A Landscape of the New Dark Silicon Design Regime

THE RISE OF DARK SILICON IS DRIVING A NEW CLASS OF ARCHITECTURAL TECHNIQUES THAT ''SPEND'' AREA TO ''BUY'' ENERGY EFFICIENCY. THIS ARTICLE EXAMINES FOUR RECENTLY PROPOSED DIRECTIONS (''THE FOUR HORSEMEN'') FOR ADAPTING TO DARK SILICON, OUTLINES A SET OF EVOLUTIONARY DARK SILICON DESIGN PRINCIPLES, AND SHOWS HOW ONE OF THE DARKEST COMPUTING ARCHITECTURES—THE HUMAN BRAIN—OFFERS INSIGHTS INTO MORE REVOLUTIONARY DIRECTIONS FOR COMPUTER ARCHITECTURE.

Michael B. Taylor
University of California, San Diego

Recent VLSI technology trends have led to a disruptive new regime for digital chip designers, where Moore's law continues but CMOS scaling provides increasingly diminished fruits. As in prior years, the computational capabilities of chips are still increasing by 2.8× per process generation. However, a utilization wall[1] limits us to only 1.4× of this benefit—causing large underclocked swaths of silicon area—hence the term *dark silicon*.[2,3]

Fortunately, simple scaling theory makes the utilization wall easy to derive, helping us to think intuitively about the problem. Transistor density continues to improve by 2× every two years, and native transistor speeds improve by 1.4×. But transistor energy efficiency improves by only 1.4×, which, under constant power budgets, causes a 2× shortfall in energy budget to power a chip at its native frequency. Therefore, our utilization of a chip's potential is falling exponentially by a jaw-dropping 2× per generation. Thus, if we are just bumping up against power limitations in the current generation, then in eight years, designs will be 93.75 percent dark!

A recent paper refers to this widespread disruptive factor informally as the ''dark silicon apocalypse,''[4] because it officially marks the end of one reality (Dennard scaling[5]), where progress could be measured by improvements in transistor speed and count, and the beginning of a new reality (post-Dennard scaling), where progress is measured by improvements in transistor energy efficiency. Previously, we tweaked our circuits to reduce transistor delays and turbo-charged them with dual-rail domino to reduce fan-out-of-4 (FO4) delays. From now on, we will tweak our circuits to minimize capacitance switched per function; we will strip our circuits down and starve them of voltage to squeeze out every femtojoule. Whereas once we would spend exponentially increasing quantities of transistors to buy performance, now we will spend these transistors to buy energy efficiency.

The CMOS scaling breakdown was the direct cause of industry's transition to multicore in 2005. Because filling chips with cores does not fundamentally circumvent utilization wall limits, multicore is not the final

Published by the IEEE Computer Society

solution to dark silicon;[3] it is merely industry's initial, transitional response to the shocking onset of the dark silicon age. Increasingly over time, the semiconductor industry is adapting to this new design regime, realizing that multicore chips will not scale as transistors shrink and that the fraction of a chip that can be filled with cores running at full frequency is dropping exponentially with each process generation.[1,3] This reality forces designers to ensure that, at any point in time, large fractions of their chips are effectively dark—either idle for long periods of time or significantly underclocked. As exponentially larger fractions of a chip's transistors become darker, silicon area becomes an exponentially cheaper resource relative to power and energy consumption. This shift calls for new architectural techniques that "spend" area to "buy" energy efficiency. This saved energy can then be applied to increase performance, or to have longer battery life or lower operating temperatures.

The utilization wall that causes dark silicon

Table 1 shows the derivation of the utilization wall[1] that causes dark silicon.[2,3] It employs a scaling factor, S, which is the ratio between the feature sizes of two processes (for example, $S = 32/22 = 1.4x$ between 32 and 22 nm). In both Dennard and post-Dennard scaling, the transistor count scales by S^2, and the transistor switching frequency scales by S. Thus, our net increase in computing performance is S^3, or $2.8x$.

However, to maintain a constant power envelope, these gains must be offset by a corresponding reduction in transistor switching energy. In both cases, scaling reduces transistor capacitance by S, improving energy efficiency by S. In Dennard scaling, we can scale the threshold voltage and thus the operating voltage, which yields another S^2 energy-efficiency improvement. However, in today's post-Dennard, leakage-limited regime, we cannot scale threshold voltage without exponentially increasing leakage, and as a result, we must hold operating voltage roughly constant. The end result is a shortfall of S^2, or 2× per process generation. This shortfall multiplies with each process generation, resulting in exponentially darker silicon over time.

Table 1. Dennard vs. post-Dennard (leakage-limited) scaling.[1] In contrast to Dennard scaling,[5] which held until 2005, under the post-Dennard regime, the total chip utilization for a fixed power budget drops by S^2 with each process generation. The result is an exponential increase in dark silicon for a fixed-sized chip under a fixed area budget.

