

Fine-Grained DVFS Using On-Chip Regulators

STIJN EYERMAN and LIEVEN EECKHOUT

Ghent University, Belgium

Limit studies on Dynamic Voltage and Frequency Scaling (DVFS) provide apparently contradictory conclusions. On the one end, early limit studies report that DVFS is effective at large timescales (on the order of million(s) of cycles) with large scaling overheads (on the order of tens of microseconds), and they conclude that there is no need for small overhead DVFS at small timescales. Recent work on the other hand — motivated by the surge of on-chip voltage regulator research — explores the potential of fine-grained DVFS and reports substantial energy savings at timescales of hundreds of cycles (while assuming no scaling overhead).

This paper unifies these apparently contradictory conclusions through a DVFS limit study that simultaneously explores timescale and scaling speed. We find that coarse-grained DVFS is unaffected by timescale and scaling speed, however, fine-grained DVFS may lead to substantial energy savings for memory-intensive workloads. Inspired by these insights, we subsequently propose a fine-grained microarchitecture-driven DVFS mechanism that scales down voltage and frequency upon individual off-chip memory accesses using on-chip regulators. Fine-grained DVFS reduces energy consumption by 12% on average and up to 23% over a collection of memory-intensive workloads for an aggressively clock-gated processor, while incurring an average 0.08% performance degradation (and at most 0.14%). We also demonstrate that the proposed fine-grained DVFS mechanism is orthogonal to existing coarse-grained DVFS policies, and further reduces energy by 6% on average and up to 11% for memory-intensive applications with limited performance impact (at most 0.7%).

Categories and Subject Descriptors: C.4 [Performance of systems]: Design studies

General Terms: Design, Performance, Experimentation

Additional Key Words and Phrases: Energy-efficiency, on-chip voltage regulators, fine-grained DVFS

1. INTRODUCTION

Dynamic Voltage and Frequency Scaling (DVFS) is a widely used power reduction technique: DVFS lowers supply voltage (V) and clock frequency (f) to reduce both dynamic and static power consumption. DVFS is being used in commercial processors across the entire computing range: from the embedded and mobile market up to the server market. Example processors are the Intel XScale processor [Intel 2004], the Transmeta Crusoe [Transmeta Corporation 2001], the AMD Mobile K6 Plus, the Intel Pentium M, the Intel Sossaman and Woodcrest CMP server

Contact information: Stijn Eyerman and Lieven Eeckhout,

ELIS Department, Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium

Email: {seyerman, leeckhou}@elis.UGent.be

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2011 ACM 0000-0000/2011/0000-0001 \$5.00

processors, and the AMD Opteron Quad-Core processor.

Traditionally, DVFS has been studied and applied only at a coarse timescale because of the high overheads DVFS introduces. DVFS typically incurs a time overhead (the time required to scale from one voltage level to another) of tens of microseconds or tens of thousands of processor cycles [Burd and Brodersen 2000]. This high scaling overhead precludes fine-grained DVFS and therefore DVFS is being applied at a coarse timescale only of on the order of milliseconds or millions of processor cycles. Coarse-grained DVFS fits in nicely with the observation made in a theoretical DVFS study by Ishihara and Yasuura [1998] which shows that for computation-dominant applications only two V/f settings and a single scaling point are needed to minimize energy consumption. As such, high-overhead DVFS does not affect overall performance as V/f needs to be scaled only once during program execution. Xie et al. [2005b] generalize this result to memory-bound applications and compound compute/memory-intensive applications, i.e., they find that only a small number of V/f settings and scaling points are required to minimize energy consumption. The conclusion from these works is that DVFS is effective at large timescales with large scaling overheads, and thus there is no need for operating DVFS at small timescales. In fact, Ishihara and Yasuura [1998] conclude that “time and power loss for voltage switching can be neglected”, and Xie et al. [2005b] found through a limit study that “there is no need to a granularity smaller than 1 million instructions”. Many proposals have been made for applying coarse-grained DVFS either through the operating system [Isci et al. 2006], managed runtime system [Wu et al. 2005], compiler [Hsu and Kremer 2003; Xie et al. 2003] or architecture [Hughes et al. 2001; Isci et al. 2006; Semeraro et al. 2002].

However, recent work by Kim et al. [2008] studies the opportunities of on-chip voltage regulators which promise much smaller timing overheads than (traditional) off-chip regulators. Using on-chip regulators, voltage can be scaled in on the order of tens of nanoseconds or multiple tens or hundreds of processor cycles (assuming GHz processors). As part of their study, Kim et al. perform a limit study by computing the optimal V/f setting at a small timescale of hundreds of processor cycles, and they demonstrate energy saving opportunities from fine-grained DVFS compared to coarse-grained DVFS. This result is in apparent contradiction with the earlier work (as discussed above) which states that a small number of scaling points are sufficient to minimize energy consumption, and that switching overhead does not affect DVFS effectiveness.

This observation motivates the work presented in this paper. We start with a limit study in which we study the DVFS energy saving opportunities across a wide range of timescales, while considering a range of scaling speeds. Our results confirm earlier findings — both observations are correct within their scope — and we unify these apparently contradictory observations in a common theory: energy saving opportunities for coarse-grained DVFS are largely unaffected by the timescale and scaling overhead, however, fine-grained DVFS can lead to significant energy saving opportunities for memory-intensive workloads. The timescale at which fine-grained DVFS results in substantial energy saving opportunities is on the order of hundreds of processor cycles, or on the order of the memory access latency. The insight from this limit study is that the benefit from fine-grained DVFS comes from scaling down

the voltage and frequency upon individual off-chip memory accesses. We also find through our limit study that optimal energy saving is achieved for DVFS scaling speeds of 20mV/ns which is conceivable with today’s on-chip voltage regulators. The potential energy saving for memory-intensive workloads is substantial: 12% on average across a range of performance targets for memory-bound applications on an aggressively clock-gated processor.

As a second step and motivated by these insights, the paper then proposes a fine-grained DVFS mechanism that scales down the supply voltage and clock frequency upon individual off-chip memory accesses using on-chip voltage regulators. This fine-grained DVFS mechanism operates at the hardware level triggered by the microarchitecture. The hardware support needed for fine-grained DVFS is very limited: a simple DVFS trigger mechanism scales down and up voltage/frequency upon a long-latency off-chip memory access and involves limited hardware overhead. We explore two mechanisms: a proactive scheme which assumes knowledge of the memory access latencies in order to proactively scale up voltage so that the processor is at the high voltage/frequency operating point by the time a memory access returns from memory; and a reactive scheme in which voltage upscaling is initiated when a memory access returns from memory. We discuss implementation issues for both schemes in the paper. The proactive mechanism leads to an average 12% (and up to 23%) energy reduction at a very small performance degradation of 0.08% on average (at most 0.14%) for memory-intensive workloads on an aggressively clock-gated processor. The reactive mechanism incurs an average 6% performance overhead (and at most 12%) while reducing energy consumption by 23% on average and up to 40% for memory-intensive workloads.

The fine-grained DVFS mechanism is orthogonal to existing coarse-grained DVFS algorithms, leading to multi-timescale DVFS, i.e., there is a fine-grained DVFS mechanism operated at the hardware level and a coarse-grained DVFS policy managed at the system software level. At the smallest timescale, the voltage and clock frequency is varied upon individual long-latency off-chip memory accesses; at a larger timescale, a coarse-grained DVFS policy operates at the millisecond timescale and optimizes system-level objectives (e.g., minimize energy consumption while meeting deadlines, performance targets or priorities). Our experimental results show that fine-grained DVFS complements coarse-grained DVFS and reduces energy by another 6% on average and up to 11% for memory-intensive workloads.

More specifically, this paper makes the following contributions:

- We propose a practical fine-grained DVFS implementation using on-chip regulators. Prior work [Li et al. 2003] in fine-grained DVFS assumed a different DVFS technology (i.e., dual-power-supply networks) and determined DVFS scaling points based on heuristics. The fine-grained DVFS approach proposed in this paper is based on fundamental insight from mechanistic performance modeling, which we show to incur a smaller performance penalty.
- Through a limit study, we determine the optimum voltage switching speed for on-chip regulators and find it to be 20mV/ns in our setup, which is in the range of today’s on-chip regulator designs. This optimum on-chip regulator configuration trades off processor energy saving and performance versus regulator scaling speed and overhead.

—We demonstrate how to cooperate a fine-grained DVFS mechanism (operating at the hardware level and triggered by the microarchitecture) with a coarse-grained DVFS policy (operated at the system software level), which is of particular importance for soft real-time applications.

This paper is organized as follows. We first revisit dynamic voltage and frequency scaling (Section 2). We then perform a DVFS limit study (Section 3) which provides insight into the opportunities of on-chip voltage regulators for fine-grained DVFS. We subsequently describe our fine-grained DVFS mechanism which is derived from insights obtained from analytical performance modeling (Section 4). We quantify the energy savings and performance impact of the proposed fine-grained DVFS mechanism in Section 5. Finally, we describe related work (Section 6) and conclude (Section 7).

2. DVFS

We first revisit how DVFS affects dynamic and static power consumption, and how voltage regulators operate.

2.1 Dynamic power consumption

The dynamic power consumption P_d of a CMOS circuit is determined by:

$$P_d = \alpha \cdot C \cdot V_{dd}^2 \cdot f,$$

with V_{dd} the supply voltage, f the clock frequency, α the switching activity level, and C the capacitance of the circuit. DVS allows one to reduce the supply voltage at run time, thereby reducing dynamic power consumption. However, reducing the supply voltage also increases the device delay, so frequency also needs to be reduced. Supply voltage and clock frequency under DVFS are related following

$$f \propto \frac{(V_{dd} - V_{th})^\beta}{V_{dd}},$$

with V_{th} the threshold voltage, and β a technology-dependent constant (typically around 1.5). Thus, for supply voltages sufficiently larger than the threshold voltage, it follows that clock frequency is roughly proportional to supply voltage. Or, in other words, the dynamic power consumption is proportional to the supply voltage and clock frequency through a cubic relationship, i.e.,

$$P_d \propto V_{dd}^3 \propto f^3.$$

Dynamic voltage and frequency scaling (DVFS) thus reduces dynamic power consumption due to its cubic dependence on the supply voltage and clock frequency. This explains the effectiveness of DVFS for power-efficiency: dynamic power consumption reduces with frequency following a cubic relationship whereas execution time increases following an at most linear relationship.

