POLITECNICO DI MILANO
DEPARTMENT ELECTRONICS, INFORMATION AND BIOENGINEERING
DOCTORAL PROGRAM IN INFORMATION TECHNOLOGY

NEAR-THRESHOLD COMPUTING WITH PERFORMANCE GUARANTEES FOR MANYCORE ARCHITECTURES

Doctoral Dissertation of:
Ioannis Stamelakos

Supervisor:
Prof. Cristina Silvano

Co-advisor:
Dr. Sotirios Xydis

Tutor:
Prof. Francesco Amigoni

The Chair of the Doctoral Program:
Prof. Andrea Bonarini

2016 – PhD Cycle XXVIII

Abstract

The power-wall problem caused by the stagnation of supply voltages in deep-submicron technology nodes, is now the major scaling barrier for towards the manycore era. Although the technology scaling enables extreme volumes of computational power, power budget violations will permit only a limited portion to be actually exploited, leading to the so called Dark Silicon. Near-Threshold voltage Computing (NTC) has emerged as a promising approach to overcome the manycore power-wall, at the expense of reduced performance values and higher sensitivity to process variation. Given that several application domains operate over specific performance constraints, the performance sustainability is considered a major issue for the wide adoption of NTC. Thus, in this thesis, we investigate how performance guarantees can be ensured when moving towards NTC manycores through variability-aware voltage and frequency allocation schemes. We propose different aggressive NTC voltage tuning and allocation strategies, showing that performance can be efficiently sustained or even optimized at the NTC regime and we show that NTC advantages and gains highly depend on the underlying workload characteristics. However, given the increased impact of variability in NTC, delivering the appropriate voltages can be a very demanding task, thus the power delivery scheme has to be evaluated and optimized. We extend our research and show that when the workload characteristics of the applications are analyzed and considered at runtime, significant power savings can be obtained even when using existing, cost-effective power delivery techniques, while meeting the application performance constraints imposed in the first place. Finally, we make

a first attempt to make show that, in NTC, there is ample place for optimizations at runtime that can provide extra energy savings: by proposing a lightweight runtime algorithm for balancing throughput under process and workload variability we managed to gain significant power savings without impacting performance.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The goal of this chapter is to introduce the reader to Near-Threshold Computing and the problems it addresses, to provide him with the necessary background, present the state of the art in this domain as well as to provide a summary of the thesis and its contributions.

1.1 Motivation

The continuous technology scaling, predicted and expressed by Moore's law [46], has brought many paradigm shifts in processor design: from the single core processor and the frequency scaling race to multicore processors and recently to manycore architectures [16], [15], [59], which are considered to be the principal strategy for continuing performance growth. However, the end of Dennard's scaling [8] has brought designers in front of the so called power/utilization wall i.e. not all resources can be used concurrently due to power and thermal constraints. Projections show that the gap between the number of cores integrated on a chip and the number of cores that can be utilized will continue to widen on future technology nodes [14]. As a result, *dark silicon* - transistor under-utilization due to power budget constraints - has recently emerged as major design challenge that jeopar-

dizes the well established core count scaling path in current and future chip generations.

To address the dark silicon problem, researchers have proposed techniques at the micro-architectural level [23], [24], [60] down to physical and device level [53], [49]. Near-Threshold Voltage Computing (NTC) [11] has been proposed as a promising technique to mitigate the effects of dark silicon, allowing a large number of cores to operate under a given manycore power envelope. NTC takes advantage of the quadratic relation between the supply voltage (V_{dd}) and the consumed power, by lowering the operating V_{dd} to a value slightly higher than the transistor threshold voltage (V_{th}). In comparison with the conventional Super-Threshold Voltage Computing (STC), computations at the NT regime are performed in a very energy efficient manner, unfortunately at the expense of reduced performance and high susceptibility to parametric process variations. Actually, frequency drops exponentially in the near-threshold region but we can compensate for the performance losses by taking advantage of the available resources (extra cores) and the inherent application parallelism. The problem with the latter is that the application needs to be able to scale, in terms of performance, with the increasing number of cores allocated to it, otherwise this will lead us to consume more energy than before due to the extra resources being used. The second major challenge that has to be tackled is the increased sensitivity to process variation when the circuit is operating in the near-threshold regime. Variability is a known phenomenon and refers to the fluctuation of the chip's parameters from its nominal values. Thus, the designer of an NT chip has to be very careful so that the behavior will be the desired one and the yield will be the highest possible. Additionally, it is not clear which is the best power management scheme to be used when dealing with an NT circuit. There are chips that can use only a single supply voltage and take advantage of the frequency scaling and others that can scale both voltage and frequency. To make the design space even larger there is the possibility to split the chip into small clusters that include more than one cores or even to perform the V/F control independently for each core.

1.2 NTC: Background and Challenges

Near-threshold voltage operation relies on the aggressive tuning of the V_{dd} of the integrated circuit very close to the transistors' threshold voltage V_{th}, to a region where V_{dd} is still slightly higher than V_{th}. This decrease of the supply voltage increases the potential for energy efficient computation, e.g.

by reducing V_{dd} from the nominal 1.1 V to 500 mV, energy gains of $10\times$ are reported [11]. While near-threshold region is not more power efficient than the theoretical limits provided in the sub-threshold/ultra-low voltage region, where $V_{dd} < V_{th}$, NTC has gained a lot of attention due to low energy operation at higher performance and easier adoption across multiple application platforms in comparison with the sub-threshold circuits. NT is the region that delivers interesting trade-offs regarding energy efficiency and transistor delay, since super-threshold V_{dd} quickly reduces energy efficiency while sub-threshold V_{dd} leads to severely slower transistors. However, NTC, as mentioned earlier, comes together with two major drawbacks, i.e. reduced performance and increased sensitivity to process variations.

Performance reduction at NTC is manifested through the restricted maximum achievable clock frequency. This is an implicit effect due to the reduction of the $V_{dd} - V_{th}$ difference, applied when moving to the NT region. Performance degradation can be compensated by exploiting trade-off points of higher task parallelism at lower clock frequencies. Thus, an important open question for NTC to be examined is the following: *Is the inherent parallelism of the existing applications enough to retain the performance levels of super-threshold design with lower power consumption, making it, therefore, worth going to near-threshold operation?* Pinckey et al. [50] studied the limits of voltage scaling together with task parallelization knobs to address the performance degradation at NTC by considering a clustered micro-architectural template with cores sharing the local cache memory, derived from [12]. They showed that under realistic application/architecture/technology features (i.e. parallelization efficiency, inter-core communication, V_{th} selection etc.) the theoretical energy optimum point ($\frac{dEnergy}{dV_{dd}} = 0$), moves from the sub-threshold to the near-threshold region. Considering a single supply voltage per die, the energy optimum point can be found within an interval of 200 mV higher than V_{th}, implicitly defining, in that way, the upper limits of the NTC region.

The second important challenge for manycore architectures operating at the NTC regime is their increased sensitivity to process variations. The transistor delay is heavily affected by the variation of V_{th} at NT voltages compared to the one in super-threshold voltages [13], [42]. In addition, failure rate of conventional SRAM cells is increased in low voltage operation [6]. As a consequence, the operating frequency of the cores varies considerably, reducing the yield. In addition, variation's effects on the total power of the chip have to be carefully considered, due to the exponential dependency of leakage current upon V_{th}. Karpuzcu et al. presented Varius-NTV [32], a micro-architectural model that adopts proper gate-delay and

SRAM cell type models to capture the increased sensitivity of manycore chips to process variations at NTC. For mitigating variation effects on performance at NTC, the EnergySmart architecture and thread assignment methodology have been recently proposed in [34]. By assuming on-chip voltage regulators of low-efficiency, EnergySmart adopts single voltage - multiple frequency islands to cope with variability.

The power budget of a manycore platform is a major constraint, thus the design of an efficient power delivery system is crucial. With recent advancements in on-chip voltage regulator (VR) design and implementation [37], it is now possible to design complex power delivery networks supporting fine grained voltage domains. When designing a power delivery system for NTC, the designer has to be extra careful because he has to consider and address issues such as: 1) the impact of the applications scalability regarding the increasing number of cores when running in NTC, 2) the increased current and 3) the many different voltage levels needed ideally by the cores because of the increased variability impact. There have been made some general purpose attempts for fine grained Dynamic Voltage and Frequency Scaling (DVFS), [39], [57] in system level, as well as others that target specifically Near-Threshold Computing: Superrange [26], voltage stacking [38], Booster [44], linear-dropout regulators LDOs [28], [47], which are discussed in more detail in Chapter 4. The main problem is that most of them address small multicore platforms, consisting of 16 up to 32 cores, and none of them validates the scalability of those approaches to manycore systems consisting of hundreds of cores and that they do not offer an insight on how near-threshold voltages can be delivered efficiently in a manycore architecture with hundreds of cores with many different voltage levels.

