International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-12 E-ISSN: 2347-2693

Optimization of delay and temperature for improved design flow in 3D IC

Simi P. Thomas[1*], Reshma Chandran[2], Neethan Elizabeth Abraham[3], Sunu Ann Thomas[4]

[1,2,3,4]Department of Electronics and Communication Engineering, Mangalam College of Engineering, Kerala, India

Available online at: www.ijcseonline.org

Received: 22/Nov/2016 Revised: 06/Dec/2016 Accepted: 20/Dec/2016 Published: 31/Dec/2016

Abstract- Thermal issue is a critical challenge in 3D IC design. To eliminate hotspots, physical layouts are always adjusted by shifting or duplicating hot blocks. However, these modifications may degrade the packing area as well as interconnect distribution greatly. In this paper, we propose some novel thermal-aware incremental changes to optimize these multiple objectives including thermal issue in 3D ICs. Furthermore, to avoid random incremental modification, which may be inefficient and need long runtime to converge, here potential gain is modeled for each candidate incremental change. Based on the potential gain, a novel thermal optimization flow to intelligently choose the best incremental operation is presented. We distinguish the thermal-aware incremental changes in three different categories: migrating computation, growing unit and moving hotspot. Mixed integer linear programming (MILP) models are devised according to these different incremental changes. Experimental results show that migrating computation, growing unit and moving hotspot can reduce max on-chip temperature by 7%, 13% and 15% respectively on MCNC/GSRC benchmarks. Still, experimental results also show that the thermal optimization flow can reduce max on-chip temperature by 14% compared to an existing 3D floorplan tool CBA,and achieve better area and total wirelength improvement than individual operations do.

Keywords- 3D IC technology, Temperature, Floor planning Problem.

I. Introduction

With the fast shrinking of device sizes, interconnect delays become the critical bottlenecks of chip performance. Three- dimensional (3D) integration, as figure 1 shows, recently has drawn much attention due to its potential for reducing the interconnect delay and complexity as well as promising high integration density.

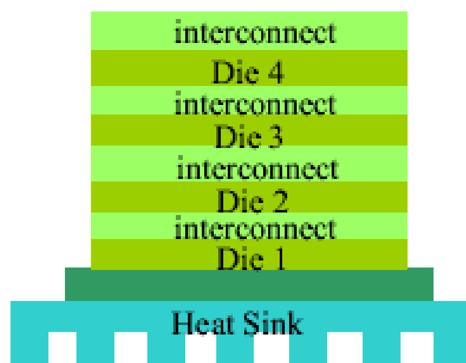


Fig. 1 3D IC technology

Though 3D IC has many advantages, there are some significant challenges along with its adoption and further development. With multi-device layers design, the vertically stacked multiple layers of active devices cause a rapid increase of power density and the thermal conductivity of the dielectric layers inserted between devicelayers for insulation is quite low. Consequently, one extremely important issue in 3D IC design is the thermal problem resulting from both higher power density and lower thermal conductivity.

Recently, several works on thermal optimization during floorplanning for 3D ICs have been proposed [1, 2, 3, 4]. [1] proposed a thermal-driven floorplanning algorithm for 3D ICs. It uses a simulated annealing with an integrated compact thermal model. [2] proposed thermal-aware floorplanning for 3D microprocessors. The power consumption of interconnect is considered during floorplanning. Though the thermal-aware SA- based approaches can indeed distribute heat evenly across the chip to mitigate thermal issue, there is no guarantee to eliminate hotspot completely, sometimes hotspot still exists. To achieve much lower on-chip temperature, minor changes may require a start-over of the floorplanning process, which suffers from long runtime and poor performance scalability. Incremental floorplanning, however, could provide a novel approach: once a good result is obtained, extra thermal improvement can be achieved effectively by eliminating the hotspot incrementally rather than restarting a new general floorplanning.

