Dark Silicon as a Challenge for Hardware/Software Co-Design

Invited Special Session Paper

Muhammad Shafique*, Siddharth Garg†, Tulika Mitra‡, Sri Parameswaran§, and Jörg Henkel*

*Chair for Embedded Systems (CES), Karlsruhe Institute of Technology, Germany
†Department of ECE, University of Waterloo, Canada
‡School of Computing, National University of Singapore, Singapore
§School of Computer Science and Engineering, University of New South Wales, Australia

Corresponding Authors: muhammad.shafique@kit.edu, s6garg@uwaterloo.ca

ABSTRACT

Dark Silicon refers to the observation that in future technology nodes, it may only be possible to power-on a fraction of on-chip resources (processing cores, hardware accelerators, cache blocks and so on) in order to stay within the power budget and safe thermal limits, while the other resources will have to be kept powered-off or "dark". In other words, chips will have an abundance of transistors, i.e., more than the number that can be simultaneously powered-on. Heterogeneous computing has been proposed as one way to effectively leverage this abundance of transistors in order to increase performance, energy efficiency and even reliability within power and thermal constraints. However, several critical challenges remain to be addressed including design, automated synthesis, design space exploration and run-time management of heterogeneous dark silicon processors. The hardware/software co-design and synthesis community has potentially much to contribute in solving these new challenges introduced by dark silicon and, in particular, heterogeneous computing. In this paper, we identify and highlight some of these critical challenges, and outline some of our early research efforts in addressing them.

1. INTRODUCTION

For decades, the Dennard Scaling model (i.e., scaling feature sizes and voltages by the same factor) has allowed chip designers to keep power density (i.e., power consumption per unit area of silicon) constant when moving from one technology node to another. More recently, however, the exponential dependence of leakage power consumption on threshold voltage has constrained further threshold- and supply-voltage scaling. As a result, the power density is now increasing with technology scaling, such that it can no longer be cooled down in cost effective ways considering the physical limitations imposed by cooling technologies and packaging. This gives rise to the so-called *Dark Silicon* problem [6,7,36].

ESWEEK'14, October 12-17, 2014, New Delhi, India.

http://dx.doi.org/10.1145/2656075.2661645.

Dark silicon refers to the constraint that a significant fraction of transistors on a chip cannot be powered-on at the nominal voltage for a *Thermal Design Power* (TDP) budget and have to remain dark, i.e., power-gated. The TDP is the maximum power budget supplied to a chip while keeping the chip temperature below the thermal safe temperature (T_{safe}). In case the TDP is exceeded, the chip temperature will start rise beyond the cooling capacity, resulting in either thermal run-away or activation of dynamic thermal management (DTM) mechanisms that will throttle the chip. Using technology data from ITRS and Intel, prior studies [6,7] have predicted that at the 8 nm technology node, 50%-80% of the chip area will be dark for both CPU and GPU-based systems executing massively parallel workloads.

Given this scenario, the question posed for the architecture, design automation, and hardware/software co-design communities is: can the (over)abundance of transistors on dark silicon chips be harnessed to improve important design metrics like performance, energy/power efficiency or even reliability under TDP constraints [32], and if so, how? Recent work in this context has explored dark silicon chip with: (i) a multitude of application-specific and general purpose accelerators [4, 19], (ii) exploiting (micro-)architectural heterogeneity [1, 6, 37] and (iii) near-threshold computing (i.e. operating at a very low voltage to power-on more cores) [14]. Moreover, the available dark silicon can also be leveraged to mitigate reliability threats in the nano-era [15, 27, 32] that include soft errors, aging and process variations [8, 9]. In all the instances above, the key idea is to *overprovision* the chip with heterogeneous computing resources — for instance, application-specific accelerators or cores with differing power, performance and reliability characteristics — and to select the subset of computing resources at run-time that maximize the desired objective within the TDP budget.

In fact, problems relating to the design and run-time management of heterogeneous computing platforms have been extensively addressed by the hardware/software co-design and systems synthesis communities, particularly in the context of application-specific multi-processor systems on chip (MPSoCs). Motivated by the dark silicon challenge, heterogeneity is now beginning to find a foothold in general-purpose processor architectures including some commercially available chips like the ARM big.LITTLE processor. However, because the general-purpose computing domain is substantively different from the application-specific domain (focus on providing best-effort versus worst-case performance guarantees, lack of well-defined application performance mod-

els, greater application diversity to name a few), existing solutions cannot simply be reused and new methodologies are required.

In this paper, we identify some critical challenges introduced by dark silicon and highlight promising solutions to these challenges, with a specific focus on the design and run-time management of future generation heterogeneous dark silicon processors. Our broader goal is to spur greater awareness and discussion of these challenges in the hardware/software co-design and systems synthesis community, and to position the dark silicon problem as one that this community can have a large impact on solving.

1.1 Critical Challenges in the Dark Silicon Era

To fully exploit the the abundance of transistors in the dark silicon era using heterogeneity, the following critical challenges must be addressed:

1. **Heterogeneous Architecture Synthesis and Design Space Exploration Challenge:** At design time, the challenge is how to optimally synthesize a chip given a library of heterogeneous cores and application-specific accelerators, in other words, how many cores and accelerators of each type should the processor be provisioned with? The constraints are the total chip area, TDP and peak temperature while the optimization objectives can include performance, energy-efficiency and even reliability. Since the design space is large, *automated* algorithms are to efficiently navigate this design space and provide high-quality solutions. In Section 2.1, we will discuss one such approach to address this challenge.
2. **On-chip Network Design Challenge:** A second design time challenge that emerges is how to effectively interconnect the large number of heterogeneous processing elements (cores and accelerators) on the chip. Compared to conventional processors, only a subset of the processing elements are simultaneously active in a dark silicon processor and this subset changes with time, giving rise to the need for a highy adaptive NoC. In Section 2.2, we will discuss one solution for an adaptive NoC that is itself heterogeneous and leverages the abundance of transistors on the chip.
3. **Run-time Power and Thermal Management Challenge:** Given a diverse set of heterogeneous computation and communication resources on the chip and TDP/thermal constraints, the run-time systems needs to perform efficient power and thermal management in order to maximize performance under TDP/thermal constraints. In conventional chips without dark silicon, maximizing performance within a TDP/thermal constraint involves simply activating all cores. On the other hand, to maximize performance for dark silicon processors, the run-time system must determine which processing elements to activate (the others remain dark) and the power state of processing element. In fact, as we will discuss, this problem is challenging even if all cores on the chip are homogeneous (see Section 2.4). In addition, in Section 2.3 we will discuss potential solutions for heterogeneous processors such as the ARM big.LITTLE.
4. **Reliability and Variability Challenge:** Although execution time, throughput, power and energy are typically thought of as the most important metrics of system performance, reliability and predictability have become increasingly important metrics in the nanoscale era. Therefore, another important question is whether the the additional transistors available on dark silicon processors can be used to increase reliability or to combat the impact of manufacturing process variations. For instance, some existing chips already use redundancy to address manufacturing defects by provisioning a chip with redundant cores. In Section 2.5, we discuss how in-field faults and aging mechanisms, and parametric process variations can also be addressed.

