Dark Silicon and the End of Multicore Scaling

Hadi Esmaeilzadeh
University of Washington

Emily Blem
University of Wisconsin—Madison

Renée St. Amant
University of Texas at Austin

Karthikeyan Sankaralingam
University of Wisconsin—Madison

Doug Burger
Microsoft Research

A KEY QUESTION FOR THE MICROPROCESSOR RESEARCH AND DESIGN COMMUNITY IS WHETHER SCALING MULTICORES WILL PROVIDE THE PERFORMANCE AND VALUE NEEDED TO SCALE DOWN MANY MORE TECHNOLOGY GENERATIONS. TO PROVIDE A QUANTITATIVE ANSWER TO THIS QUESTION, A COMPREHENSIVE STUDY THAT PROJECTS THE SPEEDUP POTENTIAL OF FUTURE MULTICORES AND EXAMINES THE UNDERUTILIZATION OF INTEGRATION CAPACITY—DARK SILICON—IS TIMELY AND CRUCIAL.

Moore's law (the doubling of transistors on chip every 18 months) has been a fundamental driver of computing.[1] For the past three decades, through device, circuit, microarchitecture, architecture, and compiler advances, Moore's law, coupled with Dennard scaling, has resulted in commensurate exponential performance increases.[2] The recent shift to multicore designs aims to increase the number of cores using the increasing transistor count to continue the proportional scaling of performance.

With the end of Dennard scaling, future technology generations can sustain the doubling of devices every generation, but with significantly less improvement in energy efficiency at the device level. This device scaling trend presages a divergence between energy-efficiency gains and transistor-density increases. For the architecture community, it is crucial to understand how effectively multicore scaling will use increased device integration capacity to deliver performance speedups in the long term. While everyone understands that power and energy are critical problems, no detailed, quantitative study has addressed how severe (or not) the power problem will be for multicore scaling, especially given the large multicore design space (CPU-like, GPU-like, symmetric, asymmetric, dynamic, composed/fused, and so forth).

To explore the speedup potential of future multicores, we conducted a decade-long performance scaling projection for multicore designs assuming fixed power and area budgets. It considers devices, core microarchitectures, chip organizations, and benchmark characteristics, applying area and power constraints at future technology nodes. Through our models we also estimate the effects of nonideal device scaling on integration capacity utilization and estimate the percentage of dark silicon (transistor integration capacity underutilization) on future multicore chips. For more information on related research, see the "Related Work in Modeling Multicore Speedup and Dark Silicon" sidebar.

Modeling multicore scaling

To project the upper bound performance achievable through multicore scaling (under current scaling assumptions), we considered technology scaling projections, single-core design scaling, multicore design choices,

Published by the IEEE Computer Society

Related Work in Modeling Multicore Speedup and Dark Silicon

Hill and Marty extend Amdahl's law to model multicore speedup with symmetric, asymmetric, and dynamic topologies and conclude that dynamic multicores are superior.[1] Their model uses area as the primary constraint and models single-core area/performance tradeoff using Pollack's rule (Performance $\propto \sqrt{\text{Area}}$) without considering technology trends.[2] Azizi et al. derive the single-core energy/performance tradeoff of Pareto frontiers using architecture-level statistical models combined with circuit-level energy/performance tradeoff functions.[3] For modeling single-core power/performance and area/performance tradeoffs, our core model derives two separate Pareto frontiers from real measurements. Furthermore, we project these tradeoff functions to the future technology nodes using our device model.

Chakraborty considers device scaling and estimates a simultaneous activity factor for technology nodes down to 32 nm.[4] 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.[5] Chung et al. study unconventional cores including custom logic, field-programmable gate arrays (FPGAs), or GPUs in heterogeneous single-chip design.[6] They rely on Pollack's rule for the area/performance and power/performance tradeoffs. Using *International Technology Roadmap for Semiconductors* (*ITRS*) projections, they report on the potential for unconventional cores considering parallel kernels. 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.[7] Using *ITRS* projections, Venkatesh et al. estimate technology-imposed utilization limits and motivate energy-efficient and application-specific core designs.[8]

Previous work largely abstracts away processor organization and application details. Our 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 tradeoffs; and exhaustively considers multicore organizations, microarchitectural features, and the behavior of real applications.

