Continuing the Scaling of Digital Computing Post Moore's Law

George Michelogiannakis, John Shalf, David Donofrio, John Bachan
Lawrence Berkeley National Laboratory
{mihelog,jshalf,ddonofrio,jdbachan }@lbl.gov

April 2016

Introduction

Computer devices are nearing the limits of scaling traditional CMOS technology that up to this day follows Gordon Moore's predictions for doubling transistor density every 24 months. In fact, CMOS scaling is now predicted to first slow down and eventually cease by the middle of the next decade (John Shalf, 2014) (Mack, 2011), with industry forecasting that technologies beyond 2 or 3nm may be infeasible or impractical due to cost, manufacturability, and the size of molecules which acts as a hard lower limit (Haron, 2008; Dhar, 2011). This approaching end of lithographic scaling does not mean the end of performance scaling for digital computing, but an invitation to identify novel methods to continue this scaling. Performance scaling will need to come with energy and cost reductions; without the density scaling they need to originate from novel architectures, technologies (devices and materials), or computational models.

The DOE has come to depend on the rapid, predictable, and cheap scaling of computing performance to meet mission needs for scientific theory, large scale experiments, and national security. For decades exponentially increasing capability could be procured at roughly constant annual cost, and that expectation has permeated computing resource planning and decision making. Society more generally has come to expect and rely upon the benefits provided by Moore's Law for consumer electronics and data centers (Lance Joneckis, 2014) (Larus, 2008). The deeper issue presented by these changes is the threat to DOE's mission and to the future economic growth of the U.S. computing industry and to society as a whole.

With the impending end of Moore's law, it is imperative for the Office of Advanced Scientific Computing Research (ASCR) to develop a research agenda to assess the viability of novel semiconductor technologies and navigate the ensuing challenges. DOE has a large investment in computing facilities and ever-growing computational demands driven by science questions. Therefore, it is incumbent on DOE to understand how future technology trends will limit performance growth, but also take a leadership role in future computing advances on behalf of science applications. The DOE HPC community is a unique national resource that understands complex scientific problems and how they map to hardware. Thus, it can both guide and understand future technology impacts to preserve performance scaling for scientific applications, as well as assess the impact of such impacts to algorithms and programming models. The investments in post Moore digital will benefit from a partnership with the DOE materials science program to develop advanced electronic materials and characterize them in DOE's advanced light sources, but will also leverage ASCR's traditional strengths in programming models and applications.

Investments in digital computing need to co-exist alongside research into new forms of computing such as neuromorphic and quantum. However, both neuromorphic and quantum computing models apply to specific domains of problems that do not include a wide variety of critical applications to DOE (John Shalf, 2014) (Aaronson, 2008) (Alán Aspuru-Guzik, 2005). As no single, general-purpose computational alternative has risen, combined with the significant investments already made to

CMOS (digital computing), a recent IDA-DARPA report concluded that CMOS is expected to remain dominant well after Moore's law ends (Lance Joneckis, 2014).

The Path Forward for Digital Computing

To address these challenges, we have identified the following areas and proposed research directions for ASCR as potential avenues for continued performance scaling of digital computing to meet DOE mission needs and to benefit society as a whole. Figure 3 provides an illustration of these directions:

- Architecture: Architecture refers to novel designs of any component of an HPC system, whether that is on-chip or off-chip. The goal of these architectures is to remove overheads in current designs, as well as offer hardware and thus more efficient support for important functionality such as security and resiliency.
- Devices and CMOS technologies: While scaling transistor density following Moore's law has historically resulted in lower energy use, there are multiple alternative technologies that fall under the same digital computing model as MOSFETs and promise more efficient operation and thus lower energy (e.g. CNFETs, TFETs, photonics, etc.). However, we need to both understand the impact of such devices to the architecture and software, as well as speed up their adoption by addressing some of their challenges.
- Advanced manufacturing techniques: Novel manufacturing methods, including 3D stacking, are increasing density enabling memory layers on top of logic layers, and even multiple memory and logic layers interleaved. This radical change challenges assumptions embedded in current architectures and software that large memories must be located relatively far – in both space and latency from processing elements..
- Programming models: In addition, we need to examine if the forthcoming changes by the above three thrusts affect how application designers interact with the machine. Existing programming models are designed with old architectures in mind. New programming models and runtimes are necessary that expose the fundamental changes in relative costs of each operation, adapt to new realities such as new location, size, and type of main memory, as well as break abstraction barriers such that the heterogeneity of future machines can be both exposed and exploited.

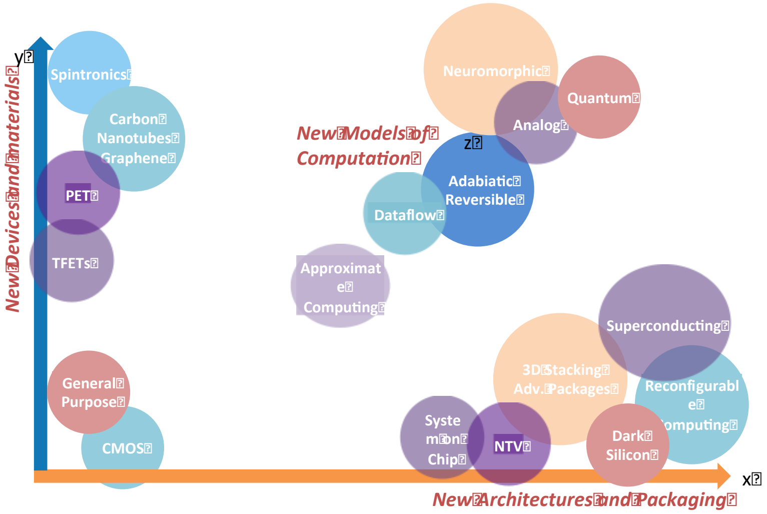

Figure 1: This figure illustrates the different thrust areas to continue digital computing scaling. The Y-axis shows that performance scaling can continue with more efficient and novel devices. The X-axis shows architectures and 3D stacking as another thrust area to also continue scaling. The Z-axis shows alternative models of computation, which themselves may rely on development of novel devices since the end of lithographic scaling affects the ability to scale ANY device. A balanced research program should enable progress along each of these axes.