2. DESIGN AND RUN-TIME MANAGEMENT OF HETEROGENEOUS DARK SILICON PROCESSORS

Fig. 1 shows a heterogeneous dark silicon processor along with its hardware and software layers: the hardware layer consists of processing cores (organized as "tiles") and the interconnect, while the software layer consists of the applications and the run-time system that maps and schedules applications and controls the power states of the hardware components.

Hardware Layer.

As shown in Fig. 1, each tile on the chip is potentially heterogeneous. We refer to these tiles as *Heterogeneous Tiles* (HTs). The heterogeneity can take one of many forms, which we enumerate below. Note that list is by no means exhaustive, and other forms of heterogeneity can also be incorporated.

1. *Functional heterogeneity* that exists in the form of processing engines with very different functional behaviors such as general-purpose cores, GPU, and special purpose accelerators.
2. *Micro-architectural heterogeneity* that is provided by cores with the same instruction-set architecture (ISA) — we refer to these as iso-ISA cores — but diverse power-performance characteristics. For example, one of the tiles in Fig. 1 contains two clusters of general-purpose cores: a cluster of small cores with simple but power-efficient micro-architecture and a cluster of big cores with complex but power-hungry micro-architecture. The small and the big cores share the same ISA, that is, the same binary executable can run on both types of cores albeit with different power-performance behavior.
3. *On-chip interconnect heterogeneity* that is provided by the existence of multiple parallel interconnection networks with different router micro-architecture (as shown in Fig. 1) and even network topology. Depending on the scenario, only one is active at any point in time while the others are dark.
4. *Reliability heterogeneity* that is provided by the cores with the same ISA but diverse reliability characteristics, i.e., cores that are protected against certain failure mechanisms, like soft-errors, to different degrees. The "Reliability Tile" in Fig. 1 contains eight different types of cores where different part of the cores (the pipeline, cache, register file, etc.) are protected using triple-modular redundancy (TMR) (see legend in Fig. 1).
5. *Technology heterogeneity* that results from using different device technologies for each component, for instance, one tile can be implemented using standard CMOS technology while another tile can be implemented using CMOS compatible steep-slope devices. Different device technologies

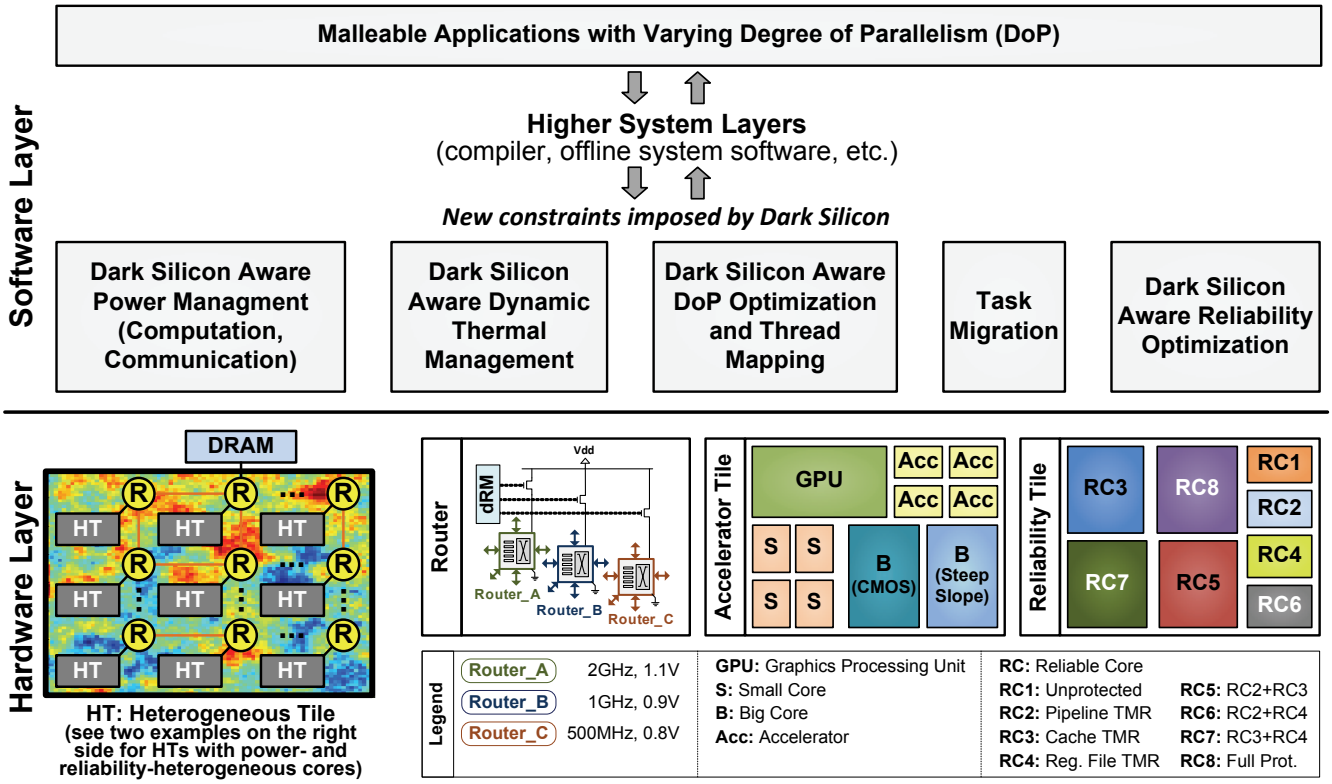

Figure 1: Hardware-software co-design for dark silicon processors.

have very different power and performance characteristics; for instance, steep-slope devices have lower leakage power than conventional CMOS, but are also slower.

6. *Process variation induced heterogeneity* that arises not from design intent, but as a consequence of the inherent randomness in the manufacturing process. As a consequence, even two identical cores or tiles on the chip (that is, identical by design) can have very different maximum operating frequency and leakage power dissipation values.

In addition to the forms of heterogeneity discussed above, a heterogeneous dark silicon processor typically consists of many voltage islands, for instance, one corresponding to each HT or group of HTs. For example, the recent Samsung Exynos 5410 Octa SoC [34] that powers Samsung Galaxy S4 devices integrates high performing, complex, out-of-order ARM Cortex-A15 and energy-efficient, simple, in-order ARM Cortex-A7 cores (ARM big.LITTLE architecture [12]) along with a GPU and multiple accelerators on the same chip. The Cortex-A7 cluster, Cortex-A15 cluster, and the GPU each have their own independent voltage islands.

