Journal of
Low Power Electronics
Vol. 3, 1–15, 2007

Simulated Annealing Based Temperature Aware Floorplanning

Yongkui Han* and Israel Koren

Department of ECE, University of Massachusetts, Amherst, MA 01002, USA

(Received: 27 March 2007; Accepted: 18 June 2007)

Power density of microprocessors is increasing with every new process generation resulting in higher maximum chip temperatures. The high temperature of the chip greatly affects its reliability, raises the leakage power consumed to unprecedented levels, and makes cooling solutions significantly more expensive. The maximum temperature of a block in a chip depends not only on its own power density, but also on the power density of the adjacent blocks. Consequently, the placement of architectural blocks, or a particular floorplan selected for a given chip, can considerably affect the maximum temperature of the chip. This paper analyzes the impact of floorplanning on the maximum temperature by using as examples the Alpha and Pentium Pro microprocessors. We show that the difference between the maximum temperatures of two different floorplans can be as high as 37 °C. We have modified a simulated annealing-based floorplanning tool to include temperature as an objective for block placement to reduce the hot spot temperature. We show that it is possible to find a floorplan that can reduce the maximum temperature of a chip by up to 21 °C compared to the original floorplan while maintaining comparable performance.

Keywords: Temperature, Floorplanning, Hot Spot, Thermal Simulation, Area, Performance.

1. INTRODUCTION

Power density is increasing in each generation of microprocessors, since feature size and frequency are scaling faster than the operating voltage. Power density directly translates into heat, and consequently processors are getting hotter. For example, Pentium 4 chips generate more heat than a kitchen hotplate and the company's projections show that the heat generated by its processors will increase sharply in the coming years, approaching that of a nuclear reactor, unless solutions to this problem can be found.[1]

In order to keep the chip temperature below a certain limit, the heat generated by the processor must be removed from the die. Since the cost of removing heat is increasing at about the same rate as power density, reducing the maximum temperature in the chip can reduce the cost of the cooling system, which constitutes a major component of the overall cost.

The operating temperature has a significant impact on microprocessor design. At higher temperatures, transistors work slower because of the degradation of carrier mobility. The interconnect metal resistivity is also higher at higher temperatures, causing longer interconnect RC delays, and thereby performance degradation. In addition, leakage power can be orders of magnitude greater at higher temperatures. Reliability is also strongly related to temperature, and increasing the temperature will exponentially decrease the lifetime of the chip. Last, but not least, a higher operating temperature increases the cost of cooling solutions. In summary, higher operating temperatures have a significant negative impact on performance, power consumption, reliability, and cooling cost.

With increases in power density of digital circuits, heat dissipation is fast becoming a limiting factor in microprocessor design. Recently temperature aware designs have been proposed and studied.[2] Skadron et al. propose temperature aware microarchitectures.[3,4] They have developed the HotSpot software,[5] which is a tool to calculate the temperature distribution among different blocks on the CPU chip.

Chu et al. propose a combinatorial optimization problem to model the thermal placement problem,[6] and present several provably good approximation algorithms. Our temperature aware floorplanning technique is different from Ref. [6] in three ways. First, they focus on a theoretical and simplified floorplanning problem where all blocks have the same size. Second, instead of using temperature as an

*Author to whom correspondence should be addressed.
Email: yhan@ecs.umass.edu

objective, they use the sum of power numbers of a partial floorplan. In contrast, we calculate the real temperature difference between different floorplans using the HotSpot software. Third, they use randomly generated power numbers in their experiments, while we use simulated power numbers for SPEC2000 benchmarks in our experiments.

Hung et al. study thermal-aware floorplanning using genetic algorithms.[7,8] They demonstrate that their combined area and thermal optimization technique decreases the peak temperature while generating floorplans with area comparable to that achieved by traditional area-oriented techniques. They do not, however, explore the performance impact of their algorithms. Our temperature aware floorplanning is based on the simulated annealing technique and evaluates the performance impact of different floorplans using an interconnection model. The studies in Refs. [7, 8] are focused on a lower circuit level, while our studies focus on the architectural level. We use real processors and their corresponding power densities to show the impact of different floorplans on temperature, while they use randomly generated power numbers in their experimental simulations. In Ref. [9], Hung et al. present a thermal-aware floorplanner for 3D architectures, which is a natural extension of 2D temperature aware floorplanning.

In Ref. [10], Sankaranarayanan et al. study thermal aware floorplanning and show that significant peak temperature reduction can be achieved by managing lateral heat spreading through floorplanning. Their results show that a thermal-aware floorplanning scheme is very competitive with dynamic temperature management techniques. Their method is similar to our method (both are based on simulated annealing), the difference is in the calculation method in each step of the simulated annealing. They calculate the temperature and the wire delay in cycles, while we use approximations (weighted wire length for performance and heat diffusion measure for temperature) to speedup the simulated annealing process. Therefore, our method is more efficient in generating temperature aware floorplans. They focus on the Alpha processor only, while we do experiments with the Pentium Pro processor and the Core 2 Duo processor as well.

In Ref. [11], Healy et al. implemented a multi-objective floorplanning algorithm for 2D and 3D ICs. It combines linear programming and simulated annealing to obtain high-quality solutions. The access latency on each interconnect is calculated based on the floorplan, and the SimpleScalar simulator[12] is used to evaluate the performance of different floorplans. The module's (dynamic) power is independent of floorplanning, but the thermal and leakage interdependence is considered in their floorplanning method. Their floorplanning method is more complex than ours, and as a result, their method runs much slower. The typical runtime of their method is about 6 hours, while our method can finish in 10 minutes.

In this paper we study the impact of floorplanning on the temperature of a chip, and provide detailed experimental results to demonstrate it. The insights learned from this paper can also be applied to devise new techniques at the architectural level. Architectural components that often affect the maximum temperature in the chip, e.g., the register file, could be banked/partitioned[14] to allow more flexibility in placement and reduced power density. We have seen similar trends to reduce power consumption in caches.[15,16] Alternatively, one might consider architectures where some of these components are replicated and the associated activity is distributed in a temperature-conscious way. Additional provisions can be added at the circuit level to reduce the power density of such components.

Our contributions in this paper are as follows:

(1) We propose a simulated-annealing-based temperature aware floorplanning, through which we can generate a floorplan that reduces the maximum temperature while maintaining comparable performance. Extensive experiments have been performed with several processors: Alpha, Pentium Pro, and Core 2 Duo.

(2) We use the weighted wire length as an easy to calculate and convenient to use approximate measure for chip performance.

(3) We use a heat diffusion measure as an approximation of the temperature, to achieve a considerable reduction in the complexity of computing the temperature while still producing good results.

This paper extends our previous workshop presentation.[13] Our floorplanning method is extended with the use of weighted wire length allowing a more accurate modeling of the chip performance. We extend our temperature aware floorplanning to multi-core microprocessors, since they are expected to be the next generation microprocessors in the near future. These extensions make our temperature aware floorplanning a more complete framework.

The rest of the paper is organized as follows. In Section 2, we demonstrate the temperature benefits of different floorplans. In Section 3, we present our temperature aware floorplanning technique and describe its implementation based on the Parquet software. The experimental results for an Alpha microprocessor, the Pentium Pro microprocessor, and the Core 2 Duo microprocessor are presented in Sections 4, 5, and 6, respectively. Conclusions are presented in Section 7.

2. PRELIMINARIES

The HotSpot software developed at the University of Virginia is a tool that models the temperature of microprocessor chips. HotSpot allows the user to specify a processor floorplan with its functional units. From this floorplan, it creates an equivalent circuit model that represents heat transfer in a processor die with specified thermal packaging. The HotSpot software[5] accounts for many important effects of the thermal interface material between the die

Table I. HotSpot model configuration.

Parameter	Alpha	Pentium Pro	Core 2 Duo	Description (unit)
t_chip	0.5	0.5	0.5	Chip thickness (mm)
c_convec	140.4	140.4	140.4	Convection capacitance (J/K)
r_convec	0.1	0.8	0.03	Convection resistance (K/W)
s_sink	60	60	72	Heat sink side (mm)
t_sink	6.9	6.9	6.9	Heat sink thickness (mm)
s_spreader	30	30	30	Heat spreader side (mm)
t_spreader	1.0	1.0	1.0	Heat spreader thickness (mm)
t_interface	0.075	0.075	0.05	Interface material thickness (mm)
ambient	40	40	40	Ambient temperature (°C)

and heat spreader and has been validated against a test chip. The parameters of the HotSpot model are shown in Table I. The same Alpha processor (0.13 μm technology) floorplan used by Skadron et al.[4] is used in our experiments (shown in Fig. 1) as well as the same power numbers of the functional blocks.

The temperature of each processor block for the *gcc* benchmark is shown in Figure 2. We do not show the temperature of the L2 cache in the figure because the L2 cache has a considerably lower temperature than the other blocks in the processor core. The power density of each block is shown in Figure 3. The block with the maximum temperature is marked with a pattern different from that used for the block with the minimum temperature. Similar distinct patterns are used for the power densities in Figure 3. The block with the maximum temperature in the chip is the integer register file *IntReg*. Its temperature is 120 °C, and it has the highest power density, 2.798 Watt/mm^2.

