

Synergistic Temperature and Energy Management in GALS Processor Architectures *

YongKang Zhu
Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627

David H. Albonesi
Computer Systems Laboratory
Cornell University
Ithaca, NY 14853

ABSTRACT

We propose a synergistic temperature and energy management scheme for GALS processors. Localized DVS is applied in domains that contain hotspots, permitting other critical domains to run unabated, thereby reducing performance cost relative to global DVS, and also creating execution slack in peripheral cooler domains that can be exploited to save energy. The reduction in energy in turn creates a steeper temperature gradient between the domains, permitting heat to flow more easily out of the hotspot domain. This symbiotic cyclical relationship between temperature and energy management leads to both significantly better performance, and lower energy, than the use of DTM alone.

Categories and Subject Descriptors: C.1.0 Processor Architectures: General

General Terms: Reliability, Performance

Keywords: Dynamic Voltage Scaling (DVS), Dynamic Temperature Management (DTM)

1. INTRODUCTION

The relentless scaling of transistor dimensions, coupled with a slowdown in supply voltage scaling and rapidly increasing leakage power, has led to unprecedentedly high on-chip power density levels. In response, microarchitectural techniques for Dynamic Temperature Management (DTM) have been proposed for maintaining suitable operating temperatures with reduced packaging and cooling costs [1, 18].

Global DTM techniques, such as Dynamic Voltage Scaling (DVS) and global clock throttling, though effective in reducing chip temperatures, have the disadvantage of impacting global microprocessor performance due to the global reduction in clock frequency, even in cases in which the thermal emergency is isolated to a small region of the die.

The differences in the logic composition and logic density among chip units, and in the utilization of these units as

*This work was supported in part by NSF grants CCR-0304574 and CCF-0541321, and by an IBM Faculty Partnership Award.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'06, October 4–6, 2006, Tegernsee, Germany.

Copyright 2006 ACM 1-59593-462-6/06/0010 ...\$5.00.

Table 1: Thermal characteristics of a fully synchronous microprocessor without any DTM control for SPEC2000 programs.

	Max Temp (degrees C)	Three Hottest Units
crafty	92.4	iExec, IntQ, iReg
eon	98.0	fAdd, fReg, LSQ
gzip	90.2	iExec, IntQ, iReg
mesa	92.8	fAdd, fReg, IntQ
equake	96.5	iExec, IntQ, iReg
facerec	89.6	fAdd, fReg, fMul
fma3d	91.7	iExec, iReg, IntQ
galgel	125.6	fMul, fAdd, fReg

applications execute, means that the thermal hotspots on the die may be isolated to a small subset of all the chip units for any given application. Table 1 shows the thermal characteristics of the SPEC2000 programs used in this paper. One observation is that the units in the front-end are never among the hottest; therefore, there is little need to ever throttle performance in that domain for temperature purposes. Note also that for any given application, the hottest units are located within at most two of the regions of the die (integer, floating point, or load-store).

These results indicate that a *localized* response to temperature emergencies may be effective in maintaining acceptable temperature levels while maintaining global performance. One such approach, localized throttling of the clock within the region of interest, was previously proposed [17]. However, this approach only impacts frequency, and therefore is often too gentle in addressing serious thermal emergencies [17]. On the other hand, *localized DVS*, which can be realized by dividing the processor into several clock/voltage domains [10, 15], has the advantage of being a localized and vigorous response to thermal emergencies.

In this paper, *localized, DVS-based, DTM* is proposed via a Globally Asynchronous, Locally Synchronous (GALS) microprocessor called MCD (Multiple Clock Domain) [15]. In MCD, the major microprocessor functions are located in separate clock/voltage domains. The advantage of this approach, in terms of DTM, is that a localized and strong response can be made to the particular unit which is overheating at any given point of execution. This effectively reduces the thermal problem at the local level, permitting other domains to maintain full speed operation, resulting in less performance overhead compared to a fully synchronous processor with DVS-based DTM. The added performance cost of MCD is, of course, inter-domain synchronization. This cost is shown to be offset by the lower performance overhead afforded by localized DVS-based DTM control.

