MCM Placement Using A Realistic Thermal Model

Craig Beebe, Jo Dale Carothers, and †Alfonso Ortega
University of Arizona

ABSTRACT —

Typically, placement algorithms attempt to minimize the total net length of a printed circuit board (PCB). However, an MCM's increased throughput and dense circuitry can easily result in failure if the board contains "hot spots". Therefore, an accurate thermal model of an MCM was needed in the development of a new placement algorithm designed to consider both total net length and thermal constraints. This algorithm uses a combination of simulated evolution and simulated annealing in an iterative approach. Each chip has a maximum thermal tolerance that it can withstand before it is known to fail. The fitness method evaluates the maximum temperature for each chip, considering every chip's thermal dissipation at the chip's hottest point. Results are presented that compare the effects of various parameters.

I. INTRODUCTION

An overwhelming majority of existing placement algorithms only attempt to optimize for the routability of a circuit. Sherwani [12] provides a summary of the classic techniques. These algorithms typically concentrate on minimizing total net length, while others focus on minimizing wire crossovers and vias [14]. These techniques are reasonable for most printed circuit boards (PCB); however, the recent popularity of multichip modules (MCM) has uncovered the fact that MCMs require a placement algorithm that seeks to optimize more than the aforementioned criteria [12].

The chips within an MCM are not individually packaged and therefore may dissipate significantly more heat than their packaged counterparts for a given temperature rise. In addition, MCM chips are placed much closer together, resulting in an overall temperature gain that is not usually seen on PCBs. It is conceivable that a placement tool that does not consider the chips' thermal properties would choose to place two or more chips close enough together to create a "hot spot". This would significantly reduce the life of the board or MCM.

While other thermal dissipation methods have been used, such as adding heat sinks or thermal vias, it is not always feasible to use these techniques to sufficiently cool the MCM. Many times heat sinks are not an option because of space limitations. Thermal vias are problematic to routing heuristics because they represent additional obstacles. The more obstacles a router has to avoid, the larger the total net length will be. In fact, too many obstacles can cause the layout to be unroutable.

Previous authors have attempted to provide a balance of temperature over the area of the substrate while reducing the total net length [2, 4, 11, 13]. Chao [2] and Osterman [11] present realistic thermal models, but the heuristics used are known to be fast rather than optimal: Chao uses the min-cut and simulated annealing algorithms, while Osterman uses the force-directed placement technique. Chu [4] presents a matrix approach for FPGA placement, but the technique cannot be applied to MCMs because the thermal model merely consists of a single value for each gate. Tang [13] uses this thermal model in an MCM placement tool. The genetic algorithm in series with a simulated annealing heuristic are combined to achieve good placements; however, the limitation of the thermal model prevents it from finding fully reliable solutions.

The motivation behind this work is to interface a realistic model of an MCM, its components, and their thermal properties with a dynamic set of optimization algorithms. The two objectives and constraints of the placement tool discussed are the total net length and the thermal dissipation of the chips.

II. PLACEMENT ALGORITHMS

The interpretation of the simulated evolution algorithm [5, 7] used here is explained in [1]. Each chromosome in the population is an MCM with an initial placement scheme, and each gene is a chip within the MCM. The strength of a chromosome is determined first by whether the thermal constraint has been violated. If so, the chromosome is given an extremely weak value for its fitness; otherwise, the total net length inversely determines a chromosome's strength.

The usage of simulated annealing [3, 6, 10] is further explained in [1]. The heuristic first initializes the fitness value for the material in question. Then, as the temperature drops, it repeatedly calls the perturb() operator and compares the new fitness with the old. If the new fitness is better, it keeps it. If not, it determines whether it wants to keep it by finding a random number between $[0.0, 1.0)$ and comparing that to $e^{\frac{oldfitness - newfitness}{currenttemperature}}$.

One drawback to many optimization heuristics is that they can fall into a local minimum and are unable to escape it. A solution to this problem is to execute more than one algorithm in series. In [13], Tang executes the evolution heuristic, followed by the annealing heuristic with a short cooling schedule. This resulted in better solutions when compared to

University of Arizona, ECE Dept. and †AME Dept., Tucson, Arizona 85721.
email: beebe@desert.ece.arizona.edu, carothers@ece.arizona.edu, ortega@u.arizona.edu
This research was supported by the National Science Foundation under grant #9554561.

```
Create and initially place chromosomes
For i = 1 to number of iterations
    Execute genetic heuristic until all chromosomes
        have equal fitness
    For j = 1 to number of chromosomes
        Execute annealing heuristic on
            chromosome(j)
    Next j
    Execute genetic hueristic on resulting
        chromosomes numGens/i times
Next i
```

Fig. 1. Pseudocode for the Hybrid Evolution and Annealing Algorithm

the simulated evolution running alone. The annealing process often assisted the candidate out of the local minimum upon which the evolution heuristic converged.

