. Temp. (K)	Peak Temp. (K)	Temp. Grad. (K)	Std. Dev. Temp. (K)	Run Time (s)
Simple Approximation on [5] based Placement	s5378	5.80	0.386135	341.64	348.81	10.28	1.9176	9.76
	s9234	21.75	0.382013	347.47	355.96	11.13	1.8134	29.25
	s13207	36.91	0.413788	356.48	366.26	12.79	1.9016	53.73
	s15850	48.76	0.393065	358.82	370.77	15.3	2.5321	74.21
	s38417	140.46	0.392271	386.40	396.91	14.03	1.9985	365.96
	s38584	162.94	0.39188	390.79	400.03	11.33	1.5563	404.16
Thermal Aware Placement for Gate Arrays[11]	s5378	5.85	0.346629	341.63	347.63	8.23	1.6899	7.55
	s9234	21.62	0.380775	347.47	355.22	10.47	1.7563	21.64
	s13207	36.70	0.412343	356.45	364.99	11.15	2.0672	39.63
	s15850	48.79	0.420067	358.82	368.95	13.12	2.1623	55.55
	s38417	139.80	0.414948	386.40	396.48	12.76	1.3680	271.95
	s38584	164.86	0.417467	390.80	399.32	10.64	1.1490	306.50
Proposed TAMPO based placement	s5378	5.62	0.388356	341.65	343.21	3.02	0.6005	10.17
	s9234	22.15	0.385854	347.50	349.65	4.23	0.8852	30.37
	s13207	36.54	0.414638	356.49	358.76	4.38	0.9118	57.02
	s15850	49.04	0.437948	358.81	362.00	5.75	0.9257	77.06
	s38417	140.11	0.393126	386.43	389.81	6.52	1.1911	377.79
	s38584	166.38	0.395270	390.84	394.19	7.02	1.5307	415.74

TABLE 10. Particulars of the optimized minimum-cell matrix placement solutions obtained from the proposed TAMPO algorithm.

Circuit	No. of Functional Cells	No. of Dummy Cells	Matrix Size	Area (10^{-6} m^2)	HPWL (m)	Max. Zonal Sum (W)	Avg. Temp. (K)	Peak Temp. (K)	Temp. Grad. (K)	Std. Dev. Temp. (K)	Run Time (s)
s5378	134	1	15 x 9	11.0	5.49	0.411075	343.10	345.22	3.65	0.9397	9.78
s9234	261	0	29 x 9	21.0	20.12	0.413459	349.79	352.93	5.97	1.3058	26.15
s13207	400	0	20 x 20	32.0	36.54	0.414638	356.49	358.76	4.38	0.9118	57.02
s15850	467	1	26 x 18	37.0	46.58	0.414898	359.62	363.49	6.4	1.2529	74.11
s38417	1115	1	36 x 31	89.0	133.34	0.415011	387.26	391.48	6.66	1.2632	355.02
s38584	1201	3	43 x 28	96.0	155.59	0.412601	391.30	396.79	8.07	1.4947	409.01

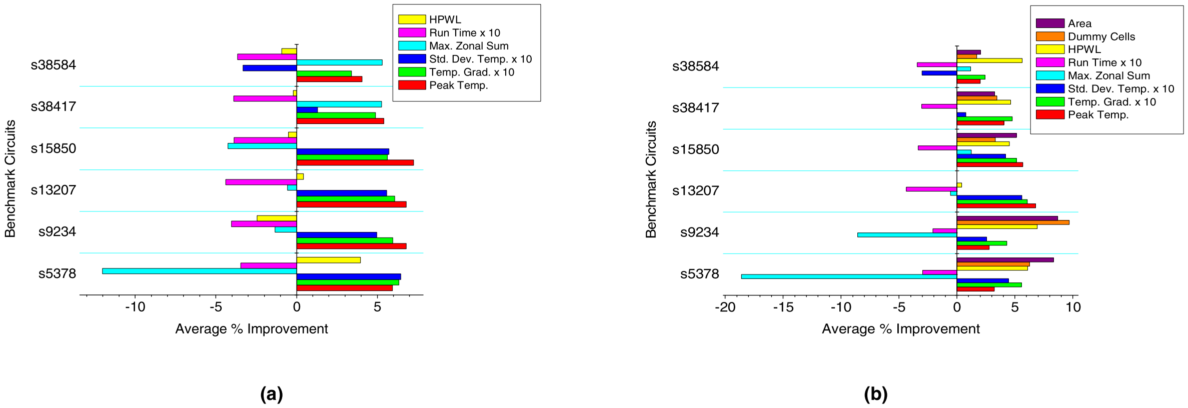


(a)

(b)

FIGURE 15. Average percentage improvement obtained from (a) the square matrix solutions of TAMPO over the square matrix solutions of Simple Approximation based placement algorithm, (b) the minimum-cell matrix solutions of TAMPO over the square matrix solutions of Simple Approximation based placement algorithm. The temperature is in °C.

