Implications of the Power Wall: Dim Cores and Reconfigurable Logic

Liang Wang
Department of Computer Science,University of Virginia
lw2aw@virginia.edu

Kevin Skadron
Department of Computer Science,University of Virginia
skadron@cs.virginia.edu

Abstract

Near-threshold operation can increase the number of simultaneously active cores, at the expense of much lower operating frequency ("dim silicon"), but dim cores suffer from diminishing returns as the number of cores increases. At this point, hardware accelerators become more efficient alternatives. To explore such a broad design space, we present an analytical model to quantify the performance limits of many-core, heterogeneous systems operating at near-threshold voltage. The model augments Amdahl's Law with detailed scaling of frequency and power, calibrated by circuit-level simulations using a modified Predictive Technology Model (PTM), and factors in effects of process variations. While our results show that dim cores do indeed boost throughput, even in the presence of process variations, significant benefits are only achieved in applications with very high parallelism or with novel architectures to mitigate variation. Reconfigurable logic that supports a variety of accelerators is more beneficial than "dim cores" or dedicated, fixed-logic accelerators, unless 1) the kernel targeted by fixed logic has overwhelming coverages across applications, or 2) the speedup of the dedicated accelerator over the reconfigurable equivalent is significant.

Keywords: Dark silicon, reconfigurable logic, accelerator, near-threshold computing, Amdahl's law

1. Introduction

Threshold voltage scales down slowly in current and future technology nodes to keep leakage power under control. The dwindling scaling on threshold voltage leads to a slower pace of supply voltage scaling. Since the switching power scales as CV^2f, Dennard Scaling can no longer be maintained, and power density keeps increasing. Furthermore, cooling cost and on-chip power delivery implications limit the increase of a chip's thermal design power (TDP). As Moore's Law continues to double transistor density across technology nodes, total power consumption will soon exceed TDP, and if high supply voltage must be maintained, future chips will only support a small fraction of active transistors, leaving others inactive, a phenomenon referred to as "dark silicon" [4, 15].

Since dark silicon poses a serious challenge for conventional multi-core scaling [4], researchers have tried different approaches to cope with dark silicon, such as "dim silicon" [10] and customized accelerators [13]. Dim silicon, unlike conventional designs working at nominal supply, aggressively lowers supply voltage close to the threshold to reduce dynamic power. The saved power can be used to activate more cores to exploit more parallelism, trading off per-core performance loss with better overall throughput [5]. However, with near-threshold supply, dim silicon designs suffer from diminishing throughput returns as the core count increases. On the other hand, customized accelerators are attracting more attention due to their orders of magnitude higher power efficiency than general-purpose processors [3, 7]. Although accelerators are promising in improving performance with less power consumption, they are built for specific applications, thus have limited utilization on general-purpose applications, and sacrifice die area that could be used for more general-purpose cores. The utilization of each incremental die area must be justified with a concomitant increase in average selling price.

To investigate the performance potential of dim cores and accelerators, we create a performance model extending Amdahl's Law, accompanied by a statistical workload model. We investigate two types of accelerators, application-specific logic (ASIC) and reconfigurable logic (RL). Due to space limits, we only model a generic reconfigurable fabric. Our model can be applied to diverse reconfigurable logic such as FPGAs, Dyser [6], etc. by adjusting the model's parameters of speedup and power-per-unit-area. In this paper, we chose the parameters as representative of what can be achieved with FPGAs, using a sampling of diverse results from papers on FPGA acceleration. We also assume that the reconfiguration overhead is dominated by sufficient utilization of each single configuration. Our main contributions are:

- Systems with dim cores achieve up to 2x throughput over conventional CMPs, but with diminishing returns when factoring in process variations.
- Hardware accelerators, especially RL, are effective to further boost the performance of dim cores.
- Reconfigurable logic is preferable over ASICs on general-purpose applications, where kernel commonality is limited.
- A dedicated ASIC accelerator is beneficial only when: 1) its targeted kernel has a significant coverage, or 2) its speedup over RL is significant.