References

1. M.D. Hill and M.R. Marty, ''Amdahl's Law in the Multicore Era,'' *Computer,* vol. 41, no. 7, 2008, pp. 33-38.
2. F. Pollack, ''New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies,'' *Proc. 32nd Ann. ACM/IEEE Int'l Symp. Microarchitecture* (Micro 99), IEEE CS, 2009, p. 2.
3. O. Azizi et al., ''Energy-Performance Tradeoffs in Processor Architecture and Circuit Design: A Marginal Cost Analysis,'' *Proc. 37th Ann Int'l Symp. Computer Architecture* (ISCA 10), ACM, 2010, pp. 26-36.
4. K. Chakraborty, ''Over-Provisioned Multicore Systems,'' doctoral thesis, Department of Computer Sciences, Univ. of Wisconsin–Madison, 2008.
5. M. Hempstead, G.-Y. Wei, and D. Brooks, ''Navigo: An Early-Stage Model to Study Power-Constrained Architectures and Specialization,'' *Workshop on Modeling, Benchmarking, and Simulations* (MoBS), 2009.
6. E.S. Chung et al., ''Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPUs?'' *Proc. 43rd Ann. IEEE/ACM Int'l Symp. Microarchitecture* (Micro 43), IEEE CS, 2010, pp. 225-236.
7. N. Hardavellas et al., ''Toward Dark Silicon in Servers,'' *IEEE Micro,* vol. 31, no. 4, 2011, pp. 6-15.
8. G. Venkatesh et al., ''Conservation Cores: Reducing the Energy of Mature Computations,'' *Proc. 15th Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 10), ACM, 2010, pp. 205-218.

actual application behavior, and microarchitectural features. We considered fixed-size and fixed-power-budget chips. We built and combined three models to project performance, as Figure 1 shows. The three models are the device scaling model (DevM), the core scaling model (CorM), and the multicore scaling model (CmpM). The models predict performance speedup and show a gap between our projected speedup and the speedup we have come to expect with each technology generation. This gap is referred to as the *dark silicon gap*. The models also project the percentage of the dark silicon as the process technology scales.

We built a device scaling model that provides the area, power, and frequency scaling factors at technology nodes from 45 nm through 8 nm. We consider aggressive *International Technology Roadmap for Semiconductors* (*ITRS*; http://www.itrs.net) projections and conservative projections from Borkar's recent study.[3]

We modeled the power/performance and area/performance of single core designs using Pareto frontiers derived from real measurements. Through Pareto-optimal curves, the core-level model provides the maximum performance that a single core can sustain for any given area. Further, it provides the minimum power that must be consumed to sustain this level of performance.

We developed an analytical model that provides per-benchmark speedup of a

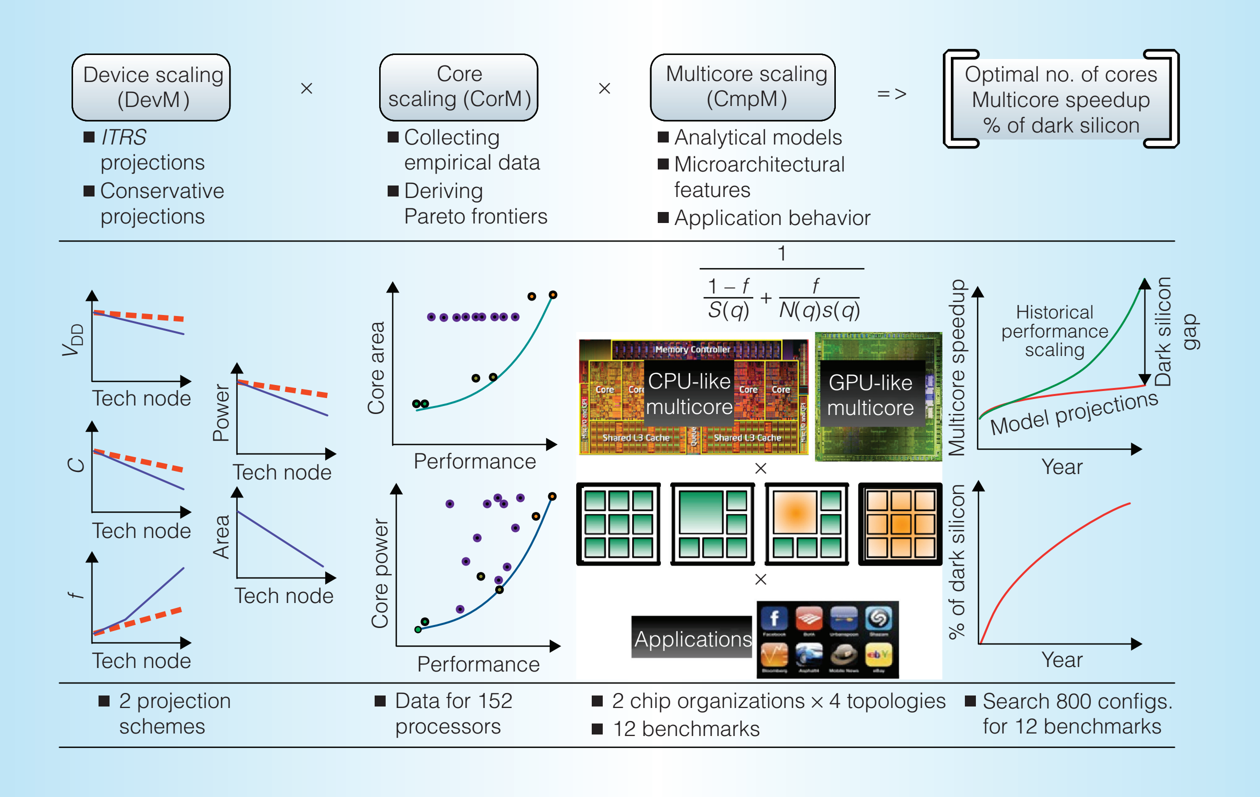

Figure 1. Overview of the methodology and models. By combining the device scaling model (DevM), core scaling model (CorM), and multicore scaling model (CmpM), we project performance speedup and reveal a gap between the projected speedup and the speedup expected with each technology generation indicated as the dark silicon gap. The three-tier model also projects the percentage of dark silicon as technology scales.