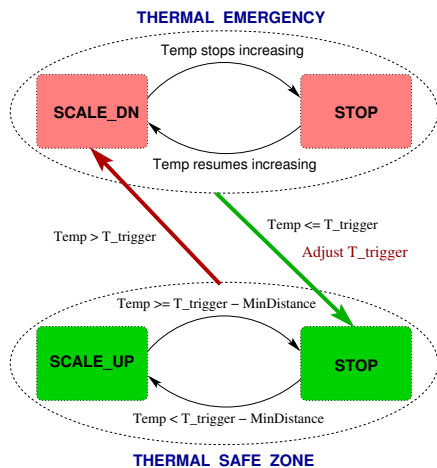


Figure 1: Adaptive DTM algorithm with dynamic setting of control parameters.

An interesting benefit of applying DVS to a subset of the chip is that this may create *execution slack* in the cooler areas of the die. This arises if one or more of the hotspot domains are on the critical path of execution at the time that it is slowed down (or becomes the case due to the slowdown). In this case, the peripheral domains need not run at full speed, and in fact, *Dynamic Energy Management (DEM)* techniques can be used to slow these domains to save energy without compromising performance. This slowing down of the peripheral domains, in turn, permits more lateral heat transfer from the hotspot domain, which reduces the degree that DVS has to be applied to reach acceptable temperature levels. This creates a *symbiotic relationship* between these two techniques (DTM within the hotspot domains and DEM within the others), in which each technique benefits from the use of the other. The application of localized DVS to achieve synergistic temperature and energy management is an interesting area for research that is explored for the first time in this paper.

The rest of this paper is organized as follows. The next section discusses related work, while Section 3 describes a new DTM algorithm that adapts its parameters to fit varying temperature characteristics. A combined DTM and DEM algorithm for MCD processors is presented in Section 4, followed by a description of the evaluation methodology in Section 5. Results are presented in Section 6, and the paper concludes and discusses areas for future work in Section 7.

2. RELATED WORK

A wealth of research has been conducted on architecture level dynamic thermal management. One approach restricts instructions from entering the processor core [1, 14, 17], while another uses dynamic frequency scaling, perhaps coupled with DVS [1, 5, 18]. Other DTM schemes include global clock gating (as used in Pentium 4 processors [6]), dual pipeline activity migration [7], instruction steering to a low power pipeline [12], and some floor planning techniques [18].

A promising approach is to combine several schemes, since different techniques may be well suited for different levels of thermal stress [16]. The framework proposed by Huang et al. [8] selects different schemes to control the temperature of different thermal phases.

Li et al. study how DTM affects the performance and power consumption of SMT and CMP architectures [11]. Furthermore, Powell et al. propose SMT thread assignment and CMP thread migration schemes to control power density [13]. The DVS-based DTM algorithm described in the next section can be extended and applied to both SMT and CMP, and the ability to tune individual domains within the processor core can potentially complement SMT and CMP approaches. This combination is left as future work.

3. ADAPTIVE DVS-BASED DTM ALGORITHM

Figure 1 shows the proposed DTM algorithm, which does not specify any target voltage. Rather, a trigger temperature is specified for engaging DVS, and temperature is sampled at a fine enough granularity to catch small changes. For a fully synchronous processor, one trigger temperature is used for all units; for an MCD machine, each domain has its own trigger temperature. In both cases, thermal sensors are located at each hotspot unit.

Each domain is initially in a thermal safe zone (Figure 1). When any monitored unit temperature exceeds the trigger temperature, the domain enters the thermal emergency zone and the voltage and frequency are reduced at a constant speed. This continues until the maximum temperature is observed to stop increasing and is below the hard limit. As in Intel’s XScale processor [4], all circuits operate during the period of voltage and frequency transition. After the scaling down process is terminated, the temperature either stays stable between the trigger temperature and the hard limit; resumes increasing sometime later, in which case DVS resumes; or gradually decreases below the trigger temperature and enters the safe zone.

A fixed trigger temperature would not fit optimally in all thermal phases, so the algorithm sets it to an initial value and then adjusts it everytime the thermal safe zone is entered after an emergency. Increasing the trigger temperature may be risky in terms of safe temperature control, and we handle this by requiring a minimum separation of 0.5 degrees between the trigger temperature and the hard limit, and forcing the upper limit of the trigger temperature to be lowered dynamically; in our algorithm, the limit is lowered by 0.5 degrees for every three thermal violations.

