B* tree based Thermal Aware Non Sliceable Floorplanning

PRABHA. J
PG Student, Dept. of ECE
Thiagarajar College of Engineering
Madurai, India
jprabha@tce.edu

D.GRACIA NIRMALA RANI
Assistant Professor, Dept. of ECE
Thiagarajar College of Engineering
Madurai, India
gracia@tce.edu

S.RAJARAM
Associate Professor, Dept. of ECE
Thiagarajar College of Engineering
Madurai, India
rajaram_siva@tce.edu

Abstract- **Scaling have increased power density which further increases the maximum temperature of the chip. This maximum temperature can be decreased thus reducing the impact of temperature on Hotspots using Thermal Aware Floorplanning. This paper involves temperature aware design which alleviates the problems in sub-micron technologies. This paper makes use of the simulated annealing algorithm to determine the global optimum solution. The floorplaning makes use of B* tree representation which is optimum for non-sliceable floorplan. The ultimate aim of this paper is to minimize the maximum temperature of the chip.**

I. INTRODUCTION

Power density has increased in recent years due to scaling thus creating difficulty in maintaining the reliability of the system [1]. Most of the power aware design does not reduce power design in hotspots [2]. Thus, Thermal aware floorplanning can be used to reduce the maximum temperature of the chip [3].

The temperature can be reduced using lateral Spreading of heat in which the hot blocks are placed beside the colder blocks [4, 5]. Hotspot tool developed by Skadron et al. can be used to calculate the temperature distribution among various blocks of the chip.

The Parquet tool designed by Han et al. uses the objective function adding temperature as one of the factor. This paper uses the combination of Normalized Polish Expression for floorplan representation and Genetic Algorithm for Optimization. The performance impact of the proposed algorithm is not addressed in this paper [7]. The temperature aware floorplanning by Lixia et al. makes use of B* tree representation and Simulated Annealing. In our proposed work, the fixed outline thermal aware floorplan is obtained.

The paper is organized as follows: Section II gives the problem definition. Section III explains the B* tree representation and the Simulated Annealing algorithm. Section IV discusses the proposed Thermal Aware Floorplanning Algorithm. Section V ends with results and discussion.

II. PROBLEM FORMULATION

A. Cost Function

Let $B = \{b_1, b_2, \ldots, b_m\}$ be a set of m rectangular modules with block $b_i$of width W_i, height H_i, area A_i, and an original power density P_i, $1 < i < m$. The aspect ratio is given by H_i/W_i. Each module is free to rotate. Let (x_i, y_i) denote the co-ordinates of the bottom-left corner of the rectangle b_i on a chip. A floorplan F is an assignment of (x_i, y_i) to the bottom left corners for each b_i such that no two modules overlap.

The objective function is given by

$$obj = ar + \alpha \frac{Area}{Area_{norm}} + \beta \frac{Wire}{Wire_{norm}} + \gamma \frac{Temp}{Temp_{norm}} \quad (1)$$

Where α,β and γ are the weights of the area, wirelength and the highest temperature respectively.

III. PROPOSED APPROACH

A. Motivation

Thermal-aware Floorplanning problem is considered a generalization of the quadratic assignment which is classified as an NP-hard problem [9]. Therefore, it is difficult to find thermally optimized solutions, especially when the number of functional units increases. The lack of proposals capable of dealing with an elevated number of functional units in a short time motivates this work. The proposed Thermal aware floorplanning method provides a multi-objective solution.

B. B* Tree Representation

Chang et al. [14] presented a binary tree based representation for a left and bottom compacted placement called B*-Tree.

IJERT

Given a placement P, we can construct a unique B*-Tree in linear time by using a recursive procedure similar to the depth first search (DFS) algorithm. The root of a B*-Tree corresponds to the module on the bottom-left corner. The left child n_j of a node n_i denotes the module m_j that is the lowest adjacent module on the right-hand side of m_i as $x_j=x_i+w_i$.

The right child n_k of a node n_i denotes the module m_k that is the lowest visible module above mi and with the same x co-ordinate as m_i as $x_k=x_i$.