In this research, a variation on Tang's hybrid approach that reaches good results alternates between simulated evolution and simulated annealing. It enters a loop whereby the evolution heuristic executes until every chromosome has the same fitness value. Assuming that at this point the evolution heuristic has reached a local minimum, the code applies the annealing heuristic, not just to the best resulting chromosome, but to every chromosome in the chromosome list. This randomizes the chromosomes somewhat, improving some of their placements while leaving others with worse placements. The evolution heuristic is executed again with the modified chromosomes, then loops back. It performs this a predetermined number of times, although it was found that MCMs with many chips require more iterations. The pseudocode of this overall algorithm is shown in Fig. 1.

III. THERMAL MODEL

The heat dissipated from a chip on a substrate is partly convected away from the surface of the chip and the remainder is conducted into the substrate. The heat conducted into the substrate spreads radially due to heat conduction.

The center of the chip is the hottest point. As the radius from the center increases, the temperature decreases monotonically in a Gaussian-like curve. If the chip is approximated as a circular source of heat of zero thickness on a substrate of thickness t, and if the upper surface cooling rate is given by a convective cooling law, the temperature is given by a one-dimensional ordinary differential equation in the radial dimension [8].

The solution to the equation gives the temperature of any given point on the substrate due to one chip. Because the governing equations and boundary conditions are linear with respect to temperature, superposition can be used to add the thermal effects of each chip on a substrate. Thus, to find the temperature gain for a given point, the thermal effect of each chip at that point can simply be superposed.

Only the hottest spots on the board are of any interest to the optimization algorithms. If one of these spots is too hot, all other thermal calculations are superfluous. Moreover, if all of the hottest spots are within a certain tolerance, no other calculations are necessary. The center of the chip under consideration is the point at which every other chip's thermal influence is evaluated. These values are summed, then added to the local maximum temperature rise for the chip being evaluated. This provides the overall thermal gain at the hottest spot for each chip. If this value exceeds a user-defined threshold, called the maximum chip to ambient thermal resistance (which can be different for each type of chip), the placement is discarded. Every chip's hot spot is similarly examined; if each are within their threshold, it is deemed a reasonable placement.

A distinct advantage that this model has over Tang's [13] is that, if two "hot" chips share several nets, it will attempt to place them as close to each other as possible, given the aforementioned constraint. Using a thermal model that attempts to balance the temperature across the board will force the algorithm to find a compromise between the thermal constraints and net length objective. Because these are conflicting factors, the compromise will not be optimal for either objective.

IV. RESULTS

Each chip's thermal model accepted two parameters as input: the maximum chip to ambient thermal resistance ratio ($MCATR$) and h, the heat transfer coefficient. Due to the lack of thermal data in standard benchmarks, these parameters, along with the substrate's thickness, were given reasonable yet somewhat random values.

Two MCMs tested were taken from the examples used to explain the EDIF format. The first of these two MCMs, named AMI33, contains 33 chips, each one different with significantly different sizes. It has 117 nets, some of which are multi-terminal. AMI49 contains 49 chips, all different, with a wider range of sizes than AMI33, as well as 407 nets.

Two other MCMs tested are the standard MCC1 and MCC2 benchmarks. They are commonly used to compare both placement and routing algorithms. MCC1 consists of 6 chips, 765 I/O pins, and contains 799 signal nets. There are two types of chips: C448, with a geometry of 550 x 550 mils and 448 pins; and C272, with a geometry of 330 x 330 mils and 272 pins. The MCM consists of four C448s and two C272s.

MCC2's substrate is 6 x 6 inches with 37 Honeywell VHSIC gate arrays and 18 high density connectors. The chips are all of the same type and have a geometry of 1.5 x 1.5 cm with 548 pins. The connectors are placed around the perimeter of the substrate. The net list contains 7118 signal nets and a total of 14659 pins.