We believe DOE is well positioned to lead research to define the future path for digital electronics, and can leverage existing expertise in the above four areas from the national laboratories. While there are expected to be parallel efforts for example from industry, those efforts are unlikely to focus on applications that are important to DOE. Novel devices and 3D stacking areas would involve a close partnership between ASCR and the DOE materials science program in Basic Energy Sciences (BES) to develop advanced electronic materials and characterize them in DOE's advanced light sources. The thrust on programming models can leverage ASCRs traditional strengths in programming models and applications.

The remainder of this document provides a deeper dive into the details for each of the identified thrust areas.

Architecture

We have previously witnessed upheavals in computer architecture that have had profound impacts across the industry. Most recently, we have seen on-chip power density limitations bring about a leveling-off of clock frequency that in turn dragged down the continued scaling of single threaded performance. The industry responded to this new design constraint by scaling performance through on-chip parallelism and broadening the use of once specialized accelerators, such as GPUs. However, even as power emerged as the primary design constraint, the difficulty of making radical changes left many fundamental architectural choices intact.

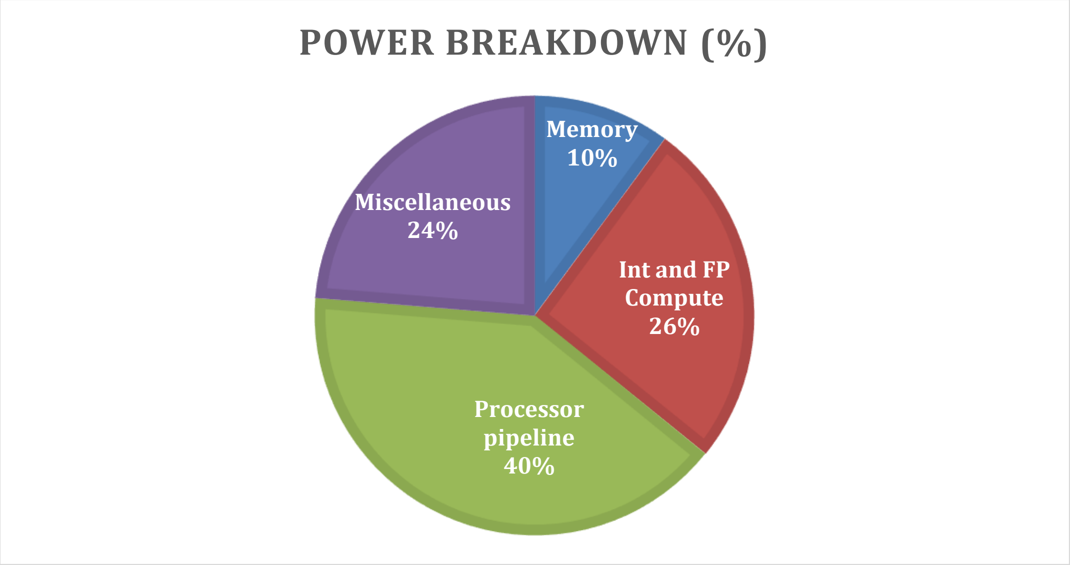

Figure 2: Current general-purpose architectures spend the minority of energy (roughly 40% in this example which includes integer and floating point computation as well as memory access) on actual computation, and the majority on other overhead such as functionality to ease programmability which includes functionality in the processor pipeline such as register renaming, fetch, decode, and scheduling (Cong J. a., 2014). This opens the door for efficiency gains from specialized architectures.

Specialization

Legacy architectural features have remained intact in the face of shifting design constraints. As a result, many current general-purpose architectures still spend the majority of their energy on non-computational operations as shown in Figure 2. Note that this figure focuses on a single core and does not include data movement to other cores, which dominates compute power even for short on-chip distances such as 2mm (John Shalf, 2014). Removing excess overhead leads towards specialized architectures, which take the form of on-chip accelerators or co-processors (Mittal, 2015). These specialized logic blocks may perform simple computations such as convolutions or more complete image processing or network protocols. The specialization removes part some of the performance and energy overhead that a general-purpose core would require for the same computation. This has already been realized by part of the community, leading to an increasing number of commercial large-scale systems with accelerators, co-processors, or graphical processing units (GPUs) acting as accelerators, as figure 3 shows. A recent study (figure 4) demonstrated speedup increases up to 28.6x and energy reduction of 78.4% for specialized architectures implemented in an FPGA, compared to a 12-core Xeon E5 CMP.

Prior to the end of the semiconductor industry lithographic roadmap, denser fabrication technologies will provide more chip area than can be utilized simultaneously due to power constraints. This inability to utilize 100% of the available area simultaneously is a logical extension to the current Dynamic Voltage and Frequency Scaling (DVFS) seen today and is often collectively referred to as "Dark Silicon." It is possible specialized accelerators will see an additional boost as a method of effectively utilizing this additional silicon acreage.