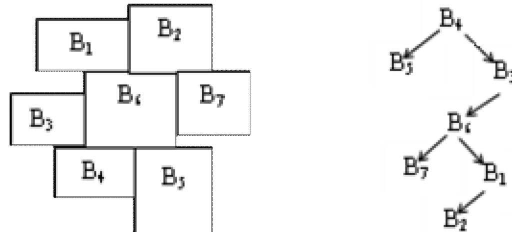

Fig.1. *B* tree representation for non-sliceable floorplan*

C. Module Cooling Strategy

The principle used in the proposed thermal aware floorplanning is lateral spreading of heat. When a hot module is placed besides a cold module, then diffusion of heat takes place which reduces the overall temperature [6]. Fig. 2 depicts the module cooling strategy. The hot modules designated as H_1 and H_2 are separated in thermal aware floorplanning.

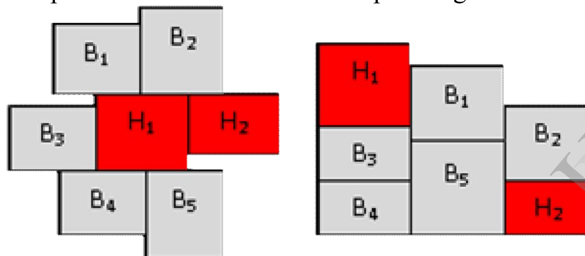

Fig.2. *Module cooling strategy*

The temperature of the chip depends on the power consumption of each functional block and the relative positions of the functional blocks. This will result in more spreading of heat, thus reducing hotspot temperatures.

D. Thermal Modelling Tool

Skadron et al. developed the HotSpot software tool, which calculates the temperature distribution among different blocks in a CPU chip.

Hotspot generates the thermal RC circuit dynamically when initialized with a CPU floorplan.HotSpot uses the fourth-order Runge-Kutta method (rk4) to solve the differential equations that describe the RC circuit, and returns the maximum temperature of blocks at each step.

The basic idea is that, we define the transfer thermal resistance R_{ij} of functional block i with respect to block j as the temperature rise at block i due to one unit of power dissipated at block j:

$$R_{ij}=\Delta T_{ij}/P_j \qquad (2)$$

such that we can get a transfer thermal resistance matrix as R^t**.** For any power distribution on the floorplan, we can calculate each block's temperature by using the following equation:

$$R^t = \begin{bmatrix} R^t_{11} & R^t_{12} & \dots & R^t_{1m} \\ R^t_{21} & R^t_{22} & \dots & R^t_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ R^t_{m1} & R^t_{m2} & \dots & R^t_{mm} \end{bmatrix} \leftrightarrow \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{bmatrix} = \begin{bmatrix} R^t_{11} & R^t_{12} & \dots & R^t_{1m} \\ R^t_{21} & R^t_{22} & \dots & R^t_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ R^t_{m1} & R^t_{m2} & \dots & R^t_{mm} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} \qquad (3)$$

Where P_i is the power consumption and T_i is the temperature of the functional block i. The transfer thermal resistance matrix can be obtained from Hotspot, given the floorplan for a set of blocks.

IV. PROPOSED THERMAL AWARE FLOORPLANNING

Thermal aware floorplanning makes use of the simulated Annealing Algorithm. The algorithm starts by randomly choosing an initial B*-Tree. The perturbation process converts one feasible B*-Tree to another. We then do the placement for the corresponding B*-Tree and evaluate the cost function. The move is accepted if the cost of the current solution is less than the previous one or with a probability that is a decreasing function of annealing temperature (Boltzmann functions) as defined in the algorithm. For all other cases, the move is rejected.

We terminate the annealing process if the number of accepted moves is less than 5% of all moves made at a certain temperature or the temperature is low enough i.e. less than the threshold. At last, we transform the resulting B*-Tree, i.e. the solution with the lowest value of cost function, to the corresponding final admissible placement.

The method used for thermal aware floorplanning in this paper is given in the flowchart as shown in fig. 3.