In the meantime, for an existing floorplan, [5] points out that allocating more die area to blocks especially to hot functional units(growing unit) actually has an immediate impact on the temperature. Still, migrating computation(MC)[6, 7] provides an attractive way to mitigate thermal issue. It requires a duplicated block of the hot block to share computation tasks, which can efficiently reduce power density of the hot block so as to reduce the max on-chip temperature. Evaluation from [8] shows that migrating computation is surely an efficient technique to decrease max on-chip temperature. Indeed, all these methods can be implemented by effective incremental

© 2016, IJCSE All Rights Reserved

modifications to avoid random operations (a) initial floor plan (b) incremental floor plan

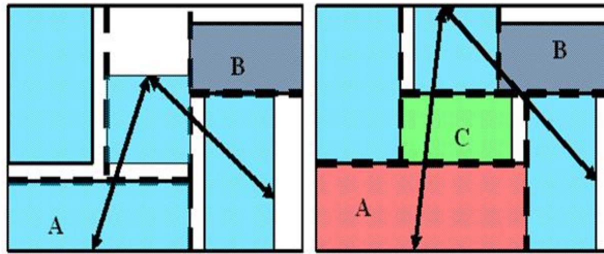


Fig. 2 Growing units and adding duplicated blocks

Obviously, migrating computation demands a new duplicated block while growing unit will enlarge the hotspot block. In fact, both these methods, say adding block and expanding block, will modify the initial floorplan, which would degrade total wirelength or overall packing area. Take Figure 2 as an illustration, the lines with arrows denote interconnections between blocks. In the initial floorplan as shown in figure 2(a), block B needs a duplicated one to migrate computation and block A needs to grow. Figure 2(b) is the incremental floorplan, where block A is enlarged and block C is a clone of block B which is newly added. After re-placing the blocks, the total wirelength might be increased.

Thus an efficient model is required for incremental modifications to achieve good tradeoff between thermal optimization and other objectives. Especially in 3D IC design, incremental optimization is a promising way to handle multi-objective optimization with complicated constraints and facilitate the design reuse technology. Several works concerned with incremental floorplanning for 2D IC design [9, 10, 11, 12] have been proposed, but none has taken thermal-aware 3D IC design into consideration. [13] proposed a LP based approach to optimize white space to facilitate thermal via insertion, but it is hard to be extended to manage such incremental changes as moving blocks between different layers.

Additionally, the model formulations alone barely guarantee preeminent results on both runtime and final objectives. For the purpose of optimizing thermal issue during floorplanning, the question is raised: there indeed exist several incremental changes to choose, but which operation is the best one that brings excellent tradeoff? Select randomly or attempt by brute-force? It seems to be not a good idea, for it may be inefficient and need long runtime to converge.

To free designer from this difficult decision-making, in this paper, we propose a novel thermal optimization flow, which can automatically choose the best procedure, based on potential gain for each possible incremental operation. The flow would bring great benefits to designers musing about how to apply incremental methods. Our contributions are summarized as follows:

- MILP based thermal-aware incremental methods. We categorize three different incremental changes in 3D ICs and provide corresponding MILP formulations respectively.
- Simultaneous optimization for chip area, total wirelength and thermal-driven incremental changes. With effective MILP desirable result of those multiple objectives. To accomplish a better tradeoff, we roll out an evaluation criterion, say potential gain, to select the most suitable operation to process the iterative flow to mitigate the thermal issue, and optimize area and total wirelength.

II. Thermal Resistance Model

For temperature profiling, we use the same thermal resistive model as [1]. The 3D circuit is divided by a two-dimensional array of tile stacks, as shown in Figure 3(a). A tile stack is modeled as a resistive network. Each tile stack is composed of several vertically- stacked tiles, as shown in Figure 3(b). These tile stacks are connected by lateral thermal resistances $R_{lateral}$. Within each tile stack, a thermal resistor R_i is modeled for the i-th device layer, while thermal resistance of the bottom layer and silicon substrate is modeled as R_b as shown in Figure 3(c).

The isothermal bases of room temperature are modeled as a voltage source. A current source is present at every node in the network to represent the heat sources. One can spatially discretize the system and solve the following equation to determine the steady-state thermal profile as a function of power profile.
Thermal-aware optimization flow. Most importantly, we propose a novel thermal-aware optimization flow, which chooses the incremental operations automatically rather than manually to cut design cost and attain high-quality results.