multicore design compared to a baseline design. The model projects performance for each hybrid configuration based on high-level application properties and microarchitectural features. We modeled the 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, heavyweight 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 considered four topologies: symmetric, asymmetric, dynamic, and composed (fused).

Table 1 outlines the four topologies in the design space and the cores' roles during serial and parallel portions of applications. Single-thread (ST) cores are uniprocessor-style cores with large caches, and many-thread (MT) cores are GPU-style cores with smaller caches.

Combining the device model with the core model provided power/performance and area/performance 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. Finally, combining all three models and performing an exhaustive design-space search produced the optimal multicore configuration and the maximum multicore speedups for each benchmark at future technology nodes while enforcing area, power, and benchmark constraints.

Future directions

As the rest of the article will elaborate, we model an upper bound on parallel application performance available from multicore

Table 1. The four multicore topologies for CPU-like and GPU-like organizations. (ST core: single-thread core; MT core: many-thread core.)

Multicore organization	Portion of code	Symmetric topology	Asymmetric topology		Dynamic topology	Composed topology
CPU multicore	Serial	1 ST core	1 large ST core		1 large ST core	1 large ST core
	Parallel	*N* ST cores	1 large ST core	+*N* small ST cores	*N* small ST cores	*N* small ST cores
GPU multicore	Serial	1 MT core (1 thread)	1 large ST core (1 thread)		1 large ST core (1 thread)	1 large ST core (1 thread)
	Parallel	*N* MT cores (multiple threads)	1 large ST core (1 thread)	+*N* small MT cores (multiple threads)	*N* small MT cores (multiple threads)	*N* small MT cores (multiple threads)

and CMOS scaling—assuming no major disruptions in process scaling or core efficiency. Using a constant area and power budget, this study shows that the space of known multicore designs (CPUs, GPUs, and their hybrids) or novel heterogeneous topologies (for example, 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. With conservative scaling, only 3.7× geometric mean speedup is achievable at 8 nm. Furthermore, with *ITRS* projections, at 22 nm, 21 percent of the chip will be dark, and at 8 nm, more than 50 percent of the chip cannot be utilized.

The article's findings and methodology are both significant and indicate that without process breakthroughs, directions beyond multicore are needed to provide performance scaling. For decades, Dennard scaling permitted more transistors, faster transistors, and more energy-efficient transistors with each new process node, which justified 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 the microprocessor research and design 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?

The glass is half-empty

A pessimistic interpretation of this study is that the performance improvements to which we have grown accustomed 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. This period is much shorter than the traditional academic time frame required for research and technology transfer. Major architecture breakthroughs in "alternative" directions such as neuromorphic computing, quantum computing, or biointegration 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 is half-full

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 our 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 toward 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. This study's results could 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, single instruction, multiple data (SIMD), and single instruction, multiple threads (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 also 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 use transistor integration capacity. Thus, improvements to the core's efficiency will impact performance improvement and will enable technology scaling even though the core consumes only 20 percent of the power budget for an entire laptop, smartphone, or tablet. We believe this study will revitalize and trigger microarchitecture innovations, making the case for their urgency and potential impact.

A case for specialization

There is emerging consensus that specialization is a promising alternative to efficiently use transistors to improve performance. Our study serves as a quantitative motivation on such work's urgency and potential impact. Furthermore, our study shows quantitatively the levels of energy improvement that specialization techniques must deliver.

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 industry's economics. However, techniques that focus on using the device integration capacity for improving security, programmer productivity, software maintainability, and so forth must consider energy efficiency as a primary factor.

Device scaling model (DevM)

The device model (DevM) provides transistor-area, power, and frequency-scaling factors from a base technology node (for example, 45 nm) to future technologies. The area-scaling factor corresponds to the shrinkage in transistor dimensions. The DevM model calculates the frequency-scaling factor based on the fanout-of-four (FO4) delay reduction. The model computes the power-scaling factor using the predicted frequency, voltage, and gate capacitance scaling factors in accordance with the $P = \alpha C V_{\mathrm{DD}}^2 f$ equation.

