Wire Congestion And Thermal Aware 3D Global Placement

Karthik Balakrishnan, Vidit Nanda, Siddharth Easwar, and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology
{gte245v, gte272u, gte581t, limsk}@ece.gatech.edu

***Abstract*— The recent popularity of 3D IC technology stems from its enhanced performance capabilities and reduced wirelength. However, wire congestion and thermal issues are exacerbated due to the compact nature of these layered technologies. In this paper, we develop techniques to reduce the maximum temperature and wire congestion of 3D circuits without compromising total wirelength and via count. Our approach consists of two phases. First, we use a multi-level min-cut placement with a modified gain function for local wire congestion and dynamic power consumption reduction. Second, we perform simulated annealing together with full-length thermal analysis and global routing for global wire congestion and maximum temperature reduction. Our experimental results show smooth tradeoff among congestion, temperature, wirelength, and via.**

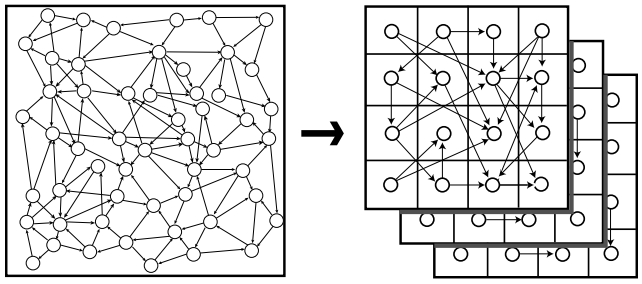

Fig. 1. Illustration of 3D global placement. Inter-layer connections are not shown for simplicity.

I. Introduction

With the recent advent of Three-dimensional Integrated Circuit technologies, there has been a positive impact on the performance and wiring length of these ICs. Typically, the layered placement of transistors in multiple planes (i.e. 3D placement) allows for a more compact chip with inherently better performance than one fabricated with traditional 2D placement techniques. However, the stacked nature of these circuits induces and aggravates problems of non-uniform thermal dissipation as well as local and global wire congestion. Simultaneously, it is necessary to minimize the wiring in single layers as well as the interconnect among different layers so as to maintain routability. Recent works in the area of 3D detailed placement include [1], [2], [3], [4], [5], [6]. Today's placement problem is divided into global and detailed placement in much the same way as in the division of global and detailed routing. Global placement determines the region for a group of cells to be located, whereas detailed placement removes overlap and performs legalization in each region while preserving the global placement solution as much as possible.

In this paper, we provide a technique to reduce both local and global congestion in a 3D circuit in order to increase the routability of the chip. We also improve the temperature profile of the circuit using state-of-the-art thermal simulator. Our approach involves a two-stage refinement procedure: Initially, we use a multilevel min-cut based method to minimize the congestion and power dissipation within confined areas of the chip. This is followed by a simulated annealing-based technique that serves to minimize the amount of congestion created from global wires as well as to improve the temperature distribution of the circuit. We show that our congestion and maximum temperature minimization does not have any significant negative impact on the inter-layer wirelength or the number of intra-layer vias. Our contributions include a flexible multi-objective optimization technique for 3D VLSI circuits, the incorporation of accurate full-length thermal analysis at the global placement phase in the design process, and a thorough analysis of the correlations among local congestion, global congestion, and thermal quality.

II. Preliminaries

A. Problem Formulation

The purpose of the 3D Global Placement Problem (3D-GP) is to assign cells in a sequential netlist NL to a given $m \times n \times l$ 3D slots while satisfying the prescribed area constraints for each slot (= alternatively called block in this paper). Given a 3D-GP solution P, our primary objective is to minimize wire congestion and maximum steady state temperature among all blocks. As secondary objectives, we minimize wirelength and via induced by P. For a net n with p pins ($|n| = p$), let $n(B)$ denote the number of pins contained in a block B. If $n(B) = q(q > 0)$, $q - 1$ is the number of edges in a spanning tree that connect these q pins in B. In addition, these edges are viewed as *local wires* since these are intra-block connections. Given a block B from a 3D-GP problem, we define the local wire congestion, denoted $LC(B)$, as follows:

$$LC(B) = \sum_{n \in N, n \cap B \neq \emptyset} |n(B)| - 1$$

This value corresponds to the total number of local wires contained in B. Then, the *local wire congestion* of a 3D-GP

solution is given by:

$$LC = \max\{LC(B_i) - LC(B_j)\},\ 1 \leq i, j \leq K \quad (1)$$

LC denotes the difference between the maximum and minimum local congestion among the blocks.