Manycore processors can provide high throughput for highly parallel workloads, making them ideal candidates for running multi-threaded parallel applications, but they are constrained by their strict power/thermal budgets. Near Threshold Computing (NTC), thus, can be used for this kind of applications running on manycore platforms because, as it has been shown [51], it provides operation at the most energy efficient point and it can compensate for the performance loss by using all the available parallelism. The target performance metric for highly parallel applications is throughput and the main goals are maximizing it given a power constraint or minimizing power consumption while meeting a desired throughput constraint. Some studies have been made in this topic, like in [58] where the authors maximize throughput by balancing power using linear integer programming, or in [30] where throughput maximization is done by creating

a model for predicting the power to frequency relation. Early NT voltage processor prototypes [18], [36] have been recently presented validating the theoretical premises, while several studies show the high efficiency of NTC for cloud- [61] and server-based workloads [48]. Generally, targeting mainly manycore architectures, NTC imposes several challenges regarding the application mapping and the resource/power management due to its increased sensitivity to parametric variation [35].

1.3 Thesis Contributions

The thesis is split in two parts and the main contributions of each one of them are:

Part I

- Introducing a variability-aware framework for exploring the potential power-efficiency of Near Threshold Computing (NTC) while sustaining performance.
- Proposing the utilization of voltage island formation combined with the operation at the near-threshold region as an effective technique for building power efficient manycore architectures that sustain performance values delivered by conventional super-threshold computing.
- Evaluating a set of aggressive NT voltage tuning and allocation strategies, showing that STC performance can be efficiently sustained or even surpassed at the NT regime for both thread-parallel and process-parallel workloads.

Part II

- Evaluating the existing power delivery architectures for near-threshold manycores under process variation.
- Optimizing voltage allocation for each application considering its own characteristics while considering the correlation between the PD system and workload characteristics to fully exploit the NTC benefits.
- Introducing an efficient, low overhead algorithm for sustaining application throughput while improving energy efficiency at NTC.
- Investigating the optimal resource allocation and voltage/frequency assignment for maximizing the energy efficiency of highly parallel applications.

1.4 Thesis Summary

In the current thesis, we explore and assess the benefits of techniques targeting Near-Threshold Computing (NTC) for manycore architectures that run highly parallel applications while meeting strict performance constraints. Building on top of that, we investigate further optimizations that can be done at both the power delivery system and the runtime.

Initially, in **Chapter 2**, we introduce a variability-aware framework for exploring the potential power-efficiency of Near Threshold Computing (NTC) while providing performance guarantees. We propose and analyze the usage of fine-grained voltage islands to cope with the increased effect of variability problem in the NT region and through extensive experimentation, we showed the optimization potentials of moving towards near-threshold voltage computing, exposing its high dependency on both workload characteristics and underlying architectural organization For the considered workloads, we found that the power impact of fine-grained voltage islands formation can be up to 35% for a 128-core chip operating at NTC region, while the adoption of a variability aware technique can bring to a power reduction of up to 43% with respect to a variability unaware technique.

Next, in **Chapter 3**, extending our approach, we propose a set of aggressive NT voltage tuning and allocation strategies, showing that STC performance can be efficiently sustained or even surpassed at the NT regime. More specifically, we show that NTC, depending on the underlying workload characteristics, can deliver average power gains of 65% for thread-parallel workloads and up to 90% for process-parallel workloads, while offering an extensive analysis on the effects of different voltage tuning/allocation strategies and voltage regulator configurations.

In **Chapter 4**, we evaluate and compare the efficiency of different power delivery schemes for NT manycore architectures under process variation while meeting performance constraints. The results indicate that for platforms operating in a predefined voltage range, simple and cost effective Power Delivery (PD) architectures can deliver average power savings ranging from 24% up to 50%, when taking into account the workload characteristics of the target applications at design time.

Finally, in **Chapter 5**.we propose a runtime management scheme for improving NTC manycore energy efficiency. Assuming a feasible, low overhead Power Delivery Network (PDN) for NTC, we propose an algorithm for balancing throughput under process (and workload) variability that sustains performance while reducing power significantly. Considering the power inefficiencies of a scalable and low overhead NT power delivery

network, our algorithm manages to reduce both the variation of the per-core throughput w.r.t. the minimum and the power consumption, on average, by 70% and 43.5% respectively, while not impacting the overall performance.

Part I

Variability-Aware Voltage Island Management for Near-Threshold Voltage Computing With Performance Guarantees

Overview

In this first part, we investigate the power efficiency potential of manycore architectures at the NT regime, considering process variation as well as power delivery architectures supporting multiple voltage domains, under strict performance constraints originated from multicore architectures at the ST regime. Unlike previous works on variation-aware voltage allocation that target the ST regime, we propose the formation of voltage islands (VIs) for the minimization of the impact of within-die variation, which is more evident at NTC, in both performance and power. Then, we show how process variation can be efficiently exploited for boosting the performance of an NT manycore. To support the aforementioned research objectives, an exploration framework for manycore architectures operating at NTC has been developed to investigate the power efficiency under different workloads, while sustaining the performance when moving from the ST to the NT region.

CHAPTER 2

Variation-Aware Voltage Island Formation

2.1 Introduction

In this chapter, we investigate the power efficiency potential of manycore architectures at the near-threshold (NT) regime, considering the process variations as well as a power delivery architecture supporting multiple voltage domains, under strict performance constraints originated from multicore architectures at the super-threshold (ST) regime. Unlike previous works on variation aware voltage allocation that target the ST regime [7], [41], [27], we propose the formation of voltage islands (VIs) for the minimization of the impact of within-die variations, which are more evident at NTC, in both performance and power. In particular, we developed a framework for manycore architectures operating at NTC to investigate the power efficiency under different workloads, while sustaining the performance when moving from the ST to the NT region. The framework has been parametrized in order to exploit different voltage island formations and to deal with variability. Additionally, to generalize the analysis, we study four clustered manycore architectural organizations – differing on the number of cores per cluster.

Extensive experimental analysis showed that for the considered work-

loads, when moving to the NT regime for a 128-core architecture, average power gains close to 65% are delivered while sustaining the performance values obtained by a 16-core architecture at STC. The power impact of fine-grained voltage island formation can be up to 35% for a 128-core chip operating at NTC. Additionally, in comparison with variation unaware techniques, the proposed variation-aware NTC voltage island formation delivers power gain up to 8% considering a single VI per chip and up to 43% when considering the fine-grained multiple VI case, that is able to deal better with variability. Finally, analyzing the V_{dd} distribution at NTC, we demonstrate that the utilization of multiple VIs together with efficient integrated regulators can be considered a feasible option at NTC to efficiently deal with the process variability.

2.2 State of the Art

Near-threshold voltage operation relies on the aggressive tuning of the V_{dd} very close to the transistor threshold voltage V_{th}, to a region where $V_{dd} > V_{th}$ still holds. This decrease of the supply voltage increases the potential for energy efficient computation, e.g. by reducing V_{dd} from the nominal 1.1 V to 500 mV, energy gains of $10\times$ are reported in [11]. NTC delivers interesting trade-offs regarding energy efficiency and transistor delay, since super-threshold V_{dd} quickly reduces energy efficiency while sub-threshold V_{dd} leads to drastically slower transistors. However, NTC comes together with two major drawbacks: *(i)* reduced performance and *(ii)* increased sensitivity to process variations.

Performance reduction at NTC is exposed through the limited maximum achievable clock frequency. This is an implicit effect due to the reduction of the $V_{dd} - V_{th}$ difference, applied when moving to the NT region. Performance degradation can be compensated by exploiting trade-off points corresponding to higher task parallelism at lower clock frequencies. Thus, an important open question for NTC to be investigated is the following: *Is the inherent parallelism of applications enough to retain the performance levels of super-threshold design with lower power consumption, thus making it worth going to near-threshold operation?* Pinckey et al. [50] studied the limits of voltage scaling together with task parallelization knobs to address the performance degradation at NTC by considering a clustered micro-architectural template with cores sharing the local cache memory. They proved that under realistic application/architecture/technology features (i.e. parallelization efficiency, inter-core communication, V_{th} selection, etc.) the theoretical energy optimum point ($\frac{dEnergy}{dV_{dd}} = 0$) moves from

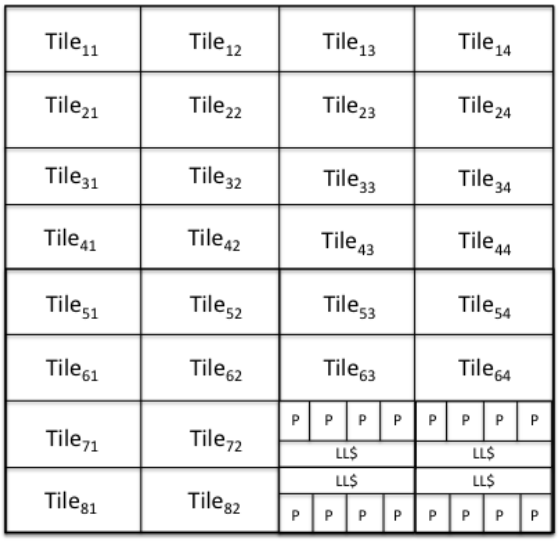
the sub-threshold to the near-threshold region. Considering a single supply voltage per die, the energy optimum point can be found within an interval of 200 mV higher V_{th}, implicitly defining, in that way, the upper limits of the NT region.