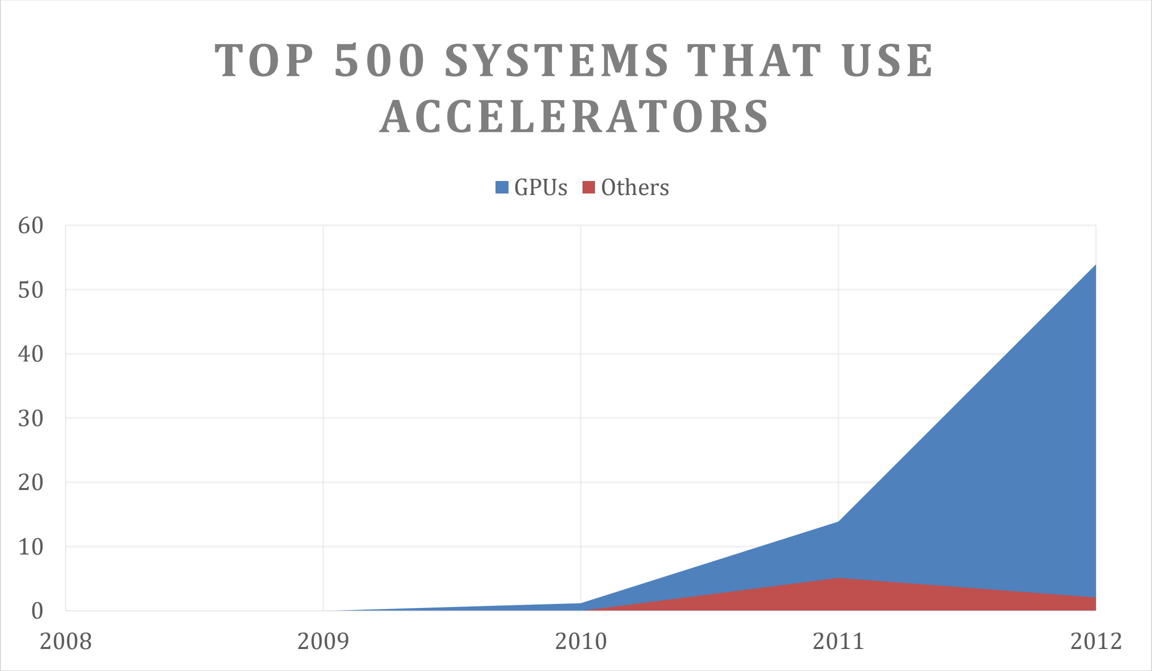

Figure 3: An increasing number of top 500 systems include GPUs or non-GPU accelerators (Intel MIC, Clearspeed CSX600, IBM PowerXCell 8i in this study) to perform specialized computations more efficiently (Cong J. a., 2014).

This trend towards specialization is important for removing overhead in general-purpose architectures and should continue in parallel with research focused on improved general-purposes cores. The development and application of these specialized architectures to HPC-focused problems is an important area of research and will require insight and investigation into not only the hardware microarchitecture but also algorithms and applications that can benefit from hardware specialization.

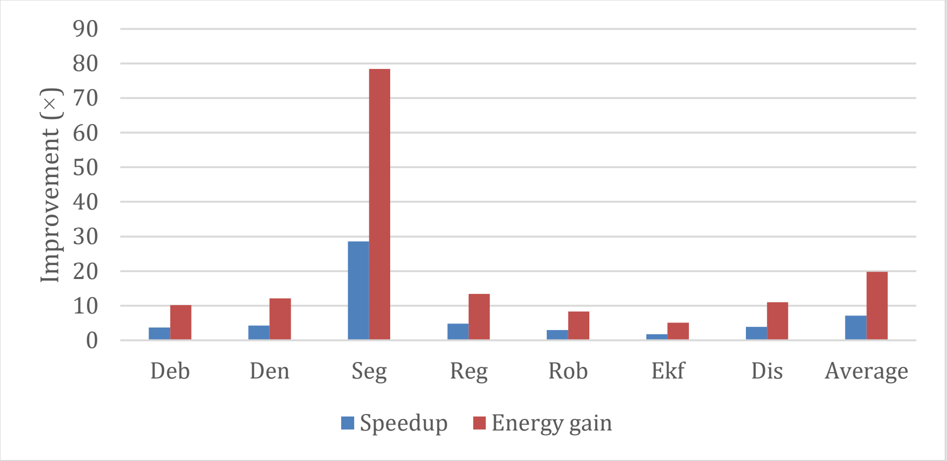

Figure 4: Speedup and energy improvement (in multiplication factors) of specialized architectures for medical imaging. Specialized architectures were implemented in a FPGA and compared against a 12-core Intel Xeon E5 CMP.

Accelerators and specialized designs are not constrained only to computation and cores. There is opportunity for optimization of on-chip networks, memory controllers, prefetchers, and other components that support common patterns in HPC applications efficiently. For instance, memory accelerators can detect and implement typical memory access patterns of different classes of applications, such as data-parallel algorithms, physics simulations, etc. While this discussion focuses on adding extra circuitry to existing multi-core chip multiprocessors, we should also explore the option of dedicated-function processor chips on the same boards as general-purpose processors. Later in this report we will discuss the need for advanced programming models that can help software developers effectively manage an increasing level of heterogeneity.

One class of specialized accelerators that is gaining significant traction is FPGA-based computing. FPGAs are composed of a large array of simple, configurable logic blocks that can be dynamically re-configured. FPGAs have continued to enjoy clock frequency scaling with each new process generation and FPGA hardware is maturing quickly, with powerful devices capable of > 10 TFlops and 5 Tb/s of IO bandwidth are on the horizon. Once relegated to the world of ASIC prototyping and signal processing, FPGAs are used now alongside ASICs as auxiliary compute engines – including being integrated into future Intel Xeon Processors. While applications ported to FPGAs can see speedups of several orders of magnitude compared to using traditional x86 style cores the tradeoff for this performance has traditionally been the long development time for application porting. Advances in tools enabling higher-level programming are beginning to address this issue and are pushing FPGA computing into the mainstream. A good example is Microsoft's FPGA accelerated node, known as Catapult, to accelerate Bing search that has provided a 2x speedup at lower cost and power consumption. This FPGA based acceleration systems maintains a high degree of programmability and flexibility as it is composed of specialized soft-cores (cores that are synthesized from hardware description languages) rather than large blocks of specialized logic. FPGAs are a great way to enable easy and ubiquitous specialization because FPGA devices are inexpensive and writing hardware description language code to implement specialized logic in them requires less effort than manufacturing chips with the same specialized logic.