2. Methodology

We use aggregate throughput as the primary performance metric. Instead of running extensive architectural simulation, we propose an analytical model extending the one proposed in [9], with physical scaling parameters calibrated by simulating a 32-bits adder with a modified Predictive Technology Model [2]. A Niagara2-like in-order core is chosen as a single-core baseline design. We scale the frequency of the Niagara2 to 45nm, and obtain characteristics of this baseline core at 45nm from McPAT [11].

Voltage-to-frequency scaling is modeled by interpolating empirical results from circuit simulations. To investigate the fluctuations of frequency subject to process variations, we simulate the test circuit with Monte-Carlo analysis, assuming a standard Gaussian distribution with 3-σ on the threshold voltage of transistors.

The workload in this paper is defined as a pool of applications. Each single application is divided into serial and parallel parts. Part of an application can be also partitioned into several computing kernels. These kernels can be accelerated by various computing units, such as multicore, possibly dim CPU cores, RL, and customized ASIC. We model the speedup and the power consumption of RL and customized ASIC for a given kernel by u-core parameters (η,ϕ). As characterized in [3], a "u-core" is any unconventional core such as RL or customized ASIC block. ϕ is the relative power efficiency to a single, basic in-order core, and η is the relative performance.[1] With our workload model, the U-core allocations tend to be small due to limited kernel coverage among a wide range of applications. In this case, the U-core power consumption will never exceed the TDPs. As a result, we consider power as a constraint only for conventional cores (e.g. dim cores). We assume that u-cores are only used to accelerate kernels that are ideally parallelized. We also assume the memory bandwidth is sufficient to match a u-core's throughputs, so that u-cores can deliver scalable performance. With these assumptions, we model the relative performance of a u-core proportional to its area, which is a simplifying first-order approximation suitable for tasks with plentiful parallelism, but overlooking synchronization and routing overheads for more complex algorithms on real RL fabrics. We add two more parameters to model the relationship between applications and kernels:

Presence (λ) A binary value indicating whether or not a kernel occurs in an application.

Coverage (ε) The time consumption in percentage for a kernel when the whole application is running with a single baseline core.

Based on application characterizations for popular suites such as PARSEC, Minebench, and SD-VBS, we propose a statistical approach to model kernels' speedup (η), presence (λ), and coverage (ε), which are summarized in Equation 1.

$$\begin{cases} \eta & \sim \mathcal{N}(\mu_s, \sigma_s^2) \\ \lambda & = B(p) \qquad \text{where } p = PDF(\eta) \\ \varepsilon & \sim \mathcal{N}(\mu_c, \sigma_c^2) \end{cases} \tag{1}$$

The speedups (η) of kernels are modeled as following a normal (N) distribution. Kernels, like library calls, are mapped to medium speedup close to the mean of the distribution, while application-specific kernels are mapped to tails of the distribution. The left tail denotes application-specific kernels that are hard to accelerate, such as control-intensive ones. The right tail denotes application-specific kernels that are highly efficient on accelerators, such as compute-intensive streaming ones. The presences (λ) of these kernels are modeled as following a Bernoulli (B) distribution with $PDF(\eta)$ as its parameter. The coverage (ε) is modeled as following another independent normal (N) distribution. We extract parameters for speedups by normalizing reported values from recent publications in reconfigurable logic community. Due to space limits, we choose FPGA as the representative RL fabric. Our model can be applied to diverse RLs by adjusting speedup and power-per-unit-area parameters. We justify parameters for coverage with values we collected from PARSEC. All the parameter values are summarized in Table 1

We assume that the power and the area of the last-level cache (LLC) remains a constant ratio to the whole system. According to [11], we assume that un-core components attribute to 50% of both the total thermal design power (TDP) and the die area; The bandwidth of the memory subsystem is assumed to

[1] In the original paper of [3], μ is used to denote the relative performance of u-cores. However, μ is commonly used as the mean in statistics. Therefore, we use η as an alternative to avoid confusions.