2.2 Static power consumption

DVFS also affects static (leakage) power consumption. Two major contributors to static power consumption P_s are subthreshold leakage and gate leakage [Kaxiras and Martonosi 2008]. Subthreshold leakage and gate leakage are affected by the

supply voltage V_{dd} , i.e., lowering the supply voltage reduces both subthreshold and gate leakage. When scaling the supply voltage V_{dd} in conjunction with Adaptive Body Biasing (ABB) by controlling the transistor body-source voltage V_{bs} , even higher leakage power savings can be achieved [Martin et al. 2002].

2.3 Supply voltage regulators

Supply voltage regulators enable dynamically changing the V/f setting, and are typically provided off-chip, on the PCB. Scaling the supply voltage (and clock frequency) using off-chip regulators incurs a significant energy overhead and performance overhead [Burd and Brodersen 2000] — the timing overhead for switching the supply voltage is typically on the order of tens of microseconds (or on the order of tens of thousands of processor cycles).

Recently, a surge of work has been done on integrating voltage regulators on-chip [Abedinpour et al. 2007; Hazucha et al. 2005; Kim et al. 2008; Schrom et al. 2007; Shrom et al. 2004; Wibben and Harjani 2007]. On-chip voltage regulators promise much smaller timing overheads than off-chip regulators: scaling the supply voltage now takes on the order of tens of nanoseconds (or multiple tens or hundreds of processor cycles). In addition, on-chip voltage regulators are smaller in size than off-chip regulators. Unfortunately, these benefits are tampered by lower energy-conversion efficiencies compared to off-chip regulators. As such, there are trade-offs between voltage regulator size, V/f scaling speed, and conversion efficiency. And these trade-offs in regulator design also have their implications at the system level in terms of system cost, overall energy saving and overall performance impact. For example, off-chip regulators have a higher conversion efficiency but can be used to drive coarse-grained DVFS only, because of the high timing overheads; on the other hand, on-chip regulators may enable fine-grained DVFS but incur a higher regulator conversion inefficiency.

On-chip regulators come with their own trade-off in terms of conversion efficiency and voltage switching time [Kim et al. 2008]. Conversion efficiency refers to the ratio of the power delivered to the processor to the total power into the regulator — the regulator itself also consumes power while delivering power to its load (the processor). Regulator losses include the switching power and resistive losses due to the power transistors in the regulator in addition to the resistive loss throughout the power delivery network. Voltage switching time refers to how fast the regulator voltage scales to a new voltage level. Conversion efficiency and voltage switching time are not independent of each other. Regulators with higher power transistor switching frequencies are capable of fast voltage scaling (i.e., short switching time) but incur higher regulator loss. Reversely, regulators with lower switching frequencies incur lower regulator loss but their switching time is larger. This trade-off lies at the foundation of the DVFS limit study that we will be presenting in the next section: i.e., without a detailed evaluation in terms of performance and total energy (including processor and regulator energy consumption and DVFS overhead) it is unclear whether the higher regulator loss incurred by fine-grained DVFS (which requires a short switching time) will offset the energy savings in processor energy consumption compared to coarse-grained DVFS which incurs lower regulator losses.

In this paper, we assume that the processor continues its operation during the scaling transient (which is done in the Intel XScale processor [Clark et al. 2001] and

is assumed by Kim et al. [2008] as well). Upon a low-to-high transition, voltage scales first to a higher level — this takes on the order of tens of nanoseconds — and once the voltage is settled at the higher level, clock frequency is changed abruptly. A high-to-low V/f transition first abruptly reduces clock frequency followed by a voltage transition of on the order of nanoseconds. The transition periods at low clock frequency where voltage is higher than nominal, incur additional power being consumed in the processor. This additional power consumption is referred to as DVFS overhead in this paper.

An additional cost during voltage upscaling is the energy needed to charge the transistors' gates to a high voltage level from a low voltage level. However, this cost is very small compared to the amount of energy saved while operating at a low voltage level. For example, scaling up from a low to a high voltage level in the L2 cache accounts for less than 0.2% of the amount of static energy saved at a low voltage level — taking into account the amount of dynamic energy saved would result in an even smaller percentage. Therefore, we do not model this effect in our evaluation.

3. DVFS LIMIT STUDY USING ON-CHIP REGULATORS

To better understand the relationship between the apparently contradictory conclusions reached in prior work, as discussed in the introduction, we now present a DVFS limit study in which we explore the impact of the DVFS timescale and scaling overhead on overall energy efficiency. This limit study is unique compared to prior limit studies [Hsu and Kremer 2003; Ishihara and Yasuura 1998; Kim et al. 2008; Qu 2001; Xie et al. 2003; 2005a] in that it targets on-chip regulators and focuses on the impact of the timescale granularity on DVFS efficacy and its interaction with scaling overhead. However, before presenting the limit study, we first explain our experimental setup.

3.1 Experimental setup

3.1.1 Benchmarks and simulator. We use the SPEC CPU2000 benchmarks; the binaries are highly optimized Alpha binaries (taken from the SimpleScalar website). To limit the simulation time in our experiments, we use representative 100M-instruction simulation points provided by SimPoint [Sherwood et al. 2002]. We use the SimpleScalar/Alpha v3.0 out-of-order simulator for all of our experiments. We assume a contemporary 4-wide superscalar out-of-order processor configuration, see Table I.

Figure 1 shows the non-pipelined fraction of the total execution time, i.e., the fraction of the execution time due to off-chip memory accesses, for all benchmarks on the baseline 2.4GHz processor, i.e., the fraction of the total execution time that is insensitive to DVFS. This graph gives insight into the computation-dominant versus memory-bound behavior of the benchmarks. For the remainder of the paper, we classify all benchmarks with a non-pipelined fraction smaller than 30% as compute-intensive; these are the applications on the lefthand side of the graph from `eon` up to `gap`. All the other benchmarks on the righthand side of the graph are classified as memory-intensive, from `parser` (33%) up to `ampp` (79%).

ROB	128 entries
LSQ	64 entries
processor width	decode, dispatch, issue and commit 4 wide fetch 8 wide
latencies	load (2), mul (3), div (20)
L1 I-cache	32KB 4-way set-assoc, 1 cycle
L1 D-cache	32KB 4-way set-assoc, 1 cycle
L2 cache	unified, 2MB 8-way set-assoc, 10 cycles
main memory	70ns access time
branch predictor	hybrid bimodal/gshare predictor
frontend pipeline	5 stages

Table I. Processor model assumed in our experimental setup.

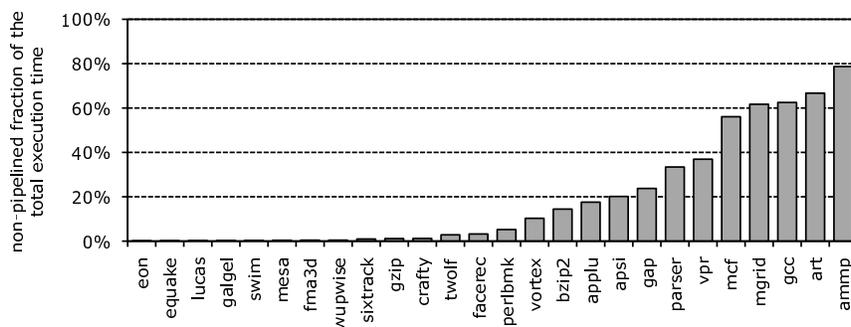


Fig. 1. Non-pipelined fraction of the total execution time on the baseline 2.4GHz processor; benchmarks are sorted by this fraction from left to right.

V	f
1V	2.4GHz
0.88V	1.8GHz
0.76V	1.2GHz
0.64V	0.6GHz

Table II. The V/f settings considered in this paper.

3.1.2 *Power model.* The dynamic power model is taken from Wattch v1.02 [Brooks et al. 2000]. We consider two power modes: (i) *cc0* which assumes that all processor structures consume dynamic power every clock cycle, i.e., there is no clock gating, and (ii) *cc1* which assumes that unused processor structures consume no dynamic power consumption, i.e., the processor is aggressively clock-gated. A real processor may not be fully clock-gated but may exhibit some degree of clock-gating; in other words, we expect performance and energy results for a real processor to lie between the non-clock-gated and clock-gated processor performance and energy numbers presented in this paper. The static power model is taken from HotLeakage [Skadron et al. 2003]. Further, we assume a 70nm CMOS chip technology, a nominal $V_{dd,n} = 1V$ supply voltage and a nominal $f_n = 2.4GHz$ clock frequency.

Static power varies between 44% of the total power consumption (at the 1V and 2.4GHz operating point) to 63% (at the 0.64V and 0.6 GHz operating point).

3.1.3 Voltage regulators. In our setup, similar to Kim et al. [2008], we assume an off-chip regulator that scales down the supply voltage from 3.7V to 1.8V. On-chip regulators scale down the 1.8V supply to a range of 0.64V to 1V. The motivation for this two-step voltage scaling setup is the inherent degradation in conversion efficiency for large step-down ratios. The various V/f settings are shown in Table II.

On-chip regulator loss depends on the power transistor switching frequency (which also determines the voltage switching speed), the output voltage of the regulator, and processor activity. In particular, regulator loss increases with voltage switching speed (see middle graph in Figure 2); regulator loss decreases and regulator efficiency increases with higher output voltages, higher processor activity and slower switching speeds (see top and bottom graphs in Figure 2). In our experiments, we model regulator loss' dependence on voltage switching speed, output voltage and processor activity following the data presented by [Kim et al. 2008]. Figure 2 shows these dependences along with the Kim et al. data and the interpolations that we performed to get a picture of the entire on-chip regulator design space.

