# Power Limitations and Dark Silicon Challenge the Future of Multicore

HADI ESMAEILZADEH, University of Washington
EMILY BLEM, University of Wisconsin-Madison
RENÉE ST. AMANT, The University of Texas at Austin
KARTHIKEYAN SANKARALINGAM, University of Wisconsin-Madison
DOUG BURGER, Microsoft Research

Since 2004, processor designers have increased core counts to exploit Moore's Law scaling, rather than focusing on single-core performance. The failure of Dennard scaling, to which the shift to multicore parts is partially a response, may soon limit multicore scaling just as single-core scaling has been curtailed. This paper models multicore scaling limits by combining device scaling, single-core scaling, and multicore scaling to measure the speedup potential for a set of parallel workloads for the next five technology generations. For device scaling, we use both the ITRS projections and a set of more conservative device scaling parameters. To model single-core scaling, we combine measurements from over 150 processors to derive Pareto-optimal frontiers for area/performance and power/performance. Finally, to model multicore scaling, we build a detailed performance model of upper-bound performance and lower-bound core power. The multicore designs we study include single-threaded CPU-like and massively threaded GPU-like multicore chip organizations with symmetric, asymmetric, dynamic, and composed topologies. The study shows that regardless of chip organization and topology, multicore scaling is power limited to a degree not widely appreciated by the computing community. Even at 22 nm (just one year from now), 21% of a fixed-size chip must be powered off, and at 8 nm, this number grows to more than 50%. Through 2024, only 7.9× average speedup is possible across commonly used parallel workloads for the topologies we study, leaving a nearly 24-fold gap from a target of doubled performance per generation.

Categories and Subject Descriptors: C.0 [**General**]: *Modeling of computer architecture; system architectures*

General Terms: Design, Measurement, Performance

Additional Key Words and Phrases: Dark silicon, modeling, power, technology scaling, multicore

**ACM Reference Format:**
Esmaeilzadeh, H., Blem, E., St. Amant, R., Sankaralingam, K., and Burger, D. 2012. Power limitations and dark silicon challenge the future of multicore. ACM Trans. Comput. Syst. 30, 3, Article 11 (August 2012), 27 pages.
DOI = 10.1145/2324876.2324879 http://doi.acm.org/10.1145/2324876.2324879

This work was supported by the National Science Foundation, under grant CCF-0845751, grant CCF-0917238, and grant CNS-0917213.

Authors' addresses: H. Esmaeilzadeh, Computer Science & Engineering, University of Washington, Box 352350, AC 101, 185 Stevens Way, Seattle, WA 98195; email: hadianeh@cs.washington.edu; E. Blem and K. Sankaralingam, Department of Computer Science, University of Wisconsin-Madison, 1210 West Dayton Street, Madison, WI 53706; R. St. Amant, Department of Computer Science, The University of Texas at Austin, 1616 Guadalupe, Suite 2.408, Austin, TX 78701; D. Burger, Microsoft Research, One Microsoft Way, Redmond, WA 98052.

DOI 10.1145/2324876.2324879 http://doi.acm.org/10.1145/2324876.2324879

## 1. INTRODUCTION

Moore's Law [Moore 1965] (the doubling of transistors on chip every 18 months) has been a fundamental driver of computing. For the past three decades, through device, circuit, microarchitecture, architecture, and compiler advances, Moore's Law, coupled with Dennard scaling [Dennard et al. 1974], has resulted in commensurate exponential performance increases. The recent shift to multicore designs has aimed to increase the number of cores along with transistor count increases, and continue the proportional scaling of performance. As a result, architecture researchers have started focusing on 100-core and 1000-core chips and related research topics and called for changes to the undergraduate curriculum to solve the parallel programming challenge for multicore designs at these scales.

With the failure of Dennard scaling–and thus slowed supply voltage scaling–core count scaling may be in jeopardy, which would leave the community with no clear scaling path to exploit continued transistor count increases. Since future designs will be power limited, higher core counts must provide performance gains despite the worsening energy and speed scaling of transistors, and given the available parallelism in applications. By studying these characteristics together, it is possible to predict for how many additional technology generations multicore scaling will provide a clear benefit. Since the energy efficiency of devices is not scaling along with integration capacity, and since few applications (even from emerging domains such as recognition, mining, and synthesis [Bienia et al. 2008]) have parallelism levels that can efficiently use a 100-core or 1000-core chip, it is critical to understand how good multicore performance will be in the long term. In 2024, will processors have 32 times the performance of processors from 2008, exploiting five generations of core doubling?

Such a study must consider devices, core microarchitectures, chip organizations, and benchmark characteristics, applying area and power limits at each technology node. This article considers all those factors together, projecting upper-bound performance achievable through multicore scaling, and measuring the effects of *non-ideal* device scaling, including the percentage of "dark silicon" (transistor under-utilization) on future multicore chips. Additional projections include best core organization, best chip-level topology, and optimal number of cores.

This article considers technology scaling projections, single-core design scaling, multicore design choices, actual application behavior, and microarchitectural features. Previous studies have also analyzed these features in various combinations, but not together [Chakraborty 2008; Cho and Melhem 2008; Chung et al. 2010; Hempstead et al. 2009; Hill and Marty 2008; Lee et al. 2009; Loh 2008; Venkatesh et al. 2010; Woo and Lee 2008]. This study builds and combines three models to project performance and the fraction of "dark silicon" (transistor integration underutilization) on fixed-size and fixed-power chips as listed here.

- Device model ($DevM$). Area, frequency, and power requirements at future technology nodes through 2024.
- Core model ($CorM$). Power/performance and area/performance single core Pareto frontiers derived from a large set of diverse microprocessor designs.
- Multicore model ($CmpM$). Area, power and performance of any application for "any" chip topology for CPU-like and GPU-like multicore performance.
- $DevM \times CorM$. Pareto frontiers at future technology nodes; any performance improvements for future cores will come only at the cost of area or power as defined by these curves.
- $DevM \times CorM \times CmpM$ and an exhaustive state-space search. Maximum multicore speedups for future technology nodes while enforcing area, power, and benchmark constraints.

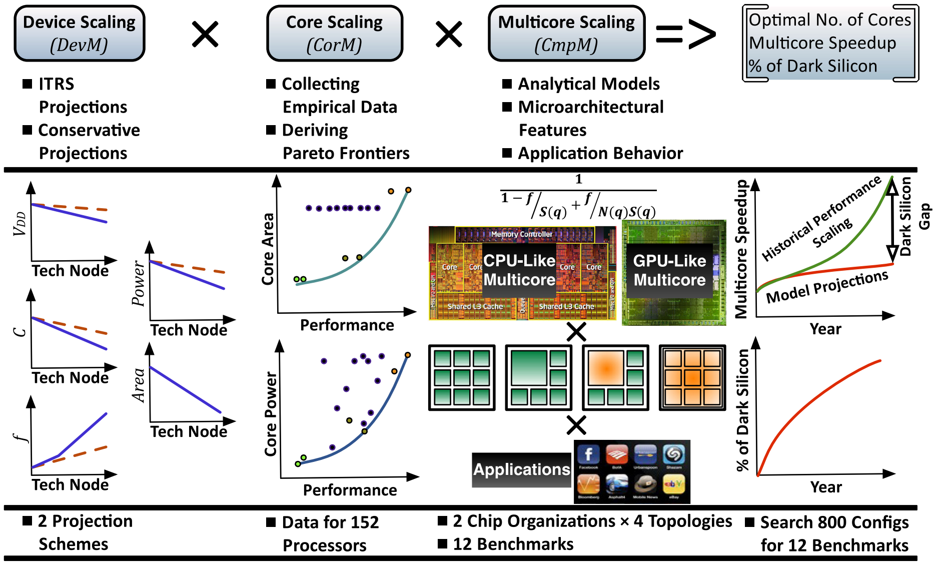


Fig. 1. Overview of the models and the methodology.

The results from this study provide detailed best-case multicore performance speedups for future technologies considering real applications from the PARSEC benchmark suite [Bienia et al. 2008]. Our results evaluating the PARSEC benchmarks and our upper-bound analysis confirm the following intuitive arguments.

(i) Contrary to conventional wisdom on performance improvements from using multicores, over five technology generations, only 7.9× average speedup is possible using ITRS scaling.
(ii) While transistor dimensions continue scaling, power limitations curtail the usable chip fraction. At 22 nm (i.e., in 2012), 21% of the chip will be dark and at 8 nm, over 50% of the chip will not be utilized using ITRS scaling.
(iii) Neither CPU-like nor GPU-like multicore designs are sufficient to achieve the expected performance speedup levels. Radical microarchitectural innovations are necessary to alter the power/performance Pareto frontier to deliver speedups commensurate with Moore's Law.

## 2. OVERVIEW

As depicted in Figure 1, we build and combine three models to project performance and integration capacity utilization as technology scales. Ultimately, the combination of these models predicts multicore performance speedup across future technology generations and shows a gap between the projected speedup and the speedup we have come to expect with each technology scaling. We refer to this speedup gap as the *dark silicon gap*. There following three models are basis for our projections.

*Device Model* (*DevM*). We build device scaling models that provides the area, power, and frequency scaling factors at technology nodes from 45 nm to 8 nm. We consider ITRS Roadmap projections [ITRS 2011] and conservative scaling parameters from Borkar's recent study [Borkar 2010]. We also build two synthetic device scaling models to study transistor scaling trends beyond ITRS and conservative projections.

*Core Model* (*CorM*). Through Pareto-optimal curves [Horowitz et al. 2005], the core-level model provides the maximum performance that a single-core can sustain for any given area. Further, it provides the minimum power (or energy) that must be consumed to sustain this level of performance. To quantify, we measure the core performance in terms of SPECmark. We consider empirical data from a large set of processors and use curve fitting to obtain the Pareto-optimal frontiers for single-core area/performance and power/performance trade-offs.