μ_s RL	μ_s ASIC	σ_s RL	σ_s ASIC	μ_c	σ_c
40x	200x	10	50	40%	10%

Table 1. Summary of statistical parameters for kernel modeling

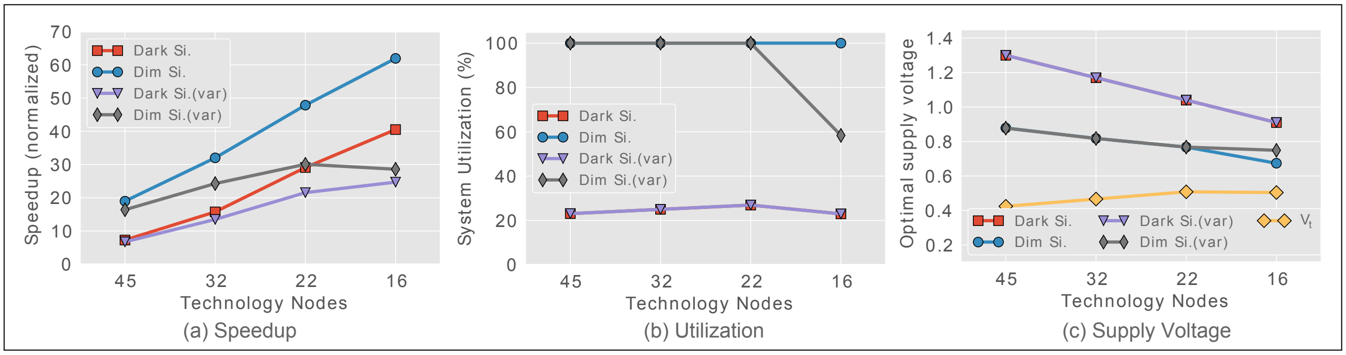
be sufficient for future technology nodes. We extract core-only TDP and die area from Oracle SPARC T4, as $120W$ and $200mm^2$, respectively.

Within an application, we assume that each kernel takes sufficient time to ignore various overheads in reconfiguring accelerators, as well as setting up the execution context. We leave the detailed overhead modeling and the modeling of transient kernel behaviors as future work.

3. Analysis

3.1. Effectiveness of Dim Silicon with Near-threshold Operation

Unlike dark silicon, which we define as maximizing the frequency of cores to improve the overall throughput at the expense of potentially turning some off, dim silicon aims to maximize chip utilization to improve overall throughput by exploiting more parallelism. Dark silicon systems apply the highest supply voltage, which is assumed to be 1.3x the nominal supply voltage in this model, to cores in parallel mode; while dim silicon systems scale down supply and set the voltage to optimize the overall throughput. Running all cores at nominal supply is merely one "dim" design point. The comparisons between dark and dim silicon are shown in Figure 1.



(a) Speedup (b) Utilization (c) Supply Voltage

Figure 1. Dark silicon vs. dim silicon, regarding the throughput based speedup (*left*), die utilization (*middle*), and the supply voltage (*right*) with optimal throughput, for a "large" system budget.

For performance, dim silicon beats dark silicon with up to 2X throughput improvement at 16nm. When variation comes into play, however, both systems experience throughput losses compared to non-variation cases respectively. Dim silicon systems suffer even more compared to dark silicon, revealing higher vulnerability of dim silicon to process variations. Dim silicon is able to utilize more cores to achieve a higher utilization, up to 100% for the large system. Similarly, the utilization of dim silicon systems is sensitive to the process variation. Being aware of variation, the large system with dim silicon sees the utilization drop to as low as 60%. Finally, the optimal supply voltage is approaching the threshold as technology advances, and comes within 20% of the threshold at 16nm. Variation raises the optimal supply at 16nm due to the dramatic increase of frequency penalty at near threshold.

Although dim cores manage to deliver higher throughput, they suffer from diminishing returns as the number of active cores increases. The lower per-core frequency of dim cores compromises the speedup improvement coming from increasing cores. As a result, it is not cost-effective to lower supply voltage aggressively to reach the optimal throughput. The limited speedup improvement from aggressive dim cores provides an opportunity to allocate some of the die area to more efficient customized hardware, such as RL, and ASICs. For the sake of studying the future trend, we choose 16nm as the technology node for the rest of analysis.