Transistor property	Dennard	Post-Dennard
Δ Quantity	S^2	S^2
Δ Frequency	S	S
Δ Capacitance	$1/S$	$1/S$
V_{DD}^2	$1/S^2$	1
⇒ Δ Power = Δ $QFCV^2$	1	S^2
⇒ Δ Utilization = 1/Power	1	$1/S^2$

This shortfall prevents multicore from being the solution to scaling.[1,3] Although advancing a single process generation would allow enough transistors to increase core count by 2×, and frequency could be 1.4× faster, the energy budget permits only a 1.4× total improvement. Per Figure 1, across two process generations ($S = 2$), designers could increase core count by 2× leaving frequency constant, or they could increase frequency by 2× with leaving core count constant, or they could choose some middle ground between the two. The remaining 4× potential remains inaccessible.

More positively stated, the true new potential of Moore's law is a 1.4× energy-efficiency improvement per generation, which could be used to increase performance by 1.4×. Additionally, if we could somehow make use of dark silicon, we could do even better.

Although the utilization wall is based on a first-order model that simplifies many factors, it has proved to be an effective tool for designers to gain intuition about the future, and has proven remarkably accurate (see the sidebar "Is Dark Silicon Real? A Reality Check"). Follow-up work[6-8] has looked at extending this early work[1,3] on dark silicon and multicore scaling with more sophisticated models that incorporate factors such as application space and cache size.

Dark silicon misconceptions

Let's clear up a few misconceptions before proceeding. First, dark silicon does not mean blank, useless, or unused silicon; it's just

Is Dark Silicon Real? A Reality Check

A quick survey of recent designs from multicore outfits such as Tilera, Intel, and AMD indicates that industry has pursued core count and frequency combinations consistent with the utilization wall. For instance, Intel's 90-nm single-core Prescott chip ran at 3.8 GHz in 2004. Dennard scaling would suggest that a 22-nm multicore version should run at 15.5 GHz, and contain 17 superscalar cores, for a total improvement of 69× in instruction throughput. Instead, the upcoming 2013 22-nm Intel Core i7 4960X runs at 3.6 GHz and has six superscalar cores, a 5.7× peak serial instruction throughput improvement. The darkness ratio is thus 91.74 percent versus the 93.75 percent predicted by the utilization wall. The latest 2012 *International Technology Roadmap for Semiconductors* also shows that scaling has proceeded consistently with post-Dennard predictions.

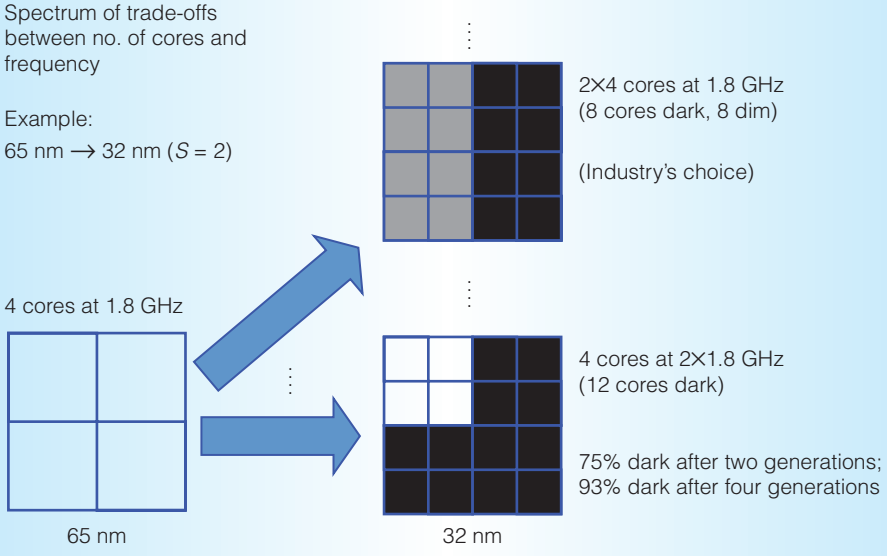

Figure 1. Multicore scaling leads to large amounts of dark silicon.[3] Across two process generations, there is a spectrum of trade-offs between frequency and core count; these include increasing core count by 2× but leaving frequency constant (top), and increasing frequency by 2× but leaving core count constant (bottom). Any of these trade-off points will have large amounts of dark silicon.

silicon that is not used all the time, or at its full frequency. Even during the best days of CMOS scaling, microprocessor and other circuits were chock full of "dark logic" used infrequently or for only some applications—for instance, caches are inherently dark because the average cache transistor is switched for far less than one percent of cycles, and FPUs remain dark in integer codes.

Soon, the exponential growth of dark silicon area will push us beyond logic targeted for direct performance benefits toward swaths of low-duty cycle logic that exists, not for direct performance benefit, but for improving energy efficiency. This improved energy efficiency can then allow an indirect performance improvement because it frees up more of the fixed power budget to be used for even more computation.

The four horsemen

Recently, researchers proposed a taxonomy—the four horsemen—that identifies four promising directions for dealing with dark silicon that have emerged as promising potential approaches as we transition beyond the initial multicore stop-gap solution. These responses originally appeared to be unlikely candidates, carrying unwelcome burdens in design, manufacturing, or programming. None is ideal from an aesthetic engineering

point of view. But the success of complex multiregime devices such as metal-oxide-semiconductor field-effect transistors (MOSFETs) has shown that engineers can tolerate complexity if the end result is better. Future chips are likely to employ not just one horseman, but all of them, in interesting and unique combinations.