Usually the block with a higher power density also has a higher temperature, but this is not always the case. The temperature of a block in a chip depends not only on its power density but also on the power density of the adjacent blocks.

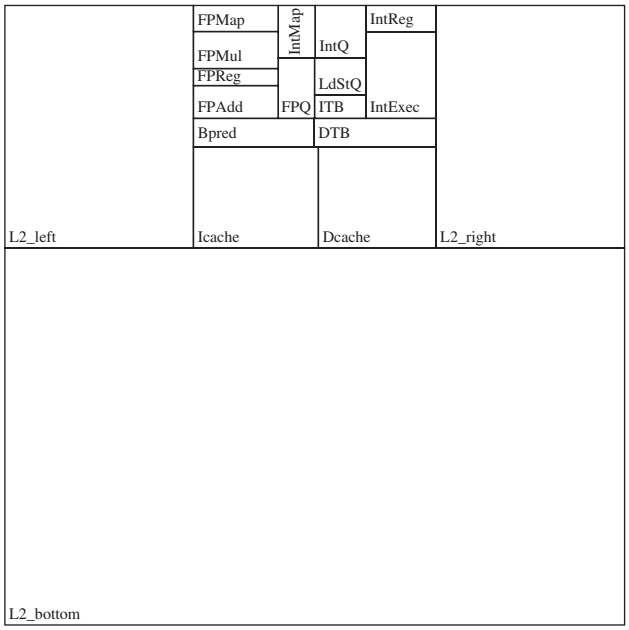


Fig. 1. The original Alpha floorplan.

The placement of a block has therefore a considerable impact on its temperature and this has motivated us to study temperature aware floorplanning.

2.1. Maximum Temperature for the SPEC2000 Benchmarks

We show the maximum temperature for 24 SPEC2000 benchmarks in Figure 4. We can see that for 12 out of the 24 benchmarks the maximum temperature of the chip is higher than 100 °C, for 8 of them the temperature exceeds 110 °C, and for 2 of them it exceeds 120 °C.

When we take a look at the hottest block in the chip, we find that it is *IntReg* for almost all SPEC2000 benchmarks except for *applu*, *lucas*, and *mgrid*. *FPReg* is the hottest block for *applu* and *FPAdd* is the hottest block for *lucas* and *mgrid*. However, their maximum temperatures are not high (67.2 °C, 59.1 °C and 75 °C, respectively). Since the temperature distribution among the blocks of the chip for these SPEC benchmarks is similar, we select as the representative benchmark in our experiments the *gcc* benchmark, which has a maximum temperature of 120 °C.

2.2. A Manually Generated Floorplan

We first tried to manually modify the Alpha floorplan to lower the maximum temperature of the chip. Figure 5 is

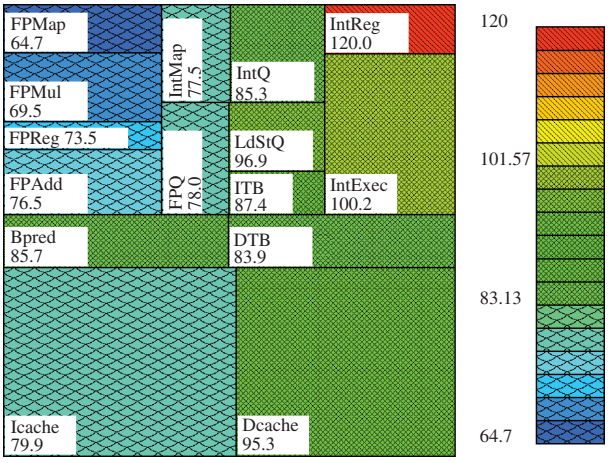


Fig. 2. Temperatures (in °C) of blocks in the core area for the *gcc* benchmark.

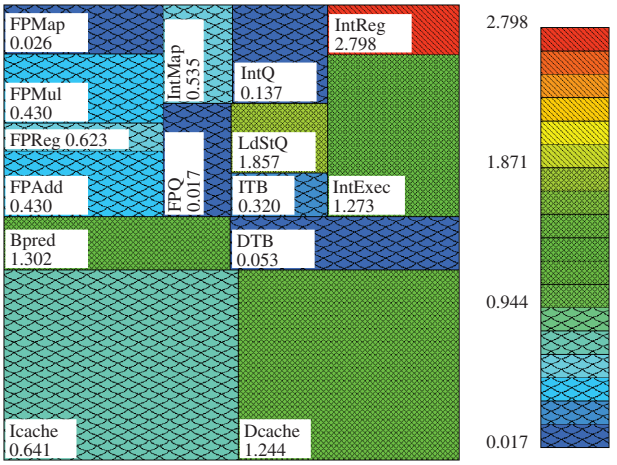


Fig. 3. Power densities (in Watt/mm^2) of blocks in the core area for the *gcc* benchmark.

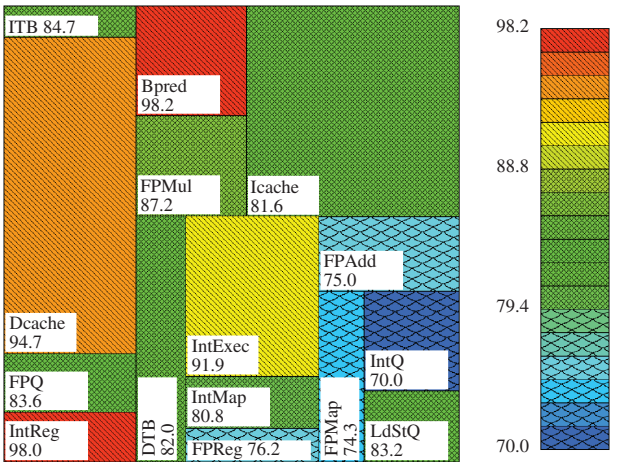


Fig. 5. The block temperatures (in °C) for the manually generated Alpha floorplan.

a new floorplan, in which we put blocks with high power density next to blocks with low power density, while maintaining the area of blocks and changing the aspect ratio of the blocks as little as possible. We show the resulting temperature of each block for the *gcc* benchmark in Figure 5. The maximum temperature and the temperature reductions for the new floorplan for SPEC2000 benchmarks are shown in Figure 6. We achieve a reduction of more than 20 °C for many benchmarks, and an average reduction of 11 °C.

For the new floorplan, only the maximum temperature for the *vortex* benchmark is a little higher than 100 °C. The maximum temperature for all other benchmarks has been reduced to below 100 °C. This is a significant improvement in temperature.

For three benchmarks, *applu*, *lucas*, and *mgrid*, the maximum temperature has increased. Their hottest blocks in the chip are *FPReg*, *FPAdd*, and *FPAdd*, respectively. Notice, however, that for these benchmarks, the maximum temperature is low, so a small increase in the maximum temperature will still keep the chip reasonably cool. Their maximum temperatures are 67 °C, 65 °C, and 91 °C, respectively.

These results motivated us to further manipulate the floorplan and modify the placement of the blocks to reduce the maximum temperature in the chip.

Any block replacement can, however, affect the performance of the chip. The performance of the chip depends on the wire length of the various interconnections among the blocks. Since only approximate values of inter-block wires' lengths are usually available during floorplanning, and in order to obtain a simple metric to be used in the search for an optimal floorplan, the total wire length among all blocks is traditionally used as a measure for performance during floorplanning. Clearly, the total wire length can not accurately reflect the individual signal delays among the various blocks. To increase the relevance of the total wire length metric, it is common to assign higher weights to wires between two blocks which carry timing critical signals, making these blocks more likely to be placed adjacently. Still, the total wire length can, at best, serve as only a first order approximation for the chip performance. In the absence of exact information regarding the criticality of individual wires in the floorplans which we have analyzed, we first use the unweighted total wire length as our measure for performance. In Section 4, we extend our experiments and use a weighted total wire length metric.

2.3. Wire Length Overhead

Since the exact number of wires between any two blocks in the Alpha chip has not been available to us, we used instead the estimated interconnect matrix shown in Table II. This matrix focuses on data signals and ignores control signals,

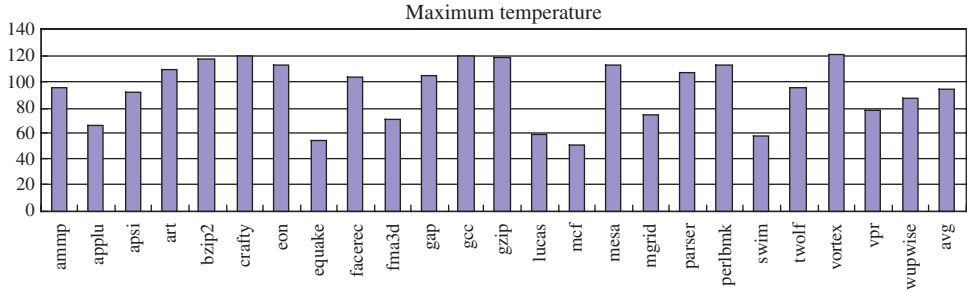


Fig. 4. Maximum temperature (in °C) for the original Alpha floorplan for SPEC2000 benchmarks.