The second important challenge for manycore architectures operating at NTC regime is their increased sensitivity to process variations. The transistor delay is heavily affected by the variation of V_{th} at NT voltages compared to the one in super-threshold voltages [13], [42]. In addition, failure rate of conventional SRAM cells is increased in low voltage operation [6], [56]. As a consequence, the operating frequency of the cores varies considerably, reducing the yield. In addition, variation's effects on the total power of the chip have to be carefully considered, due to the exponential dependency of leakage current upon V_{th}.

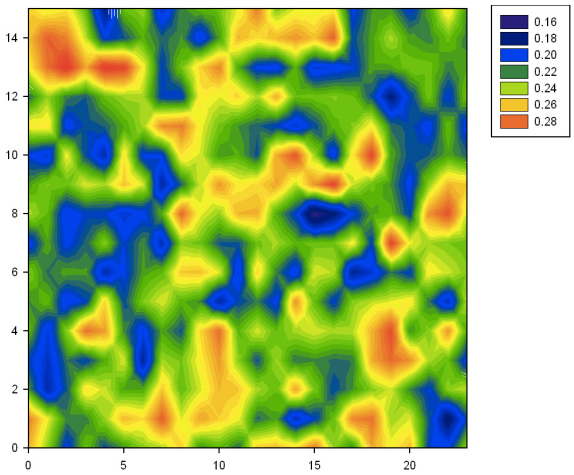
We focus our study on the NTC design space defined by [11] and [34]. Specifically, we target power efficient NT manycore architectures that sustain ST performance levels by considering their increased sensitivity to process variation. Performance sustainability is a critical issue for the adoption of NTC, since best effort approaches are more suitable for managing performance fluctuations due to process variability. In comparison to previous works [11], [34] where only a single system-wide power domain is considered, we differentiate our approach by exploring multiple voltage domain NT architectures through variation-aware voltage island (VI) formation techniques.

2.3 Micro-architecture, Process-Variation and Power Delivery Modelling

Micro-architecture model: We focus our study on tile-based architectures, including the ones proposed in [12] and [34]. Figure 2.1a shows an abstract view of the tile-based manycore architecture, as well as the different intra-tile organizations. We consider four intra-tile architectures by varying the number of cores per tile and the memory configuration of the last level cache (LLC) per tile. Each core owns a private instruction and data cache (P$). The LLC (LL$) is shared among the different cores composing a tile. The Intel Nehalem processor [31] configuration for the core and the P$ has been adopted. While the P$ size remains constant across the different intra-tile configurations, the size of the (LL$) is scaled according to the number of cores in the tiles, keeping the total chip area constant. We use the following abbreviations for differentiating between the manycore architectures, based on four tile types: (i) *S1:* each core owns a Last



(a) *ManyCore Architecture*



(b) V_{th} *variation map*

Figure 2.1: *Tile-based manycore architecture (a) and corresponding* V_{th} *variation map (b).*

Level LL$, (ii) *S2:* LL$ is shared between 2 adjacent cores, (iii) *S4:* LL$ is shared among 4 adjacent cores, (iv) *S8:* LL$ is shared among 8 adjacent cores. The different configurations are depicted in Figure 2.2. While S4 and S8 resemble the cluster organizations proposed in [12], [34], we also explored more fine-grained clusters, i.e. S1 and S2. The tile's type defines the minimum VI granularity supported by each manycore configuration.

Thus, for a Si manycore platform the finest granularity of a voltage domain is i cores per VI.

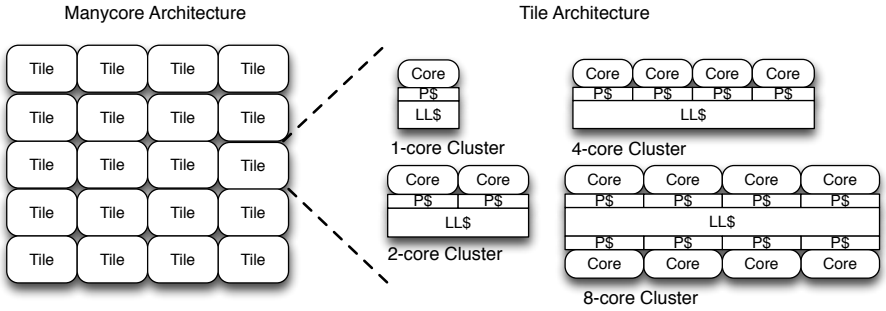


Figure 2.2: *Tile-based manycore architecture and the S1-S8 type of clusters.*

Process variation model: In order to capture the process variation at the NT regime, we integrate the Varius-NTV [32] microarchitectural model within the proposed framework. While Varius-NTV reuses the spherical distance function in [55] for modeling the intra-die spatial correlations, it heavily extends [55] by updating the STC micro-architectural delay and SRAM cell models to reflect in a more accurate manner the higher sensitivity of NTC on process variation. Specifically, it (i) calculates gate-delay following the EKV model [42] , (ii) it incorporates a 8T SRAM cell model for reliable read/write operations at NTC and (iii) it considers a larger set of memory timing and stability failure modes. We used ArchFP [17] to automatically generate the floorplan for the target manycore architectures. Based on the provided manycore floorplan, Varius-NTV generates the corresponding variation maps accounting for the within-die (WID) and die-to-die (D2D) process variations. Figure 2.1b shows a sample instance of its V_{th} variation map.

Assuming B as the set of component blocks found in the floorplan and D the set of dies, we now define $V_{th}^{(i,j)}, i \in B, j \in D$ that corresponds to the V_{th} of the architecture's component i in sample die j. Once extracted, $V_{th}^{(i,j)}$ is used for allocating to each component the lowest possible $V_{dd}^{(i,j)}$ for sustaining f_{NTC} frequency constraint given that:

Power Delivery Architecture: Generally, the power delivery network can be divided into two components:

1. Off-chip network: one or more power supply rails, powered by off-chip voltage regulators, deliver the appropriate voltages to the chip.

2. On-chip network: a second layer, connected with the off-chip net-

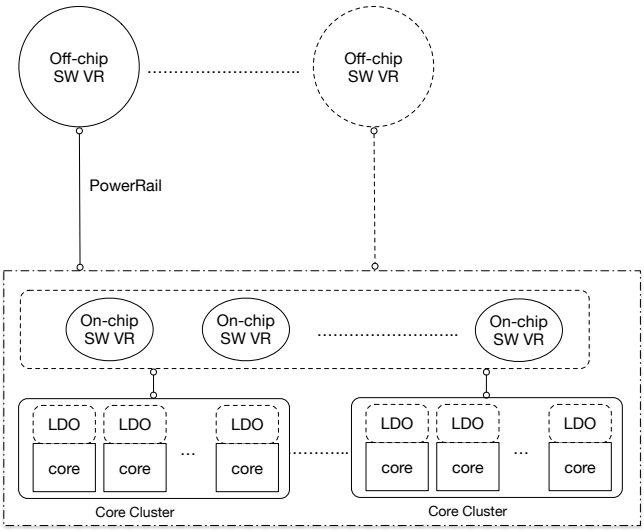


Figure 2.3: *Abstract view of the power delivery architecture.*

work, consisting of voltage regulators that step down the voltage and deliver it to the cores. The VRs considered here are of two types:

- Switching: They have a very good efficiency (~90%) but they consume a lot of area and it is hard to integrate them on chip.
- Low Dropout: An LDO is a linear regulator and its efficiency can be calculated as follows:

$$\eta_{ldo} = \frac{V_{out}}{V_{in}} \tag{2.1}$$

We consider a power delivery architecture model similar to the one shown in Fig. 2.3. As mentioned in [57], this scheme forms a realistic approach to be used for per-core or per-VI delivery scheme. Initial experimental results reported that the overhead compared with the ideal case where every voltage is precisely delivered would be around 25% on average. This is because the power supply rail, depending on the platform's variability would have to provide the worst case voltage required leading to a low LDO efficiency. This can be improved by providing extra rails or an extra layer of switching regulators that will downgrade the voltage to an intermediate level. In this case, the experiments show that the overhead will drop to 15%, which is still quite big but it is a good starting point for improvement and optimization.

