

6-1-2012

Architecturally Homogeneous Power-Performance Heterogeneous Multicore Systems

Koushik Chakraborty
Utah State University

S. Roy

Recommended Citation

Chakraborty, Koushik and Roy, S., "Architecturally Homogeneous Power-Performance Heterogeneous Multicore Systems" (2012).
ECE Faculty Publications. Paper 62.
http://digitalcommons.usu.edu/ece_facpub/62

This Article is brought to you for free and open access by the Electrical and Computer Engineering, Department of at DigitalCommons@USU. It has been accepted for inclusion in ECE Faculty Publications by an authorized administrator of DigitalCommons@USU. For more information, please contact becky.thoms@usu.edu.



Architecturally Homogeneous Power-Performance Heterogeneous Multicore Systems

Koushik Chakraborty and Sanghamitra Roy, *Member, IEEE*

Abstract—Dynamic voltage and frequency scaling (DVFS), a widely adopted technique to ensure safe thermal characteristics while delivering superior energy efficiency, is rapidly becoming inefficient with technology scaling due to two critical factors: 1) inability to scale the supply voltage due to reliability concerns and 2) dynamic adaptations through DVFS cannot alter underlying power hungry circuit characteristics, designed for the nominal frequency. In this paper, we show that DVFS scaled circuits substantially lag in energy efficiency, by 22%–86%, compared to ground up designs for target frequency levels. We propose *architecturally homogeneous power-performance heterogeneous* multicore systems, a fundamentally alternate means to design energy efficient multicore systems. Using a system level computer-aided design (CAD) approach, we seamlessly integrate architecturally identical cores, designed for different voltage-frequency domains. We use a combination of standard cell library based CAD flow and full system architectural simulation to demonstrate 11%–22% improvement in energy efficiency using our design paradigm.

Index Terms—Dynamic voltage frequency scaling (DVFS), energy efficiency, multicore systems.

I. INTRODUCTION

GLOBAL concerns about green environment, coupled with the rapid evolution of commodity multicore systems, present unique challenges to improve the energy efficiency—an estimate of the performance delivered per watt of power drawn by the system. Multicore chip designs throughout the last decade have witnessed the trend of increasing number of on-chip cores. For example, IBM POWER7 and Sun ROCK processors have eight and 16 on-chip cores, respectively [1], [2]. Utilizing several on-chip cores, these systems can run many applications simultaneously, creating a wide diversity in power-performance requirements. In such an environment, meeting the steep energy efficiency demands requires a flexible multicore platform with the ability to exercise fine grain control [3]. On the other hand, interaction of multiple on-chip cores plays havoc in maintaining safe thermal conditions. Dynamic thermal management (DTM), a broad class of techniques that aim to deliver energy efficiency within safe thermal limits, has now become ubiquitous.

Manuscript received November 2, 2011; revised March 1, 2012; accepted April 22, 2012. This work was supported in part by the National Science Foundation under Grant CNS-1117425.

The authors are with the Electrical and Computer Engineering Department, Utah State University, Logan, UT 84322 USA (e-mail: koushik.chakraborty@usu.edu; sroy@engineering.usu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVLSI.2012.2199142

Across the entire spectrum of microprocessor design, dynamic voltage and frequency scaling (DVFS) has been the basic foundation for DTM [4]–[7]. Driven by an increasingly diverse application pool, the runtime utilization rate of on-chip cores often falls substantially below the nominal frequency. DVFS is beneficial during these phases, as it saves energy consumption with minimal loss in performance.

The effectiveness of DVFS, however, is rapidly degrading with technology scaling. This growing DVFS inefficiency stems from two critical factors: 1) dynamic adaptations fail to alter the intrinsic circuit characteristics (such as gate sizes and threshold voltages), which are designed for the nominal frequency, and tend to be power hungry and 2) forthcoming technology generations are restricting the supply voltage scaling margins [8], [9], which is the key component behind power savings through DVFS. Hence, it is now critical to find alternative scalable means of achieving energy efficiency.

In this paper, we demonstrate that dynamic adaptations using DVFS are markedly energy inefficient than techniques that design circuits ground up for lower performance. Fundamentally, lower performance requirements allow certain circuit components to choose appropriate gate sizes and device attributes, for lower power. This design style can yield significantly better energy efficiency than achieved by applying DVFS on circuits designed for higher performance. The key question is how to use this design style at the system level, and analyze its impact on real workload execution.

Using a system level computer-aided design (CAD) approach, we propose *architecturally homogeneous power-performance heterogeneous* multicore systems (AHPH): a novel power efficient design paradigm. In this approach, on-chip cores are topologically identical, but they are designed to be power-performance optimal for separate voltage-frequency (VF) domains. Our approach is in contrast to the conventional design style, where *all cores are designed to be power-performance optimal at the nominal frequency* [1], [2], [10]. Our design paradigm yields heterogeneous circuit characteristics in various on-chip cores, without altering their fundamental micro-architecture and circuit topology. To the best of our knowledge, this design paradigm is the first of its kind, capturing the low-cost benefit of a homogeneous core architecture, and delivering energy-efficiency advantages of ground up circuit designs for target latencies in a multicore system.

Our paper makes several contributions in the broad area of circuit-architecture co-design. First, we demonstrate how circuits designed ground up at specific latencies are more energy efficient than using DVFS on circuits designed at

the nominal latency, *by up to 7 \times* (Section III). Second, we propose AHPH, a novel power efficient multicore design paradigm that combines ground up circuit design style with architectural-level modular design (Section IV). Third, we present a simulated annealing algorithm for selecting appropriate VF domains for a target multicore (Section V), and discuss AHPH implementation (Section VI). Fourth, using a rigorous experimental setup combining standard cell-based CAD flow with architectural full system simulation, we analyze our design and several state-of-the-art DTM techniques (Section VIII). Overall, we observe 11%–22% improvement in system-level energy efficiency in representative multicore workloads.

II. RELATED WORK

With the advent of multicore systems and growing power density induced thermal constraint, DTM has received significant attention in the research community. To the best of our knowledge, the underlying multicore in all of these works uses the conventional practice of on-chip cores designed for a fixed nominal frequency. Our work in this paper is fundamentally different, as we propose a novel systemlevel CAD approach to design multicore systems, where individual cores are designed *ground up to operate at different frequency levels*.

Donald *et al.* classified existing DTM techniques into two types: distributed (per-core) or global [11]. Per core DVFS offers higher flexibility, and several works have shown its power-performance advantages [3], [6], [12]–[14], as well as growing advantage with the number of on-chip cores [15]. Although its implementation is substantially complex, most recent multicore systems are already offering this capability [10]. With multicores integrating a growing number of cores [1], [2], we take this forward looking approach in this paper to assume per core VF domains. Local architecture adaptation has been combined with global DVFS to yield some of the advantages of per-core DVFS [16].