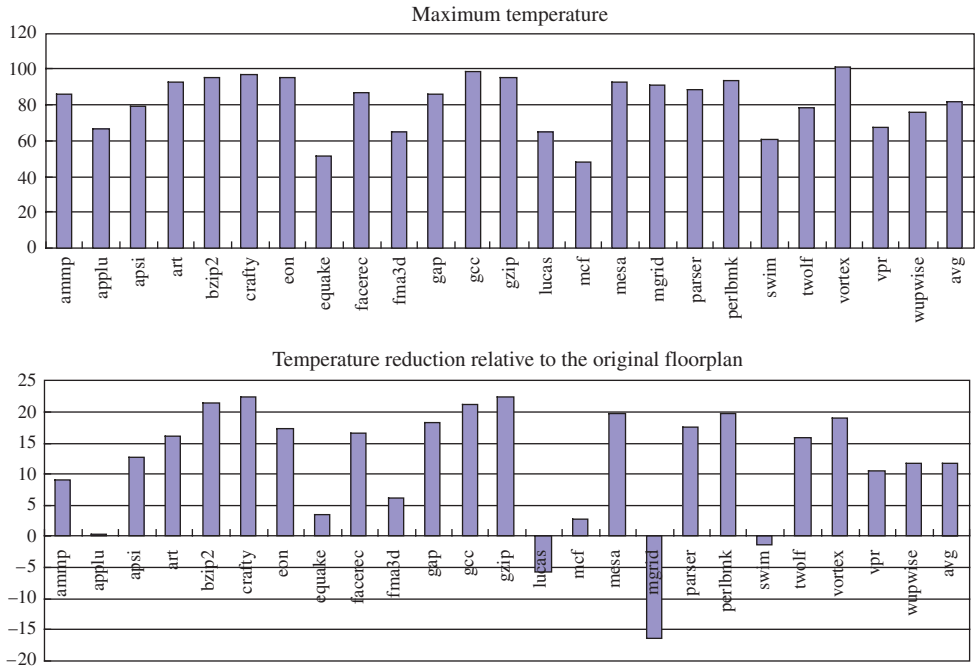


Fig. 6. Maximum temperatures and temperature reductions (in °C) for the manually generated Alpha floorplan.

Table II. An interconnect matrix.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
L2_left	1	0	128	128	128	128	0	0	0	0	0	0	0	0	0	0	0	0	0
L2_bottom	2	128	0	128	128	128	0	0	0	0	0	0	0	0	0	0	0	0	0
L2_right	3	128	128	0	128	128	0	0	0	0	0	0	0	0	0	0	0	0	0
Icache	4	128	128	128	0	0	128	0	0	0	0	128	128	0	0	0	0	0	128
Dcache	5	128	128	128	0	0	0	64	0	0	0	0	0	0	0	0	0	128	0
Bpred	6	0	0	0	128	0	0	0	0	0	0	0	0	0	0	0	0	0	128
DTB	7	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0
FPAdd	8	0	0	0	0	0	0	0	0	128	0	0	0	0	0	0	128	128	0
FPReg	9	0	0	0	0	0	0	0	128	0	128	0	0	0	0	0	128	128	0
FPMul	10	0	0	0	0	0	0	0	0	128	0	0	0	0	0	0	128	128	0
FPMap	11	0	0	0	128	0	0	0	0	0	0	0	0	0	0	0	128	0	0
IntMap	12	0	0	0	128	0	0	0	0	0	0	0	0	128	0	0	0	0	0
IntQ	13	0	0	0	0	0	0	0	0	0	0	0	128	0	128	128	0	0	0
IntReg	14	0	0	0	0	0	0	0	0	0	0	0	0	128	0	192	0	128	0
IntExec	15	0	0	0	0	0	0	0	0	0	0	0	0	128	192	0	0	128	0
FPQ	16	0	0	0	0	0	0	0	128	128	128	128	0	0	0	0	0	0	0
LdStQ	17	0	0	0	0	128	0	0	128	128	128	0	0	0	128	128	0	0	0
ITB	18	0	0	0	128	0	128	0	0	0	0	0	0	0	0	0	0	0	0

Table III. Core area, wire length and maximum temperature for Alpha floorplans.

Floorplan	Core area (mm^2)	Increase (%)	Total wire length (m)	Increase (%)	Temp (°C)	Reduction (°C)
Original	38.76	0	17.93	0	120.0	0
Manual	38.76	0	23.21	29.45	98.2	21.8
Rotated	38.76	0	18.32	2.18	103.1	16.9
Low-temp	39.16	1.02	24.77	38.15	95.2	24.8
Wire-temp	39.55	2.04	18.05	0.67	98.9	21.1
Short-wire	39.66	2.31	17.20	−4.07	120.1	−0.1
High-temp	39.55	2.04	19.07	6.36	132.3	−12.3

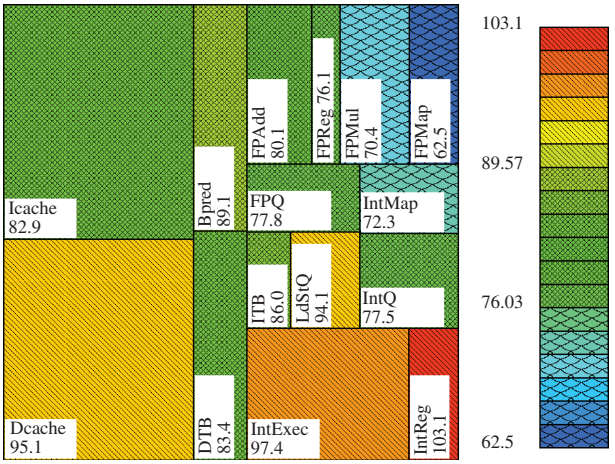


Fig. 7. The block temperature (in °C) for the rotated core Alpha floorplan.

and is used here for illustration purposes only. The columns (and rows) of the interconnect matrix are in the order *L2_left, L2_bottom, L2_right, Icache, Dcache, Bpred, DTB, FPAdd, FPReg, FPMul, FPMap, IntMap, IntQ, IntReg, IntExec, FPQ, LdStQ, ITB*.

To calculate the wire length we adopt the widely used method of HPWL (Half Perimeter Wire Length), i.e., the wire length between two blocks is calculated as follows:

$$WireLength = |x1 - x2| + |y1 - y2|$$

where $x1, y1, x2, y2$ are the coordinates of the centers of the blocks.

The increase in wire length of the manually drawn Alpha floorplan is 29% (see Table III). Such a wire length increase is excessive and the question arises whether any attempt to reduce the maximum temperature will result in a substantial increase in wire length. In the next section, we will show that this is not necessarily the case.

2.4. The Rotated Alpha Floorplan

Since the CPU core is surrounded by the low power density portions of the L2 cache, we experimented with a 90° rotation of the core, a simple floorplan modification that is expected to result in a small wire length increase. The rotated floorplan and the resulting temperatures for the *gcc* benchmark are shown in Figure 7. The new maximum temperature in the chip and the reduction in maximum temperature for all the 24 SPEC benchmarks are given in Figure 8. The rotated floorplan reduces the maximum temperature of the chip for the *gcc* benchmark by 16.1 °C with a wire length increase of only 2.18% (see Table III). For most benchmarks, we obtain considerable temperature improvements. We can achieve an average reduction of 9 °C for the SPEC2000 benchmarks (which is smaller than the 21.8 °C reduction achieved by the manually generated floorplan).

Manually searching for a floorplan that may further reduce the maximum temperature with a low wire length overhead is time-consuming and inefficient. We decided therefore to modify an existing floorplanning tool to allow us to generate temperature aware floorplans.

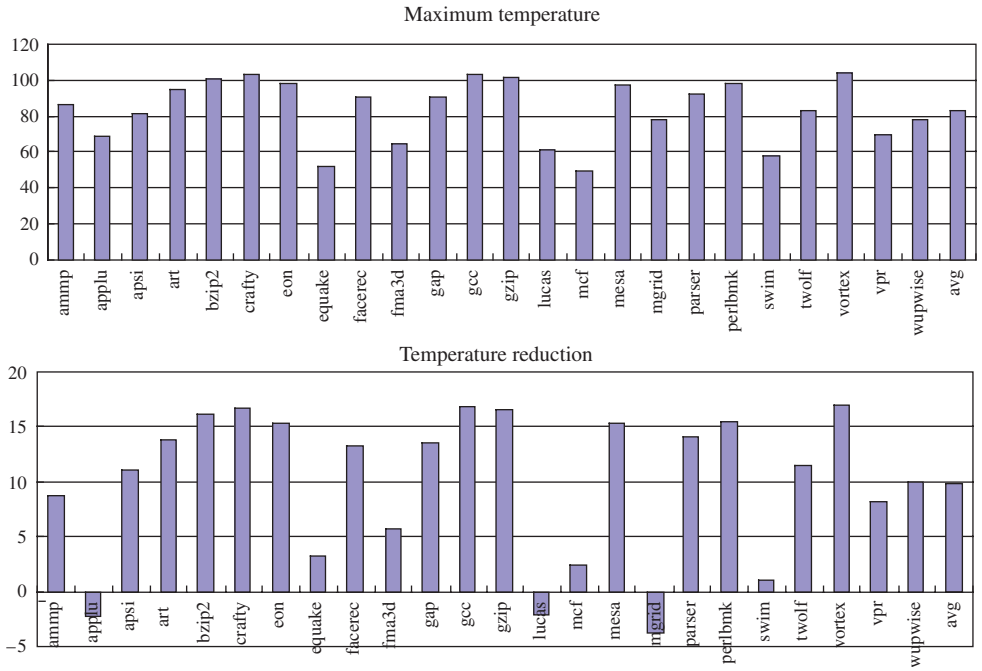


Fig. 8. Maximum temperatures and temperature reductions (in °C) for the rotated core Alpha floorplan.