2.4 Methodology & Framework

Voltage island formation when combined with voltage and frequency tuning can provide four different power management schemes that mitigate variability and deliver different power/complexity trade-offs:

1. Single-Voltage/Single-Frequency (SVSF): all the cores have the same voltage and frequency, leading to low complexity implementation but overdesigned power management decisions.

2. Single-Voltage/Multiple-Frequencies (SVMF): the frequency can be tuned individually for each core, enabling in that way the boosting or downgrading of the desired cores' performance, but the flexibility of this approach is constrained by the shared voltage.

3. Multiple-Voltages/Single-Frequency (MVSF): voltage scaling can be performed per core or per cluster of cores while the frequency is the same for the whole chip. The benefit of this approach is that the voltage can either be increased so that a higher frequency is achieved or decreased in order to consume less power.

4. Multiple-Voltages/Multiple-Frequencies (MVMF): the two knobs (voltage and frequency) provided in this scheme deliver the benefits of both SVMF and MVSF, leading to big power savings and fine-grained variability reduction, on the expense of high complexity both in implementation and management.

As mentioned before, the effects of process variation are exacerbated in NTC, but except for that, in order to exploit its energy efficiency potential, we should be able to provide performance guarantees to the applications running in an NT manyocre platform, with the ideal case being sustaining their ST performance. This becomes more evident if we consider the emerging paradigm of data centers and cloud computing. To further motivate the aforementioned claim, Figure 2.4 shows the performance distribution for a 128-core NT manycore that implements the best-effort EnergySmart power management SVMF approach [34]. The results are obtained from the execution of the *BARNES* application over 100 different variation maps. The normalized performance value of 1 corresponds to the *nominal* performance of the application. As shown, the performance of NT manycore platforms are not controllable and spread out over a wide range of normalized values (from 1 to 3.7) due to the underlying process variability. Thus, the adoption of NTC for applications, exhibiting specific performance and/or throughput constraints, requires careful selection and tuning

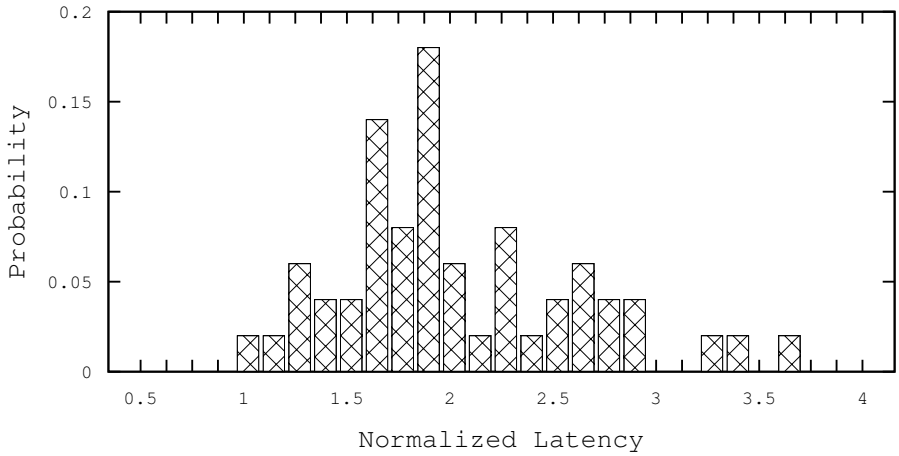


Figure 2.4: *Performance distribution on a 128-core NTC manycore implementing the EnergySmart [34] approach.*

of the power management scheme. In the following sections, we propose an exploration framework for variation-aware VI formation and we use it to evaluate several variation-aware power management tuning strategies that will enable performance sustainability in NTC.

The overall exploration framework for variation-aware VI formation at NTC is shown in Figure 2.5. It accepts as main inputs the performance, power and area characterization curves of the target application at the ST regime. The super threshold characterization is performed by adjusting the number of cores of the underlying manycore architecture template and then scaling accordingly the application's degree of parallelism. The Sniper multicore simulator [5] and the McPAT power modeling framework [40] have been used for the performance and power characterization, respectively. Designer/architect specific constraints are provided regarding the minimum allowed performance, L_{min} and the maximum core count constraint, C_{max} for the near-threshold manycore. Given the aforementioned inputs, the proposed exploration framework generates the VI configurations and the corresponding V_{dd} allocation decisions per VI, for a manycore architecture with C_{max} number of cores operating at the NT regime and satisfying the performance constraint L_{min} under parametric process variation. In the remainder of this section, we describe the basic components of the proposed methodology.

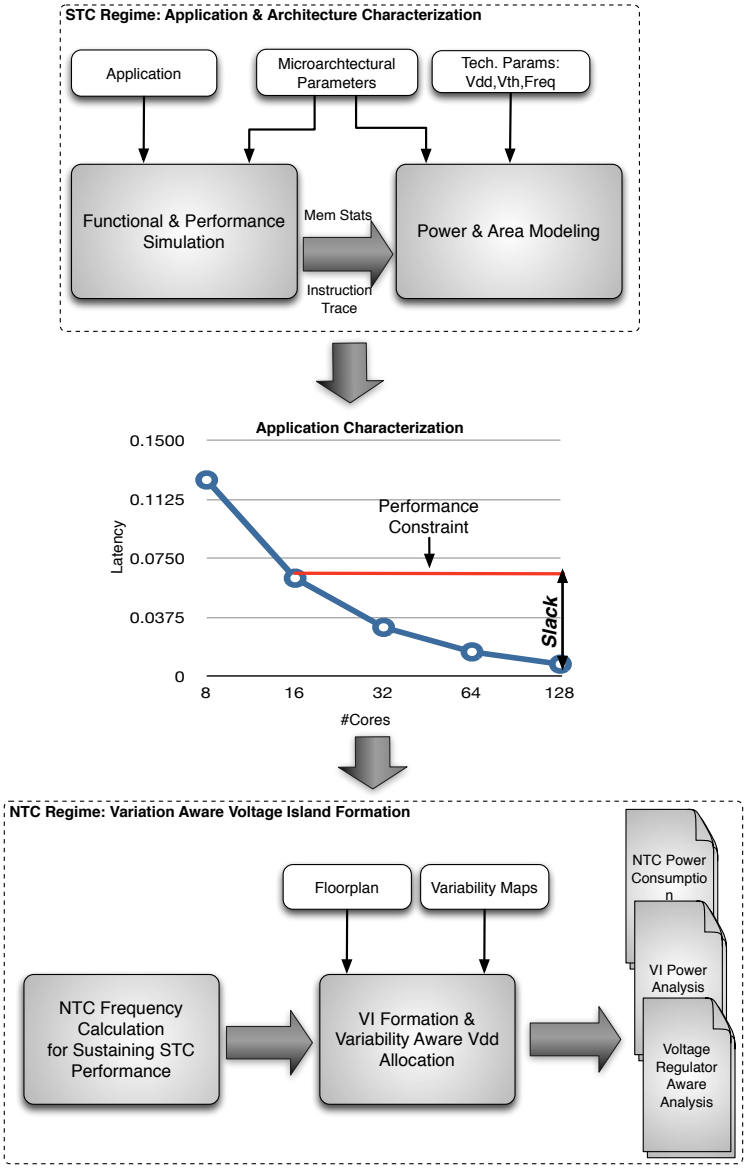


Figure 2.5: *Framework for variation-aware VI formation.*

2.4.1 Sustaining the ST Performance I: Workload Dependent NT Frequency Assignment

So far, application workloads have been originally developed and characterized for the ST regime. In order to sustain ST performance figures (i.e. latency or throughput) when moving to the NT regime, the inher-

ent parallelism of the applications should be exploited [50] to alleviate the impact of the reduced clock frequencies at NTC. Assuming a minimum allowed latency L_{min} and maximum core count constraint, C_{max} for the NT manycore, we first calculate the clock frequency of the platform at the NT regime, f_{NTC}, that satisfies the performance constraint. Let $L_{C_{max}}$ be the performance, in terms of latency, at the ST regime of a manycore architecture with C_{max} number of cores, running at f_{STC}. At ST region, $L_{min} - L_{C_{max}} > 0$ is the available latency slack due to the higher degree of parallelism of the architecture, that can be exploited to run the application at lower frequency. Utilizing this positive slack, the f_{NTC} is calculated as follows:

$$f_{NTC} = \frac{L_{C_{max}}}{L_{min}} \times f_{STC} \tag{2.2}$$

The calculated f_{NTC} refers to the target clock frequency of each core at NTC for sustaining ST performance, without considering the spatial effects of process variations. Assuming B as the set of component blocks in the floorplan and D the set of dies, we define $V_{th}^{(i,j)}$, $i \in B$, $j \in D$ that corresponds to the V_{th} of the architecture's component i in sample die j. Once extracted, $V_{th}^{(i,j)}$ is used for allocating to each component the lowest possible $V_{dd}^{(i,j)}$ for sustaining the f_{NTC} frequency constraint given that:

$$f_{NTC} \propto \frac{(V_{dd}^{(i,j)} - V_{th}^{(i,j)})^{\beta}}{V_{dd}^{(i,j)}} \tag{2.3}$$

where β is a technology-dependent constant (≈ 1.5). The extraction of the f_{NTC} and the per component $V_{dd}^{(i,j)}$, enables the adoption of different power management schemes for NTC operation with guaranteed performance sustainability.