3.2 Limit algorithm

The input to the DVFS limit algorithm is a series of execution time and energy consumption numbers per time interval at multiple V/f settings. The smallest time interval is 20 cycles; larger time intervals are obtained by aggregating the 20-cycles time intervals. The DVFS limit algorithm then determines the appropriate V/f setting per interval to arrive at a globally optimal energy versus performance solution. More specifically, the algorithm searches for the optimal V/f setting per time interval that minimizes overall energy consumption while respecting a preset overall performance target. The algorithm takes into account total energy consumption which includes (i) dynamic and static power consumption by the processor, (ii) regulator loss (which depends on voltage switching speed, regulator output voltage and processor activity), and (iii) DVFS overhead or energy overhead due to scaling voltage and frequency.

In our work, our limit algorithm is inspired by the linear-time heuristic algorithm proposed by Xie et al. [2005a]; these researchers also propose an exact algorithm, however, it is much more time consuming than the linear-time heuristic algorithm — we confirmed and verified that the linear-time heuristic algorithm yields comparable results to the exact algorithm. The linear-time algorithm computes the optimal V/f settings per interval for a single performance target, however, in our limit study we are interested in evaluating DVFS across multiple performance targets, i.e., we are interested in computing so called Pareto-optimal solutions, or solutions for which there is no other solution that performs better in terms of both energy and performance. One way of computing these Pareto-optimal solutions is to run the Xie et al. algorithm multiple times for each performance target. A more practical solution (which is what we did) is to extend the Xie et al. algorithm to compute multiple solutions from a single sweep through the program execution by keeping track of multiple intermediate solutions. The additional state that needs to be kept track of is limited according to our experiments which makes it feasible to do in

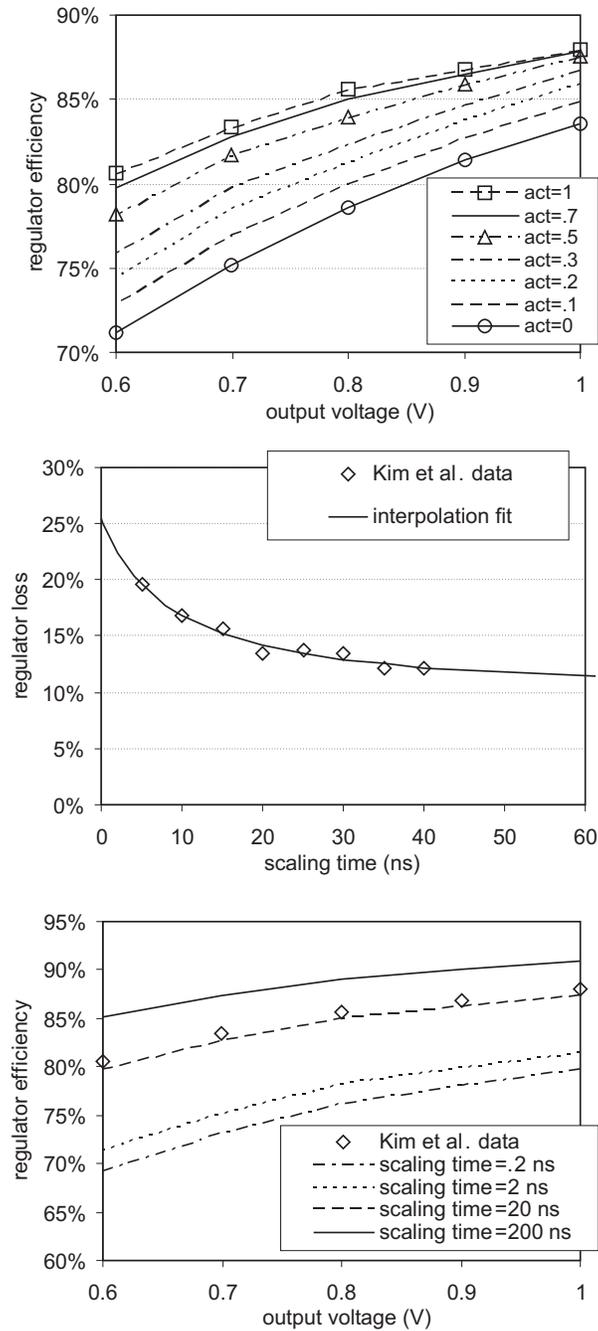


Fig. 2. On-chip voltage regulator characterization: regulator loss and efficiency as a function of processor activity, voltage switching time and output voltage.

practice.

3.3 Results

Using the limit algorithm and the setup outlined above, we now explore the impact of the DVFS timescale granularity and scaling speed on energy efficiency. Energy efficiency in this study is quantified using a Pareto frontier which represents multiple optimal solutions in terms of energy versus performance. For constructing the Pareto frontier we consider 5 performance targets uniformly spread between the fastest solution (at the highest V/f setting) and the middle point between the fastest and slowest solution; and for each of these performance targets we then determine the optimal DVFS scheduling that minimizes energy consumption using the DVFS limit algorithm explained above. Figure 3 shows the Pareto frontiers for the `ammp`, `parser` and `eon` benchmarks; the two curves in each graph represent ‘fine-grained DVFS’ assuming on-chip regulators, a 20-cycle timescale granularity and a 20mV/ns voltage switching speed, versus ‘coarse-grained DVFS’ assuming an off-chip regulator¹ and a million-cycle timescale granularity. The vertical axis represents normalized energy consumption (which includes dynamic and static power consumption, regulator loss and DVFS overhead); the horizontal axis represents normalized execution time. These graphs clearly illustrate that fine-grained DVFS leads to substantially lower energy consumption compared to coarse-grained DVFS for memory-intensive workloads. Energy savings through fine-grained DVFS go from 42% for `ammp` to 11% for `parser` for the same level of performance as coarse-grained DVFS. For compute-intensive applications (see `eon`), we observe slightly higher energy consumption (because of regulator loss). In what follows, we will quantify energy efficiency as the average normalized energy consumption across these five Pareto-optimal solutions compared to coarse-grained DVFS. The reason for taking the average across multiple Pareto-optimal solutions instead of a single operating point is to cover a broad range of trade-offs in terms of performance versus energy.

The graphs in Figures 4 and 5 explore the impact of DVFS timescale granularity versus voltage switching speed assuming a clock-gated and non-clocked-gated processor, respectively. These graphs show average normalized energy consumption on the vertical axis versus voltage switching speed (in mV/ns) on the horizontal axis for different timescale granularities (in clock cycles) — the different curves in Figures 4 and 5 show different timescales and are limited to at most 10,000 cycles for readability (larger timescales show the same result as 10,000 cycles). Energy consumption is barely affected by the timescale granularity for the compute-intensive workloads (see the top graphs in Figures 4 and 5). However, energy consumption increases substantially with increasing voltage switching speeds. This increase is due to increasing regulator loss as a result of higher voltage switching speeds. The picture is very different for memory-intensive workloads (see the bottom graphs in Figures 4 and 5). Energy efficiency is very sensitive to timescale granularity, especially for relatively high voltage switching speeds (higher than 5mV/ns). A DVFS timescale granularity on the order of 20 cycles can lead to dramatic reductions in energy consumption, 18% on average for a non-clock-gated processor and around

¹Regulator loss of approximately 10%.

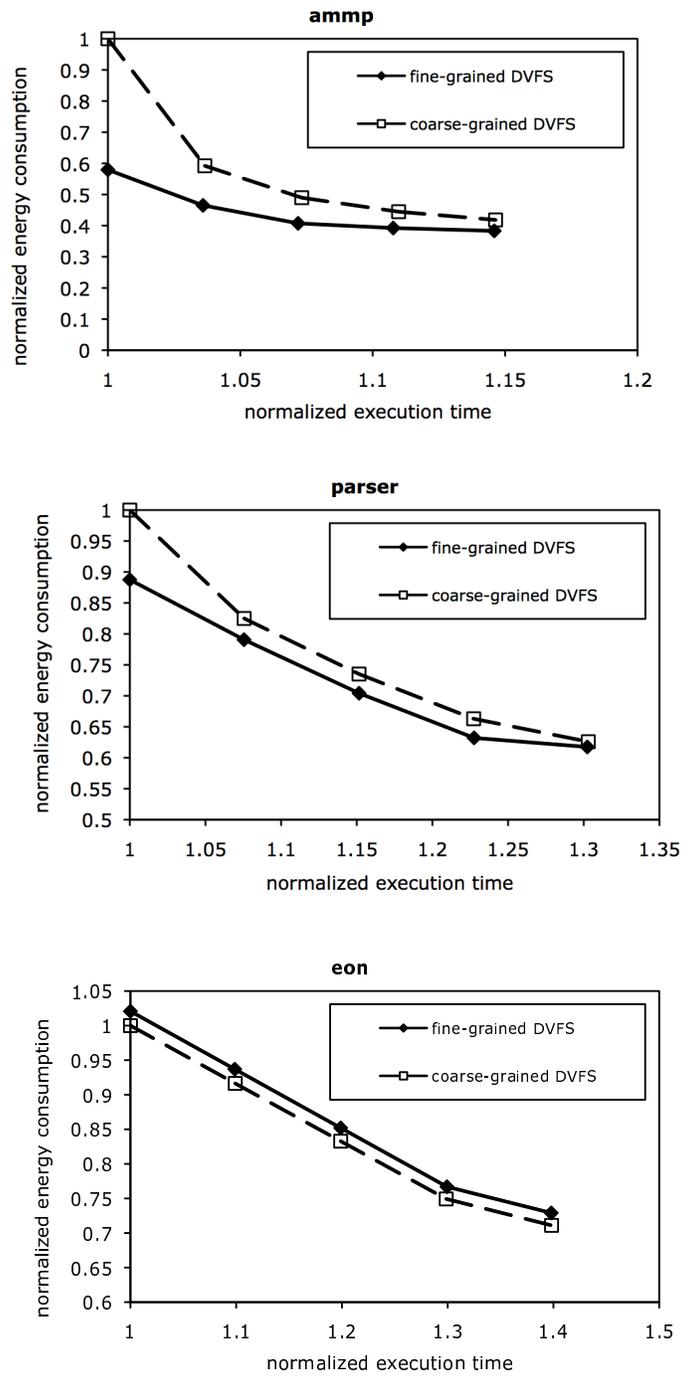


Fig. 3. Pareto frontiers for ampp (top), parser (middle) and eon (bottom): energy versus performance.

