

Thermal Management with Asymmetric Dual Core Designs

Soraya Ghiasi and Dirk Grunwald

University of Colorado

Department of Computer Science

Boulder, CO 80309

{ghiasi, grunwald}@cs.colorado.edu

Abstract

Thermal considerations play an increasingly important role in the design of new processors. Thermal concerns impact how chips are laid out, how quickly processors can be clocked, processor reliability, and how expensive the packaging is. Current production chips are often packaged to dissipate maximum typical power, rather than maximum absolute power. We investigate techniques which would allow the use of an even lower thermal threshold, thus further reducing packaging costs.

We examine single core and dual core solutions to deal with thermal overloads. Our single core solutions include techniques already deployed by chip manufacturers and others proposed by researchers. We also present symmetric and asymmetric dual core techniques. We find that our dual core techniques compare favorably to single core techniques in their ability to reduce thermal loads. Our asymmetric dual core techniques provide additional advantages over single core techniques, by providing an additional low power core that can be used to improve overall system throughput.

1 Introduction

Power and thermal considerations play an increasingly important role in the design of modern processors. Millions of transistors are employed to increase performance, but the heat generated by these transistors must be dissipated. Many approaches to *energy efficient* microprocessors have been examined; only recently have researchers tried to assess how those mechanisms impact thermal dissipation. Skadron et. al, found power and temperature to be poorly correlated [14]; it may be that while mechanisms that save power usually reduce heat dissipation, better thermal regulation may be possible using mechanisms designed specifically for such regulation.

Thermal problems are becoming more severe, particularly for embedded and mobile applications. Packaging is expensive - the cost for effective thermal regulation increases the overall system cost. Power increases as cfV^2 , where c is capacitance, f is frequency and V is voltage. Processors already run at ultra-low voltages, and continued decreases in voltage will be difficult to engineer. The dynamic component of power can be controlled in other ways including designing simpler processors (reducing c), scaling processors to smaller technologies (reducing c , but increasing leakage), or by reducing processor frequency.

Thermal regulation also allows processor cooling systems to be designed for the common case. For example, the “thermal design point” of Intel Pentium-4 processors is set to be 75% of the maximum power because applications rarely exceed this threshold. This means that cheaper packaging can be selected because it is not necessary to dissipate as much power. Moreover, the ability to dissipate power depends on the ambient temperature and it having an active thermal management solution provides better reliability and processor longevity.

There are a broad range of thermal management techniques, including mechanisms controlled by the microarchitecture (*e.g.*, shutting down processor components), platform (such as reducing the processor clock), operating system (voltage scaling or managed system shutdown). In this paper, we examine mechanisms that are invisible to operating systems. We compare a variety of mechanisms and propose a new mechanism, asymmetric dual-core processors, for thermal regulation. We show that this mechanism achieves better performance

than competing mechanisms. The primary contribution of this paper is to explore the efficacy of dual-core processor designs thermal regulation.

The primary contribution of this paper is to explore the efficacy of dual-core processor designs thermal regulation. Researchers at Intel labs have proposed a dual core mechanism to prevent thermal emergencies. In their variation, two identical cores are placed on a single chip. Instructions are scheduled to a core for some specified scheduling quanta and then processing is switched to the second core. This computational interleaving allows one processor to cool while the other one is in use. Although this mechanism is wasteful of die are, it should reduce the occurrence of thermal emergencies.

By comparison, our mechanism uses *asymmetric dual cores*, or a combination of two processor cores with equivalent functionality but differing implementations. Typically, one processor has a significantly simpler implementation than the other – this reduces the cost invested to prevent the (hopefully rare) thermal emergencies while still providing a reasonable level of performance during those emergencies.

Thermal regulation techniques may be either *reactive* (being applied when a thermal overload is detected) or *preventive* (attempting to avoid the onset of thermal overload). We studied one technique from each category. We also compare these to some pre-existing single core mechanisms for thermal regulation.

In Section 2 we discuss related work. We briefly describe our modeling and simulation environment in Section 3. Section 4 presents our methodology followed by our evaluation criteria in Section 5. Our results are presented in Section 6. Our conclusions and future work are in Section 7.

2 Related Works

Our work merges two disparate areas of research; thermal management and dual core design. A considerable amount of research effort has recently been made in the area of thermal management. Brooks and Martonosi [1] consider dynamic thermal management mechanisms in a single core system. They explore a variety of hardware and software mechanisms ranging from instruction cache toggling to dynamic voltage scaling. Their work uses

power to represent the current temperature of the system.

Skadron et. al, found power and temperature to be poorly correlated [14] and instead use an RC based thermal model. They introduce control-theoretic techniques for dynamic thermal management. They use these techniques to evaluate different mechanisms for reducing the time spent in thermal emergencies [12, 13]. They explore a variety of single core solutions including the possibility of migration to a new functional block, in this case the register file.

Functional block migration on a single core is a general case of the dual pipeline technique proposed by Lim et. al [9]. Their work adds a second, lower power pipeline to the core. In cases of thermal emergencies, the secondary pipeline is used until the primary pipeline has had sufficient time to recover.

The previously mentioned works have all examined single core, reactive mechanisms. Intel has proposed a dual core mechanism to prevent thermal emergencies [8]. Two identical cores are placed on a single chip. Instructions are scheduled to a core for some specified scheduling quanta and then processing is switched to the second core. Although this mechanism is computationally wasteful, it should reduce the occurrence of thermal emergencies.

Our work differs from the prior work in this area with its introduction of asymmetric dual core chips. We study the feasibility of using two general purpose processors of different sizes and complexities to address thermal emergencies. Techniques may be either reactive or preventive; we studied one technique from each category. We also compare these to some pre-existing single core solutions.