If after returning to the safe zone the temperature still decreases, the voltage is increased until the temperature is close to the trigger temperature. This minimum temperature distance (called *MinDistance*) must be carefully chosen. If the value is too small, the temperature may oscillate around the trigger thus incurring an alternating scaling down and up process. Too large a value increases stability, but may degrade performance. While dynamic adjustment of the minimum distance is possible, a static value is used in the algorithm.

The last parameter is the voltage transition speed. In the proposed algorithm, the scaling up process occurs at the fastest practical speed, while scaling down occurs at a slower rate. This reduces the chances that a scaling down process results in a subsequent need to scale up, thereby incurring less oscillation and potentially benefiting performance.

4. SYNERGISTIC LOCALIZED DTM AND DEM

Since localized heating occurs much faster than chip wide heating due to slow lateral heat propagation [18], localized

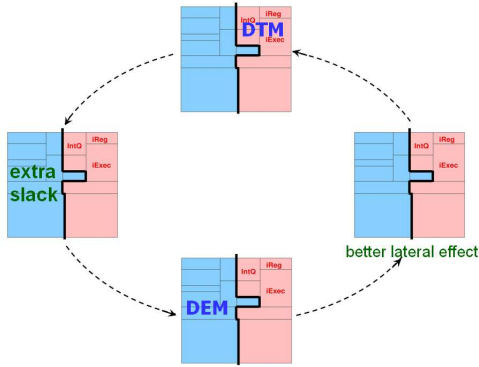


Figure 2: Synergy between DTM and DEM. Hotspot domains are colored with *pink* (lighter shade, right) and non-hotspot domains are colored with *blue* (darker, left).

DTM techniques, such as DVS within a given MCD domain, can be effective in controlling overall chip temperature. To circumvent a potential thermal violation, the throughput of these hot domains is inevitably reduced. This has the salient advantage of introducing execution slack in other domains which can be exploited by energy saving techniques.

On the other hand, effective energy saving techniques often reduce temperature in a local region as a by-product of reducing power consumption. If the local units happen to be the hot spots, then these techniques may avert a thermal emergency that would have otherwise occurred.

If the units being addressed by the energy management technique are peripheral to the hot spot region, the cooling of these neighboring regions permits additional heat transfer from the hot spots due to the better lateral effect resulting from a steeper temperature gradient. This, in turn, requires a less severe response in the hot spot domain, reducing the performance overhead associated with temperature control. This symbiotic cycle of mutually beneficial operation of the two techniques is shown in Figure 2.

The DEM algorithm adopted in this paper is similar to the improved *Attack/Decay* DVS algorithm proposed in [21] except that there are two queue occupancy change thresholds, one for each change direction. Having two thresholds permits more flexible control, and by setting different threshold values, the algorithm can be made more performance oriented or more energy aware.

The combined DTM and DEM approach operates as follows. Temperature sensors located at the hot spots independently trigger DTM control within each domain (using the algorithm described in Section 3). Hardware monitors embedded within each domain track statistics for the DEM algorithm. Within each domain, the DTM algorithm always has priority; the DEM algorithm operates only when DTM is not engaged. Whenever DTM is triggered, the DEM algorithm is disabled; and only when the thermal safe zone is reached again can the DEM algorithm be re-engaged. Note that since the DEM algorithm can only lower voltage and frequency below their nominal operating points that it cannot aggravate the ability of the DTM algorithm to control temperature. Clearly, if there is slack that is being exploited within a domain by the DEM algorithm then either DTM triggering may be avoided altogether, or when DTM is triggered, the severity of the DTM response is lower, reducing the performance overhead.

Table 2: Microarchitectural parameters.

