RETROSPECTIVE:
Dark Silicon and the End of Multicore Scaling

Hadi Esmaeilzadeh[†] Emily Blem[‡] Renée St. Amant Karthikeyan Sankaralingam[§] Doug Burger[◇]

[†]University of California, San Diego [‡]Google [§]UW-Madison, NVIDIA [◇]Microsoft
hadi@ucsd.edu emilyblem@google.com renee.st.amant@gmail.com karu@cs.wisc.edu dburger@microsoft.com

I. The Past: Economical and Historical Context

Economic context. Historically, architectural advancements in conjunction with device improvements realized the twin benefits of Moore's law and Dennard scaling – a doubling of the performance of *general-purpose* microprocessors on a consistent two-year cadence. Until the early 2000s, the computing industry benefited enormously from this consistent, exponential growth in singled-threaded performance. The consistency of these performance improvements enabled powerful computing to become a commodity, reaching masses of developers and generating a Cambrian explosion in the domains and uses of computing. The continued performance growth allowed the computing industry to operate on the economics of new capabilities (growing markets), rather than the economics of replacement in a stable or saturated market.

By providing *higher performance* for general-purpose computing at a *fixed cost*, computer architecture advances were then, and still are, a key enabler of the computing industry's economic cycle. That is, our innovations enable software to harness technology advances into new capabilities.

The multicore era. In the early 2000s, the decline and eventual end of Dennard scaling benefits (for devices below 32nm) became clear. There was also a growing awareness that single-thread performance growth would slow appreciably below historical trends. Facing the clear need to continue growing capabilities, with acceptable disruption to software, the computing industry pivoted to a multicore strategy. That is, increasing the number of cores per die each generation to continue performance improvements, while sustaining small improvements in single-core efficiency. Shortly before we began the investigation that led to this paper, some of us attended a panel at HPCA 2008. The majority of the panelists expressed the view that exponential increases in CPU core counts (as many as thousands of cores on a chip), would be the principal way to scale general-purpose performance going forward. The majority also agreed that the community must solve the software parallelization problem to get there. That consensus inspired us to investigate whether traditional parallel computing approaches were a viable path forward in an era of power-limited silicon.

Combining the insights of device scaling with architecture trends, there was a growing concern that the failure of Dennard Scaling meant that power would not scale down proportionally as the core count increased across technology generations. However, the severity of the issues at the transistor level and how those would play out with respect to core scaling and application parallelism was not well understood at the time. The main question that we tried to address was: *How effective would multicore designs be in a power-limited era, for differing degrees of parallelism*?

II. Technical Summary

To answer this question, the paper generated a ten-year performance scaling projection for future multicore designs through a model containing multiple factors. The multicore modeling took into account transistor scaling trends, processor core design options, chip multiprocessor organizations, and benchmark characteristics, while applying area and power constraints at future technology nodes. The model combined these factors to project the upper bound speedup achievable through multicore scaling under various technology scaling trends. The paper's principal result showed that the power issue, even assuming simpler cores, would result in lower utilization of the silicon (**dark silicon**) – the fraction of the chip that must be powered off to meet power constraints. Dark silicon was a useful concept and term to illustrate the challenge. However, it was unclear to us whether the power limits would result in designs with substantial fractions powered off, smaller chips, or lower average activity factor across active silicon. This result appeared to us to indicate that techniques other than traditional multicore scaling would be required to advance application performance. This conclusion was counter to the directions many in the architecture community were pursuing at the time. The paper's modeling considered the trends from 2008, when 45 nm microprocessors were available, to 2018. The evaluations showed that with optimistic projections from International Technology Roadmap for Semiconductors (ITRS), only 7.9× average speedup would be possible for commonly used parallel workloads. This limited improvement would leave a nearly 24-fold gap from a "doubled performance per generation" expectation. This gap would grow to 28-fold under conservative scaling projections where only 3.7× speedup would be achievable.

Based on these results, we predicted that power limits and imperfect parallelism would drive a shift in architectures–that parallel applications running on an increasing number of CPU (or GPU) cores would not be the principal vector of performance scaling. Without a shift, it seemed unlikely that historical levels of application performance increases would be viable.

Origins of the term "Dark Silicon". Dark Silicon became a popular term used as short-hand for the limited power problem. In 2010, we had been explicitly thinking of a good term to capture this phenomenon. The best we came up with was "pinhole processing," but it was insufficient. In a 2010 meeting with ARM, their then-CTO, Mike Muller, referred to the concept as "dark silicon" and it was clearly the right term. We did not coin the term, we got it from Mark.

At that time, we were not the only researchers exploring and modeling this phenomenon. Concurrently to us, Hardavellas *et al.* were investigating dark silicon and multicore scaling in servers [1]. Also, Conservation Cores [2] considered the utilization wall that leads to dark silicon.