Dual core designs have typically focused on either symmetric or asymmetric approaches. Symmetric designs make use of two identical cores. Asymmetric designs instead allow for specialized cores. Little work appears to have been done on general purpose asymmetric designs. Chip manufactures scale existing processor designs to new smaller generations, but have not placed new and older, scaled processors together on the same die. Some implementations of the “Itanium” processor design do incorporate a small IA-32 processor on-die, but this is used to execute IA-32 instructions. Such an implementation is an example of a specialized asymmetric design.

Intel, AMD, HP, and Sun have all announced symmetric multi-core designs using their processors. IBM is already shipping symmetric dual core designs [10, 11], with two occurrences of the same core are located on a single die. This is done primarily to increase the processing power by allowing two independent processes to be scheduled. Alternatively, it can be treated as a small multi-processor with low communication latencies due to shared caches. Academic efforts focusing on chip multiprocessors, such as Hydra [4], have also used symmetric designs.

STMicroelectronics' STLC1502 represents a typical asymmetric core design with a DSP core and a general purpose RISC core on the same chip [7]. Texas Instruments TMS320C80 presents a similar solution with 4 DSP cores and one general purpose RISC core. Other implementations use asymmetric cores to handle network communication via TCP/IP, MPEG encoding and decoding, or cryptographic functions.

Our work differs from the traditional efforts in dual core design by focusing on asymmetric designs for thermal management and the use of two general purpose cores, rather than a combination of general purpose and domain specific cores.

3 Modeling and Simulation

We use a simplified version of the RC-thermal model used by Skadron et. al. Rather than model individual functional blocks, we use an RC-thermal model of the whole processor. This approach was first introduced by Dhodapkar, et al [6] in the Tempest power model. We plan to extend our model in the future to model thermal dissipation at the functional block level and to include the effects of adjacent blocks and adjacent cores.

The temperature contribution T_i from the core at cycle i is governed by the following equation:

$$\begin{aligned} T_i &= T_{power} + (T_{i-1} - T_{power})e^{radiative\ factors} \\ &= P_i R + (T_{i-1} - P_i R)e^{\frac{-1}{fRC}} \end{aligned}$$

This can be simplified in cases where $\frac{1}{fRC} \ll 1$.

$$T_i = P_i R + (T_{i-1} - P_i R) \left(1 - \frac{1}{fRC}\right)$$

Power P is a calculated quantity during simulation and depends upon both the design of the core and the activity occurring in it. f is the frequency of the processor and determines the amount of much time the core has to dissipate or gain heat before the next set of activities occur. R and C are dependent upon the materials used, the area over which heat can be transferred and the thickness of the material. $R = \rho t A$, where ρ is the thermal resistivity, t is the die thickness and A is the surface area of the core. $C = c A t$, where c is the thermal capacitance and t and A are as above.

We present a subset of results from Skadron, et al. In order to make a direct comparison, we selected the same values for p , c and t that they used in their own work. $\rho = 10^2 mK/W$, $c = 10^6 J/m^3 K$ and $t = 0.5 mm$.

We model only the temperature of the core and assume a steady state for both the ambient air and heat sink temperatures. For our efforts, we treat these as constants give the high thermal resistance and thus slow thermal dissipation involved. We assume an ambient air temperature of 30C and a heatsink temperature of 70C.

We define thermal emergencies to occur any time the temperature exceeds 75C. A threshold of 75C is artificially low because we want to induce many thermal emergencies to study the effects of our techniques. Although others have pursued thermal management techniques with much high thermal thresholds, we feel justified in choosing such a low value because of the variety of real world processors that operate within this range. In addition, our intent is to find techniques that will allow for a significant reduction in packaging costs. The new temperature is calculated each cycle and compared against a thermal threshold of 75C.

Our NoNameToGiveUsAway simulator has extended the HotLeakage simulator [15] by adding the simplified model of RC-thermal modeling discussed above. HotLeakage includes a small degree of temperature sensitivity, but not dynamic temperature sensitivity. Future versions of NoNameToGiveUsAway will include dynamic temperature sensitivity. HotLeakage is built upon Wattch [2] and extends the leakage power modeling used by Wattch. Wattch is a power estimation tool built upon the SimpleScalar 3.0 microarchitectural simulator [3].

NoNameToGiveUsAway has also been extended to handle dual cores. Architected state, power, caches, temperature, and other attributes are recorded for each core. The simulator supports both shared and separate L2 caches. Switches between cores are naturally penalized by the need to retrieve instructions and data from either a shared L2 cache or main memory. The simulator currently handles sequential, but not parallel execution on two cores.

4 Mechanisms and Simulation Parameters

A number of different dynamic thermal management techniques have been explored by previous researchers or are already in use in commercial processors. We consider a few of the existing techniques as well as new dual core techniques.

4.1 Single Core Mechanisms

Single core mechanisms rely on features that are local to the core. These range from designs with replicated functional blocks to dynamic voltage scaling. We explore a small subset of the possible mechanisms and describe these below.

4.1.1 Global Clock Gating - Duty Cycles and Intel Thermal Management

In the Pentium 4, Intel uses a combination of global clock gating and a BIOS defined duty cycle. The duty cycle determines the period at which an external clock is ANDed with global clock. For example, a duty cycle of 50% means that the global clock is passed to the core 50% of the time and that during the periods it is passed, the cycle time is the same as it is in the absence of any mechanism. We refer to this mechanism as duty cycle-based global clock scaling throughout the remainder of this paper.

4.1.2 Global Clock Gating - A Cheap Way of Frequency Scaling

Intel’s thermal management solution suggests that it is possible to simulate frequency scaling by the appropriate selection of a external clock signal. We simulate a core that uses an $f/2$ clock to reduce the operating frequency of the core by half. We refer to this technique as frequency scaling throughout the rest of the document.

The difference between these two techniques is shown in Figure 1. At cycle 0, when the mechanisms are triggered, both techniques gate the global clock. In frequency scaling, the clock is gated every other cycle. In a duty cycle based scheme, the clock is gated for the duration of the predefined duty cycle and then normal clock signalling is resumed.