The shrinking horseman

When confronted with the possibility of dark silicon, many chip designers insist that area is expensive, and that they would just build smaller chips instead of having dark silicon in their designs. Among the four horsemen, these "shrinking chips" are the most pessimistic outcome. Although all chips may eventually shrink somewhat, the ones that shrink the most will be those for which dark silicon cannot be applied fruitfully to improve the product. These chips will rapidly turn into low-margin businesses for which further generations of Moore's law provide small benefit. Below is an examination of the spectrum of second-order effects associated with shrinking chips.

Cost side of shrinking silicon. Understanding shrinking chips requires considering semiconductor economics. The "build smaller chips" argument has a ring of truth; after all, designers spend much of their time trying to meet area budgets for existing chip designs. But exponentially smaller chips are not exponentially cheaper; even if silicon begins as 50 percent of system cost, after a few process generations, it will be a tiny fraction. Mask costs, design costs, and I/O pad area will fail to be amortized, leading to rising costs per mm^2 of silicon, which ultimately will eliminate incentives to move the design to the next process generation. These designs will be "left behind" on older generations.

Revenue side of shrinking silicon. Shrinking silicon can also shrink the chip selling price. In a competitive market, if there is a way to use the next process generation's bounty of dark silicon to attain a benefit to the end product, then competition will force companies to do so. Otherwise, they will generally be forced into low-end, low-margin, high-competition markets, and their competitor will take the high end and enjoy high margins. Thus, in scenarios where dark silicon could be used profitably, decreasing area in lieu of exploiting it would certainly decrease system costs, but would catastrophically decrease sale price. Hence, the shrinking-chips scenario is likely to happen only if we can find no practical use for dark silicon.

Power and packaging issues with shrinking chips. A major consequence of exponentially shrinking chips is a corresponding exponential rise in power density. Recent analysis of many-core thermal characteristics has shown that peak hotspot temperature rise can be modeled as $T_{max} = TDP \times (R_{conv} + k/A)$, where T_{max} is the rise in temperature, TDP is the target chip thermal design power, R_{conv} is the heat sink thermal convection resistance (lower is a better heat sink), k incorporates many-core design properties, and A is chip area.[8] If area drops exponentially, the second term dominates and chip temperatures rise exponentially. This in turn will force a lower TDP so that temperature limits are met, and reduce scaling below even the nominal 1.4× expected energy-efficiency gain. Thus, if thermals drive your shrinking-chip strategy, it is much better to hold your frequency constant and increase cores by 1.4× with a net area decrease of 1.4× than it is to increase your frequency by 1.4× and shrink your chip by 2×.

The dim horseman

As exponentially larger fractions of a chip's transistors become dark transistors, silicon area becomes an exponentially cheaper resource relative to power and energy consumption. This shift calls for new architectural techniques that spend area to buy energy efficiency. If we move past unhappy thoughts of shrinking silicon and consider populating dark silicon area with logic that we use only part of the time, then we are led to some interesting new design possibilities.

The term *dim silicon* refers to techniques that put large amounts of otherwise-dark silicon area to productive use by employing heavy underclocking or infrequent use to meet the power budget—that is, the

architecture is strategically managing the chip-wide transistor duty cycle to enforce the overall power constraint.[8,9] Whereas early 90-nm designs such as Cell and Prescott were dimmed because actual power exceeded design-estimated power, we are converging on increasingly more elegant methods that make better trade-offs.

Dim silicon techniques include dynamically varying the frequency with the number of cores being used, scaling up the amount of cache logic, employing near-threshold voltage (NTV) processor designs, and redesigning the architecture to accommodate bursts that temporarily allow the power budget to be exceeded, such as Turbo Boost and computational sprinting.[10]

Turbo Boost 1.0. Although first-generation multicores had a ship-time-determined top frequency that was invariant of the number of currently active cores, Intel's Turbo Boost 1.0 enabled second-generation multicores to make real-time trade-offs between active core count and the frequency the cores ran at: the fewer the cores, the higher the frequency. When Turbo Boost is enabled, it uses the energy gained from turning off cores to increase the voltage and then the frequency of the active cores. This technique, known as dynamic voltage and frequency scaling (DVFS), increases power proportional to the cube of the increase in frequency.

NTV processors. In the past, DVFS was also used to save cubic power when frequencies were decreased. However, today, processor manufacturers operate transistors at reduced voltages—around 2.5× the threshold voltage, an energy-delay optimal point. This point is right at the edge of an operating regime where frequency starts to drop precipitously as voltage is reduced, which makes downward-DVFS much less effective.