*Multicore Model* (*CmpM*). We model two mainstream classes of multicore organizations, multi-core CPUs and many-thread GPUs, which represent two extreme points in the threads-per-core spectrum. The CPU multicore organization represents Intel Nehalem-like, heavy-weight multicore designs with fast caches and high single-thread performance. The GPU multicore organization represents NVIDIA Tesla-like lightweight cores with heavy multithreading support and poor single-thread performance. For each multicore organization, we consider four topologies: symmetric, asymmetric, dynamic, and composed (also called “fused” in the literature).

*Symmetric Multicore.* The symmetric, or homogeneous, multicore topology consists of multiple copies of the same core operating at the same voltage and frequency 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. The design leverages 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 [Chakraborty 2008; Suleman et al. 2009].

*Composed Multicore.* The composed multicore topology consists of a collection of small cores that can logically fuse together to compose a high-performance large core for the execution of the serial portion of code [Ipek et al. 2007; Kim et al. 2007]. In either serial or parallel cases, the large core or the small cores are used exclusively.

Table I outlines the design space we explore and explains the roles of the cores during serial and parallel portions of applications. Single-thread (ST) cores are uni-processor style cores with large caches and many-thread (MT) cores are GPU-style cores with smaller caches; both are described in more detail in Section 5.

This article describes an analytic model that provides system-level performance using as input the core's performance (obtained from *CorM*) and the multicore's organization (CPU-like or GPU-like). Unlike previous studies, the model considers application behavior, its memory access pattern, the amount of thread-level parallelism in the workload, and microarchitectural features such as cache size, memory bandwidth, etc. We choose the PARSEC benchmarks because they represent a set of highly parallel applications that are widely studied in the research community.

Heterogeneous configurations such as AMD Fusion and Intel Sandy Bridge combine CPU and GPU designs on a single chip. The asymmetric and dynamic GPU topologies resemble those two designs, and the composed topology models configurations similar to AMD Bulldozer. For GPU-like multicores, this study assumes that the single ST core does not participate in parallel work. Finally, our methodology implicitly models heterogeneous cores of different types (mix of issue widths, frequencies, etc.) integrated

Table I. CPU and GPU Topologies (ST Core: Single-Thread Core and MT: Many-Thread Core)

| | Code | Symmetric | Asymmetric | Dynamic | Composed |
|---|---|---|---|---|---|
| **CPU Multicores** | Serial Fraction | 1 ST Core | 1 Large ST Core | 1 Large ST Core | 1 Large ST Core |
| | Parallel Fraction | *N* ST Cores | 1 Large ST Core + *N* Small ST Cores | *N* Small ST Cores | *N* Small ST Cores |
| **GPU Multicores** | Serial Fraction | 1 MT Core<br>(1 Thread) | 1 Large ST Core<br>(1 Thread) | 1 Large ST Core<br>(1 Thread) | 1 Large ST Core<br>(1 Thread) |
| | Parallel Fraction | *N* MT Cores<br>(Multiple Threads) | 1 Large ST Core + *N* Small MT Cores<br>(1 Thread) (Multiple Threads) | *N* Small MT Cores<br>(Multiple Threads) | *N* Small MT Cores<br>(Multiple Threads) |

on one chip. Since we perform a per-benchmark optimal search for each organization and topology, we implicitly cover the upper-bound of this heterogeneous case.

## 3. DEVICE MODEL

We begin with a brief historical perspective on device scaling and then describe our model and its implementation.

*Historical Perspective on Device Scaling.* Historically, transistor scaling has reliably delivered significant performance improvements each generation by allowing more transistors on chip that run faster without increasing overall chip power consumption. The CMOS transistor scaling has traditionally followed Dennard's scaling scheme [Dennard et al. 1974]. Dennard scaling calls for a 30% reduction in transistor dimensions each generation, keeping the electric fields within the transistor constant. 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 in each technology generation, in accordance with Moore's Law [Moore 1965]. Additionally, the 30% delay reduction that results from decreased transistor length allows for a 40% increase in frequency and, therefore, performance. To keep the electric field constant, supply voltage is reduced by 30%. Oxide thickness is also reduced by 30%, resulting in a 30% decrease in capacitance, and ultimately power is reduced by 50%. Therefore, doubling the number of transistors on chip does not result in an increase in chip power consumption. Dennard scaling stopped at 90nm [Dennard et al. 2007].

*Recent Device Scaling Trends.* As supply voltage scales down, so must the transistor threshold voltage to preserve reliable binary margins at the gate level. However, leakage current increases exponentially with a decrease in threshold voltage. Controlling leakage has limited reductions in threshold voltage, thereby limiting supply voltage scaling. Physical limitations in further scaling gate oxide thickness, as well as limited supply voltage scaling, restricts the device performance increase and device power reduction that can be achieved with transistor scaling. On the other hand, the dimensions of the transistors are scaling with historical rates doubling the device integration capacity every process technology generation. The disparity between the transistor area reduction and its power reduction results in dark silicon, underutilization of the device integration capacity [Chakraborty 2008]. In addition, the transistor switching delay is not decreasing at historical rates, which further decreases the potential to extract speedup from device scaling. These two device scaling trends create a gap between the speedup levels that can be achieved in the future and the speedup levels our industry has grown accustomed to. We refer to this speedup gap as the dark silicon gap. Our projections show that this gap imminently increases as we go further down the path of technology scaling even with tuned optimal multicore designs. Here, we present our device model that captures the device scaling trends for future technology generations.

### 3.1. Model Structure

The device model provides transistor area, power, and frequency scaling factors from a base technology node (e.g., 45 nm) to future technologies. The area scaling factor corresponds to the shrinkage in transistor dimensions. The frequency scaling factor is calculated based on the FO4 inverter delay reduction. 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 build four device models. Two *original* models, conservative and ITRS, and two *synthetic* models derived from the original ones, midpoint and aggressive. The parameters used for calculating the power and performance scaling factors are summarized in Table II.

*Original DevMs.* The conservative model is based on predictions presented by Borkar and represents a less optimistic view [Borkar 2010]. The ITRS model uses projections from the ITRS 2010 technology roadmap [ITRS 2011]. As shown in Table II, the ITRS roadmap predicts that multi-gate MOSFETs, such as FinFETs, will supersede planar bulk at 22 nm [ITRS 2011]. The large increase in frequency, 2.2× as shown in Table II, and substantial reduction in capacitance, 47%, from 32 nm to 22 nm is the result of this technology change. The ITRS roadmap predicts that by changing the transistor technology to multi-gate MOSFETs, the device power decreases by 18%, despite a frequency increase of 2.2×. Based on ITRS projections, the device switching delay increases from 11 nm to 8 nm, while its power decreases.

*Synthetic DevMs.* The midpoint model is the middle ground scaling projection between conservative scaling and the ITRS projection. At each technology node, the frequency scaling and the power scaling factors are computed as the average of conservative and ITRS factors. For the aggressive model, which is one step more optimistic than ITRS, the frequency and power scaling factors are computed such that the ITRS factors are the average of midpoint and aggressive factors. In all the four models, the energy scaling factor is computed based on the frequency and power factors as the (power scaling factor) × 1/(frequency scaling factor). The area scaling factor is the same across all models: a 50% area reduction per process scaling in accordance with Moore's Law [Moore 1965].

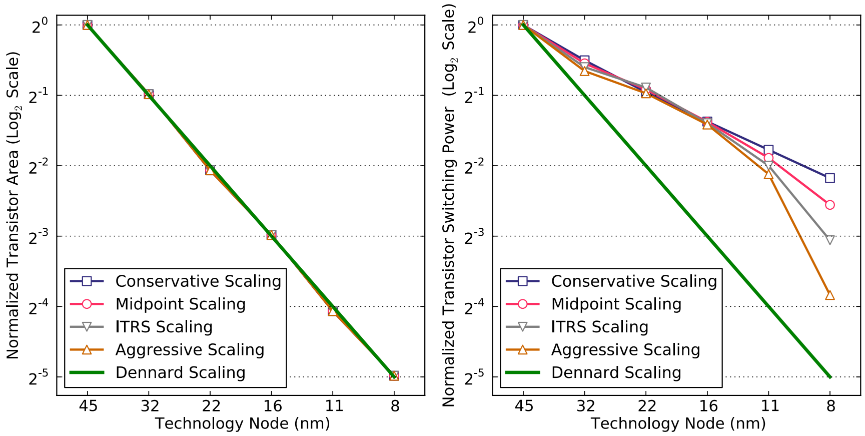
Figure 2 shows the device area and switching power scaling trends for our four *DevM* models across the future technology nodes compared to classical Dennard scaling. As illustrated, even though the device area is scaling according to historical rates, there is a growing gap between device power and the historical rate of transistor power reduction. This growing gap is one of the main sources of dark silicon. At 8 nm, the gap is 2.3× between Dennard scaling and the device switching power projected by the aggressive model. This gap becomes as large as 7.3× with conservative scaling at 8 nm.