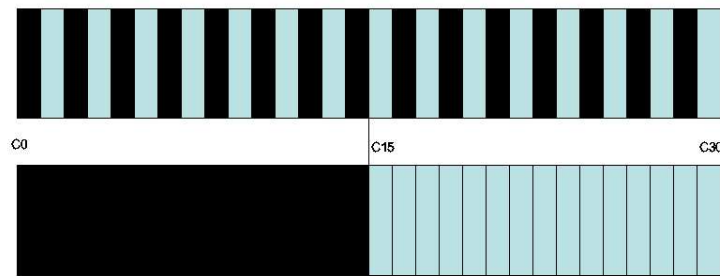


Figure 1: Frequency Scaling (top) versus Duty Cycle-based Global Clock Gating (bottom)

4.1.3 Instruction Fetch Toggling

Brooks, et al. found instruction fetch toggling to be an effective method for reducing thermal emergencies. The instruction fetch queue is disabled every n th cycle. We explore this mechanism with $n = 1$. In this case, fetch stops entirely until the criteria for disengaging the mechanism is met. Brooks [1] and Skadron [14] both found instruction fetch toggling with $n = 1$ to be an effective thermal management mechanism so we have included it in our study.

4.2 Dual Core Mechanisms

Dual core mechanisms can exploit the same features used by single core designs. However, they also have additional features and degrees of freedom which make this a rich design space. Intra- and inter-core features can be used to optimize power, performance or thermal behavior. In this work, we focus on thermal behavior. We explore four different mechanisms and include two variations of each mechanism. It is important to consider the effects of shared and non-shared L2 caches on the thermal and performance behavior of dual core mechanisms. Shared caches are potentially more energy efficient and suffer less of a performance penalty on switch. However, they require additional work on the part of the designer when merging two already existing processors and can have a negative impact on performance due to contention.

4.2.1 Dual Core Stripping - Intel and Symmetric Core

Intel has proposed the use of two identical cores to reduce the occurrence of thermal emergencies [11]. Processing occurs on core A for a preset duration and then processing is switched to core B. The duration is chosen ahead of time and is long enough for a processor that is near a thermal emergency to cool significantly.

4.2.2 Asymmetric Dual Core Stripping

The idea of stripping work across two identical cores can be extended to stripping across asymmetric cores provided that both are general purpose cores that support the same instruction set. We do not explore the effects of a dual core design with one general purpose core and one specific purpose core. This lies outside the scope of our work and does not provide a general solution to the problem of thermal dissipation in general purpose processors.

4.2.3 Symmetric Dual Core Offloading

Single core mechanisms generally use microarchitectural features, some form of scaling, or a limited duration suspension of processing to reduce the power and thus the temperature. Offloading to a less powerful core can be performed in lieu of these solutions. When a thermal emergency arises, processing is switched to the second core until the primary core is again available for processing.

4.2.4 Asymmetric Dual Core Offloading

This mechanism is identical to the Symmetric Dual Core Offloading except both cores are no longer identical. When a thermal overload occurs, processing is switched to the lower power core.

4.3 Simulator Parameters

Our single core mechanisms are all evaluated on a high power modern processor called Core A. Core A has an extremely fast L1 instruction cache, a large instruction window, and a high operating frequency. Our dual core mechanisms are combinations of Core A and either a duplicate of Core A or a less powerful Core B. Core B is designed to be the direct ancestor of Core A scaled to the same process technology as Core A. It has a lower voltage, a lower operating frequency, and a much smaller area than Core A. For simplicity, both cores use the basic five-stage pipeline provided by SimpleScalar. Both cores are shown in Table 1. Although the voltages, areas, frequencies and general microarchitectural trends are taken from real processors, no direct comparisons should be made because the differences in the fabricated and simulated designs are significant.

5 Evaluation Criteria

The techniques we examine have different effects on the a variety of different aspects of processing. We are interested in both the efficacy of our techniques as thermal management tools and in the performance implications of their use. With these guiding principles in mind we chose the following criteria to analyze our techniques.

Parameters	Core A	Core B
Area	116 mm^2	21.9 mm^2
Voltage	1.35 V	1.15 V
Frequency	2.8 GHz	1.4 GHz
Technology	0.13 μ	0.13 μ
Machine Width	4 wide fetch, 4 wide issue, 4 wide commit	
Window Size	128 entry RUU 64 entry load/store queue	64 entry RUU 32 entry load/store queue
Branch Misprediction Latency	19 cycles	12 cycles
L1 Icache	16K, 4-way 32 byte lines 2 cycle latency	16K, 4-way 32 byte lines 3 cycle latency
L1 Data Cache	8K, 4-way 32 byte lines 2 cycle latency	16 K, 4-way 32 byte lines 3 cycle latency
L2 Combined	512K, 8-way 128 byte lines 10 cycle latency	256K, 4-way 32 byte lines 25 cycle latency
Memory	128 bit wide 92 cycle latency	128 bit wide 41 cycle latency
BTB	4096 entry, 4-way set-associative 32 entry return address stack	512 entry, 4-way set-associative 32 entry return address stack
TLB	128 entry (I), 128 entry (D) 4-way set-associative 30 cycle miss latency	64 entry (I), 64 entry (D) 4-way set-associative 30 cycle miss latency
Functional Units and Latency (total/issue)	2 Int ALU (1/1), 1 Int Mult (2/2) / Div(2/2) 4 Load/Store (2/1), 1 FP Add (5/3) 1 FP Mult (6/5) / Div (6/5) / Sqrt (6/5)	1 Int ALU (1/1), 1 Int Mult (2/2) / Div(2/2) 2 Load/Store (2/1), 1 FP Add (5/3) 1 FP Mult (6/5) / Div (6/5) / Sqrt (6/5)

Table 1: Simulation Parameters

5.1 Effectiveness at Reducing Temperature

The first criteria we examine is simply the efficacy of a technique in reducing the temperature. The temperature is set at slightly above the threshold temperature T_{thresh} . Each mechanism is applied for a set length of time and the temperatures at the end of the period are compared. This metric gives an indication of how quickly a mechanism is able to cool off a core. In the two cases of global clock scaling examined, work continues on the single core. Fetch gating allows only a small amount of already inflight work to continue. All dual core techniques prevent work from happening on a given core. In all cases a core must dissipate at least leakage power to be able to cool below T_{thresh} .