Nonetheless, researchers have begun to explore this regime. One recent approach is Near-Threshold Voltage (NTV) logic,[11] which operates transistors in the near-threshold regime slightly above the threshold voltage, providing more palatable trade-offs between energy and delay than subthreshold circuits, for which frequency drops exponentially with voltage decreases. Researchers have explored wide-SIMD NTV processors,[12] which seek to exploit data parallelism, along with NTV many-core processors[13] and an NTV x86 processor.[14]

Although NTV per-processor performance drops faster than the corresponding savings in energy-per-instruction (5× energy improvement for an 8× performance cost), the performance loss can be offset by using 8× more processors in parallel if the workload allows it. Then, an additional 5× processors could turn the energy efficiency gains into additional performance. So, with ideal parallelization, NTV could offer 5× the throughput improvement by absorbing 40× the area. But this would also require 40× more free parallelism in the workload relative to the parallelism consumed by an equivalent energy-limited super-threshold many-core processor.

In practice, for many applications, 40× additional parallelism can be elusive. For chips with large power budgets that can already sustain hundreds of cores, applications that have this much spare parallelism are relatively rare. Interestingly, because of this effect, NTV's applicability across applications increases in low-energy environments because the energy-limited baseline super-threshold design has consumed less of the available parallelism. Furthermore, NTV clearly becomes more applicable for workloads with extremely large amounts of parallelism.

NTV presents several circuit-related challenges that have seen active investigation, especially because technology scaling will exacerbate rather than ameliorate these factors. A significant NTV challenge has been susceptibility to process variability. As operating voltages drop, variation in transistor threshold due to random dopant fluctuation is proportionally higher, and leakage and operating frequency can vary greatly. Because NTV designs can expand the area consumption by approximately 8× or more, variation issues are exacerbated. Other challenges include the penalties involved in designing low-operating voltage static RAMs (SRAMs) and the increased interconnection energy consumption due to greater spreading across cores.

Bigger caches. An often-proposed dim-silicon alternative is to simply allocate otherwise dark silicon area for caches. Because only a

subset of cache transistors (such as a word-line) is accessed each cycle, cache memories have low duty cycles and thus are inherently dark. Compared to general-purpose logic, a level-1 (L1) cache clocked at its maximum frequency can be about 10× darker per square millimeter, and larger caches can be even darker. Thus, adding cache is one way to simultaneously increase performance and lower power density per square millimeter. We can imagine, for instance, expanding per-core cache at a rate that soaks up the remaining dark silicon area: 1.4 to 2× more cache per core per generation. However, many applications do not benefit much from additional cache, and upcoming TSV-integrated DRAM will reduce the cache benefit for those applications that do.

Computational sprinting and Turbo Boost. Other techniques employ "temporal dimness" as opposed to "spatial dimness," temporarily exceeding the nominal thermal budget but relying on thermal capacitance to buffer against temperature increases, and then ramping back to a comparatively dark state. Intel's Turbo Boost 2.0 uses this approach to boost performance up until the processor reaches nominal temperature, relying on the heat sink's innate thermal capacitance. ARM's big.LITTLE employs four A15 cores until the thermal envelope is exceeded (anecdotally, about 10 seconds), then switches over to four lower-energy, lower-performance A7 cores. Computational sprinting carries this a step further, employing phase-change materials that let chips exceed their sustainable thermal budget by an order of magnitude for several seconds, providing a short but substantial computational boost. These modes are especially useful for "race to finish" computations, such as webpage rendering, for which response latency is important, or for which speeding up the transition of both the processor and its support logic to a low-power state reduces energy consumption.

The specialized horseman

The specialized horseman uses dark silicon to implement a host of specialized coprocessors, each either much faster or much more energy efficient (100 to 1,000×) than a general-purpose processor.[1] Execution hops between coprocessors and general-purpose cores, executing where it is most efficient. The unused cores are power- and clock-gated to keep them from consuming precious energy. Unlike dim silicon, which tends to focus on manipulating voltages, frequencies, and duty cycles as ways to manage power, specialized logic focuses on reducing the amount of capacitance that needs to be switched to perform a particular operation.

The promise for a future of widespread specialization is already being realized: we are seeing a proliferation of specialized accelerators that span diverse areas such as baseband processing, graphics, computer vision, and media coding. These accelerators enable orders-of-magnitude improvements in energy efficiency and performance, especially for computations that are highly parallel. Recent proposals have extrapolated this trend and anticipate that the near future will see systems comprising more coprocessors than general-purpose processors.[1,7] This article refers to these systems as coprocessor-dominated architectures, or CoDAs.

As specialization usage grows to combat the dark silicon problem, we are faced with a modern-day specialization "Tower of Babel" crisis that fragments our notion of general-purpose computation and eliminates the traditional clear lines of communication between programmers and software and the underlying hardware. Already, we see the deployment of specialized languages such as CUDA that are not usable between similar architectures (for example, AMD and Nvidia). We see overspecialization problems between accelerators that cause them to become inapplicable to closely related classes of computations (such as double-precision scientific codes running incorrectly on a GPU's non-IEEE-compliant floating-point hardware). Adoption problems are also caused by the excessive costs of programming heterogeneous hardware (such as the slow uptake of Sony PlayStation 3 versus Xbox). Moreover, specialized hardware risks obsolescence as standards are revised (for example, a JPEG standard revision).