Software Layer.

In a heterogeneous dark silicon processor, there may exist multiple run-time *contexts*, each satisfying the given TDP and thermal constraints but with very different performance, reliability and spatio-thermal characteristics. A run-time context denotes a set of cores or HTs that are active, their locations, temperature, V/F-level, and the active NoC layer. In addition, for a given context, the following decisions need to be made: (i) how to parallelize each application, i.e., the number of parallel threads which we will refer to its degree-of-parallelism (DoP); and (ii) mapping of threads to active cores. All of the decisions above are made by a *run-time system manager* that is typically implemented in software (in the operating system) or potentially, jointly between hardware and software.

In the following, we will discuss our preliminary ideas on addressing the four challenges that we highlighted in Section 1.1 in the context of the prototypical heterogeneous dark silicon processor that we described above. In particular, we start by addressing the automatic synthesis and design space exploration challenge.

2.1 Synthesis and Design Space Exploration of Heterogeneous Dark Silicon Processors

The synthesis challenge for heterogeneous dark silicon processors is to optimally provision a chip with heterogeneous computational resources which can, in general, include accelerators, and functionally and/or micro-architecturally heterogeneous processing cores. We begin with a restricted version of the synthesis problem in which we only consider micro-architecturally heterogeneous cores in our component library.

The problem overview is shown in Figure 2: along with a library of micro-architecturally heterogeneous cores, we are given a set of multi-threaded benchmark applications (sequential, single-threaded applications are a special case and thus easily incorporated in this framework), a chip area budget and a TDP constraint. The goal is to minimize execution time, averaged over the benchmark suite.

For this synthesis problem, one approach is to assume that each application will execute with a static, user specified DoP. However, allowing each application to execute with its

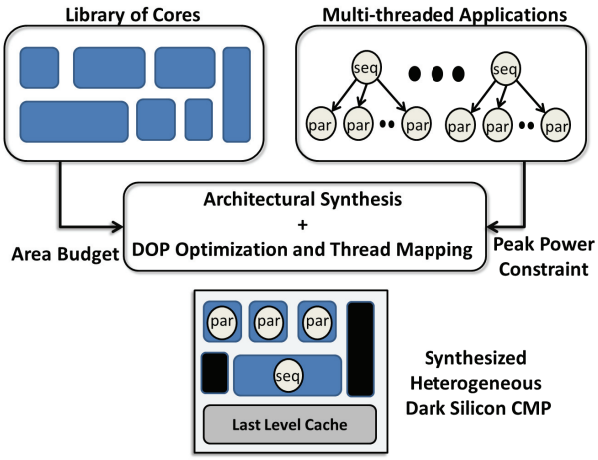

Figure 2: Overview of the architecture synthesis problem for heterogeneous dark silicon processors. Figure reproduced from [38].

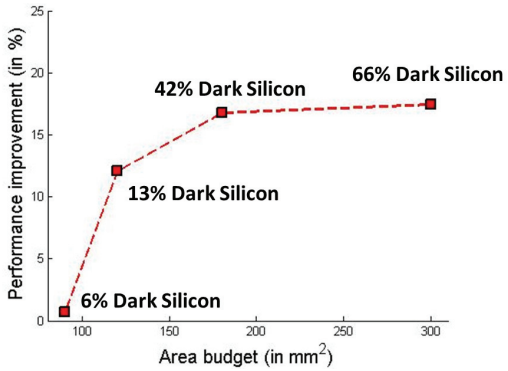

Figure 3: Performance improvement as a function of chip area (and the percentage dark silicon) for the same TDP. Figure reproduced from [38].

own optimal DoP provides further opportunities for optimization, particularly for a heterogeneous dark silicon processor. For instance, an application could execute with low DoP on a small number of high power/performance cores, or with high DoP on a larger number of low power/performance cores. While the power consumption in both scenarios might be identical, the execution time can be very different.

The introduction of the DoP knob results in two new challenges, as highlighted below.

- First, the optimal DoP and the optimal mapping of threads to cores depend on the number of cores of each type in the synthesized heterogeneous processor, resulting in a "chicken-and-egg" problem (should the number of cores of each type be determined first or the DoP of each benchmark?).
- Second, analytical model for the execution time of an application that are functions of both its DoP and mapping of threads to (heterogeneous) cores are lacking. If available, such a model can then be plugged into a a mathematical optimization formulation of the synthesis problem.

In recent work, we have taken the first step towards addressing the challenges mentioned above [38]. We observe, as others have, that the execution time of an application as a function of its DoP is governed by Amdahl's Law [10] which splits up execution into serial and parallel phases. Parallel phases are sped-up proportional to DoP while serial phases are unaffected. Furthermore, we observe that in the heterogeneous setting, each thread in a parallel phase can be mapped to a core of a different type — consequently, the execution time of the parallel phase will be determined by the *slowest* thread. Together, these observations allow us to determine a simple but accurate analytical model for execution time as a function of DoP and thread to core mapping. We have verified the validity of our model across a wide range of multi-threaded benchmark applications.

The proposed analytical model enables an integer-linear programming (ILP) formulation of the heterogeneous multi-core synthesis problem that synergistically optimizes the DoP and thread to core mapping for each application. Empirically, however, we find that the ILP solver converges slowly, even for relatively small problem instances Instead, we propose an *iterative procedure* that first keeps DoP fixed and optimizes the number of cores of each type, and then optimizes DoP. This repeats till convergence, or till the improvement in execution time saturates. We have show that this iterative process generates high-quality solutions while being significantly faster than the ILP approach.

An interesting consequence of the proposed approach is that we can now study the design space of heterogeneous dark silicon processors. For instance we can answer questions like: how much does performance improve as chip area is improved, while keeping the TDP the same? This is shown in Figure 3 (more details on the experimental set-up can be found in [38]). The key observation is that the performance benefits are quite significant at first, but saturate with increasing chip area. This suggests that micro-architecturally heterogeneity alone is not itself sufficient to best utilize the abundance of transistors in the dark silicon era.

Of course, there still remain several open challenges. For instance, the performance models we proposed for multi-threaded applications are only accurate for data parallel workloads, but not for pipeline or thread-pool based parallelism. In addition, our performance metric, i.e., execution time, is not appropriate for server settings where new jobs are continuosly arriving and the queueing delay jobs must be taken into account (our recent paper [26] sheds more light on this setting). Finally, automated synthesis techniques must be extended to incorporate not only processing cores, but also accelerators, reconfiurable logic and GPUs.