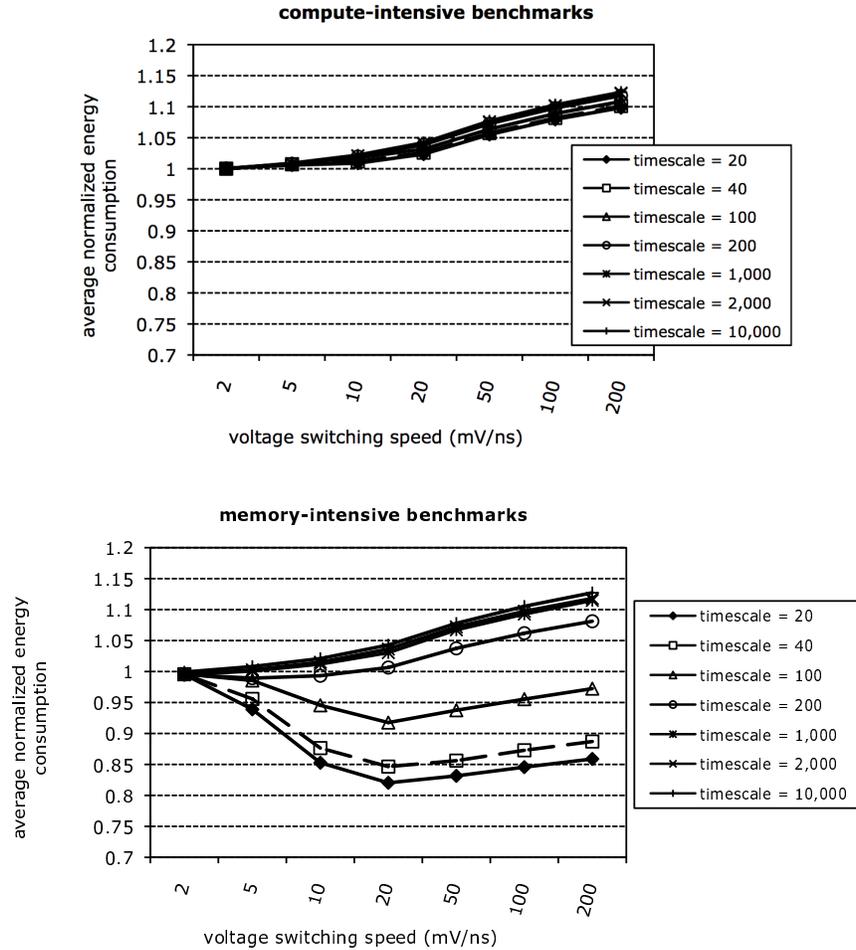


Fig. 4. Normalized energy consumption as a function of DVFS timescale and voltage switching speed for a non-clock-gated processor: compute-intensive workloads (top graph) and memory-intensive workloads (bottom graph).

12% on average for a clock-gated processor.

The key insight from these graphs is that the substantial reduction in energy consumption at small time scale granularities comes from scaling voltage and frequency upon individual off-chip memory accesses. Accessing off-chip memory takes 70ns or around 170 cycles for the 2.4GHz processor configuration, and small timescale DVFS exploits the fine-grained slack due to memory accesses. Interestingly, there is an optimum that balances regulator loss and processor energy saving as a function of voltage switching speed. This optimum appears around 20mV/ns, which is the range of feasible on-chip regulator designs according to Kim et al. [2008].

These results confirm and unify the (apparently) contradictory results reported in prior research on DVFS: energy-efficiency is largely insensitive to the timescale in

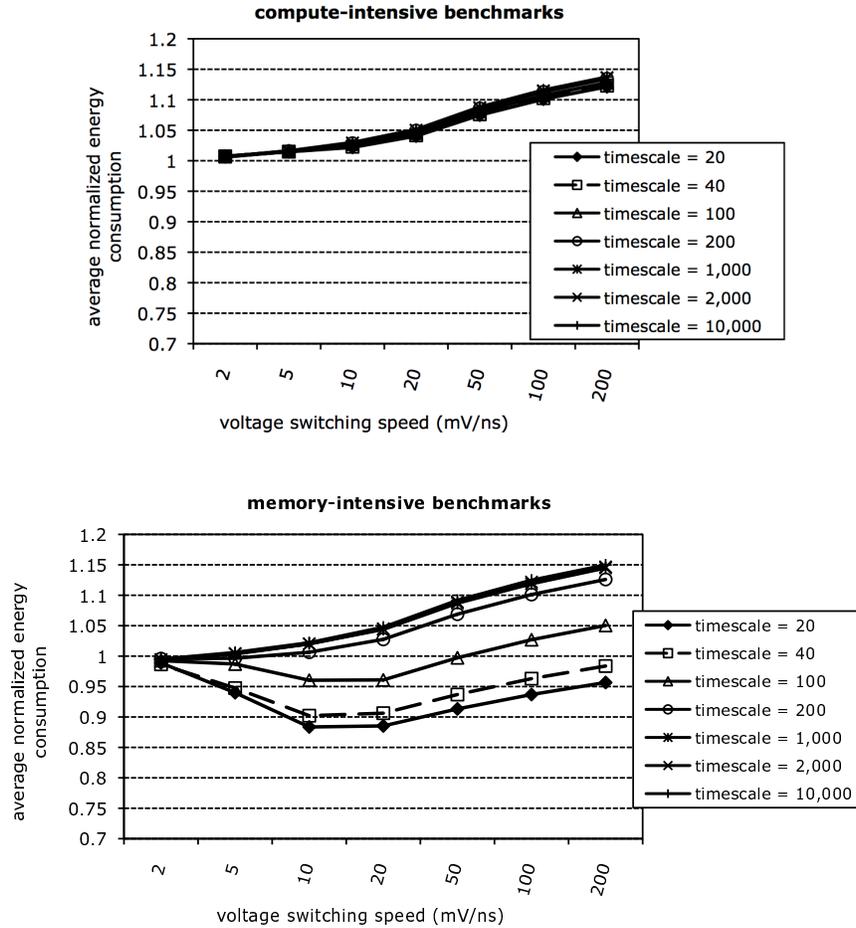


Fig. 5. Normalized energy consumption as a function of DVFS timescale and voltage switching speed for a clock-gated processor: compute-intensive workloads (top graph) and memory-intensive workloads (bottom graph).

case of slow voltage switching speeds, however, there are significant energy saving opportunities for memory-intensive applications through fine-grained DVFS at the timescale of tens or hundreds of processor cycles, i.e., when applying DVFS at the granularity of individual off-chip memory accesses. The switching speeds offered by existing on-chip regulators are in a speed range that enable dramatic energy savings for memory-intensive workloads. This insight motivates the fine-grained DVFS mechanism as described in the next section.

4. FINE-GRAINED DVFS

The idea behind the proposed fine-grained DVFS mechanism is to scale down the V/f setting at the granularity of individual off-chip memory accesses, i.e., we scale

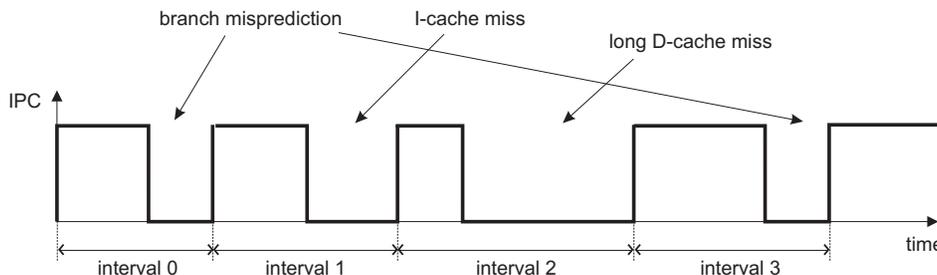


Fig. 6. Performance can be analyzed by dividing time into intervals between miss events.

down the V/f setting upon an off-chip memory access, and when the data returns from memory, we scale up again. Given the high scaling speeds of on-chip regulators, scaling up and down can be done on the order of multiple tens or hundreds of processor cycles. Fine-grained DVFS requires a microarchitecture mechanism that triggers the voltage regulator to scale up and down the V/f setting. For explaining the fine-grained DVFS mechanism we focus on the main off-chip access contributors and make a distinction between off-chip memory accesses due to instruction cache misses and off-chip data cache misses due to loads; TLB misses incur similar overheads, and we therefore treat them collectively with off-chip cache misses.

For deriving when to scale up and down, we build on a recently proposed mechanistic superscalar processor performance model, called interval analysis [Eyerma et al. 2009]. The basis for interval analysis is the observation that, in the absence of miss events such as branch mispredictions and cache misses, a well-balanced superscalar out-of-order processor should smoothly stream instructions through its pipelines, buffers, and functional units. Under ideal conditions the processor sustains a level of performance (instructions per cycle) roughly equal to the superscalar dispatch bandwidth. (We define dispatch as the stage where instructions make it from the front-end of the pipeline into the reorder buffer and issue queues.) However, the smooth dispatch of instructions is intermittently disrupted by miss events. The effects of these miss events at the dispatch stage divide execution time into intervals, and these intervals serve as the fundamental entity for analysis and modeling, see Figure 6. Interval analysis thereby provides a way of visualizing the significant performance events that take place in an out-of-order processor without requiring detailed tracking of extremely large numbers of individual instructions.