We generated two device scaling models: *ITRS* scaling and conservative scaling. The *ITRS* model uses projections from the 2010 *ITRS*. The conservative model is based on predictions presented by Borkar[3] and represents a less optimistic view. Table 2 summarizes the parameters used for calculating the power and performance-scaling factors. We allocated 20 percent of the chip power budget to leakage power and assumed chip designers can maintain this ratio.

Table 2. Scaling factors with *International Technology Roadmap for Semiconductors* (*ITRS*) and conservative projections. *ITRS* projections show an average 31 percent frequency increase and 35 percent power reduction per node, compared to an average 6 percent frequency increase and 23 percent power reduction per node for conservative projections.

Device scaling model	Year	Technology node (nm)	Frequency scaling factor (45 nm)	V_{DD} scaling factor (45 nm)	Capacitance scaling factor (45 nm)	Power scaling factor (45 nm)
ITRS scaling	2010	45*	1.00	1.00	1.00	1.00
	2012	32*	1.09	0.93	0.70	0.66
	2015	22†	2.38	0.84	0.33	0.54
	2018	16†	3.21	0.75	0.21	0.38
	2021	11†	4.17	0.68	0.13	0.25
	2024	8†	3.85	0.62	0.08	0.12
Conservative scaling	2008	45	1.00	1.00	1.00	1.00
	2010	32	1.10	0.93	0.75	0.71
	2012	22	1.19	0.88	0.56	0.52
	2014	16	1.25	0.86	0.42	0.39
	2016	11	1.30	0.84	0.32	0.29
	2018	8	1.34	0.84	0.24	0.22

* Extended Planar Bulk Transistors; † Multi-Gate Transistors.

Core scaling model (CorM)

We built the technology-scalable core model (CorM) 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. The q is a core's single-threaded performance. These frontiers capture the optimal area/performance and power/performance tradeoffs for a core while abstracting away specific details of the core.

As Figure 2 shows, we populated the two design spaces at 45 nm using 20 representative Intel and Advanced Micro Devices (AMD) processors and derive the Pareto frontiers. The curve that bounds all power/performance (area/performance) points in the design space and indicates the minimum amount of power (area) required for a given performance level constructs the Pareto frontier. The $P(q)$ and $A(q)$ pair, which are polynomial equations, constitute the core model. The core performance (q) is the processor's SPECmark and is collected from the SPEC website (http://www.spec.org). We estimated the core power budget using the thermal design power (TDP) reported in processor datasheets. The TDP is the chip power budget, or the amount of power the chip can dissipate without exceeding the transistor junction temperature. After excluding the share of uncore components from the power budget, we divided the power budget allocated to the cores to the number of cores to estimate the core power budget. 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 [L2] and Level-3 [L3] caches). Because this work's focus 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' share.

As Figure 2 illustrates, we fit a cubic polynomial, $P(q)$, to the points along the edge of the power/performance design space, and a quadratic polynomial (Pollack's rule[4]), $A(q)$, to the points along the edge of the area/performance design space. The Intel Atom Z520 with an estimated 1.89 W core TDP

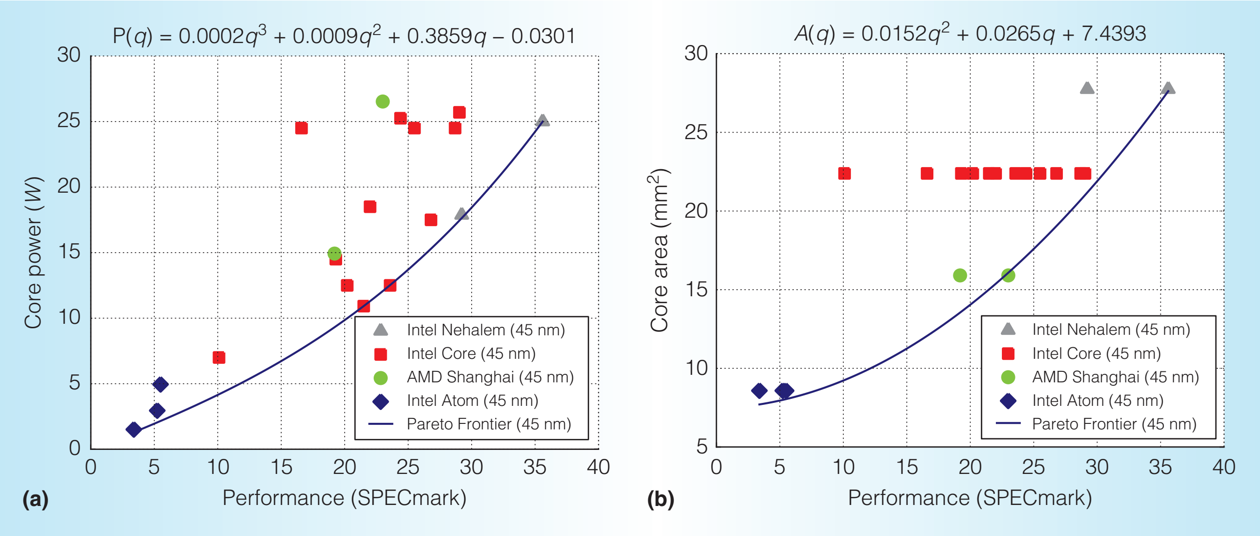

Figure 2. Design space and the derived Pareto frontiers. Power/performance frontier, 45 nm (a); area/performance frontier, 45 nm (b).