For any two adjacent blocks B_i, B_j from a 3D-GP solution, N_{ij} denotes a set of all nets having pins in both B_i and B_j. Then, the *global wire congestion* at boundary $[B_i, B_j]$, denoted $GC(B_i, B_j)$, is simply $|N_{ij}|$. This value denotes the total number of global wires routed through the given boundary. Then, the *global wire congestion* of a 3D-GP solution, denoted GC, is simply the maximum global wire congestion among all boundaries in the 3D grid.

B. Overview of the Approach

During the first stage, mincut placement is performed to divide the input netlist into $K(= m{\cdot}n{\cdot}l)$ blocks and place them in a 3D grid. The objective during the recursive bipartitioning is to balance local wire congestion between two sub-blocks while minimizing dynamic power consumption. We introduce the concept of *local congestion gain* to minimize local wire congestion. In addition, we use the *dynamic power gain* discussed in [7] to minimize the dynamic power consumpsion (which has a direct impact on thermal profile of the final global placement). We then perform multi-level partitioning [8] to minimize the weighted cutsize. During the second stage, simulated annealing based refinement is performed to improve global wire congestion and maximim temperature. We generate a new candidate 3D global placement solution by swapping a pair of random blocks and measure its quality in terms of temperature and global wire congestion. For this purpose, we perform thermal simulation to compute the steady state temperatures of the blocks and global routing to compute the global wire congestion among all boundaries in the 3D global placement. We perform incremental 3D maze routing to accurately and efficiently measure global wire congestion for each candidate solution. However, temperature calculation cannot be done in the same way due to its prohibitive runtime. Therefore, we perform thermal simulation periodically and use these results to interpolate temperature values in between.

III. 3D Mincut Global Placement

The purpose of this step is to balance the local wire congestion while minimizing the dynamic power consumed by the nets that are cut during the 3D partitioning. Our cut sequence is an extension of the two cut sequence techniques used in [2]. Their first method performs via-minimizing inter-layer cuts (z cuts) before performing intra-layer cuts (x, y) to minimize the 2D wirelength. Their second cut sequence does the opposite, making all (x, y) cuts first before performing z-cuts to achieve minimal wirelength. For the purposes of maintaining a balanced combination of via count and wirelength during our algorithm, we devise a new cut sequence, $(z, x, y, z, x, y, \cdots)$. We experimentally determined that the best results in terms of balanced wirelength and via count were produced by this new cut sequence.

A. Local Congestion Gain Computation

For a given cell c, if moved from B_i to B_j, the change in local wire congestion for block B_i, denoted δ_i, is computed as follows:

$$\delta_i = LC(B_i) - LC(B_i - \{c\})$$

δ_j is computed using $LC(B_j)$. The *local wire congestion balance gain* of cell c for moving from block B_i to block B_j is given by:

$$g_l = |LC(B_i) - LC(B_j)| - |LC(B_i) - LC(B_j) + \delta_i + \delta_j|$$

This gain represents how much $|LC(B_i)$- $LC(B_j)|$ is reduced. Thus, the cell move based on this gain will improve the balance between the local wire congestion of the two blocks. Note that g_l can be computed by looking at only the neighboring nodes and current local wire congestion values of the two blocks. Thus, we can incrementally update g_l for all neighboring cells as well as $LC(B_i)$ and $LC(B_j)$ upon each cell move efficiently.

We note that performing cell moves purely based on g_l has negative impact on cutsize minimization. Therefore, we use g_l only when the difference between $LC(B_i)$ and $LC(B_j)$ is greater than a threshold value. Otherwise, we use the conventional cutsize again g_c for cutsize minimization. Therefore, we maintain both g_l and g_c for all cells and selectively use them so that we focus on local wire congestion balancing only when it is necessary.