3.2. Dim Silicon with Accelerators(RL and ASIC) on a General-Purpose workload

To study the potential benefits of accelerators, we build a synthetic workload from applications following characteristics defined by Equation 1 and Table 1. (Our speedups are derived from RL and ASIC speedups in the literature, but note that these tend to be reported for applications that benefit the most from such architectures. Some tasks will be much more difficult to speed up.) We choose ten synthetic kernels with speedups evenly spread from $\mu - 3\sigma$ through $\mu + 3\sigma$, and various probabilities of occurance. We refer to these kernels with identifiers 0 through 9: the speedup of kernels goes up from kernel 0 through kernel 9, while the probability of presence peaks at kernel 4 and 5, and is the smallest at kernel 0 and 9 (i.e. normal distribution). We assume a fixed scaling ratio of 5x for ASIC performance relative to their corresponding RL implementations. We draw a sample population of 500 applications as the workload. We use the mean of speedups on all applications as the performance metric. We do a brute-force search to find the optimal allocations for U-cores.

With RL the only accelerator, the system manages to achieve an optimal throughput on average that is almost 3x larger than a non-RL system with only dim cores. Only a relatively small area allocation of RL is required, e.g., 20%-30%, across a range of kernel characteristics. Additional RL is not cost-effective because it limits the number of available dim cores to accelerate the non-kernel portion of the application, delivering worse overall performance according to Amdahl's Law. More details on the RL benefit under alternative scenarios are available in [16].

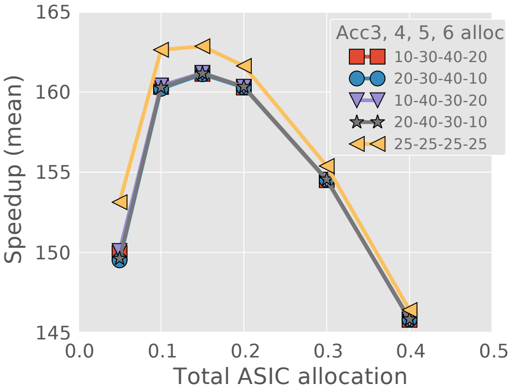


Figure 2. Speedup of a system composed of dim silicon and four dedicated ASIC accelerators. Kernels are chosen from the synthetic set of kernels as library-call kernels numbered 3 through 6. The X-axis is the total allocation to ASIC kernels, relative to the die area budget. The legend labels show the allocation of a kernel relative to the total ASIC allocation, for example, "10-30-40-20" indicates 10%, 30%, 40%, 20% allocation out of the overall ASIC allocation on accelerators targeted kernel 3 through 6.

We then discuss systems with only dedicated ASIC accelerators. Although specialized accelerators can achieve tremendous speedup for application-specific kernels, limited utilization of a targeted kernel within a general-purpose workload leads to limited overall performance benefit for the workload as a whole. However, accelerators for library-call kernels tend to be more beneficial in overall speedup of the workload due to their higher overall coverage. We plot potential allocations of four accelerators for library-call kernels (numbered as 3 through 6 from the synthetic set of kernels mentioned before) in Figure 2. To achieve the best overall throughput, the area allocation tends to be evenly distributed among all accelerators to achieve the best performance. It is no more than 20% that the total area allocation on all hardware accelerators to achieve the optimal throughput. However, when compared to RL-only systems, ASICs-only systems have inferior throughput. More evaluations on ASIC-only systems can be found in [16].