2.2 Heterogeneous Networks-on-Chip for Dark Silicon

Having discussed heterogeneity in the context of the computational resources on the chip, we now discuss hetergoeneous communication fabrics for dark silicon processors. In particular, we will describe a novel NoC architecture, named *darkNoC* [2], where multiple network layers consisting of architecturally identical routers, but optimized to operate within different voltage and frequency ranges during synthesis are used. Only one network layer is active at a given time while the rest of the network layers are dark (deactivated). We will show that a heterogeneous NoC is another way to

leverage the "spare" transistors on a dark silicon chip to either increase performance within a TDP. Alternatively, it can be used to reduce communication power without sacrificing communication latency and throughput, thus providing a larger share of the TDP to the computational components.

Most fabrication foundries characterize cell libraries for various gate threshold voltage (Vt) values such as normal Vt (NVt), Low Vt (LVt), and High Vt (HVt). LVt cells can switch at a much faster speed than HVt cells. However, LVt cells can be up to 5× leakier than their HVt counterparts. Modern CAD tools exploit the power-delay characteristics of multi-Vt cell libraries and slacks in path delays to synthesize power efficient circuits [13].

We exploited the multi-Vt circuit optimization available in CAD tools to synthesize architecturally identical NoC routers for a set of target VF levels: [1GHz, 0.9V], [750 MHz, 0.81V], [500 MHz, 0.81V] and [250 MHz, 0.72V]. Figure 4 reports the network power for operation at [500 MHz, 0.81V] and [250 MHz, 0.72V]. We can observe that for operation at [500 MHz, 0.81V], the NoC designed particularly for [500 MHz, 0.81V] VF level is on average 35% and 16% more power efficient than applying DVFS on a NoC designed for [1GHz, 0.9V] and [750 MHz, 0.81V], respectively. Similar observations can be made for other modes of operation as well. This observation shows that, unlike traditional NoC with a single layer of routers, it may be beneficial in terms of power to have multiple layers of routers in a NoC such that each layer is optimized for a particular VF level.

The darkNoC contains different logical network layers, where each layer is optimized at design-time to operate in a certain VF range. That is, multi-Vt circuit optimization of CAD tools is used to optimize all the routers of a network layer for a particular VF range. All the layers in the darkNoC are managed by a hardware-based darkNoC Layer Manager (*dLM*). The function of the *dLM* is to switch

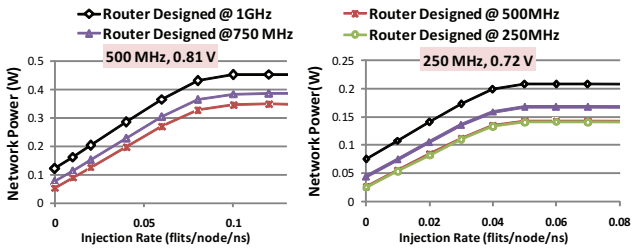

Figure 4: NoC Power for Transpose Traffic for a)(left)[500 MHz, 0.81V] VF level, b)(right) [250 MHz, 0.72V] VF level [2]

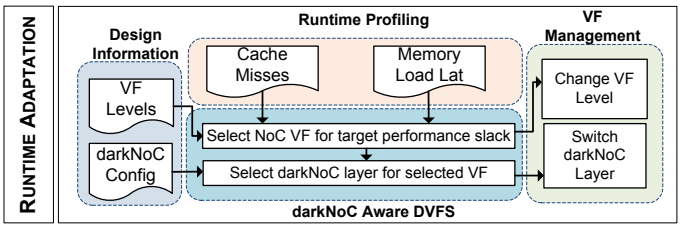

Figure 5: Overview of Runtime darkNoC management [2]

between network layers when directed by the system-level DVFS manager. At each network node, multiple routers are managed by a local hardware-based darkRouter Manager (*dRM*) which controls the power-gating and port enabling signals for each router.

The switch-over between network layers is an important design requirement for our darkNoC architecture. The main challenges are: a) the lossless data communication property of packet-switched buffered NoC should be preserved, b) the switch-over mechanism should be transparent to software, and c) the switch-over mechanism should be efficient in terms of time and energy overhead. In our solution, the darkNoC Layer Manager (*dLM*) and the darkNoC Router Managers (*dRMs*) autonomously coordinate with each other to realize a switch-over mechanism with the aforementioned requirements. In fact, on average, the switch-over procedure on average takes only 200 and 600 cycles in 4×4 and 8×8 mesh NoC, respectively.

At runtime, the NoC power manager monitors various application characteristics to decide the required VF level. If the NoC power manager decides to switch to *i-th* VF level and there is a network layer optimized for *i-th* VF level, then the NoC power manager initiates a switch-over to that particular network layer. On the other hand, if there is no network layer optimized for *i-th* VF level, then the NoC power manager can decide to switch-over to the network layer optimized for the closest yet higher VF level, and will scale VF of the selected network layer. For example, in darkNoC1 configuration (see Fig. 6), if the NoC power manager decides to operate NoC at [250 MHz, 0.72V], then the [750 MHz, 0.81V] network layer will be scaled down to operate at [250 MHz, 0.72V] rather than the [1 GHz, 0.9V] network layer.

For evaluation, we used different NoC configurations in our full system simulations, which are as follows:

- *baselineNoC:* Traditional NoC with router designed for [1Ghz,0.9V]
- *darkNoC1:* darkNoC with 2 VF-optimized network layers for [1GHz, 0.9V] and [750MHz, 0.81V]
- *darkNoC2:* darkNoC with 2 VF-optimized network layers for [1GHz, 0.9V] and [500MHz, 0.81V]
- *darkNoC3:* darkNoC with 3 VF-optimized network layers for [1GHz, 0.9V], [500MHz, 0.81V], and [500MHz, 0.81V]

For darkNoC evaluation, we performed experiments on a 16-core mesh NoC-based dark silicon manycore processor. We used two step system simulation methodology where memory access trace of each application executing on a processor is collected from Xtensa instruction set simulator. These memory access traces are then simulated through a closed loop cycle-accurate NoC and DRAM simulator. Our NoC simulator also modeled different VF levels accurately for the NoC. We used eight applications from Mediabench suite and created diverse multi-programmed application mixes **(AM)**. We created two designs of NoC power managers based upon the application requirements: *DVFS-1 with a target performance loss of 15%* and *DVFS-2 with a target performance loss of 10%.* based on technique introduced by Chen et al. [3].