Table II. Scaling Factors for Conservative, Midpoint, ITRS and Aggressive Projections

| | Year | Tech Node (nm) | Frequency Scaling Factor (/45 nm) | Vdd Scaling Factor (/45 nm) | Capacitance Scaling Factor (/45 nm) | Power Scaling Factor (/45 nm) | Energy Scaling Factor (/45 nm) | Area Scaling Factor (/45 nm) |
|---|---|---|---|---|---|---|---|---|
| **Conservative** | 2008 | 45 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | $2^{0}$ |
| | 2010 | 32 | 1.10 | 0.93 | 0.75 | 0.71 | 0.65 | $2^{-1}$ |
| | 2012 | 22 | 1.19 | 0.88 | 0.56 | 0.52 | 0.44 | $2^{-2}$ |
| | 2014 | 16 | 1.25 | 0.86 | 0.42 | 0.39 | 0.31 | $2^{-3}$ |
| | 2016 | 11 | 1.30 | 0.84 | 0.32 | 0.29 | 0.22 | $2^{-4}$ |
| | 2018 | 8 | 1.34 | 0.84 | 0.24 | 0.22 | 0.16 | $2^{-5}$ |
| 6% frequency increase, 23% power reduction, and 30% energy reduction per node | | | | | | | | |
| **Midpoint** | – | 45 | 1.00 | – | – | 1.00 | 1.00 | $2^{0}$ |
| | – | 32 | 1.10 | – | – | 0.69 | 0.63 | $2^{-1}$ |
| | – | 22 | 1.79 | – | – | 0.53 | 0.30 | $2^{-2}$ |
| | – | 16 | 2.23 | – | – | 0.39 | 0.17 | $2^{-3}$ |
| | – | 11 | 2.74 | – | – | 0.27 | 0.10 | $2^{-4}$ |
| | – | 8 | 2.60 | – | – | 0.17 | 0.07 | $2^{-5}$ |
| 21% frequency increase, 30% power reduction, and 42% energy reduction per node | | | | | | | | |
| **ITRS** | 2010 | 45* | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | $2^{0}$ |
| | 2012 | 32* | 1.09 | 0.93 | 0.70 | 0.66 | 0.61 | $2^{-1}$ |
| | 2015 | 22† | 2.38 | 0.84 | 0.33 | 0.54 | 0.23 | $2^{-2}$ |
| | 2018 | 16† | 3.21 | 0.75 | 0.21 | 0.38 | 0.12 | $2^{-3}$ |
| | 2021 | 11† | 4.17 | 0.68 | 0.13 | 0.25 | 0.06 | $2^{-4}$ |
| | 2024 | 8† | 3.85 | 0.62 | 0.08 | 0.12 | 0.03 | $2^{-5}$ |
| 31% frequency increase, 35% power reduction, and 50% energy reduction per node | | | | | | | | |
| **Aggressive** | – | 45 | 1.00 | – | – | 1.00 | 1.00 | $2^{0}$ |
| | – | 32 | 1.11 | – | – | 0.64 | 0.57 | $2^{-1}$ |
| | – | 22 | 2.98 | – | – | 0.51 | 0.17 | $2^{-2}$ |
| | – | 16 | 4.19 | – | – | 0.38 | 0.09 | $2^{-3}$ |
| | – | 11 | 5.61 | – | – | 0.23 | 0.04 | $2^{-4}$ |
| | – | 8 | 5.11 | – | – | 0.07 | 0.01 | $2^{-5}$ |
| 39% frequency increase and 41% power reduction, and 58% energy reduction per node | | | | | | | | |

∗: Extended Planar Bulk Transistors, †:Multi-Gate Transistors

*Leakage.* We allocate 20% of the chip power budget to leakage power. As shown in Nose and Sakurai [2000], the transistor threshold voltage can be selected so that the maximum leakage power is always an acceptable ratio of the chip power budget while



(a) Device area (b) Device switching power

Fig. 2. Device scaling trends across future technology nodes with four different scaling projections compared to classical Dennard scaling.

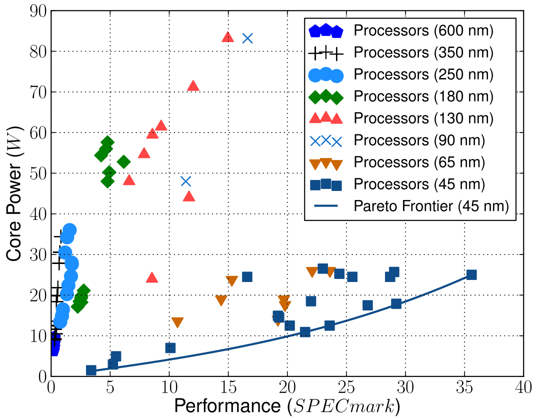


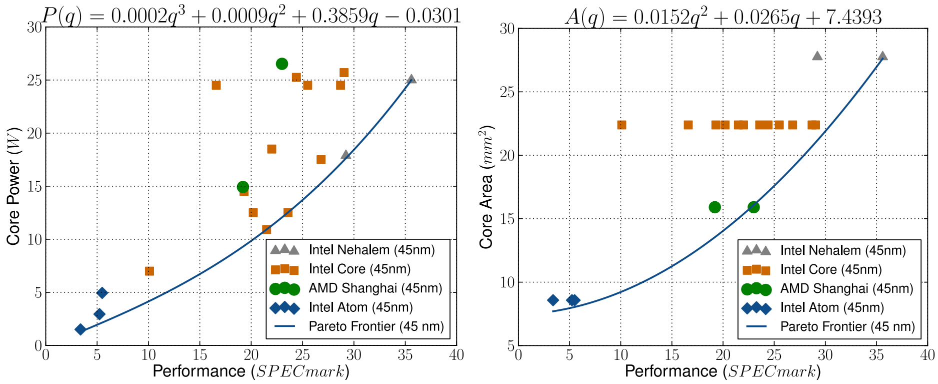
Fig. 3. Power/performance design space for 152 real processors (from P54C Pentium to Nehalem-based i7) fabricated at 600 nm through 45 nm. The design space boundary that comprises the power/performance optimal points constructs the Pareto frontier.

still meeting power and performance constraints. We also observe that with 10% or 30% leakage power, we do not see significant changes in optimal configurations.

## 4. CORE MODEL

### 4.1. Model Structure

We build the technology-scalable core model by populating the area/performance and power/performance design spaces with the data collected for a set of processors as depicted in Figure 3. 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. The $q$ is the single-threaded performance of a core. These frontiers capture the optimal area/performance and power/performance trade-offs 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.



(a) Power/performance frontier, 45 nm

(b) Area/performance frontier, 45 nm

Fig. 4. Single-core (a) power/performance and (b) area/performance design space at 45 nm and the corresponding Pareto frontiers.

### 4.2. Model Implementation

As Figure 4 depicts, we populate the two design spaces at 45 nm using 20 representative Intel and AMD processors[1] 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 website [SPEC 2011]. 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 used 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. To compute the power budget of a single core, the power budget allocated to the level 2 and level 3 caches is estimated and deducted from the chip TDP. In the case of a multicore CPU, the remainder of the chip power budget is divided by the number of cores, resulting in the power budget allocated to a single core.

As illustrated in Figure 4, 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 [Pollack 1999]), $A(q)$, to the points along the edge of the area/performance design space. We used the least square regression method for curve fitting such that the frontiers enclose all design points. Figures 4(a) and 4(b) show the 45-nm processor points and identify the power/performance and area/performance Pareto frontiers. The power/performance cubic polynomial $P(q)$ function (Figure 4(a)) and the area/performance quadratic polynomial $A(q)$ (Figure 4(b)) are the *core model*. The Intel Atom Z520 with an estimated 1.89 W core TDP represents the lowest power

[1] Atom Z520, Atom 230, Atom D510, Core 2 Duo T9500, Core 2 Extreme QX9650, Core 2 Quad Q8400, Opteron 2393SE, Opteron 2381HE, Core 2 Duo E7600, Core 2 Duo E8600, Core 2 Quad Q9650, Core 2 Quad QX9770, Core 2 Duo T9900, Pentium SU2700, Xeon E5405, Xeon E5205, Xeon X3440, Xeon E7450, Core i7-965 ExEd.

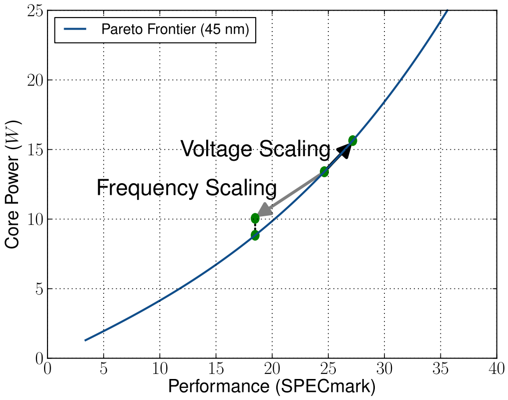


Fig. 5. Voltage and frequency scaling on the power/performance Pareto frontiers.

design (lower-left frontier point), and the Nehalem-based Intel Core i7-965 Extreme Edition with an estimated 31.25 W core TDP 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-scaling model.

*Decoupling Power/Performance Trade-Off from Area/Performance Trade-Off.* Previous studies on multicore performance modeling [Chakraborty 2008; Cho and Melhem 2008; Chung et al. 2010; Hill and Marty 2008; Lee et al. 2009; Loh 2008; Woo and Lee 2008] use Pollack's rule [Pollack 1999] to denote the trade-off between area and performance. Furthermore, these studies consider the power consumption of a core to be directly proportional to its area. This assumption makes power an area-dependent constraint. However, power is a function of not only area, but also supply voltage and frequency. Since these no longer scale at historical rates, Pollack's rule is insufficient for modeling core power. Thus, it is necessary to decouple area and power into two independent constraints.

*Voltage and Frequency Scaling.* Our device and core models do not explicitly consider dynamic voltage and frequency scaling; instead, we take an optimistic approach to account for its best-case impact. When deriving the Pareto frontiers, each processor data point was assumed to operate at its optimal voltage and frequency setting ($Vdd_{min}$, $Freq_{max}$). Figure 5 shows the result of voltage/frequency scaling on the design points along the power/performance frontier. At a fixed $Vdd$ 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 voltage up and operating at a new ($Vdd'_{min}$, $Freq'_{max}$) setting results in a different power-performance point that is still on the optimal frontier. Furthermore, if an application dissipates less than the power budget, we assume that the voltage and frequency scaling will be utilized to achieve the highest possible performance with the minimum power increase. This is possible since voltage and frequency scaling only changes the operating condition in a Pareto-optimal fashion. Since we investigate all of the points along the frontier to find the optimal multicore configuration, our study covers multicore designs that introduce heterogeneity to symmetric topologies through dynamic voltage and frequency scaling.