In the previous two paragraphs, we have showed the benefits of accompanying conventional but dim cores with RL and ASIC accelerators, respectively. Both of them exhibit performance improvement over a baseline system composed of dim cores only. However, it is not necessary to achieve a better performance by combining both types of U-cores. As shown in Figure 3, the best performance is achieved with an RL-only system organization. This counter-intuitive result comes from two reasons. First, with a general-purpose workload, the average coverage of a kernel is small, due to either small coverage among applications (library-call kernels) or rare presence (application-specific kernels). This exaggerates the cost of a single ASIC accelerator, which is only helpful for a specific kernel. Second, we scale accelerator performance proportionally to its area, and make a conservative assumption on the performance ratio between ASIC and RL (see Section 3.4 for impacts of alternative performance ratios). With these assumptions, RL will be more powerful on a kernel than the corresponding ASIC implementation, because the entire RL is available, and is much larger than the ASIC accelerators (and a large RL allocation is justified by the high utilization achievable across multiple kernels). Consequently, the system ends up with an RL-only configuration. As a result, the RL implementation is more favorable due to its versatility across kernels. In the following two sections, we are going to explore two alternative scenarios which: 1) Alter the coverage of kernels to see the benefit of ASIC accelerators; 2) Alter the performance ratio assumption between RL and ASICs.

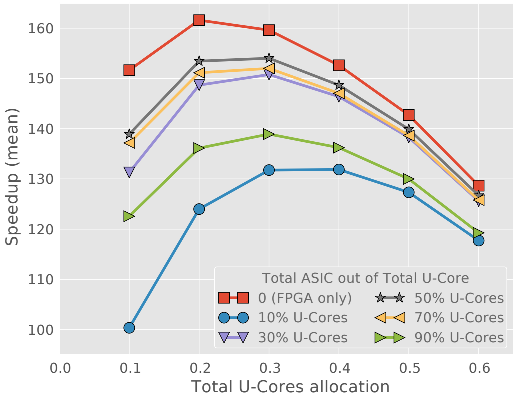


Figure 3. Speedup of a system composed of dim silicon, RL and ASICs. The X-axis is the total allocation to U-cores, including both RL and ASIC accelerators, relative to the die area budget. Legend labels indicate the total allocation of ASIC accelerators, relative to the total U-cores allocation. We assume an evenly distributed allocation among ASIC accelerators, since it shows the best performance in previous analysis.