III. MOTIVATION

Our work is inspired by a combination of multiple emerging technology trends. Modern multicore processors have several on-chip cores, which are designed for optimal power-performance at the highest utilization rate (nominal frequency). A diverse pool of applications running on these cores dictate wide variations in power-performance requirements. To cope with such variations, DVFS is employed to improve energy efficiency during low utilization, and prevents thermal violations during sustained high utilization. In this section, we present a quantitative analysis on expected DVFS efficiency, in the light of increasing obstacles of voltage scaling: the key driver behind DVFS [8], [17].

A. Background on Area-Speed-Power Tradeoffs

Techniques, such as gate sizing and multiple threshold voltage assignment are widely used for optimizing power and performance in integrated circuits [18]–[22]. A particular class of these techniques generally exploits any available slack in the circuit for reducing power. For example, low threshold

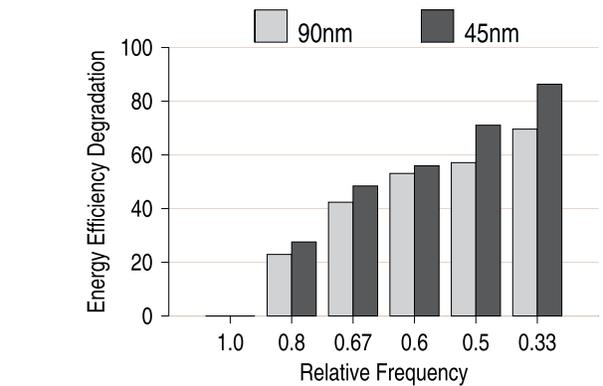


Fig. 1. c7552: energy efficiency degradation in DVFS (lower is better).

voltage devices (LVT) are very high speed, but also consume extremely high leakage power. Increasing the threshold voltage levels (RVT, HVT) causes exponential reduction in leakage power, while also slowing down the speed of the logic gates. Likewise, as the gate size is increased, the speed of a gate improves at the cost of additional power and area. Hence, the logic paths in a circuit having positive slack, can replace some of their gates with high threshold voltage gates as well as smaller sized gates to save power without any performance hit. Efficient gate sizing and threshold voltage assignment can exploit the available slack in reducing overall power and energy consumption of a logic circuit.

B. Dynamic Adaptation Versus Ground Up Optimal Design

The key motivation of our work is to exploit the increasing number of on-chip cores in high performance microprocessors in a power-performance optimal way. For example, a core may be designed for operation at 3 GHz, but during a particular program phase, the processor may perform DVFS to operate it at 2 GHz. However, the intrinsic circuit characteristics (e.g., distribution of gate sizes and threshold voltages) of the core are substantially different from that of a core *designed ground up* to be power-performance optimal at 2 GHz. The key issue to understand is their comparative power-performance, when the former is dynamically adapted to operate at 2 GHz.

We investigate this circuit design tradeoff using a combinational 32-bit ALU from the ISCAS benchmark suite (c7552) as well as a sequential floating point unit (FPU) from the OpenSPARC T1 processor [23]. We use two standard cell libraries for this paper: a UMC 90-nm industrial library [24] and a TSMC 45-nm technology library. Both libraries are characterized for multiple threshold voltages: low V_t (LVT), regular V_t (RVT), and high V_t (HVT). We allow 20% voltage scaling to model the voltage scaling constraints of current and forthcoming technology generations.

1) *Technology Trend DVFS Inefficiency*: Figs. 1 and 2 present a quantitative comparison of how much DVFS techniques lag their corresponding ground up optimal designs in terms of energy efficiency. We report results for six operating frequencies (1 = nominal). The ALU and FPU designed at the nominal frequency are DVFS scaled to operate at the lower operating frequencies. At each operating point, we report the percentage degradation in energy efficiency of the *DVFS*

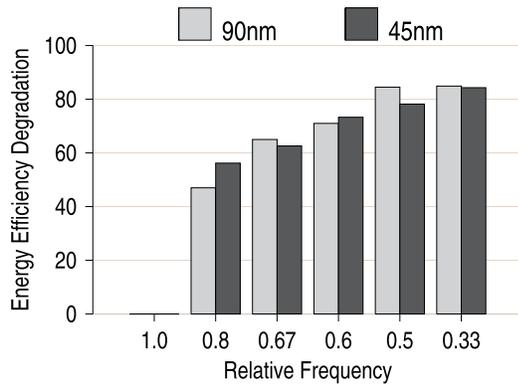


Fig. 2. OpenSPARC floating point unit: energy efficiency degradation in DVFS (lower is better).

scaled nominal design with respect to the ground up optimal design at that operating frequency. Despite employing allowable DVFS, the nominal ALU suffers 57% (90 nm) and 71% (45 nm) energy efficiency degradation at 50% of the nominal frequency (0.5), compared to the ALU designed to operate at that frequency. Similar energy efficiency degradation is also observed in the nominal FPU. Interestingly, this degradation in energy efficiency grows rapidly as a circuit diverges further away from its design frequency. The nominal ALU suffers 69% and 86% loss in energy efficiency, respectively, when it is DVFS scaled to operate at the relative frequency of 0.33. At the same relative frequency, the nominal FPU suffers 92% and 86% loss in energy efficiency at 90 nm and 45 nm, respectively. As leakage becomes more dominant at lower technology nodes, the degradation also becomes more pronounced as ground up designs can replace leaky devices with low leakage devices (Tables I and II).

2) *Methodology*: We use Synopsys Design Compiler for synthesizing our design with circuit optimizations. We use multiple threshold voltages (LVT, RVT, and HVT) to optimize power at the nominal frequency. Subsequently, we calculate several operating frequencies with delays progressively higher than the nominal. At each operating frequency, we perform:

- 1) ground up optimal design. For a given latency, we synthesize the circuit with leakage and dynamic power optimization at multiple voltages (between 100% and 80% of nominal voltage) and pick the design with minimum power;
- 2) voltage frequency scaling of nominal design. We apply DVFS on the ALU and FPU designed at the nominal frequency, to measure the scaled power components when they are operated at a frequency lower than nominal.

Tables I and II present detailed internal circuit characteristics of ground up optimal designs at the six operating points for 90- and 45-nm nodes, which lead to the energy efficiency trend seen in Figs. 1 and 2. We report: 1) total area; 2) percentage area distribution of LVT, RVT, and HVT gates in the circuit; and 3) operating voltage. It is seen that the internal circuit characteristics of the same functional unit vary substantially when optimized at different performance points. When designed for the highest performance, the circuit takes

TABLE I
C7552 GROUND UP DESIGNS

Rel. freq.	1.0	0.8	0.67	0.6	0.5	0.33
Tech. node	90 nm					
Area	11879	8955	8808	7402	5603	4463
HVT	3.6%	8.6	8.3%	8.6%	20.6%	49.8%
RVT	16.2%	43.1%	64.3%	86.1%	76.26%	47.3%
LVT	80.2%	48.2%	27.4%	5.3%	3.19%	2.8%
Voltage V	1.0	0.95	0.85	0.85	0.85	0.85
Tech. node	45 nm					
Area	2120	1306	1151	1071	1012	980
HVT	13.0%	22.4%	38.9%	43.6%	56.9%	71.1%
RVT	24.0%	34.1%	33.9%	32.4%	30.5%	22.8%
LVT	63.0%	43.5%	27.2%	24.0%	12.6%	6.1%
Voltage V	0.99	0.84	0.81	0.81	0.81	0.81