To understand how far specialization can go to improve performance it is possible to create a full-custom architecture for specific computations that eschews programmability. Such a design would approach the upper limit of performance per unit power for a particular algorithm and provide a basis for comparison against programmable architectures. Contemporary examples are the Anton systems that represent an implementation where general-purpose programmability was sacrificed for raw performance and for molecular dynamics codes achieved *two orders of magnitude* speedup over contemporary leadership class HPC systems.

While we cannot identify the upper limit of computational efficiency that we can ever achieve with specialized architectures because we cannot possibly explore all possible designs, designing full-custom architectures for specific computations that eschew programmability will show us that the upper limit is at least as good as these architectures, because any modification to increase programmability to make this solution usable will only increase power overheads. In other words, these full-custom architectures will show us what performance efficiency is possible by sacrificing programmability, which will serve as a practical upper limit we can strive to reach with architecture optimizations of programmable architectures.

We expect to identify a significant gap between current architectures and specialized designs. This is a critical study to guide the community in how much effort is valuable to spend in architecture. In addition, it will also guide our research because establishing the ideal architecture and then adding just enough functionality to make it programmable is a top-bottom design approach that also starts from a clean state and thus avoids common overheads of existing architectures.

Approximate and Reconfigurable Computing

Support for approximate computing (Trancoso, 2014) also has significant potential given that many applications relevant to DOE belong in the domain of applications that can tolerate errors with no noticeable impact, and the performance and error tolerance benefits approximate computing has demonstrated. Specifically, approximate computing has demonstrated an up to 20x energy reduction in imaging applications with very little reduction in output quality (Swagath Venkataramani, 2015). Approximate computing is a promising approach not only because of its performance and gains, but because it can be a natural fit for some future manufacturing technologies and devices with larger error rates than existing MOSFET technology.

Support for approximate computing involves both the software and hardware, but promises to simplify large parts of each. Not only do both software and hardware need to perform less strict error checking, but the hardware can tolerate more manufacturing defects that can be used to reduce manufacturing costs especially in future denser technologies and new devices. However, the DOE needs to carefully consider which applications can tolerate error and to what degree. This will be combined with new devices and manufacturing technologies on the hardware side to produce appropriate approximate computing architectures.

Reconfigurable computing has also gained traction in domains including HPC (Wim Vanderbauwhede, 2013). Reconfigurable computing provides hardware that dynamically adjusts its functionality according to the application or other run-time inputs, often using FPGAs in order to take advantage of their reconfiguration capabilities. Reconfigurable computing can be combined with specialized architectures such as to provide the functionality that the application needs at runtime. This way, one reconfigurable device is needed instead of potentially an array of specialized architectures (accelerators), each for a different application. However, performing efficient and accurate dynamic reconfiguration is a challenge.

Using Architecture To Alleviate Challenges of New Devices and Technologies

New devices and manufacturing technologies each come with a new set of challenges. We can use advanced and novel architectures to alleviate some of those challenges such as to increase the adoptability of new devices. In addition, we can use advanced architectures to deal with problems that are currently present and are expects to become more significant in future machines. Examples include resiliency, security, and debug-ability.

Resiliency is a prime example because it is expected to become more important at future compute scales, but also because some new devices are more prone to both soft and hard errors. Thus, inherent microarchitecture support for resiliency in the form of error detection and correction will remove today's software overheads that would prove prohibitive in scales necessary for exascale computing that projects a failure in the order of a few minutes (Snir, 2013) (Hensbergen, 2013). We can use this resiliency support to not expose the software to additional complexity because of the increased numbers of errors, but also to reduce performance and energy penalties of handling errors. In addition, architectural support for security, in the form of hardware-based encryption, permission checking, and isolation, promises to remove further software overheads that can impede continued performance scaling. Security is expected to be a critical issue to a large number of computational domains. Hardware architecture support is an important direction that has repeatedly demonstrated large benefits, such as for large-scale error-free floating-point computations (Demmel, 2013) (George Michelogiannakis, 2015).

Technological advancements have demonstrated that they can be flexible in the face of new constraints and the end of Moore's law presents an opportunity for architectural innovation to address challenges in power consumption, resiliency, and security, all while increasing performance. By starting research now DOE can anticipate and influence post-Moore architectural trends.

Devices and CMOS Technologies

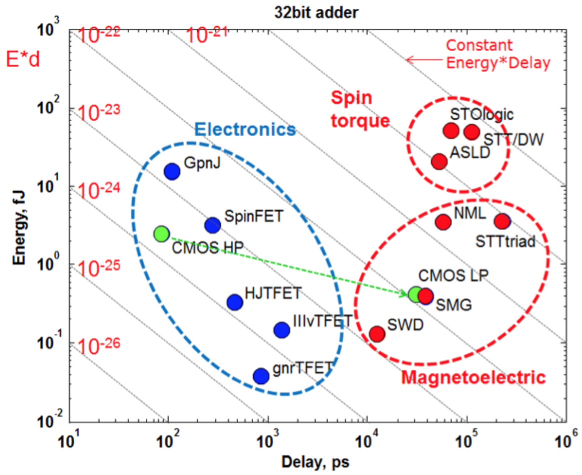

Figure 5: A variety of alternative technologies that conform to the digital computing model are in progress. This figure shows many alternatives compared to CMOS high performance (HP) and low power (LP). The figure shows the energy-delay product. In this figure, lower and to the left are better (Nikonov, 2013).

While CMOS scaling is affected by the slowdown of Moore's law, there are a number of alternative technologies that are still emerging and promise to keep digital computing performance over cost (in Watts or $) scaling without changing the computational model from classical digital computing as we have it today. With these devices we can preserve digital computing scaling as we have up to this point. Therefore, it is a fallacy to consider CMOS scaling slowdown as a slowdown for digital computing performance scaling as well. Figure 5 shows a variety of alternative technologies emerging today, and how they compare with CMOS. A survey and layouts of the alternative devices shown in the figure can be found in (Nikonov, 2013).