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 core TDP represents the highest-performing design (upper-right frontier point). We used the points along the scaled Pareto frontier as the search space for determining the best core configuration by the multicore scaling model.

Multicore scaling model (CmpM)

We developed 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. proposed a model for studying the first-order impacts of microarchitectural features (cache organization, memory bandwidth, threads per core, and so forth) and workload behavior (memory access patterns).[5] Their model considers stalls due to memory dependences and resource constraints (bandwidth or functional units). We extended 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 1. Table 3 lists the input parameters to the model, and how the multicore design choices impact them, if at all.

Microarchitectural features

Equation 1 calculates the multithreaded performance (*Perf*) of either a CPU-like or GPU-like multicore organization running a fully parallel ($f = 1$) and multithreaded application in terms of instructions per second by multiplying the number of cores (N) by the core utilization (η) and scaling by the ratio of the processor frequency to CPI_{exe}:

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

The CPI_{exe} parameter does not include stalls due to cache accesses, which are considered separately in core utilization (η). The core utilization (η) is the fraction of time that a thread running on the core can keep it busy. It is modeled as a function of the

Table 3. 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)	3,200	Core performance
CPI_{exe}	Cycles per instruction (zero-latency cache accesses)	1	Core performance, application
C_{L1}	Level-1 (L1) cache size per core (Kbytes)	64	Core style
C_{L2}	Level-2 (L2) cache size per chip (Mbytes)	2	Core style, multicore topology
t_{L1}	L1 access time (cycles)	3	N/A
t_{L2}	L2 access time (cycles)	20	N/A
t_{mem}	Memory access time (cycles)	426	Core performance
BW_{max}	Maximum memory bandwidth (Gbytes/s)	200	Technology node
b	Bytes per memory access (bytes)	64	N/A
f	Fraction of code that can be parallel	Varies	Application
r_m	Fraction of instructions that are memory accesses	Varies	Application
α_{L1}, β_{L1}	L1 cache miss rate function constants	Varies	Application
α_{L2}, β_{L2}	L2 cache miss rate function constants	Varies	Application

average time spent waiting for each memory access (t), fraction of instructions that access the memory (r_m), and the CPI_{exe}:

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

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}):

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

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

Multicore topologies

The multicore model is an extended Amdahl's law[6] equation that incorporates the multicore performance ($Perf$) calculated from Equations 1 through 4:

$$Speedup = 1/\left(\frac{f}{S_{parallel}} + \frac{1-f}{S_{serial}}\right) \tag{5}$$

The CmpM model (Equation 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 $S_{Serial} = Perf_S/Perf_B$.

We calculated the number of cores that can fit on the chip based on the multicore's topology, area budget (AREA), power budget (TDP), and each core's area [$A(q)$] and 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 large core's single-threaded performance. The area overhead of supporting composability is τ, while no power overhead is assumed for composability support.

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.

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 obtained this data from two previous studies.[7,8] 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.

To incorporate the Pareto-optimal curves into the CmpM model, we converted the SPECmark scores (q) into an estimated CPI_{exe} and core frequency. We assumed that 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} depends on its instruction mix and use of hardware optimizations such as functional units and out-of-order processing. Since the measured CPI_{exe} for each benchmark at each technology node is not available, we used 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 searched for the CPI_{exe} at each processor design point. We started 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 CmpM performance to the Nehalem core's CmpM performance is the same as the ratio of their SPECmark scores (q). We assumed that the CPI_{exe} does not change with technology node, while frequency scales.

A key component of the detailed model is the set of input parameters modeling the cores' microarchitecture. For single-thread cores, we assumed that each core has a 64-Kbyte L1 cache, and chips with only single-thread cores have an L2 cache that is 30 percent of the chip area. MT cores have small L1 caches (32 Kbytes for every eight cores), support multiple hardware contexts (1,024 threads per eight cores), a thread register file, and no L2 cache. From Atom and Tesla die photos, we estimated that eight small many-thread cores, their shared L1 cache, and their thread register file can fit in the same area as one Atom processor. We assumed that off-chip bandwidth (BW_{max}) increases linearly as process technology scales down and while the memory access time is constant.

We assumed that τ increases from 10 percent up to 400 percent, depending on the composed core's total area. The composed core's performance cannot exceed performance of a single Nehalem core at 45 nm.

We derived the area and power budgets 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's solely allocated to the cores, not caches and other uncore components. The CmpM's speedup baseline is a quad-Nehalem multicore.

