DOI:10.1145/2408776.2408797

Power Challenges May End the Multicore Era

By Hadi Esmaeilzadeh, Emily Blem, Renée St. Amant, Karthikeyan Sankaralingam, and Doug Burger

Abstract

Starting in 2004, the microprocessor industry has shifted to multicore scaling—increasing the number of cores per die each generation—as its principal strategy for continuing performance growth. Many in the research community believe that this exponential core scaling will continue into the hundreds or thousands of cores per chip, auguring a parallelism revolution in hardware or software. However, while transistor count increases continue at traditional Moore's Law rates, the per-transistor speed and energy efficiency improvements have slowed dramatically. Under these conditions, more cores are only possible if the cores are slower, simpler, or less utilized with each additional technology generation. This paper brings together transistor technology, processor core, and application models to understand whether multicore scaling can sustain the historical exponential performance growth in this energy-limited era. As the number of cores increases, power constraints may prevent powering of all cores at their full speed, requiring a fraction of the cores to be powered off at all times. According to our models, the fraction of these chips that is "dark" may be as much as 50% within three process generations. The low utility of this "dark silicon" may prevent both scaling to higher core counts and ultimately the economic viability of continued silicon scaling. Our results show that core count scaling provides much less performance gain than conventional wisdom suggests. Under (highly) optimistic scaling assumptions—*for parallel workloads*—multicore scaling provides a 7.9× (23% per year) over ten years. Under more conservative (realistic) assumptions, multicore scaling provides a total performance gain of 3.7× (14% per year) over ten years, and obviously less when sufficiently parallel workloads are unavailable. Without a breakthrough in process technology or microarchitecture, other directions are needed to continue the historical rate of performance improvement.

1. INTRODUCTION

Moore's Law[18] (the doubling of transistors on chip every 18 months) has been a fundamental driver of computing. For more than four decades, through transistor, circuit, microarchitecture, architecture, and compiler advances, Moore's Law, coupled with Dennard scaling,[9] has resulted in consistent exponential performance increases. Dennard's scaling theory showed how to reduce the dimensions and the electrical characteristics of a transistor proportionally to enable successive shrinks that simultaneously improved density, speed, and energy efficiency. According to Dennard's theory with a scaling ratio of $\frac{1}{\sqrt{2}}$, the transistor count doubles (Moore's Law), frequency increases by 40%, and the total chip power stays the same from one generation of process technology to the next on a fixed chip area. With the end of Dennard scaling, process technology scaling can sustain doubling the transistor count every generation, but with significantly less improvement in transistor switching speed and energy efficiency. This transistor scaling trend presages a divergence between energy efficiency gains and transistor density increases. The recent shift to multicore designs, which was partly a response to the end of Dennard scaling, aimed to continue proportional performance scaling by utilizing the increasing transistor count to integrate more cores, which leverage application and/or task parallelism.

Given the transistor scaling trends and challenges, it is timely and crucial for the broader computing community to examine whether multicore scaling will utilize each generation's doubling transistor count effectively to sustain the performance improvements we have come to expect from technology scaling. Even though power and energy have become the primary concern in system design, no one knows how severe (or not) the power problem will be for multicore scaling, especially given the large multicore design space.

Through comprehensive modeling, this paper provides a decade-long performance scaling projection for future multicore designs. Our multicore modeling takes into account transistor scaling trends, processor core design options, chip multiprocessor organizations, and benchmark characteristics, while applying area and power constraints at future technology nodes. The model combines these factors to project the upper bound speedup achievable through multicore scaling under current technology scaling trends. The model also estimates the effects of nonideal transistor scaling, including the percentage of *dark silicon*—the fraction of the chip that needs to be powered off at all times—in future multicore chips. Our modeling also discovers the best core organization, the best chip-level topology, and the optimal number of cores for the workloads studied. We do not believe or advocate that designs with dark silicon are ideal or even desirable; in our view smaller chips are more likely. Nonetheless, our modeling shows that—even with the best multicore organization, assuming constant chip size and fixed power budget—a significant portion of the chip will remain dark.

The study shows that regardless of chip organization and topology, multicore scaling is power limited to a degree not

A previous version of this article appears in *Proceedings of the 38th International Symposium on Computer Architecture* (June 2011). Parts of this article appear in *IEEE Micro Top Picks from the Computer Architecture Conferences of 2011* (May/June 2012).

widely appreciated by the computing community. In just five generations, at 8nm, the percentage of dark silicon in a fixed-size chip may grow to 50%. Given the recent trend of technology scaling, the 8nm technology node is expected to be available in 2018. Over this period of ten years (from 2008 when 45nm microprocessors were available), with optimistic international technology roadmap for semiconductors (ITRS) scaling projections,[16] only 7.9× average speedup is possible for commonly used parallel workloads,[4] leaving a nearly 24-fold gap from a target of doubled performance per generation. This gap grows to 28-fold with conservative scaling projections,[5] with which only 3.7× speedup is achievable in the same period. Further investigations also show that beyond a certain point increasing the core count does not translate to meaningful performance gains. These power and parallelism challenges threaten to end the multicore era, defined as the era during which core counts grow appreciably.

2. OVERVIEW

Figure 1 shows how we build and combine three models to project the performance of future multicores. Ultimately, the model predicts the speedup achievable by multicore scaling and shows a gap between our model's projected speedup and the expected exponential speedup with each technology generation. We refer to this gap as the *dark silicon performance gap*, since it is partly the result of the dark silicon phenomenon, or the nonideal transistor scaling that prevents fully utilizing the exponential increases in transistor count. Our modeling considers transistor scaling projections, single-core design scaling, multicore design choices, application characteristics, and microarchitectural features. This study assumes that the die size and the power budget stay the same as technology scales, an assumption in line with the common practice for microprocessor design. Below we briefly discuss each of the three models.