5.2 Performance Loss

Thermal overloads can occur in any processor regardless of how it is deployed. Certain techniques may be more applicable to mobile platforms where a certain amount of performance loss is acceptable. These same techniques may be unsuitable for use in a performance critical server farms. The performance of applications in the presence and absence of thermal reduction techniques is compared. We expect single core techniques to suffer more performance loss than dual core techniques.

5.3 Number of Thermal Threshold Violations

We also examine the effectiveness of a technique in reducing the number of cycles where $T > T_{thresh}$. We compare the cycles spent above T_{thresh} for applications run with and without the use of a thermal reduction technique. The single core techniques and the dual core offloading technique are reactive measures. The dual core swapping techniques are intended as a preventive measure and this should be reflected in the number of violations observed.

5.4 Unused Cycles

The amount of wasted work is measured by the cycles in which a core is unused. It is only applicable for the dual core measures. In the case of dual core swapping, 50% of the time on any given core is unused. This time could be spent completing additional work. The amount of useful work that could be accomplished in this time is constrained by the frequency of the unused core and its performance. Our current simulator does not support concurrent execution of jobs, but we plan to revisit this

6 Results

We present results only for SPEC CPU2000 benchmarks which exceeded T_{thresh} . Because we chose a low threshold temperature (75C) many applications which exceed the threshold do so for much of their lives. The cycles spent above the threshold range from 5% to 95% of the cycles. All benchmarks are run for 100 million instructions starting from an idle, but not cold processor. The starting temperature influences the temperatures observed during an application's run.

Figure 2 illustrates the time-lagged nature of the relationship between temperature and power. Both power and temperature have been scaled to the same interval. No conclusions should be drawn about the magnitudes. Instead, the key point is that small, infrequent power spikes are relatively unimportant. Long term phases of high power consumption lead to significantly increased temperatures. The temperature slowly drops after the end of such phases.

6.1 Single Core Mechanisms

All three single core techniques studied are reactive in nature. As soon as a situation occurs in which $T > T_{thresh}$, a technique is invoked for a certain duration.

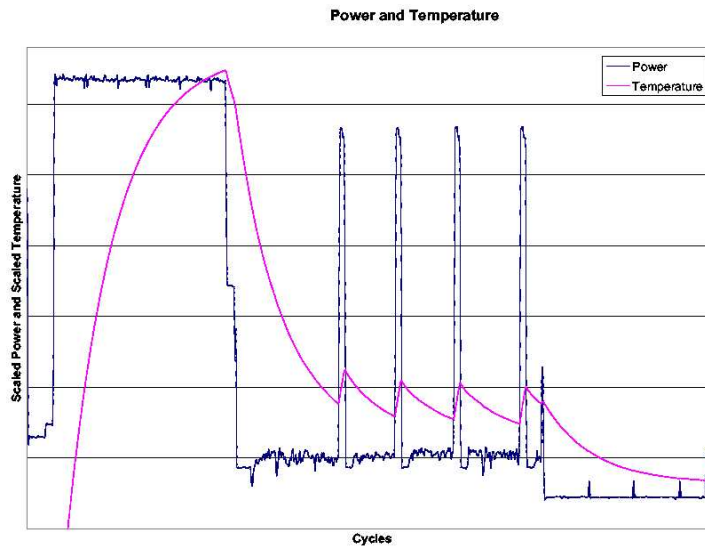


Figure 2: Temperature lags behind changes in power consumption.

6.1.1 Effectiveness at Reducing Temperature

All techniques were applied for the same duration. Applications were allowed to reach T_{thresh} by processing instructions. Once T_{thresh} is reached, a given technique is applied for 6 ms.

Figure 3 shows how effective the single core techniques are at reducing the temperature of a core once T_{thresh} is reached. Results are shown for mgrid. Frequency scaling without voltage scaling was found to be ineffective. Simply gating the global clock every other cycle does not allow enough time for the core to dissipate the excess heat without scaling the voltage as well. Duty cycle-based global clock gating proves to be more effective, but still does not effectively cool the core. In this case, we are limited by the amount of time the core can be gated without loss of data. For example, the longest non-active period used in Intel’s Pentium 4 is approximately $3\mu s$ [5]. It does reduce the extent to which the overall temperature rises, but cannot prevent thermal overloads from occurring. A duty cycle of 12.5%, rather than 50%, is more effective. The final single core technique considered is fetch gating. In this case, fetch is gated for the entire duration and the temperature eventually drops to the idle temperature.