2.4.2 Sustaining STC Performance II: VI Formation and Variability Aware V_{dd} Allocation at NTC

Given this NTC scenario, the f_{NTC} and the $V_{dd}^{(i,j)}$ values are used by a MVSF power management scheme to form the voltage island domains and allocate their NT voltages. The adoption of the MVSF scheme mitigates variability effects, while at the same time it derives an iso-frequency view of the manycore platform. The iso-frequency view of the platform facilitates the application development and porting, because it enables a symmetric platform from the performance point of view. Once the VIs have been defined, we compute the per island V_{dd} assignment that satisfies the f_{NTC} constraint.

Table 2.1: *Experimental Setup (Chapter 2): Platform Parameters*

Parameters	Value
Process Technology	22nm
STC Frequency	3.2GHz
STC Supply Voltage	1.05V
Nominal $V_{th}/\sigma_{V_{th}}$	0.23V/0.025
Number of Cores/Core Area	128/6mm^2
Private Cache Size/Area	320KB/4.14mm^2
Last Level Cache Size – Area	8 MB / 15.52mm^2

More specifically, for the jth die, $j \in D$, each VI, $k \in VI$, operates in its own $V_{dd}^{(k,j)}$, tuned for the VI$_{k,j}$ group of processors and memories. In VI$_{k,j}$, the core with the highest $V_{th}^{(i,j)}$, $i \in B$, $j \in D$ determines the V_{dd} for the specific voltage island, to satisfy the VI$_k$'s critical path timing. Analyzing the trade-off by moving towards coarse grained VI granularities, we reduce area cost since less voltage regulation logic is allocated at the expenses of degrading the power efficiency of the manycore with respect to the finest possible granularity. For $B_k, k \in VI$, the set of resources found in VI$_k$ and from Eq. 5.3, we calculate $V_{dd}^{(k,j)}$ according to the following relation:

$$V_{dd}^{(k,j)} = \max_{i \in B_k, j \in D} \left[V_{dd}^{(i,j)} \right] \tag{2.4}$$

2.5 Experimental Results

In this section, we experimentally evaluate the efficiency of the proposed framework. Without loss of generality, we consider that the performance L_{min} corresponds to a 16 core multicore in the STC regime, while the constraint C_{max} targets a 128 core many-core chip at NTC, at 22nm technology node. Maximum V_{dd} was set to 1.05V and the frequency to 3.2 GHz for the STC regime, according to parameter values derived from [4] for conservative technology scaling. From Varius-NTV, we extract 100 different variation maps by using a 24x16 grid based on the core/cache granularity. The most significant parameters and their values are summarized in Table 5.1.

For our experiments we used five applications from the SPLASH-2 benchmark suite [62] by using the "large dataset" provided within Sniper [5]. The applications considered exhibit different behaviors by scaling from 16 to 128 cores: close to ideal (*RADIOSITY*), medium (*BARNES, WATER-NSQ*) and limited scaling (*RAYTRACE, WATER-SP*). Additionally, we examined

an *average* case workload, that aggregates in the execution sequence the five aforementioned applications. Specifically, this *average* case workload is like executing all the aforementioned applications, one after the other and then treating it as a single benchmark. In that way, we manage to see what happens in an average case since it includes benchmarks that scale well and others that don't scale well. We present results regarding the power efficiency delivered by adopting the proposed approach and we provide a sensitivity oriented analysis regarding parameters of the voltage regulation structure.

2.5.1 Power estimation for the NTV regime

Given the V_{dd} allocation per VI from Eq. 2.4, $V_{dd}^{(k,j)}, k \in VI, j \in D$, and the power characterization for the manycore with C_{max} number of cores at STC, we can calculate the power of each component in NTC. For $i \in B_k, j \in D, k \in VI$, the dynamic, DP and leakage, LP, power scaling factors. Since McPat is not validated against near-threshold voltages, we use it for obtaining the ST values and then we scale accordingly in order to calculate the near threshold ones, using the models for NTC provided in [43]

$$SF_{DP}^{(i,j,k)} = \left(\frac{V_{dd}^{(k,j)}}{V_{dd_{STC}}}\right)^2 \times \left(\frac{f_{NTC}}{f_{STC}}\right) \tag{2.5}$$

$$SF_{LP}^{(i,j,k)} = \left(\frac{V_{dd}^{(k,j)}}{V_{dd_{STC}}}\right) \times \exp\left(\frac{V_{th_{STC}} - V_{th}^{(i,j)} + DIBL}{n \times V_{thermal}}\right) \tag{2.6}$$

$$DIBL = \lambda(V_{dd}^{(k,j)} - V_{dd_{STC}}) \tag{2.7}$$

where $DIBL$ is the coefficient modeling the Drain-Induced Barrier Lowering effect, $V_{thermal}$ is the thermal voltage, and n is the sub-threshold slope coefficient. The DIBL effect is a deep-submicron effect and is related to the reduction of the threshold voltage as a function of the drain voltage. DIBL is enhanced at higher drain voltage and tends to become more severe with process scaling to shorter gate lengths. Lowering supply voltage provides an exponential reduction in sub-threshold current resulting from the DIBL effect. Figure 2.6 shows the impact of DIBL effect on the reduction of leakage power in manycore architectures at NTC regime. As shown, by moving from STC multicore (16 cores) to NTC manycore (128 cores) architecture configurations, the DIBL effect accounts for a significant portion of the total power of the system.

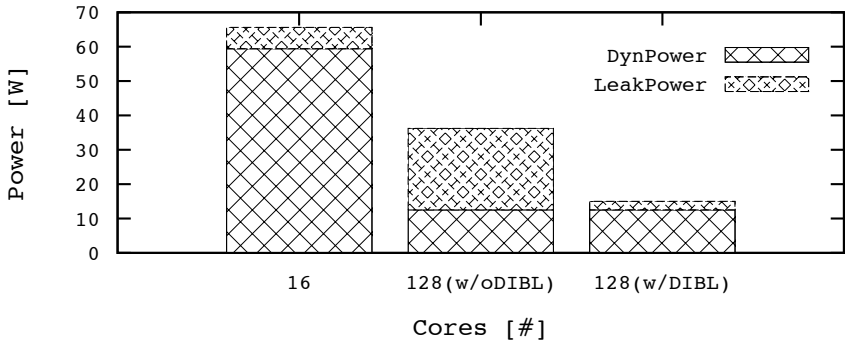


Figure 2.6: *Power breakdown for STC-16core and NTC-128core architectures with and without DIBL effect*

2.5.2 Analysis of Power Gains at NTC Regime

Moving from 16 STC Cores to 128 NTC Cores

Figure 2.7 shows the power consumption when moving from 16 cores at STC to 128 cores at NTC for each benchmark. Radiosity delivers the highest power gains since it scales almost ideally in terms of performance as the number of cores increases. For the radiosity benchmark we observed a 95% decrement in power while for the barnes and water_nsq that exhibit a medium scaling behavior, we observe a reduction of 77%. As shown, the scaling behavior of the applications with respect to increasing the number of cores, heavily affects the power efficiency at NTC, since the application takes advantage of the available performance slack when moving to a large number of cores. Thus, besides frequency we can aggressively scale down V_{dd} as well, and reduce power drastically, taking advantage of V_{dd}'s quadratic relation with dynamic power and its linear plus DIBL relation with leakage current. Raytrace and water_sp exhibit lower scaling degrees, limiting performance boost when their load is split and distributed to 128 cores. In this case, V_{dd} assignment acquires higher values restricting power gains. The last column of Figure 2.7 depicts the power gains delivered in the average workload. Although benchmarks that don't scale well are included, a near threshold V_{dd} ($\sim$0.4V) is acquired, delivering a 65% power reduction with respect to the 16 core STC manycore.