IJERT

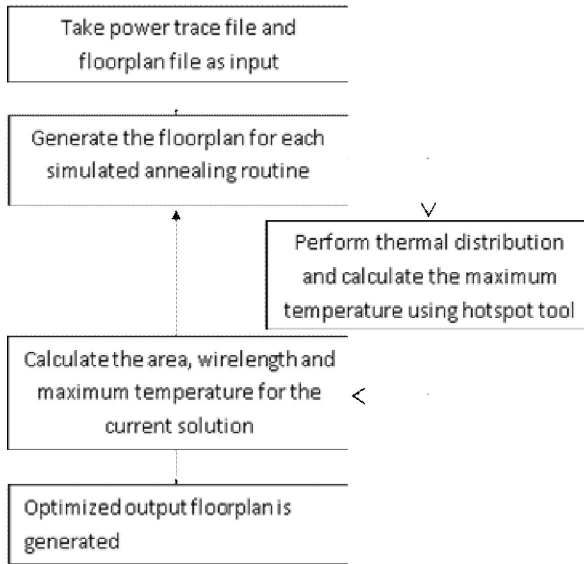

Fig.3. *Flowchart of the proposed Thermal aware floorplanning*

IV. RESULTS AND DISCUSSION

Skadron et al. developed the HotSpot software tool, which calculates the temperature distribution among different blocks in a CPU chip. The algorithm was written in C++ and was simulated with Intel core i5 processor and 4 GB RAM. The MCNC benchmark circuits namely apte, hp, Xerox, ami33, ami49 were used to test the correctness of the tool. The MCNC benchmark circuit hp that is flipped diagonally is shown in fig. 4(a). The blocks labeled as Unit 8 and Unit9 is taken as the hotspots. These blocks have the temperature greater than the average die temperature. The MCNC benchmark circuit characteristics are shown in table 1.

IJERT

Table I. MCNC benchmark circuit Characteristics

Circuit	Block#	Net#	Pin#	Pad#
Apte	9	97	214	73
Xerox	10	203	696	2
Hp	11	83	264	45
ami33	33	123	480	42
ami49	49	408	931	22

The input block/net/pl files and the randomly generated experimentally tested ptrace file were taken as input. The thermal aware floorplanner was run first with area weight as 0.35, wirelength weight as 0.30 and the temperature weight as 0.35.

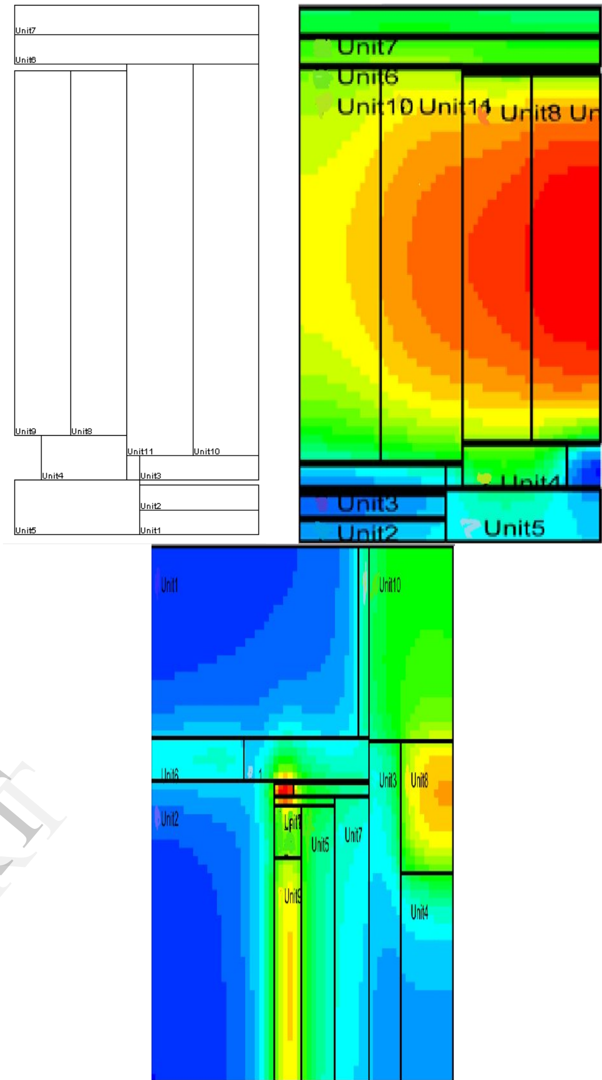

Fig.4. (a) MCNC benchmark circuit hp (b) Thermal distribution of the hp circuit based on the power density (c) Output Thermal aware floorplan using Simulated annealing algorithm for the hp circuit.