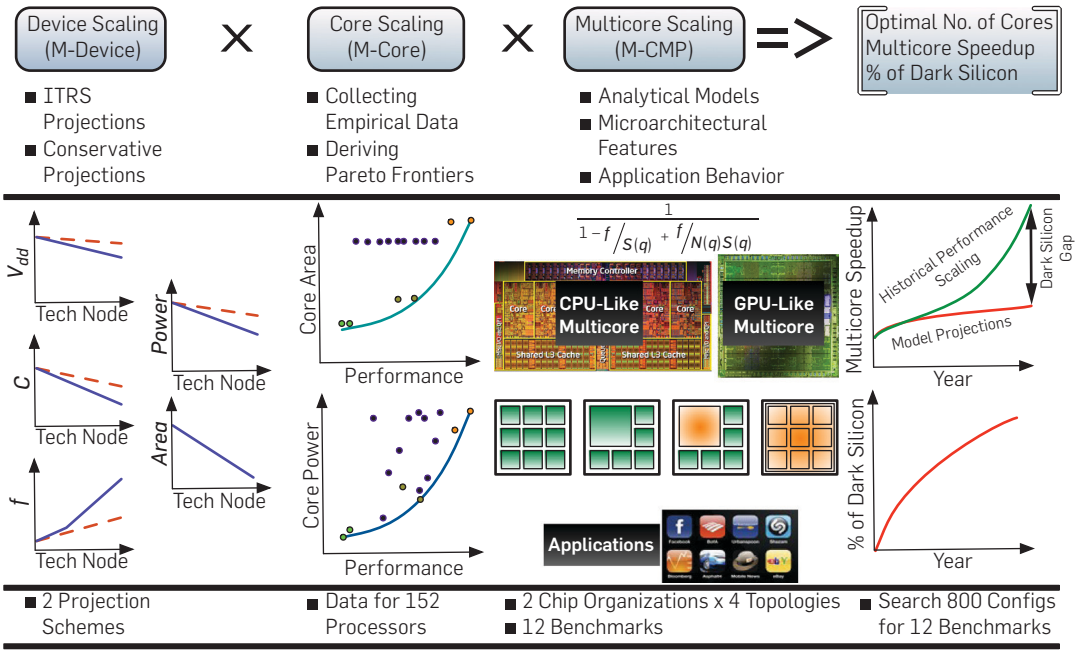
Device scaling model (M-Device). Two device (transistor) scaling models provide the area, power, and frequency scaling factors at technology nodes from 45nm through 8nm. One model is based on aggressive ITRS projections[16] while the other builds on more conservative predictions from Shekhar Borkar's recent study.[5]

Core scaling model (M-Core). Through Pareto-optimal curves, derived from measured data, the M-Core model provides the maximum performance that a single-core can sustain for any given area. Further, it provides the minimum power that is consumed to sustain this level of performance. At each technology node, these two Pareto frontiers, which constitute the M-Core model, define the best-case design space of single cores.

Multicore scaling model (M-CMP). The M-CMP covers two mainstream classes of multicore organizations, multicore CPUs and many-thread GPUs, which represent two extreme points in the threads-per-core spectrum. The CPU multicore organization represents Intel Nehalem-like multicore designs that benefit from large caches and offer relatively high single-thread performance. The GPU multicore organization represents NVIDIA Tesla-like lightweight cores with heavy multithreading support and poor single-thread performance. In modeling each of the two multicore organizations, we consider four topologies: symmetric, asymmetric, dynamic, and composed (also called "fused" in the literature[15]).

Symmetric multicore. The symmetric, or homogeneous, multicore topology consists of multiple copies of the same core operating at the same voltage and frequency

Figure 1. Overview of the methodology and the models.



setting. In a symmetric multicore, the resources, including the power and the area budget, are shared equally across all cores.

Asymmetric multicore. The asymmetric multicore topology consists of one large monolithic core and many identical small cores. This design uses the high-performing large core for the serial portion of code and leverages the numerous small cores as well as the large core to exploit the parallel portion of code.

Dynamic multicore. The dynamic multicore topology is a variation of the asymmetric multicore topology. During parallel code portions, the large core is shut down and, conversely, during the serial portion, the small cores are turned off and the code runs only on the large core.[6, 21] Switching the cores off and on allows integrating more cores or using a higher voltage and frequency operational setting.

Composed multicore. The composed multicore topology is a collection of small cores that can dynamically merge together and compose a logical higher performance large core.[15, 17] While providing a parallel substrate for the parallel portion of code when unmerged, the small cores merge and compose a logical core that offers higher single-threaded performance for the serial portion.

The multicore model is an analytic model that computes the multicore performance and takes the core performance as input (obtained from M-Core), the multicore organization (CPU-like or GPU-like), and multicore topology (symmetric, asymmetric, dynamic, and composed). Unlike previous studies, the model also takes into account application characteristics such as memory access pattern, and the amount of thread-level parallelism in the workload as well as the microarchitectural features such as cache size and memory bandwidth. We choose the PARSEC benchmarks[4] to study the multicore scaling potential for successfully parallelized applications. PARSEC is a set of highly parallel applications that are widely used to support the parallel architecture research.

Modeling future multicore chips. To model future multicore chips, we first model the building blocks, the future cores. We combine the device and core models to project the best-case design space of single cores—the Pareto frontiers—at future technology nodes. Any performance improvement for future cores will come at the cost of area or power as defined by the projected Pareto frontiers. Then, we combine all three models and perform an exhaustive design-space search to find the optimal multicore configuration for each individual application considering its characteristics. The optimal configuration delivers the maximum multicore speedup for each benchmark at future technology nodes while enforcing area and power constraints. The gap between the projected speedup and the speedup we have come to expect with each technology generation is the dark silicon performance gap.

Related work. Other work has studied various subsets of the problem that we study comprehensively. Hill and Marty extend Amdahl's Law to model multicore architectures with different topologies.[14] Hempstead et al. introduce a variant of Amdahl's Law to estimate the amount of specialization required to maintain 1.5× performance growth per year, assuming completely parallelizable code.[13] Chung et al. study unconventional cores including custom logic, FPGAs, or GPUs in heterogeneous single-chip designs.[7] Azizi et al. derive the single-core energy/performance tradeoff as Pareto frontiers using architecture-level statistical models combined with circuit-level energy-performance tradeoff functions.[2] Chakraborty considers device-scaling and estimates a simultaneous activity factor for technology nodes down to 32nm.[6] Venkatesh et al. estimate technology-imposed utilization limits and motivate energy-efficient and application-specific core designs.[22] Hardavellas et al. forecast the limits of multicore scaling and the emergence of dark silicon in servers with workloads that have an inherent abundance of parallelism.[12]

3. DEVICE MODEL (M-DEVICE)

The first step in projecting gains from more cores is developing a model that captures future transistor scaling trends. To highlight the challenges of nonideal device scaling, first we present a simplified overview of historical Dennard scaling and the more recent scaling trends.