The geometric mean of the resulting temperature across all benchmarks which exceed T_{thresh} for all three

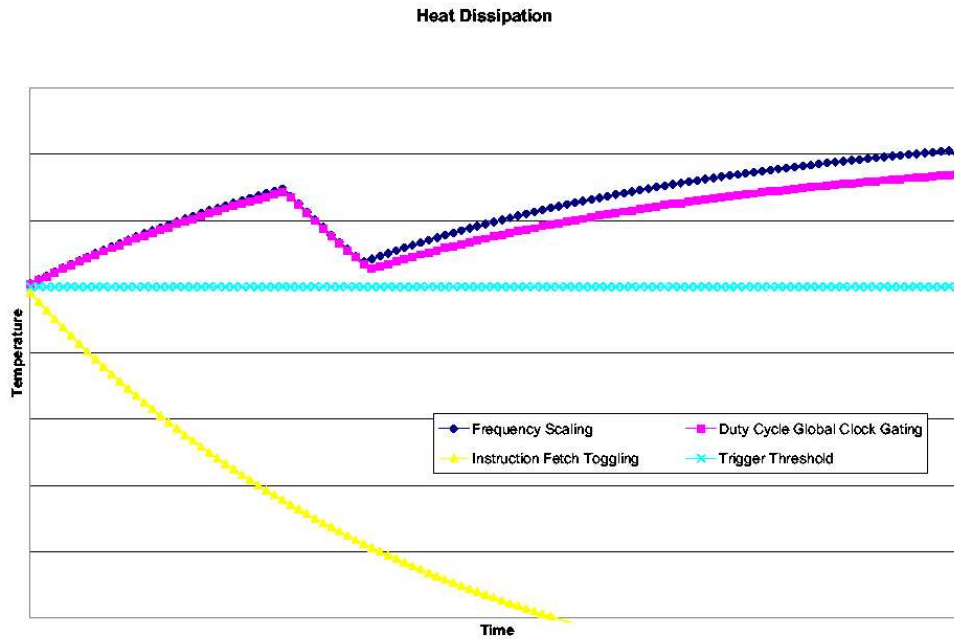


Figure 3: Single core techniques and their effectiveness at cooling the core while running mgrid

techniques is shown in Figure 4. On average the techniques are valid, but as Figure 3 demonstrates, there are cases where they may fail.

6.1.2 Performance Loss

Figure 5 shows the performance lost by the applying thermal reduction techniques. It demonstrates that the more effective a technique is at reducing the temperature and reducing the number of thermal violations, the larger the performance impact on the applications is. This will be discussed further in the next criterion analysis.

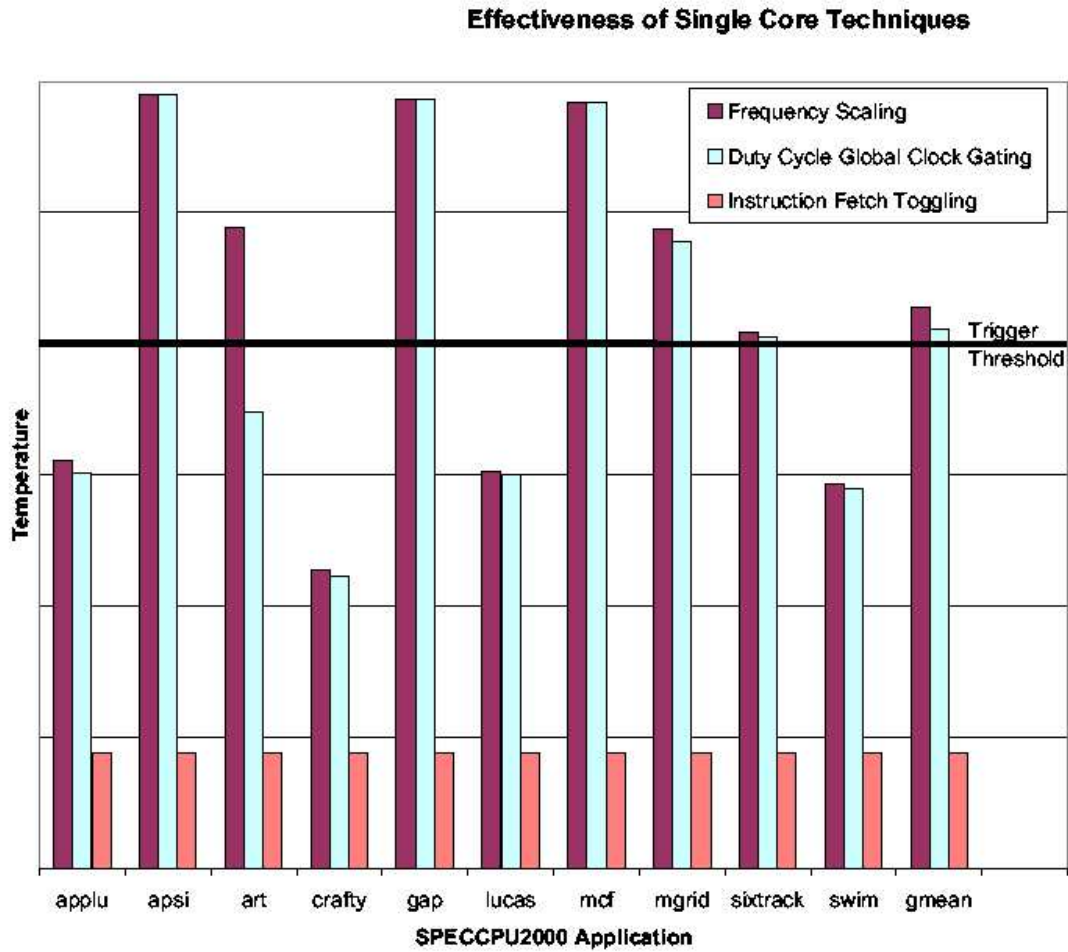


Figure 4: Single core techniques and their effectiveness at cooling the core

6.1.3 Number of Thermal Threshold Violations

The percentage of cycles spent above T_{thresh} is shown in Figure 6. This figure, when considered in conjunction with Figure 4 illustrates both the effect of an initially high percentage of cycles of cycles over T_{thresh} . applu, apsi, and art initially spent over 60% of their time above T_{thresh} . applu and apsi suffer the large performance losses under all techniques. art performs much better, but its temperature was frequently just above the threshold, allowing it to be cooled more effectively by the techniques examined. On average, we found these techniques to retain only 60-72% of their original performance. In most cases, they were able to reduce the number of thermal