Table III. *CmpM* Parameters with Default Values from 45 nm Nehalem

| Parameter | Description | Default | Impacted By |
|---|---|---|---|
| $N$ | Number of cores | 4 | Multicore Topology |
| $T$ | Number of threads per core | 1 | Core Style |
| $freq$ | Core frequency (MHz) | 3200 | Core Performance |
| $CPI_{exe}$ | Cycles per instruction (zero-latency cache accesses) | 1 | Core Performance, Application |
| $C_{L1}$ | L1 cache size per core (KB) | 64 | Core Style |
| $C_{L2}$ | L2 cache size per chip (MB) | 2 | Core Style, Multicore Topology |
| $t_{L1}$ | L1 access time (cycles) | 3 | - |
| $t_{L2}$ | L2 access time (cycles) | 20 | - |
| $t_{mem}$ | Memory access time (cycles) | 426 | Core Performance |
| $BW_{max}$ | 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 |
| $r_m$ | Fraction of instructions that are memory accesses | varies | Application |
| $\alpha_{L1}$, $\beta_{L1}$ | L1 cache miss rate function constants | varies | Application |
| $\alpha_{L2}$, $\beta_{L2}$ | L2 cache miss rate function constants | varies | Application |

## 5. MULTICORE MODEL

### 5.1. Model Structure

We develop a detailed chip-level model (*CmpM*) 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). Guz et al. [2009] proposed a model to consider first-order impacts of microarchitectural features (cache organization, memory bandwidth, number of threads per core, etc.) and workload behavior (memory access pattern). 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 behaviors, microarchitectural features, and physical constraints, and covers both homogeneous and heterogeneous multicore topologies.

Using this model, we consider 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 space across all four topologies, as described in Table I. The input parameters to the model, and how, if at all, they are impacted by the multicore design choices are listed in Table III.

*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 (1) by multiplying the number of cores ($N$) by the core utilization ($\eta$) and scaling by the ratio of the processor frequency to $CPI_{exe}$:

$$Perf = \min\left(N\frac{freq}{CPI_{exe}}\eta, \frac{BW_{max}}{r_m \times m_{L1} \times m_{L2} \times b}\right). \tag{1}$$

The $CPI_{exe}$ parameter does not include stalls due to cache accesses, which are considered separately in the core utilization ($\eta$). 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 $CPI_{exe}$:

$$\eta = \min\left(1, \frac{T}{1 + t\frac{r_m}{CPI_{exe}}}\right). \tag{2}$$

Table IV. Number of Cores for Different Multicore Topologies

| Topology | Number of Cores |
|---|---|
| Symmetric | $N_{Symm}(q) = min\left(\frac{DIE_{AREA}}{A(q)}, \frac{TDP}{P(q)}\right)$ |
| Asymmetric | $N_{Asym}(q_L, q_S) = min\left(\frac{DIE_{AREA} - A(q_L)}{A(q_S)}, \frac{TDP - P(q_L)}{P(q_S)}\right)$ |
| Dynamic | $N_{Dynm}(q_L, q_S) = min\left(\frac{DIE_{AREA} - A(q_L)}{A(q_S)}, \frac{TDP}{P(q_S)}\right)$ |
| Composed | $N_{Comp}(q_L, q_S) = min\left(\frac{DIE_{AREA}}{(1 + \tau)A(q_S)}, \frac{TDP}{P(q_S)}\right)$ |

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 $m_{L2}$):

$$t = (1 - m_{L1})t_{L1} + m_{L1}(1 - m_{L2})t_{L2} + m_{L1}m_{L2}t_{mem} \tag{3}$$

$$m_{L1} = \left(\frac{C_{L1}}{T\beta_{L1}}\right)^{1-\alpha_{L1}} \text{ and } m_{L2} = \left(\frac{C_{L2}}{NT\beta_{L2}}\right)^{1-\alpha_{L2}}. \tag{4}$$

*Multicore Topologies.* The multicore model (5) is an extended Amdahl's Law [Amdahl 1967] that incorporates the multicore performance ($Perf$) calculated from (1)–(4):

$$Speedup = 1/\left(\frac{f}{S_{Parallel}} + \frac{1 - f}{S_{Serial}}\right). \tag{5}$$

The $CmpM$ model (5) 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 $S_{Serial} = Perf_S/Perf_B$. The number of cores that fit on the chip is calculated through the equations in Table IV based on the multicore topology, its area budget ($DIE_{AREA}$), its power budget ($TDP$), area of each core ($A(q)$), and power consumption of each core ($P(q)$). 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 $\tau$, while no power overhead is assumed for composability support.

### 5.2. Model Implementation

One of the contributions of this work is the incorporation of Pareto frontiers, physical constraints, real application behavior, and realistic microarchitectural features into the multicore speedup projections as discussed below.

*Application Behavior.* 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 [Bhadauria et al. 2009; Bienia et al. 2008]. 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. This fit shows values of $f$ between 0.75 and 0.9999 for individual benchmarks.

*Obtaining Frequency and $CPI_{exe}$ from Pareto Frontiers.* To incorporate the Pareto-optimal curves into the *CmpM* 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 optimizations (e.g., functional units and out-of-order processing). Since the measured $CPI_{exe}$ for each benchmark at each technology node is not available, we use the *CmpM* 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 $\ell$. Then, the smallest core, an Atom processor, should have a $CPI_{exe}$ such that the ratio of its *CmpM* performance to the Nehalem core's *CmpM* performance is the same as the ratio of their SPECmark scores ($q$). We assume $CPI_{exe}$ does not change with technology node, while frequency scales as discussed in Section 4.

*Microarchitectural Features.* A key component 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 64 KB L1 cache, and chips with only ST cores have an L2 cache that is 30% of the chip area. Many-thread (MT) cores[2] have small L1 caches (32 KB for every 8 cores), support multiple hardware contexts (1024 threads per 8 cores), a thread register file, and 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 $\tau$ increases from 10% up 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 45 nm.

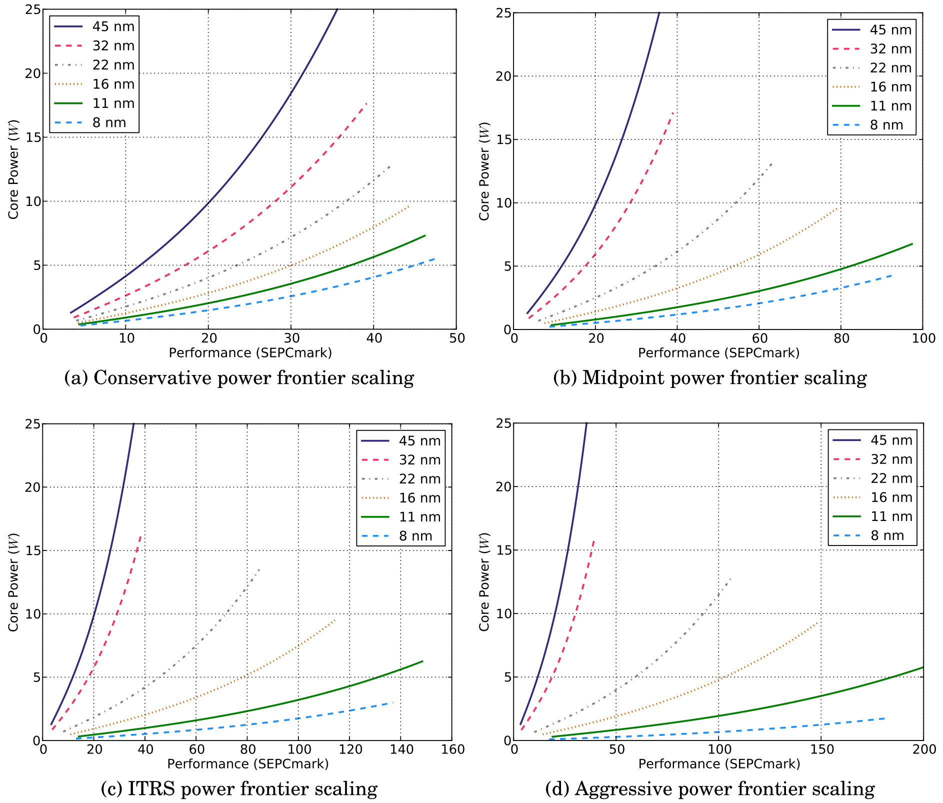
*Baselines and Constraints.* The *CmpM* multicore speedup baseline is a quad-Nehalem multicore that fits in the area and power budgets. The area and power budgets are derived from the same quad-core Nehalem multicore at 45 nm excluding the L2 and L3 caches. They are 111 mm$^2$ and 125 W, respectively. The reported dark silicon projections are for the area budget that is solely allocated to the cores, not caches and other "uncore" components.

## 6. COMBINING MODELS

### 6.1. Device × Core Model

To study core scaling in future technology nodes, we scale the 45-nm Pareto frontiers to 8 nm by scaling the power and performance of each processor data point using the projected *DevM* scaling factors and then re-fitting the Pareto optimal curves at each technology node. Performance, measured in SPECmark, is assumed to scale linearly with frequency. This is an optimistic assumption which ignores the effects of memory latency and bandwidth on the performance. Thus, actual performance through scaling is likely to be lower. Figure 6 shows the scaled power Pareto frontiers with the conservative, midpoint, ITRS, and aggressive device scaling models. As illustrated in Figure 6(a), conservative scaling suggests that performance will increase only by 34%, and power will decrease by 74% as a core scales from 45 nm to 8 nm. Figure 6(b)

[2] In the GPU projections, each core is a shader processor (SP), a group of which can form a stream multiprocessor (SM).