3. FLOORPLANNING WITH A TEMPERATURE OBJECTIVE

3.1. Parquet Floorplanner

For our purposes, we have selected Parquet,[18] which is a floorplanning tool developed at the University of Michigan. The Parquet floorplanner is a fixed-outline, hierarchical design package and is based on the widely used sequence-pair representation. It is intended to solve multi-objective problems (area and wire length) using the simulated annealing algorithm.

We allow all the blocks of the chip to be "soft" blocks, that is, their aspect ratio can change (in a controlled manner) in each movement but their area is fixed. We next describe our temperature aware floorplanning.

3.2. Problem Description

The temperature aware floorplanning problem is that of placing n rectangular modules in the chip area satisfying the following constraints:
(1) Module i has a fixed area A_i but its height h_i and width w_i can be changed so that $h_i * w_i = A_i$. The aspect ratio of module i, h_i/w_i, must be limited to the range $r_i \leq h_i/w_i \leq s_i$ $(i = 1, ..., n)$.
(2) The wiring length is calculated using an interconnect matrix $C_{n*n} = [C_{ij}]$, where C_{ij} is the number of wires connecting modules i and j.
(3) The chip area is $A = H * W$, where H and W are the height and width of the chip, respectively. The chip aspect ratio is also constrained to a given range $R \leq H/W \leq S$.
(4) Module i has a known power consumption P_i and power density $d_i = P_i/A_i$ $(i = 1, ..., n)$.
(5) The objectives of the floorplanning process are: low chip area A, low total wire length L, and low maximum temperature T.

3.3. Objective Function

In Parquet, the global objective is to minimize a linear combination of the total area and the total wire length. We add the temperature to the original objective function as follows:

$$Obj = C_A * A + C_L * L + C_T * T$$

where C_A, C_L and C_T are the weights of the area, the wire length and the maximum temperature in the chip, respectively.

The Parquet software performs millions of movements in each simulated annealing run. It is therefore, prohibitively time-consuming to calculate the steady-state temperature of the blocks for each movement since this requires the construction of a new thermal resistance matrix. We need therefore to find an approximate measure to represent the maximum temperature in the chip.

3.4. Maximum Temperature Estimation

An estimate for the maximum temperature should have the following properties:
(1) Reflect the goodness of a floorplan with respect to the maximum temperature, i.e., a floorplan with a lower maximum temperature must have a lower value than a floorplan with a higher maximum temperature.
(2) Should be easy to calculate.

The essence of temperature interaction between adjacent blocks is the *heat diffusion* between them. Thus, the heat diffusion between adjacent blocks can serve as a good approximation for the temperature.

3.5. Heat Diffusion Measure

The heat diffusion between two adjacent blocks is proportional to their temperature difference and the length of the shared block boundary between them.

$$H(T1, T2) = (T1 - T2) * shared\_length$$

where H is the heat diffusion, $T1$ and $T2$ are the temperatures of the two blocks, and *shared_length* is the length of their shared boundary.

Since we do not know the exact temperatures of the blocks at each simulation step, we can not use them to calculate the heat diffusion directly, and we must replace them by estimates. To this end, consider an isolated block whose temperature can be calculated as:

$$T = P * R_t = P * \delta/(k * A) = (P/A) * (\delta/k) = (\delta/k) * d$$

where T is the steady-state temperature, P is the power consumption, R_t is the thermal resistance between the block and the environment, δ is the thickness of the chip, k is the thermal conductivity of the material, A is the area of the block, and d is its power density.

This expression shows that the temperature of an isolated block depends linearly on its power density. We will, therefore, use the power density of a block as an estimate of its temperature.

Thus, we define the following measure as an approximation for the *heat diffusion* between two adjacent blocks:

$$H(d_i, d_j) = (d_i - d_j) * shared\_length$$

where d_i and d_j are the power densities of the two blocks. For each block, its total *heat diffusion* will be:

$$H_T(d) = \Sigma_i H(d, d_i)$$

over all its neighbors.

We calculate the heat diffusion of the chip as an approximation for the chip temperature, but in this calculation we do not consider all the blocks. If the power density of a block is very small, it is impossible for it to become the hottest block and as a result, its position is not important.

We only need to care about the heat diffusion of blocks which may become the hottest block in the chip.

To select the *possibly-hot* blocks which may become the hottest blocks, we pick the top m $(1 \leq m \leq n)$ blocks with the highest power density.

If we take too many blocks into consideration, the final result may not place the block with the highest power density and the block with the lowest power density next to each other. If we take too few blocks into consideration, e.g., only the block with the highest power density, then other blocks with high power density can become the hottest block in the chip. It is important therefore to determine the number of *possibly-hot* blocks carefully.

In our experiments, we tried the selection of the top 1, 2, 3, or 4 blocks with the highest power density as *possibly-hot* blocks. We found that the selection of 2 *possibly-hot* blocks produced the best results for the Alpha processor.

We calculated the heat diffusion H_T of all the selected *possibly-hot* blocks, and then calculated the total *thermal diffusion* D_T which is defined as the sum of the heat diffusions of all *possibly-hot* blocks:

$$D_T = \Sigma H_T(d)$$

for all *possibly-hot* blocks.

D_T was used as an approximation for the maximum temperature in our experiments. Thus, the final objective function for the Parquet floorplanner is:

$$Obj = C_A * A + C_L * L - C_D * D_T$$

where C_D is the weight of the thermal diffusion. C_D has a negative sign because we want to maximize the *thermal diffusion* D_T.

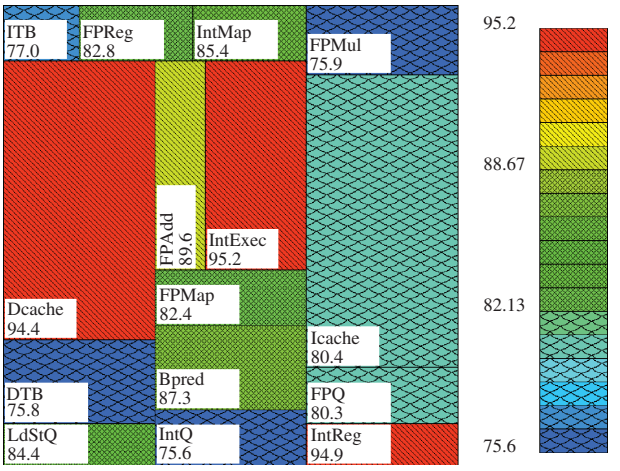


Fig. 9. **Low-temp**: a floorplan with low maximum temperature.

4. EXPERIMENTAL RESULTS

4.1. Parquet Generated Floorplans

Each run of Parquet takes about 5 seconds, and we ran this tool thousands of times and then selected the best result. For the Alpha floorplan, we only manipulated the positions of the blocks in the core area, while keeping that of the L2 cache fixed. Since the Parquet generated floorplan may have some unused space, the area of some blocks is increased to fill the unused space. Such an increase in area may affect the temperature of the chip. However, the increase in area that we have observed has been very small, usually less than 1%, so its impact is negligible.

In what follows we show several Parquet generated floorplans. **Low-temp** (Fig. 9) is a floorplan obtained when we

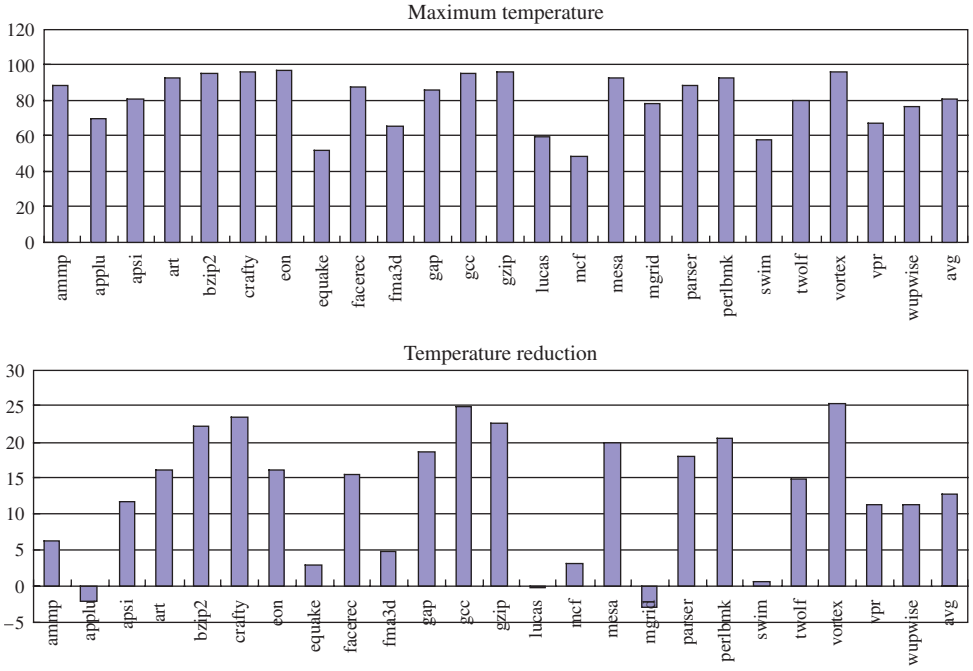


Fig. 10. Maximum temperature and temperature reductions (in °C) of **Low-temp** floorplan for SPEC2000 benchmarks.