4.1 Off-chip instruction cache misses

Figure 7(a) shows a schematic drawing of the timing behavior for an off-chip memory access due to an instruction cache miss. The vertical axis shows the processor's dispatch behavior as a function of time on the horizontal axis. Initially, the processor dispatches D instructions per cycle from the front-end pipeline into the reorder buffer and issue queues. At some point, an off-chip instruction cache miss occurs. Then it takes L cycles before dispatch stops, with L the number of front-end pipeline stages, i.e., the front-end pipeline depth. In the meanwhile, the off-chip memory access is being handled. When the off-chip memory access returns, the processor will resume fetching instructions, and L cycles later, dispatch will resume. The penalty

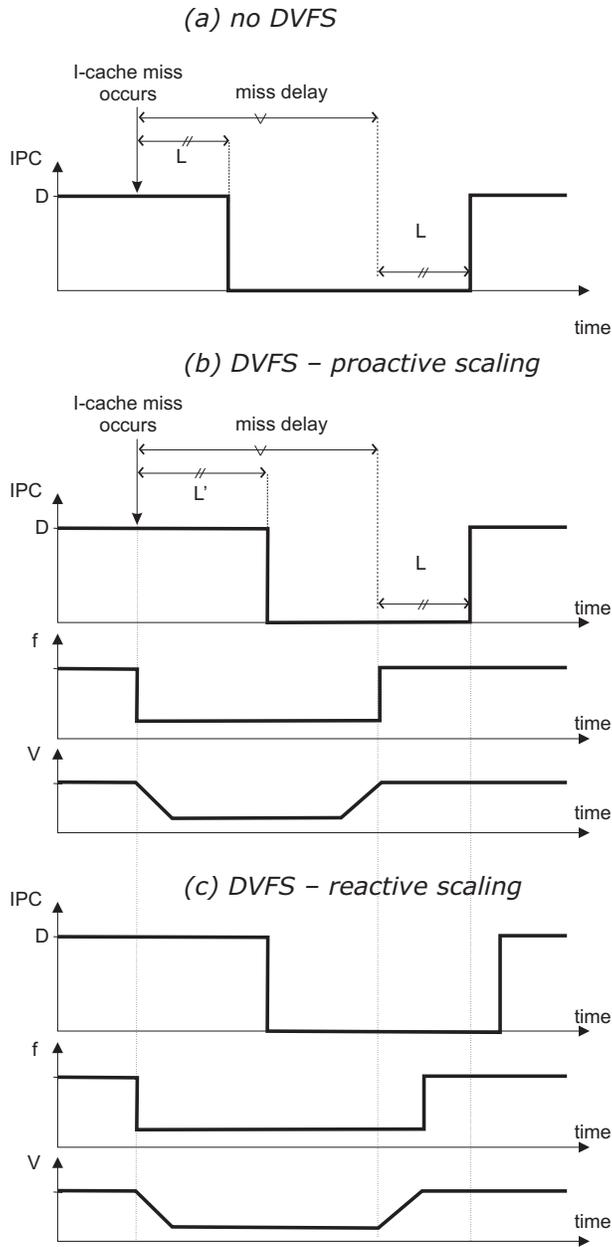


Fig. 7. Timing behavior of an off-chip instruction cache miss: (a) assuming no DVFS, (b) assuming DVFS and proactive scaling, and (c) DVFS and reactive scaling.

for an off-chip instruction cache miss thus equals the off-chip memory access latency as draining and refilling the front-end pipeline offset each other.

The fine-grained DVFS mechanism scales down the V/f setting as soon as the off-chip memory access initiates, and scales up again when the off-chip memory access returns from memory. Figure 7(b) and (c) shows the timing, and voltage and frequency scaling behavior. The clock frequency is scaled down abruptly when the off-chip memory access is initiated, and voltage is scaled down subsequently during a transient period. A consequence of the abrupt frequency scaling is that it takes longer to drain the front-end pipeline, however, this does not affect overall performance because the front-end pipeline drain is hidden underneath the memory access. When the off-chip memory access is about to return, voltage is scaled up again during a transient, and when the voltage reaches its high level, frequency is scaled up abruptly. Ideally, frequency is at its high level when the front-end pipeline is about to be refilled with instructions; this is referred to as *proactive scaling*, see Figure 7(b). This assumes perfect knowledge about the timing of off-chip memory accesses because voltage needs to be scaled up first to reach its high level. An alternative and more pragmatic approach is to initiate the voltage transient when the memory access has returned (*reactive scaling*), see Figure 7(c); this however incurs a performance penalty because refilling the front-end pipeline will not be done at the highest frequency. In the evaluation section of this paper, we will compare proactive against reactive scaling.

4.2 Off-chip load misses

Off-chip load cache misses are more complicated to handle, and we therefore make a distinction between an isolated off-chip load miss and overlapping off-chip load misses.

4.2.1 Isolated long-latency load. Figure 8(a) shows the timing behavior for an isolated load miss. At some point in time, the off-chip memory load access is initiated. Underneath the handling of the off-chip memory access, the processor will continue dispatching instructions until either (i) the reorder buffer completely fills up and the long-latency load blocks the head of the reorder buffer, (ii) the issue queues fill up because of instructions that are dependent on the long-latency load, and (iii) the number of rename registers gets exhausted. Eventually, dispatch ceases for a long period of time while the off-chip memory access is being handled. When the data gets back from memory, dispatch resumes.

4.2.2 Overlapping long-latency loads. Figure 8(b) shows the timing behavior for two independent overlapping long-latency load misses. At some point while the first long-latency load miss is being serviced, the second miss will occur. Both miss penalties will overlap, and memory-level parallelism (MLP) gets exposed [Chou et al. 2004; Glew 1998; Karkhanis and Smith 2004], i.e., the memory access latency of the second load miss is hidden underneath the first load miss. This observation generalizes to multiple independent long-latency misses.

Because of overlapping long-latency load misses, scaling voltage and frequency as soon as the first miss occurs is likely to incur a performance penalty when the miss returns, i.e., the subsequent independent long-latency load misses would be issued later and would return later as well thereby prolonging the total execution time.

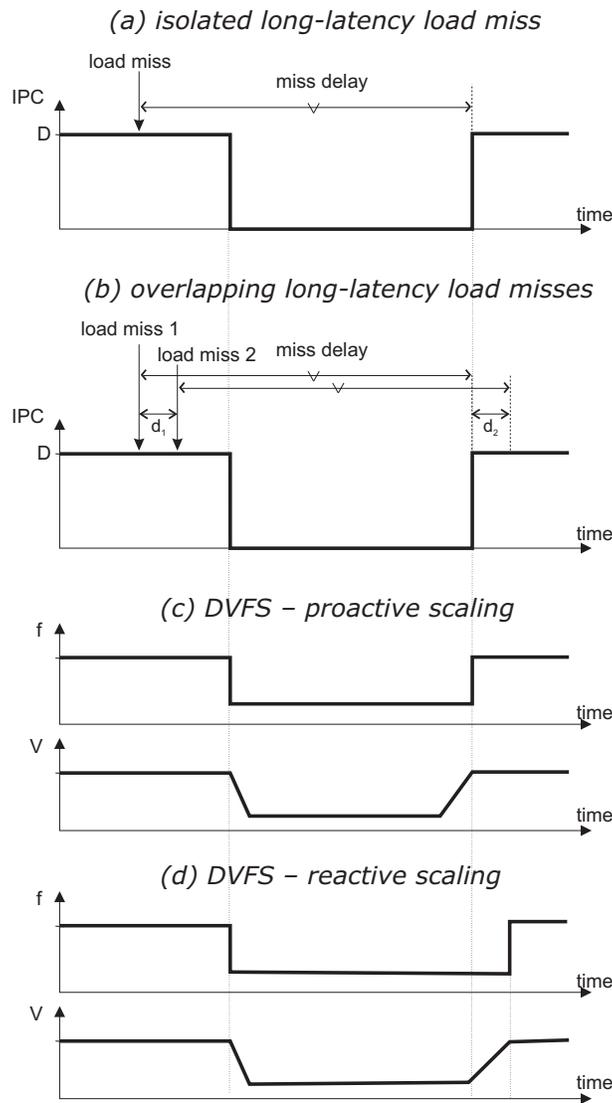


Fig. 8. Timing behavior of (a) an isolated long-latency load miss, (b) overlapping long-latency load misses, (c) DVFS under proactive scaling, and (d) DVFS under reactive scaling.

Instead, we scale voltage and frequency when dispatch stalls, see Figure 8 (c) and (d), in which case the independent long-latency load misses will issue faster, and thus return faster from memory. Similarly as for I-cache misses, scaling up can be done in a proactive way or reactive way, see Figure 8 (c) and (d), respectively.

4.3 Hardware mechanism

Based on the above, we can now describe our hardware mechanism for driving reactive fine-grained DVFS. There are two hardware monitors for scaling voltage and frequency. The one monitor observes the front-end pipeline. In case of an off-chip instruction cache miss, the monitor triggers DVFS as soon as the miss occurs. The second monitor observes the dispatch stage of the pipeline. When dispatch stalls upon an off-chip load miss at the head of the reorder buffer, the second monitor triggers DVFS, and the voltage and frequency are scaled down. Waiting until dispatch stalls upon an off-chip load miss at the head of the reorder buffer naturally handles isolated and overlapping off-chip load misses in a simple but effective manner. For handling overlaps between off-chip load misses and instruction cache misses, we ramp up voltage and frequency upon the first off-chip miss returning from memory. The hardware cost for these fine-grained DVFS monitors is very limited; both monitors are trivial to implement.

As alluded to before and as will be shown in the next section, reactive scaling incurs some performance penalty. A proactive mechanism does not incur overhead, however, it requires that the memory access latency is known. One possible implementation of a proactive scheme involves communication with the memory controller. The memory controller, which is an on-chip structure in most contemporary processors, accepts memory requests from the CPU and schedules them to the memory banks in a way that the timing constraints of the DRAM banks are satisfied. A memory request can reside some time in the memory controller until it is scheduled, but as soon as the memory controller sends a memory access to a memory bank, the memory controller knows how long the memory access will take; the outstanding memory access time depends on the current row buffer content². The memory controller communicates the outstanding memory access latency to the DVFS mechanism, which is then capable of scheduling a voltage ramp-up on time (or at least sooner than would be the case for reactive scaling) in case the remaining memory latency is smaller than the ramp-up time, e.g., on a row buffer hit. Another way of implementing proactive scaling is off-chip memory access latency prediction. The work by Mutlu and Moscibroda [2007] may be a good first step towards this goal.