Combining models

Our three-tier modeling approach allows us to exhaustively explore the design space of future multicores, project their upper bound performance, and estimate the amount of integration capacity underutilization, dark silicon.

Device × core model

To study core scaling in future technology nodes, we scaled the 45 nm Pareto frontiers down to 8 nm by scaling each processor data point's power and performance using the DevM model and then refitting the Pareto optimal curves at each technology node. We assumed that performance, which we measured in SPECmark, would scale linearly with frequency. By making this assumption, we ignored the effects of memory latency and bandwidth on the core performance. Thus, actual performance gains through scaling could be lower. Based on the optimistic *ITRS* model, scaling a microarchitecture (core) from 45 nm to 8 nm will result in a 3.9× performance improvement and an 88 percent reduction in power consumption. Conservative scaling, however, suggests that performance will increase only by 34 percent and that power will decrease by 74 percent.

Device × core × multicore model

We combined all three models to produce final projections for optimal multicore speedup, number of cores, and amount of dark silicon. To determine the best multicore configuration at each technology node, we swept the design points along the scaled area/performance and power/performance Pareto frontiers (DevM × CorM) because these points represent the most efficient designs. For each core design, we constructed a multicore consisting of one such core at each technology node. For a symmetric multicore, we iteratively added identical cores one by one until we hit the area or power budget or until performance improvement was limited. We swept the frontier and constructed a symmetric multicore for each processor design point. From this set of symmetric multicores, we picked the multicore with the best speedup as the optimal symmetric multicore for that technology node. The procedure is similar for other topologies. We performed this procedure 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 that is only allocated to the cores.

Scaling and future multicores

We used 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 12 real applications from the PARSEC suite.

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 or GPU-like) × (symmetric, asymmetric, dynamic, or composed). The exponential performance curve matches transistor count growth as process technology scales.

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

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

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

Dark silicon projections

To understand whether parallelism or the power budget is the primary source of the dark silicon speedup gap, we varied each of these factors in two experiments at 8 nm. First, we kept the power budget constant (our default budget is 125 W) and varied the level of parallelism in the PARSEC applications from 0.75 to 0.99, assuming that programmer effort can realize this improvement. Performance improved slowly as the parallelism level increased, with most benchmarks reaching a speedup of about only 15× at 99 percent parallelism. Provided that the power budget is the only limiting factor, typical upper-bound *ITRS*-scaling speedups will still be limited to 15×. With conservative scaling, this best-case speedup is limited to 6.3×.

For the second experiment, we kept each application's parallelism at its real level and varied the power budget from 50 W to 500 W. Eight of 12 benchmarks showed no more than 10× speedup even with a practically unlimited power budget. In other words, increasing core counts beyond a certain point did not improve performance because of the limited parallelism in the applications and Amdahl's law. Only four benchmarks have sufficient parallelism to even hypothetically sustain speedup levels that matches the exponential transistor count growth, Moore's law.