3.3. Benefit of ASIC Accelerators

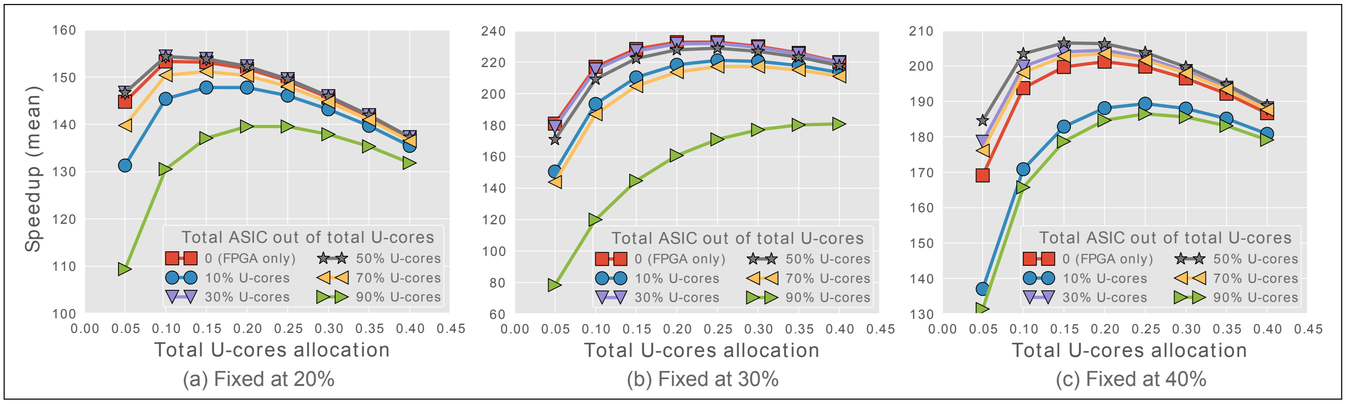


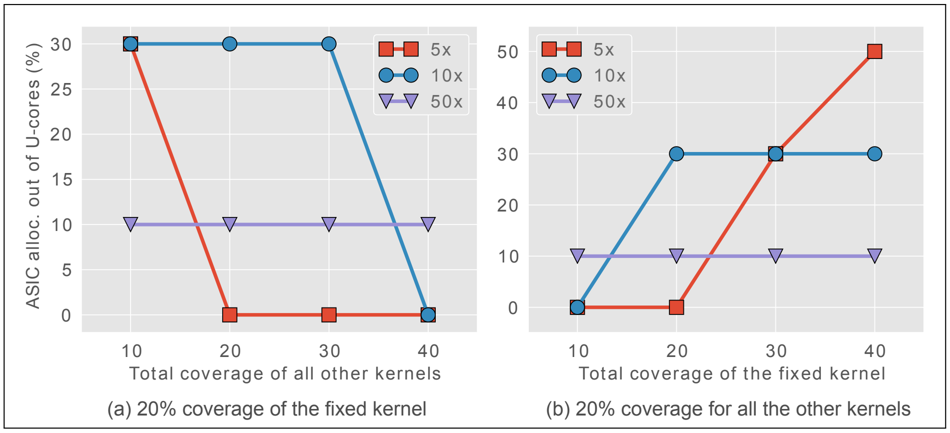
Figure 4. 10% coverage for all the other kernels, while the fixed kernel has 20% (*left*), 30% (*middle*), and 40% (*right*) coverage across all applications in a workload.

Although RL works better with general-purpose workloads, an ASIC accelerator becomes beneficial when its targeted kernel is common enough across applications in a workload. In order to capture this scenario, we add one more kernel to the set of ten kernels we have used in previous analyses. The coverage of this new kernel is fixed across all applications. We also hold the total coverage of all the other kernels. By varying the coverage of the new kernel and all the other kernels, we have generated several alternative workloads, within which the new kernel has a considerable amount of coverage across all applications. We show results of 10% coverage of all the other kernels in Figure 4, leaving more results in [16]. One of the most significant observations from these plots is that the dedicated ASIC accelerator is not beneficial at all until its targeted kernel covers more than the total of all the other kernels within an application. This observation is consistent with commercial MPSoC designs such as TI's OMAP4470, which has dedicated accelerators for only image processing, video encoding/decoding, graphics rendering, since these functions are expected to be quite common for the targeted mobile device workloads.

3.4. Sensitivity on ASIC Performance Ratio

We have assumed that a fixed logic accelerator provides 5x better performance than the corresponding RL accelerator. This assumption could be quite conservative, especially when the reconfiguration overhead can not be amortized or hidden effectively. Alternatively, we increase the performance ratio to 10x and 50x, and plot the dedicated ASIC allocation for the kernel of a fixed coverage when the system achieves its optimal performance, regarding the total coverage of all the other kernels in Figure 5(a) and fixed coverages of the kernel targeted by dedicated ASIC in Figure 5(b).

In Figure 5(a), the coverage of the dedicated ASIC's kernel is fixed at 20%, while the total coverage of all the other kernels varies from 10% through 40%. When the performance of dedicated ASIC accelerators is merely 5x better than RL, the system ends up with zero allocation to the dedicated accelerator, unless the coverage of its targeting kernel (20%) is larger than the total coverage of all the other kernels (10%). However, when the performance of the dedicated accelerator boosts to 50x better than RL, the dedicated accelerator will always hold its place with 10% allocation out of all u-cores area. In the case of 10x performance ratio, a dedicated accelerator is beneficial, unless the total coverage of all the other



(a) 20% coverage of the fixed kernel

(b) 20% coverage for all the other kernels

Figure 5. Sensitivity study on ASIC performance ratio.

kernels is as large as 40%, overwhelming the dedicated kernel's coverage of 20%. A similar trend is observed in Figure 5(b), where the total coverage of all the other kernels is fixed at 20% and the dedicated kernel's coverage varies from 10% through 40%: a dedicated kernel is favored when there is a huge gap between either: 1) the performance of the dedicated and the reconfigurable accelerator (e.g. 50x), or 2) the coverage of the dedicated kernel and the total coverage of all the other kernels (e.g. 2x larger).