5. EVALUATION

For evaluating the proposed fine-grained DVFS mechanism, we consider two scenarios to demonstrate the general applicability of fine-grained DVFS. The first scenario targets general-purpose applications, and the goal of fine-grained DVFS is to minimize energy consumption with no (or very limited) impact on performance. The second scenario targets (soft) real-time applications for which there is a coarse-grained DVFS policy in place for meeting the deadlines while reducing energy consumption by exploiting slack. Employing fine-grained DVFS in conjunction with coarse-grained DVFS leads to a multi-timescale DVFS system: the fine-grained DVFS mechanism operating at the hardware level further reduces energy

²This is a simplifying assumption. Large-scale systems with multiple chips in the same coherency domain may not satisfy this property. Memory access time after an on-chip memory controller issues a request may not be fixed.

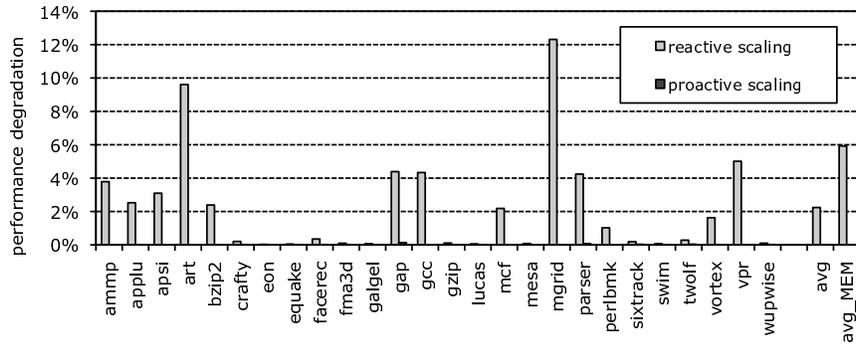


Fig. 9. Performance degradation for fine-grained reactive and proactive DVFS.

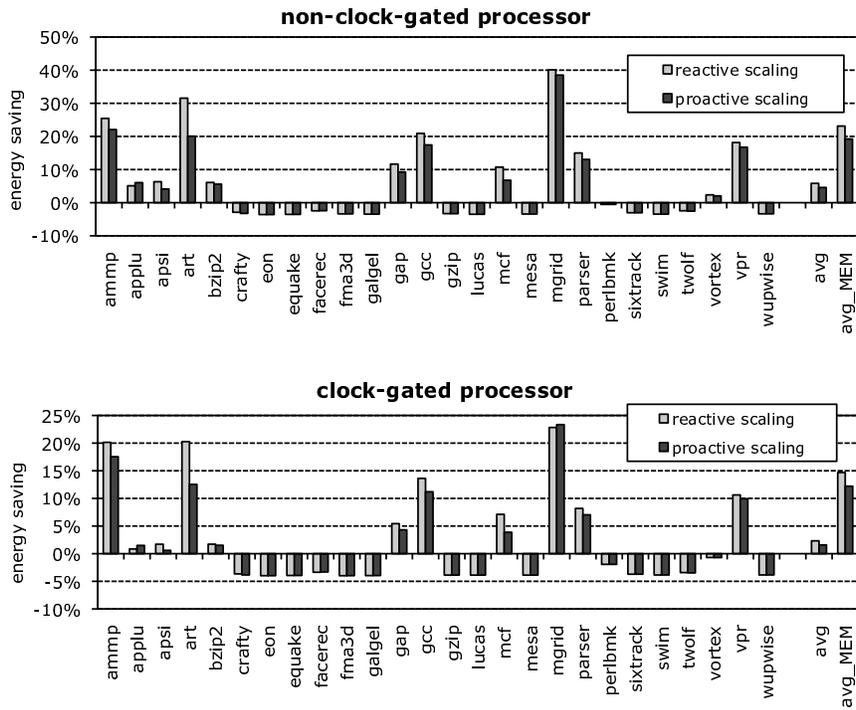


Fig. 10. Normalized energy consumption for fine-grained reactive and proactive DVFS; non-clock-gated processor (top graph) and clock-gated processor (bottom graph).

consumption upon the coarse-grained DVFS policy which operates at the system software level.

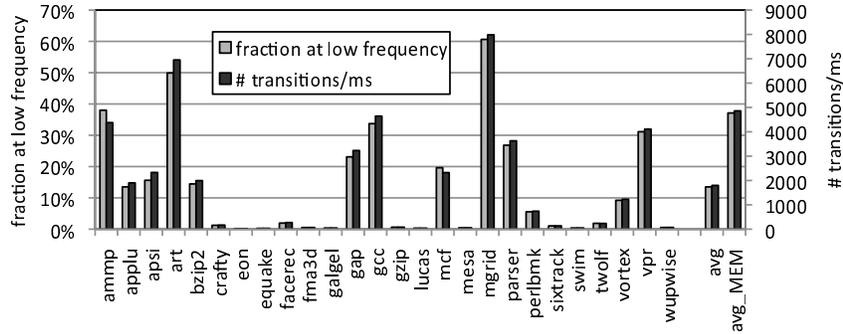


Fig. 11. The fraction of time frequency is set at the lowest level and the number of transitions (low to high) per millisecond.

5.1 Fine-grained DVFS

For the evaluation of fine-grained DVFS for general-purpose applications, we assume that the processor runs at the highest V/f setting throughout the entire execution and scales down to the lowest V/f setting upon off-chip memory accesses. We then compare fine-grained DVFS against a benchmark run at the highest V/f setting, i.e., no DVFS. The on-chip regulator for fine-grained DVFS is assumed to be relatively fast (20mV/ns); for the no DVFS case, we assume an off-chip regulator. Figures 9 and 10 quantify performance and energy, respectively, for fine-grained DVFS and both proactive and reactive scaling against no DVFS. Proactive scaling incurs a very small performance degradation only (0.08% on average and at most 0.14%), and energy savings are substantial: 6% on average and up to 39% for the non-clock-gated processor, and 1.6% on average and up to 23% for the clock-gated processor. For the memory-intensive workloads, we obtain an average energy reduction of 12% and 19% for a clock-gated and non-clock-gated processor, respectively. Compute-intensive workloads on the other hand, incur a small energy increase due to regulator loss, at most 3.6% and 4% for the non-clock-gated and clock-gated processor, respectively. This illustrates the large potential energy savings through fine-grained DVFS, i.e., across this set of benchmarks, energy savings through fine-grained DVFS offset regulator loss.

A limitation with proactive scaling though is that it requires perfect miss delay knowledge which may be difficult to achieve in practice because of non-constant miss delays. Initiating the scale-up earlier than perfect will not hurt (or improve) performance but will likely reduce energy savings. Initiating the scale-up later than perfect on the other hand will degrade performance (but will likely increase energy savings). The latest point for scale-up is represented by reactive scaling: the maximum performance degradation to be expected is no more than 2.2% on average (6% for memory-intensive workloads) and at most 12% (mgrid), see Figure 9. Energy savings are substantial under reactive scaling: 2.3% and 5.8% on average and up to 23% and 40% for the clock-gated and non-clock-gated processor, respectively; for memory-intensive workloads, the average energy saving is around 15% and 23% for the clock-gated and non-clock-gated processor, respectively, see Figure 10.

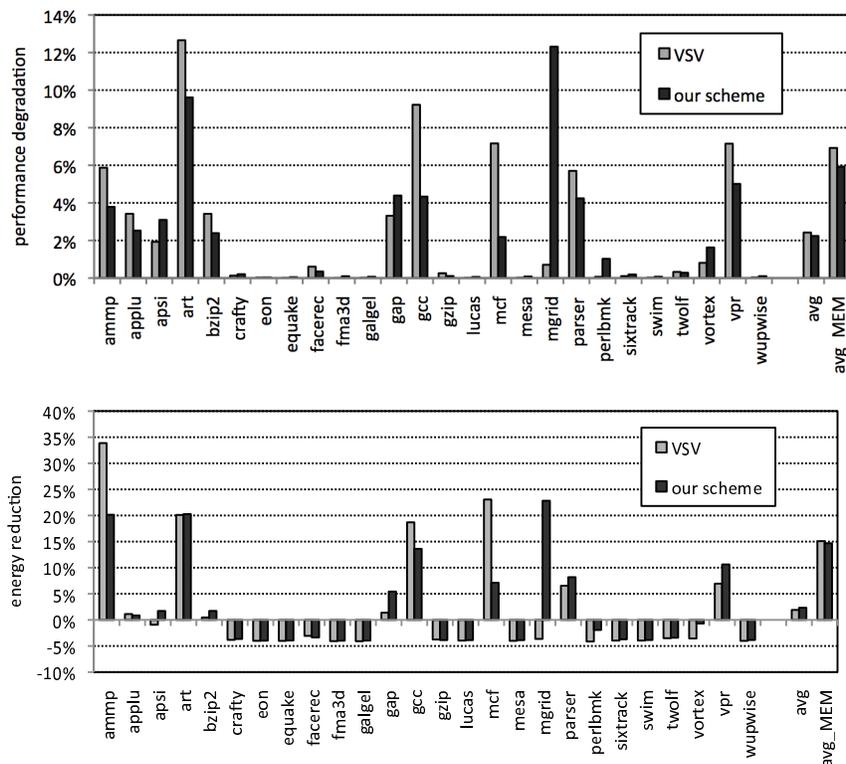


Fig. 12. Comparison to VSV in terms of performance (top graph) and energy (bottom graph).

Figure 11 quantifies the fraction of time spent at the lowest voltage/frequency operating point during fine-grained DVFS, as well as the number of transitions per millisecond to the lowest operating point. Memory-intensive workloads spend a substantial fraction of their time at the lowest operating point, 36% on average and up to 60%.