The tests were performed on a Sun Ultra 1 running Solaris 2.5. The code was compiled using g++ version 2.7.2. Comparisons of net length and execution time to other thermal placement algorithms can be found in [1].

Five placements of MCC1 were found to illustrate how d-

TABLE I
HEAT TRANSFER COEFFICIENT (h) AND $MCATR$ (M) VALUES FOR MCC1

Chip	h_1 M_1	h_2 M_2	h_3 M_3	h_4 M_4	h_5 M_5
C448	10	7	10	10	10
	1.3	1.3	1.29	1.28	1.27
C227	12.5	8.5	12.5	12.5	12.5
	1.4	1.4	1.39	1.38	1.37

TABLE II
NET LENGTHS FOR MCC1 (mm)

$MCC1_1$	$MCC1_2$	$MCC1_3$	$MCC1_4$	$MCC1_5$
22,700	22,800	23,600	24,700	24,900

ifferent values for the chips' heat transfer coefficient h and the $MCATR$ ratio affects both the resulting layout and the total net length. See Table I for specific values. The other parameters were kept constant between the five placements: number of simulated evolution generations = 100, number of chromosomes = 40, random number seed = 782271, initial placement spacing = 11%, and number of iterations = 8. Table II compares the approximated net lengths of the five placements. Each run took 14 minutes to complete.

MCC2 is more complex because of the number of nets involved. MCC1 used 40 chromosomes, and could have used more if it had needed them. In fact, 100 chromosomes were used initially, but the execution time was too slow to justify the small improvement in net length. Table III compares the number of chromosomes used for MCC2 to the resulting net lengths and execution times. As with MCC1, the other parameters were left constant.

The variation in results due to the number of chromosomes used in MCC2 led to another series of tests on the other benchmarks to determine if there exists a correlation between the number of chromosomes used and the resulting net length. The results are shown in Tables IV, V, and VI. Net Length 1 began with an initial spacing of 5%, Net Length 2 began with 8%, and Net Length 3 began with 11%. Some of the best results for a given circuit and spacing are underlined. Note that these results are not always produced with the largest number of chromosomes. Generally speaking, there is a local minimum in the range of 35 - 40 chro-

TABLE III
NUMBER OF CHROMOSOMES FOR MCC2 VS. NET LENGTH, PERCENT IMPROVEMENT, AND EXECUTION TIME

Num	Len (mm)	Improvement	Time (min)
10	330,635	42%	38
15	317,625	34%	60
20	319,932	34%	76
25	330,176	32%	100

TABLE IV
AMI33 WITH INCREASING CHROMOSOME COUNT

Num	Len1 (mm)	Len2 (mm)	Len3 (mm)
10	163.7	164.5	158.9
15	163.7	139.7	143.5
20	164.0	144.4	142.0
25	163.2	134.6	138.1
30	160.9	141.6	136.4
35	161.2	137.6	133.7
40	152.7	134.2	135.0
45	154.6	134.3	132.4
50	165.1	137.5	137.6
55	153.7	135.2	139.1
60	148.7	137.0	135.3

TABLE V
AMI49 WITH INCREASING CHROMOSOME COUNT

Num	Len1 (mm)	Len2 (mm)	Len3 (mm)
10	1,730	1,788	1,859
15	1,788	1,798	1,880
20	1,702	1,786	1,814
25	1,681	1,703	1,761
30	1,759	1,779	1,776
35	1,659	1,688	1,797
40	1,687	1,747	1,679
45	1,693	1,700	1,717
50	1,666	1,678	1,777
55	1,672	1,757	1,755
60	1,653	1,646	1,673

TABLE VI
MCC1 WITH INCREASING CHROMOSOME COUNT

Num	Len1 (mm)	Len2 (mm)	Len3 (mm)
10	n/a	24,990	25,360
15	23,322	23,017	24,910
20	24,413	23,711	29,108
25	23,579	23,596	23,539
30	23,524	23,476	23,377
35	23,629	23,307	22,609
40	23,507	23,302	21,552
45	22,748	23,556	22,123
50	23,073	23,249	21,327
55	23,077	23,150	21,482
60	23,500	23,183	21,585

TABLE VII
MCC NET LENGTH IMPROVEMENT

Circuit	MTP Improvement	MPH Improvement
MCC1	10%	11%
MCC2	11%	11%

mosomes for MCMs with a small number of nets (AMI33 and AMI49), while MCC1 results in a local minimum of 45 - 55 chromosomes.