(a) Conservative power frontier scaling

(b) Midpoint power frontier scaling

(c) ITRS power frontier scaling

(d) Aggressive power frontier scaling

Fig. 6. Combining *DevM* and *CorM* results in scaled core models for future technology nodes by providing the scaled power/performance and area/performance Pareto frontiers.

shows that the core performance from 45 nm through 8 nm increases 2.6× and the core power decreases by 83% with midpoint scaling. Based on the optimistic ITRS predictions, however, scaling a microarchitecture (core) from 45 nm to 8 nm will result in a 3.9× performance improvement and an 88% reduction in its power consumption (Figure 6(c)). As shown in Figures 6(d), with aggressive scaling the single-threaded performance at 8 nm increases by a factor of 5.1× while its power dissipation decreases by 93%. The current trends of frequency scaling in microprocessor design is far from the predictions of ITRS. We believe that based on the current trends, without any disruptive innovations in transistor design, such frequency improvements may not be possible.

### 6.2. Device × Core × Multicore Model

All three models are combined to produce final projections for 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 (*DevM* × *CorM*) as these points represent the most efficient designs. First, we uniformly quantize the scaled Pareto frontier to 100 points. At each technology node, for each core design on the scaled frontiers, we construct a multicore consisting of one such core. For a symmetric multicore, we iteratively add identical cores one by one until the area or power budget is hit, or performance improvement is limited (less than 10% performance gain

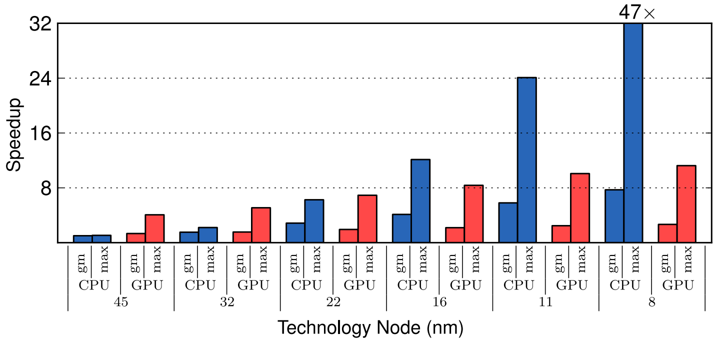


Fig. 7. Speedup projections for CPU-like and GPU-like symmetric multicore topology across technology generations with ITRS scaling.

for doubling the number cores). 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 designated for 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 detailed analysis of multicore behavior for future technologies considering 12 real applications from the PARSEC suite. Details for all applications and topologies are presented in Figure 17. Unless otherwise stated, the model is used to find the optimal multicore configuration with the highest possible speedup for each individual benchmark.

### 7.1. Speedup Projections

Figure 7 shows the geometric mean of speedup and the best-case speedup among the benchmarks for a symmetric topology using the optimistic ITRS scaling. The symmetric topology achieves the lower bound on speedups. With speedups that are no more than 10% higher, the dynamic and composed topologies achieve the upper-bound. The results are presented for both CPU-like and GPU-like multicore organizations. To conduct a fair comparison between different design points, all speedup results are normalized to the performance of a quad-core Nehalem multicore at 45 nm that fits in the same power and area budget. The results over five technology generations with the four device scaling projections are summarized as follows.

| | | **Conservative** | | **Midpoint** | | **ITRS** | | **Aggressive** | |
|---|---|---|---|---|---|---|---|---|---|
| | Topology | CPU | GPU | CPU | GPU | CPU | GPU | CPU | GPU |
| Geomean Speedup | Symmetric | 3.4× | 2.4× | 5.4× | 2.4× | 7.7× | 2.7× | 10.2× | 2.7× |
| Geomean Speedup | Asymmetric | 3.5× | 2.4× | 5.5× | 2.4× | 7.9× | 2.7× | 10.3× | 2.7× |
| Geomean Speedup | Dynamic | 3.5× | 2.4× | 5.5× | 2.4× | 7.9× | 2.7× | 10.3× | 2.7× |
| Geomean Speedup | Composed | 3.7× | 2.3× | 5.1× | 2.3× | 6.2× | 2.5× | 7.2× | 2.5× |
| Maximum Speedup | All | 10.9× | 10.1× | 27.5× | 10.1× | 46.6× | 11.2× | 91.7× | 11.2× |

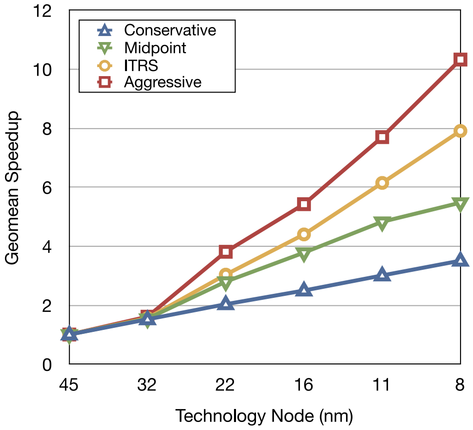


Fig. 8. Speedup projections for dynamic CPU-like multicore topologies with four transistor scaling models.

Figure 8 shows the geometric mean speedup projections across the technology nodes for all the device scaling models. As depicted, improvements in process technology have a direct impact in bridging the dark silicon speedup gap. We believe that reality will be closer to the midpoint projections that leaves a large dark silicon speedup gap. However, a disruptive breakthrough in transistor fabrication that matches the aggressive scaling predictions could improve potential multicore scaling significantly.

Figure 9 summarizes all of the speedup projections in a single scatter plot for conservative and ITRS scaling models. 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 in Figure 9 shows the exponential performance scaling—doubling performance every generation.

With optimal multicore configurations for each individual application, at 8 nm, only 3.7× (conservative scaling), 5.5× (midpoint scaling), 7.9× (ITRS scaling), or 10.3× (aggressive scaling) geometric mean speedup is possible.

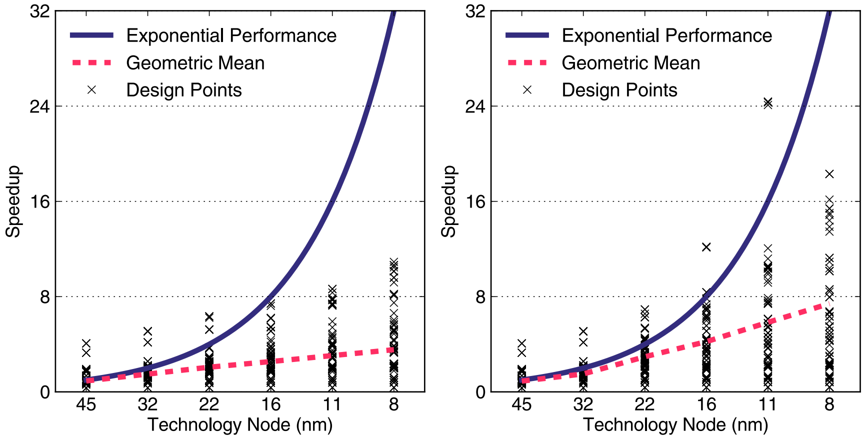
Highly parallel workloads with a degree of parallelism higher than 99% will continue to benefit from multicore scaling.

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

Improvements in transistor process technology are directly reflected as multicore speedup; however, to bridge the dark silicon speedup gap even a disruptive breakthrough that matches our aggressive scaling model is not enough.

### 7.2. Core Count Projections

Figure 10 illustrates the geometric mean number of cores across the 12 PARSEC benchmarks for a symmetric topology with ITRS scaling when each individual benchmark has its optimum number of cores. Different applications saturate performance improvements at different core counts, but the geometric mean number of cores is less than 64. We consider as an *ideal* configuration the chip configuration that provides



(a) Conservative Scaling

(b) ITRS Scaling

Fig. 9. Speedup across process technology nodes across all organizations and topologies with PARSEC benchmarks.

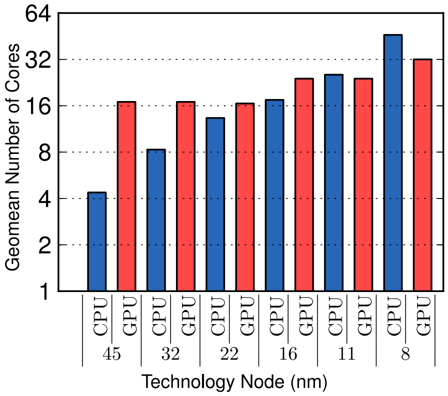
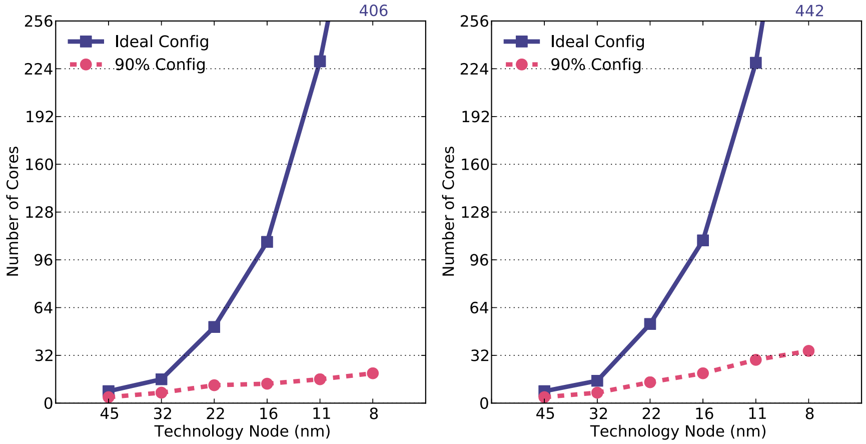


Fig. 10. Geometric mean number of cores across 12 PARSEC benchmarks for symmetric topology with ITRS scaling.

the best speedups for all applications. Figure 11 shows the number of cores (solid line) for the ideal CPU-like dynamic multicore configuration across technology generations, since dynamic configurations performed best. 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 8 nm; however, 35 cores reach the 90% of the speedup achievable by 442 cores. With conservative scaling, the 90% speedup core count is 20 at 8 nm.