IV. 3D Global Placement Refinement

We use a simulated annealing-based refinement for minimizing the maximum temperature, global wire congestion, wirelength, and via count of the 3D global placement solution obtained from our 3D mincut global placement. We generate a new candidate 3D global placement solution by swapping two blocks. We then use the following cost function to measure its quality.

$$C = \alpha \cdot XT + \beta \cdot GC + \gamma \cdot wirelength + \delta \cdot via$$

where XT and GC respectively denote the maximum temperature and global wire congestion discussed in Section II-A. As is standard with all annealing algorithms, improvements are guaranteed only at a significant runtime expense. In order to make the procedure as efficient as possible, it becomes necessary to perform highly optimized incremental evaluation of these metrics.

A. Incremental Congestion Analysis

We extend the existing maze routing algorithm to accurately estimate global wire congestion in a given 3D global placement solution. Before we start the annealing process, we decompose each multi-terminal net into a set of two-terminal nets. We then visit each net and find the shortest path between the source and the sink blocks. In this case, we ignore the current usage of the boundaries in the 3D grid. After we finish routing all nets, we begin the annealing process. When a pair

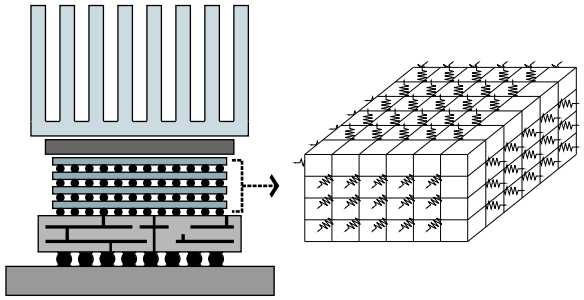

Fig. 2. 3D thermal modeling

of blocks is swapped during each move, we find a set of nets that are affected by this swap, i.e., all nets that have connection to either block for re-routing. We then find the weighted shortest path for all nets in this set, where the weight represent the current usage of the boundaries each net uses. This ensures that the global wire congestion is considered during the re-routing. This routing update can be done efficiently since the number of nets that are affected is not huge.

B. 3D Thermal Analysis

The temperature profile of a design is governed by the following partial differential equation for heat conduction:

$$\rho C_p \frac{\partial T(\vec{r},t)}{\partial t} = \nabla \cdot [\kappa(\vec{r},T)\nabla T(\vec{r},t)] + g(\vec{r},t) \qquad (2)$$

where ρ is the density of the material, C_p is the specific heat of the material, κ is the thermal conductivity of the material, T is the time dependent temperature, and g is the heat generation rate. The solution for this equation is set the by the following boundary condition equation:

$$\kappa(\vec{r},T)\frac{\partial T(\vec{r},t)}{\partial n_i} + h_i T(\vec{r},t) = f_i(\vec{r}_{s_i},t)$$

where h_i is the heat transfer coefficient, and f_i is an arbitrary function on the boundary surface.

Our 3D circuit thermal modeling is shown in Figure 2, where we compute $T(x,y,z,t_s)$, the steady state temperature at each grid point (x,y,z) at time t, by solving Equation 2. For this purpose, we use the 3D Alternate Direction Implicit Method (3D-ADI) presented in [9]. Each time step is split into three equal smaller time steps, with only one direction–x, y, or z–implicit in each step. Since each of the equations in ADI method can be represented as a tridiagonal system of equations, and hence be solved using the Thomas algorithm, run time is linearly proportional to the number of temperature nodes. Once a steady state temperature profile has been achieved, various physical parameters such as maximum temperture, temperture gradient, and average temperature are calculated.

Although 3D-ADI thermal simulator has a linear runtime and memory requirement and is unconditionally stable, it is still too expensive from a runtime point of view to call a full thermal simulation at every move in simulated annealing. In order to tackle this problem, we perform thermal simulation at every m moves and use these results to interpolate the maximum temperature for every candidate solution. We to allow the value of the interpolated temperature value to vary only within a certain range so that we prevent a pathological configuration from becoming the best solution.