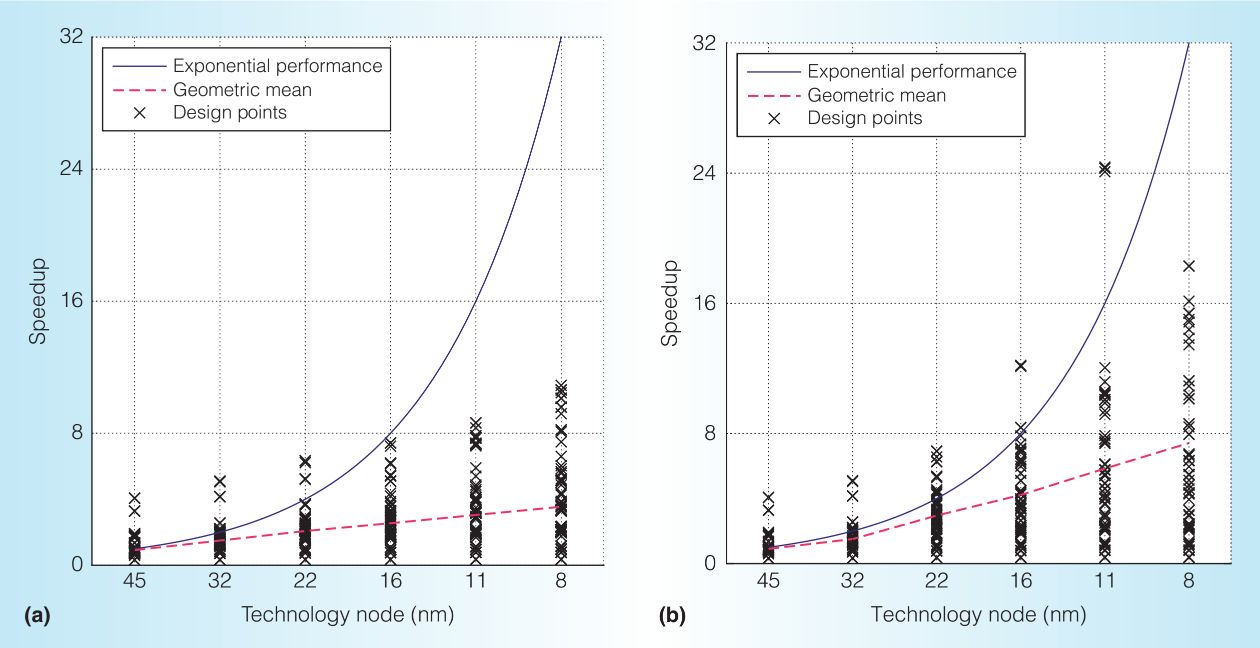

Figure 3. Speedup across process technology nodes across all organizations and topologies with PARSEC benchmarks. The exponential performance curve matches transistor count growth. Conservative scaling (a); *ITRS* scaling (b).

Finding: With *ITRS* projections, at 22 nm, 21 percent of the chip will be dark, and at 8 nm, more than 50 percent of the chip cannot be utilized.

Finding: 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.

Core count projections

Different applications saturate performance improvements at different core counts. We considered the chip configuration that provided the best speedups for all applications to be an ideal configuration. Figure 4 shows the number of cores (solid line) for the ideal CPU-like dynamic multicore configuration across technology generations, because dynamic configurations performed best. The dashed line illustrates the number of cores required to achieve 90 percent 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 percent of the speedup achievable by 442 cores. With conservative scaling, the 90 percent speedup core count is 20 at 8 nm.

Finding: 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.

Sensitivity studies

We performed sensitivity studies on the impact of various features, including L2 cache sizes, memory bandwidth, simultaneous multithreading (SMT) support, and the percentage of total power allocated to leakage. Quantitatively, these studies show that these features have limited impact on multicore performance.

Limitations

Our device and core models do not explicitly consider dynamic voltage and frequency scaling (DVFS). Instead, we take an optimistic approach to account for its best-case impact. When deriving the Pareto frontiers, we assume that each processor data point operates at its optimal voltage and

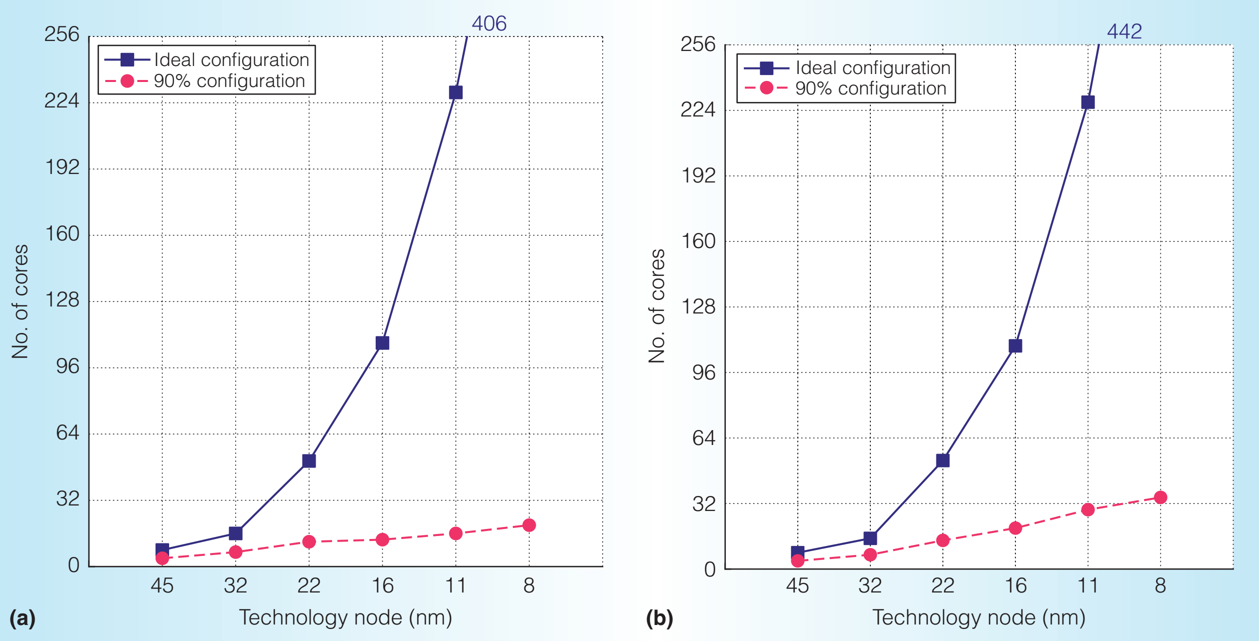

Figure 4. Number of cores for the ideal CPU-like dynamic multicore configurations and the number of cores delivering 90 percent of the speedup achievable by the ideal configurations across the PARSEC benchmarks. Conservative scaling (a); *ITRS* scaling (b).