For the PARSEC benchmarks that we studied, the typical number of cores for individual benchmarks is less than 64 for CPUs and less than 256 SP cores for GPUs with both conservative and ITRS scaling.

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



(a) Conservative Scaling (b) ITRS Scaling

Fig. 11. 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.

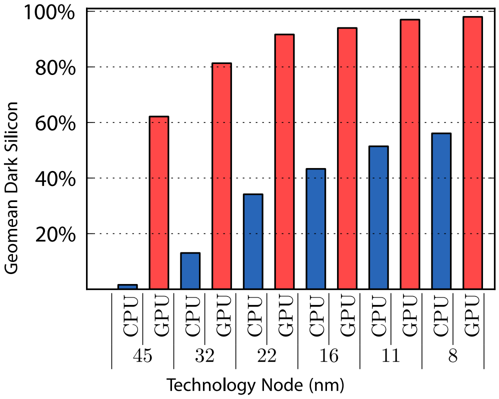


Fig. 12. Percentage of dark silicon (geometric mean across all 12 PARSEC benchmarks) for symmetric topology and ITRS scaling.

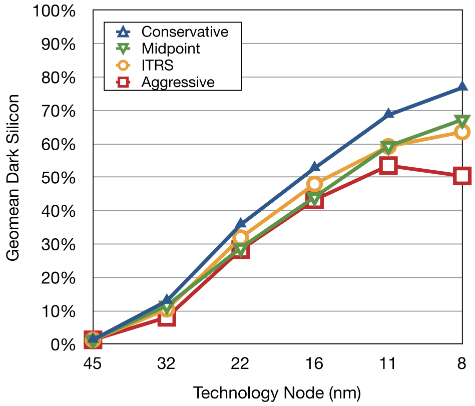


Fig. 13. Dark silicon projections across technology generation for dynamic CPU-like multicores with the four device scaling models.

The optimal number of cores projected by our study seems small compared to chips such as the NVIDIA Fermi, which has 512 cores at 45 nm. There are two reasons for this discrepancy. First, in our study we are optimizing for a fixed power budget, whereas with real GPUs the power has been slightly increasing. Second, our study optimizes core count and multicore configuration for general purpose workloads similar to the PARSEC suite. We assume Fermi is optimized for graphics rendering. When we applied our methodology to a graphics kernel (ray tracing) in an asymmetric topology, we obtained higher speedups and an optimal core count of 4864 at 8 nm, with 8% dark silicon.

### 7.3. Dark Silicon Projections

Figure 13 illustrates the dark silicon projections for dynamic CPU multicore topology with the four transistor scaling models. As depicted, the midpoint scaling almost matches the ITRS projections and aggressive scaling lowers the portion of dark silicon. However, as shown in Figure 2, even with aggressive scaling, the energy efficiency of transistors is significantly below the ideal Dennard scaling which enables Moore's Law. This significant gap between the ideal transistor scaling prevents even huge improvements in process technology from bridging the dark silicon underutilization gap. Microarchitectural innovations that can efficiently trade area for energy are vital to tackle the dark silicon problem.

Figure 12 depicts the geometric mean percentage of dark silicon across the PARSEC benchmarks for symmetric multicores with ITRS scaling. In these projections, we optimistically use the core counts that achieve the highest speedup for individual benchmarks. The trend is similar for other topologies.

With conservative scaling, dark silicon dominates in 2016 for CPU-like and in 2012 for GPU-like multicores. With ITRS scaling, dark silicon dominates in 2021 for CPU-like multicores and in 2015 for GPU-like multicores.

With ITRS projections, at 22 nm (2012) 21% of the chip will be dark and at 8 nm, over 50% of the chip cannot be utilized.

Mere improvements in process technology even as significant as aggressive scaling projections cannot bridge the dark silicon underutilization gap. Microarchitecture innovations are vital to justify continuous scaling.

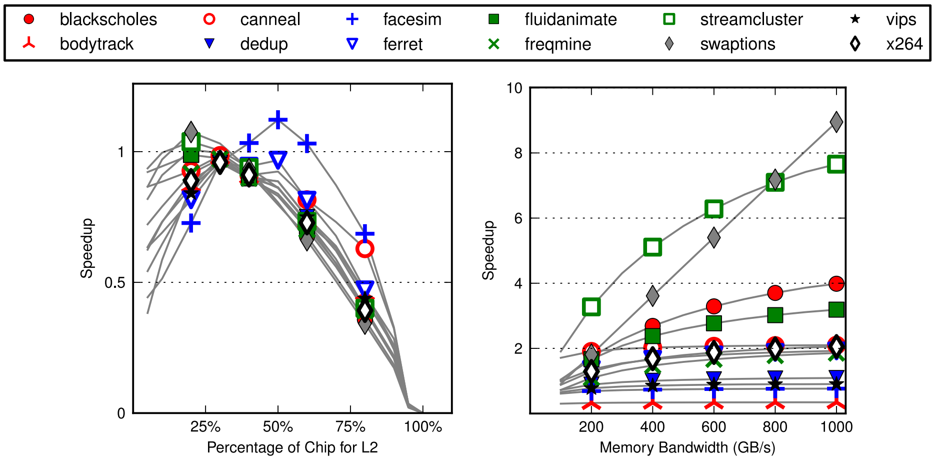
### 7.4. Bridging the Dark Silicon Gap

Our analysis thus far examined "typical" configurations and showed poor scalability for the multicore approach. A natural question is, can simple configuration changes (percentage cache area, memory bandwidth, etc.) provide significant benefits? We elaborate on three representative studies of simple changes (L2 cache size, memory bandwidth, and SMT) below. Further, to understand whether parallelism or the power budget is the primary source of the dark silicon speedup gap, we vary each of these factors in two experiments at 8 nm. Our model is flexible enough to perform these types of studies.

*L2 Cache Area.* Figure 14(a) shows the optimal speedup at 45 nm 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 a 30% cache area, using optimal cache area only improves performance by at most 20% across all benchmarks.

*Memory Bandwidth.* Figure 14(b) illustrates the sensitivity of PARSEC performance to the available memory bandwidth for symmetric GPU multicores at 45 nm. As the memory bandwidth increases, the speedup improves as the bandwidth can keep more threads fed with data; however, the increases 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, 200 GB/s.

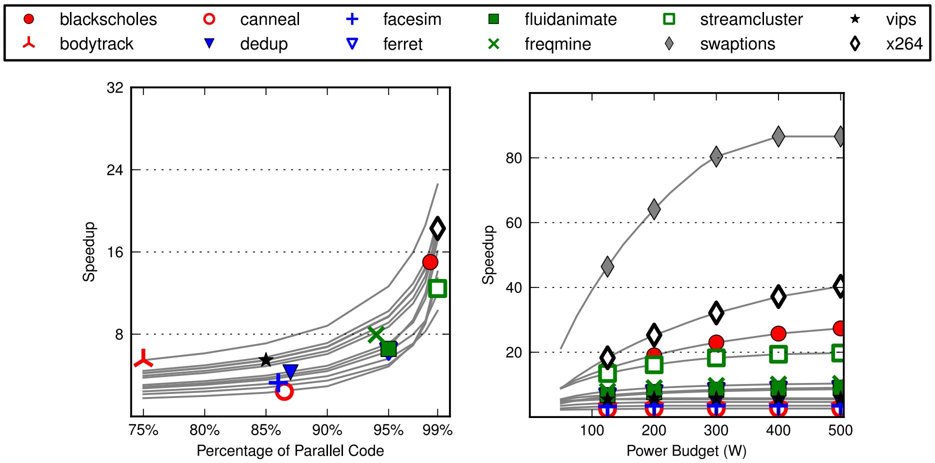
*SMT.* To Simplify the discussion, we did not consider SMT support for the processors (cores) in the CPU multicore organization. SMT support can improve the power



(a) L2 size (CPU)

(b) Memory bandwidth (GPU)

Fig. 14. Impact of L2 size and memory bandwidth on speedup at 45 nm.



(a) Parallelism ($f$ actual at marker)

(b) Power budget

Fig. 15. Impact of application parallelism and power budget on speedup at 8 nm.

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×.

*Application Parallelism.* First, 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 (Figure 15). 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×.

Table V. Effect of Assumptions on *CmpM* Accuracy

| | Assumption | Impact on CPU Speed | Impact on GPU Speed |
|---|---|---|---|
| $\mu$arch | Memory Contention: 0 | ↑ | ⇑ |
| | Interconnection Network Latency: 0 | ⇑ | ⇑ |
| | Thread Swap Time: 0 | ↑ | ↑ |
| Application | Cache Hit Rate Function | ↑ or ↓ | ↑ or ↓ |
| | Thread Synch & Communication: 0 | ↑ | ⇑ |
| | Thread Data Sharing: 0 | ↓ | — |
| | Workload Type: Homogeneous | ↑ | ↑ |

Assumptions lead to ↑ (slightly higher), ⇑ (higher) or ↓ (slightly lower) predicted speedups (or have no effect (—)).

*Power-Budget/Lower-Power Cores.* 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. 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 Amdahl's Law. Only four benchmarks have sufficient parallelism to even hypothetically sustain Moore's Law level speedups.