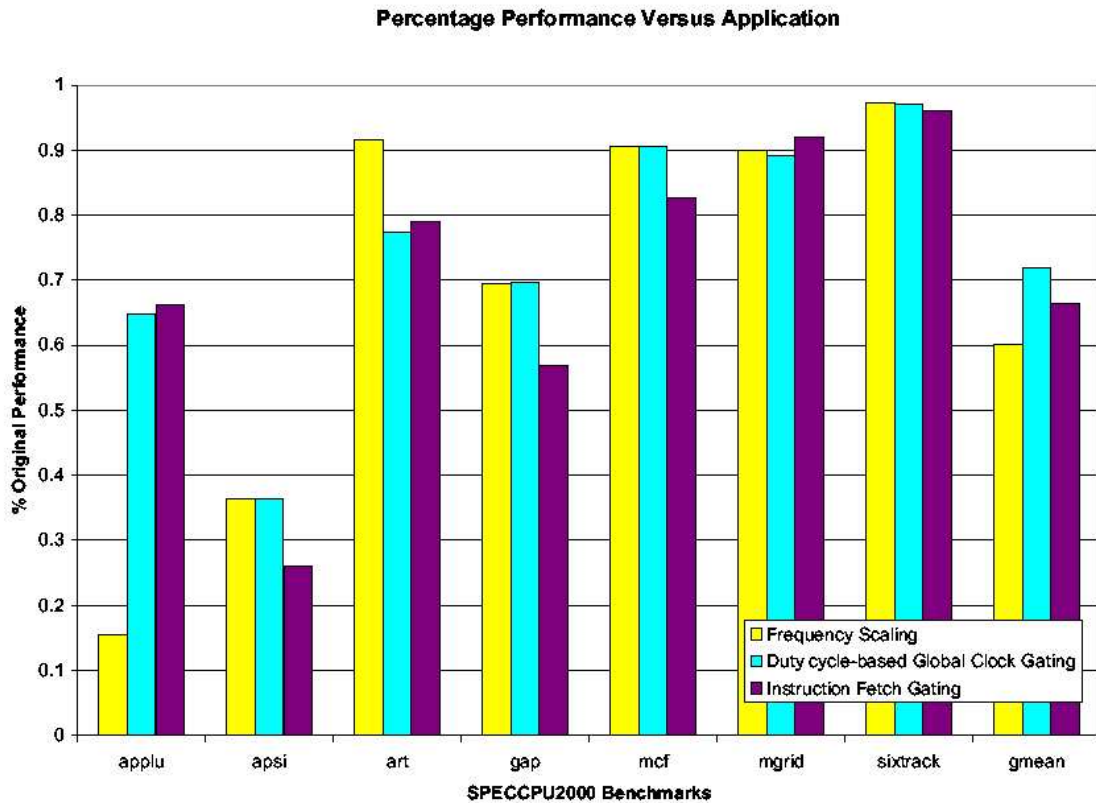


Figure 5: Performance loss due to single core techniques

violations as well. mcf is a notable exception to this general trend.

6.1.4 Unused Cycles

The number of unused cycles is not an important for single core techniques. In general, when a single core is not in use it is cooling. Scheduling any additional work to it during this time would prevent cooling from occurring.

6.2 Dual Core Mechanisms

The dual core techniques are divided into preventive and reactive techniques. Dual core offloading for both symmetric and asymmetric cores is a reactive technique. Dual core swapping is a preventive technique.