Insulating humans from complexity. These factors speak to potential exponential

increases in design, verification, and programming effort for these CoDAs. Combating the Tower of Babel problem requires defining a new paradigm for how specialization is expressed and exploited in future processing systems. We need new scalable architectural schemas that employ pervasively specialized hardware to minimize energy and maximize performance while at the same time insulating the hardware designer and programmer from such systems' underlying complexity.

Overcoming Amdahl-imposed limits on specialization. Amdahl's law provides an additional roadblock for specialization. To save energy across the majority of the computation, we must find broad-based specialization approaches that apply to both regular, parallel code and irregular code. We must also ensure that communicating specialized processors doesn't fritter away their energy savings on costly cross-chip communication or shared-memory accesses.

Recent efforts. The UCSD GreenDroid processor (see Figure 2)[3,15] is one such CoDA-based system that seeks to address both complexity issues and Amdahl limits. GreenDroid is a mobile application processor that implements Android mobile environment hotspots using hundreds of specialized cores called *conservation cores*, or c-cores.[1,9] C-cores, which target both irregular and regular code, are automatically generated from C or C source code, and support a patching mechanism that lets them track software changes. They attain an estimated ~8 to 10× energy-efficiency improvement, at no loss in serial performance, even on nonparallel code, and without any user or programmer intervention.

Unlike NTV processors, c-cores need not find additional parallelism in the workload to cover a serial performance loss. Thus, c-cores are likely to work across a wider range of workloads, including collections of serial programs. However, for highly parallel workloads in which execution time is loosely concentrated, NTV processors might hold an area advantage because of their reconfigurability.

Other specialized processors such as the University of Wisconsin-Madison's DySER[16] and the University of Michigan's Beret[17] propose alternative architectures that exploit specialization like c-cores, but focus on improving reconfigurability at the cost of energy savings. Recent efforts have also examined the use of approximate neural-network-based computing as an elegant way to package programmability, reconfigurability, and specialization.[18]

The "deus ex machina" horseman

Of the four horsemen, this is by far the most unpredictable. "Deus ex machina" refers to a plot device in literature or theater in which the protagonists seem increasingly doomed until the very last moment, when something completely unexpected comes out of nowhere to save the day. For dark silicon, one deus ex machina would be a breakthrough in semiconductor devices. However, as we shall see, the breakthroughs that would be required would have to be quite fundamental—in fact, we most likely would have to build transistors out of devices other than MOSFETs. Why? Because MOSFET leakage is set by fundamental principles of device physics, and is limited to a subthreshold slope of 60 mV/decade at room temperature; this corresponds to a reduction of 10× leakage current for every 60 mV that the threshold voltage is above the *V*ss, which is determined by properties of thermionic emission of carriers across a potential well. Thus, although innovations such as Intel's FinFET/TriGate transistor and high-*K* dielectrics represent significant achievements maintaining a subthreshold slope close to their historical values, they still remain within the scope of the MOSFET-imposed limits and are one-time improvements rather than scalable changes.

Two VLSI candidates that bypass these limits because they are not based on thermal injection are tunnel field-effect transistors (TFETs),[19] which are based on tunneling effects, and nanoelectromechanical system (NEMS) switches,[20] which are based on physical relays. TFETs are reputed to have subthreshold slopes on the order of 30 mV/decade—twice as good as the ideal MOSFET—but with lower on-currents than MOSFETs, limiting their use in high-performance circuits. NEMS devices have essentially a near-zero subthreshold