The maximum temperature of the floorplan was reduced significantly with only slight increase in area and wirelength. The hotspots are placed apart so that the peak temperature is reduced. This is done in the thermal aware floorplanning using simulated annealing algorithm. Fig. 4(c) depicts the output floorplan that is thermal aware with their thermal distribution. This shows that the peak temperature is reduced resulting in the elimination of hotspots.
The MCNC benchmark circuits ami33 and ami49 shows reduction in the peak temperature of the chip whereas the apte circuit shows a slight increase in temperature. The results have

also shown that the proposed algorithm reduces the chip's maximum and average temperature. The reduction in temperature is shown in Fig. 6.

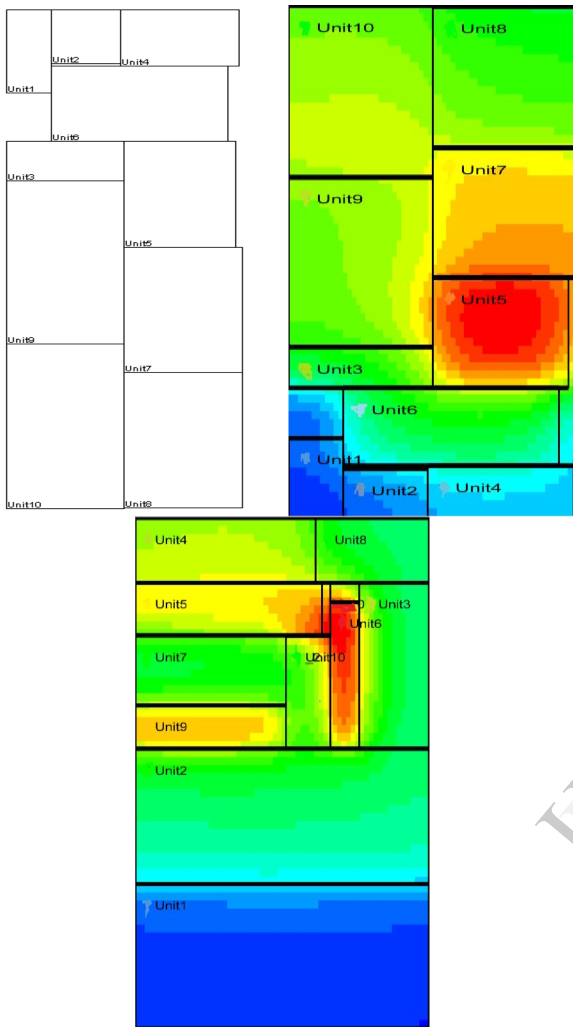

Fig. 5 (a) MCNC benchmark circuit xerox (b) Thermal distribution of the xerox circuit based on the power density (c) Output Thermal aware floorplan using Simulated annealing algorithm for the xerox circuit**.**

Table II. Comparison between existing approach and proposed approach

Circuit	Existing Approach [7]			Proposed Approach (Randomly Selecting modules)		
	Area (in mm^2)	Wire length (in mm)	Temperature (in °C)	Area (in mm^2)	Wire length (in mm)	Temperature (in °C)
apte	48.66	417	78	48.36	465	79.11
hp	9.78	162	89	9.363	167	88.85
ami33	1.27	68	87	1.156	70	58.85
ami49	38.86	898	95	35.54	907	87.85

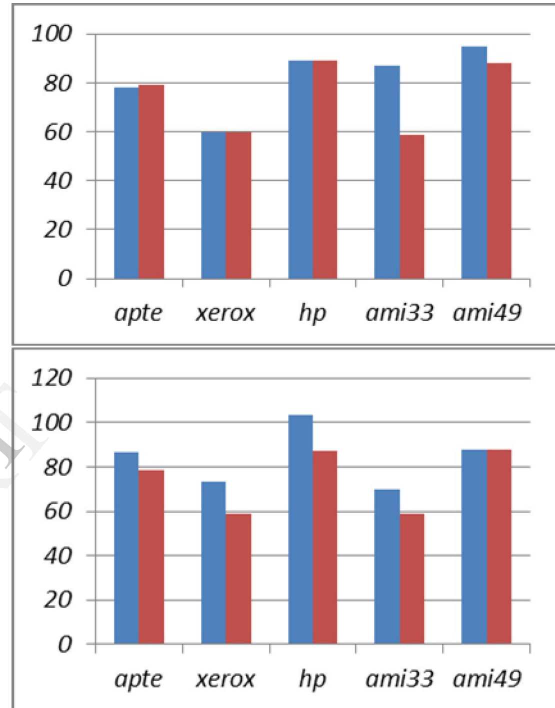

Fig. 6 Graph showing reduction in peak and average temperature in Existing and Proposed Approach

IJERT

V. CONCLUSION

This paper has implemented the novel thermal aware floorplanning method using simulated annealing algorithm. Simulated annealing is used to find the optimum solution no matter where the initial solution begins. The optimized floorplan is obtained that separates the two hot modules thus decreasing the overall die temperature. The future scope of this work is to apply hybrid evolutionary algorithms for the non-sliceable thermal aware floorplanning.