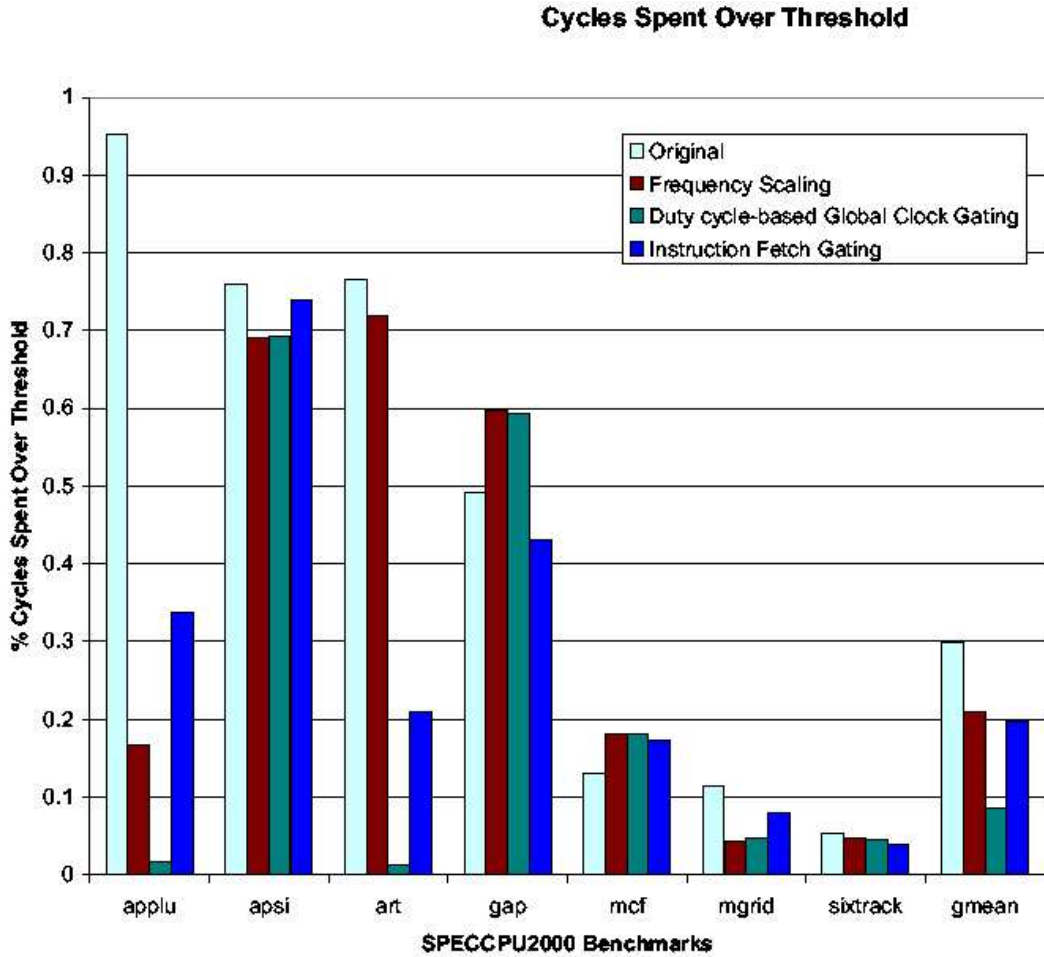


Figure 6: The number of cycles spent above T_{thresh}

6.2.1 Effectiveness at Reducing Temperature

In all eight cases studied, temperature reduction on the now idle core follows the same pattern as it does for instruction fetch gating. Please refer to Figure 3 for comparison. The thermal constants are the same for all cores in the symmetric core case. The thermal constants, and hence the decay time, are different on the asymmetric CoreB, but the curve retains the same overall shape.

6.2.2 Performance Loss

Figure 7 shows the performance lost by the primary application when running a dual core technique. Dual core swapping loses very little of the original performance. Even the worst case technique, asymmetric offloading with unshared caches, still provides reasonable performance. The performance for offloading could be further tuned by adjusting the offload duration, but that has not been done in this case.

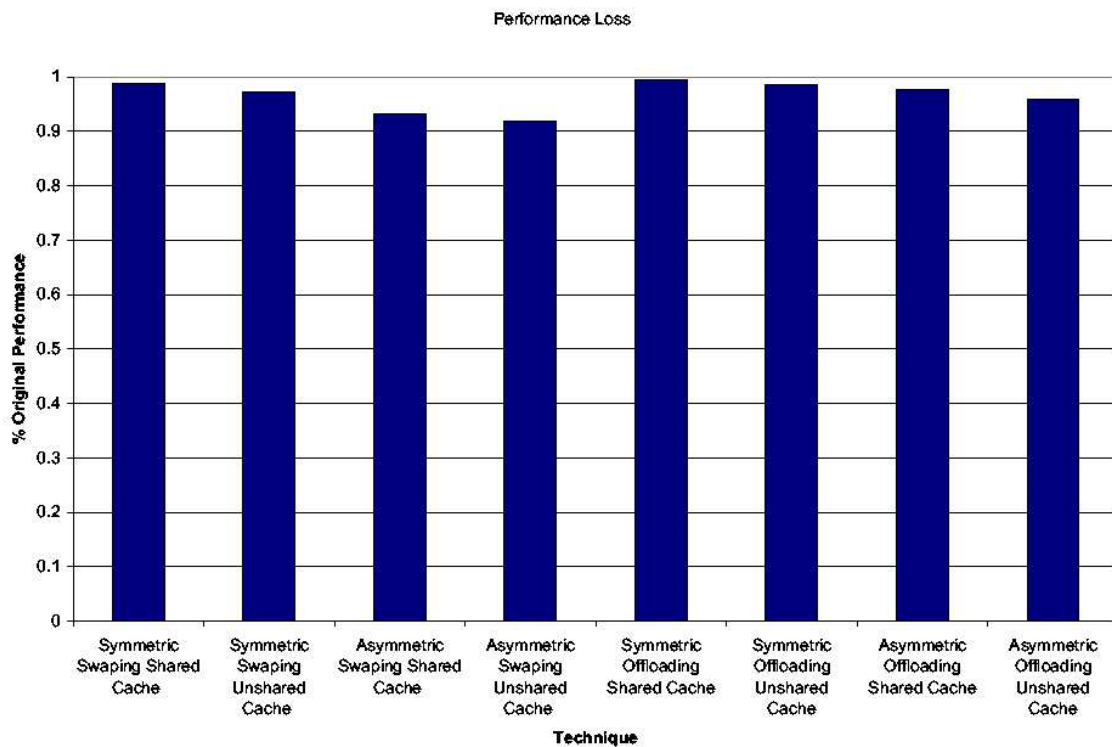


Figure 7: Some performance is lost by applying dual core thermal techniques

6.2.3 Number of Thermal Threshold Violations

The number of thermal violations is reduced to a small fraction of the original number of violations. On average, only 2% of the violations remain. During these cycles, no additional work should be scheduled to the cooling core, but this has not been taken into account in the analysis on unused cycles below.

6.2.4 Unused Cycles

Figure 8 shows the unused cycles for the dual core techniques. For asymmetric cores, the amount of unused work on the secondary core is scaled by the geometric mean of IPC on all applications for the relevant core. This scaling represents the situation where another job could have made use of the resources. An offloaded core can not be used for additional work while it is cooling, but a swapped core may. Care must be taken that a formerly idle core is scheduled with “cool” running jobs to allow some thermal dissipation to continue.