Configuration Parameter	Value
Branch predictor:	
Level 1	1024 entries, history 10
Level 2	1024 entries
Bimodal predictor size	1024
Combining predictor size	4096
BTB	4096 sets, 2-way
Branch Mispredict Penalty	7
Decode/Issue/Retire Width	4/6/11
L1 Data Cache	64KB, 2-way set associative
L1 Instruction Cache	64KB, 2-way set associative
L2 Unified Cache	1MB, direct mapped
L1 cache latency	2 cycles
L2 cache latency	12 cycles
Integer ALUs	4 + 1 mult/div unit
Floating-Point ALUs	2 + 1 mult/div/sqrt unit
INT Issue Queue Size	20 entries
FP Issue Queue Size	15 entries
Load/Store Queue Size	64
Physical Register File Size	72 integer, 72 floating-point
Reorder Buffer Size	80

Table 3: Temperature modeling and thermal management parameters.

Temperature sampling interval	10000 cycles of a 3GHz clock
Thermal threshold	85 Degrees (Celsius)
Nominal frequency	3.0 GHz
Nominal voltage	1.4 Volt
Ambient air temperature	45 Degrees (Celsius)
Convection thermal resistance	0.8 K/W
Convection thermal capacitance	140.4 J/K
Die	0.5 mm thick
Heat spreader	1.0 mm thick, 3 cm × 3 cm
Heat sink	6.9 mm thick, 6 cm × 6 cm
Max temp sensor reading error	0.5 Degrees (Celsius)
Temp sensor resolution	0.5 Degrees (Celsius)

The algorithm used in this paper is a slight modification of that described above, as its first priority is to minimize the performance cost of DTM, with energy efficiency a secondary concern. The algorithm attempts to minimize the performance effects of the simultaneous engagement of DTM and DEM in the same domain. Therefore, once a domain enters the thermal emergency zone for a given application, the modified *Attack/Decay* DEM algorithm is disabled within that domain for the rest of the application run. (In practice, the DEM algorithm could be periodically re-enabled to account for phase behavior.) Thus, geographically, each of the algorithms controls different parts of the chip, with the DTM scheme operating in the domains that contain hot spots, and the DEM algorithm in those that do not. (However, as demonstrated in the next section, the triggering of DEM early in application execution may prevent DTM from ever needing to be engaged.) This non-overlapping of controlled domains avoids complex interactions between these two algorithms and yet achieves both good temperature control and energy efficiency.

To summarize, there are three effects that make localized DVS within an MCD processor efficient: a. it provides a localized, but vigorous, response to the particular area of the die that is undergoing a thermal emergency, permitting unaffected areas to continue to operate at full speed; b. the use of DEM within a particular domain may reduce the number of thermal emergencies in that domain, and the severity of the response that is required by the DTM algorithm; and

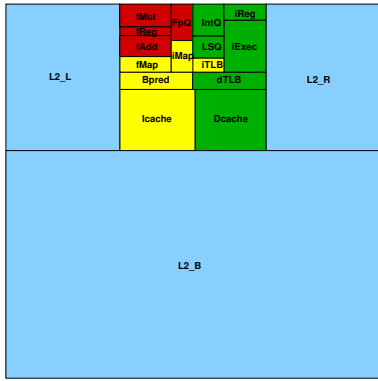


Figure 3: Floorplan (top) and logical domain partitioning (bottom) of the MCD processor.

c. a temperature response in one or two domains may create execution slack in adjacent domains. This permits DVS in these peripheral domains to be engaged, creating lateral heat flow away from the hotspot domains. This in turn permits a gentler response within the hotspot domains, leading to less performance loss.

5. EVALUATION METHODOLOGY

The evaluation methodology uses the MCD simulation framework [15], which is based on the SimpleScalar and Wattch toolkits [2, 3] and the HotSpot temperature modeling tool [9]. The microarchitecture and temperature modeling parameters are shown in Tables 2 and 3. Temperature sensors are placed at all relevant units and assumed to have a maximum reading error of 0.5 degrees Celsius, and a resolution of 0.5 degrees Celsius as well. The maximum voltage is 1.4V and there are 11 frequency levels, ranging from 3GHz to 1GHz. The fastest voltage transition speed is 16.7 mV per μ s. The chip floorplan, and the logical domain partitioning of the MCD processor (proposed in [21]), are shown in Figure 3.