Historical device scaling trends. According to Dennard scaling, as the geometric dimensions of transistors scale, the electric field within the transistors stays constant if other physical features, such as the gate oxide thickness and doping concentrations, are reduced proportionally. To keep the electric field constant, the supply voltage (the switch on voltage) as well as the threshold voltage (the voltage level below which the transistor switches off) need to be scaled at the same rate as the dimensions of the transistor. With Dennard scaling, a 30% reduction in transistor length and width results in a 50% decrease in transistor area, doubling the number of transistors that can fit on chip with each technology generation (Moore's Law[18]). Furthermore, the decrease in transistor sizes results in a 30% reduction in delay. In total, Dennard scaling suggests a 30% reduction in delay (hence 40% increase in frequency), a 50% reduction in area, and a 50% reduction in power per transistor. As a result, the chip power stays the same as the number of transistors doubles from one technology node to the next in the same area.

Recent device scaling trends. At recent technology nodes, the rate of supply voltage scaling has dramatically slowed due to limits in threshold voltage scaling. Leakage current increases exponentially when the threshold voltage is decreased, limiting threshold voltage scaling, and making leakage power a significant and first-order constraint. Additionally, as technology scales to smaller nodes, physics limits decreases in gate oxide thickness. These two phenomena were not considered in the original Dennard scaling theory, since leakage power was not dominant in the older generations, and the physical limits of scaling oxide thickness were too far out to be considered. Consequently, Dennard scaling stopped at 90nm.[8] That is, transistor area continues to scale at the historic rate, which allows for doubling the number of transistors, while the power per transistor is not scaling at the same rate. This disparity will translate to an increase in chip power if the fraction of active transistors is not reduced from one technology generation

to the next.[6] The shift to multicore architectures was partly a response to the end of Dennard scaling.

3.1. Model structure

The device model provides transistor area, power, and frequency scaling factors from a base technology node (e.g. 45nm) to future technologies. The area scaling factor corresponds to the shrinkage in transistor dimensions. The frequency scaling factor is calculated based on the fan-out of 4 (FO4) delay reduction. FO4 is a process independent delay metric used to measure the delay of CMOS logic that identifies the processor frequency. FO4 is the delay of an inverter, driven by an inverter 4× smaller than itself, and driving an inverter 4× larger than itself. The power scaling factor is computed using the predicted frequency, voltage, and gate capacitance scaling factors in accordance with the $P = \alpha C V_{dd}^2 f$ equation.

3.2. Model implementation

We generate two device scaling models: ITRS and conservative. The ITRS model uses projections from the ITRS 2010 technology roadmap.[16] The conservative model is based on predictions by Shekhar Borkar and represents a less optimistic view.[5] The parameters used for calculating the power and performance scaling factors are summarized in Table 1. We allocate 20% of the chip-power budget to leakage power and assume chip designers can maintain this ratio.

4. CORE MODEL (M-CORE)

The second step in estimating future multicore performance is modeling a key building block, the processor core.

4.1. Model structure

We build a technology-scalable core model by populating the area/performance and power/performance design spaces with the data collected for a set of processors; all fabricated in the same technology node. The core model is the combination of the area/performance Pareto frontier, $A(q)$, and the power/performance Pareto frontier, $P(q)$, for these two design spaces, where q is the single-threaded performance of a core. These frontiers capture the best-case area/performance and power/performance tradeoffs for a core while abstracting away specific details of the core. We use the device scaling model to project the frontiers to future technologies and model performance, area, and power of cores fabricated at those nodes.

Table 1. Scaling factors with ITRS and conservative projections.

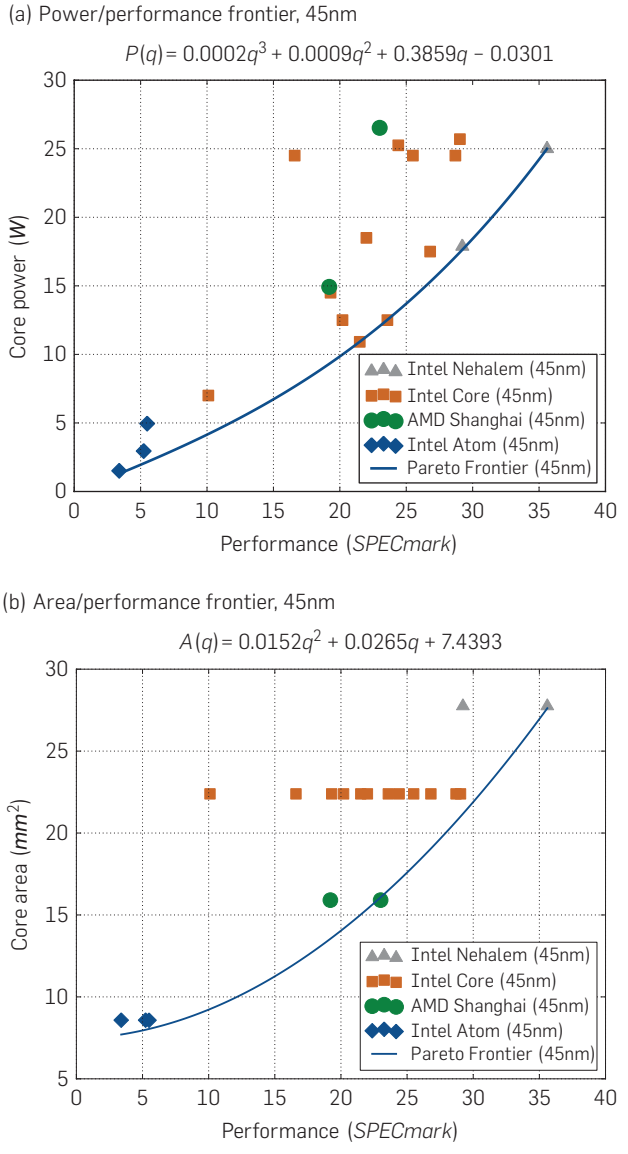
	Tech node (nm)	Frequency scaling factor (/45nm)	V_{dd} scaling factor (/45nm)	Capacitance scaling factor (/45nm)	Power scaling factor (/45nm)
ITRS	45*	1.00	1.00	1.00	1.00
	32*	1.09	0.93	0.70	0.66
	22†	2.38	0.84	0.33	0.54
	16†	3.21	0.75	0.21	0.38
	11†	4.17	0.68	0.13	0.25
	8†	3.85	0.62	0.08	0.12
	31% frequency increase and 35% power reduction per node				
Conservative	45	1.00	1.00	1.00	1.00
	32	1.10	0.93	0.75	0.71
	22	1.19	0.88	0.56	0.52
	16	1.25	0.86	0.42	0.39
	11	1.30	0.84	0.32	0.29
	8	1.34	0.84	0.24	0.22
	6% frequency increase and 23% power reduction per node				

*Extended planar bulk transistors.
†Multi-gate transistors.