In this work the thermal quality of placement has been assessed on the basis of the peak temperature ‘Peak Temp.’, temperature gradient ‘Temp. Grad.’, and the standard deviation in cell temperature ‘Std. Dev. Temp.’. The peak (t x t) submatrix sum has been denoted by ‘Max. Zonal Sum’. From Table 8 and Table 9 it can be observed that the Hotspot tool based placement algorithm gives improved thermal results than the placement algorithms based on Simple Approximation and [11]. However, both the placement algorithms based on Simple Approximation and [11] give an enormous 99% (approximate) improvement in execution time over the Hotspot tool based algorithm, thereby



(a)

(b)

FIGURE 16. Average percentage improvement obtained from (a) the square matrix solutions of TAMPO over the square matrix solutions of Thermal-aware placement algorithm [11], (b) the minimum-cell matrix solutions of TAMPO over the square matrix solutions of thermal-aware placement algorithm [11]. The temperature is in °C.

TABLE 11. Attributes of the optimized square matrix and minimum-cell matrix placement solutions obtained from the proposed Co-optimized TAMPO algorithm with equal weightage ($W_1 = 0.5$, $W_2 = 0.5$) for thermal metric (μ_{CTM}) and the wirelength (HPWL) refinement.

Placement Category		Thermal and Wire Length Co-Optimization						
	Circuit	HPWL (m)	Max. Zonal Sum (W)	Avg. Temp. (K)	Peak Temp. (K)	Temp. Grad. (K)	Std. Dev. Temp. (K)	Run Time (s)
Square Matrix Placement	s5378	3.0606	0.397622	341.6504	344.57	5.44	1.512841	158.54
	s9234	15.1736	0.392739	347.4554	352.73	7.69	2.042465	782.8
	s13207	26.379	0.41561	356.4677	361.9	7.84	1.868138	1552.21
	s15850	35.5178	0.397504	358.8	363.3	6.99	1.783361	2240
	s38417	65.0514	0.396593	386.4041	391.67	8.26	1.739143	10315
	s38584	133.3132	0.406463	390.7973	396.67	8.77	1.398223	13402
Minimum Cell Matrix Placement	s5378	2.8788	0.42145	343.081	346.56	5.48	1.074244	124.12
	s9234	15.163	0.416874	349.7352	354.44	7.69	2.00475	717.78
	s13207	26.379	0.41561	356.4677	361.9	7.84	1.868138	1552.21
	s15850	34.1864	0.41719	359.5992	365.08	8.26	1.755333	2205
	s38417	63.1462	0.419323	387.2406	392.6	7.97	1.523262	9985
	s38584	127.0992	0.451989	391.2674	398.85	10.09	1.734525	12650

stressing the importance of adopting alternative algorithms like the Simple Approximation and [11] for designing placement algorithms. From Table 8 and Table 9 it can also be observed that the proposed algorithm TAMPO gives almost equivalent thermally good solutions as the Hotspot tool based algorithm while (TAMPO) giving an enormous 99% improvement in the execution time over the Hotspot tool based placement algorithm. From Table 8 and Table 9 it can be observed that the average temperature (Avg. Temp.) of the square matrix solutions of the corresponding circuits obtained from the placement algorithms is almost the same.