5.2 Comparison to prior work

Li et al. [2003] propose variable supply-voltage scaling (VSV), a mechanism that reduces voltage and frequency on the occurrence of L2 misses using a dual-power-supply network. They employ two finite state machines that scale the supply voltage based on the amount of instruction-level parallelism (ILP) in the instruction stream. When an L2 cache miss is detected, instruction issue is monitored for the next 10 cycles. If no instructions issue in 3 consecutive cycles, ILP is assumed to be low and voltage is scaled down. When the miss returns from memory, voltage is scaled up again if there was only one outstanding miss; in case of multiple outstanding misses, voltage is scaled up if, during the next 10 cycles, there are at least 3 cycles during which at least one instruction was issued.

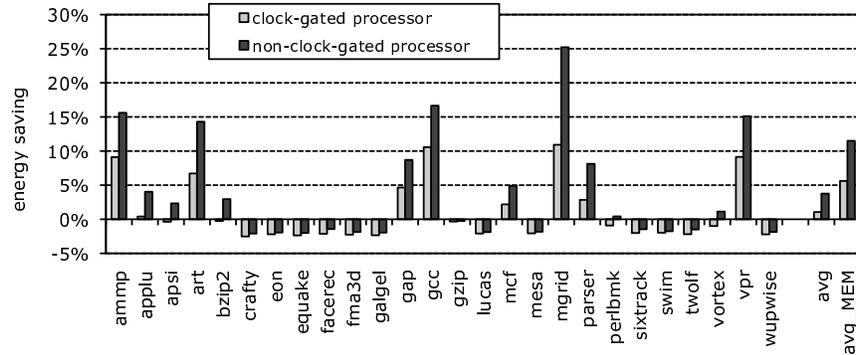


Fig. 13. Energy saving through multi-timescale DVFS compared to coarse-grained DVFS.

Figure 12 shows the performance decrease and energy reduction of VSV and our mechanism using reactive scaling (both using on-chip voltage regulators), compared to not using DVFS. The performance overhead is higher for VSV on average³, while the energy reduction is comparable. The reason for the higher performance hit is due to the delay induced by the finite state machines: VSV scales up (and down) frequency later than our approach, which explains the larger performance penalty. In addition, our proposal has the potential to use memory latency information to implement proactive scaling with almost no performance penalty, which is not done in VSV.

5.3 Multi-timescale DVFS

To demonstrate that fine-grained DVFS complements coarse-grained DVFS, we now employ our fine-grained DVFS mechanism in addition to a coarse-grained DVFS policy similar to the one proposed by Xie et al. [2005b]. This leads to multi-timescale DVFS, i.e., fine-grained DVFS operates at the hardware level and coarse-grained DVFS operates at the system level. The coarse-grained DVFS policy works as follows. We compute the pipelined and non-pipelined fraction of the application ($t_{pipelined}$ and $t_{non-pipelined}$, respectively) through profiling by running the application at frequency f_0 . The lowest frequency f to meet the deadline $t_{deadline}$ can then be found by solving the equation $t_{deadline} = t_{pipelined} \frac{f_0}{f} + t_{non-pipelined}$; frequency f is the optimal frequency for meeting the deadline. We then select the two V/f operating points that are just below and above the optimal frequency, and determine the time that should be spent at each of the frequencies (using the Ishi-

³With mgrid as an extreme counter example: mgrid has very dense dependency paths with many L2 misses. This means that it takes longer for an L2 miss to reach the head of the ROB (because the instructions before the L2 miss take longer to execute due to the long chains of dependent instructions), and that the time between the L2 miss hitting the head of the ROB and the memory operation returning is short. The time period where frequency can be scaled down is thus short, which implies that the relative performance impact of scaling up late (reactive scaling) is larger. VSV tackles this situation by detecting ILP after the L2 miss and thus it does not scale down frequency. This performance hit does not occur in case of proactive scaling.

hara and Yasuura [1998] approach). Per interval of 1M cycles, the non-pipelined fraction of the next interval is predicted (using a last-value predictor; more complicated predictors were only marginally more accurate), and if the non-pipelined fraction is higher than average, the next interval is executed at the lowest frequency, otherwise the highest frequency is chosen. Like that, memory-intensive phases of the program execution are run at a lower frequency, reducing the useless CPU power spent during memory accesses. If the time spent at one frequency equals the calculated total time that should be spent at that frequency, the remainder of the program is executed at the other frequency. Figure 13 reports the additional energy savings through multi-timescale DVFS compared to coarse-grained DVFS. Fine-grained DVFS reduces energy by 6% on average and up to 11% compared to coarse-grained DVFS for memory-intensive workloads on the clock-gated processor. The reason for this additional energy saving over coarse-grained DVFS is that coarse-grained DVFS cannot exploit the slack of individual off-chip memory accesses.

6. RELATED WORK

A large body of work has been done on DVFS. In this section, we only describe the work most related to our work.

6.1 DVFS limit studies

A number of DVFS limit studies have been done prior to this work. Some of these studies focus on compile-time DVFS, see for example Xie et al. [2003] and Hsu and Kremer [2003]. Compile-time DVFS is limited by the fact that mode-set instructions need to be inserted statically which affect all the dynamic executions of the static code fragments on which the mode-setting applies. Run-time DVFS is thus likely to achieve better energy savings than compile-time DVFS because the V/f settings can be adjusted at run time for specific dynamic executions.

Ishihara and Yasuura [1998] formulate the DVFS problem as an integer linear programming (ILP) problem and compute the upper bound on DVFS assuming no cost for V/f switching. They prove that two V/f settings and a single switching point are sufficient to minimize energy consumption; this however assumes no time-varying program behavior. Qu [2001] assumes continuous voltage scaling in his limit study. Xie et al. [2005a] propose an exact as well as a linear-time heuristic algorithm for computing the upper bound energy savings of run-time DVFS. Kim et al. [2008] explore the energy saving opportunities of fine-grained DVFS assuming no switching overheads. None of these prior works however studied the combined impact of the DVFS timescale granularity and on-chip voltage regulator scaling speed.

6.2 Fine-grained DVFS

Marculescu [2000] studies fine-grained DVFS under idealistic circumstances assuming no DVFS overhead and perfect regulator conversion efficiency. Our work demonstrates the feasibility of fine-grained DVFS under realistic circumstances while taking into account DVFS overhead and imperfect regulator conversion efficiency. Moreover, we consider both a limit study as well as a realistic implementation.

In Section 5.2, we showed that our approach has a smaller performance penalty than VSV, previously proposed by Li et al. [2003]. Apart from that, our work is different from Li et al.'s work in four other ways. First, Li et al. assume a dual-power-supply network for driving fine-grained DVFS. Our work shows that a different technology, namely on-chip regulators, can be used for fine-grained DVFS as well. As mentioned earlier, without a detailed analysis as done in this paper, it is unclear whether the energy overhead of on-chip regulators would be offset by the processor energy savings. Second, Li et al. do not study how fine-grained DVFS should cooperate with coarse-grained DVFS, which is important for soft real-time applications. We provide the solution that fine-grained DVFS should be done at the hardware level while coarse-grained DVFS is being dealt with at the system level (e.g., by system software). Third, Li et al. use heuristics to drive fine-grained DVFS. Our approach instead uses fundamental insights from mechanistic performance modeling to determine when to scale down (i.e., when an I-cache miss occurs and when dispatch stalls upon a long-latency load miss), and explores the performance versus energy trade-off for upscaling (i.e., reactive versus proactive scaling). Fourth, Li et al. assume a non-fully clock-gated processor in a 180nm technology assuming no leakage power consumption. Our work assumes a fully clock-gated processor in a 70nm technology including leakage power consumption, and we show that even under these conditions fine-grained DVFS is beneficial.

6.3 Coarse-grained DVFS

A large number of DVFS proposals have been made at a coarse-grained timescale, across multiple layers in the execution stack, from the operating system [Isci et al. 2006], to the managed runtime system [Wu et al. 2005], to the compiler [Xie et al. 2003; Hsu and Kremer 2003], and to the hardware [Hughes et al. 2001; Isci et al. 2006; Semeraro et al. 2002]. For example, Hughes et al. [2001] use DVFS to reduce energy consumption of multimedia workloads running on general-purpose processors, i.e., they scale clock frequency to reduce energy consumption while meeting the soft real-time deadlines. As another example, Isci et al. [2006] employ per-core DVFS for managing chip-level power consumption, i.e., they adjust per-core clock frequency and supply voltage to maximize system throughput while not exceeding the chip-level power budget. Commercial multi-core processors also feature DVFS mechanisms for maximizing system throughput while staying within a given power budget, see for example the AMD Opteron Quad-Core processor [Dorsey et al. 2007] with its core-level frequency scaling technology, and the Intel Itanium Montecito processor [McGowen et al. 2006] with its Foxtan hardware circuitry which scales up voltage and frequency when power consumption is below the budget and scales down voltage and frequency when above the power budget. Kim et al. [2008] explore core-level DVFS in multi-core processors using on-chip regulators. They demonstrate substantial energy saving potentials (greater than 20%) using multiple on-chip regulators compared to conventional, single power domain, off-chip regulators.

6.4 On-chip voltage regulators

Kim et al. [2008] provide a limit study considering on-chip voltage regulators in order to demonstrate the potential of fine-grained DVFS. They conclude that on-

chip regulators can offer considerable energy savings, however, regulator loss and DVFS overhead can offset the potential gains. The limit study presented here in this paper is more elaborate in the sense that it includes the impact of regulator loss and DVFS overhead, it optimizes over the course of the entire program execution (in contrast to the per 2M cycles window optimization by Kim et al.), it evaluates smaller timescales (10ns compared to 100ns), and it compares the impact of different voltage switching speeds to find the optimum speed. In addition, Kim et al. do not propose a practical mechanism to implement fine-grained DVFS, but instead they assume a policy that switches frequency at fixed interval times, driven by an offline analysis. The mechanism presented here is able to realize the potential benefits of fine-grained DVFS in practice using hardware control, and it can be combined with coarse-grained OS-level DVFS mechanisms to achieve even higher energy savings.