TABLE II
OPENSARC FPU GROUND UP DESIGNS

Rel. freq.	1.0	0.8	0.67	0.6	0.5	0.33
Tech. node	90 nm					
Area	286140	282621	280131	275695	260376	239775
HVT	31.2%	39.2%	54.6%	53.4%	72.1%	98.8%
RVT	35.1%	27.5%	33.8%	36.8%	23.2%	1.2%
LVT	33.7%	33.3%	12.7%	9.8%	4.6%	0.1%
Voltage V	1.0	1.0	1.0	0.97	0.97	0.97
Tech. node	45 nm					
Area	61530	59598	57542	52172	50626	48224
HVT	33.0%	42.8%	56.9%	65.3%	77.2%	94.5%
RVT	16.5%	19.4%	17.7%	15.2%	13.7%	4.9%
LVT	50.5%	37.8%	25.4%	19.5%	9.1%	0.6%
Voltage V	0.99	0.99	0.81	0.81	0.81	0.81

TABLE III
C7552 GROUND UP DESIGNS WITH RVT TRANSISTORS

Rel. freq.	1.0	0.8	0.67	0.6	0.5	0.33
Tech. node	90 nm					
Area	11237	8507	8338	6958	5256	4142
Voltage V	1.0	0.95	0.90	0.85	0.85	0.85
Tech. node	45 nm					
Area	2056	1269	1126	1039	980	955
Voltage V	0.99	0.84	0.84	0.81	0.81	0.81

up the largest area as different gates in the circuit are sized up. As expected, we find the largest concentration of LVT devices at the highest performance level (1.0). At lower performance levels, the power consumption drops significantly due to: 1) smaller area (lower gate capacitance); 2) lower leakage due to the reduction of LVT devices; and 3) lower supply voltage.

C. Dynamic Adaptation Versus Ground Up Design Using Single Threshold Transistors

A large portion of the energy efficiency benefits of ground up optimal designs over their DVFS scaled counterparts comes from reduced leakage power. To analyze the energy benefits of this scheme outside the leakage benefits, we repeat our analysis with single threshold voltage transistors (RVT only). Tables III and IV show the circuit characteristics using only RVT transistors.

Figs. 3 and 4 show the energy efficiency degradation of DVFS designs compared to their ground up counterparts. We can see from the results that even using single threshold transistors, ground up designs show better energy

TABLE IV

OPENSPARC FPU GROUND UP DESIGNS WITH RVT TRANSISTORS

Rel. freq.	1.0	0.8	0.67	0.6	0.5	0.33
Tech. node	90 nm					
Area	271489	266280	254523	245202	233252	221009
Voltage V	1.0	1.0	0.94	0.85	0.82	0.82
Tech. node	45 nm					
Area	60182	58330	54789	53723	52580	51629
Voltage V	0.99	0.99	0.81	0.81	0.81	0.81

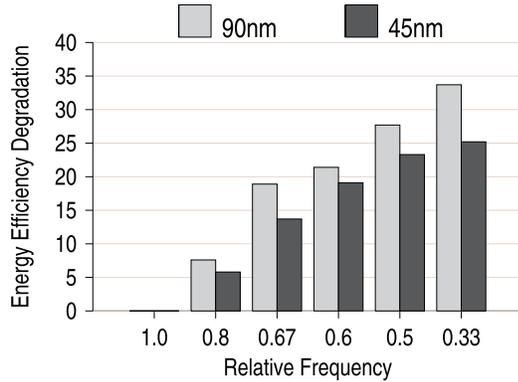


Fig. 3. c7552: energy efficiency degradation in DVFS with RVT transistors (lower is better).

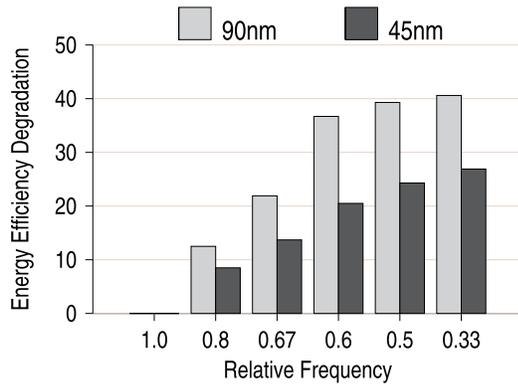


Fig. 4. OpenSPARC floating point unit: energy efficiency degradation in DVFS with RVT transistors (lower is better).

efficiency compared to DVFS designs. This benefit is however, much smaller than with multiple threshold transistors. The energy efficiency under the RVT only designs is drawn from: 1) smaller area (lower gate capacitance) and 2) lower supply voltage. However, no benefit is obtained from lower leakage in this case.

D. Performance Impact of Lower Frequency

At the system level, the performance impact of lower operating frequency is diverse: *not all applications are equally affected by the core frequency*. Fig. 5 shows this diversity in SPEC CPU2006 workloads measured using the methodology in Section VIII. We categorize the diversity in three different classes that collectively capture the range of IPCs (instructions per cycle) seen in these workloads (see Table VI). *High IPC*

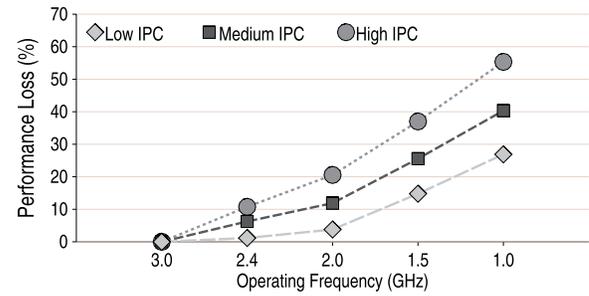


Fig. 5. Performance loss from core frequency scaling.

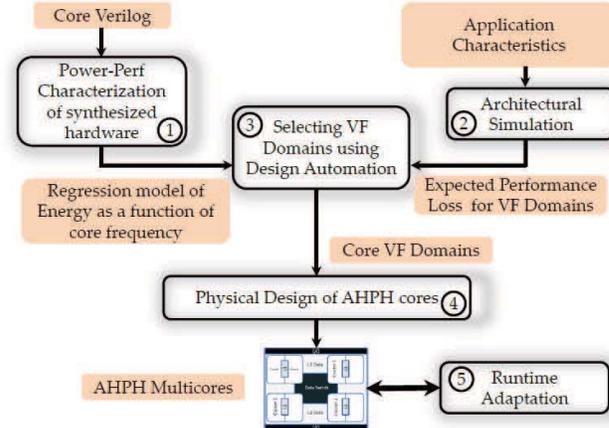


Fig. 6. Overview of the AHPH multicore system design.

benchmarks include *h264*, *hmmr* and *sjeng*. Mid IPC benchmarks include *astar-lakes*, *gcc*, *gobmk*, *bzip2*, and *perlbench*. Low IPC benchmarks include *mcf*, *xalancbmk*, *libquantum*, and *omnetpp*. We observe that low IPC workloads suffer substantially lower performance loss at lower frequencies compared to high IPC workloads.