From the results in Table 9, an analysis has been done in Fig. 15a between the square matrix solutions of both the Simple Approximation based placement algorithm and the TAMPO. In this case, the placement algorithm based on Simple Approximation gives an average improvement of 2.38% in peak submatrix sum over TAMPO. However, TAMPO gives an average improvement of 7.07% peak temperature, 58.73% temperature gradient, 46.23% standard deviation in cell temperature compared to the former. TAMPO incurs an average overhead of 4.02% in execution time and a marginal 0.028% average overhead in wirelength (HPWL) with respect to the Simple Approximation based placement algorithm. Again utilizing the results in Table 9 and Table 10 an analysis has been made in Fig. 15b between the square matrix solutions of the Simple Approximation based placement algorithm and the minimum-cell matrix solutions of the TAMPO. In this case, the placement algorithm based on Simple Approximation gives an improvement of 5.26% in the peak submatrix sum. But TAMPO gives an average improvement of 5.14% peak temperature, 52.68 % temperature gradient, and 37.05% standard deviation in cell temperature with respect to the former. TAMPO also gives an average improvement of 4% dummy cells, 4.58% area, 4.65% half perimeter wirelength (HPWL), and 1.03% execution time over the Simple Approximation based placement algorithm. However, the solutions generated by TAMPO have a slightly higher average temperature compared to the former counterpart.