REFERENCES

[1] Y. Liu, R.P. Dick, L. Shang, and H. Yang, "Accurate temperature-dependent integrated circuit leakage power estimation is easy", *DATE*

'07: Proceedings of the Conference on Design, Automation and Test , Europe, pp. 1526-1531, 2007.
[2] K. Skadron, M. Stan, M.R. Huang, S. Velusamy, K. Sankaranarayanan and D. Tarjan, "Temperature aware microarchitecture", *International Symposium on Computer Architecture*, pp. 2-13, 2003.
[3] K. Sankaranarayanan, S.Velusamy, M.R. Stan and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level", *Journal of Instruction-level Parallelism*, pp. 8-16, 2005.
[4] K. Skadron, M. Stan, M.R. Huang, S. Velusamy and K. Sankaranarayanan, "A case for thermal aware floorplanning at the microarchitectural level", *Journal of Instruction-level Parallelism*, pp. 1-16, 2005.
[5] W. Huang, M.R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh and S. Velusamy, "Compact thermal modeling for temperature-aware design", *41st Annual Conference (DAC)*, pp. 878-883, 2004.
[6] W.L. Hung, C. Addo-Quaye, T. Theocharides, Y. Xie, N. Vijaykrishnan and M.J. Irvin, "Thermal aware floorplanning using genetic algorithms", *International Symposium on Quality Electronic Design (ISQED)*, pp. 634-639, March 2005.
[7] Lixia Qi, Yinshui Xia, Lunyao Wang, "Simulated Annealing Based Thermal-Aware Floorplanning" International Conference on Electronics, Communications and Control (ICECC), pp. 463 – 466, Sept 2011.
[8] Y. Han, I. Koren and C.A. Moritz, "Temperature aware floorplanning", *Workshop on Temperature Aware Microarchitectures*, 2005.
[9] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
[10] D. F.Wong, and C. L. Liu, "A New Algorithm for Floorplan Design," *Proc. DAC*, pp. 101–107, 1986.
[11] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-Packing Based Module Placement," *Proc. ICCAD*, pp. 472–479, 1995.
[12] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module Placement on BSGStructure and IC Layout Applications," *Proc. ICCAD*, pp. 484–491, 1996.
[13] P.-N. Guo, C.-K. Cheng, and T. Yoshimura, "An O-Tree Representation of Non-Slicing Floorplan and Its Applications,"*Proc.DAC*, pp. 268–273, 1999.
[14] Y.C. Chang, Y.W. Chang, G.M. Wu and S.W. Wu, "B* trees: a new representation for non-slicing floorplans", *Proceedings of the 37th Annual design Automation Conference*, pp. 458-463, 2000.
[15] 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 Transactions on VLSI System* , Vol.14, No.5, pp. 501-513, May 2006.
[16] Ying-Chieh Chen, Yiming Li, "Temperature-aware floorplanning via geometric programming" *Mathematical and Computer Modeling*, Vol.51, No.7-8, pp. 927-934, April 2010.
[17] Srinivasan J., Adve S.V., Bose P., and Rivers J., "*The Impact of Technology Scaling on Lifetime Reliability*". Proceedings of International Conference on Dependable Systems and Networks, June 2004.
[18] Skadron K., Stan M., Huang W., Velusamy S., Sankaranarayanan K., and Tarjan D., "*Temperature-aware microarchitecture: Modelling and implementation*," Association for Computing Machinery Transactions on Architecture and Code Optimization, Vol. 1, pp. 94, 2004.
[19] "Hotspot-howto," Available: [Online] [http://lava.cs.virginia.edu/HotSpot/HotSpot-HOWTO.htm].
[20] "Mcnc-benchmark," Available: [Online] [http://vlsicad.eecs.umich.edu/BK/MCNCbench/], 2006.

IJERT