4.2. Model implementation

As Figure 2 depicts, we populate the two design spaces at 45nm using 20 representative Intel and AMD processors and derive the Pareto frontiers. The curve that bounds all power(area)/performance points in the design space and minimizes power(area) required for a given level of performance constructs the Pareto frontier. The polynomials $P(q)$ and $A(q)$ are the core model. The core performance, q, is the processor's SPECmark and is collected from the SPEC

Figure 2. Design space and derivation of the Pareto frontiers.



Website.[20] We estimate the core power budget using the TDP reported in processor datasheets. TDP is the chip-power budget, or the amount of power the chip can dissipate without exceeding the transistor junction temperature. We use die photos of the four microarchitectures, Intel Atom, Intel Core, AMD Shanghai, and Intel Nehalem, to estimate the core areas (excluding level 2 and level 3 caches). Since the focus of this work is to study the impact of technology constraints on logic scaling rather than cache scaling, we derive the Pareto frontiers using only the portion of power budget and area allocated to the core in each processor excluding the uncore components.

As illustrated in Figure 2, a cubic polynomial, $P(q)$, is fit to the points along the edge of the power/performance design space and a quadratic polynomial (Pollack's rule[19]), $A(q)$, to the points along the edge of the area/performance design space. The Intel Atom Z520 with an estimated 1.89 W TDP per core represents the lowest power design (lower-left frontier point), and the Nehalem-based Intel Core i7-965 Extreme Edition with an estimated 31.25 W TDP per core represents the highest performing design (upper-right frontier point). The points along the scaled Pareto frontier are used as the search space for determining the best core configuration by the multicore model.

5. MULTICORE MODEL (M-CMP)

The last step in modeling multicore scaling is to develop a detailed chip-level model (M-CMP) that integrates the area and power frontiers, microarchitectural features, and application behavior, while accounting for the chip organization (CPU-like or GPU-like) and its topology (symmetric, asymmetric, dynamic, or composed).

5.1. Model structure

Guz et al. proposed a model to consider first-order impacts of microarchitectural features (cache organization, memory bandwidth, number of threads per core, etc.) and workload characteristics (memory access pattern).[10] To first order, their model considers stalls due to memory dependences and resource constraints (bandwidth or functional units). We extend their approach to build our multicore model. Our extensions incorporate additional application characteristics, microarchitectural features, and physical constraints, and covers both homogeneous and heterogeneous multicore topologies.

This model uses single-threaded cores with large caches to cover the CPU multicore design space and massively threaded cores with minimal caches to cover the GPU multicore design while modeling all four topologies. The input parameters to the model, and how, if at all, they are affected by the multicore design choices are listed in Table 2.

Multicore topologies. The multicore model is an extended Amdahl's Law[1] equation that incorporates the multicore performance (*Perf*) calculated from (2)–(5):

$$Speedup = 1 / \left(\frac{f}{S_{Parallel}} + \frac{1-f}{S_{Serial}} \right) \quad (1)$$

The M-CMP model (1) measures the multicore speedup with respect to a baseline multicore ($Perf_B$). That is, the parallel portion of code (f) is sped up by $S_{Parallel} = Perf_P/Perf_B$ and the serial portion of code ($1 - f$) is sped up by $S_{Serial} = Perf_S/Perf_B$.

Table 2. M-CMP parameters with default values from 45nm Nehalem.

Parameter	Description	Default	Affected by
N	Number of cores	4	Multicore topology
T	Number of threads per core	1	Core style
$freq$	Core frequency (MHz)	3200	Core performance
CPI	Cycles per instruction (zero-latency cache accesses)	1	Core performance, application
C_1	L1 cache size per core (KB)	64	Core style
C_2	L2 cache size per chip (MB)	2	Core style, multicore topology
t_1	L1 access time (cycles)	3	–
t_2	L2 access time (cycles)	20	–
t	Memory access time (cycles)	426	Core performance
BW	Maximum memory bandwidth (GB/s)	200	Technology node
b	Bytes per memory access (B)	64	–
f	Fraction of code that can be parallel	Varies	Application
r	Fraction of instructions that are memory accesses	Varies	Application
α_1, β_1	L1 cache miss rate function constants	Varies	Application
α_2, β_2	L2 cache miss rate function constants	Varies	Application

The number of cores that fit on the chip is calculated as follows based on the topology of the multicore, its area budget (*AREA*), its power budget (*TDP*), each core's area ($A(q)$), and each core's power ($P(q)$).

$$N_{Symm}(q) = \min\left(\frac{AREA}{A(q)}, \frac{TDP}{P(q)} \right)$$

$$N_{Asym}(q_L, q_S) = \min\left(\frac{AREA - A(q_L)}{A(q_S)}, \frac{TDP - P(q_L)}{P(q_S)} \right)$$

$$N_{dynm}(q_L, q_S) = \min\left(\frac{AREA - A(q_L)}{A(q_S)}, \frac{TDP}{P(q_S)} \right)$$

$$N_{Comp}(q_L, q_S) = \min\left(\frac{AREA}{(1+\tau)A(q_S)}, \frac{TDP}{P(q_S)} \right)$$

For heterogeneous multicores, q_S is the single-threaded performance of the small cores and q_L is the single-threaded performance of the large core. The area overhead of supporting composability is τ; however, no power overhead is assumed for composability support.

Microarchitectural features. Multithreaded performance (*Perf*) of an either CPU-like or GPU-like multicore running a fully parallel (f = 1) and multithreaded application is

calculated in terms of instructions per second in (2) by multiplying the number of cores (N) by the core utilization (η) and scaling by the ratio of the processor frequency to CPI_{exe}:

$$Perf = \min\left(N\frac{freq}{CPI_{exe}}\eta, \frac{BW_{max}}{r_m \times m_{L1} \times m_{L2} \times b}\right) \quad (2)$$

The CPI_{exe} parameter does not include stalls due to cache accesses, which are considered separately in the core utilization (η). The core utilization is the fraction of time that a thread running on the core can keep it busy. It is modeled as a function of the average time spent waiting for each memory access (t), fraction of instructions that access the memory (r_m), and the CPI_{exe}:

$$\eta = \min\left(1, \frac{T}{1 + t\frac{r_m}{CPI_{exe}}}\right) \quad (3)$$

The average time spent waiting for memory accesses (t) is a function of the time to access the caches (t_{L1} and t_{L2}), time to visit memory (t_{mem}), and the predicted cache miss rate (m_{L1} and m_{L2}):

$$t = (1-m_{L1})t_{L1} + m_{L1}(1-m_{L2})t_{L2} + m_{L1}m_{L2}t_{mem} \quad (4)$$

$$m_{L1} = \left(\frac{C_{L1}}{T\beta_{L1}}\right)^{1-\alpha_{L1}} \text{ and } m_{L2} = \left(\frac{C_{L2}}{NT\beta_{L2}}\right)^{1-\alpha_{L2}} \quad (5)$$

5.2. Model implementation

The M-CMP model incorporates the Pareto frontiers, physical constraints, real application characteristics, and realistic microarchitectural features into the multicore speedup projections as discussed below.