Fig. 6 reports the savings in NoC EDP for the four application mixes, four NoC configurations discussed above and two NoC power managers. Overall, darkNoC configurations provide significant improvement in EDP over baselineNoC, indicating that a significant increase in energy efficiency can be obtained at the expense of NoC transistor count and sil-

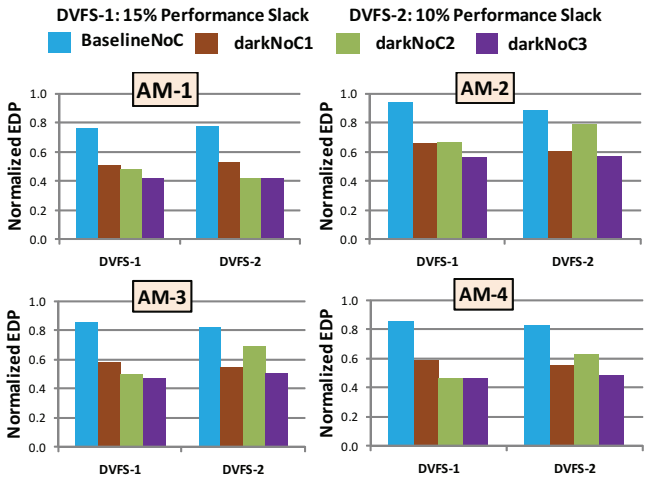

Figure 6: darkNoC Energy-Delay Product (EDP) normalized with respect to a baseline NoC operating at highest VF level [2].

icon area. However, since dark silicon architectures are primarily power and thermally constrained, such a trade-off is indeed desirable.

2.3 Run-time Power Management for Heterogeneous Dark Silicon Processors

Runtime management of heterogeneous dark silicon manycore processors is notably more complex compared to the homogeneous manycore architectures simply because of multiple iso-TDP run-time contexts and the spectrum of choices available for power-performance optimization as well as the thermal constraints. The ultimate objective of an overarching solution that would work for a generic heterogeneous dark silicon processor as shown in Fig. 1 presents several exciting research opportunities for the next decade.

To give an idea of the kinds of solutions that are required, we focus in this section on so-called *clustered* processor architectures. Such processors consist of multiple clusters of processing elements — each cluster has processing elements of the same type, but the clusters are different from each other. For example, the "accelerator tile" in Figure 1 can be thought of as a clustered processor, with two clusters of general purpose cores, a cluster of accelerators and a GPU cluster. Each cluster has its own voltage island.

5The Cortex-A7 cluster, Cortex-A15 cluster, and the GPU each have their own independent voltage islands.

The heterogeneity of the cores along with the voltage-frequency setting of each cluster opens up a gamut of choices and trade-offs for efficient application execution. Let us first closely investigate the constraints that need to be enforced before proceeding to detail the available mechanisms or choices for power management. First, the dark silicon era implies that the chip has to operate under a strict TDP constraint; running all the clusters at the highest frequency level will violate the safe thermal thresholds. Thus the power budget has to be judiciously allocated to the different clusters at runtime depending on the operating conditions and the current applications. Secondly, the applications, especially on mobile platforms such as smartphones, demand a certain performance level or quality-of-service (QoS) that need to be satisfied as best as possible while maintaining the power below the TDP. Finally, for battery-operated devices, energy is a first-class design consideration. These constraints render the runtime management decidedly challenging.

We now introduce the knobs exposed for power management in a heterogeneous architecture. First of all, we may employ per-cluster dynamic power management (DPM) and dynamic voltage-frequency scaling (DVFS). With DPM, a cluster is switched to low-power state when idle leading to drastically reduced energy consumption. But switching to/from low-power state incurs non-negligible time and energy overhead and hence such state switching should be performed with care. DVFS allows the voltage and the clock frequency to be set to one of the available discrete levels to trade time for energy. The power budget for each cluster essentially determines its power state and the frequency level. Thus we need a coordinated power management strategy across the clusters so as to meet the performance demands of the currently executing applications under the thermal constraint. For example, [3] [24] demonstrates the advantages of a collaborative CPU-GPU DVFS management approach as opposed an independent approach in the context of high-end mobile 3D games.

The core-level functional and power/performance heterogeneity present additional mechanisms for power-performance trade-off. For example, a programming framework such as OpenCL [35] enables collaborative execution of a single data-parallel kernel across the CPU and the GPU [23]. The runtime layer needs to orchestrate the execution by partitioning the workload between the CPU and the GPU so as to achieve the best energy-performance objective while respecting the thermal constraints. The dark silicon aware runtime management system is also responsible for mapping each task to the most appropriate core (small or big) at runtime. Finally, an application may have distinct phases that may benefit from different core complexity and the runtime system should perform migration to take advantage of the heterogeneity in improving energy-efficiency.

In summary, the runtime power management of a heterogeneous dark silicon processor involves task partitioning, task mapping, and task migration in conjunction with DVFS and DPM per-cluster with the objective of maximizing energy-efficiency of the entire system while enforcing TDP and QoS constraints.

The first step towards identifying the appropriate core that fits an application or the phase of an application is to estimate the power-performance behavior of the application on cores with different micro-architectural complexity and at different voltage-frequency level. This estimation is challenging, as the cores can be dramatically different in terms of micro-architecture — not just in the pipeline organization but also in terms of memory hierarchy and the branch predictors. A solution is proposed in [25] overcomes these challenges through a combination of static (compile time) program analysis, mechanistic modeling, which builds an analytical model from an understanding of the underlying architecture, and empirical modeling, which employs statistical inference techniques like regression to create an analytical model.

Given the power-performance estimation models, the operating system (runtime layer) needs to make decisions re-

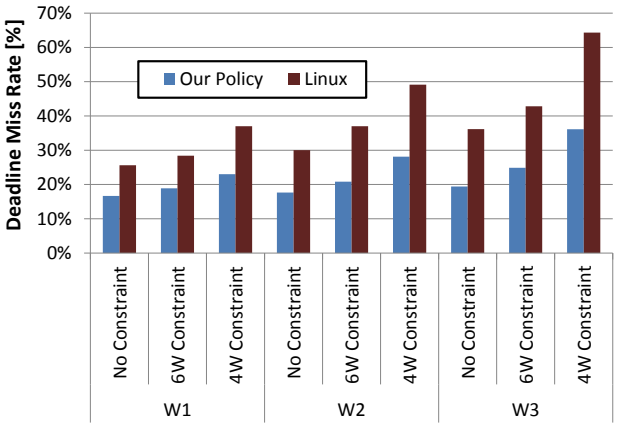

Figure 7: Dark silicon TDP constraint aware power management policy ("Our Policy") compared to the stock Linux scheduler for different workloads and different TDP constraints.