Variation Aware versus Overdesign NTC operation

We compared the power gains delivered by the proposed variation aware VI formation versus an overdesign approach to mitigate variation effects.

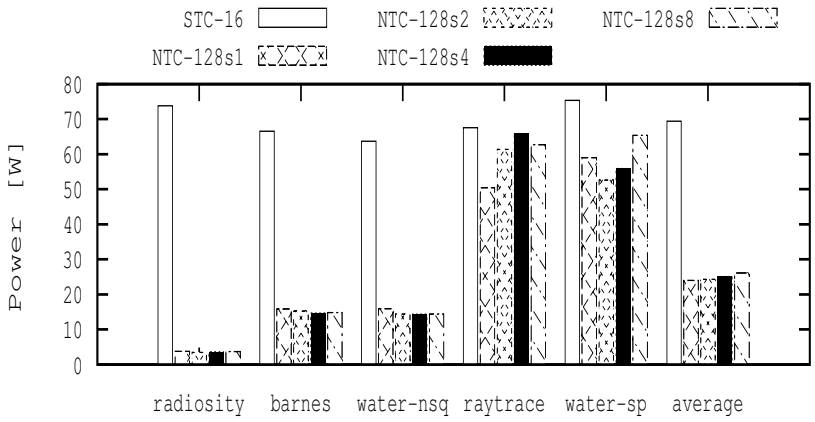


Figure 2.7: *Power Reduction: 16-core STC chip versus 128-core NTC*

From the V_{th} distribution, we calculate the V_{dd} of architectural components according to Eq. 5.3, with V_{th}'s overdesign value being equal to $\mu_{V_{th}} + 3\sigma_{V_{th}}$. Figure 2.8 reports the gains of the variability aware approach over the overdesign one. The histograms with the *singleVI* annotation represent power gains when having only one VI, and as a consequence one V_{dd} for the whole chip. Under a *singleVI* configuration, the variation aware approach achieves power gains around 5%, for all the available cluster architectures (S_i, $i \in \{1, 2, 4, 8\}$). On the contrary, the histograms with the *finestVI* annotation show the power gains achieved by considering the finest VI granularity possible for each architecture. Since S1 enables the finest 1×1 VI granularity to be exploited, it delivers the highest gains over the overdesign approach, that range between 34-42%. In the rest of the architectures, namely (S2, S4, S8), the gains vary between 29-34%, 25-28% and 18-23%, respectively.

Analysis of VI Granularity on NTC Power Efficiency

Figure 2.9 shows the impact of the different voltage island configurations in terms of power consumption at the NTC regime. The voltage island formation that has been analyzed includes all the possible combination in terms of power-of-two of the cores. We restrict the voltage island granularities to an aspect ratio between 1 and 1/4 (considering a voltage island configuration $c \times r$ the aspect ratio is c/r). In each case, we considered the tile size as the smallest possible voltage island. The constant trend over all the workloads and architectures, is that finer the granularity of the voltage island higher

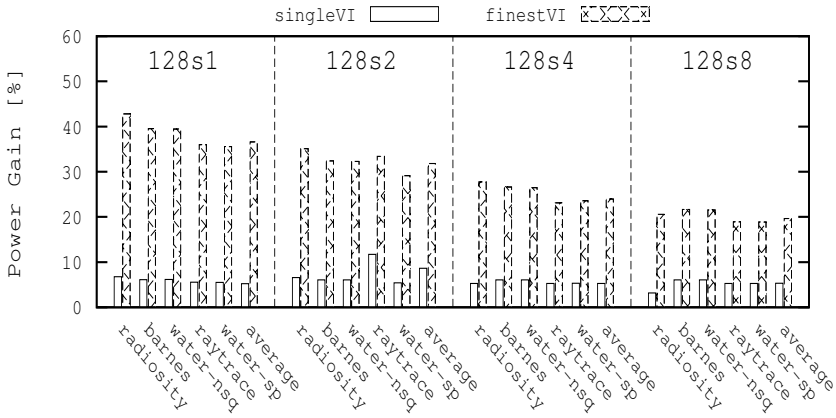


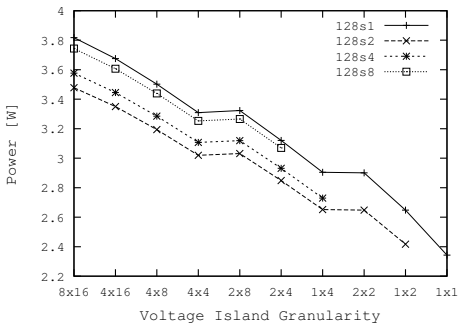
Figure 2.8: *Power gains of variability-aware NTC technique w.r.t. overdesign*

the power savings. In fact, selecting smaller voltage islands, we can cope with the variability in a more aggressive way by using a fine-grained tuning of the V_{dd} over the entire chip. The advantage passing from the single voltage island to the finest voltage island depends on the different architectural configuration: 30-35% for 128S1, 24-30% for 128S2, 19-24% for 128S4 and 14-18% for 128S8. In addition, the impact of the different voltage island configurations (when composed of the same number of cores), such as passing from 4×4 to 2×8 or passing from 1×4 to 2×2, is very limited. Despite the global trend, analyzing in more detail the power behavior over all the four clustered architectures, we noticed that the 128S1 architecture is not constantly the best configuration across the various VI granularities. In some cases the best configuration shifts over the 128S2 architecture (see water-sp in Figure 2.9). This phenomenon depends on the application scalability over the different clustered architectures.

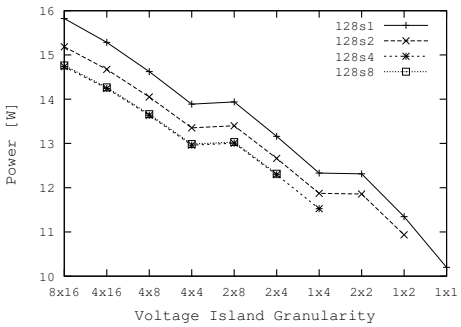
2.5.3 Voltage Regulation Oriented Analysis

The analysis conducted so far considers the ability to ideally deliver all the requested voltage levels. Since this is not a realistic scenario according to current state-of-art power supply architectures, hereafter we analyze the impact of the on-chip voltage regulator resolution on power efficiency.

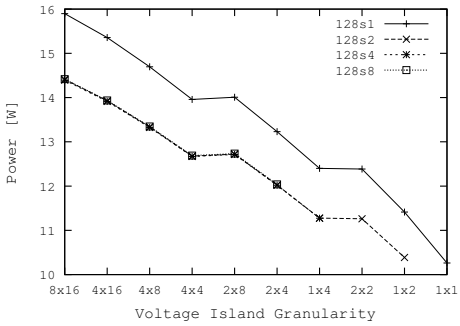
We analyzed three different voltage regulator resolutions, delivering voltage with a precision of (i) 12.5mV, (ii) 25mV and (iii) 50mV. Adopting the aforementioned schemes, we demonstrate the effect of allocating integrated regulators in the NTC region (from $[V_{th}] \longleftrightarrow [V_{th} + 200mV]$) that includes



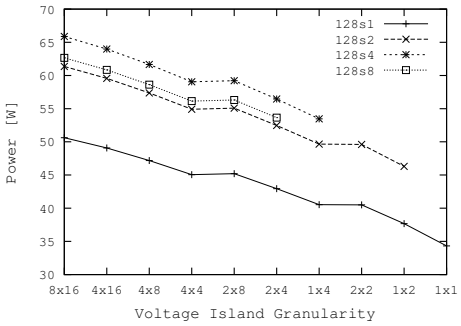
(a) *radiocity*



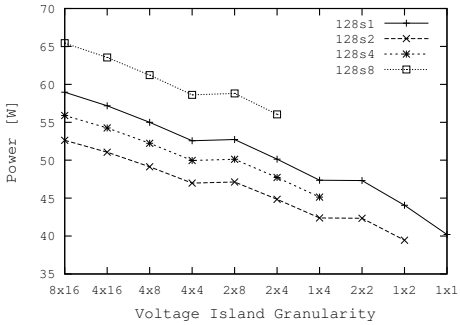
(b) *barnes*



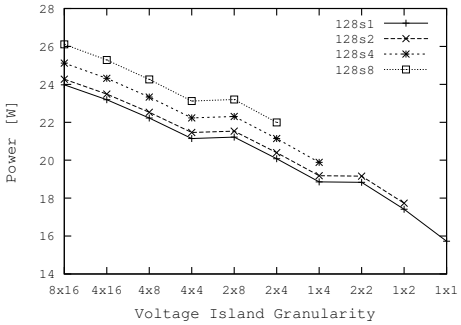
(c) *water-nsq*



(d) *raytrace*



(e) *water-sp*



(f) *average*

Figure 2.9: *Impact of voltage island granularity on power consumption*

respectively 16, 8, and 4 voltage quantization levels. Figure 2.10 presents the average power overhead for each one of the voltage regulator precisions. Power overhead refers to the normalized difference between the power consumed in the ideal case (voltage regulator delivering arbitrary V_{dd} values) and the power with the specific value of voltage precision. The results are

the average values for all the benchmarks and all the four architectures that we investigated. As expected, the higher is the resolution the smaller is the overhead since we are closer to the ideal case, passing from a 12% at 50mV to less than 3% at 12.5mV. For the applications that exhibit ideal or medium scaling with respect to increasing the number of cores, such as radiosity, barnes and water-nsq, the overhead of 12% can be efficiently compensated by the low-power consumption at NTC regime. On the opposite, for the case of applications with limited scaling, such as raytrace and water-sp, an integrated voltage regulation scheme that provides high resolution of the delivered V_{dd} is preferable.