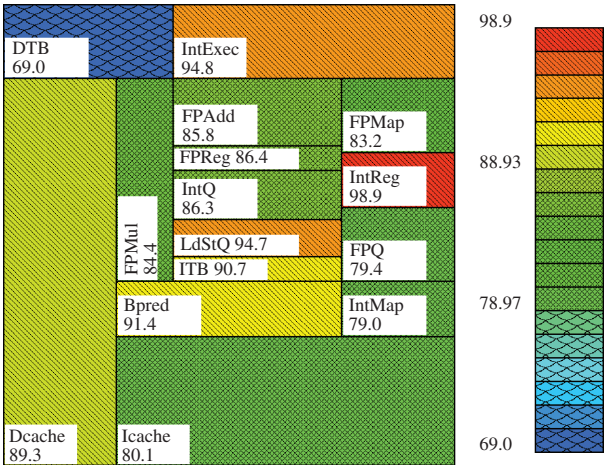


Fig. 11. **Wire-temp**: a floorplan with both short wire length and low maximum temperature.

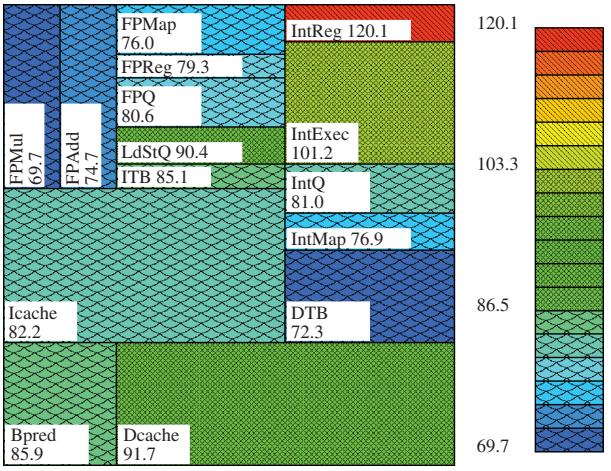


Fig. 13. **Short-wire**: a floorplan with short wire length.

optimize area and temperature and ignore the wire length ($C_A = 0.4$, $C_L = 0$, $C_D = 0.6$). **Wire-temp** (Fig. 11) is a floorplan which takes temperature, area and wire length into account ($C_A = 0.3$, $C_L = 0.4$, $C_D = 0.3$). **Short-wire** (Fig. 13) is a floorplan which optimizes only the wire length and area but ignores the temperature ($C_A = 0.4$, $C_L = 0.6$, $C_D = 0$). **High-temp** (Fig. 14) is a result of an attempt to generate a floorplan with the highest maximum temperature.

The maximum temperature of the **Low-temp** floorplan for the *gcc* benchmark is 95 °C, which is 25 °C lower than that of the original floorplan. The steady-state temperatures for SPEC2000 benchmark are shown in Figure 10. We can see that the average maximum temperature has decreased from 94 °C to 81 °C, the average reduction of the maximum temperature is thus 13 °C, which is 2 °C better than that obtained by the manually generated floorplan. The maximum temperatures for all the benchmarks have been reduced to below 97 °C. Although the temperatures of the *applu*, *lucas*, and *mgrid* benchmarks have increased by several degrees, their maximum temperatures are still below 80 °C.

The maximum temperature of the **Wire-temp** floorplan for the *gcc* benchmark is 99 °C, which is 21 °C lower than the original one. The maximum temperatures for the 24 SPEC2000 benchmarks are shown in Figure 12. We can see that the average maximum temperature has decreased from 94 °C to 82 °C, i.e., a reduction of 12 °C. The maximum

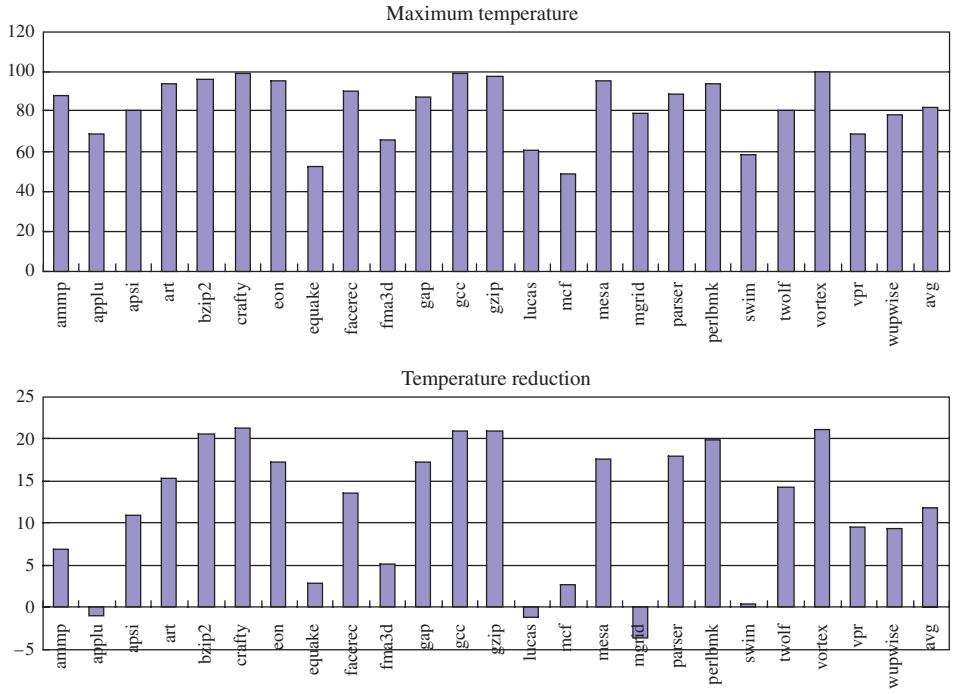


Fig. 12. Maximum temperatures and temperature reductions (in °C) of **Wire-temp** floorplan for SPEC2000 benchmarks.

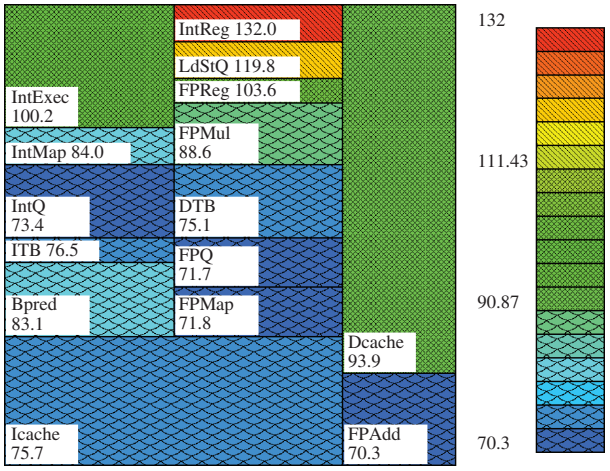


Fig. 14. **High-temp**: a floorplan with high maximum temperature.

temperatures for all the benchmarks have been reduced to below 100 °C.

The maximum temperature of the **Short-wire** floorplan (see Fig. 13) for the *gcc* benchmark is 120.1 °C, which is almost the same as that of the original floorplan. This demonstrates that in order to get a lower chip temperature, we must include temperature as an objective in the floorplanning tool.

If we do not include the temperature in the objective function, Parquet may generate floorplans with very high maximum temperatures. We show such a floorplan **High-temp** in Figure 14. This floorplan has a maximum temperature of 132 °C, which is 12 °C larger than the original one and 37 °C larger than the floorplan **Low-temp** with the lowest maximum temperature.

4.2. Area and Wire Length Overhead

The area increase, wire length increase, and temperature reduction for all the generated floorplans are listed in Table III. The area increase of all the floorplans is very small, less than 1%.

The floorplan **Low-temp** reduces the maximum temperature by 25 °C with a wire length increase of 38%. If this is considered excessive, we can instead select the floorplan **Wire-temp** which reduces the maximum temperature by 21 °C while keeping the wire length practically unchanged. The small reduction in wire length for the floorplan **Short-wire** shown in Table III does not necessarily mean that this floorplan is better than the original one with respect to wiring cost but can probably be attributed to the inaccuracies in our wire length calculations.

Table IV. Weights of all interconnections.

Interconnection	Weight
IntExec-IntQ, Bpred-ICache	15
IntReg-IntExec, IntQ-IntReg, IntMap-IntQ ITB-Icache, DTB-Dcache, LdStQ-Dcache	10
FPReg-FPAdd, FPReg-FPMul, FPMap-FPQ FPQ-FPAdd, FPQ-FPMul	5
All others	1

4.3. Weighted Interconnection Matrix

In the previous experiments, we assumed all interconnections to be equally critical for performance. In practice, some interconnections are more critical than others and it is very important to keep certain units adjacent to each other. In our temperature aware floorplanning we can take this fact into account by using weighted interconnects, assigning larger weights to more critical interconnections.