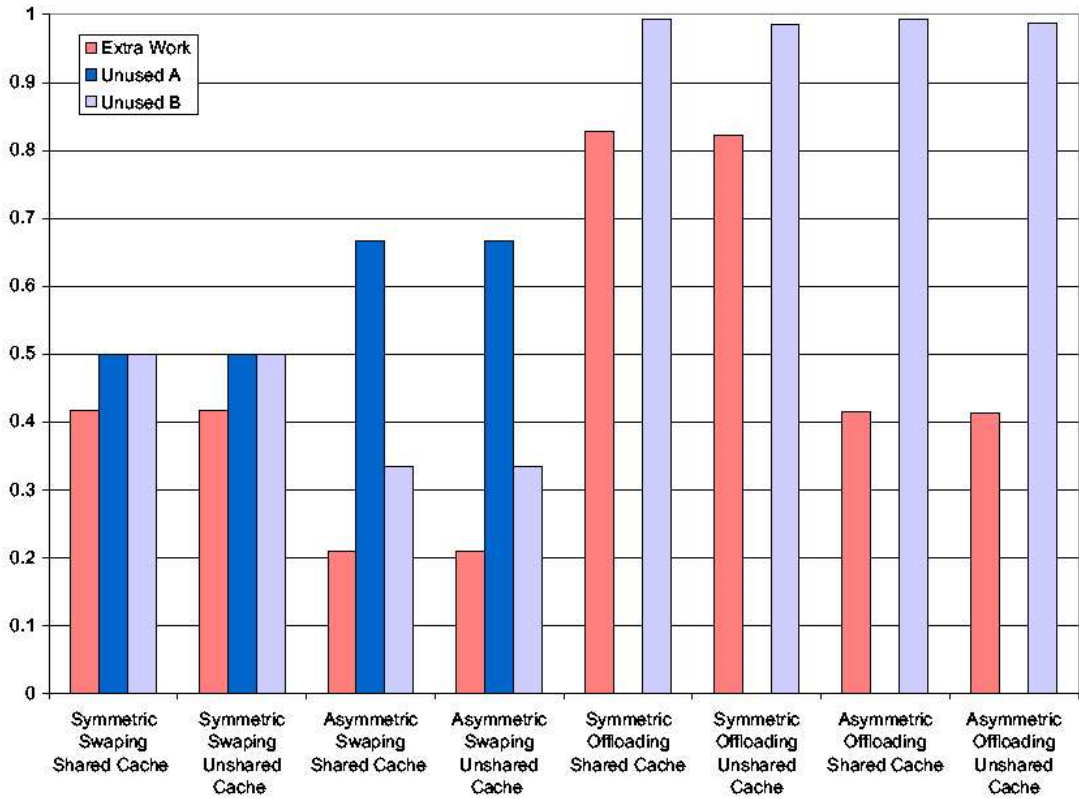


Figure 8: Extra work that may be performed by dual cores

6.3 Single Core Versus Dual Core Mechanisms

Overall, we find dual core solutions to provide both better performance and greater thermal violation reduction than single core techniques. They are also capable of providing cycles to perform useful work that are otherwise

unoccupied in our current scheme. These features do come at the expense of additional power and energy consumption, but this effect can be mitigated through the use of a well designed low power core.

7 Conclusions and Future Work

We find that dual core techniques make an excellent choice for thermal management of processors by providing both good performance and a significant reduction in thermal exceptions. The slight additional power and energy costs of asymmetric solutions alleviate many of the drawbacks of using a dual core solution. Dual core solutions allow for the selection of less expensive packaging than single core solutions at the expense of additional fabrication costs.

Our future work in this area includes significant enhancements to our simulator to enable a more thorough analysis of the relative merits of the techniques proposed here. We believe that both swapping and offloading provide ample opportunity to enhance performance by using otherwise idle cycles, but the thermal implications of running jobs on both cores has yet to be studied.

References

- [1] David Brooks and Margaret Martonosi. Dynamic Thermal Management for High-Performance Microprocessors. In *Proceedings of the 7th International Symposium on High Performance Computer Architecture*, Monterrey, Mexico, January 2001.
- [2] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimization. In *Proceedings of the 27th International Symposium on Computer Architecture*, pages 83–94, Vancouver, Canada, June 2000.
- [3] D.C. Burger and T.M. Austin. The SimpleScalar Tool Set, Version 2.0. *Computer Architecture News*, 25(3):13–25, 1997.
- [4] T-F. Chen and J-L. Baer. Effective Hardware-Based Data Prefetching for High-Performance Processors. *IEEE Transactions on Computers*, 44(5):609–623, May 1995.
- [5] Intel Corp. *Intel Pentium 4 Processor in the 478 pin package Thermal Design Guidelines*, 2001.
- [6] Ashutosh Dhodapkar, Chee How Lim, George Cai, and W. Robert Daasch. TEM2P2EST: A Thermal Enabled Multi-model Power/Performance ESTimator . In *Workshop on Power Aware Computer Systems*, pages 112–125, Boston, November 2000.
- [7] Veronica Hendricks. Dual-Core SoC Simplifies VoIP Terminals and Gateways. *CommsDesign*, Jun 2001.
- [8] Michael Kanellos. At Intel - the chip with two brains. *C—Net news.com*, Aug 2002.

- [9] Chee How Lim, Robert Daash, and George Cai. A Thermal-Aware Superscalar Microarchitecture. In *Proceedings of the International Symposium on Quality Electronic Design*, pages 517–522, San Jose, California, USA, March 2002.
- [10] ARM Ltd. ARM Extends PrimeXsys Family with With Introduction of Dual Core Platform for Networking Applications. *Press Release*, Jun 2002.
- [11] Stephen Shankland. Intel see dual core Itanium by 2005. *C—Net news.com*, Sept 2002.
- [12] Kevin Skadron, Tarek Abdelzaher, and Mircea Stan. Control-Theoretic and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management. Technical Report CS-2001-27, University of Virginia, November 2001.
- [13] Kevin Skadron, Tarek Abdelzaher, and Mircea Stan. Control-Theoretic and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management. In *Proceedings of the 8th International Symposium on High Performance Computer Architecture*, Cambridge, MA, USA, February 2002.
- [14] Kevin Skadron, Mircea Stan, Wei Huang, and Sivakumar Velusamy. Temperature Aware Microarchitecture. In *Proceedings of the 30th International Symposium on Computer Architecture*, San Diego, California, USA, June 2003.
- [15] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects. Technical report, University of Virginia, March 2003.