7. CONCLUSION

Through a limit study, we unify two apparently contradictory results in the DVFS literature, and we find that the effectiveness of coarse-grained DVFS is largely unaffected by timescale and scaling speed, however, fine-grained DVFS at small timescales may lead to substantial energy savings for memory-intensive workloads. We find that fine-grained DVFS is effective at a timescale of on the order of tens or hundreds of processor cycles, or on the order of the memory access latency. In other words, energy consumption can be reduced substantially through fine-grained DVFS if the V/f setting can be scaled down upon individual off-chip memory accesses. This of course requires that the voltage regulators be fast and efficient, which we found to be the case for contemporary on-chip voltage regulators. The potential energy savings for memory-intensive workloads are substantial: 14% on average across a range of performance targets for memory-bound applications on an aggressively clock-gated processor.

Motivated by this insight, we subsequently propose a fine-grained DVFS mechanism driven by the microarchitecture that scales the V/f setting upon individual off-chip memory accesses. Using a mechanistic analytical processor performance model we determine when to scale down the V/f setting: for off-chip I-cache misses, we scale down when the miss occurs, and for off-chip load misses, we scale down when dispatch stalls upon an off-chip load miss at the head of the reorder buffer. This fine-grained DVFS mechanism potentially achieves an on average 12% (and up to 23%) energy reduction for memory-intensive workloads at a small performance degradation (0.08% on average and at most 0.14%); this assumes proactive upscaling, i.e., the V/f setting is scaled up to its upper level by the time the off-chip memory access returns. Reactive upscaling, i.e., scaling up when the miss returns, leads to an on average 23% (and up to 40%) energy reduction for an on average 6% performance degradation for memory-intensive workloads. The fine-grained DVFS mechanism assumes on-chip regulators and incurs very little hardware overhead. We also demonstrate that the proposed fine-grained DVFS mechanism is orthogonal to coarse-grained DVFS policies leading to a multi-timescale DVFS system: the coarse-grained DVFS policy optimizes system-level objectives such as meeting soft real-time deadlines while minimizing energy consumption, and the fine-grained DVFS mechanism further reduces energy consumption by 6% on average and up

to 11% for memory-intensive workloads for an aggressively clock-gated processor (with minimal impact on performance of at most 0.4%).

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and suggestions. Stijn Eyerman is supported through a postdoctoral fellowship by the Research Foundation–Flanders (FWO). Additional support is provided by the FWO projects G.0232.06, G.0255.08, and G.0179.10, and the UGent-BOF projects 01J14407 and 01Z04109.

REFERENCES

- ABEDINPOUR, S., BAKKALOGLU, B., AND KIAEI, S. 2007. A multi-stage interleaved synchronous buck converter with integrated output filter in a 0.18 μ m SiGe process. *IEEE Transactions on Power Electronics* 22, 6 (Nov.), 2164–2175.
- BROOKS, D., TIWARI, V., AND MARTONOSI, M. 2000. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th Annual International Symposium on Computer Architecture (ISCA)*. 83–94.
- BURD, T. AND BRODERSEN, R. 2000. Design issues for dynamic voltage scaling. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*. 9–14.
- CHOU, Y., FAHS, B., AND ABRAHAM, S. 2004. Microarchitecture optimizations for exploiting memory-level parallelism. In *Proceedings of the 31st Annual International Symposium on Computer Architecture (ISCA)*. 76–87.
- CLARK, L. T., HOFFMAN, E. J., MILLER, J., BIYANI, M., LIAO, Y., STRAZDUS, S., MORROW, M., VELARDE, K. E., AND YARCH, M. A. 2001. An embedded 32-b microprocessor core for low-power and high-performance applications. *IEEE Journal of Solid-State Circuits* 36, 11 (Nov.), 1599–1607.
- DORSEY, J., SEARLES, S., CIRAULA, M., JOHNSON, S., BUJANOS, N., WU, D., BRAGANZA, M., MEYERS, S., FANG, E., AND KUMAR, R. 2007. An integrated quad-core Opteron processor. In *Proceedings of the International Solid State Circuits Conference (ISSCC)*. 102–103.
- EYERMAN, S., EECKHOUT, L., KARKHANIS, T., AND SMITH, J. E. 2009. A mechanistic performance model for superscalar out-of-order processors. *ACM Transactions on Computer Systems (TOCS)* 27, 2 (May).
- GLEW, A. 1998. MLP yes! ILP no! In *ASPLOS Wild and Crazy Idea Session*.
- HAZUCHA, P., SCHROM, G., JAEHONG, K., BLOECHEL, B., HACK, P., DERMER, G., NARENDRA, S., GARDNER, D., KARNIK, T., DE, V., AND BORKAR, S. 2005. A 233-MHz 80%-87% efficiency four-phase DC-DC converter utilizing air-core inductors on package. *IEEE Journal of Solid-State Circuits* 40, 4 (Apr.), 838–845.
- HSU, C.-H. AND KREMER, U. 2003. The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction. In *Proceedings of the International Symposium on Programming Language Design and Implementation (PLDI)*. 38–48.
- HUGHES, C. J., SRINIVASAN, J., AND ADVE, S. V. 2001. Saving energy with architectural and frequency adaptations for multimedia applications. In *Proceedings of the 34th Annual International Symposium on Microarchitecture (MICRO)*. 250–261.
- Intel 2004. *Intel XScale Core Developer's Manual*. Intel. 273473-002.
- ISCI, C., BUYUKTOSUNOGLU, A., CHER, C.-Y., BOSE, P., AND MARTONOSI, M. 2006. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*. 347–358.
- ISCI, C., CONTRERAS, G., AND MARTONOSI, M. 2006. Live, runtime phase monitoring and prediction on real systems and application to dynamic power management. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*. 359–370.

- ISHIHARA, T. AND YASUURA, H. 1998. Voltage scheduling problem for dynamically variable voltage processors. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*. 197–202.
- KARKHANIS, T. AND SMITH, J. E. 2004. A first-order superscalar processor model. In *Proceedings of the 31st Annual International Symposium on Computer Architecture (ISCA)*. 338–349.
- KAXIRAS, S. AND MARTONOSI, M. 2008. *Computer Architecture Techniques for Power-Efficiency*. Morgan & Claypool Publishers.
- KIM, W., GUPTA, M. S., WEI, G.-Y., AND BROOKS, D. 2008. System level analysis of fast, per-core DVFS using on-chip switching regulators. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*. 123–134.
- LI, H., CHER, C.-Y., VIJAYKUMAR, T. N., AND ROY, K. 2003. VSV: L2-miss-driven variable supply-voltage scaling for low power. In *Proceedings of the Annual International Symposium on Microarchitecture (MICRO)*. 19–28.
- MARCULESCU, D. 2000. On the use of microarchitecture-driven dynamic voltage scaling. In *Proceedings of the Workshop on Complexity-Effective Design (WCED), held in conjunction with ISCA*.
- MARTIN, S., FLAUTNER, K., BLAAUW, D., AND MUDGE, T. 2002. Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 721–725.
- MCGOWEN, R., POIRIER, C. A., BOSTAK, C., IGNOWSKI, J., MILLICAN, M., PARKS, W. H., AND NAFFZIGER, S. 2006. Power and temperature control on a 90-nm Itanium family processor. *IEEE Journal of Solid-State Circuits* 41, 1 (Jan.), 229–237.
- MUTLU, O. AND MOSCIBRODA, T. 2007. Stall-time fair memory access scheduling for chip multiprocessors. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 146–160.
- QU, G. 2001. What is the limit of energy saving by dynamic voltage scaling? In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 560–563.
- SCHROM, G., HAZUCHA, P., PAILLET, F., RENNIE, D. J., MOON, S. T., GARDNER, D. S., KARNIK, T., SUN, P., NGUYEN, T. T., HILL, M. J., RADHAKRISHNAN, K., AND MEMIOGLU, T. 2007. A 100MHz eight-phase buck converter delivering 12A in 25mm² using air-core inductors. In *IEEE Applied Power Electronics Conference*. 727–730.
- SEMERARO, G., ALBONESI, D. H., DROPSHO, S. G., MAGKLIS, G., DWARKADAS, S., AND SCOTT, M. L. 2002. Dynamic frequency and voltage control for a multiple clock domain microarchitecture. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*. 356–367.
- SHERWOOD, T., PERELMAN, E., HAMERLY, G., AND CALDER, B. 2002. Automatically characterizing large scale program behavior. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. 45–57.
- SHROM, G., HAZUCHA, P., HAHN, J., GARDNER, D., BLOECHEL, B., DERMER, G., NARENDRA, S., KARNIK, T., AND DE, V. 2004. A 480-MHz, multi-phase interleaved buck DC-DC converter with hysteretic control. In *IEEE Power Electronics Specialist Conference*. Vol. 6. 4702–4707.
- SKADRON, K., STAN, M. R., HUANG, W., VELUSAMY, S., SANKARANARAYANAN, K., AND TARJAN, D. 2003. Temperature-aware microarchitecture. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*. 2–13.
- Transmeta Corporation 2001. *LongRun Power Management: Dynamic Power Management for Crusoe Processors*. Transmeta Corporation.
- WIBBEN, J. AND HARJANI, R. 2007. A high efficiency DC-DC converter using 2nH on-chip inductors. In *IEEE Symposium on VLSI Circuits*. 22–23.
- WU, Q., REDDI, V. J., WU, Y., LEE, J., CONNORS, D., BROOKS, D., MARTONOSI, M., AND CLARK, D. W. 2005. A dynamic compilation framework for controlling microprocessor energy and performance. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*. 271–282.
- XIE, F., MARTONOSI, M., AND MALIK, S. 2003. Compile-time dynamic voltage scaling settings: Opportunities and limits. In *Proceedings of the International Symposium on Programming Language Design and Implementation (PLDI)*. 49–62.

- XIE, F., MARTONOSI, M., AND MALIK, S. 2005a. Bounds on power savings using runtime dynamic voltage scaling: An exact algorithm and a linear-time heuristic approximation. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*. 287–292.
- XIE, F., MARTONOSI, M., AND MALIK, S. 2005b. Efficient behavior-driven runtime dynamic voltage scaling policies. In *Proceedings of the International Conference on Hardware Software Codesign (CODES)*. 105–110.