E. Summary

We demonstrated that ground up optimal designs at any performance level are substantially more energy efficient than their DVFS scaled counterparts. Given the diverse latency requirements of on-chip cores in modern multicores, we explore design automation techniques to apply this circuit design style for an entire system design next.

IV. AHPH MULTICORE: OVERVIEW

In this paper, we propose a cost-effective heterogeneous multicore design paradigm as an energy efficient alternative to DVFS-based multicore systems. This paradigm creates several *architecturally identical* cores, designed ground up to be power-performance optimal at specific VF domains.

Fig. 6 shows an overview of designing AHPH cores in a multicore system. The AHPH multicore design is comprised of five phases, marked as 1–5 in Fig. 6. In Phase 1, we use the Synopsys Design Compiler tool to synthesize the Verilog description of a 64-bit out-of-order microprocessor core at various target frequencies (using a methodology similar

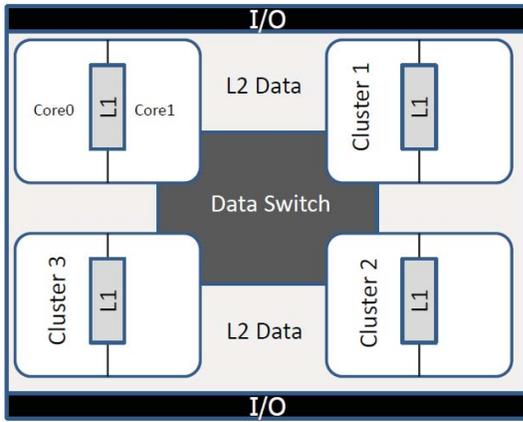


Fig. 7. Clustered multicore system.

to Section III). Subsequently, we do a power-performance analysis of the synthesized hardware to develop regression models of energy as a function of the core frequency.

In Phase 2, we determine the expected performance loss at specific VF domains using full-system architectural simulation (collecting data similar to Fig. 5). To handle the large workload space of applications, we use a methodology employing design automation algorithm based on yield detailed in Section V.

In Phase 3, we use a simulated annealing optimization algorithm to select the VF domain levels of each core for maximizing the energy efficiency of the multicore system (Section V). The algorithm for VF selection depends on the number of cores, the workload characteristics as well as the technology node. Once we select the VF domains of the target multicore, Phase 4 develops ground-up designs of the cores at the target VF domains, using the Synopsys Design Compiler tool (Section VI).

The output from Phase 4 is the synthesized design of our AHPH multicore system, where each core is designed ground up for the target VF domain obtained from the simulated annealing algorithm. To dynamically exploit the AHPH cores and accommodate variations in workload characteristics, we use *runtime adaptation* techniques in Phase 5 (Section VI-A).

V. SELECTING VF DOMAINS

In this section, we formally describe the VF domain selection problem, analyze its complexity, and describe our simulated annealing-based stochastic algorithm.

For the ease of comprehension, we distinguish *application* and *workload* in the following way. Applications for a multicore system consist of individual programs, such as SPEC CPU2006 benchmarks. Workloads, on the other hand, contain a collection of applications to run on the target multicore. Detailed description of our multicore workloads based on virtual machine consolidation is given in Section VII-B.

A. Formalizing the VF Selection Problem

We are given a set of k permissible VF domain levels denoted by the set \mathcal{PD} , where

$$\mathcal{PD} = \{(v_1, f_1), \dots, (v_k, f_k) : f_k > f_{k-1} > \dots > f_2 > f_1\}.$$

Our goal is to select a VF domain assignment for a multicore system with n cores to maximize the system level energy efficiency, while minimizing the expected performance loss. We denote a possible VF assignment by the set \mathcal{V} , where

$$\mathcal{V} = \{(V_i, F_i) : (V_i, F_i) \in \mathcal{PD}, \quad 1 \leq i \leq n\}.$$

In order to maximize the overall energy efficiency, we use an objective function that correlates with the total energy of the VF assignment under a bound on the performance loss. We use the methodology in Section III to develop a regression model of energy $E(F_i)$ as a function of the core frequency F_i . Hence, our objective function for assignment \mathcal{V} is given by $\sum_{i=1}^n E(F_i(\mathcal{V}))$.

Workload Space: Our selection of VF domains further needs to be constrained by an estimation of the expected performance loss of the system. For a general purpose microprocessor, a large number of applications can combine in various phases creating a very large workload space. For example, characterizing the application phases into p IPC classes creates a workload space of size p^n with exponential complexity. Exhaustive evaluation of all possible workload combinations becomes an extremely hard problem.

We use a simulated annealing-based stochastic optimization to formulate our VF domain selection. We define *multicore performance yield* (Y) to estimate the performance loss in a large workload space as

$$Y(\mathcal{V}) = Pr\left(\frac{1}{n} \sum_{i=1}^n \frac{IPC_i(\mathcal{V})}{IPC_{base}} \geq L_0\right) \quad (1)$$

where, IPC_{base} is the IPC of an application running in a core with the highest VF domain, while $IPC_i(\mathcal{V})$ is the IPC of the same application running on the i th core with a VF tuple (V_i, F_i) in \mathcal{V} . $Y(\mathcal{V})$ is the probability that a VF assignment \mathcal{V} produces a relative performance higher than L_0 under diverse workload combinations. Note that Y uses a conservative estimate of IPC_{base} , as certain applications may not be able to continually execute at the highest VF level due to their thermal profile. Therefore, a thermally aware throughput of AHPH can be higher than a multicore comprised of cores designed for the nominal frequency (details in Section VIII). Throughput here refers to the number of instructions executed by the multicore system per unit time.

We use Monte Carlo simulations with 1000 randomly selected samples of applications to estimate $Y(\mathcal{V})$ using (1) and characterized performance loss data similar to Fig. 5. Our VF selection problem can be formulated as

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^n E(F_i(\mathcal{V})) \\ & \text{s.t.} \quad Y(\mathcal{V}) \geq Y_0. \end{aligned} \quad (2)$$

B. Simulated Annealing Optimization

Algorithm 1 shows our simulated annealing optimization. The algorithm follows the traditional simulated annealing schedule where anneal temperature T is initialized to a high value T_0 and cools down to ϵ .