The weighted interconnect matrix used in our experiments is shown in Table V. This matrix assigns larger weights to the following interconnections: *IntReg-IntExec, IntQ-IntReg, IntMap-IntQ, IntExec-IntQ, ITB-Icache, DTB-Dcache, Bpred-Icache, LdStQ-Dcache, FPReg-FPAdd,*

Table V. The weighted interconnect matrix.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
L2_left	1	0	128	128	128	128	0	0	0	0	0	0	0	0	0	0	0	0	0
L2_bottom	2	128	0	128	128	128	0	0	0	0	0	0	0	0	0	0	0	0	0
L2_right	3	128	128	0	128	128	0	0	0	0	0	0	0	0	0	0	0	0	0
Icache	4	128	128	128	0	0	128	0	0	0	0	128	128	0	0	0	0	0	128
Dcache	5	128	128	128	0	0	0	64	0	0	0	0	0	0	0	0	0	128	0
Bpred	6	0	0	0	1920	0	0	0	0	0	0	0	0	0	0	0	0	0	128
DTB	7	0	0	0	0	640	0	0	0	0	0	0	0	0	0	0	0	0	0
FPAdd	8	0	0	0	0	0	0	0	0	128	0	0	0	0	0	0	128	128	0
FPReg	9	0	0	0	0	0	0	0	640	0	640	0	0	0	0	0	128	128	0
FPMul	10	0	0	0	0	0	0	0	0	640	0	0	0	0	0	0	128	128	0
FPMap	11	0	0	0	128	0	0	0	0	0	0	0	0	0	0	0	640	0	0
IntMap	12	0	0	0	128	0	0	0	0	0	0	0	0	1280	0	0	0	0	0
IntQ	13	0	0	0	0	0	0	0	0	0	0	0	1280	0	1280	1920	0	0	0
IntReg	14	0	0	0	0	0	0	0	0	0	0	0	0	1280	0	1960	0	128	0
IntExec	15	0	0	0	0	0	0	0	0	0	0	0	0	1920	1960	0	0	128	0
FPQ	16	0	0	0	0	0	0	0	640	128	640	640	0	0	0	0	0	0	0
LdStQ	17	0	0	0	0	1280	0	0	128	128	128	0	0	0	128	128	0	0	0
ITB	18	0	0	0	1280	0	128	0	0	0	0	0	0	0	0	0	0	0	0

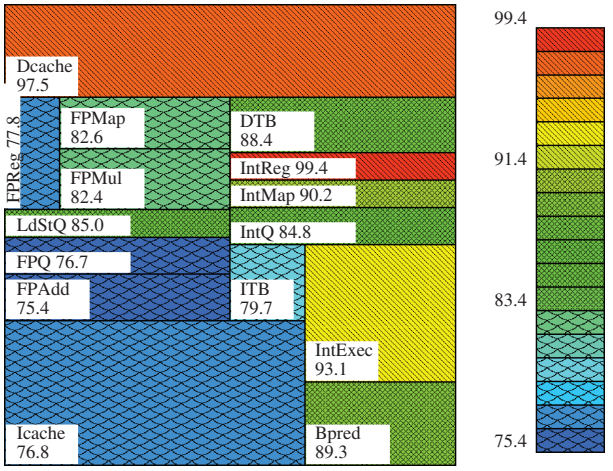


Fig. 15. Weighted: a floorplan with larger weights on critical interconnections.

FPReg-FPMul, FPMap-FPQ, FPQ-FPAdd, and *FPQ-FPMul*. Based on the performance impact of varying the delay on these interconnections,[10] three levels of weights have been assigned in order to demonstrate the capability and flexibility of our temperature aware floorplanning in handling such cases. The assigned weights are shown in Table IV.

4.4. Experimental Results with Weighted Interconnections

Figure 15 shows the Parquet generated floorplan **Weighted** with weighted interconnections. The resulting area increase, total weighted wire length overhead, and temperature reduction are listed in Table VI that provides these values for all the floorplans that have been discussed for the Alpha chip. The **Weighted** floorplan has a comparable wire length to the original one: the total weighted wire length is increased by only 1.70%.

Table VI. Core area, total weighted wire length, and maximum temperature for Alpha floorplans.

Floorplan	Core area (mm^2)	Increase (%)	Total weighted wire length (m)	Increase (%)	Temp (°C)	Reduction (°C)
Original	38.76	0	47.30	0	120.0	0
Manual	38.76	0	80.40	70.00	98.2	21.8
Rotated	38.76	0	47.68	0.82	103.1	16.9
Low-temp	39.16	1.02	82.18	73.76	95.2	24.8
Wire-temp	39.55	2.04	53.50	13.12	98.9	21.1
Short-wire	39.66	2.31	44.45	−6.02	120.1	−0.1
High-temp	39.55	2.04	53.79	13.73	132.3	−12.3
Weighted	39.56	2.06	48.10	1.70	99.4	20.6

We achieve considerable peak temperature reductions in the generated floorplan. The floorplan **Weighted** reduces the maximum temperature by 20.6 °C. We can see from Table VI that our temperature aware floorplanning is able to find a floorplan with both low peak temperature and comparable total weighted wire length.

The maximum temperature of the **Weighted** floorplan for the *gcc* benchmark is 99.4 °C, which is 20.6 °C lower than the original one. The maximum temperatures for the SPEC2000 benchmarks are shown in Figure 16. We can see that the average maximum temperature has decreased from

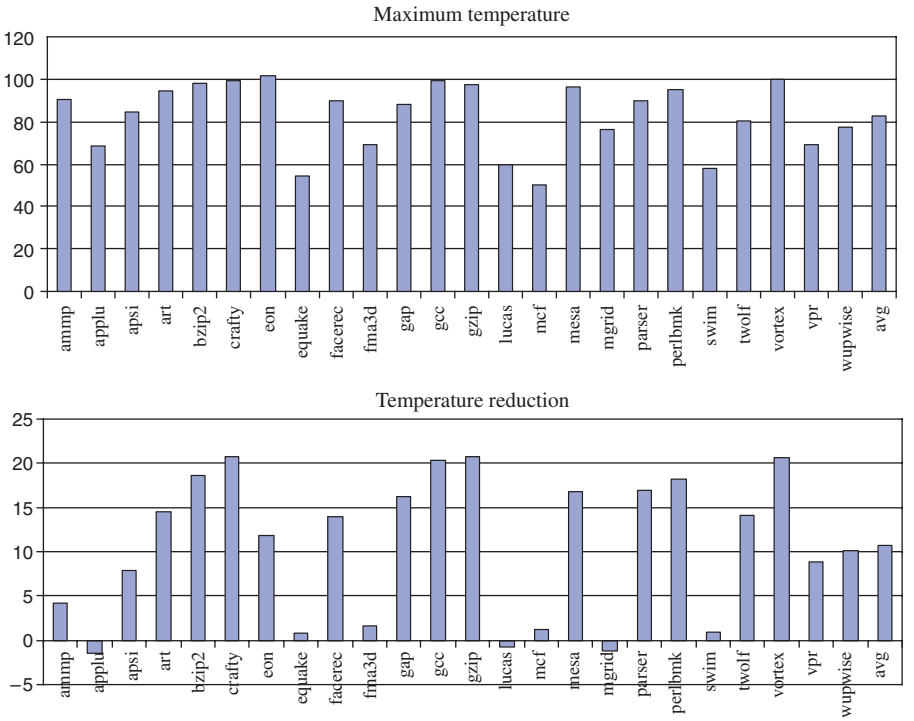


Fig. 16. Maximum temperatures and temperature reductions (in °C) of **Weighted** floorplan for SPEC2000 benchmarks.

94 °C to 83 °C, i.e., an average reduction of 11 °C. The maximum temperatures for all the benchmarks have been reduced to below 100 °C except for the *eon* benchmark (101.4 °C).

In Ref. [10], the authors analyze the floorplan of the same Alpha chip. Their experiments show similar results as ours: a temperature reduction of 22 °C for the *gcc* benchmark. Our approach is much faster than theirs because we use approximations to speedup the simulated annealing process.

From these generated floorplans, we can see that even with weighted wires, our temperature aware floorplanning is able to reduce the peak temperature of the chip with comparable total wire length. This again proves the effectiveness of our temperature aware floorplanning approach.

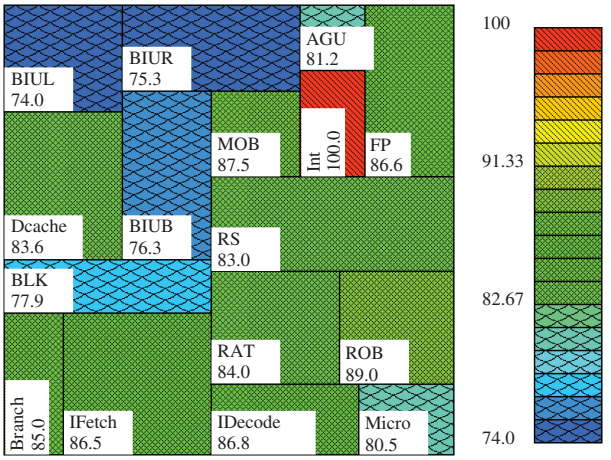


Fig. 17. The original Pentium Pro floorplan.