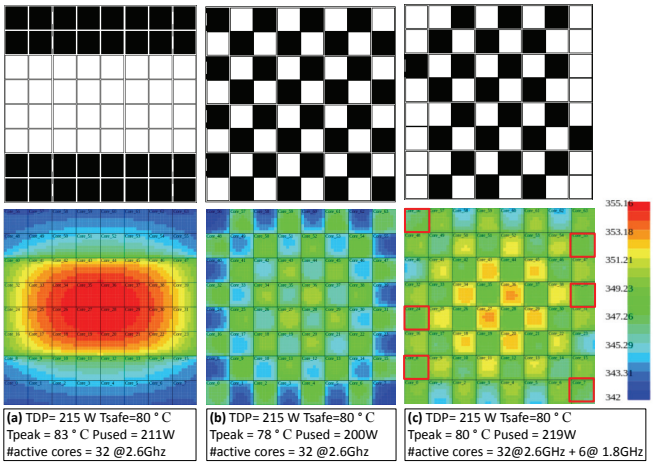

Figure 8: Dark Silicon Patterning: Illustrating the impact of dark silicon decisions on the thermal profile and performance boosting of the chip.

garding the initial mapping of an application to the appropriate core as well as migrating the application across cores in case of phase-change behaviour within the application at a later point in time. The task mapping and migration have to be carefully orchestrated with the cluster-level DPM and DVFS to reach the full performance potential of the application without transgressing the thermal bounds. A hierarchical control-theory based power management framework is introduced in [21] that employs multiple PID controllers (one for each cluster and one for each application) in a synergistic fashion and manages to achieve optimal power-performance efficiency while respecting the TDP budget. The drawback of this approach is the poor scalability with increasing number of clusters as a centralized component allocates the power budget to the different clusters. To solve these issues, a comprehensive, unified, distributed, and scalable power management strategy is proposed in [20]. It is based on price theory that strictly follows the supply-demand based market mechanisms to select the core for each task and the frequency-level for each cluster. When the supply from the core (defined by the core type and its frequency) is equal to the demand of the task (defined in terms of QoS), the system reaches stability and is working at the most energy-efficient point. Otherwise, frequency scaling and/or task migration have to be invoked to achieve the supply-demand equilibrium.

Fig. 7 shows the impact of the runtime management layer that is aware of the TDP constraints imposed by the dark silicon era. We compose four different workloads (W1, W2, W3, W4) consisting of multiple soft real-time applications running on heterogeneous multi-core architecture with three simple ARM Cortex A7 cores and two complex ARM Cortex A15 cores. The figure shows the average deadline miss rate of the workload with standard Linux that is not aware of the TDP constraint and our modified version of Linux that schedules workload and manipulates frequency based on the TDP constraint. We experiment with no thermal constraint and TDP constraint of 4W and 6W, respectively. When the total power exceeds the TDP constraint, the system automatically powers down the cores to keep the power within the constraint. Clearly, the stricter the TDP constraint, the higher the deadline miss rates for both policies. But our TDP-aware runtime management fares much better in meeting the performance demand compared to stock Linux that is agnostic to the TDP constraint. This demonstrates the urgent need for sophisticated runtime management policies for the dark silicon processors

2.4 Run-Time Thermal Management for Dark Silicon Processors

As mentioned above, new run-time power management policies need to be devised to ensure that heterogeneous dark silicon processors operate within their TDP budgets. Of course, the TDP constraint itself (in units of power consumption) is merely a conservative way of ensuring that the maximum chip temperature does not exceed safe limits. As opposed to dynamic power management, dynamic thermal management directly tries to ensure that the chip temperature always remains below the safe temperature, T_{safe}.

Although dynamic thermal management has been extensively studied for conventional chips, the problem acquires an added dimension in the dark silicon era. Particularly, unlike conventional chips where only one TDP mode is available (i.e. all cores are powered-on at full voltage-frequency), dark silicon processors exhibit multiple TDP modes (i.e. a different set of cores may be powered-on) that result in starkly different thermal profiles. To illustrate this fact, we first define *Dark Silicon Patterning* as the spatial and temporal shutdown of on-chip resources while aiming at minimizing peak temperature and respecting the TDP constraint. Our experimental analysis in Figure 8 (for a 64-core chip, TDP=215 W) illustrate that different dark silicon patterns result in different thermal profiles even for the same set of concurrently execution applications and same number of active cores at the full voltage-frequency setting. This is because of the improved heat dissipation due to the "dark cores". A reduced peak temperature allows supplying more power (i.e. beyond TDP) while still keeping the temperature within the safe operating limits. Note that TDP is specified regardless the number and positions of active cores. Therefore, differ-

ent application mapping decisions together with dark silicon patterning also affect the chip thermal profiles.

Fig.8 shows three different thermal profiles generated by different mappings. In the first two cases, i.e. Fig.8 (a) and (b), only 32 cores are powered-on while others are kept dark (i.e. power-gated). In case of contiguous dark cores, (Fig.8 (a)), the power density and temperature are high near the chip center. Moreover, even power is within the TDP bounds, the peak temperature violates the safe temperature constraints in this particular experiment. However, a better dark dilicon pattern (Fig.8 (b)) alleviates the power densities through efficient heat dissipation and contributes towards lowering the peak temperature. The temperature headroom is then exploited to power-on more cores to boost the performance as shown in Fig.8 (c). It is important to note that in case of dependent threads, decision of dark silicon patterning needs to account for the communication overhead between different threads in a distributed memory paradigm.

In short, dark silicon Patterning introduces new opportunities to optimize the thermal profile and/or performance boosting by choosing amongst one of many available TDP modes. However finding an appropriate dark silicon pattern and corresponding application mapping are open research problems.

The above discussion and experiments in Fig.8(c) also hint that the traditional notion of TDP specification is too pessimistic, thus requiring for novel power budgeting methods. In [22], we proposed the **Thermal Safe Power** (TSP) as a fundamentally new power budgeting concept which provides safe power constraint values as a function of the number of active cores without triggering DTM and keeping temperature within safe operating limits. It alleviates the pessimistic bounds of TDP and thereby enables hardware/software designers to explore new techniques for performance improvements at different abstraction layers in the dark silicon era. In [22], we also compare it with Intel's Turbo Boost over a constant power budget per-chip [11, 31]. The algorithms to compute TSP are implemented as an open-source tool available for download[1].

2.5 Addressing Reliability and Variability

Although it is typically assumed that dark silicon processors will be provisioned with heterogeneous accelerators and cores to improve performance and energy-efficiency, the abundance of transistors on the chip can also be exploited to enhance reliability and address the impact of process variations.