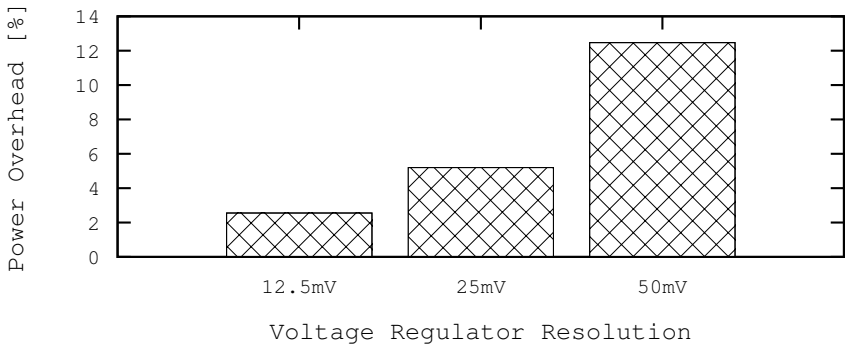


Figure 2.10: *Impact of voltage regulator resolution on power efficiency at NTC.*

Finally, Figure 2.11 shows the V_{dd} probability distribution considering the 12.5mV as the regulator's granularity for the *barnes* application running on the 128core architecture and operating at NTC, across all the examined *S1-S8* tile types. We can observe that the V_{dd} distribution is very concentrated across the mean value $\mu = 0.388V$ with $\sigma = 0.071V$. The V_{dd} distributions of the rest of the applications resemble the depicted one on Figure 2.11 with similar σ and slightly shifted μ values, according to the scaling behavior of the application. Narrow V_{dd} distributions together with low power consumption at NTC, instruct that the integrated voltage regulation circuitry can efficiently supply the requested power without the need of allocating multiple levels of voltage supplies, showing the efficiency of moving towards multiple VIs for supporting NTC operation on manycore chips.

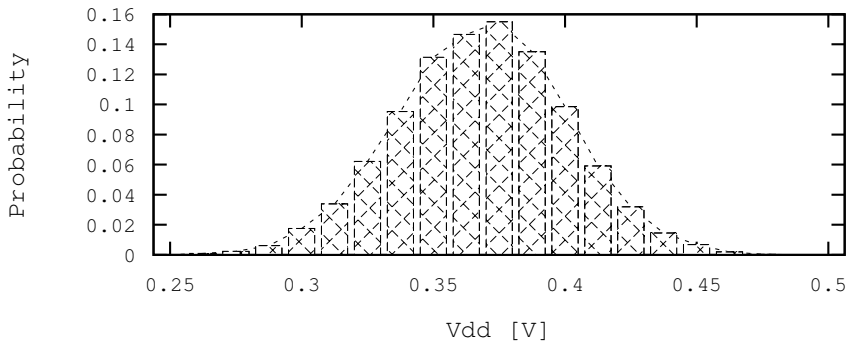


Figure 2.11: *Distribution of Vdd voltage at NT region.*

2.6 Conclusion

In this chapter, we presented a variability-aware framework for exploring the power-efficiency of Near-Threshold Computing. Motivated by recent advancement on power delivery systems, we proposed the utilization of voltage island formation combined with the operation at the near-threshold region as an effective technique for building power efficient manycore architectures that sustain performance values delivered by conventional super-threshold computing. Through extensive experimentation, we showed the optimization potentials of moving towards near-threshold voltage computation, exposing its high dependency on both workload characteristics and underlying architectural organization.

CHAPTER 3

Voltage Island Management in Near Threshold Manycore Architectures to Mitigate Dark Silicon

3.1 Introduction

In the previous chapter, we introduced a variability-aware framework for exploring the power-efficiency of Near-Threshold Computing. We proposed the utilization of voltage island formation combined with the operation at the near-threshold region as an effective technique for building power efficient manycore architectures that sustain performance values delivered by conventional super-threshold computing (STC). Through extensive experimentation, we showed the optimization potentials of moving towards near-threshold voltage computation, exposing its high dependency on both workload characteristics and underlying architectural organization. In this chapter, we extend our techniques and our experiments even further:

- instead of having an iso-frequency approach, we relax the performance constraint and try to go even faster than STC.
- motivated by the emerging cloud/server applications, we evaluate pro-

cess parallel workload in order to see if and how they are benefited by NTC.

- we experiment with a core-cache decoupling scheme, in order to experiment with the limits of fine grained voltage-island granularity.

Evaluation results, on both thread-parallel (parallel-application view - high synchronization) and process-parallel (cloud-based application view - low synchronization) workloads, show the high dependence of NTC efficiency to the workload's characteristics. Moving to NT regime for a 128-core architecture, while sustaining performance values obtained by a 16-core architecture at STC, average power gains $>90\%$ are delivered for process-parallel workloads, while 65% power gains for the thread-parallel workload set. We also show that given a best-effort V_{dd} tuning scenario (i.e. let NTC manycore to run faster than the requested STC constraint), a performance improvement of 27% can be achieved at the expense of 45% NTC power overhead. However, even with 45% power overhead, the maximum power dissipated by the NTC manycore is around 10W. Finally, analyzing the V_{dd} distributions at NTC, we demonstrate that the utilization of multiple VIs together with efficient integrated regulators can be considered a feasible option at NTC to efficiently deal with the process variability.

3.2 Methodology

As mentioned in *Section 2.4*, voltage island formation combined with V_{dd} and frequency tuning have been proved very efficient for mitigating core-to-core frequency and leakage variations [41]. There are four power management schemes supporting voltage/frequency islands: Single-Voltage/Single-Frequency (SVSF) for all cores, Single-Voltage/Multiple-Frequencies (SVMF), Multiple-Voltages/Single-Frequency (MVSF) and Multiple-Voltages/Multiple-Frequencies (MVMF). While the SVSF scheme usually leads to overdesigned power management decisions, the SVMF, MVSF and MVMF schemes provide a larger set of tuning knobs for mitigating process variations. The tuning of these knobs considering only variability mitigation scenarios [34] provides no guarantees regarding the performance of the NTC manycore. In order to exploit the energy efficiency potential of NTC for realistic workloads, applications running at NTC mode should ideally sustain their ST performance figures. Moving to NTC considering only the case of targeting a best-effort application domain, will limit NTC's applicability since the notion of service level agreements (SLAs), used in current data-center infrastructures and emerging cloud-based workloads, would not be efficiently

supported. In the current chapter, except for the MVSF scheme, we employ the MVMF one as well. The methodology and framework are adopted and extended from *Section 2.4*: for sustaining performance we use the same procedure and equations described in *Subsections 2.4.1* and *2.4.2*. Additionally, we also use the tile-based manycore architecture as well as the intra-tile organization depicted in Figures 2.1a and 2.2, but in this we limit the analysis to a 4 core per tile. The discussion is general and can be extended to other cluster organizations such as those proposed in the previous chapter, exploring in that way more coarse/fine-grained clusters. The intra-tile architecture is composed of 4 cores per tile and a last level cache (LL$) shared among all the cores in the tile. Each core owns a private instruction and data cache (P$). The Intel Nehalem processor [31] configuration for the core and the P$ has been adopted as reference.