III. Overview of Thermal-AwareIncremental Floorplanning Problem

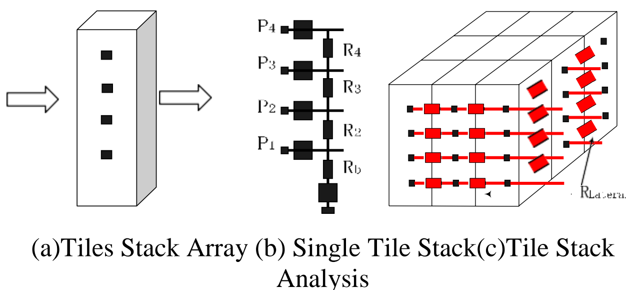
The hotspot, with significantly higher temperature than surrounding cooler regions, could reduce chip reliability and lead to catastrophic failure. To effectively eliminate the hotspots, some incremental changes can be used while the original packing does not need to be changed significantly.

Our thermal-aware incremental floorplanning is an iterative optimization flow. The corresponding problem can be described as: Given a multilayer packing with a set of n blocks M={M_1, M_2, Ă, M_n} in K layers, where w_i and h_i specify the dimensions of block M_i respectively, a set of nets N={N_1, N_2, Ă, N_m} where N_i, i=1,2,Ă, m describes the connections between blocks, we want to generate a new packing where the original topological relations between most blocks remain unchanged so that: 1) the max on-chip temperature can be reduced as much as possible; 2) total wirelength and chip area are degraded little compared with the original design. To mitigate thermal issue, three different incremental floorplanning strategies

© 2016, IJCSE All Rights Reserved

can be applied: Growing unit to allocate more die area around hotspot to reduce max on-chip temperature. In growing unit, the power density is decreased proportionally to the increase of the die area, which can effectively reduce temperature increase from the isothermal point according to [5].

Migrating computation among duplicated blocks. In migrating computation, the hotspot block requires a duplicated one to share computation tasks, which means to halve the power density to reduce the temperature of where A is an K□ K sparse thermal conductivity matrix. T and P(t) are K□ 1 temperature and power vectors. K is the number of thermal conduction edges.



(a)Tiles Stack Array (b) Single Tile Stack(c)Tile Stack Analysis

Fig. 3 Resistive thermal model for 3D ICs

IV. Milp Formulation For GeneralFloorplans

To develop MILP based methods for thermal-aware incremental changes in 3D ICs, multiple objectives and various constraints should be considered at the same time. In this section, we will firstly show how to model these issues in general floorplanning. We use the techniques from [17] in the following subsections A and B.

A. LP model for certain topological Relations

Given a multilayer packing, it is easy to represent the topological relations in linear constraints to prevent overlapping between any pair of rectangular blocks i and j on the same layer, following the techniques in [17]. In the incremental optimization, blocks would deviate from the original packing positions. Let (x_i,y_i) and (x_j,y_j) denote the positions of the lower left corners of block i and j respectively. From the existing floorplan based on certain multilayer representation such as CBA[1] and LTCG[15], we can find the corresponding relative positions of blocks, which keeps unchanged in the optimization process. As a consequence, to prevent overlapping between original blocks i and j on the same layer, one of the following linear inequalities must hold hotspot block.

Moving certain hot block to cooler regions. to relatively cooler area, actually can reduce thermal coupling in the hottest region and decrease the max on-chip temperature, since it could reduce power density and improve the heat dissipation of the hottest area,.

It must be noticed that these modifications are just basic incremental changes in 3D floor planning which waits for the designer to choose. Moreover, just MILP model for uncertain topological relations.

If a new block is added to the existing packing, the relative relations between this new block and the old blocks are unknown.To ensure that one of the inequalities in (2) holds such that the new block does not overlap the present blocks, two additional 0-1 integer variables x_{ij} and y_{ij}, which take only either 0 or 1 value, can be introduced as in [17]. Let us define bounding constants B_w and B_h such that we always have $|x_i-x_j|$”B_w and $|y_i-y_j|$”B_h. Possible choices for B_w and B_h are: B_w =™w_i and B_h =™h_i. Assume block i is the newly added block, we can derive the following constraints:

V. Milp Based Thermal Aware Incremental Floorplanning Methods

A. Special 3D constrained modifications

Our formulation could provide a flexible way to handle constrained modifications in 3D ICs. Here we demonstrate two kinds of such modifications: adding blocks with alignment Where constraints and moving blocks between layers.

Adding blocks with alignment constraints**:** Suppose we have two blocks M_i and M_j in the existing floorplan which must be aligned from high-level design requirement. When a new block is added, this constraint should still be satisfied undoubtedly.