Algorithm 1 VF_select

```

1: Initialize:  $\mathcal{V} \leftarrow \mathcal{V}_{in}; T \leftarrow T_0$ 
2: Calculate  $O \leftarrow \sum_{i=1}^n E(F_i(\mathcal{V}))$ 
3: while  $T > \epsilon$  do
4:   while moves  $< M$  do
5:      $\mathcal{V}_{new} = \text{APPLY RANDOM MOVE}(\mathcal{V})$ 
6:      $O_{new} \leftarrow \sum_{i=1}^n E(F_i(\mathcal{V}_{new}))$ 
7:     Calculate  $Y(\mathcal{V}_{new})$ 
8:     if ANNEAL CRITERIA MET ( $O, O_{new}$ ) then
9:        $\mathcal{V} \leftarrow \mathcal{V}_{new}; O \leftarrow O_{new}$ 
10:    end if
11:  end while
12:   $T \leftarrow \lambda T$ 
13: end while

```

Initial Solution: A trivial initial solution is $\mathcal{V}_{in} = \{(V_i, F_i) : V_i = v_k, F_i = f_k, 1 \leq i \leq n\}$. This solution obeys the yield constraint in (2) as it assigns the highest VF domain to each core. Since, more than one core can have the same VF domain assigned, the size of the solution space is given by $\binom{k+n-1}{n}$.

Annealing Moves: We perturb our current solution \mathcal{V} using two types of moves, which combine to explore the entire solution space. Move M_0 randomly selects a VF domain in \mathcal{V} and raises it to the next higher VF level in \mathcal{PD} . Move M_1 randomly selects a VF domain in \mathcal{V} and lowers it to the immediate lower VF level in \mathcal{PD} . APPLY RANDOM MOVE(\mathcal{V}) randomly selects and applies a move to \mathcal{V} .

ANNEAL CRITERIA MET(O, O_{new}) evaluates to true if: 1) O_{new} is smaller than O and 2) O_{new} is larger than O with a probability of accepting uphill moves that decreases with the progress of annealing.

1) *Application Diversity:* To derive the best multicore VF domain configuration using our proposed algorithm, we randomly selected 1000 multicore applications. Each of these applications is essentially comprised of eight randomly selected SPEC CPU2006 benchmarks, we analyze in this paper (Table VI shows the SPEC CPU2006 benchmarks). However, none of these 1000 randomly selected application is comprised of the exact set of benchmarks combined in the consolidated virtual machines (Table VII), we have used in our experimental results (Section VIII).

VI. IMPLEMENTATION

After selecting the target VF domains in a multicore, we design individual cores for their target frequencies, in a ground up manner using the methodology in Section III-B2. A key challenge beyond that is to adapt to the runtime variation in workload characteristics, described next.

A. Runtime Adaptation

The runtime adaptation scheme is loosely based on the hardware task migration proposed in [12]. The basic objective of the technique is to reassign workloads on the available AHPH cores to better fit their runtime execution characteristics, while also alleviating thermal emergencies.

TABLE V
MULTICORE VF DOMAIN CONFIGURATION

Cluster	Core configuration
Cluster 0	Core 0: 0.99 V, 3.0 GHz; Core 1: 0.84 V, 2.4 GHz
Cluster 1	Core 0: 0.99 V, 3.0 GHz; Core 1: 0.81 V, 2.0 GHz
Cluster 2	Core 0: 0.84 V, 2.4 GHz; Core 1: 0.81 V, 2.0 GHz
Cluster 3	Core 0: 0.81 V 1.8 GHz; Core 1: 0.81 V, 1.5 GHz

At each scheduling epoch, we reassign the workloads on different clusters or different cores within the cluster. During task migration, the hardware saves and restores the register state using the on-chip caches [12]. Using the performance counters to monitor IPC, the hardware selects the best possible VF domain for an application. Within the same cluster, the migration overhead is small, and thus such reassignments are always favored over inter-cluster reassignments. An inter-cluster reassignment is only accomplished when the VF domains differ by at least two discrete frequency levels.

Overhead: The hardware task migration avoids context switch to the OS, but incurs overhead due to: 1) saving and restoring register state; 2) effect of cold cache; and 3) loss of branch predictor state. The cold cache effect is the major component, but it is incurred only during inter-cluster migration, as first level caches are shared within a cluster (Fig. 7). In our full system architectural simulator infrastructure, we faithfully model both the latency and energy overhead of task migration in results presented in Section VIII. Across all workloads, latency overheads are less than 2%, while the power overheads are less than 1%, similar to [12].

VII. EXPERIMENTAL METHODOLOGY

We combine an elaborate standard cell-based CAD design flow with a detailed cycle accurate full system architectural simulation, to evaluate the energy efficiency of the system using real workloads.

A. Multicore System

We use full-system simulation built on top of Virtutech SIMICS [21]. SIMICS provides the functional model of several popular ISAs, in sufficient detail to boot an unmodified operating system and run commercial applications. For our experiments, we use the SPARC V9 ISA, and use our own detailed timing model to enforce timing characteristics of eight-core clustered multicore system. Fig. 7 shows the clustered multicore system we use for this paper. This model is similar to the Sun's ROCK multicore chip [30], where first level caches are shared within a cluster.

In our model, we use four clusters, each containing two on-chip cores that share both L1 Instruction and Data caches. L2 is shared between all clusters, and uses MESI coherence protocol. On-chip cores have seven stage pipelines, with dual issue 32 entry out-of-order issue window. L1 (32 KB 4-way split Instruction and Data) has a single cycle latency, while 16-way 8 MB L2 and main memory are accessed in 25 and 240 cycles, respectively. We use the 45-nm technology node, with 0.99 V VDD and 3.0-GHz nominal frequency. L1 and L2 caches use this nominal VF domain (0.99 V and 3.0 GHz).

TABLE VI
SPEC CPU2006 BENCHMARK IPC

Benchmark	Phase I IPC	Phase II IPC
astar-lakes	0.43	1.59
sjeng	1.85	1.74
gcc	0.71	1.41
hmmmer	2.54	2.54
gobmk	0.65	1.31
mcf	0.34	0.24
bzip2	1.84	0.178
xalancbmk	0.39	0.41
h264	2.17	2.31
libquantum	0.51	0.31
omnetpp	0.55	0.59
perlbench	0.79	1.08

TABLE VII
WORKLOAD COMPOSITION. SPECIFIC PHASES INCLUDED FROM EACH BENCHMARK IS SHOWN IN PARENTHESES

Workloads	Composition
VM1	astar-lakes(both), xalancbmk (2), sjeng(2), h264(both), libquantum(2), hmmmer(1)
VM2	mcf(both), hmmmer(both), h264(1), xalancbmk(1), gobmk(both)
VM3	hmmmer(2), gcc(both), omnetpp(both), libquantum(1), perlbench(2)
VM4	gobmk(both), perlbench(1), sjeng(1), h264(both), mcf(2), libquantum(2)
VM5	mcf(2), sjeng(1), astar-lakes(both), omnetpp(2), h264(1), gcc(both)
VM6	libquantum(both), xalancbmk(2), gcc(both), mcf(both), bzip2(1)
VM7	mcf(2), xalancbmk(2), h264(both), omnetpp(both), astar-lakes(2), bzip2(2)
VM8	xalancbmk(both), perlbench(both), bzip2(1), mcf(2), sjeng(1), omnetpp(1)

B. Workloads

We use several consolidated virtual machines, each running a SPEC CPU2006 benchmark on Solaris nine operating system. The representative portions of each application are extracted using the SimPoint toolset [28]. We select two most representative phases from these SPEC benchmarks. The IPC characteristics for these benchmarks are shown in Table VI. Table VII shows the composition of our consolidated virtual machine workloads. Workloads are run for 100 million instructions.