Looking at reliability first, we note that the conventional solution to protect the execution of an application from soft-errors is the use of TMR at the architecture level. However, providing full-scale redundancy incurs significant power overhead. In our previous studies [28, 29, 30, 33] we have shown that different applications exhibit dissimilar instruction profiles and correspondingly different vulnerabilities to soft errors. Moreover, due to their varying data and control flow properties these applications have distinct inherent resilience, i.e., error masking properties. Therefore, not all applications require full TMR and it may be beneficial to design iso-ISA cores with different reliability, power/performance, and area properties. These so-called *reliability-heterogeneous cores* provide power versus reliability tradeoffs and range from a fully-protected core to partially-protected cores (i.e. only pipeline, register file or cache or a combination of these is protected) to the unprotected/baseline core as shown in Figure 9. In [15, 16, 17, 18] we have developed different designs of such reliability-heterogeneous cores that can be used to generate a library.

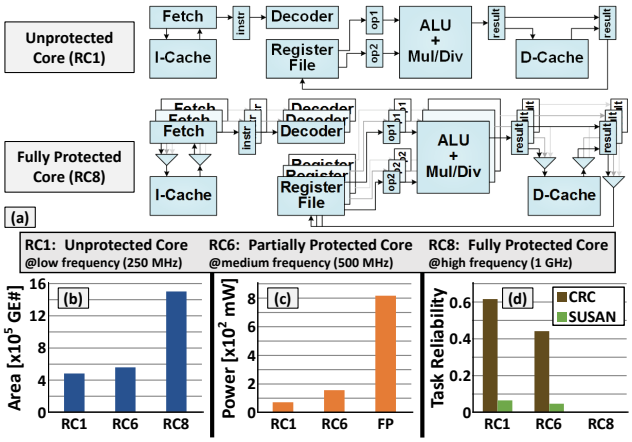

Figure 9: Comparison of iso-ISA reliability-heterogeneous cores (LEON3) without protection and with full protection [15].

Given a library of such reliability-heterogeneous cores, the first challenge is the architectural synthesis challenge similar to the one described in Section 2.1, but with an added focus on reliability. In [15], we have formulated this problem as a Bounded Knapsack Problem and developed a polynomial time algorithm to synthesize target processors. Fig. 10 (left-side) shows two steps for obtaining an example example *darkRHP* template.

The next challenge is to design a dark silicon aware run-time system that dynamically manages the reliability of concurrently executing multi-threaded applications under the TDP/thermal constraints. Fig. 10 (see right-side) illustrates example run-time scenarios where only a subset of reliability-heterogeneous cores is powered-on, while other cores are kept *dark* — we describe the design of such a run-time manager in [15]. Our experimental results show that that significant improvement in reliability can be obtained using the proposed techniques.

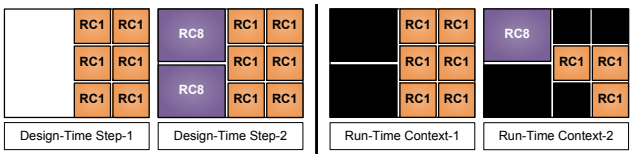

Figure 10: Example for design-time customization (left) and different run-time contexts with two types of cores (right).

Besides reliability, the availability of spare transistors in the dark silicon era provides another opportunity to address a major problem with technology scaling — the impact of manufacturing process variations. As a consequence of process variations, even identical cores on the same chip can have very different leakage power consumption and operating frequency [5]. In other words, process variations introduce unintended heterogeneity between cores on the same chip. By provisioning a chip with *more* cores than can be simultaneously powered on, one can therefore pick the *best*

[1] http://ces.itec.kit.edu/download

subset of cores that maximized performance within the chip TDP, while keeping the others dark. We refer to this idea as *cherrypicking* [27]. In our previous work, we have shown that by over-provisioning a chip with redundant cores, more than 30% increase in performance is achievable over a large suite of multi-threaded benchmark applications.

3. CONCLUSION

In this paper, we have highlighted some key challenges that must be addressed to mitigate the so-called dark silicon problem, a potentially major hurdle for future technology scaling and transistor integration. In particular, we have focused on the design, automated synthesis and runtime management of heterogeneous dark silicon processor architectures, and highlighted some early research efforts that attempt to leverage heterogeneity in order to increase performance, energy-efficiency and reliability within Thermal Design Power (TDP) and safe operating temperature (T_{safe}) constraints. Nonetheless, several fundamental challenges still remain to be addressed, and it is our belief that the hardware/software co-design and systems synthesis community is key in solving these challenges.

4. ACKNOWLEDGMENTS

This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre "Invasive Computing" (SFB/TR 89); http://invasic.de, and the National Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Ph.D. students of their labs (Santiago Pagani, Heba Khdr Florian Kriebel, Semeen Rehman, Bharathwaj Ragunathan, Haseeb Bokhari, Haris Javaid) for assistance with experiments.

5. REFERENCES

[1] Jason Allred, Sanghamitra Roy, and Koushik Chakraborty. Designing for dark silicon: a methodological perspective on energy efficient systems. In *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2012.

[2] Haseeb Bokhari, Haris Javaid, Muhammad Shafique, Jörg Henkel, and Sri Parameswaran. darknoc: Designing energy-efficient network-on-chip with multi-vt cells for dark silicon. In *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*, pages 1–6. ACM, 2014.

[3] Xi Chen, Zheng Xu, Hyungjun Kim, Paul V. Gratz, Jiang Hu, Michael Kishinevsky, Umit Ogras, and Raid Ayoub. Dynamic voltage and frequency scaling for shared resources in multicore processor designs. In *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, pages 114:1–114:7, 2013.

[4] Jason Cong, Mohammad Ali Ghodrat, Michael Gill, Beayna Grigorian, and Glenn Reinman. Architecture support for accelerator-rich cmps. In *Proceedings of the ACM 49th Annual Design Automation Conference (DAC)*, 2012.

[5] Saurabh Dighe, Sriram R Vangal, Paolo Aseron, Shasi Kumar, Tiju Jacob, Keith A Bowman, Jason Howard, James Tschanz, Vasantha Erraguntla, Nitin Borkar, et al. Within-die variation-aware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-core teraflops processor. *Solid-State Circuits, IEEE Journal of*, 46(1):184–193, 2011.

[6] Hadi Esmaeilzadeh, Emily Blem, RenÃl'e St. Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 365 –376, 2011.

[7] Nikos Hardavellas, Michael Ferdman, Babak Falsafi, and Anastasia Ailamaki. Toward dark silicon in servers. *Micro, IEEE*, 31(4):6–15, 2011.

[8] Jörg Henkel, Lars Bauer, Nikil Dutt, Puneet Gupta, Sani Nassif, Muhammad Shafique, Mehdi Tahoori, and Norbert Wehn. Reliable on-chip systems in the nano-era: Lessons learnt and future trends. In *DAC*, 2013.

[9] Jörg Henkel, Lars Bauer, Hongyan Zhang, Semeen Rehman, and Muhammad Shafique. Multi-layer dependability: From microarchitecture to application level. In *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*, DAC '14, pages 47:1–47:6, 2014.