B. LP formulation of chip area

We propose a new optimization model for chip area. Assume W and H are the width and the height of the original packing respectively, to preserve the initial minimum packing area, additional inequalities for each block i are needed as follows:moving hot blocks of other type will be introduced in subsection B. Each action is executed only once here. Because the initial floorplans are packed tightly, the final packings are all enlarged to facilitate addition and growth of blocks. Table 2 shows the experimental results of our approaches. Growing unit enlarges the hotspot block by 3 times on average. As can be seen, growing unit, migrating computation and moving hotspot block reduce max on- chip temperature by 7%, 13% and 15% respectively. Moving hotspot is the best in mitigating thermal issue. Growing unit decreases temperature the least, but it brings the slightest area increase and can reduce total wirelength as well. Migrating computation can notably reduce max temperature but enlarge total wirelength since the duplicated block introduce extra connections with others blocks.

B. Optimization flow for the 3D chip

We run our flow that includes five possible incremental changes on those floorplans generated from CBA[1]. The flow will not exit until no objective improvement can be achieved, or the chip area is enlarged by more than 10% though it may bring great thermal abatement. Table 3

© 2016, IJCSE All Rights Reserved

shows the results of the optimization flow. As can be seen from the table, compared with CBA, our approach can reduce the maximal temperature by about 14%, introducing little time overhead, which shows rapid design convergence. Meanwhile, total wirelength is also decreased by 2%. Because the original floorplans are packed tightly, the chip area is enlarged by 3% and the optimization iterates only a few times according the area constraint. The runtime is mainly spent on solving the MILP formulations rather than invoking the solver. Note that this flow seems to have almost the same thermal optimization effects as moving hotspot block does, however, the flow can attain smaller chip area and total wirelength, which shows the effectiveness of the flow to bring better tradeoff.

VI. Conclusion

In this paper, we propose some novel thermal-aware incremental changes to optimize these multiple objectives including thermal issue in 3D ICs. Furthermore, to improve time-to-market via design cycle reduction, incremental design must move from an expert methodology to a mainstream design methodology: one that is automated, integrated, reliable, and repeatable. To avoid random incremental modification, which may be inefficient and need long runtime to converge, here potential gain is modeled for each candidate incremental change. Based on the potential gain, a novel thermal optimization flow to intelligently choose the best incremental operation is presented. We distinguish the thermal-aware incremental changes in three different categories: migrating computation, growing unit and moving hotspot. Mixed integer linear programming (MILP) models are devised according to these different incremental changes. Experimental results show that migrating computation, growing unit and moving hotspot block can reduce max on-chip temperature by 7%, 13% and 15% respectively on MCNC/GSRC benchmarks. Still, experimental results also show that the thermal optimization flow can reduce max on-chip temperature by 14% compared to an existing 3D floorplan tool CBA, and achieve better area and total wirelength improvement than individual operations do.