5. PENTIUM PRO FLOORPLAN

We also performed some experiments on the Pentium Pro processor (0.35 μm technology).[19] Figure 17 shows the original floorplan for this microprocessor[20] and the temperature of each block. The power numbers of the functional blocks used in our experiments are obtained from Ref. [19]. The maximum temperature is 100 °C and it is the temperature of the integer execution unit *Int*.

To calculate the wire length for the Pentium Pro, we used the estimated interconnect matrix shown in Table VII. This matrix focuses on data signals and ignores control signals, and is used here for illustration purposes only. The columns (and rows) of the interconnect matrix are in the order: *Branch, IFetch, IDecode, Micro, RAT, ROB, RS, MOB, Int, FP, Dcache, BIUL, BIUR, BIUB, AGU*, and *BLK*.

We use the modified Parquet software to find a better floorplan with respect to the maximum temperature for the Pentium Pro processor. **Pro-low** (Fig. 18) is a floorplan generated by Parquet. The block with the maximum temperature is still the integer execution unit *Int*, but the maximum temperature has been reduced by 6.3 °C to 93.7 °C.

Pro-high (Fig. 19) is a floorplan that we have generated in an attempt to obtain the highest maximum temperature. The maximum temperature for this floorplan is 110 °C, which is 10 °C higher than that of the original floorplan. Thus, we may conclude that the range for the maximum temperature for the possible floorplans is between 93.7 °C and 110 °C.

The area of the floorplan **Pro-low** has increased by 1.5%, and its total wire length has increased by 13%. These are the penalties we pay for the improvement in the maximum temperature.

The benefits of modifying the block placement to improve the temperature for the Pentium Pro processor are not as impressive as those for the Alpha processor, because the temperature differences between blocks in the Pentium Pro chip are not as large. For the original Pentium Pro floorplan, the maximum temperature is 100 °C, and the minimum temperature is 74 °C. The difference is only

Table VII. An interconnect matrix for Pentium Pro.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Branch	1	0	64	0	0	0	64	64	0	0	0	0	0	0	0	0	0
IFetch	2	0	0	128	0	0	0	0	0	0	0	0	0	0	0	0	0
IDecode	3	0	0	0	64	64	96	0	0	0	0	0	0	0	0	0	0
Micro	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RAT	5	0	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0
ROB	6	0	0	0	0	0	0	64	0	64	64	128	0	0	0	0	0
RS	7	0	0	0	0	0	0	0	0	64	64	0	0	0	0	0	0
MOB	8	0	64	0	0	0	0	0	0	0	0	64	128	0	0	0	0
Int	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FP	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dcache	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIUL	12	0	256	0	0	0	0	0	0	0	0	256	0	0	0	0	0
BIUR	13	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0
BIUB	14	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0
AGU	15	0	0	0	0	0	0	0	64	64	0	0	128	0	0	0	0
BLK	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

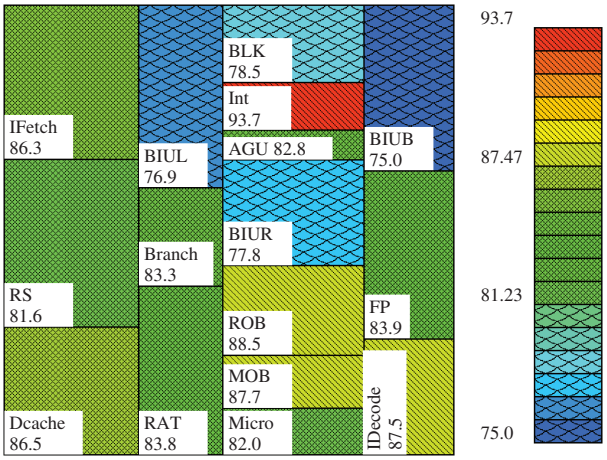


Fig. 18. **Pro-low**: a floorplan with low maximum temperature for Pentium Pro.

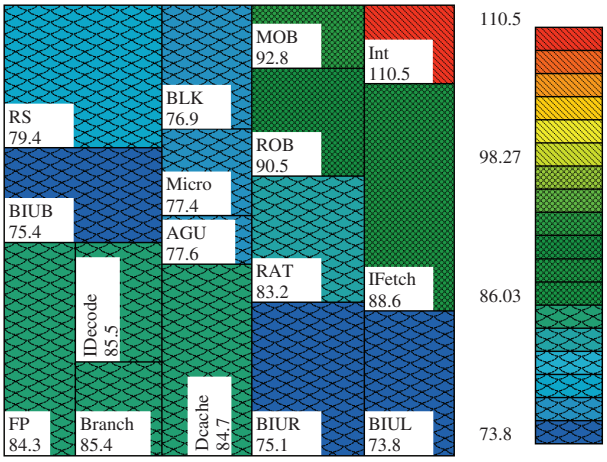


Fig. 19. **Pro-high**: a floorplan with high maximum temperature for Pentium Pro.

26 °C, while the difference for the original Alpha floorplan is about 70 °C (the temperature of the L2 cache is about 50 °C). Still, the difference between the "best" floorplan (**Pro-low** with maximum temperature of 93.7 °C) and the "worst" floorplan (**Pro-high** with maximum temperature of 110.5 °C) shows that even in this case it is worthwhile to consider the temperature when deciding on the floorplan.

6. FLOORPLANS FOR MULTI-CORE PROCESSORS

In recent years, the trend to integrate a number of similar cores into one package to produce a multi-core processor became evident. In multi-core processor chips, designers usually place the multiple cores side by side on one side of the chip, and the cache on the other side. Unfortunately, this is not an optimal floorplan in terms of temperature. As a demonstration, we performed some experiments to study temperature aware floorplanning for the dual-core

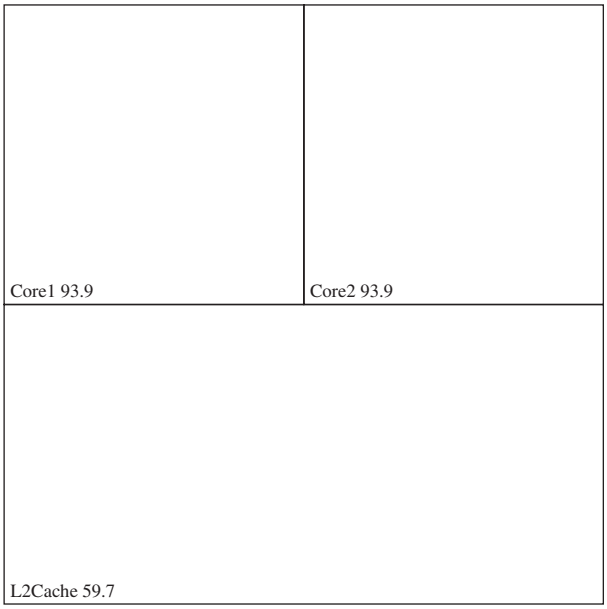


Fig. 20. The original Core 2 Duo floorplan **Core2-orig**.

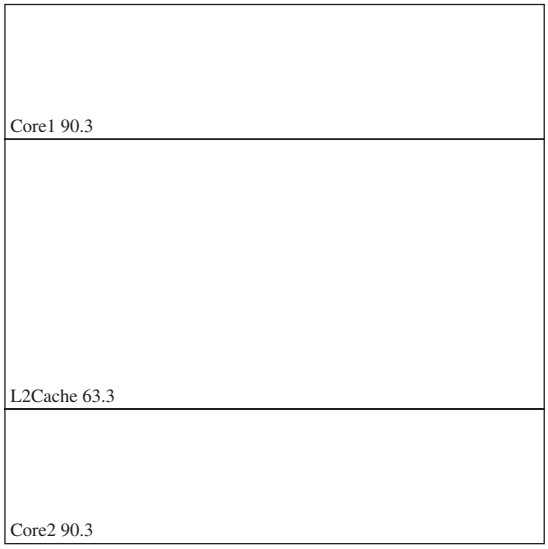
microprocessor, Intel's Core 2 Duo.[21] The original floorplan of the Core 2 Duo CPU is shown in Figure 20.[22] There are two cores occupying the top half of the chip. The other half of the chip is occupied by the level 2 cache which is shared by the two cores. The core temperature of the Core 2 Duo CPU is 93.9 °C.[23, 24]

Two new floorplans for the Core 2 Duo CPU are shown in Figure 21. In the floorplan **Core2-low1**, the two cores are separated and the level 2 cache is placed between the two. In the floorplan **Core2-low2**, the level 2 cache is divided into two parts that are placed separately. In the first floorplan, the aspect ratio of the core is significantly changed. This may result in routing difficulties and is likely to impact the performance. In contrast, the second floorplan keeps the aspect ratios of the cores unchanged, so no routing problems should arise, and since the two cores do not communicate directly with each other, the performance impact is expected to be negligible.