Of the SPEC2000 benchmark programs, the eight with the most severe thermal problems were chosen (refer to Table 1 for their characteristics); the remaining benchmarks generated no or very few thermal emergencies. For each DTM result with each benchmark, three simulation runs were conducted, with each run taking the steady state temperatures from the previous run as the initial temperatures, except for the first run which sets the initial temperature at 80 degrees Celsius and operated without any DTM control. For runs with DTM control, initial temperatures were clipped based on the pre-specified hard limit, which is set at 85 degrees Celsius.

Each benchmark was fast-forwarded 2 billion instructions, followed by the two-phase warm up for 300 million instruc-

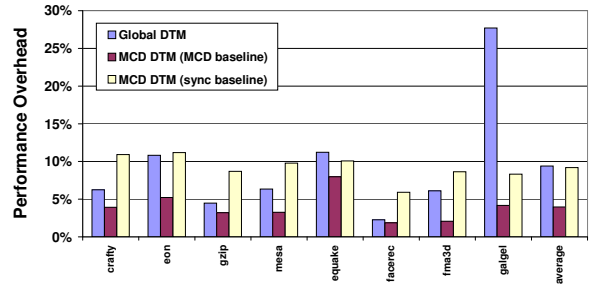


Figure 4: Performance degradation of Global DTM relative to fully synchronous without DTM, MCD DTM relative to MCD without DTM, and MCD DTM relative to fully synchronous without DTM.

tions total as suggested in [18], the warm up of various structures (like branch predictor and caches) for 100 million instructions, and then the warm up of different chip units to reach representative temperature values, for another 200 million instructions. The statistics used to generate the bar graphs in the next section were then gathered for the next 300 million instructions. However, Figures 5 and 7 include the data from the last 500 million instructions of execution.

6. RESULTS

In this section, localized DTM within an MCD processor is first compared with global DTM in a fully synchronous machine without considering DEM. Then, the combined DTM and DEM algorithm is evaluated in Section 6.2.

6.1 Localized Versus Global DVS-based DTM

The DVS-based DTM algorithm described in Section 3 was applied to both fully synchronous and MCD processors (each domain independently implementing the algorithm). Figure 4 shows the corresponding performance degradation. Comparing the bars on the left (fully synchronous microprocessor with DVS-based DTM) with the maximum temperatures in Table 1 shows that for high temperature programs like *galgel*, *eon* and *equake*, the performance cost is high as well, more than 10%. The worst performance cost is 27% for *galgel*, which is the program that has the highest temperature. For lower temperature programs such as *facerec* and *gzip*, the performance cost is also lower, as expected.

The performance cost of using localized DVS within MCD (relative to the baseline MCD machine) is on average 2.5 times less than that of global DVS (relative to the performance of a fully synchronous machine). Since domains in an MCD processor are independent, the performance impact of DVS is largely confined within the domain where the hot spots are located. Maintaining full speed operation in other domains is especially important when one or all of the other domains are very performance critical. If there happens to be slack in the hot spot domain at the time DVS is applied, then the performance loss is further reduced. Due to the lower performance cost of targeted, localized DTM on an MCD machine, even when the inter-domain synchronization performance cost of MCD is accounted for (over 5% on average – see Figure 6), its performance overall is competitive with that of the fully synchronous machine with DTM.

While for all benchmarks the performance cost of localized DTM within MCD is less than global DTM, the difference is particularly striking for *galgel*. Since *galgel* has the most severe thermal problem among all the programs (Table 1), it also requires the largest voltage and frequency reduction