The future of most of these devices is encouraging, with recent projections stating significant movement towards lower energy-delay products (EDPs) that is shown in the lower left of the figure, which is exactly what is necessary for continued performance scaling. However, harvesting those benefits is not automatic as it used to be with Moore's law because new devices present different tradeoffs, strengths, and weakness that the architecture and potentially software may have to adapt to. This necessitates significant research that DOE is well-versed to perform in order to accelerate applications critical to DOE. Parallel efforts from other entities, such as industry, are unlikely to focus on the scientific applications that are important to DOE.

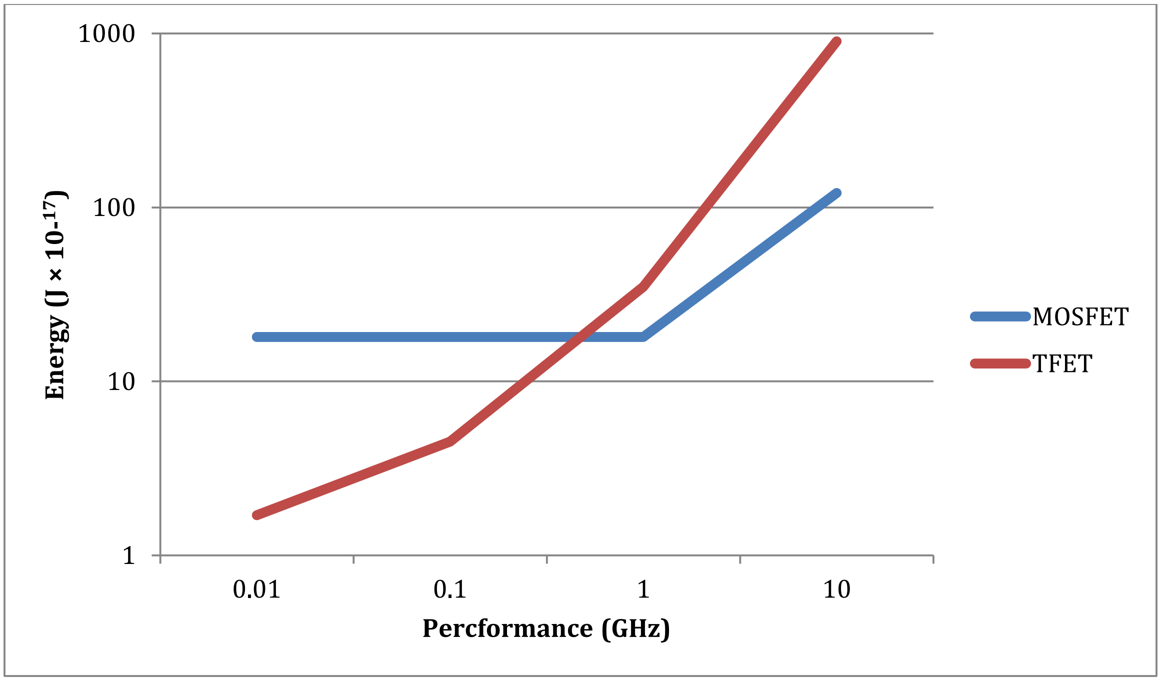

Figure 6: The tunnel field-effect transistor reduces switching energy by an order of magnitude compared to the MOSFET, but only at low clock rates.

To begin with, we need to understand the impact of these technologies, as well as implications for the programming models and other parts of the system (software or hardware). Even though the new devices we consider fit the traditional digital computing model, they may still have large impacts to the software such as runtimes due to their impact to their architecture.

Let us consider the tunnel field-effect transistor (TFET), which is one of the primary candidates for replacing MOSFETs. Figure 6 shows the performance-energy tradeoff it presents compared to the MOSFET. As shown, the TFET can provide an order of magnitude reduction in energy, but only at low clock frequencies. TFETs can achieve higher frequencies but at higher energy. Therefore, to use the TFET to increase performance, we need to tradeoff the energy reduction for increased parallelism. Therefore, we need an order of magnitude more parallelism than today. This will have an important impact to the rest of the system architecture, as well as software. We need to delve more into this observation, quantify the impacts, and propose solutions to these challenges so we can harvest the savings the TFET promises.

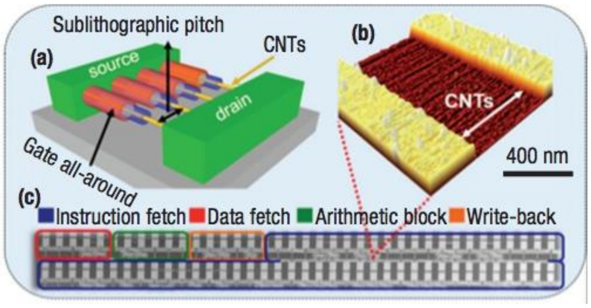

Figure 7: Carbon nanotube transistors (CNFETs) consist of multiple carbon nanotubes (CNs) and follow the conventional digital transistor model (Aly, et al., 2015).

Two other promising device technologies are carbon nanotube transistors (CNFETs (Aly, et al., 2015)) and negative capacitance transistors (Lee, et al., 2013). Negative capacitance transistors can also operate at lower voltage levels, similar to TFETs but with different tradeoffs. CNFETs (figure 7) have excellent electrical, thermal and mechanical properties. In fact, a recently fabricated chip with 2 million CNFETs demonstrated a 1000x improvement of the energy-delay product (EDP) for memory-bound applications, and 10x EDP improvement for compute-bound applications. Even though the maturity of this technology today is sufficient for these impressive gains, the future of CNFETs holds further promise by scaling to billions of transistors early in the next decade. In addition, these impressive gains were attained simply with a general-purpose architecture. This shows the promise of architecture specialization with new devices in preserving digital computing performance scaling, because significant gains from specialization will be combined with large gains from the new devices and technologies.