Application characteristics. The input parameters that characterize an application are its cache behavior, fraction of instructions that are loads or stores, and fraction of parallel code. For the PARSEC benchmarks, we obtain this data from two previous studies.[3, 4] To obtain the fraction of parallel code (f) for each benchmark, we fit an Amdahl's Law-based curve to the reported speedups across different numbers of cores from both studies. The value of f ranges from 0.75 to 0.9999 depending on the benchmark.

Obtaining frequency and CPI_{exe} from Pareto frontiers. To incorporate the Pareto-optimal curves into the M-CMP model, we convert the SPECmark scores (q) into an estimated CPI_{exe} and core frequency. We assume the core frequency scales linearly with performance, from 1.5 GHz for an Atom core to 3.2 GHz for a Nehalem core. Each application's CPI_{exe} is dependent on its instruction mix and use of hardware resources (e.g., functional units and out-of-order issue width). Since the measured CPI_{exe} for each benchmark at each technology node is not available, we use the M-CMP model to generate per benchmark CPI_{exe} estimates for each design point along the Pareto frontier. With all other model inputs kept constant, we iteratively search for the CPI_{exe} at each processor design point. We start by assuming that the Nehalem core has a CPI_{exe} of ℓ. Then, the smallest core, an Atom processor, should have a CPI_{exe} such that the ratio of its M-CMP performance to the Nehalem core's M-CMP performance is the same as the ratio of their SPECmark scores (q). We assume CPI_{exe} does not change as technology scales, while frequency does change as discussed in Section 6.1.

Microarchitectural features. A key part of the detailed model is the set of input parameters that model the microarchitecture of the cores. For single-thread (ST) cores, we assume each core has a 64KB L1 cache, and chips with only ST cores have an L2 cache that is 30% of the chip area. Many-thread (MT) cores have small L1 caches (32KB for every 8 cores), support multiple hardware contexts (1024 threads per 8 cores) and a thread register file, and have no L2 cache. From Atom and Tesla die photos, we estimate that 8 small MT cores, their shared L1 cache, and their thread register file can fit in the same area as one Atom processor. We assume that off-chip bandwidth (BW_{max}) increases linearly as process technology scales down while the memory access time is constant.

Composed multicores. We assume that τ (area overhead of composability) increases from 10% to 400% depending on the total area of the composed core and performance of the composed core cannot exceed performance of a single Nehalem core at 45nm.

Constraints and baseline. The area and power budgets are derived from the highest-performing quad-core Nehalem multicore at 45nm excluding the L2 and L3 caches. They are 111 mm^2 and 125 W, respectively. The M-CMP multicore speedup baseline is a quad-Nehalem multicore that fits in the area and power budgets. The reported dark silicon projections are for the area budget that is solely allocated to the cores, not caches and other 'uncore' components. The actual fraction of chip that goes dark may be higher.

6. COMBINING MODELS

6.1. Device × core model

To study core scaling in future technology nodes, we scaled the 45nm Pareto frontiers down to 8nm by scaling the power and performance of each processor data point using the *DevM* model and then re-fitting the Pareto optimal curves at each technology node. Performance, measured in SPECmark, is assumed to scale linearly with frequency. This optimistic assumption ignores the effects of memory latency and bandwidth on the core performance, and thus actual performance gains through scaling may be lower. Based on the optimistic ITRS model, scaling a microarchitecture (core) from 45nm to 8nm will result in a 3.9× performance improvement and an 88% reduction in power consumption. Conservative scaling, however, suggests that performance will increase only by 34%, and power will decrease by 74%.

6.2. Device × core × multicore model

All three models are combined to produce final projections on optimal multicore speedup, optimal number of cores, and amount of dark silicon. To determine the

best multicore configuration at each technology node, we sweep the design points along the scaled area/performance and power/performance Pareto frontiers (M-Device × M-Core) as these points represent the most efficient designs. At each technology node, for each core design on the scaled frontiers, we construct a multicore chip consisting of one such core. For a symmetric multicore chip, we iteratively add identical cores one by one until the area or power budget is hit, or performance improvement is limited. We sweep the frontier and construct a symmetric multicore for each processor design point. From this set of symmetric multicores, we pick the multicore with the best speedup as the optimal symmetric multicore for that technology node. The procedure is similar for other topologies. This procedure is performed separately for CPU-like and GPU-like organizations. The amount of dark silicon is the difference between the area occupied by cores for the optimal multicore and the area budget allocated to the cores.

7. SCALING AND FUTURE MULTICORES

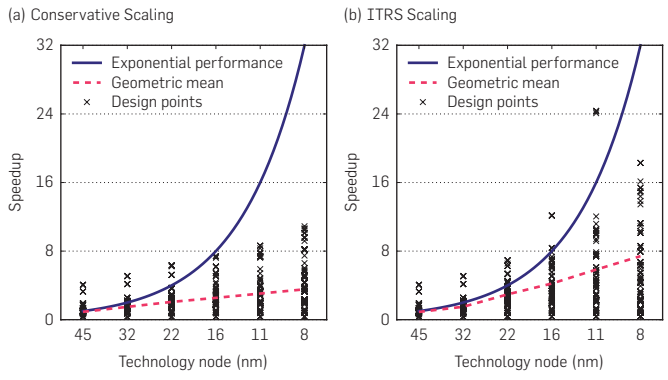
We apply the combined models to study the future of multicore designs and their performance-limiting factors. The results from this study provide a detailed analysis of multicore behavior for future technologies considering 12 real applications from the PARSEC suite.

7.1. Speedup projections

Figure 3 summarizes all of the speedup projections in a single scatter plot. For every benchmark at each technology node, we plot the speedup of eight possible multicore configurations (CPU-like, GPU-like) × (symmetric, asymmetric, dynamic, composed). The solid line is exponential performance scaling–doubling performance every technology generation.