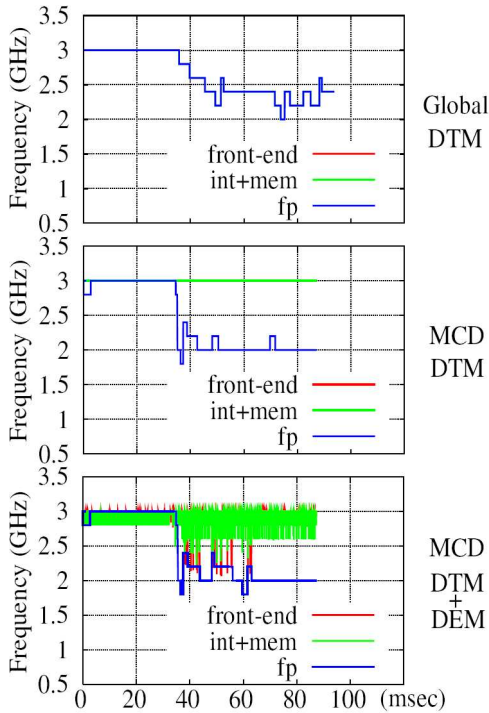


Figure 5: Frequency profiles for *galgel* when running the DTM algorithm on a fully synchronous machine (top, where all three curves overlap), an MCD machine (middle), and an MCD machine with DEM also applied in the non-hotspot domains (bottom).

to maintain acceptable temperatures. As shown in Figure 5, for the MCD DTM case, only the floating point domain frequency is reduced, as the other two domains do not contain hot spots. For Global DTM, all three domain frequencies must be reduced by the same amount. This has a significant performance cost for *galgel* since for this benchmark, all three domains are performance critical, each containing critical paths through the execution dataflow graph. Therefore, the ability to maintain full speed operation in two of the three domains through localized DTM yields a large performance advantage.

One advantage of global DVS is the cooling of neighboring units results in better heat removal from the hotspot due to a steeper temperature gradient; therefore, the voltage does not have to be lowered as much in the affected domain as in the MCD case, as seen for the floating point domain in *galgel* (Figure 5). However, this factor did not have nearly as large a performance impact as the ability to keep the front-end and integer/memory domains running at full frequency.

6.2 Combined DTM and DEM Algorithms

Figure 6 shows the performance overhead and energy savings, relative to the baseline fully synchronous machine without DTM or DEM, of the Global and MCD machines with DTM, and the MCD machine with both DTM and DEM. The last bar in each set shows the performance degradation (due to synchronization) of the baseline MCD machine. (The other MCD bars include this baseline degradation.) For MCD, the combined DTM and DEM approach achieves over a factor of two greater energy savings *and* better performance compared to the use of DTM alone. The energy benefit is a result of the overall lower voltage in all domains:

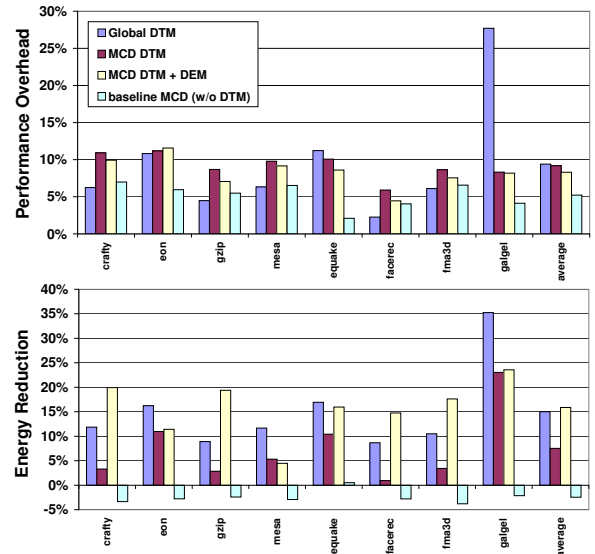


Figure 6: Performance degradation and energy savings for different schemes. The baseline is a fully synchronous machine without any DTM control or energy saving techniques.

in the hotspot domains due to DTM and/or DEM, and in the other domains due to the additional execution slack that can be exploited. The performance benefit comes from the DEM algorithm exploiting the extra slack in non-hotspot domains through DVS, creating a better lateral temperature effect, or from DEM being engaged in the hotspot domains preventing them from heating to the same degree; thus, the DTM algorithm operating in the hotspot domains does not need to reduce voltage and frequency as much to maintain acceptable temperature levels. The only exception is *eon*, which has a slightly higher performance overhead with the combined approach. This is due to the fact that *eon* is the only program among the eight where the DEM algorithm fails to save energy, and therefore the lateral effect is actually slightly degraded. Although the effect is minor, a DEM algorithm based on a more formal control theoretic approach [19, 20] may yield more consistent results.