In summary, the benefit of ASIC accelerators is quite dependent to its relative throughput to RL accelerators on the same kernel. With a large throughput gap (e.g. 50x), ASIC accelerators are beneficial to be included in the optimal design, even when the targeted kernel has a limited presence across applications (e.g. as low as 10%). Otherwise, RL is preferable, as long as reconfiguration overhead can be neglected. However, in this paper we have focused on high-end desktop and server chips, which the TDP is relatively high. In cases where TDP is very low relative to area (mobile chips such as NVIDIA's Tegra 2), only a small portion of the chip can be active at any point in time, so even a large allocation of RL cannot be fully utilized. This means ASICs become more beneficial, even when their utilization may be low.

3.5. Alternative Serial Cores

Although massive parallelism has been observed in several computing domains, such as high performance computing, there are many applications with periods of serial activities. In this case, adding a high-performance out-of-order (O3) core is more beneficial, especially when dim cores get diminishing returns on throughput. We also assume that the serial core is gated off in parallel mode to save power for throughput-oriented dim cores. We study both conventional out-of-order core, such as Core i7, as well as core-selectability [12]. We find that even with an embarrassingly parallel application, the die area investment on a dedicated O3 core is still beneficial. In the case of ideal parallelism of 100%, the performance loss by introducing a O3 core is around 14% at 45nm, While at 16nm, the performance loss is less than 1%, due to diminishing performance returns from a larger number of throughput cores, and lower percentage area impact of one O3 core. More details on alternative serial core analysis can be found in [16].

4. Related Work

The power issue in future technology scaling has been recognized as one of the most important design constraints by architecture designers [12, 15]. Esmaeilzadeh et al. performed a comprehensive design space exploration on future technology scaling with an analytical performance model in [4]. They did not consider lowering supply voltage, and concluded that future chips would inevitably suffer from a large portion of dark silicon. In [1], Borkar and Chien indicated potential benefits of near-threshold computing with aggressive voltage-scaling to improve the aggregate throughput. We evaluate near threshold in more detail with the help of an analytical model calibrated with circuit simulations. In [10], Huang et al. performed a design space exploration for future technology nodes. They recommended dim silicon and briefly mentioned the possibility of near-threshold computing. Our work exploits circuit simulation to model technology scaling and evaluates in detail the potential benefits of improving aggregate throughput by near-threshold computing. There are numerous works studying the benefit of hardware accelerators as a response to dark silicon. Chung et al. studied the performance potential of GPU, FPGA and ASIC in [3], regarding physical constraints of area, power, and memory bandwidth. Although a limited number of applications were studied in their work, our work corroborates that reconfigurable accelerators, such as FPGA, are more competitive than dedicated ASICs. Wu and Kim did a study across several popular benchmark suites characterizing the kernels to be accelerated in [17]. Their work suggested that a large number of dedicated accelerators are required to achieve a moderate speedup, due to the minimal functional level commonality in benchmark suites like SPEC2006. This is consistent with our observation that dedicated fixed logic accelerators are less beneficial due to limited utilization across applications in a general-purpose workload, and the importance of efficient, reconfigurable accelerators. In [14], Tseng and Brooks build an analytical model to study tradeoffs between latency and throughput performance under different power budgets and workloads. However, the model lacks the support of voltage and frequency scaling, especially near threshold, and the capability of hardware accelerator performance modeling, limiting the design space exploration of future heterogeneous architectures. In [18], Zidenberg et al. propose the MultiAmdahl model to study the optimal allocations to each computational units, including conventional cores and u-cores. The MultiAmdahl model can not support voltage scaling and near threshold effects, so its design space exploration capability is limited for heterogeneous systems under dark/dim silicon projections. In [8], Hempstead et al. propose the Navigo model to study power-constrained architectures and specialization. The Navigo model did not model process variations at near threshold. Moreover, the specialization modeling was limited by a single kernel, compared to multiple kernels and accelerators in our model.