V. Experimental Results

Our algorithms were implemented in C++/STL, compiled with gcc v2.96 with -O3, and run on Pentium III 746 MHz machines. The benchmark set consisted of six circuits from ISCAS89 and five circuits from ITC99 suites. We used $8\times8\times4$ for our 3D grid size. We use 10% as the threshold during local wire congestion balancing, i.e., if the unbalance between two blocks is more than 10%, we use the congestion gain during cell move.

Table I shows our mincut-based global placement results. First, we observe from comparing the wirelength/via-driven algorithm with the local congestion-driven algorithm that the improvement on local wire congestion balance is significant–the average improvement is 65%. However, this gain came at the cost of 77% increase in wirelength and 109% increase in via counts. The runtime also increased significantly. We note that the total number of passes used in each bipartitioning tends to increase in case of local congestion optimization. Second, our dynamic power optimization has positive impact on reducing the maximum temperature among the blocks. The average improvement on the temperature is 22% compared to the baseline (= wirelength/via-driven algorithm). In addition, the increase in wirelength and via is not significant–only 17$ and 10% on average, respectively. Our multi-level partitioning with net weighting scheme was successful in improving the thermal cost in 3D global placement. Third, by combining all four objectives (= wirelength, via, congestion, and power) in our 3D mincut global placement, we obtained a very smooth tradeoff curve–a 25% and 12% reduction on local congestion balance and maximum temperature came at the cost of 18% and 15% increase in wirelength and via, respectively.

Table II shows our simulated annealing-based refinement results. First, our incremental maze routing-based global congestion control proves to be effective–an average improvement of 20% was obtained compared to the baseline (= wirelength/via-driven algorithm). This improvement came at the cost of wirelength and via increase. Interestingly, the maximum temperature also dropped by 21% on average. This implies that global wire congestion can adversely affect the thermal profile of 3D global placement. Second, our thermal-driven algorithm using thermal simulation and interpolation is also effective in reducing the maximum block temperature–an average improvement of 23% was obtained at the cost of 20 to 30% increase in other objectives. The runtime also increased due to frequent thermal simulation. Third, by combining all four objectives (= wirelength, via, congestion, and temperature) in our 3D placement refinement, we obtained a very smooth tradeoff curve–a 7% and 25% reduction on global wire congestion and maximum temperature came at the cost of 8% and 10% increase in wirelength and via, respectively.

TABLE I

LOCAL WIRE CONGESTION AND THERMAL OPTIMIZATION RESULTS

ckts	wirelength/via-driven				local-congestion-driven				power-driven				wl+via+con+power			
name	wire	via	l-wc	temp	wire	via	l-wc	temp	wire	via	l-wc	temp	wire	via	l-wc	temp
s5378	2315	232	26	12.39	3386	418	14	11.09	2517	281	21	13.5	2374	267	19	13.5
s9234	2232	250	59	30.23	3931	476	16	19.52	2639	276	54	16.98	2726	271	36	26.24
s13207	2332	293	77	11.7	4793	618	21	27.78	2934	347	99	21.48	3139	413	63	18.23
b14_opt	5633	635	40	34.81	8204	1110	17	44.67	7045	724	39	30.95	7024	727	22	34.3
b15_opt	8478	960	40	20.86	12418	1856	18	22.04	9445	1081	41	22.66	9355	1119	39	29.42
b20_opt	8859	1039	70	19.51	15185	2184	27	23.5	10582	1057	60	20.81	11035	1123	49	17.36
b21_opt	9445	1069	86	28.95	14750	1732	27	20.54	10631	1182	62	16.31	11259	1304	56	18.92
b22_opt	11020	1410	93	25.73	19351	2847	32	16.75	12748	1346	87	18.4	12154	1321	73	19.57
s38417	3945	433	144	10.08	9560	1548	55	9.94	4234	445	135	14.2	4578	510	110	12.07
s35932	2922	323	72	45.93	7292	944	30	18.99	3951	393	80	10.47	3864	389	72	11.48
s38584	4645	407	120	16.15	10845	1021	31	20.03	5420	623	126	14.34	5524	636	84	25.75
RATIO	1.00	1.00	1.00	1.00	1.77	2.09	0.35	0.92	1.17	1.10	0.97	0.78	1.18	1.15	0.75	0.88
TIME	216				1467				844				1622			