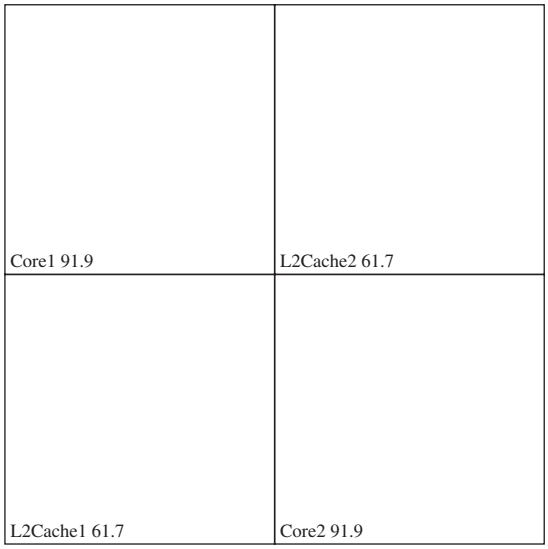
The core temperature in the floorplans **Core2-low1** and **Core2-low2** is reduced to 90.3 °C and 91.9 °C, respectively. The reduction in the core temperature is 3.6 °C and 2 °C for the floorplans **Core2-low1** and **Core2-low2**, respectively. We can see from Figure 21 that placing the two cores far apart helps reduce the core temperature.

The typical power consumption of the Core 2 Duo CPU is 75 Watt,[21] but we do not have the exact power numbers of the individual blocks on the chip. The temperatures of the blocks in Figures 20 and 21 are calculated assuming the power of each core is 30 Watt.

A reasonable range for the power of each core is from 25 to 35 Watt. We performed some experiments to see the impact of the core power on the core temperature of the three floorplans. The results are shown in Figure 22. For the floorplan **Core2-low1**, the core temperature is reduced



(a) Core2-low 1



(b) Core2-low 2

Fig. 21. New floorplans for Core 2 Duo: **Core2-low1** and **Core2-low2**.

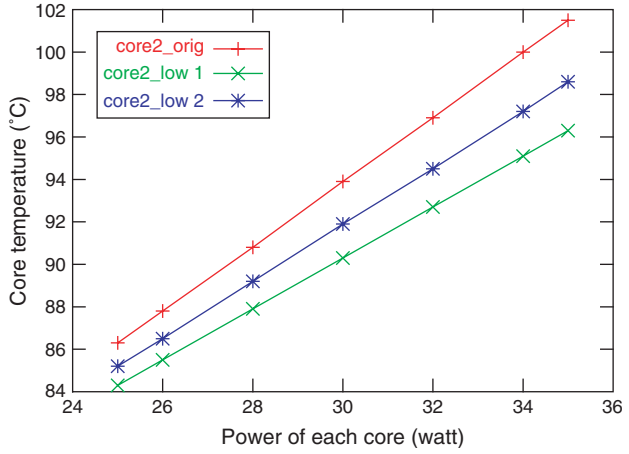


Fig. 22. The core temperatures of the three Core 2 Duo floorplans.

by 2 °C to 5.2 °C for a power range of 25 to 35 Watt. For the floorplan **Core2-low2**, the core temperature is reduced by 1.1 °C to 3 °C for the same range of core power.

7. CONCLUSIONS

In this paper, we have shown how the temperature distribution of a chip can be improved through temperature aware floorplanning. Through experiments on the Alpha, Pentium Pro, and Core 2 Duo microprocessors, we have shown that we can obtain a temperature reduction of 21 °C while keeping a comparable wire length for the Alpha processor, or a 6 °C reduction in the maximum temperature for the Pentium Pro processor with a penalty of 13% in terms of the total wire length, or a 2 °C reduction in the maximum temperature for the Core 2 Duo processor without significant performance degradation. In future designs based on deep sub-micron technology, chip temperatures are expected to further increase, making the benefits of temperature aware floorplanning even more prominent.

Acknowledgments: This work has been supported in part by NSF grant ITR-0205212 and by NSF grant EIA-0102696. The authors wish to thank the LAVA group (the Laboratory for Computer Architecture at Virginia) at the University of Virginia for providing the HotSpot simulator, and especially Karthik Sankaranarayanan for providing the power traces of SPEC benchmark programs.

References

1. S. Lemon, Intel tries to keep its cool. *PC World* **(2004)**.
2. W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam, Compact thermal modeling for temperature-aware design. *Proceedings of the 41st Annual Conference on Design Automation (DAC)* **(2004)**, pp. 878–883.
3. K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, Temperature-aware microarchitecture. *Proceedings of the 30th Annual International Symposium on Computer Architecture (ISCA)* **(2003)**, pp. 2–13.
4. 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.* **(2004)**, Vol. 1, pp. 94–125.
5. http://lava.cs.virginia.edu/hotspot.
6. C. C. N. Chu and D. F. Wong, A matrix synthesis approach to thermal placement. *Proceedings of the 1997 international symposium on Physical design* **(1997)**, pp. 163–168.
7. W.-L. Hung, C. Addo-Quaye, T. Theocharides, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, Thermal-aware IP virtualization and placement for networks-on-chip architecture. *Proceedings of International Conference on Computer Design (ICCD)* **(2004)**, pp. 430–437.
8. W.-L. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T. Theocharides, and M. J. Irwin, Thermal-aware floorplanning using genetic algorithms. *Proceedings of International Symposium on Quality Electronic Design (ISQED)* **(2005)**, pp. 634–639.
9. W.-L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, Interconnect and thermal-aware floorplanning for 3D microprocessors. *Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED)* **(2006)**, pp. 98–104.

10. 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* **(2005)**, Vol. 7, pp. 8–16.
11. M. Healy, M. Vittes, M. Ekpanyapong, C. S. Ballapuram, S. K. Lim, Lee, and G. H. Loh, Multiobjective microarchitectural floorplanning for 2-D and 3-D ICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **(2007)**, Vol. 26, pp. 38–52.
12. D. C. Burger and T. M. Austin, The SimpleScalar tool set. Version 2.0, Tech. Rep. CS-TR-1997-1342 **(1997)**.
13. Y. Han, I. Koren, and C. A. Moritz, Temperature aware floorplanning. *Second Workshop on Temperature-Aware Computer Systems (TACS-2)* **(2005)**.
14. J.-L. Cruz, A. González, M. Valero, and N. P. Topham, Multiple-banked register file architectures. *Proceedings of the 27th Annual International Symposium on Computer Architecture* **(2000)**, pp. 316–325.
15. K. Ghose and M. B. Kamble, Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation. *Proceedings of the 1999 International Symposium on Low Power Electronics and Design* **(1999)**, pp. 70–75.
16. T. Juan, J. J. Navarro, and O. Temam, Data caches for superscalar processors. *Proceedings of the 11th International Conference on Supercomputing* **(1997)**, pp. 60–67.
17. http://www.spec.org
18. S. N. Adya and I. L. Markov, Fixed-outline floorplanning: Enabling hierarchical design. *IEEE Trans. on VLSI* **(2003)**, Vol. 11, pp. 1120–1135.
19. http://www.sandpile.org/impl/p6.htm
20. http://www.faculty.iu-bremen.de/birk/lectures/PC101-2003/02pentium/pentium%20webpage/Index.htm
21. http://www.sandpile.org/impl/core.htm
22. S. Gochman, A. Mendelson, A. Naveh, and E. Rotem, Introduction to Intel Core Duo processor architecture. *Intel Technology Journal* **(2006)**, Vol. 10, pp. 89–97.
23. http://download.intel.com/design/intarch/designgd/31116101.pdf
24. E. Rotem, A. Cohen, J. Hermerding, and H. Cain, Temperature measurement in the Intel Core Duo processor. *Proceedings of 12th International Workshop on Thermal Investigations of ICs (THERMINIC 2006)* **(2006)**, pp. 23–27.

Yongkui Han

Yongkui Han *received the B.S. and M.S. degrees in the department of electronic engineering from Tsinghua University in 1999 and 2002, respectively. Since 2002, he has been a Ph.D. student at the University of Massachusetts Amherst. The topic of the Ph.D. program is power, energy, and temperature aware computer systems.*

Israel Koren

Israel Koren *received the D.Sc. degree from the Technion-Israel Institute of Technology, Haifa, in 1975 in Electrical Engineering. He is currently a professor of Electrical and Computer Engineering at the University of Massachusetts, Amherst. Previously he was with the Technion-Israel Institute of Technology. He also held visiting positions with the University of California at Berkeley, University of Southern California, Los Angeles, and University of California, Santa Barbara. He has been a consultant to several companies, including IBM, Intel, Analog Devices, AMD, Digital Equipment Corp., National Semiconductor, and Tolerant Systems. Dr. Koren's current research interests include fault-tolerant architectures, power and temperature aware techniques, yield and reliability enhancement, and computer arithmetic. He has published extensively in several IEEE Transactions and has more than 200 publications in refereed journals and conferences. He currently serves on the editorial board of the IEEE Computer Architecture Letters and the VLSI Design Journal. He was a co-guest editor for the IEEE Transactions on Computers, special issues on high yield VLSI systems, April 1989, on computer arithmetic, July 2000, and on fault diagnosis and tolerance in cryptography, September 2006. He served on the editorial board of the IEEE Transactions on Computers between 1992 and 1997 and the IEEE Transactions on VLSI Systems between 2001 and 2007. He also served as general chair, program chair, and program committee member for numerous conferences. He is the author of the textbook Computer Arithmetic Algorithms (A. K. Peters, Ltd., 2002), and a co-author of the textbook Fault-Tolerant Systems (Morgan-Kaufmann, 2007). He is a fellow of the IEEE.*