[10] Mark D Hill and Michael R Marty. Amdahl's law in the multicore era. *IEEE Computer*, 41(7):33–38, 2008.

[11] Intel Corporation. Dual-core intel xeon processor 5100 series datasheet, revision 003, August 2007.

[12] Brian Jeff. Advances in big.little technology for power and energy savings. 2012.

[13] Tanay Karnik, Yibin Ye, James Tschanz, Liqiong Wei, Steven Burns, Venkatesh Govindarajulu, Vivek De, and Shekhar Borkar. Total power optimization by simultaneous dual-vt allocation and device sizing in high performance microprocessors. In *Design Automation Conference, 2002. Proceedings. 39th*, pages 486–491, 2002.

[14] Himanshu Kaul, Mark Anders, Steven Hsu, Amit Agarwal, Ram Krishnamurthy, and Shekhar Borkar. Near-threshold voltage (ntv) design: opportunities and challenges. In *Proceedings of the 49th Annual Design Automation Conference*, pages 1153–1158. ACM, 2012.

[15] Florian Kriebel, Semeen Rehman, Duo Sun, Muhammad Shafique, and Jörg Henkel. Aser: Adaptive soft error resilience for reliability-heterogeneous processors in the dark silicon era. In *Design Automation Conference (DAC)*, 2014.

[16] Tuo Li, Muhammad Shafique, Jude Angelo Ambrose, Semeen Rehman, Jörg Henkel, and Sri Parameswaran. Raster: runtime adaptive spatial/temporal error resiliency for embedded processors. In *DAC*, page 62, 2013.

[17] Tuo Li, Muhammad Shafique, Semeen Rehman, Jude Angelo Ambrose, Jörg Henkel, and Sri Parameswaran. DHASER: dynamic heterogeneous adaptation for soft-error resiliency in ASIP-based multi-core systems. In *ICCAD*, pages 646–653, 2013.

[18] Tuo Li, Muhammad Shafique, Semeen Rehman, Swarnalatha Radhakrishnan, Roshan G. Ragel, Jude Angelo Ambrose, Jörg Henkel, and Sri Parameswaran. Cser: Hw/sw configurable soft-error resiliency for application specific instruction-set

processors. In *DATE*, pages 707–712, 2013.

[19] Michael J. Lyons, Mark Hempstead, Gu-Yeon Wei, and David Brooks. The accelerator store: A shared memory framework for accelerator-based systems. *ACM Trans. Archit. Code Optim.*, 8(4):48:1–48:22, 2012.

[20] Thannirmalai Somu Muthukaruppan, Anuj Pathania, and Tulika Mitra. Price theory based power management for heterogeneous multi-cores. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2014.

[21] Thannirmalai Somu Muthukaruppan, Mihai Pricopi, Vanchinathan Vanchinathan, Tulika Mitra, and Sanjay Vishin. Hierarchical power management for asymmetric multi-core in dark silicon era. In *Design Automation Conference (DAC)*, 2013.

[22] Santiago Pagani, Heba Khdr, Waqaas Munawar, Jian-Jia Chen, Muhammad Shafique, Minming Li, and Jörg Henkel. TSP: Thermal Safe Power - efficient power budgeting for many-core systems in dark silicon. In *Proceedings of the IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2014.

[23] Prasanna Pandit and R. Govindarajan. Fluidic kernels: Cooperative execution of opencl programs on multiple heterogeneous devices. In *International Symposium on Code Generation and Optimization (CGO)*, 2014.

[24] Anuj Pathania, Qing Jiao, Alok Prakash, and Tulika Mitra. Integrated cpu-gpu power management for 3d mobile games. In *Design Automation Conference (DAC)*, 2014.

[25] Mihai Pricopi, Thannirmalai Somu Muthukaruppan, Vanchinathan Venkataramani, Tulika Mitra, and Sanjay Vishin. Power-performance modeling on asymmetric multi-cores. In *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*, 2013.

[26] Bharathwaj Raghunathan and Siddharth Garg. Job arrival rate aware scheduling for asymmetric multi-core servers in the dark silicon era. In *Proceedings of the IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2014.

[27] Bharathwaj Raghunathan, Yatish Turakhia, Siddharth Garg, and Diana Marculescu. Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 39–44. EDA Consortium, 2013.

[28] Semeen Rehman, Muhammad Shafique, Pau Vilimelis Aceituno, Florian Kriebel, Jian-Jia Chen, and Jörg Henkel. Leveraging variable function resilience for selective software reliability on unreliable hardware. In *DATE*, pages 1759–1764, 2013.

[29] Semeen Rehman, Muhammad Shafique, Florian Kriebel, and Jörg Henkel. Reliable software for unreliable hardware: embedded code generation aiming at reliability. In *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 237–246, 2011.

[30] Semeen Rehman, Anas Toma, Florian Kriebel, Muhammad Shafique, Jian-Jia Chen, and Jörg Henkel. Reliable code generation and execution on unreliable hardware under joint functional and timing reliability considerations. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 273–282, 2013.

[31] Efi Rotem et al. Power-management architecture of the intel microarchitecture code-named sandy bridge. *IEEE Micro*, 32(2):20–27, 2012.

[32] Muhammad Shafique, Siddharth Garg, Jörg Henkel, and Diana Marculescu. The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives. In *Design Automation Conference (DAC)*, 2014.

[33] Muhammad Shafique, Semeen Rehman, Pau Vilimelis Aceituno, and Jörg Henkel. Exploiting program-level masking and error propagation for constrained reliability optimization. In *DAC*, page 17, 2013.

[34] Youngmin Shin et al. 28nm high- metal-gate heterogeneous quad-core cpus for high-performance and energy-efficient mobile application processor. In *International Solid-State Circuits Conference (ISSCC)*, 2013.

[35] John E. Stone, D. Gohara, and G. Shi. Opencl: A parallel programming standard for heterogeneous computing systems. In *Computing in science and engineering*, volume 12.3, 2010.

[36] M. Taylor. Is dark silicon useful?: harnessing the four horsemen of the coming dark silicon apocalypse. In *Proceedings of the 49th ACM Annual Design Automation Conference (DAC)*, pages 1131–1136, 2012.

[37] Y. Turakhia et al. Hades: Architectural synthesis for heterogeneous dark silicon chip multi-processors. In *Proceedings of the 50th ACM Design Automation Conference (DAC)*, 2013.

[38] Yatish Turakhia, Bharathwaj Raghunathan, Siddharth Garg, and Diana Marculescu. Hades: architectural synthesis for heterogeneous dark silicon chip multi-processors. In *Proceedings of the 50th Annual Design Automation Conference*, page 173. ACM, 2013.