C. Power and Timing Analysis

We generate accurate power and timing information of designs tuned at multiple operating frequencies, using an extensive standard cell-based CAD flow. The energy consumption is then calculated by combining the cycle level usage information from our architectural simulation with power and timing from the standard cell library flow (detailed next).

1) *Verilog Modeling and Synthesis*: We design a 64-bit out-of-order microprocessor core containing the following critical modules in Verilog: 64-bit ALU, 64-bit register file with four banks, 64-bit Rename logic incorporating a CAM table and freelist, and 64-bit Instruction Scheduler (32 entry). These designs are developed and verified using the ModelSim simulation tool. Next, the microprocessor core is synthesized for various target frequencies using Synopsys Design Compiler

and a 45-nm TSMC standard cell library (see Section III-B2). We measure the frequency, dynamic, and leakage power of the synthesized cores at each VF operating point. Cache powers are estimated using Cacti 6.0 [22].

2) *Temperature Estimation Using HotSpot*: To estimate the workload-dependent runtime core power, we combine the dynamic power and leakage power characteristics of the synthesized hardware, and the runtime usage for processor components from the architectural simulator. Multicore power dissipation data is then fed to HotSpot 4.2 [29], which is integrated with our architectural simulator to obtain transient temperatures. Our overall model is similar to the Wattch-based techniques [4]. However, we use a more accurate power estimation using synthesized hardware, instead of linear scaling. Before the simulation run, we feed the power trace for each workload to derive steady state thermal characteristics of the multicore system. Subsequently, transient temperatures are used to model thermal emergencies triggered by on-chip thermal sensors. In our experiments, we use 90 °C as the thermal threshold. We use a sampling interval of 0.05 msec to communicate runtime power dissipation with Hotspot. Transient temperatures are also used to adjust the temperature-dependent leakage power in our experiments.

VIII. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our system level techniques to combine CAD-driven AHPH core design and runtime adaptation.

A. Selected VF Domains

Based on the workload and multicore configuration, Table V shows VF domains selected by our algorithm in Section V, where L_0 and Y_0 are set to 0.85 and 0.8, respectively.

B. Comparative Schemes

A large number of schemes are possible within the entire design spectrum of Multicore DTM. We choose the following pivotal multicore designs that capture the state-of-the-art spectrum wide characteristics.

- 1) *Homogeneous Cores with DVFS (D-DVFS)*: Per core DVFS is applied where frequency levels are set using feedback control [11]. Changing the VF domain of a core must incur transition delay costs. Since cores are designed for the nominal frequency, altering the frequency levels make them power inefficient. No task migration is performed.
- 2) *Homogeneous Cores without VF Transition Delays (ADVFS)*: This scheme models recent developments in altering the clock frequency without pausing execution, by using a digital phase locked loop (DPLL) [30]. Although voltage transition do not cause paused execution, these transitions happen only at 32ms interval with a 6.25 mV step [30]. Consequently, they offer minimal throughput loss, but energy savings are also limited in this scheme.
- 3) *Homogeneous Cores with Static VF (SVFS)*: Homogeneous cores, designed for nominal frequency, are tuned

TABLE VIII
SUMMARY OF EXPERIMENTAL MULTICORE SCHEMES

Scheme	H/W task migration	Overhead	Power efficient
D-DVFS	No	VF transition delays	No
ADVFS	No	No	No
SVFS	Yes	task migration	No
AHPH	No	VF transition delays	Yes
AHPH-TM	Yes	task migration	Yes

statically to operate at different VF domains. Frequent hardware-driven task migration is performed, and the overall design is similar in spirit to [12]. We use DVFS to avoid thermal emergencies.

- 4) *AHPH*: This is a simplified version of our proposed multicore system, comprised of topologically homogeneous power-performance heterogeneous cores. Each core has identical logic depth and pipelining. However, the choices of gate sizes and threshold voltages in these cores are different to provide the right tradeoff between speed and power. Workload assignments are static, and no task migrations are performed. DVFS is performed only to avoid thermal emergencies.
- 5) *AHPH-TM*: This is similar to *AHPH*, but we also perform task migration to adapt to transient variations in workload characteristics.

A summary of the fundamental characteristics of the above schemes is shown in Table VIII. Except in D-DVFS, VF domains are fixed in the other three designs, except to avoid thermal emergencies (90.0 °C in our setup). We assume 10- μ s transition delay for switching between VF domains [11]. We use 10 000 cycles as the scheduling epoch for our runtime adaptation, and model all aspects of our runtime task migration overhead as described in Section VI-A.

C. Throughput

Fig. 8 presents the throughput comparison between the different schemes. Overall, hardware-driven task migration in SVFS is able to improve the throughput by avoiding VF transition delays, and utilizing high-speed task migration. Using DPLL, ADVFS scheme also avoids VF transition delays, but the ability to lower clock frequency without thread migration further improves its throughput. AHPH performs comparably to the D-DVFS scheme, but without task migration it is unable to exploit the opportunities to reassign the workloads. However, more power efficient cores lead to substantially smaller thermal violations, which avoid VF transition delays. With task migration, AHPH-TM performs similar to SVFS, but outperforms in a few workloads.

D. Energy Efficiency

Fig. 9 presents the energy efficiency comparison. We measure the energy efficiency as throughput (Billion Instruction per Second)/Watt. Across all workloads, the use of power efficient cores in our schemes substantially improves the energy efficiency. AHPH shows an overall improvement of 7%–14%, which is primarily due to conservation

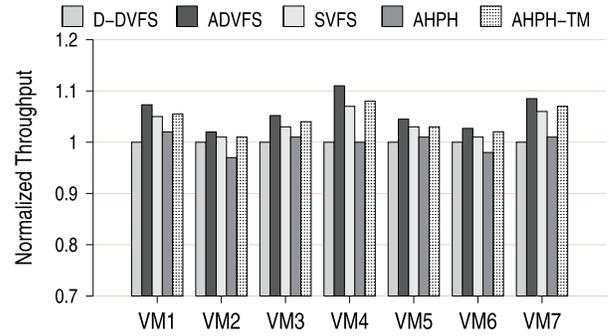


Fig. 8. Throughput comparison (higher is better).