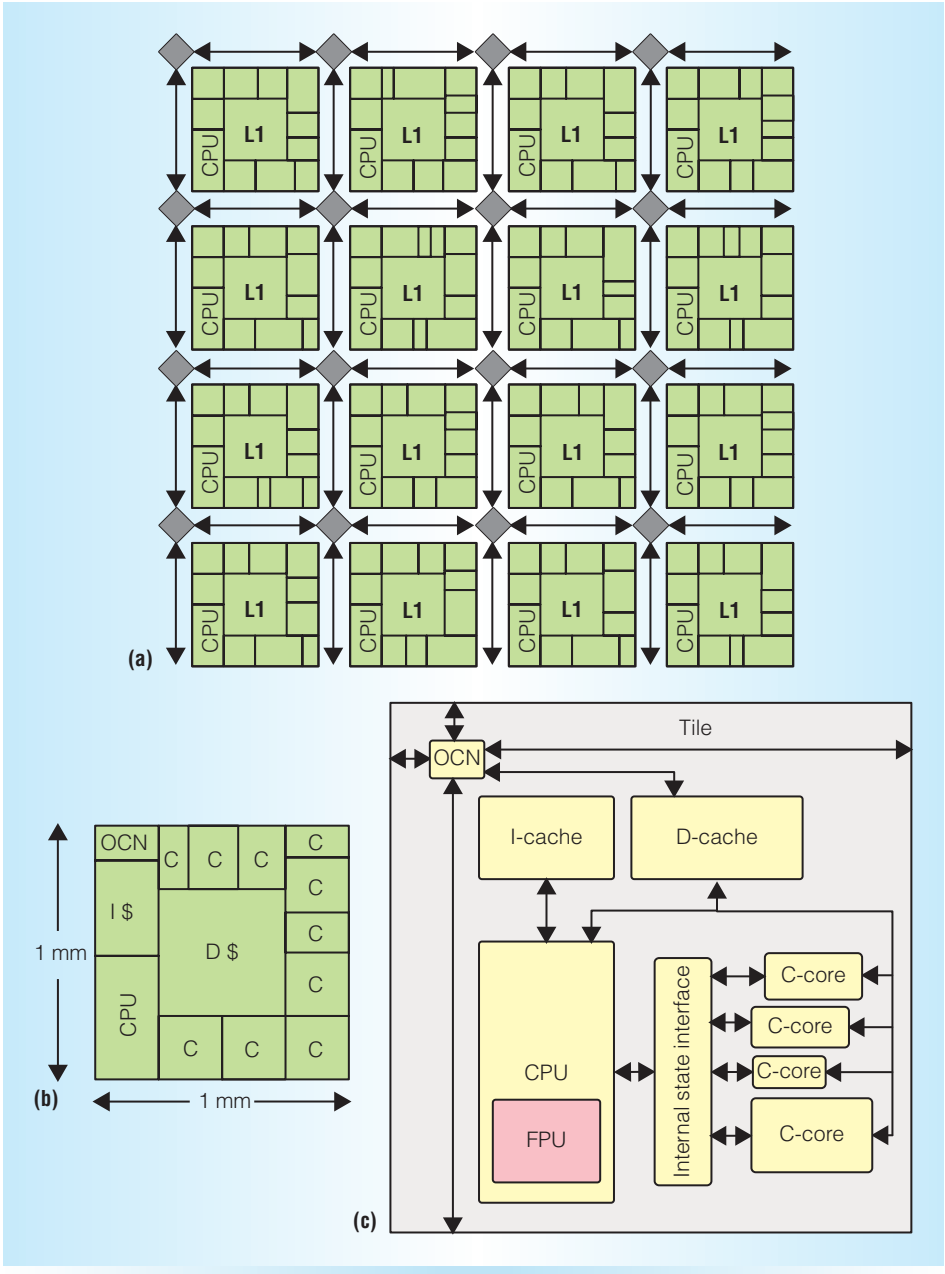

Figure 2. The GreenDroid architecture, an example of a coprocessor-dominated architecture (CoDA). The GreenDroid Mobile Application Processor comprises 16 nonidentical tiles (a). Each tile (b) holds components common to every tile—the CPU, on-chip network (OCN), and shared level-1 (L1) data cache—and provides space for multiple conservation cores, or c-cores, of various sizes. A variety of in-tile networks (c) connect components and c-cores.

slope but slow switching times. Both TFETs and NEMS devices thus hint at orders-of-magnitude improvements in leakage but remain untamed and fall short of being integrated into real chips.

Realizing the importance of the fourth horseman, a recent $194 million DARPA/MARCO STARnet program is funding four centers, each focusing on a key direction for beyond-CMOS approaches: developing electron spin-based memory computation devices (C-SPIN), formulating new information-processing models that can leverage statistical (that is, nondeterministic)

beyond-CMOS devices (SONIC), engineering nonconventional atomic scale engineered materials (FAME), and creating new devices that extend prior work on TFETs to operate at even lower voltages (LEAST).

Evolutionary design principles for dark silicon

While researchers work to mature the new ideas represented by the four horsemen, what principles should guide today's designs that must tackle dark silicon? Listed below is a set of evolutionary, rather than revolutionary, dark silicon design principles that are motivated by changing trade-offs created by dark silicon:

- *Moving to the next generation will provide an automatic 1.4× energy-efficiency increase. Figure out how you will use it.* As a baseline, chip capabilities will scale with energy, whether it is allocated to frequency or more cores. You can increase or decrease frequency or transistor counts, but transistors switched per unit time can increase by only 1.4×.
- *The next generation will create a large amount of dark area. Determine, for your domain, how to trade mostly dark area for energy.* If the die area is fixed, any scaling is going to have a surplus of transistors. Which combination of the four horsemen is most effective in your domain? Should you go dim—more caches? Underclocked arrays of cores? NTV on top of that? Add accelerators or c-cores? Use new kinds of devices? Shrink your chip?
- *Pipelining makes less sense than it used to. Figure out if faster transistor delays will allow you to fit more in a pipeline stage without reducing frequency.* Pipelining increases duty cycle and introduces additional capacitance in circuits (registers, prediction circuits, bypassing, and clock tree fan out), neither of which is dark silicon friendly. Reducing pipeline depth and increasing FO4 depths reduces capacitive overhead. Note, too, that excessive pipelining and frequency exacerbates the gap between processing and memory.
- *Architectural multiplexing and logic sharing are becoming increasingly questionable optimizations. See if they still make sense.* Sharing introduces additional energy consumption because it requires sharers to have longer wires to the shared logic, and it introduces additional performance and energy overheads from the control logic that manages the sharing. For example, architectures that have repositories of nonshared state that share physical pipelines (such as large-scale multithreading) pay large wire capacitances inside these memories to share that state. As area gets cheaper, it will make less sense to pay these overheads, and the degree of sharing will decrease so that the energy cost of pulling state out of these state repositories will be reduced.
- *Multiplexing and RAMs that facilitate sharing of program data are still a good idea. Keep them.* If different threads of control are truly sharing data, multiplexed structures, such as shared RAM, or crossbars, are often still more efficient than coherence protocols or other schemes.
- *Architectural techniques for saving transistors should only be applied if they do not worsen energy efficiency.* Transistors are getting exponentially cheaper, and we can't use them all at once. Why are we trying to save transistors? Locally, transistor-saving optimizations make sense, but an exponential wind is blowing against these optimizations in the long run.
- *Power rails are the new clocks. Design with them in mind.* Ten years ago, it was a big step to move beyond a few clock domains. Now, chips can have hundreds of clock domains, all with their own clock gates. With dark silicon, we will see the same effect with power rails; we will have hundreds and maybe thousands of power rails in the future, all with their own power gates, to manage the leakage for the many heterogeneous system components.
- *Heterogeneity results from the shift from a 1D objective function (performance) to a 2D objective function (performance and energy). Design with the shape of this function in mind.* The past lacked in