There are two reasons why DEM lessens the performance impact of DTM. The first is the lateral effect as mentioned previously. Figure 5 shows this effect for *galgel*. In the MCD DTM+DEM case, DEM is activated for both the front-end and integer+memory domains, while DTM operates in the floating point domain. In comparing the floating point frequency curves, once DTM is activated, the floating point frequency for the DTM+DEM case is not as aggressively scaled as compared to DTM alone. From the period in which the floating point frequency drops to 2GHz, to the point where both the DTM and DTM+DEM floating point frequency curves remain flat (at roughly 72ms), the DTM+DEM floating point frequency is about 40MHz higher on average than for DTM alone.

On the other hand, Figure 7 shows the frequency curves for *facerec* for both DTM and DTM+DEM. For the latter, DEM is aggressively engaged immediately at execution, long before the DTM algorithm is engaged. This results in a significant energy savings over the DTM case, thereby obviating the need for DTM to be triggered. This case, in which DEM alone maintains acceptable temperatures (although DTM is of course available should this situation change),

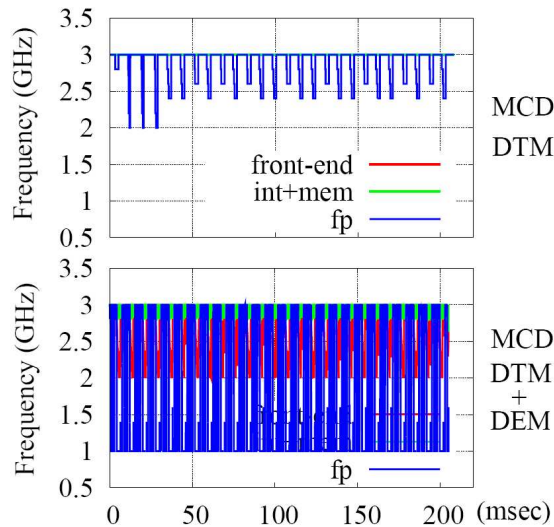


Figure 7: Frequency profiles with DTM alone (top) and with both DTM and DEM (bottom) for *facerec*.

happens for several of the benchmarks, while others benefit more from the lateral effect as in *galgel*.

Compared to DTM on a fully synchronous machine, MCD with DTM and DEM achieves a lower performance overhead and greater energy savings, even with the performance cost of inter-domain synchronization taken into account. While much of the performance benefit comes from *galgel*, most of the benchmarks demonstrate a comparable performance and energy tradeoff with MCD compared to Global. The ability to perform vigorous localized temperature management in MCD, and to exploit the synergy with DEM, provides an advantage in a thermally constrained environment that serves to offset the synchronization cost.

7. CONCLUSIONS AND FUTURE WORK

In this paper, a DVS-based algorithm for localized temperature control in MCD processors was proposed and compared with its use in a conventional fully synchronous design. The ability to provide a focused and vigorous response was shown to have a significantly lower performance cost. Furthermore, the symbiotic relationships between localized DTM and DEM within an MCD processor was examined. Due to several complementary effects, the addition of localized DEM to a localized DTM approach yields a significant performance benefit *and* improves energy efficiency. The use of these complementary techniques permits an MCD processor to be performance competitive with a fully synchronous design in a temperature constrained environment, even when accounting for the synchronization costs.

An interesting area for future work is to explore the integration of localized intra-core DTM and DEM policies with higher level CMP DTM approaches. Furthermore, the robustness of the new DTM algorithm will be compared against other DVS-based DTM approaches.

Acknowledgments

The authors would like to thank Alper Buyuktosunoglu for his guidance and feedback.

8. REFERENCES

[1] D. Brooks and M. Martonosi. Dynamic Thermal Management for High Performance Microprocessors. In *Proc., of the 7th Intl. Symp. on High Performance Computer Architecture*, Jan. 2001.

[2] D. Brooks, V. Tiwari, and M. Martonosi. Watch: A Framework for Architectural-Level Analysis and Optimization. In *Proc., of the 27th Intl. Symp. on Computer Architecture*, Jun. 2000.

[3] D. Burger and T. Austin. The SimpleScalar Tool Set, Version 2.0. Technical Report 1342, Dept. of Computer Science, Univ. of Wisconsin, Jun. 1997.