With optimal multicore configurations for each individual application, at 8nm, only 3.7× (conservative scaling) or 7.9× (ITRS scaling) geometric mean speedup is possible, as shown by the dashed line in Figure 3.

Figure 3. Speedup across process technology nodes across all organizations and topologies with PARSEC benchmarks.



Highly parallel workloads with a degree of parallelism higher than 99% will continue to benefit from multicore scaling.

At 8nm, the geometric mean speedup for dynamic and composed topologies is only 10% higher than the geometric mean speedup for symmetric topologies.

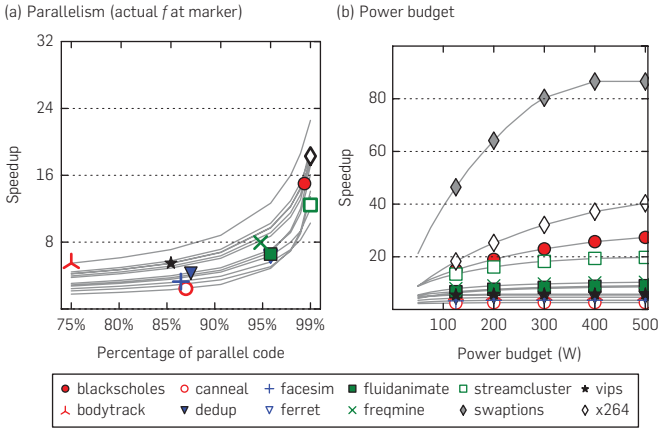
7.2. Dark silicon projections

With ITRS projections, at 8nm, over 50% of the chip will be dark and cannot be utilized.

To understand whether parallelism or the power constraint is the primary source of the dark silicon performance gap, we vary each of these factors in two experiments at 8nm. First, as depicted in Figure 4(a), we keep the power budget constant (our default budget is 125 W), and vary the level of parallelism in the PARSEC applications from 0.75 to 0.99, assuming programmer effort can realize this improvement. We see performance improves slowly as the parallelism level increases, with most benchmarks reaching a speedup of about only 15× at 99% parallelism. Provided that the power budget is the only limiting factor, typical upper-bound ITRS-scaling speedups would still be limited to 15×. With conservative scaling, this best-case speedup is limited to 6.3×.

For the second experiment, we keep each application's parallelism at its real level and vary the power budget from 50 W to 500 W. As Figure 4(b) shows, eight of 12 benchmarks show no more than 10× speedup even with a practically unlimited power budget. That is, increasing core counts beyond a certain point does not improve performance due to the limited parallelism in the applications and the Amdahl's Law. Only four benchmarks have sufficient parallelism to even hypothetically sustain the exponential level of speedup.

Figure 4. Impact of application parallelism and power budget on speedup at 8nm.



The level of parallelism in PARSEC applications is the primary contributor to the dark silicon performance gap. However, in realistic settings the dark silicon resulting from power constraints limits the achievable speedup.

7.3. Core count projections

Different applications saturate performance improvements at different core counts. We consider the chip configuration that provides the best speedup for all of the applications as an *ideal* configuration. Figure 5 shows the number of cores (solid line) for the ideal CPU-like dynamic multicore configuration across technology generations. We choose the dynamic topology since it delivers the highest performance. The dashed line illustrates the number of cores required to achieve 90% of the ideal configuration's geometric mean speedup across PARSEC benchmarks. As depicted, with ITRS scaling, the ideal configuration integrates 442 cores at 8nm; however, 35 cores reach the 90% of the speedup achievable by 442 cores. With conservative scaling, the 90% speedup core count is 20 at 8nm.

Due to limited parallelism in the PARSEC suite, even with novel heterogeneous topologies and optimistic ITRS scaling, integrating more than 35 cores improves performance only slightly for CPU-like topologies.

7.4. Sensitivity studies

Our analysis thus far examined typical configurations and showed poor scalability for the multicore designs. A natural question is, *can simple configuration changes (percentage cache area, memory bandwidth, etc.) provide significant benefits that can bridge the dark silicon gap?* We investigate three such simple changes: L2 cache size, memory bandwidth, and simultaneous multithreading (SMT).

L2 cache area. Figure 6(a) shows the optimal speedup at 45nm as the amount of a symmetric CPU's chip area devoted to L2 cache varies from 0% to 100%. In this study we ignore any increase in L2 cache power or increase in L2 cache access latency. Across the PARSEC benchmarks, the optimal percentage of chip devoted to cache varies from 20% to 50% depending on benchmark memory access characteristics. Compared to the unified 30% cache area for all the applications, using each application's optimal cache area improves performance merely by at most 20% across all benchmarks.

Memory bandwidth. Figure 6(b) illustrates the sensitivity of PARSEC performance to the available memory bandwidth for symmetric GPU multicores at 45nm. As the memory bandwidth increases, the speedup improves since more threads can be fed with data; however, the benefits are limited by power and/or parallelism and in 10 out of 12 benchmarks speedups do not increase by more than 2× compared to the baseline, 200GB/s.

SMT. To simplify the discussion, we did not consider SMT support for the processors (cores) in the CPU multicore organization. SMT support can improve power efficiency of the cores for parallel workloads to some extent. We studied 2-way, 4-way, and 8-way SMT with no area or energy penalty, and observed that speedup improves with 2-way SMT by 1.5× in the best case and decreases as much as 0.6× in the worst case due to increased cache contention; the range for 8-way SMT is 0.3–2.5×.

8. ASSUMPTIONS AND LIMITATIONS

We discuss some of the important limitations of our model and argue that they do not significantly change our final results.

Dynamic voltage and frequency scaling (DVFS). Our device and core models do not explicitly consider dynamic voltage and frequency scaling; instead, we take an optimistic approach to account for their best-case settings. When deriving the Pareto frontiers, we set each processor to operate at its optimal voltage and frequency setting ($V_{dd_{min}}$, $Freq_{max}$). At a fixed V_{dd} setting, scaling down the frequency from $Freq_{max}$ results in a power/performance point inside the optimal Pareto curve, a suboptimal design point. However, scaling

Figure 5. Number of cores for the ideal CPU-like dynamic multicore configurations and the number of cores delivering 90% of the speedup achievable by the ideal configurations across the PARSEC benchmarks.