frequency setting ($V_{\mathrm{DD}_{\min}}, Freq_{\max}$). At a fixed V_{DD} setting, scaling down the frequency from $Freq_{max}$ results in a power/performance point inside the optimal Pareto curve, which is a suboptimal design point. However, scaling voltage up and operating at a new ($V'_{\mathrm{DD}_{\min}}, Freq'_{\max}$) setting results in a different power-performance point that is still on the optimal frontier. Because 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 DVFS. The multicore model considers the first-order impact of caching, parallelism, and threading under assumptions that result only in optimistic projections. Comparing the CmpM model's output against published empirical results confirms that our model always overpredicts multicore performance. The model optimistically assumes that the workload is homogeneous; that work is infinitely parallel during parallel sections of code; that memory accesses never stall due to a previous access; and that no thread synchronization, operating system serialization, or swapping occurs.

This work makes two key contributions: projecting multicore speedup limits and quantifying the dark silicon effect, and providing a novel and extendible model that integrates device scaling trends, core design tradeoffs, and multicore configurations. While abstracting away many details, the model can find optimal configurations and project performance for CPU- and GPU-style multicores while considering microarchitectural features and high-level application properties. We made 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 researchers and engineers can adopt as a tool to study limits of their solutions. MICRO

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 the NSF under grants CCF-0845751, CCF-0917238, and CNS-0917213.

References

1. G.E. Moore, ''Cramming More Components onto Integrated Circuits,'' *Electronics,* vol. 38, no. 8, 1965, pp. 56-59.
2. R.H. Dennard et al., ''Design of Ion-Implanted Mosfet's with Very Small Physical Dimensions,'' *IEEE J. Solid-State Circuits,* vol. 9, no. 5, 1974, pp. 256-268.
3. S. Borkar, ''The Exascale Challenge,'' *Proc. Int'l Symp. on VLSI Design, Automation and Test* (VLSI-DAT 10), IEEE CS, 2010, pp. 2-3.
4. F. Pollack, ''New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies,'' *Proc. 32nd Ann. ACM/IEEE Int'l Symp. Microarchitecture* (Micro 99), IEEE CS, 2009, p. 2.
5. Z. Guz et al., ''Many-Core vs. Many-Thread Machines: Stay Away From the Valley,'' *IEEE Computer Architecture Letters,* vol. 8, no. 1, 2009, pp. 25-28.
6. G.M. Amdahl, ''Validity of the Single Processor Approach to Achieving Large-scale Computing Capabilities,'' *Proc. Joint Computer Conf. American Federation of Information Processing Societies* (AFIPS 67), ACM, 1967, doi:10.1145/1465482.1465560.
7. M. Bhadauria, V. Weaver, and S. McKee, ''Understanding PARSEC Performance on Contemporary CMPs,'' *Proc. IEEE Int'l Symp. Workload Characterization* (IISWC 09), IEEE CS, 2009, pp. 98-107.
8. C. Bienia et al., ''The PARSEC Benchmark Suite: Characterization and Architectural Implications,'' *Proc. 17th Int'l Conf. Parallel Architectures and Compilation Techniques* (PACT 08), ACM, 2008, pp. 72-81.

Hadi Esmaeilzadeh is a PhD student in the Department of Computer Science and Engineering at the University of Washington. His research interests include power-efficient architectures, approximate general-purpose computing, mixed-signal architectures, machine learning, and compilers. Esmaeilzadeh has an MS in computer science from the University of Texas at Austin and an MS in electrical and computer engineering from the University of Tehran.

Emily Blem is a PhD student in the Department of Computer Sciences at the University of Wisconsin–Madison. Her research interests include energy and performance tradeoffs in computer architecture and quantifying them using analytic performance modeling. Blem has an MS in computer science from the University of Wisconsin–Madison.

Renée St. Amant is a PhD student in the Department of Computer Science at the University of Texas at Austin. Her research interests include computer architecture, low-power microarchitectures, mixed-signal approximate computation, new computing technologies, and storage design for approximate computing. St. Amant has an MS in computer science from the University of Texas at Austin.

Karthikeyan Sankaralingam is an assistant professor in the Department of Computer Sciences at the University of Wisconsin–Madison, where he also leads the Vertical Research Group. His research interests include microarchitecture, architecture, and very-large-scale integration (VLSI). Sankaralingam has a PhD in computer science from the University of Texas at Austin.

Doug Burger is the director of client and cloud applications at Microsoft Research, where he manages multiple strategic research projects covering new user interfaces, datacenter specialization, cloud architectures, and platforms that support personalized online services. Burger has a PhD in computer science from the University of Wisconsin. He is a fellow of IEEE and the ACM.

Direct questions and comments about this article to Hadi Esmaeilzadeh, University of Washington, Computer Science & Engineering, Box 352350, AC 101, 185 Stevens Way, Seattle, WA 98195; hadianeh@cs.washington.edu.