Different technologies present different tradeoffs. For example, nanophotonics and spintronics promise to alleviate the growing cost of data movement in the on-chip environment. The same is true for optics, which is a related technology. By using one of these options to make data movement cheaper (which directly contradicts observations that have led research in recent years), we need to re-visit architectural changes and re-direct future research to the right path.

Ultimately, the community is in need of a comprehensive study of the impact each technology will have, compared against the promise it holds. This is important for eventually choosing an alternative technology when MOSFET technology stops scaling. Currently, the community is flooded with multiple options having considerably different tradeoffs but little guidance on what to choose and why. This study needs to consider the HPC context, scales of computation for exascale computing, as well as the applications that are important to DOE. Thus, it is important for DOE to have a assume a leading role.

In addition to evaluating the impact of future technologies, we need to understand resiliency and manufacturing challenges that each technology introduces. This study will also act as a predictor to manufacturing cost. As a next step, we can use architecture to alleviate any challenges new technologies introduce (as a tradeoff for better performance efficiency). For instance, some of the architectural solutions to resiliency that we discussed in the previous section could be used to reduce the error rate of new technologies with lower reliability, which could otherwise hinder adoption. This study needs to be conducted in collaboration with the material sciences and VLSI communities.

Packaging and 3D Stacking

3D stacking is projected to have high impact in future computing and HPC. Today's options are mostly limited around 3D-stacked memory (Seth H Pugsley, 2014). Such technologies already show large increases in bandwidth with modest decreases in power and potentially a reduction in latency.

Although these technologies have been available for a few years, we do not yet possess an understanding of the broader-scale performance impact of 3D stacked memory or how to maximize the benefit. Initial studies report that the hybrid memory cube (HMC) does not maximize its performance with a sequential access stream that follows memory address order (as DRAM did). When we understand how HMC's performance is affected by different access patterns, we can re-design applications and the rest of the microarchitecture to present more optimized access patterns. This is important given that memory bandwidth and power are already primary constraints towards continued system-wide performance scaling and increased parallelism (Brian Rogers, 2009).

3D stacking can also have substantial impact to processing in memory (PIM). PIM has been recently proposed to mitigate increasing data movement costs, and relies on memory controllers to perform simple computations and in some cases filter the data (Ignatowski, 2013). The HPC community needs to develop both architectural and software supports to make use of PIM for HPC systems. This can also motivate novel approaches, such as combining accelerators with PIM to perform specialized functions as previously described, but located next to main memory in order to eliminate data movement. Similar concepts can be expanded to networks, including off-chip (system-wide). Such networks can perform computation in routers to avoid interrupting processors as well as reduce contention. This can be a major boost to collective operations such as reduction trees.

In the context of 3D stacked memory, PIM has higher potential than simple arithmetic operations at the memory controller. Recent 3D stacked memories including the HMC include a component of logic at each memory layer along with a large logic base, shown in figure 8. Both can be used to support more sophisticated PIM functionality than is found today, such as complicated accelerators. The logic components at each logic layer are particularly promising because they can access data without any vertical data movement. Thus, we can explore popular patterns such as tree-like communication patterns and how they map to 3D stacked memories. These will be presented to applications as accelerators.

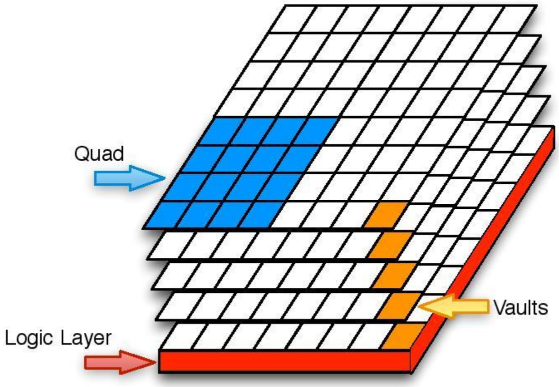

Figure 8: The HMC consists a logic base that is responsible for scheduling memory accesses (in red), as well as scheduling and access logic at every memory layer referred to as vaults, shown in orange.

3D stacked memory is just the beginning of 3D stacking. Future manufacturing processes promise multiple levels of memory interleaved with multiple levels of logic, as shown in figure 9. Such a setting presents a shift to state-of-the-art architecture assumptions, which programming models and applications have been built around. For instance, with interleaved 3D stacked logic and memory, accessing memory is no longer limited by chip pins, dramatically increasing memory bandwidth and decreasing power. In addition, each processor chip can now have hundreds of GBs of memory right on the same die, which was previously impossible because large memory resided far away. In fact, the bandwidth between layers of the same chip is expected to further increase than what is feasible today due to denser VIAs (inter-layer connections). This, combined with an increased number of memory controllers, is expected to both increase memory bandwidth and also de-sensitize memory performance with the access patterns; both aspects break fundamental assumptions about main memory performance today. Furthermore, the number of cores in the same processor chip will now skyrocket from hundreds or thousands with 2D logic, to multiple times that. This is combined with different tradeoffs we need to understand regarding data movement in the horizontal versus vertical axes.

Furthermore, there are multiple types of memory available with different tradeoffs. For instance, compared to traditional DRAM, NVRAM requires more energy to write new data. What's more, new memory technologies such as resistive RAM (Akinaga & Shima, 2010) and magnetic RAM (Dason, Kumar, & Kirubaraj, 2011) have shown fast access to large memory arrays, while at the same time providing *non-volatile* storage (data are not erased when the RAM is powered down). Such large non-volatile memory on top of every processor threatens to change our view of multiple levels of memory to just one that satisfies the needs for low-latency access, large capacities, high bandwidth, and non-volatile storage.