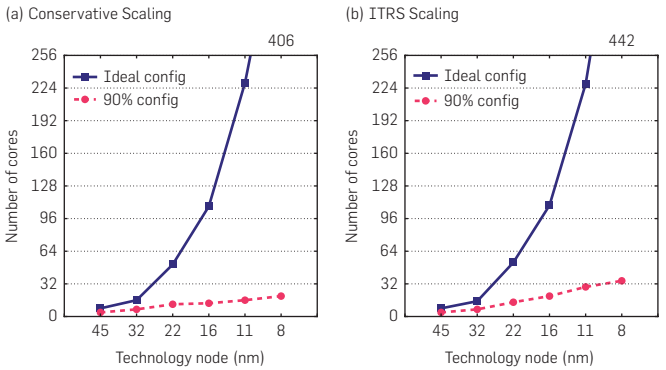
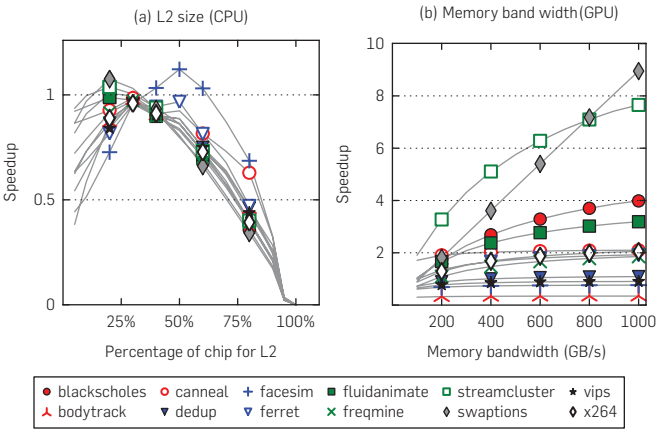


Figure 6. Effect of L2 size and memory bandwidth on speedup at 45nm.



voltage up and operating at a new ($V'_{dd_{min}}, Freq'_{max}$) setting results in a different power-performance point that is still at the optimal frontier. Since we investigate all of the points along the frontier to find the optimal multicore configuration, our study covers multicore designs that induce heterogeneity to symmetric topologies through dynamic voltage and frequency scaling.

Architecture details in multicore model and validation. The multicore model considers the first-order impact of caching, parallelism, and threading under assumptions that result only in optimistic projections (i.e., favorable multicore scaling). Comparing the output of the M-CMP model against published empirical results confirm that our model always overpredicts multicore performance. The model optimistically assumes that the workload is homogeneous, work is infinitely parallel during parallel sections of code, memory accesses never stall due to a previous access, and no thread synchronization, operating system serialization, or swapping occurs.

Server workloads. We do not directly study the server workloads, a domain where applications are highly concurrent and embarrassingly parallel. However, even in these types of workloads, resource scheduling and structural hazards such as competition for cache, memory bandwidth, DRAM storage, SSD IO, network IO, etc. limit parallelism. These factors induce a serial portion to the execution of the workloads. The key challenge is to measure the amount of serialization from these structural hazards, which is an interesting future study. Once the amount of serialization is measured, our models can be applied to the server workloads to project the amount of dark silicon and its effects. Generally, if the effective parallelism is less than 99%, the results suggest that dark silicon and its effects will manifest. Furthermore, Hardavellas et al. forecast the limits of multicore scaling and the emergence of dark silicon in servers with workloads that have an inherent abundance of parallelism.[12] They project that for server workloads such as online transaction processing (OLTP), decision support systems (DSS), and Web server (Apache), even with 3D-stacked memory, a significant amount of a multicore chip will be dark as technology scales. They determine that power and limited and nonscalable off-chip bandwidth are the primary limiting factors to multicore performance scaling and result in dark silicon for server workloads.

Alternative cores. We do not consider embedded ARM or Tilera cores in this work because they are designed for restricted domains and their SPECmark scores are not available for a meaningful comparison.

9. A PATH FORWARD?

For decades, Moore's Law *plus* Dennard scaling permitted more transistors, faster transistors, and more energy efficient transistors with each new process node, justifying the enormous costs required to develop each new process node. Dennard scaling's failure led industry to race down the multicore path, which for sometime permitted performance scaling for parallel and multitasked workloads, allowing the economics of process scaling to hold. A key question for the computing community is whether scaling multicores will provide the performance and value needed to scale down many more technology generations. Are we in a long-term "multicore era," or will it instead be a "multicore decade" (2004–2014)? Will industry need to move in different, perhaps radical, directions to justify the cost of scaling? To answer the question, this paper models an upper bound on parallel application performance available from multicore and CMOS scaling–assuming no major disruptions in process scaling or core efficiency. Using a constant area and power budget, this study showed that the space of known multicore designs (CPU, GPU, their hybrids) or novel heterogeneous topologies (e.g., dynamic or composable) falls far short of the historical performance gains to which the microprocessor industry is accustomed. Even with aggressive ITRS scaling projections, scaling cores achieve a geometric mean 7.9× speedup in 10 years at 8nm—a 23% annual gain. Our findings suggest that without process breakthroughs, directions beyond multicore are needed to provide performance scaling. There are reasons to be both pessimistic and optimistic.

9.1. Pessimistic view

A pessimistic interpretation of this study is that the performance improvements to which we have grown accustomed over the past 40 years are unlikely to continue with multicore scaling as the primary driver. The transition from multicore to a new approach is likely to be more disruptive than the transition to multicore. Furthermore, to sustain the current cadence of Moore's Law, the transition needs to be made in only a few years, much shorter than the traditional academic time frame for research and technology transfer. Major architecture breakthroughs in "alternative" directions such as neuromorphic computing, quantum computing, or bio-integration will require even more time to enter the industrial product cycle. Furthermore, while a slowing of Moore's Law will obviously not be fatal, it has significant economic implications for the semiconductor industry.

9.2. Optimistic view

Technology. The study shows if energy efficiency breakthroughs are made on supply voltage and process scaling, the performance improvement potential for multicore scaling is still high for applications with very high degrees of parallelism.

The need for microarchitecture innovations. Our study shows that fundamental processing limitations emanate from the processor core. The limited improvements on single-threaded performance is the inhibiting factor. Clearly, architectures that move well past the power/performance Pareto-optimal frontier of today's designs are necessary to bridge the dark silicon gap and utilize the increases in transistor count. Hence, improvements to the processor core efficiency will have significant impact on performance improvement and will enable technology scaling even though the core consumes

only 20% of the power budget for an entire laptop, smartphone, tablet, etc. When performance becomes limited, microarchitectural techniques that occasionally use parts of the chip to deliver outcomes orthogonal to performance, such as security, programmer productivity, and software maintainability are ways to sustain the economics of the industry. We believe this study will revitalize and trigger microarchitecture innovations, making the case for their urgency and their potential impact.