TABLE II

GLOBAL WIRE CONGESTION AND THERMAL OPTIMIZATION RESULTS

ckts	wirelength/via-driven				global-congestion-driven				thermal-driven				wl+via+con+ther			
name	wire	via	g-wc	temp	wire	via	g-wc	temp	wire	via	g-wc	temp	wire	via	g-wc	temp
s5378	2095	257	27	13.40	3296	298	23	9.72	2741	299	29	13.50	2235	278	27	11.13
s9234	2068	244	30	33.18	3998	360	33	18.58	2845	301	42	15.79	2106	255	38	17.54
s13207	2357	215	31	12.85	4514	491	24	26.84	3120	355	55	20.15	2587	233	29	24.91
b14_opt	5071	649	57	37.34	8067	953	60	43.73	7641	768	84	32.79	5548	688	66	32.09
b15_opt	7491	942	100	25.39	12633	1734	82	21.10	10245	1254	108	20.81	7698	1054	98	24.17
b20_opt	8883	945	88	22.28	15076	2033	61	22.56	11598	1200	99	19.31	9056	956	72	21.63
b21_opt	9189	1034	95	30.15	13514	1576	60	19.60	12455	1278	124	16.55	9312	1154	71	15.39
b22_opt	10146	1242	102	29.58	21399	2936	92	15.81	14106	1569	146	17.56	12454	1465	103	19.21
s38417	3615	464	49	10.30	9520	1433	24	9.01	4415	455	56	13.14	3945	497	28	18.31
s35932	3059	325	38	49.98	7343	823	24	18.05	4046	416	42	9.53	3264	377	29	12.23
s38584	4628	566	54	19.71	10501	890	52	19.09	5745	667	88	12.30	4956	604	66	15.70
RATIO	1.00	1.00	1.00	1.00	1.87	1.97	0.80	0.79	1.35	1.24	1.30	0.67	1.08	1.10	0.93	0.75
TIME	2034				7936				9951				11114			

VI. CONCLUSIONS

We presented a 3D global placement algorithm for wire congestion and maximum temperature reduction. We first performed a multi-level min-cut placement with a modified gain function for local wire congestion and dynamic power consumption reduction. We then performed simulated annealing together with full-length thermal analysis and global routing for global wire congestion and maximum temperature reduction. We obtained smooth tradeoff among congestion, temperature, wirelength, and via costs.

ACKNOWLEDGMENT

This research is partially supposed by the National Science Foundation under project number EEC-9402723.

REFERENCES

[1] T. Tanprasert, "An analytical 3-D placement that reserves routing space," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 2000.

[2] S. Das, A. Chandrakasan, and R. Reif, "Design tools for 3-D integrated circuits," in *Proc. Asia and South Pacific Design Automation Conf.*, 2003.

[3] R. Zhang, K. Roy, C.-K. Koh, and D. B. Janes, "Exploring SOI device structures and interconnect architectures for 3-dimensional integration," in *Proc. ACM Design Automation Conf.*, 2001.

[4] S. Das, A. Chandrakasan, and R. Reif, "Timing, energy, and thermal performance of three-dimensional integrated circuits," in *Proc. Great Lakes Symposum on VLSI*, 2004.

[5] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2003.

[6] G. Chen and S. Sapatnekar, "Partition-driven standard cell thermal placement," in *Proc. Int. Symp. on Physical Design*, 2003.

[7] M. Ekpanyapong, K. Balakrishnan, V. Nanda, and S. K. Lim, "Simultaneous delay and power optimization for multi-level partitioning and floorplanning with retiming," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 2004.

[8] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning : Application in VLSI domain," in *Proc. ACM Design Automation Conf.*, 1997, pp. 526–529.

[9] T.-Y. Wang and C. C.-P. Chen, "3-d thermal-adi: A linear-time chip level transient thermal simulator," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1434–1445, 2002.