All these changes in the architecture have to be reflected in the software in order to be fully taken advantage of. Therefore, we need to investigate how current applications will operate in this environment, and identify necessary changes to both the algorithms and programming models. We expect certain algorithms to not experience a benefit, others to see improvements, and a third class to only experience benefits with modifications. For instance, new data placement algorithms will be necessary system-wide because data reside on the same die with processors, not in independent remote locations. This presents a multi-constraint problem in case multiple processors access the same data.

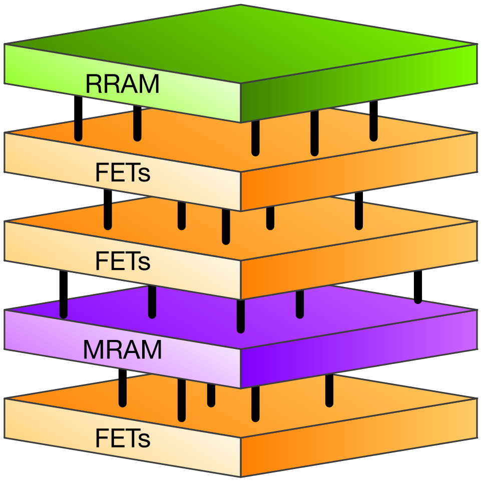

Figure 9: 3D stacking technology offers the potential for multiple levels of different types of memory, interleaved with multiple layers of logic. Logic layers can be different, with some being general purpose, others GPUs, accelerators, etc.

In addition to the impact study to the software, DOE should work closely with vendors to alleviate some challenges that currently make adopting 3D stacked logic and memory challenging. For instance, as chip area increases, manufacturing yield (the probability of producing a chip with tolerable defects) decreases. This will worsen with 3D stacking, because now *all* layers need to be manufacturing error free, which further decreases the yield. Therefore, we need mechanisms to isolate manufacturing defects and still meet performance goals, in order to increase yield.

Furthermore, heat extraction now becomes more of a challenge because the center parts of the middle layers in the stack are not close to the surface. We propose to investigate architectural techniques that selectively throttle or power down parts of the chip, by transferring work or scheduling accordingly to still provide balanced performance. 3D stacking is a promising technology and we believe the DOE should invest in both its development and paving the way for using it to its full potential.

Programming Models

Given the unknown nature of how to coordinate such extreme parallelism, proliferation of hardware specialization, and other technological changes we discussed previously, most HPC application teams will be utterly understaffed to deal with the portability concerns of targeting such a diverse and exotic pool of future hardware. Therefore, a programming model that allows them to express their algorithms in a portable way, with compilers and tools doing the low-level translation, will become necessary. Currently, there are few production quality solutions to the diversity of even today's modest ecosystem of hardware. The prevalent example is stencil array languages, which usually tout the ability to target both CPUs and GPUs without modifying the application kernel code. Research of this sort must be pursued aggressively, and expanded to include codes that can run in future hardware accelerators. More computational patterns must be recognized and generalized into lean programming models. These programming models will be scored both on their expressiveness and inclusiveness of HPC applications, as well as the flexibility and efficacy with which the compiler can explore transformations when targeting the varied architectures.

Given a specific programming model, the key to facilitating the compiler in the effort of generating a well performing program will be the availability of high quality information from both the hardware and the application. The hardware will need the expose appropriate cost models to guide the compiler towards constructing quality programs. The software will need to present constraints culling the space of valid program transformations (hopefully embodied by the structure of the code under the semantics of the given programming mode) and annotations indicating likely behavioral scenarios (such as data sizes and access patterns). Research into distilling good hardware cost models, salient application behaviors, and evaluating compiler heuristics/auto-tuning strategies for choosing transformations will be paramount.

To compound the pains of dealing with greater diversity, heterogeneity will also become the norm. Not only can we expect HPC applications to target multiple instances of drastically different hardware and their accompanying programming models, but they will also have to coordinate kernels executing in these different models concurrently, potentially all on chip. This implies another axis of variability pertaining to how these different accelerators will communicate (register sharing, cache coherency, DMA, message passing, etc.). Not only will we require novel programming models to portably map kernels to specialized hardware, there will be a need for a soup-like programming environment to express the communication between the kernels. An example of such a programming environment that has seen success in HPC is CUDA. CUDA presents a single environment that allows the programmer to write sections of code in either the CPU or GPU programming models with many of the difficulties of how data is transferred between the two abstracted away. As the future will bring a much greater variety than just a CPU and GPU, the scope of something like CUDA will have to grow to encompass the plethora of computational paradigms working together.

Conclusion

Digital computing will remain dominant after Moore's law because a larger variety of problems rely on it, but the end of technology scaling will have a substantial detrimental effect on the execution of DOE science missions in scientific discovery and national security. Navigating this oncoming transition is critically important for DOE because many mission applications depend on unique capabilities of digital computing (features not replicated by neuromorphic or quantum). DOE has the breadth of capability and the mission need to maintain performance scaling of digital computing using a combination of architecture, device, manufacturing, and software directions. On the software side, DOE has an enormous investment and expertise which can be used to guide future hardware and assess the impact technology trends have on software. On the hardware size, DOE can take a leading role in future advances in digital computing post Moore's law such as to preserve performance scaling for scientific applications. This white paper presents future opportunities and challenges in each of those four directions, and how we can use them to preserve digital computing scaling. We believe ASCR can take a leading role in doing so by starting up research programs on novel architectures, novel electronic materials, and 3D stacking technologies, with the purpose of continuing performance scaling of electronics. In addition, the existing directions in programming models and runtimes can be augmented to towards the same goal of digital computing scaling, and work in collaboration with hardware and advanced materials researchers.

This transition will require effort on a decadal time scale, so despite whether the CMOS digital computing roadmap has 10, 20, or more years of vitality left, it is important to be laying the strategic foundation for change now.

Acknowledgments

This work was supported by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE- AC02-05CH11231.

Disclaimer

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Bibliography

Aaronson, S. (2008). The Limits of Quantum. *Scientific American*(298), pp. 62-69.