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

## 8. MODEL ASSUMPTIONS, VALIDATION, AND LIMITATIONS

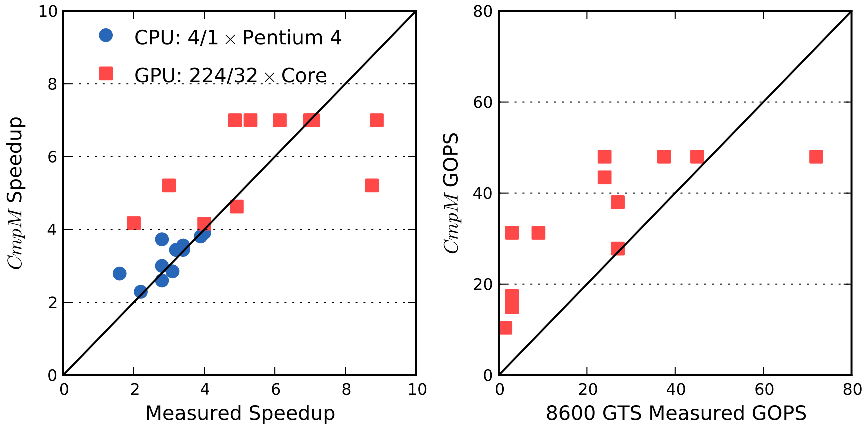
We elaborate on the assumptions of the model and through validation against empirical results demonstrate that they are carefully considered and consistently optimistic with respect to the multicore speedup projections. In addition, our modeling includes certain limitations, which we argue they do not significantly change the results.

### 8.1. Model Assumptions

The *CmpM* model allows us to estimate the first-order impact of caching, parallelism, and threading under several key assumptions. Table V qualitatively describes the impact of these assumptions. The model optimistically assumes that the workload is homogeneous, work is infinitely parallel during parallel sections of code, and no thread synchronization, operating system serialization, or swapping occurs. We also assume memory accesses never stall due to a previous access. Each of these assumptions results in over-prediction of multicore performance, making the model and hence projected speedups optimistic. Cache behaviors may lead to over- or under-prediction. The model assumes that each thread only sees its own slice of cache and thus the model may over or underestimate the hit rate. However, comparing the model's output to the published empirical results confirms that it only over-predicts multicore performance.

### 8.2. Model Validation

To validate the *CmpM* model, we compare the speedup projections from the model to measurement and simulation results for existing CPU and GPU multicores. For the CPU case, we compare the model's speedup predictions to speedup measurements for



(a) Speedup (b) Performance

Fig. 16. *CmpM* model validation.

a quad-Pentium 4 multicore [Bhadauria et al. 2009]. The model is configured to match this real multicore. We validate GPU speedup projections by comparing the model's output simulation results from GPGPUSim [Bakhoda et al. 2009]. Both model and simulator compare speedups of a 224-core GPU over a 32-core GPU. We use GPGPUSim's 12 CUDA benchmarks since GPU implementations of PARSEC are not available. Figure 16(a), which includes both CPU and GPU data, shows that the model is optimistic and over predicts the speedups. *CmpM* underpredicts speedups for two benchmarks for which the simulation results show a speedup of greater than 7× , the increase in number of cores.

To strongly advance our GPU claim, we also need to prove the model's raw performance projection is accurate or optimistic. As depicted in Figure 16(b), the model's GPU performance projection is validated by comparing its output to the results from a real system, NVIDIA 8600 GTS, using the data from Bakhoda et al. [2009]. Except for a known anomaly that also occurs in GPGPUsim, *CmpM* consistently over-predicts raw performance.

Furthermore, using our model, we find 4× geometric-mean and 12× maximum speedup for PARSEC benchmarks on Tesla compared to a quad-core Nehalem. While our results are impressively close to Intel's empirical measurements using similar benchmarks [Lee et al. 2010], the match in the model's maximum speedup prediction (12× vs 11× in the Intel study) is an anomaly. Our model does not account for specialized compute units, which contribute to the speedup in Lee et al. [2010].

### 8.3. Model Limitations

*Different Workloads.* Workloads with significantly different behavior than the workloads we studied could result in different findings.

*Other Types of Cores.* We do not consider embedded ARM or Tilera cores in this work because they are designed for different application domains and their SPECmark scores are not available for a meaningful comparison.

*Power Impact of Uncore.* We ignore the power impact of uncore components such as the memory subsystem. There is consensus that the number of these components will increase and hence they will further eat into the power budget, reducing speedups.

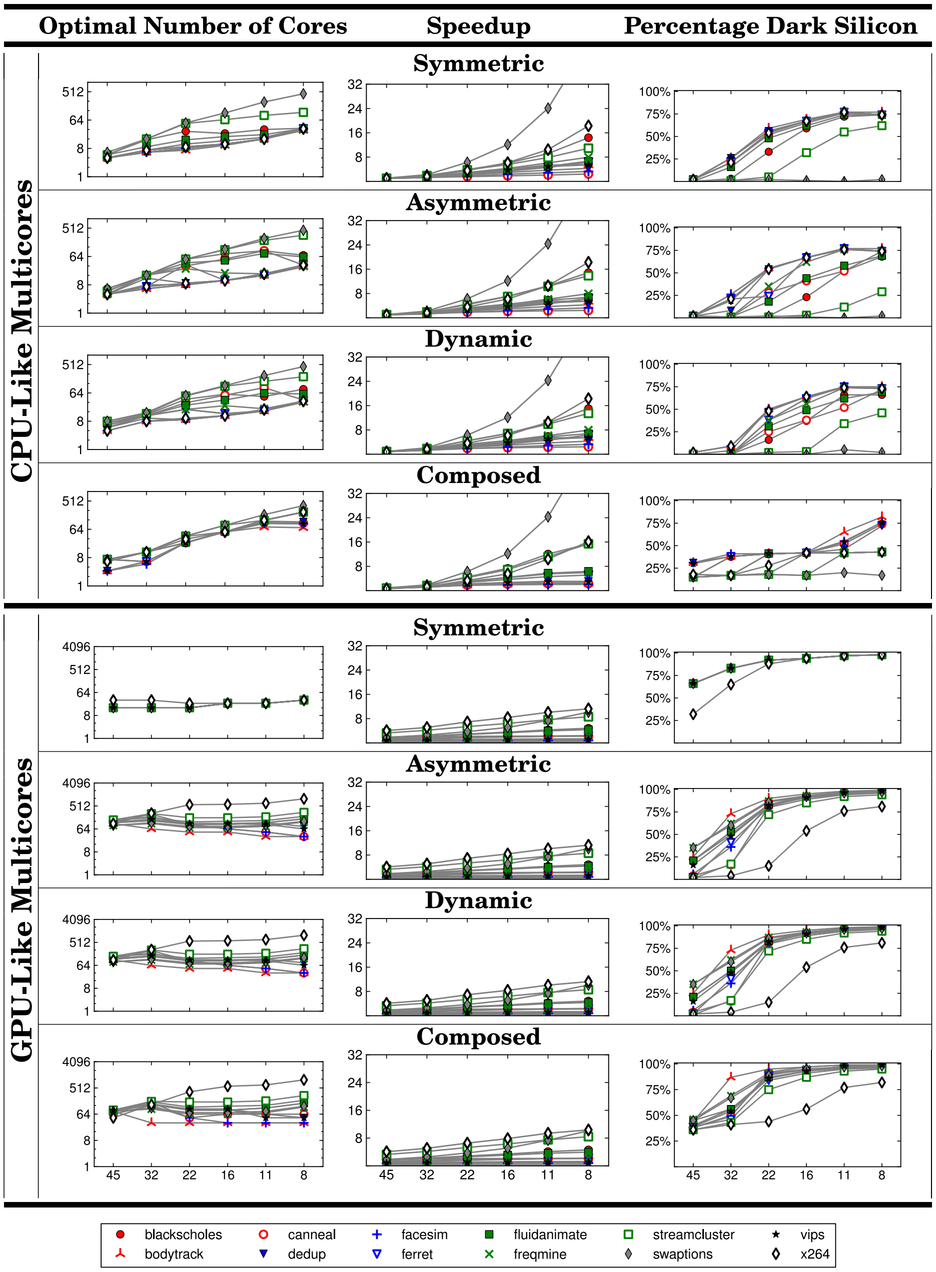


Fig. 17. Optimal number of cores, speedup over quad-Nehalem at 45 nm, and percentage dark silicon under ITRS scaling projections.

*GPU Methodology*. Our GPU methodology may over-estimate the GPU power budget, so we investigated the impact of 10%–50% improved energy efficiency for GPUs and found that total chip speedup and percentage of dark silicon were not impacted.

We acknowledge that we make a number of assumptions in this work to build a useful model. Questions may still linger on the model's accuracy and whether its assumptions contribute to the performance projections that fall well below the ideal 32×. First, in all instances, we selected parameter values that would be favorable towards multicore performance. Second, our validation against real and simulated systems shows the model always over-predicts multicore performance.

## 9. RELATED WORK

Hill and Marty [2008] extend Amdahl's Law to model multicore speedup with symmetric, asymmetric, and dynamic topologies and conclude dynamic multicores are superior. Several extensions to the Hill and Marty model have been developed for modeling "uncore" components (e.g., interconnection network and last level cache), [Loh 2008], computing core configuration optimal for energy [Cho and Melhem 2008; Lee et al. 2009], and leakage power [Woo and Lee 2008]. All these model uses area as the primary constraint and model single-core area/performance tradeoff using Pollack's rule (Performance $\propto \sqrt{Area}$ [Pollack 1999]) without considering technology trends.

Azizi et al. [2010] derive the single-core energy/performance trade-off as Pareto frontiers using architecture-level statistical models combined with circuit-level energy-performance trade-off functions. For modeling single-core power/performance and area/performance trade-offs, our core model derives two separate Pareto frontiers from empirical data. Further, we project these trade-off functions to the future technology nodes using our device model. Esmaeilzadeh et al. [2011] perform a power/energy Pareto efficiency analysis at 45 nm using total chip power measurements in the context of a retrospective workload and microarchitecture analysis. In contrast to the total chip power measurements for specific workloads, we use the power and area budget allocated to a single-core to derive the Pareto frontiers and combine those with our device and chip-level models to study the future of multicore design and the implications of technology scaling.