heterogeneity, because designs were largely measured according to a single axis—performance. To first order, there was a single optimal design point. Now that performance and energy are both important, a Pareto curve trades off performance and energy, and there is no one optimal design across that curve; there are many optimal points. Optimal designs will incorporate several such points across these curves.

These rules of thumb will guide our existing designs along an evolutionary path to become increasingly dark silicon friendly—but what then of more revolutionary approaches?

Insights from the brain: a dark technology

Perhaps one promising indicator that low-duty cycle, "dark technology" can be mastered, unlocking new application domains, is the efficiency and density of the human brain. The brain, even today, can perform many tasks that computers cannot, especially vision-related tasks. With 80 billion neurons and 100 trillion synapses operating at less than 100 mV, the brain embodies an existence proof of highly parallel, reliable, and dark operation, and embodies three of the horsemen—dim, specialized, and deus ex machina. Neurons operate with extremely low-duty cycles compared to processors—at best, 1 kilohertz. Although computing with silicon-simulated neurons introduces excessive "interpretive" overheads—neurons and transistors have fundamentally different properties—the brain can offer us insight and long-term ideas about how we can redesign systems for the extremely low-duty cycles and low voltages called for by dark silicon. Here are some of these properties, which may give us insight on more revolutionary extensions to the evolutionary principles proposed in the last section:

- *Specialization.* As with the specialized horseman, different groups of neurons serve different functions in cognitive processing, connect to different sensory organs, and allow reconfiguration, evolving with time synaptic connections customized to the computation.
- *Very dark operation.* Neurons fire at a maximum rate of approximately 1,000 switches per second. Compare this to arithmetic logic unit (ALU) transistors that toggle at three billion times per second. The most active neuron's activity is a millionth of that of processing transistors in today's processors.
- *Low-voltage operation.* Brain cells operate at approximately 100 mV, yielding CV^2 energy savings of 100× versus 1-V operation, in a clear parallel to the dim horseman's NTV circuits. Communication is low swing and low voltage, saving large amounts of energy.
- *Limited sharing and memory multiplexing.* Any given neuron can switch only 1,000 times per second, by definition, so it must have extremely limited sharing, because a point of multiplexing would be a bottleneck in parallel processing. The human visual system starts with 6M cones in the retina, similar to a 2-megapixel display, processes it with local neurons, and then sends it on the 1M-neuron optic nerve to the visual cortex. There is no central memory store; each pixel has a set of its own ALUs, so to speak, so energy waste due to multiplexing is minimal.
- *Data decimation.* The human brain reduces the data size at each step and operates on concise but approximate representations. If using 2 megapixels suffices to handle color-related vision tasks, why use more than that? Larger sensors would just require more neurons to store and compute on the data. We should ensure that we are processing no more data than necessary to achieve the final outcome.
- *Analog operation.* The neuron performs a more complex basic operation than the typical digital transistor. On the input side, neurons combine information from many other neurons; and on the output, despite producing rail-to-rail digital pulses, encode multiple bits of information via spikes timings. Could this suggest that there are more efficient ways to map operations onto silicon-based technologies? In RF wireless front-end communications, analog processing enables computations that would be impossible to do at speed

digitally. However, analog techniques might not scale well to deep nanometer technology.

- *Fast, static, "gather, reduce, and broadcast" operators.* Neurons have fan out and fan in of approximately 7,000 to other neurons that are located significant distances away. Effectively, they can perform efficient operations that combine vector-style gather memory accesses to large numbers of static-memory locations, with a vector-style reduction operator and a broadcast. Do more efficient ways exist for implementing these operations in silicon? It could be useful for computations that operate on finite-sized static graphs.

Recently, both the EU and US governments have proposed initiatives to enable greater studies of the computational capabilities of the brain. Although brain-inspired computing has already come and gone several times in the brief history of manmade computers, dark silicon may cause these approaches to become increasingly relevant.