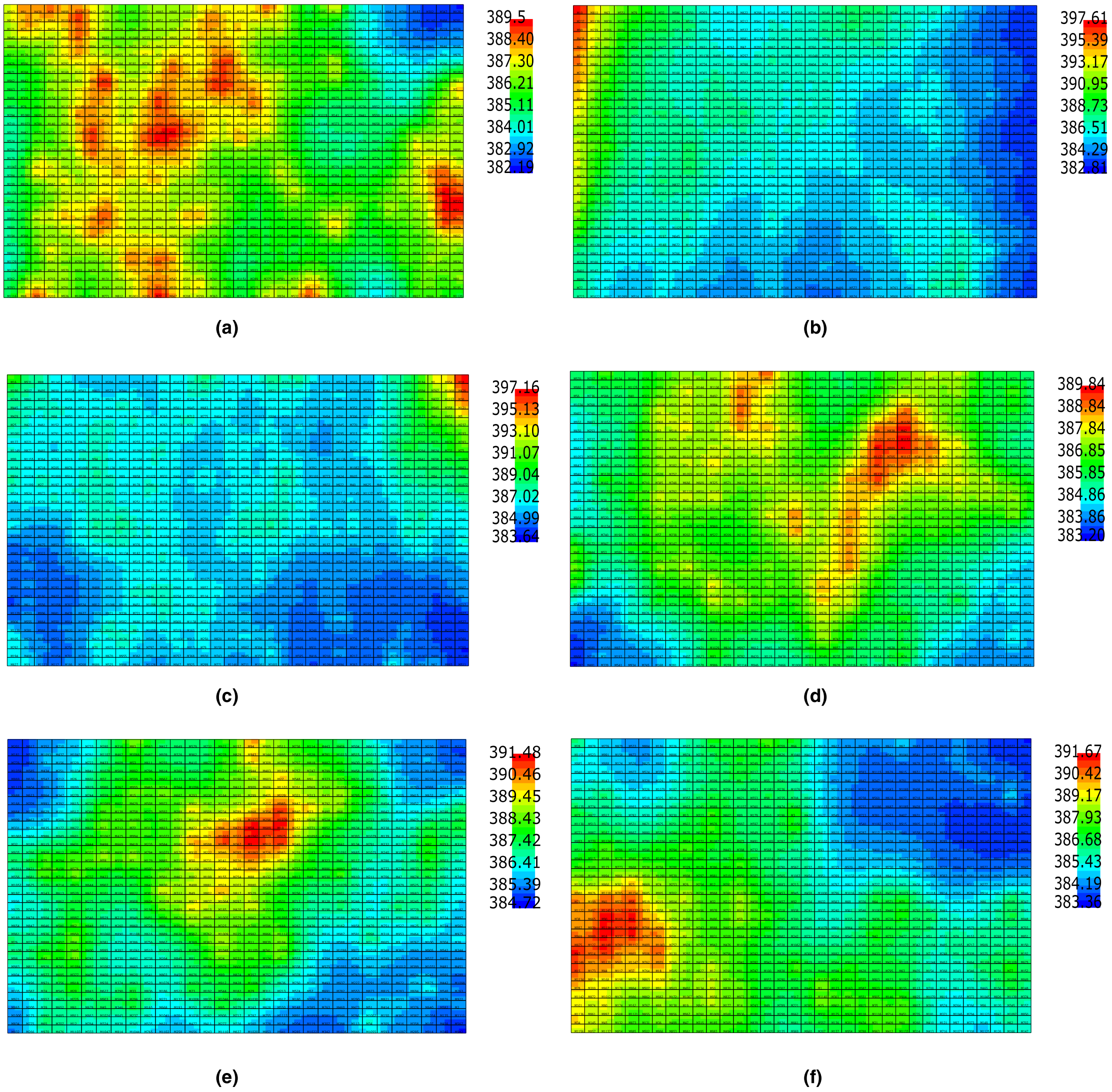


FIGURE 17. **Thermal map of the optimized placement solutions of s38417 circuit obtained as, (a) Square matrix solution of Hotspot tool based placement algorithm, (b) Square matrix solution of Simple Approximation based placement algorithm, (c) Square matrix solution of thermal awrae placement algorithm [11], (d) Square matrix solution of proposed TAMPO placement algorithm, (e) Minimum-cell matrix solution of proposed TAMPO placement algorithm. (f) Square matrix solution of proposed Co-optimized TAMPO placement algorithm. The temperature is in Kelvin.**

Now from the results in Table 9, an analysis has been done in Fig. 16a between the square matrix solutions of both the thermal-aware placement algorithm [11] and the TAMPO. In this case, the thermal-aware placement algorithm [11] gives an average improvement of 1.27% in the peak submatrix sum or the critical threshold over the TAMPO. On the contrary, TAMPO gives an average improvement of 6.04% peak temperature, 53.79% temperature gradient, 34.47% standard deviation in cell temperature compared to the former. TAMPO gives a marginal 0.05% average improvement in half perimeter wirelength (HPWL) over the thermal-aware placement algorithm [11]. However, TAMPO incurs an average overhead of 38.86% in execution time with respect to the former counterpart. Again from the results in Table 9 and Table 10, an analysis has been made in Fig. 16b between the square matrix solutions of the thermal-aware placement algorithm [11] and the minimum-cell matrix solutions of the proposed algorithm TAMPO. In this case, the thermal-aware placement algorithm [11] gives an average improvement of 4.23% in the peak submatrix sum or the critical threshold over the TAMPO. Whereas, TAMPO gives an average improvement of 4.09% peak temperature, 47.09% temperature gradient, 24.26% standard deviation in cell temperature compared to the former. TAMPO also gives an average improvement of 4% dummy cells and 4.58% area, 4.71% half perimeter wirelength (HPWL) over the thermal-aware placement algorithm [11]. However, TAMPO incurs an average overhead of 32.1% in execution time with respect to the former counterpart.

Table 11 shows the results of the solutions obtained from the proposed Co-optimized TAMPO (discussed in section VII H) where equal weightage ($W_1 = 0.5$, $W_2 = 0.5$) has been given for the optimization of the thermal metric and wirelength (HPWL). A comparative analysis has been done between the results of the TAMPO (thermal aware only) and Co-optimized TAMPO placement algorithms given in Table 9 and Table 11 respectively, relating to the square matrix as well as the minimum-cell matrix solutions. In the case of square matrix solutions, Co-optimized TAMPO gives an average improvement of 34.31% wire length (HPWL) over TAMPO. However, Co-optimized TAMPO incurs an average overhead of 2.46% peak temperature, 52.35% temperature gradient, and 86.26% standard deviation in cell temperature with respect to TAMPO. In this case, the execution time of TAMPO is on average 95.97% faster than Co-optimized TAMPO. In the case of the minimum-cell matrix solutions, Co-optimized TAMPO gives an average improvement of 32.93% wire length (HPWL) over TAMPO. But again Co-optimized TAMPO costs an average overhead of 1.97% peak temperature, 38.62% temperature gradient, and 41.58% standard deviation in cell temperature in comparison to TAMPO. In this case, the execution time of TAMPO is on average 95.78% faster than Co-optimized TAMPO. The excess runtime overhead in Co-optimized TAMPO is due to the additional wirelength (HPWL) estimation task of partial placement in every iteration step of the optimization process. Since TAMPO only tries to optimize the thermal metric, it gives thermally superior solutions in comparison to Co-optimized TAMPO where 50% of weightage has been given for the thermal improvement. However, the Co-optimized TAMPO gives a substantial improvement in the wirelength and still gives improved thermally aware solutions in comparison to the placement algorithms based on Simple Approximation [5] and the thermal-aware placement algorithm [11].