Chakraborty [2008] considers device-scaling and estimates a simultaneous activity factor for technology nodes down to 32 nm. Hempstead et al. [2009] 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. Chung et al. [2010] study unconventional cores including custom logic, FPGAs, or GPUs in heterogeneous single-chip design. They rely on Pollack's rule for the area/performance and power/performance tradeoffs. Using ITRS projections, they report on the potential for unconventional cores considering parallel kernels. Hardavellas et al. [2011] forecast the limits of multicore scaling and the emergence of dark silicon in servers with workloads that have an inherent abundance of parallelism. Using ITRS projections, Venkatesh et al. [2010] estimate technology-imposed utilization limits and motivate energy-efficient and application-specific core designs.

Previous work largely abstracts away processor organization and application details. This study provides a comprehensive model that considers the implications of process technology scaling, decouples power/area constraints, uses real measurements to model single-core design trade-offs, and exhaustively considers multicore organizations, microarchitectural features, and real applications and their behavior.

## 10. CONCLUSIONS AND FUTURE DIRECTIONS

For decades, 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 some time permitted performance scaling for parallel and multitasked workloads, permitting the economics of process scaling to hold. A key question for our 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 industry need to move in different, perhaps radical, directions to justify the cost of scaling? This article 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 our industry is accustomed to. Even with aggressive ITRS scaling projections, scaling cores achieves a geometric mean 7.9× speedup through 2024 at 8 nm. The article's findings and methodology are both significant and indicate that without process breakthroughs, directions beyond multicore are needed to provide performance scaling.

*The Glass Half-Empty.* A pessimistic interpretation of this study is that the performance improvements we have grown accustomed to over the past 30 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, and to sustain the current cadence of Moore's Law, must occur 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 industry product cycle. Furthermore, while a slowing of Moore's Law will obviously not be fatal, it has significant economic implications for the semiconductor industry.

*The Glass Half-Full.* The study shows that performance speedup is left from multicore scaling. If energy efficiency breakthroughs are made on supply voltage and process scaling, the performance improvement potential is high for applications with very high degrees of parallelism.

*Rethinking Multicore's Long-Term Potential.* We hope that this article's quantitative findings trigger some analyses in both academia and industry on the long-term potential of the multicore strategy. Academia is now making a major investment in research focusing on multicore and its related problems of expressing and managing parallelism. Research projects assuming hundreds or thousands of capable cores should consider this model and the power requirements under various scaling projections before assuming that the cores will inevitably arrive. The paradigm shift towards multicores that started in the high-performance general purpose market has already percolated to mobile and embedded markets. The qualitative trends we predict and our modeling methodology hold true for all markets even though our study considers the high-end desktop market. The results from this study may help break industry's current widespread consensus that multicore scaling is the viable forward path.

*Model Points to Opportunities.* Our study is based on a model that takes into account properties of devices, processor core, multicore organization and topology. Thus the model inherently provides the places 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. The results show that radical departures are needed and the model shows quantitative ways to measure the impact of such techniques.

*A Case for Microarchitecture Innovation.* Our study shows that fundamental processing limitations emanate from the processor core. 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 transistor integration capacity. Hence, improvements to the core's efficiency will have 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. We believe this study will revitalize and trigger microarchitecture innovations, making the case for their urgency and their potential impact. Our article serves as a quantitative and resounding affirmative answer to a recent ISCA panel questioning the relevance of microarchitecture research.

*A Case for Specialization.* There is emerging consensus that specialization is a promising alternative to energy-efficiently utilize transistors to improve performance. Our study serves as a quantitative motivation on the urgency of such work and its potential impact. Further, our study shows quantitatively the levels of energy improvement that must be delivered by specialization techniques.

*A Case for Complementing the Core.* Our study also shows that when performance becomes limited, techniques that occasionally use parts of the chip to deliver outcomes orthogonal to performance are ways to sustain the economics of the industry. However, techniques that focus on using the device integration capacity for improving security, programmer productivity, software maintainability, etc., must consider energy-efficiency as a primary factor.

*Contributions.* Our work makes two key contributions: (1) projecting multicore speedup limits and quantifying the dark silicon effect and (2) providing a novel and extendible model that integrates device scaling trends, core design trade-offs, and multicore configurations. While abstracting away many details, the model can find optimal configurations and project performance for CPU- and GPU-style multicores while taking into account microarchitectural features and high-level application properties. We make our model publicly available at `http://research.cs.wisc.edu/vertical/DarkSilicon`. We believe this study makes the case for innovation's urgency and its potential for high impact while providing a model that can be adopted as a tool by researchers and engineers to study limits of their solutions.

## REFERENCES

AMDAHL, G. M. 1967. Validity of the single processor approach to achieving large-scale computing capabilities. In *Proceedings of the AFIPS Conference*.

AZIZI, O., MAHESRI, A., LEE, B. C., PATEL, S. J., AND HOROWITZ, M. 2010. Energy-performance tradeoffs in processor architecture and circuit design: A marginal cost analysis. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*.

BAKHODA, A., YUAN, G. L., FUNG, W. W. L., WONG, H., AND AAMODT, T. M. 2009. Analyzing CUDA workloads using a detailed GPU simulator. In *Proceedings of the 2009 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*.

BHADAURIA, M., WEAVER, V., AND MCKEE, S. 2009. Understanding PARSEC performance on contemporary CMPs. In *Proceedings of the IEEE International Symposium on Workload Characterization (IISWC)*.

BIENIA, C., KUMAR, S., SINGH, J. P., AND LI, K. 2008. The PARSEC benchmark suite: Characterization and architectural implications. In *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques (PACT)*.

BORKAR, S. 2010. The exascale challenge. In *Proceedings of the International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*.

CHAKRABORTY, K. 2008. Over-provisioned multicore systems. Ph.D. thesis, University of Wisconsin-Madison.

CHO, S. AND MELHEM, R. 2008. Corollaries to Amdahl's law for energy. *Comput. Arch. Lett. 7,* 1.

CHUNG, E. S., MILDER, P. A., HOE, J. C., AND MAI, K. 2010. Single-chip heterogeneous computing: Does the future include custom logic, FPGAs, and GPUs? In *Proceedings of MICRO*.

DENNARD, R. H., GAENSSLEN, F. H., RIDEOUT, V. L., BASSOUS, E., AND LEBLANC, A. R. 1974. Design of ion-implanted mosfet's with very small physical dimensions. *IEEE J. Solid-State Circ. 9*.

DENNARD, R. H., CAI, J., AND KUMAR, A. 2007. A perspective on today's scaling challenges and possible future directions. *Solid-State Electron. 5*, 4, 518–525.

ESMAEILZADEH, H., CAO, T., YANG, X., BLACKBURN, S. M., AND MCKINLEY, K. S. 2011. Looking back on the language and hardware revolutions: Measured power, performance, and scaling. In *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'11)*. ACM.

GUZ, Z., BOLOTIN, E., KEIDAR, I., KOLODNY, A., MENDELSON, A., AND WEISER, U. C. 2009. Many-core vs. many-thread machines: Stay away from the valley. *IEEE Comput. Arch. Lett. 8*.

HARDAVELLAS, N., FERDMAN, M., FALSAFI, B., AND AILAMAKI, A. 2011. Toward dark silicon in servers. *IEEE Micro 31,* 4.

HEMPSTEAD, M., WEI, G.-Y., AND BROOKS, D. 2009. Navigo: An early-stage model to study power-contrained architectures and specialization. In *Proceedings of MoBS*.

HILL, M. D. AND MARTY, M. R. 2008. Amdahl's law in the multicore era. *Computer 41,* 7.

HOROWITZ, M., ALON, E., PATIL, D., NAFFZIGER, S., KUMAR, R., AND BERNSTEIN, K. 2005. Scaling, power, and the future of CMOS. In *Proceedings of the 2005 International Electron Devices Meeting (IEDM)*.

IPEK, E., KIRMAN, M., KIRMAN, N., AND MARTINEZ, J. F. 2007. Core fusion: Accommodating software diversity in chip multiprocessors. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*.

ITRS. 2011. International technology roadmap for semiconductors, 2010 update.

KIM, C., SETHUMADHAVAN, S., GOVINDAN, M. S., RANGANATHAN, N., GULATI, D., BURGER, D., AND KECKLER, S. W. 2007. Composable lightweight processors. In *Proceedings of MICRO*.

LEE, J.-G., JUNG, E., AND SHIN, W. 2009. An asymptotic performance/energy analysis and optimization of multi-core architectures. In *Proceedings of the International Conference on Distributed Computing and Networking (ICDCN)*.

LEE, V. W., KIM, E., CHHUGANI, J., DEISHER, M., KIM, D., NGUYEN, A. D., SATISH, N., SMELYANSKIY, M., CHENNUPATY, S., HAMMARLUND, P., SINGHAL, R., AND DUBEY, P. 2010. Debunking the 100X GPU vs. CPU myth: An evaluation of throughput computing on CPU and GPU. In *Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA'10)*. ACM, New York, 451–460.

LOH, G. 2008. The cost of uncore in throughput-oriented many-core processors. In *Proceedings of the Workshop on Architectures and Languages for Troughput Applications (ALTA)*.

MOORE, G. E. 1965. Cramming more components onto integrated circuits. *Electronics 38,* 8.

NOSE, K. AND SAKURAI, T. 2000. Optimization of VDD and VTH for low-power and high speed applications. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*.

POLLACK, F. 1999. New microarchitecture challenges in the coming generations of CMOS process technologies. In *Proceedings of MICRO*.

SPEC. 2011. Standard performance evaluation corporation.

SULEMAN, A. M., MUTLU, O., QURESHI, M. K., AND PATT, Y. N. 2009. Accelerating critical section execution with asymmetric multi-core architectures. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

VENKATESH, G., SAMPSON, J., GOULDING, N., GARCIA, S., BRYKSIN, V., LUGO-MARTINEZ, J., SWANSON, S., AND TAYLOR, M. B. 2010. Conservation cores: Reducing the energy of mature computations. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

WOO, D. H. AND LEE, H.-H. S. 2008. Extending Amdahl's law for energy-efficient computing in the many-core era. *Computer 41,* 12.

Received March 2012; accepted May 2012