[4] L. T. Clark. Circuit Design of XScale Microprocessors. In *2001 Symp. on VLSI Circuits, Short Course on Physical Design for Low Power and High Performance Microprocessors*, Jun. 2001.

[5] M. Fleischmann. Crusoe Power Management – Reducing the Operating Power with LongRun. In *Proc., of the HOT CHIPS Symp. XII*, Aug. 2000.

[6] S. Gunther, F. Binns, D. M. Carmean, and J. C. Hall. Managing the Impact of Increasing Microprocessor Power Consumption. In *Intel Technology Journal*, 2001.

[7] S. Heo, K. Barr, and K. Asanovic. Reducing Power Density through Activity Migration. In *Proc., of the 2003 Intl. Symp. on Low-Power Electronics and Design*, Aug. 2003.

[8] M. Huang, J. Renau, S. M. Yoo, and J. Torrellas. A Framework for Dynamic Energy Efficiency and Temperature Management. In *Proc., of the 33rd Intl. Symp. on Microarchitecture*, Dec. 2003.

[9] W. Huang, S. Ghosh, K. Sankaranarayanan, K. Skadron, and M. R. Stan. Compact Thermal Modeling for Temperature-Aware Design. In *Proc., of the 41st Design Automation Conf.*, Jun. 2004.

[10] A. Iyer and D. Marculescu. Power and Performance Evaluation of Globally Asynchronous Locally Synchronous Processors. In *Proc., of the 29th Intl. Symp. on Computer Architecture*, May 2002.

[11] Y. Li, K. Skadron, Z. Hu, and D. Brooks. Performance, Energy and Thermal Considerations for SMT and CMP Architectures. In *Proc., of the 11th Intl. Symp. on High Performance Computer Architecture*, Feb. 2002.

[12] C. H. Lim, W. Daasch, and G. Cai. A Thermal-Aware Superscalar Microprocessor. In *Proc., of the 3rd IEEE Intl. Symp. on Quality Electronic Design*, Mar. 2002.

[13] M. Powell, M. Goma, and T. N. Vijaykumar. Heat-and-run: Leveraging SMT and CMP to Manage Power Density Through the Operating System. In *Proc., of the 11th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems*, Oct. 2004.

[14] H. Sanchez, B. Kuttanna, T. Olson, M. Alexander, G. Gerosa, R. Philip, and J. Alvarez. Thermal Management System for High Performance PowerPC Microprocessors. In *Proc., of the 42nd IEEE Intl. Computer Conf.*, 1997.

[15] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott. Energy-Efficient Processor Design Using Multiple Clock Domains with Dynamic Voltage and Frequency Scaling. In *Proc., of the 8th Intl. Symp. on High Performance Computer Architecture*, Feb. 2002.

[16] K. Skadron. Hybrid Architectural Dynamic Thermal Management. In *Proc., of the 2004 Conf. on Design, Automation and Test in Europe*, Feb. 2004.

[17] K. Skadron, T. Abdelzaher, and M. Stan. Control-Theoretic Techniques and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management. In *Proc., of the 8th Intl. Symp. on High Performance Computer Architecture*, Feb. 2002.

[18] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture: Extended Discussion and Results. Technical Report CS-2003-08, Dept. of Computer Science, Univ. of Virginia, Aug. 2003.

[19] Q. Wu, P. Juang, M. Martonosi, and D. W. Clark. Formal Online Methods for Voltage/Frequency Control in Multiple Clock Domain Microprocessors. In *Proc., of the 11th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems*, Oct. 2004.

[20] Q. Wu, P. Juang, M. Martonosi, and D. W. Clark. Voltage and Frequency Control with Adaptive Reaction Time in Multiple Clock Domain Processors. In *Proc., of the 11th Intl. Symp. on High Performance Computer Architecture*, Feb. 2005.

[21] Y. Zhu, D. H. Albonesi, and A. Buyuktosunoglu. A High Performance, Energy Efficient, GALS Processor Microarchitecture with Reduced Implementation Complexity. In *Proc., of the 2005 IEEE Intl. Symp. on Performance Analysis of Systems and Software*, Mar. 2005.