The MCC benchmarks define their own placement so routing algorithms can compare their results with other routers. This is also helpful to compare placement algorithms. For example, Tang used the MCG router [9] to compare his placement of the MCC benchmarks with their original placements. A similar approach was taken with this work: the MCG router was also applied to both the original placements and the best placement found above. The resulting net lengths are compared to their respective original placements. Table VII shows the percent improvement in net length for both benchmarks and for both placements algorithms. This illustrates that the improved thermal model did not adversely affect the routability.

V. CONCLUSIONS

The algorithms and heuristics used in this work yielded excellent results, especially when fused into the hybrid approach. The results were comparable to those of other placement algorithms, so the thermal constraint has been shown to be a reasonable consideration. It was shown that the two thermal parameters, h and $MCATR$, affected the placement algorithms' decisions and therefore the resulting placements.

The implementation of the thermal model was crucial to the resulting execution time. The assumption that the hot spot of each chip will be at its center allowed for a tremendous speedup in the thermal calculations. Since only $C(C-1)$ calculations of the temperature gradient per placement were required, where C is the number of chips, the overall thermal dissipation could be evaluated quickly enough to be useful to a placement algorithm.

References

[1] C. Beebe, J. D. Carothers, A. Ortega, "Object-Oriented Thermal Placement Using an Accurate Heat Model," *Hawaii International Conference on Systems Sciences*, Wailea, Hawaii, 1998.

[2] K. Chao, D. Wong, "Thermal Placement for High-Performance Multichip Modules," *International Conference on Computer Design*, Austin, Texas, pp 218 - 223, 1995.

[3] A. Chatterjee, R. Hartley, "A New Simultaneous Circuit Partitioning and Chip Placement Approach Based on Simulated Annealing," *Proceedings of Design Automation Conference*, pp 36 - 39, 1990.

[4] C. Chu, D. Wong, "A Matrix Synthesis Approach to Thermal Placement," *International Symposium on Physical Design*, Napa Valley, California, pp 163 - 168, 1997.

[5] J. P. Cohoon, W. Paris, "Genetic Placement," *Proceedings IEEE International Conference on Computer-Aided Design*, pp 422 - 425, 1986.

[6] S. Kirkpatrick, C. D. Gellat, M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220 (4598), pp 671 - 680, 13 May 1983.

[7] R. M. Kling, "Placement by Simulated Evolution," *Master's Thesis*, Coordinated Science Laboratory, College of Engineering, University of Illinois at Urbana-Champaign, 1987.

[8] B. Lall, A. Ortega, H. Kabir, "Thermal Design Rules for Electronic Components on Conducting Boards in Passively Cooled Enclosures," *InterSociety Conference on Thermal Phenomena*, Washington DC, pp 50 - 61, 1994.

[9] D. Li, J. D. Carothers, "MCM Routing Algorithm Based on a Compatibility Graph Approach", *Electronics Letters*, Vol. 32 (1), 1996, pp 5 - 6. *Ph.D. Dissertation*, Computer Systems Design Laboratory, Department of Electrical and Computer Engineering, University of Arizona, 1995.

[10] N. Metropolis, A. Rosenbluth, M. Rosenbluth, "Equation of State Calculations by Fast Computing Machines," *Journal of Chemistry and Physics*, pp 1087 - 1092, 1953.

[11] M. Osterman, M. Pecht, "Placement for Reliability and Routability of Convectively Cooled PWBs," *IEEE Transactions on CAD*, Vol. 9 (7), pp 734 - 744, July 1990.

[12] N. Sherwani, *Algorithms for VLSI Physical Design Automation*, Kluwer Academic Publishing, Massachusetts, 1995.

[13] M. C. Tang, J. D. Carothers, "Consideration of Thermal Constraints During Multichip Module Placement," *Electronic Letters*, Vol. 33, no. 12, pp 1043 - 1045, 1997.

[14] G. Wippler, M. Wiesal, D. Mlynski, "A Combined Force and Cut Algorithm for Computer Logic Graphs," *Proceedings 19th Automation Conference*, pp 671 - 677, 1982.

VI. ACKNOWLEDGMENTS

The authors would like to thank Prof. Jerzy Rozenblit and Mr. Kusnadi for their contributions.