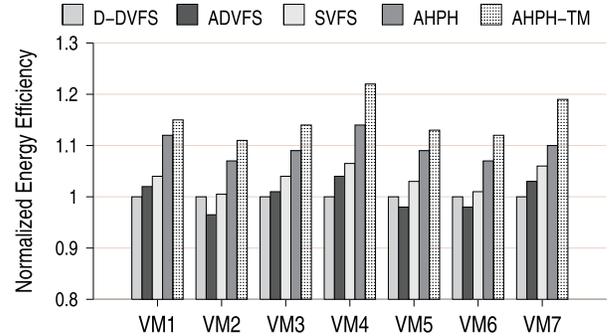


Fig. 9. Energy efficiency comparison (higher is better).

of energy consumption through a combination of high-, medium-, and low-power cores. AHPH-TM is able to boost the throughput while saving energy consumption, delivering an overall improvement of 11%–22% in energy efficiency. Compared to SVFS that uses the same VF domains in homogeneous cores, both the AHPH schemes perform substantially better due to our power efficient design paradigm. ADVFS shows a loss in energy efficiency in several benchmarks as without pausing execution, voltage scaling can happen at a very fine grain (6.25 mV per 32 ms [30]), thereby limiting the energy savings obtained during low throughput phases in these benchmarks.

E. Thermal Violations

We show a comparison of DVFS induced by thermal sensors across different schemes in Fig. 10. The figure shows percentage change in these events compared to D-DVFS. When a temperature threshold is indicated in a given core, both D-DVFS and AHPH immediately reduce their VF domain. However, SVFS and AHPH-TM attempt to tackle the problem by applying task migration. If the same sensor again indicates temperatures beyond the threshold, the VF domain is lowered from the current one, indicating a thermal-induced transition.

AHPH lacks both feedback-driven control and task migration capability, which can lead to higher thermal emergency responses. However, using power efficient cores in AHPH substantially reduce these events. Overall, we notice up to 4% reduction in thermal-induced transitions in AHPH. Using task migration reduces these transitions in SVFS and we observe up to 6% reduction. AHPH-TM is able to further reduce these

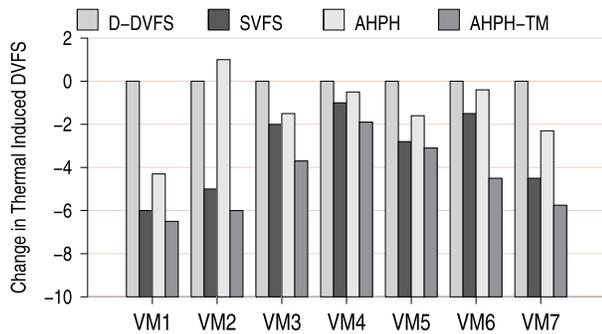


Fig. 10. Percentage change in thermal-induced DVFS (lower is better).

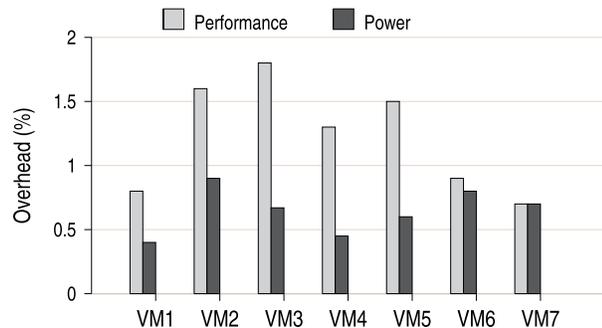


Fig. 11. Overhead from task migration in AHPH-TM.

events, as it benefits from both task migration and power efficient cores, improving its thermal characteristics.

In our ADVFS scheme, no thermally induced DVFS are triggered, and thus omitted from Fig. 10. In this scheme, when the temperature reaches the threshold, we aggressively reduce the frequency by 50%. In all cases, such drastic reduction in clock frequency is enough to reduce the temperature in the next epoch. Subsequently, the frequency is slowly ramped up to the original level to limit the performance overhead.

F. Task Migration Overhead

Fig. 11 shows the task migration overhead in the AHPH-TM scheme. The figure shows both the performance overhead as well as the power overhead. These overheads, however, only include the directly measurable overheads incurred during task migration: latency and power consumption of transferring register state from one core to another. The penalty from cold caches is not shown here, but is included in the power-performance results presented before. We notice fairly small overheads in power-performance across these benchmarks.

G. Restricted VF Domain Assignment

In our study, we have thus far allowed unrestricted VF domain assignments to processing cores in a cluster. With a shared L1, this approach may lead to additional design issues related to synchronization. While recent progress in low overhead and high data rate synchronization can eliminate bulk of the overhead [31], we also wanted to explore the efficacy

TABLE IX
RESTRICTED MULTICORE VF DOMAIN CONFIGURATION

Cluster	Core configuration
Cluster 0	Core 0: 0.99 V, 3.0 GHz; Core 1: 0.99 V, 3.0 GHz
Cluster 1	Core 0: 0.84 V, 2.4 GHz; Core 1: 0.84 V, 2.4 GHz
Cluster 2	Core 0: 0.81 V, 2.0 GHz; Core 1: 0.81 V, 2.0 GHz
Cluster 3	Core 0: 0.81 V 1.5 GHz; Core 1: 0.81 V, 1.5 GHz

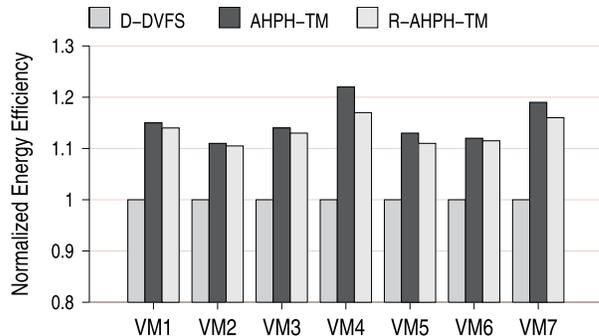


Fig. 12. Energy efficiency with restricted VF domain assignment.

of our scheme when cores within a cluster must operate in the same VF domain. Imposing this additional constraint, yields the multicore configuration shown in Table IX. All L1 caches use the same VF domain as their parent cluster. Similar to the AHPH system, L2 caches use the nominal VF domain, although it is possible for the L2s to use a lower VF domain.

Using this multicore, the energy efficiency obtained in our schemes are shown in Fig. 12, where we show the results for the baseline D-DVFS, AHPH-TM, and this restricted multicore (R-AHPH-TM). We notice that R-AHPH-TM is able to offer almost similar advantages in energy efficiency barring VM4 and VM7. These two benchmarks offer substantial opportunity in intra-cluster thread migration in AHPH-TM. However, R-AHPH-TM cannot exploit those as processing cores within a cluster operate in the same VF domain.

IX. CONCLUSION

We presented a novel energy efficient design paradigm: *AHPH* multicore systems. This paradigm captures the low-cost benefit of homogeneous core design and verification, while delivering remarkable energy efficiency improvements over dynamic adaptation schemes. Using a rigorous experimental methodology combining standard cell library based CAD flows with architectural full system simulation, we demonstrated 11%–22% improvement over state-of-the-art DTM techniques.