The experimentally generated thermal profile showing the temperature distribution of the different optimized matrix placement solutions of the s38417 circuit, synthesized with the reference placement algorithms as well as the proposed placement algorithms have been shown in Fig. 17. Hence the placement algorithms based on [5] and [11] give more improvement in the peak submatrix sum of power or the critical threshold value over the proposed algorithm TAMPO; however, TAMPO gives better thermal aware solutions. Also, the proposed algorithm Co-optimized TAMPO gives solutions that are improved in peak temperature, temperature gradient, and wirelength with respect to the placement algorithms based on [5] and [11].

X. CONCLUSION

In this paper, the proposed Thermal Aware Matrix Placement Optimizer (TAMPO) has proved to be an efficient framework for generating improved thermal aware matrix placements of gate arrays. Experimental results suggest that TAMPO has successfully improved the philosophy of the Matrix Synthesis Problem (MSP) to generate solutions that are thermally superior in terms of peak temperature, temperature gradient, and the standard deviation in cell temperature with respect to the existing methodologies of Simple Approximation and [11]. Work done in this paper quantifies the thermal improvements in terms of temperature which the previous works on thermal aware matrix placement lack. Since the framework avoids the expensive overhead of exact temperature estimation during the placement synthesis, it is fast and hence maintaining almost the same quality of solutions, it gives significant run time improvement over the Hotspot tool based placement algorithm. Experimental results also show that the extended version of TAMPO, termed as the Co-optimized TAMPO gives efficient thermal aware placement of cells while maintaining a reasonable reduction in the wirelength as well. As a future scope, the work can be extended for incorporating the metrics like routing congestion and interconnect delay along with the thermal metrics of a chip in a multi-objective optimization problem. It can also be extended for the placement problem of 3D ICs.

REFERENCES

[1] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The impact of technology scaling on lifetime reliability," in *Proc. Int. Conf. Dependable Syst. Netw.*, Florence, Italy, 2004, pp. 177–186.

[2] A. Vassighi and M. Sachdev, "Thermal runaway in integrated circuits," *IEEE Trans. Device Mater. Rel.*, vol. 6, no. 2, pp. 300–305, Jun. 2006.

[3] S. Chaudhury, "A tutorial and survey on thermal-aware VLSI design: Tools and techniques," *Int. J. Recent Trends Eng.*, vol. 2, no. 8, pp. 18–21, 2009.

[4] N. Lamaison, J. B. Marcinichen, and J. R. Thome, "Recent advances in on-chip cooling systems: Experimental evaluation and dynamic modeling," in *Proc. 15th Int. Heat Transf. Conf.*, Kyoto, Japan, 2014, pp. 100–134.

[5] C. C. N. Chu and D. F. Wong, "A matrix synthesis approach to thermal placement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 11, pp. 1166–1174, Nov. 1998.

[6] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[7] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, Jan. 2004.

[8] S. R. Choudhury and S. N. Pradhan, "DOTFloor—A diffusion oriented time-improved floorplanner for macrocells," *IEEE Access*, vol. 7, pp. 172074–172087, 2019.

[9] Y. Han, I. Koren, and C. A. Moritz, "Temperature aware floorplanning," in *Proc. 2nd Workshop Temp.-Aware Comput. Syst. (TACS)*, 2005, pp. 1–10.

[10] Y. Han and I. Koren, "Simulated annealing based temperature aware floorplanning," *J. Low Power Electron.*, vol. 3, no. 2, pp. 141–155, Aug. 2007.

[11] P. Ghosal, T. Samanta, H. Rahaman, and P. Dasgupta, "Thermal-aware placement of standard cells and gate arrays: Studies and observations," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Montpellier, France, 2008, pp. 369–374.

[12] R. B. Roy, D. Saha, and P. Dasgupta, "A novel approach for solving multi-objective optimization problems: A case of VLSI thermal placement," IIM Calcutta Work. Paper Ser., Calcutta, India, WPS no. 661, Aug. 2010, doi: 10.13140/2.1.2487.3283.

[13] A. Sinharay, P. Roy, and H. Rahaman, "VLSI thermal placement issues: A cooperative game theory based approach," in *Proc. 6th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Patna, India, Dec. 2016, pp. 106–111.

[14] P. Ghosal, H. Rahaman, and P. Dasgupta, "Thermal aware placement in 3D ICs," in *Proc. Int. Conf. Adv. Recent Technol. Commun. Comput.*, Kottayam, India, Oct. 2010, pp. 66–70.