III. The Present: The Post MultiCore Era

Multicore architectures (in the sense of scaling the number of CPUs) did not end up being the primary vehicle for accelerating applications. However, high core counts did become extremely important for task-level parallelism as a means to improve the cost efficiency of virtual machines in datacenters. Today, multicore chips are ubiquitous at every level with task parallelism being the primary use case. Today, even on multicore architectures, we do not see large swaths of the chip being "dark." We instead see **dim silicon**, with large areas of the chip devoted to caches alongside the use of aggressive voltage/frequency scaling. We continue to see modest inter-generational (10%) improvements in CPU performance, which come from both process scaling and continued microarchitectural improvements.

Specialization and ISAs The timing of this paper shortly proceeded a rapid rise of research and development in Domain-Specific Architectures (DSAs). This shift would certainly have happened without this paper. Much of this paper's visibiliy was due to timing, it was published between the widespread "multicore consensus" in the late 2000's and the rise of large-scale deployments of accelerators starting in the mid 2010s. It is important to note that multicore consensus was correct; however, in its focus on parallelism. All of the specialized accelerators leverage parallelism in different ways. These forms of parallelism have mostly departed from parallel CPU ISAs running on many general cores. Instead, accelerators leverage different underlying parallelism models with higher-level software frameworks such as Pytorch expressing domain-specific parallel semantics.

Deep learning - a pleasant surprise. While we predicted that neither CPU-like or GPU-like multicore designs were sufficient to deliver the expected computing advancements, the computer architecture community (and many other communities) did not anticipate the rapid adoption of specialized hardware for Deep Neural Networks (DNNs). Deep learning was starting to emerge at the time this paper was published, but differed significantly from the structure of traditional parallel workloads. These workloads are embarrassingly parallel, with ample amounts of optimization opportunities in the memory system, arithmetic, computing silicon, and at the system level. The rather general applicability of deep learning to numerous use cases in various industries, such as healthcare, finance, and automotive, justify the economics of specialization for this domain. The economics of this trend are so powerful that CPUs, GPUs, and FPGAs now all include some form of specialization for deep learning, and chips, such as Google TPUs and Amazon Inferentia/Trainium, are built by software companies entirely for deep learning. GPUs have since become the dominant architecture for accelerating deep learning. Successive GPU generations spanning 28 nm to 7 nm (Tesla, Pascal, Volta, Ampere, Hopper) have increased sustained performance on deep learning by 16× in a decade. The deep learning revolution has been fueled by a virtuous cycle of rapid algorithmic, hardware, and software changes.

Approximate computing. Another example of an avenue of radical innovation that continues to unfold is that of approximate computing. While accelerators relax the long-held ISA abstraction, approximation relaxes the notion of perfect precision. Examples of approximate computing include reduced bit arithmetic, dual-voltage calculations, stepping over computation, computation substitution with less-precise alternatives, value prediction, transforming imperative code to neural networks for digital and analog implementation, etc.

IV. The Future

When we were writing the paper in 2010, we were thinking about how the end of Dennard scaling would affect the major directions in computer architecture. We were asking whether multicore chips would be the dominant path forward or whether there would be a shift. Core counts did continue to increase, cloud computing grew enormously in importance, mobile devices became more capable, and many specialized extensions to CPUs enabled these classes of computing. In retrospect, the biggest disruption has been driven by algorithms, particularly deep learning and generative inference. Increasingly deep learning seems to be becoming a "second great computing platform" (imperative code running on CPUs being the first). This new workload is driving rapid changes in architecture, algorithms, and systems, akin to the level of computer architecture innovation in the 1960's and 1970's. In the past we thought of these workloads as "specialized computing." We think it will increasingly be seen as a second class of general-purpose computation. The class that relies on statistical pattern identification rather than determinism. The small-to-negligible improvements in CMOS power efficiency still pose a challenge, which may motivate more radical innovations in computer architecture: photonics, quantum computing, analog computing, and biological computing with neurons could emerge as a cross-cutting enabler to address the continuing power challenge. When we wrote this paper, we predicted an inflection point and a shift in architectures. Deep learning and generative AI, fueled by data,is likely driving another inflection point in how data are generated, consumed, monitored and monetized. Some prognosticators compare this shift to seismic changes like the industrial revolution. Computer architecture innovations fuel these changes and their economic benefits, as we build the abstractions and mechanisms that harvest physics to transmute data to these new capabilities. Our collective work has been instrumental in making CPU-based computing, and now AI (currently GPU)-based computing, an integral part of the human experience. For computer architecture, there are exciting questions regarding not only how to enable these capabilities, but also how to make use of them responsibly.

References

[1] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki, "Toward dark silicon in servers," *IEEE Micro*, vol. 31, no. 4, pp. 6–15, 2011.

[2] G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin, J. Lugo-Martinez, S. Swanson, and M. B. Taylor, "Conservation cores: Reducing the energy of mature computations," in *ASPLOS*, 2010.