References

[1] J.Cong, J. Wei and Y. Zhang, "A Thermal-Driven Floorplanning Algorithm for 3D ICs", in Procceedings of ICCAD, 2004
[2] W. L. Huang, G.M. Link, Y. Xie, N. Vijaykrishnan and M.J Irwin, "Interconnect and Thermal-Driven floorplanning for 3D microprocessors", in Procceedings of ISQED, Mar. 2006
[3] Z.P. Gu, Y. Yang, J. Wang, R.P. Dick and L. Shang, "TAPHS: Thermal aware unified physical-level and high-level synthesis", in Procceedings of ASP-DAC, 2006
[4] P. Zhou, Y. Ma, Z. Li, R.P. Dick, L. Shang, H. zhou, X.L. Hong and Q. Zhou, "3D-STAF: Scalable Temperature and Leakage Aware Floorplanning for Three Dimensional Integrated Circuits", In procceedings of ICCAD, 2007
[5] C.H. Tsai and S.M.S Kang, "Standard cell placement for even on-chip thermal distribution", in Procceedings of ISPD, 1999
[6] K. Skadron, M.R Stan, W. Huang, S. Velusamy, K. Sankaranarayanan D. Tarjan, "Temperature-aware Microarchitecture", in Procceedings of ISCA, 2003.
[7] S. Heo, K. Barr and K. Asanovic, "Reducing power density through activity migration", in Procceedings of ISLPED, Aug., 2003.
[8] T.D. Richardson and Y. Xie, "Evaluation of Thermal-aware design Techniques for Microprocessors", in Proceedings of ASICON, 2005.
[9] J. Cong and M. Sarrafzadeh, "Incremental Physical Design", in Procceedings of ISPD, 2000.
[10] J. Creshaw, M. Sarrafzadeh, P. Banerjee, P. Prabhakaran, "An incremental floorplanner", in Proceedings of GLSVLSI,1999.
[11] S. Liao, M.A. Lopez and D. Mehta, "Constrained Polygon Transformations for Incremental Floorplanning", ACM Trans. On DAES, Vol.6, No.3, July 2001.
[12] X. Tang, R. Tian and M.D.F Wong, "Optimal Redistribution of White Space for Wire length Minimization", In procceedings of ASP-DAC, 2005
[13] X. Li, Y. Ma, X.L. Hong, S. Dong and J. Cong, "LP Based White Space Redistribution for Thermal Via Planning and Performance Optimization in 3D ICs", in procceedings of ASP-DAC, 2008
[14] J. Cong and M. Sarrafzadeh, "Incremental Physical Design", in Procceedings of ISPD, pp.84-92, may, 2000.
[15] H.Y. Jill, E.F.Y Young and R.L.S. Ching, "Block alignment in 3D floorplan using layered TCG", in Procceedings of GLSVLSI, 2006.
[16] www.gnu.org/software/glpk/
[17] S. Sutanthavibul, E. Shragowitz and J.B. Rosen, "An Analytical Approach to Floorplan Design and Optimization", in Procceedings of DAC, 1990.
[18] S.N. Adya, I.L. Markov, "Fixed-outline Floorplanning: Enabling Hierarchical Design", IEEE Trans. On VLSI systems, Vol.11,No.1,pp.1120-1135, Dec.2003.
[19] P. Chen and E.S. Kuh, "Floorplan Sizing By linear Programming Approximation", in Proceedings of DAC, 2000
[20] B. Lall, A. Ortega and H. Kabir, "Thermal Design Rules for Electronic Components on Conducting Boards in Passively Cooled Enclosures", in Proceedings of inter-society Conference on Thermal Phenomerna, 1994

Table 1: Results of different thermal-aware Incremental Floorplannings

	Growing unit				Migrating computation				Moving hotspot block			
	Area(um^2)	T_{max}(^oC)	WL(um)	Cpu(s)	Area(um^2)	T_{max}(^oC)	WL(um)	Cpu(s)	Area(um^2)	T_{max}(^oC)	WL(um)	Cpu(s)
Ami33	364077	409.45	25015	0.3	361883	337.77	33111	1.6s	361883	322.14	30524	1.7
Ami49	12413700	349.63	179368	1.6	12413700	342.15	198711	6.1	12413700	334.39	181969	6.9
N100	53918	290.68	60586	8.6	57431	273.24	72622	21.0	57431	272.78	71043	20.8
N200	65345	273.05	165801	18.4	68362	237.13	179565	85.7	68362	233.37	177421	80.6
N300	93227	275.37	262830	29.0	94050	269.5	257049	256.2	94050	268.42	255435	247.3
Avg.	**1.04**	**0.93**	**0.92**		**1.06**	**0.87**	**1.04**		**1.06**	**0.85**	**1.00**	

© 2016, IJCSE All Rights Reserved

Table 2: Results of the iterative optimization flow

Benchmark	CBA						CBA + optimization flow				
	Block#	Net#	Area(um^2)	T_{max}(^oC)	WL(um)	Cpu(s)	Area(um^2)	T_{max}(^oC)	WL(um)	Cpu(s)	Iteration#
Ami33	33	123	342504	462.3	29640	31	361883	322.1	30524	32.7	1
Ami49	49	408	12413700	358.8	187065	97	12309800	284.6	196600	111	3
N100	100	885	51983	303.7	68899	401	55615	287.6	60946	422.1	1
N200	200	1585	60652	303.4	174921	2273	63736	271.3	165265	2464.9	3
N300	300	1893	91025	288.7	267944	4081	91025	274.0	268107	4592.3	2
Avg.			1	1	1	1	**1.03**	**0.86**	**0.98**	1.09	

© 2016, IJCSE All Rights Reserved