[15] G. Feng, F. Ge, N. Wu, L. Zhou, and J. Liu, "MSP based thermal-aware mapping approach for 3D network-on-chip under performance constraints," *IEICE Electron. Exp.*, vol. 13, no. 7, 2016, Art. no. 20160082.

[16] J. Song, Y.-M. Lee, and C.-T. Ho, "ThermPL: Thermal-aware placement based on thermal contribution and locality," in *Proc. Int. Symp. VLSI Design, Automat. Test (VLSI-DAT)*, Hsinchu, Taiwan, Apr. 2016, pp. 1–4.

[17] S. H. Gade, P. Kumar, and S. Deb, "A pre-RTL floorplanner tool for automated CMP design space exploration with thermal awareness," in *Proc. 20th Int. Symp. VLSI Design Test (VDAT)*, Guwahati, India, May 2016, pp. 1–6.

[18] A. T. Winther, W. Liu, A. Nannarelli, and S. Vrudhula, "Thermal aware floorplanning incorporating temperature dependent wire delay estimation," *Microprocessors Microsystems*, vol. 39, no. 8, pp. 807–815, Nov. 2015.

[19] P. Sivaranjani and A. S. Kumar, "Thermal-aware non-slicing VLSI floorplanning using a smart decision-making PSO-GA based hybrid algorithm," *Circuits, Syst., Signal Process.*, vol. 34, no. 11, pp. 3521–3542, Nov. 2015.

[20] Z. Jiang and N. Xu, "HotSpot thermal floorplan solver using conjugate gradient to speed up," *Mobile Inf. Syst.*, vol. 2018, Apr. 2018, Art. no. 2921451.

[21] J.-M. Lin, T.-T. Chen, Y.-F. Chang, W.-Y. Chang, Y.-T. Shyu, Y.-J. Chang, and J.-M. Lu, "A fast thermal-aware fixed-outline floorplanning methodology based on analytical models," in *Proc. Int. Conf. Comput.-Aided Design*, San Diego, CA, USA, Nov. 2018, pp. 1–8.

[22] W. Huang, M. R. Stan, and K. Skadron, "Parameterized physical compact thermal modeling," *IEEE Trans. Compon. Packag. Technol.*, vol. 28, no. 4, pp. 615–622, Dec. 2005.

[23] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," in *Proc. DAC*, San Diego, CA, USA, 2004, pp. 878–883.

[24] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science), vol. 272. Cham, Switzerland: Springer, 2019, ch. 1, pp. 1–35.

[25] S. M. Sait and H. Youssef, *VLSI Physical Design Automation: Theory and Practice*. Singapore: World Scientific, 2001.

[26] B. N. B. Ray, S. K. Mohanty, D. Sethy, and R. B. Ray, "HPWL formulation for analytical placement using Gaussian error function," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Bhubaneswar, India, Dec. 2017, pp. 56–61.

[27] G. M. Link and N. Vijaykrishnan, "Thermal trends in emerging technologies," in *Proc. ISQED*, San Jose, CA, USA, 2006, pp. 625–632.

[28] *HotSpot Tool Package*. Accessed: May 15, 2017. [Online]. Available: http://lava.cs.virginia.edu/hotspot

[29] *ISCAS89 Circuits*. Accessed: Jul. 7, 2018. [Online]. Available: https://s2.smu.edu/ dhoungninou/Benchmarks/ISCAS89/VERILOG

SOMNATH ROY CHOUDHURY received the M.Tech. degree in VLSI design from the Vellore Institute of Technology, India, in 2009. He is currently pursuing the Ph.D. degree with the Department of Electronics and Communication Engineering, National Institute of Technology Agartala, India. He is also an Assistant Professor with the Department of Electronics and Communication Engineering, National Institute of Technology Agartala. His research interests include VLSI physical design automation, thermal management in processors, and CAD tool development.

SAMBHU NATH PRADHAN received the M.E. degree from Bengal Engineering and Science University, India, in 2004, and the Ph.D. degree from the Indian Institute of Technology Kharagpur, India, in 2010. He is currently an Associate Professor with the Department of Electronics and Communication Engineering, National Institute of Technology Agartala, India. His research interests include VLSI physical design automation, low-power design and testing, and thermal aware logic synthesis.