REFERENCES

- [1] R. N. Kalla, B. Sinharoy, W. J. Starke, and M. S. Floyd, "Power7: IBM's next-generation server processor," *IEEE Micro*, vol. 30, no. 2, pp. 7–15, Mar.–Apr. 2010.
- [2] M. Tremblay and S. Chaudhry, "A third generation 65 nm 16-core 32-thread plus 32-scout-thread CMT SPARC processor," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2008, pp. 82–83.
- [3] U. Y. Ogras, R. Marculescu, P. Choudhary, and D. Marculescu, "Voltage-frequency island partitioning for GALS-based networks-on-chip," in *Proc. 44th ACM/IEEE Design Autom. Conf.*, Jun. 2007, pp. 110–115.

- [4] G. Semeraro, D. H. Albonesi, S. Dropsho, G. Magklis, S. Dwarkadas, and M. L. Scott, "Dynamic frequency and voltage control for a multiple clock domain microarchitecture," in *Proc. 35th Annu. ACM/IEEE Int. Symp. Microarch.*, Nov. 2002, pp. 356–367.
- [5] S. Herbert and D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," in *Proc. Int. Symp. Low Power Electron. Design*, 2007, pp. 38–43.
- [6] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi, "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *Proc. 39th Annu. IEEE/ACM Int. Symp. Microarch.*, Dec. 2006, pp. 347–358.
- [7] H. Li, C. Y. Cher, K. Roy, and T. N. Vijaykumar, "Combined circuit and architectural level variable supply-voltage scaling for low power," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 5, pp. 564–576, May 2005.
- [8] S. Borkar, "Design perspectives on 22 nm CMOS and beyond," in *Proc. 46th Annu. Design Autom. Conf.*, 2009, pp. 93–94.
- [9] J. Lee and N. S. Kim, "Optimizing total power of many-core processors considering voltage scaling limit and process variations," in *Proc. 14th ACM/IEEE Int. Symp. Low Power Electron. Design*, Aug. 2009, pp. 201–206.
- [10] R. Kumar and G. Hinton, "A family of 45 nm IA processors," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2009, pp. 58–59.
- [11] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proc. Int. Symp. Comput. Arch.*, 2006, pp. 78–88.
- [12] K. K. Rangan, G.-Y. Wei, and D. Brooks, "Thread motion: Fine-grained power management for multi-core systems," in *Proc. 36th Annu. Int. Symp. Comput. Arch.*, Jun. 2009, pp. 302–313.
- [13] V. Hanumaiah, S. Vrudhula, and K. S. Chatha, "Maximizing performance of thermally constrained multi-core processors by dynamic voltage and frequency control," in *Proc. Int. Conf. Comput.-Aided Design*, 2009, pp. 310–313.
- [14] R. Kotla, S. Ghiasi, T. Keller, and F. Rawson, "Scheduling processor voltage and frequency in server and cluster systems," in *Proc. IEEE 19th Int. Parallel Distrib. Process. Symp.*, Apr. 2005, pp. 1–8.
- [15] J. Lee and N. S. Kim, "Optimizing throughput of power- and thermal-constrained multicore processors using DVFS and per-core power-gating," in *Proc. 46th ACM/IEEE Design Autom. Conf.*, Jul. 2009, pp. 47–50.
- [16] R. Jayaseelan and T. Mitra, "A hybrid local-global approach for multi-core thermal management," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2009, pp. 314–320.
- [17] K. Roy, J. P. Kulkarni, and S. K. Gupta, "Device/circuit interactions at 22 nm technology node," in *Proc. 46th ACM/IEEE Design Autom. Conf.*, Jul. 2009, pp. 97–102.
- [18] T.-H. Wu and A. Davoodi, "PaRS: Fast and near-optimal grid-based cell sizing for library-based design," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2008, pp. 107–111.
- [19] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proc. Int. Conf. Comput.-Aided Design*, 1985, pp. 326–328.
- [20] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S.-M. Kang, "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 11, pp. 1621–1634, Nov. 1993.
- [21] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 1994, pp. 1–7.
- [22] M. Ketkar and S. Sapatnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2002, pp. 375–378.
- [23] *OpenSPARC: World's First Free 64-bit Microprocessors*. (2005) [Online]. Available: <http://www.opensparc.net>
- [24] *UMC Free Library*. (2004) [Online]. Available: <http://freelibrary.faraday-tech.com>
- [25] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hällberg, J. Högberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A full system simulation platform," *IEEE Comput.*, vol. 35, no. 2, pp. 50–58, Feb. 2002.
- [26] T. Sherwood, E. Perelman, and B. Calder, "Basic block distribution analysis to find periodic behavior and simulation points in applications," in *Proc. Int. Conf. Parallel Arch. Compilat. Tech.*, Sep. 2001, pp. 3–14.
- [27] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "Architecting efficient interconnects for large caches with CACTI 6.0," *IEEE Micro*, vol. 28, no. 1, pp. 69–79, Jan. 2008.
- [28] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Arch. Code Optim.*, vol. 1, no. 1, pp. 94–125, Mar. 2004.
- [29] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. 27th Annu. Int. Symp. Comput. Arch.*, 2000, pp. 83–94.
- [30] C. R. Lefurgy, A. J. Darke, M. S. Floyd, M. S. Allen-Ware, B. Brock, J. A. Tierno, and J. B. Carter, "Active management of timing guardband to save energy in POWER7," in *Proc. 44th Annu. IEEE/ACM Int. Symp. Microarch.*, Dec. 2011, pp. 1–11.
- [31] R. R. Dobkin, R. Ginosar, and C. P. Sotiriou, "High rate data synchronization in GALS SoCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 10, pp. 1063–1074, Oct. 2006.



Koushik Chakraborty received the Ph.D. degree from the University of Wisconsin, Madison, in 2008.

He is an Assistant Professor with the Electrical and Computer Engineering Department, Utah State University, Logan. His research is funded by the National Science Foundation, Micron Inc. and the State of Utah. His current research interests include power and application aware computer architecture, multicore systems, dynamic specialization, and techniques for holistic system design.



Sanghamitra Roy received the M.S. degree in computer engineering from Northwestern University, Evanston, IL, in 2003, and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin-Madison, Madison. Her doctoral research was sponsored by Intel Strategic CAD Laboratories and National Science Foundation.

She is an Assistant Professor with the Department of Electrical and Computer Engineering, Utah State University, Logan. She has authored over 30 peer reviewed publications in top tier journals and conferences as well as a book chapter in VLSI Design Automation. Her research is funded by the National Science Foundation, Micron Inc. and the State of Utah. Her current research interests include VLSI circuit design and optimization and exploring reliability aware novel circuit styles and architectures.

Dr. Roy serves as a reviewer for the IEEE TVLSI, the IEEE TCAD, Integration—the *VLSI Journal*, the IET Computers and Digital Techniques, the IEEE International Conference on VLSI Design, the IEEE/ACM Design Automation Conference, and the IEEE/ACM International Symposium on Quality Electronic Design. She was a recipient of Best Paper Award nominations from the IEEE Design Automation and Test in Europe in 2011, the IEEE/ACM International Conference on Computer Aided Design in 2005, and the IEEE 23rd International Conference on VLSI Design in 2010.