Efficiency through specialization. Recent work has quantified three orders of magnitude difference in efficiency between general-purpose processors and ASICs.[11] However, there is a well-known tension between efficiency and programmability. Designing ASICs for the massive base of quickly changing general-purpose applications is currently infeasible. Programmable accelerators, such as GPUs and FPGAs, and specialized hardware can provide an intermediate point between the efficiency of ASICs and the generality of conventional processors, gaining significant improvements for specific domains of applications. Even though there is an emerging consensus that specialization and acceleration is a promising approach for efficiently utilizing the growing number of transistors, developing programming abstractions that allow general-purpose applications to leverage specialized hardware and programmable accelerators remain challenging.

Opportunity for disruptive innovations. Our study is based on a model that takes into account properties of devices, processor cores, multicore organizations, and topologies. Thus the model inherently provides the areas to focus on for innovation. To surpass the dark silicon performance barrier highlighted by our work, designers must develop systems that use significantly more energy-efficient techniques. Some examples include device abstractions beyond digital logic (error-prone devices); processing paradigms beyond superscalar, SIMD, and SIMT; and program semantic abstractions allowing probabilistic and approximate computation. There is an emerging synergy between the applications that can tolerate approximation and the unreliability in the computation fabric as technology scales down. If done in a disciplined manner, relaxing the high tax of providing perfect accuracy at the device, circuit, and architecture level can provide a huge opportunity to improve performance and energy efficiency for the domains in which applications can tolerate approximate computation yet deliver acceptable outputs. Our results show that such radical departures are needed and the model provides quantitative measures to examine the impact of such techniques.

The model we have developed in the paper is useful to determine an optimal multicore configuration given a workload set, a power and area budget, and a technology generation. It can also be used to project expected multicore performance for the best configurations under a range of assumptions. We have made the models available for general use at the following URL: http://research.cs.wisc.edu/vertical/DarkSilicon.

Acknowledgments

We thank Shekhar Borkar for sharing his personal views on how CMOS devices are likely to scale. Support for this research was provided by NSF under the following grants: CCF-0845751, CCF-0917238, and CNS-0917213. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

References

1. Amdahl, G.M. Validity of the single processor approach to achieving large-scale computing capabilities. In *AFIPS* (1967).
2. Azizi, O., Mahesri, A., Lee, B.C., Patel, S.J., Horowitz, M. Energy-performance tradeoffs in processor architecture and circuit design: A marginal cost analysis. In *ISCA* (2010).
3. Bhadauria, M., Weaver, V., McKee, S. Understanding PARSEC performance on contemporary CMPs. In *IISWC* (2009).
4. Bienia, C., Kumar, S., Singh, J.P., Li, K. The PARSEC benchmark suite: Characterization and architectural implications. In *PACT* (2008).
5. Borkar, S. The exascale challenge. Keynote at *VLSI-DAT* (2010).
6. Chakraborty, K. Over-provisioned multicore systems. PhD thesis, University of Wisconsin-Madison (2008).
7. Chung, E.S., Milder, P.A., Hoe, J.C., Mai, K. Single-chip heterogeneous computing: Does the future include custom logic, FPGAs, and GPUs? In *MICRO* (2010).
8. Dennard, R.H., Cai, J., Kumar, A. A perspective on today's scaling challenges and possible future directions. *Solid-State Electron. 5,* 4 (Apr. 2007).
9. Dennard, R.H., Gaensslen, F.H., Rideout, V.L., Bassous, E., LeBlanc, A.R. Design of ion-implanted mosfet's with very small physical dimensions. *IEEE J. Solid-State Circuits 9* (Oct. 1974).
10. Guz, Z., Bolotin, E., Keidar, I., Kolodny, A., Mendelson, A., Weiser, U.C. Many-core vs. many-thread machines: Stay away from the valley. *IEEE Comput. Archit. Lett. 8* (Jan. 2009).
11. Hameed, R., Qadeer, W., Wachs, M., Azizi, O., Solomatnikov, A., Lee, B.C., Richardson, S., Kozyrakis, C., Horowitz, M. Understanding sources of inefficiency in general-purpose chips. In *ISCA* (2010).
12. Hardavellas, N., Ferdman, M., Falsafi, B., Ailamaki, A. Toward dark silicon in servers. *IEEE Micro 31,* 4 (Jul.–Aug. 2011).
13. Hempstead, M., Wei, G.Y., Brooks, D. Navigo: An early-stage model to study power-contrained architectures and specialization. In *MoBS* (2009).
14. Hill, M.D., Marty, M.R. Amdahl's law in the multicore era. *Computer 41,* 7 (Jul. 2008).
15. Ipek, E., Kirman, M., Kirman, N., Martinez, J.F. Core fusion: Accommodating software diversity in chip multiprocessors. In *ISCA* (2007).
16. ITRS. International technology roadmap for semiconductors, the 2010 update, http://www.itrs.net (2011).
17. Kim, C., Sethumadhavan, S., Govindan, M.S., Ranganathan, N., Gulati, D., Burger, D., Keckler, S.W. Composable lightweight processors. In *MICRO* (2007).
18. Moore, G.E. Cramming more components onto integrated circuits. *Electronics 38,* 8 (Apr. 1965).
19. Pollack, F. New microarchitecture challenges in the coming generations of CMOS process technologies. Keynote at *MICRO* (1999).
20. SPEC. Standard performance evaluation corporation, http://www.spec.org (2011).
21. Suleman, A.M., Mutlu, O., Qureshi, M.K., Patt, Y.N. Accelerating critical section execution with asymmetric multi-core architectures. In *ASPLOS* (2009).
22. Venkatesh, G., Sampson, J., Goulding, N., Garcia, S., Bryksin, V., Lugo-Martinez, J., Swanson, S., Taylor, M.B. Conservation cores: Reducing the energy of mature computations. In *ASPLOS* (2010).

Hadi Esmaeilzadeh (hadianeh@cs.washington.edu), University of Washington.

Emily Blem (blem@cs.wisc.edu), University of Wisconsin-Madison.

Renée St. Amant (stamant@cs.utexas.edu), The University of Texas at Austin.

Karthikeyan Sankaralingam (karu@cs.wisc.edu), University of Wisconsin-Madison.

Doug Burger (dburger@microsoft.com), Microsoft Research.