Although silicon is getting darker, for researchers the future is bright and exciting. Dark silicon will cause a transformation of the computational stack and provide many opportunities for investigation. MICRO

Acknowledgments

This work was partially supported by NSF awards 0846152, 1018850, 0811794, and 1228992, Nokia and AMD gifts, and by STARnet, an SRC program sponsored by MARCO and DARPA. I thank the anonymous reviewers for their valuable insights and suggestions.

References

1. G. Venkatesh et al., ''Conservation Cores: Reducing the Energy of Mature Computations,'' *Proc. 15th Architectural Support for Programming Languages and Operating Systems Conf.,* ACM, 2010, pp. 205-218.
2. R. Merrit, ''ARM CTO: Power Surge Could Create 'Dark Silicon,''' *EE Times,* 22 Oct. 2009.
3. N. Goulding et al., ''GreenDroid: A Mobile Application Processor for a Future of Dark Silicon,'' Hot Chips Symp., 2010.
4. M. Taylor, ''Is Dark Silicon Useful? Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse,'' *Proc. 49th Ann. Design Automation Conf.* (DAC 12), ACM, 2012, pp. 1131-1136.
5. R.H. Dennard, ''Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions,'' *IEEE J. Solid-State Circuits,* vol. SC-9, 1974, pp. 256-268.
6. H. Esmaeilzadeh et al., ''Dark Silicon and the End of Multicore Scaling,'' *ACM SIGARCH Computer Architecture News,* vol. 39, no. 3, 2011, pp. 365-376.
7. N. Hardavellas et al., ''Toward Dark Silicon in Servers,'' *IEEE Micro,* vol. 31, no. 4, 2011, pp. 6-15.
8. W. Huang et al., ''Scaling with Design Constraints: Predicting the Future of Big Chips,'' *IEEE Micro,* vol. 31, no. 4, 2011, pp. 16-29.
9. J. Sampson et al., ''Efficient Complex Operators for Irregular Codes,'' *Proc. 17th Int'l Symp. High Performance Computer Architecture* (HPCA 11), IEEE CS, 2011, pp. 491-502.
10. A. Raghavan et al., ''Computational Sprinting,'' *Proc. IEEE 18th Int'l Symp. High-Performance Computer Architecture* (HPCA 12), IEEE CS, 2012, doi:10.1109/HPCA.2012.6169031.
11. R. Dreslinski et al., ''Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits,'' *Proc. IEEE,* vol. 98, no. 2, 2010, pp. 253-266.
12. E. Krimer et al., ''Synctium: A Near-Threshold Stream Processor for Energy-Constrained Parallel Applications,'' *IEEE Computer Architecture Letters,* Jan. 2010, pp. 21-24.
13. D. Fick et al., ''Centip3de: A 3930 DMIPS/W Configurable Near-Threshold 3D Stacked System with 64 ARM Cortex-M3 Cores,'' *Proc. IEEE Int'l Solid-State Circuits Conf.,* IEEE, 2012, pp. 190-192.
14. S. Jain et al., ''A 280 mV-to-1.2 V Wide-Operating-Range IA-32 Processor in 32 nm CMOS,'' *Proc. IEEE Int'l Solid-State Circuits Conf.,* IEEE, 2012, pp. 66-68.
15. N. Goulding-Hotta et al., ''The GreenDroid Mobile Application Processor: An Architecture for Silicon's Dark Future,'' *IEEE Micro,* vol. 31, no. 2, 2011, pp. 86-95.

16. V. Govindaraju, C.-H. Ho, and K. Sankaralingam, ''Dynamically Specialized Datapaths for Energy Efficient Computing,'' *Proc. IEEE 17th Int'l Symp. High-Performance Computer Architecture* (HPCA 11), IEEE CS, 2011, doi:10.1109/HPCA.2011.5749755.
17. S. Gupta et al., ''Bundled Execution of Recurring Traces for Energy-Efficient General Purpose Processing,'' *Proc. 44th Ann. IEEE/ACM Int'l Symp. Microarchitecture,* ACM, 2011, pp. 12-23.
18. H. Esmaeilzadeh et al., ''Neural Acceleration for General-Purpose Approximate Programs,'' *Proc. 45th Ann. IEEE/ACM Int'l Symp. Microarchitecture,* IEEE CS, 2012, pp. 449-460.
19. A. Ionescu et al., ''Tunnel Field-Effect Transistors as Energy-Efficient Electronic Switches,'' *Nature,* 17 Nov. 2011, pp. 329-337.
20. F. Chen et al., ''Demonstration of Integrated Micro-Electro-Mechanical Switch Circuits for VLSI Applications,'' *Proc. IEEE Int'l Solid-State Circuits Conf.,* IEEE, 2010, pp. 150-151.

Michael B. Taylor is an associate professor in the Department of Computer Science and Engineering at the University of California, San Diego, where he leads the Center for Dark Silicon. His research interests include dark silicon, chip design, parallelization tools, and Bitcoin computing systems. Taylor has a PhD in electrical engineering and computer science from the Massachusetts Institute of Technology.

Direct questions and comments about this article to Michael B. Taylor, 9500 Gilman Drive, MC 0404 EBU 3B 3202, La Jolla, CA 92093-0404; mbtaylor@ucsd.edu.