Akinaga, H., & Shima, H. (2010, October). Resistive Random Access Memory (ReRAM) Based on Metal Oxides. *Proceedings of the IEEE, 98*(12), 2237 - 2251.

Alán Aspuru-Guzik, A. D.-G. (2005, September 9). Simulated Quantum Computation of Molecular Energies. *Science, 309*(5741), 1704-1707.

Aly, M. M., Gao, M., Hills, G., Lee, C.-S., Pitner, G., Shulaker, M. M., . . . Mitra, S. (2015, December 29). Energy-Efficient Abundant-Data Computing: The N3XT 1,000x. *Computer, 48*(12), 24 - 33.

Ball, P. (2012, March). The unavoidable cost of computation revealed. *Nature*.

Brian Rogers, A. K. (2009). Scaling the bandwidth wall: challenges in and avenues for CMP scaling. *Proceedings of the 36th annual international symposium on Computer architecture* (pp. 371-382). ACM.

Cong, J. a. (2014). Accelerator-Rich Architectures: Opportunities and Progresses. *Proceedings of the 51st Annual Design Automation Conference* (pp. 1-6). ACM.

Dason, I. B., Kumar, V. R., & Kirubaraj, A. A. (2011). Realization of Magnetic RAM using Magnetic Tunneling Junction in atomic level. *3rd international conference on Electronics Computer Technology (ICECT)* (pp. 397 - 401). IEEE.

Demmel, J. a. (2013). Fast Reproducible Floating-Point Summation. *21st symposium on Computer Arithmetic (ARITH)* (pp. 163 - 172). IEEE.

Dhar, S. a. (2011, January). Advancement in nanoscale CMOS device design en route to ultra-low-power applications. *VLSI Design*, 2.

George Michelogiannakis, X. S. (2015, December). Extending Summation Precision for Network Reduction Operations. *Springer International Journal of Parallel Programming, 43*(6), 1218-1243.

Haron, N. Z. (2008). Why is CMOS scaling coming to an END? *3rd International Design and Test Workshop* (pp. 98 - 103). IEEE.

Hemsoth, N. (2014, June). Are Supercomputing's Elite Turning Backs on Accelerators? *HPC wire*.

Hensbergen, M. S. (2013, 03 25). *Addressing Failures in Exascale Computing.* Retrieved from http://www.mcs.anl.gov/papers/P5022-0913.pdf

Ignatowski, G. H. (2013). A Processing-in-Memory Taxonomy and a Case for Studying Fixed-function PIM. *WoNDP: 1st Workshop on Near-Data Processing.*

John Shalf, R. L. (2014). *Computing Beyond the end of Moore's law.* Lawrence Berkeley National Laboratory. Berkeley: LBNL.

Lance Joneckis, D. K. (2014). *An Initial Look at Alternative Computing Technologies for the Intelligence Community.* DARPA, IDA. IDA-DARPA.

Larus, J. R. (2008). *Spending Moore's Dividend.* Microsoft Research. Microsoft.

Lee, M. H., Lin, J.-C., Wei, Y. -T., Chen, C.-W., Tu, W.-H., Zhuang, H.-K., & Tang, M. (2013). Ferroelectric negative capacitance hetero-tunnel field-effect-transistors with internal voltage amplification. *Electron Devices Meeting (IEDM)* (pp. 4.5.1 - 4.5.4). IEEE.

Mack, C. A. (2011, May). Fifty Years of Moore's Law. *Transactions on semicondictor manufacturing*, 202-207.

Merolla, P. A. (2014, August). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, pp. 668-673.

Michael F. Wehner, L. O.-S. (2011, April). Hardware/Software Co-design of Global Cloud System Resolving Models. *Journal of advances in Modelling earth systems, 3*(4).

Mittal, S. a. (2015, July). A Survey of CPU-GPU Heterogeneous Computing Techniques. *Computing Surveys (CSUR), 47*(4), 69.

Nikonov, D. E. (2013, December). Overview of Beyond-CMOS Devices and a Uniform Methodology for Their Benchmarking. *Proceedings of the IEEE, 101*(12), pp. 2498-2533.

Seth H Pugsley, J. J. (2014). NDC: Analyzing the impact of 3D-stacked memory+logic devices on MapReduce workloads. *International Symposium on Performance Analysis of Systems and Softwar* (pp. 190-200). IEEE.

Snir, M. W. (2013). Addressing Failures in Exascale Computing. *International Journal of High Performance Computing*, ANL/MCS-P5022-0913.

Suman Dattaa, H. L. (2014, May). Tunnel FET technology: A reliability perspective. *Microelectronics Reliability, 54*(5), 861–874.

Swagath Venkataramani, S. T. (2015). Approximate computing and the quest for computing efficiency. *Proceedings of the 52nd Annual Design Automation Conference.* ACN.

Thomson, I. (2014). D-Wave to bust 1,000-qubit barrier with new quantum compute device. *The Register*.

Trancoso, P. (2014). Getting ready for approximate computing: trading parallelism for accuracy for DSS workloads. *Proceedings of the 11th ACM Conference on Computing Frontiers* (p. 8). ACM.

Wang, L.-W. (2008). *A special purpose computer for ab initio molecular dynamic simulations .* Lawrence Berkeley National Laboratory, Computational Material Science and Nano Science Group .

Wim van Dam, M. M. (2001). How powerful is adiabatic quantum computation? *42nd IEEE Symposium on Foundations of Computer Science* (pp. 279 - 287). IEEE.

Wim Vanderbauwhede, K. B. (2013). *High-Performance Computing Using FPGAs.* Springer.

Zhu, Q., Akin, B., Sumbul, H. E., Sadi, F., Hoe, J. C., Pileggi, L., & Franchetti, F. (2013). 3D-stacked logic-in-memory accelerator for application-specific data intensive computing. *Conference on 3D Systems Integration Conference (3DIC)* (pp. 1-7). IEEE.