5. Conclusions

In this paper, we propose an analytical model for performance of dim silicon, calibrated by circuit simulations. We find that dim cores to provide a moderate speedup up to 2x over conventional CMP architecture that will otherwise suffer from poor speedup with further technology scaling. However, the poor per-core frequency of dim cores leads to a diminishing returns in throughput, creating opportunity for more efficient hardware accelerators such as RL and ASIC. We extend the performance model with a model of general-purpose workloads via a statistical approach. We show that RL is more favorable with general-purpose applications, where commonality of kernels is limited. A dedicated ASIC accelerator will not be beneficial unless the average coverage of its targeted kernel is twice as large as that of all the other kernels or its speedup over RL is significant (e.g. 10x-50x). However, it is hard to identify common function-level hotspots in real world applications. In fact, the most important conclusion from this work is the need for efficient, on-chip, RL resources that can be rapidly reconfigured to implement a wide variety of accelerators. However, ultra-low-power systems, in which die area is not a major constraint, will be able to afford a larger number of ASIC units, even though they may be only intermittently utilized.

6. Acknowledgements

We thank Mircea Stan and Martha Kim for their helpful comments. This work was supported by the SRC under GRC task 1972.001 and the NSF under grants MCDA-0903471 and CNS-0916908 and by DARPA MTO under contract HR0011-13-C-0022. The views expressed are those of the authors and do not reflect the official policy or position of the sponsors.

References

[1] S. Borkar et al. The Future of Microprocessors. *Communication of the ACM*, 54(5), May 2011.

[2] B. H. Calhoun et al. Sub-threshold Circuit Design with Shrinking CMOS Devices. In *ISCAS*, March 2009.

[3] E. S. Chung et al. Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPGPUs? In *MICRO*, 2010.

[4] H. Esmaeilzadeh et al. Dark Silicon and the End of Multicore Scaling. In *ISCA*, 2011.

[5] D. Fick et al. Centip3De: A 3930DMIPS/W configurable near-threshold 3D stacked system with 64 ARM Cortex-M3 cores. In *ISSCC*, 2012.

[6] V. Govindaraju et al. Dynamically Specialized Datapaths for Energy Efficient Computing. In *HPCA*, 2011.

[7] R. Hameed et al. Understanding Sources of Inefficiency in General-purpose Chips. In *ISCA*, 2010.

[8] M. Hempstead et al. Navigo: An Early-Stage Model to Study Power-Constrained Architectures and Specialization. In *Workshop on Modeling, Benchmarking, and Simulations*, MoBS '09, 2009.

[9] M. D. Hill and M. R. Marty. Amdahl's Law in the Multicore Era. *Computer*, 41(7), July 2008.

[10] W. Huang et al. Scaling with Design Constraints: Predicting the Future of Big Chips. *IEEE Micro*, 31(4), July 2011.

[11] S. Li et al. McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures. In *MICRO*, 2009.

[12] H. H. Najaf-abadi et al. Core-Selectability in Chip Multiprocessors. In *PACT*, 2009.

[13] M. B. Taylor. Is Dark Silicon Useful?: Harnessing The Four Horsemen of the Coming Dark Silicon Apocalypse. In *DAC*, 2012.

[14] A. C.-N. Tseng and D. Brooks. Analytical Latency-Throughput Model of Future Power Constrained Multicore Processors. In *Workshop on Energy-Efficient Design*, 2012.

[15] G. Venkatesh and othres. Conservation Cores: Reducing the Energy of Mature Computations. In *ASPLOS*, 2010.

[16] L. Wang and K. Skadron. Dark vs. Dim Silicon and Near-Threshold Computing Extended Results. Technical Report UVA-CS-13-01, Department of Computer Science, University of Virginia, 2013.

[17] L. Wu and M. A. Kim. Acceleration Targets: A Study of Popular Benchmark Suites. In *Dark Silicon Workshop*, 2012.

[18] T. Zidenberg et al. MultiAmdahl: How Should I Divide My Heterogeneous Chip? *CAL*, 11(2):65–68, 2012.