3.2.1 Exceeding STC Performance: Combining V_{dd} Allocation with Best-effort f_{NTC} Assignment under Performance Guarantees

The MVSF approach presented in the previous section guarantees the performance at NTC by allocating in a variability-aware manner the V_{dd} to each VI, in order to enable each VI to run at f_{NTC} (i.e. the minimum clock frequency requested to sustain STC performance without timing violations). However, as shown in Figure 2.4, the effects of process variability are not monolithic: process variation might generate on-chip regions with higher V_{th} values that reduce the achievable clock frequency as well as regions with lower V_{th} values that enable clock frequencies higher than the f_{NTC} to be allocated. The existence of positive frequency slacks at specific regions of the manycore platform can be exploited by moving from the previous MVSF approach to a MVMF power management scheme to further push system performance. The adoption of a MVMF scheme enables multiple frequencies to be allocated within a single VI tailored to the performance capabilities of the VI's components, i.e. the underlying tile architecture. However, it is worth noting that MVMF will not impact the V_{dd} allocation of the VIs, which depends on the maximum V_{th} found within each VI, thus performance guarantees continue to be valid. Thus, under the MVMF scenario, the NTC manycore is becoming heterogeneous, by including tiles of processing cores that run at least as fast as f_{NTC} or even faster, implying that the performance is not only sustained, but even optimized with respect to the STC reference configuration.

The frequency allocation within each VI is performed by applying locally the EnergySmart approach [34], since each VI can be considered as

an SVMF configuration. Since the $V_{dd}^{(k,j)}$, $k \in VI$, $j \in D$, is allocated according to Eq. 2.4, it implies that the maximum achievable frequency, $f_{tile}^{(k,j)}$, of each tile within VI_k is bounded as follows:

$$f_{NTC} \leq f_{tile}^{(k,j)} \leq f_{MAX}^{k,j} \tag{3.1}$$

where $f_{MAX}^{k,j}$ corresponds to the maximum frequency supported by $V_{dd}^{(k,j)}$ and f_{NTC} is the minimum frequency to sustain the performance. Given the NTC voltage allocation, the power overheads of allowing higher clock frequencies than f_{NTC} to be assigned, is expected to be limited due to the linear but upper bounded frequency increment. We foresee the proposed MVMF scheme to be proved very advantageous for multi-process workloads exhibiting efficient scalability due to limited synchronization, where performance boost of a single core leads to direct throughput improvements.

3.2.2 Fine-grained VI Formation by Decoupling Cores from Cache Hierarchies

The two aforementioned VI formation strategies consider the tile as the finest granularity. However, the coarser the granularity, the smaller the optimization impact of the tuning procedure, because the average or worst case effects are becoming the dominant coefficients. Providing voltage and frequency knobs at the finest granularity, the tuning procedure is becoming more complex, but also more aggressive, thus offering further optimization potentials. Given the tile-based NTC manycore architectural template considered so far, we identify the finest possible granularity by decoupling within each tile the V_{dd} of the cores from the V_{dd} allocated to the cache memory hierarchy. Recent advances in memory design have shown that extreme voltage and frequency scaling of SRAM modules close to NTC regime with sufficient resilience regarding memory content flipping hazards is now available [20]. The core-cache decoupling will enable each tile component to be tailored according to its own process variability features. Performance guarantees could be satisfied with less emphasis on the platform's components, thus leading to extra power efficiency. The basic core-cache decoupling presents a power reduction due to the reduced granularity of the VI that we measured around 3%. However this decoupling can open a research path towards the exploitation of more specific cache optimization approaches (such as [56]) to get further power savings.

So far, a major barrier to such fine-grained tuning is the low efficiency of on-chip voltage regulators, showing 10%-15% efficiency loss. How-

Table 3.1: *Experimental Setup (Chapter 3): Platform Parameters*

Parameters	Value
Process Technology	22nm
STC Frequency	3.2GHz
STC Supply Voltage	1.05V
Nominal $V_{th}/\sigma_{V_{th}}$	0.23V/0.025
Number of Cores/Core Area	128/$6mm^2$
Tile/VI Size	4cores/4tiles
Private Cache Size/Area	320KB/$4.14mm^2$
Last Level Cache Size – Area	8 MB / $15.52mm^2$

ever, recent advancements in fully-integrated voltage regulators like Intel's FIVR technology [25], or the low-drop out (LDO) voltage regulator scheme proposed in [21], show that cost- and power-effective on-chip voltage regulation at fine-grained does not represent anymore a visionary scenario.

3.3 Experimental Results

3.3.1 Experimental Setup

For our experiments, we replicate the setup of Section 2.5: the Sniper multicore simulator [5] and the McPAT power modeling framework [40] have been used for the performance and power characterization respectively, while the Various-NTV microarchitectural model [32] has been employed to capture the process variation at the NT regime. A summary of the experimental setup used to evaluate the methodology is presented in Table 5.1. Core and caches types, sizes and area are taken from the Intel Nehalem architecture. The target platform is a 128 core many-core chip at NTC (at 22nm technology node) composed of 32 tiles, each one including 4 cores and a shared last level cache (LL$) of 8MB and 8 voltage islands (4 tiles each). Although in this chapter we are going to present the results obtained by considering single values for the tile size and VI granularity, the approach can be easily generalized to other architectural topologies.

Maximum V_{dd} has been set to 1.05V and the frequency to 3.2 GHz for the STC regime, according to parameter values derived from [4] for conservative technology scaling. By assuming a maximum power budget of 80W at STC, the performance to be sustained at NTC (L_{min}) corresponds to a 16 core architecture in the STC regime. From Various-NTV, we extracted 100 different variation maps by using a 24x16 grid based on the core/cache granularity.

Finally, in order to be consistent with our previous results, we use the same applications from the SPLASH-2 benchmark suite [62]. This time, though,the target applications have been used for the validation in two different scenarios. The first scenario consists of the single application multiple threads (SAMT) approach, where we supposed to run a single application on the entire platform by using its inherent parallelism at thread level (128 threads). The second scenario consists of multiple applications multiple threads (MAMT), where multiple instances of the same application are running (one per tile) and the internal parallelism at the thread-level is used within each tile (4 threads). This second version gives a sort of "cloud-oriented" view of the platform. The applications considered in the SAMT version exhibit different behaviors by scaling from 16 to 128 cores: close to ideal (*RADIOSITY*), medium (*BARNES, WATER-NSQ*) and limited scaling (*RAYTRACE, WATER-SP*). Additionally, we examined an *AVERAGE* case workload, that aggregates in a single execution sequence the five applications, treating them as a single benchmark. In that way, we manage to see what happens in an *average* case, where there is a combination of benchmarks that scale well and others that don't scale well. On the opposite, all the applications in the MAMT version present an almost ideal scaling passing from 16 cores (2 application instances over 2 tiles) to 128 cores (32 application instances).

3.3.2 Power Gains: NTC vs STC

Figure 3.1 shows the power consumption comparison when passing from 16 cores at STC to 128 cores at NTC for each benchmark in both SAMT and MAMT versions. The power values for the same benchmark on SAMT and MAMT versions are not comparable because the application performance are different in the two cases. All the MAMT versions of the applications and the RADIOSITY-SAMT deliver large power gains ($> 90\%$) due to the almost ideal performance scaling as the number of cores increases. The rest of the applications in SAMT version present a power gain that depends on the scaling capability, since it impacts the minimum frequency to be sustained and thus the minimum V_{dd} to be deployed to the voltage islands. For the remaining applications, Figure 3.1 shows a 75% decrement in power for BARNES and WATER-NSQ, around 25% for WATER-SP and an almost identical power for RAYTRACE. The AVERAGE-SAMT workload (composed of a sequential mix of all applications) delivers a power gain of 65%.

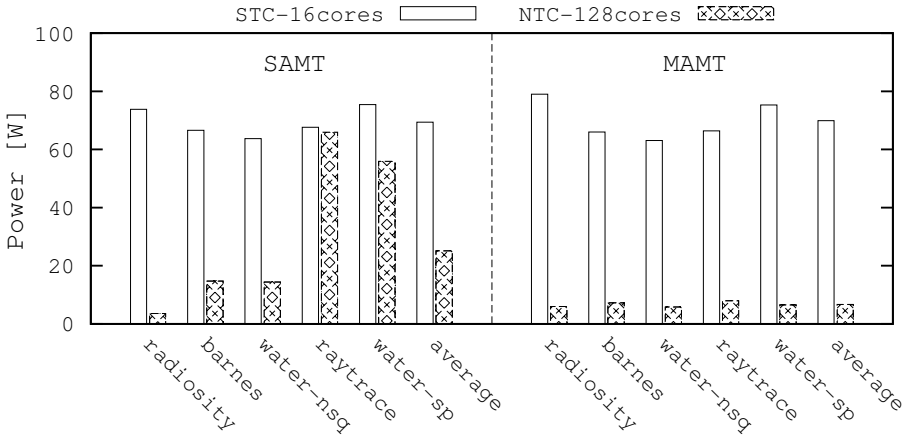


Figure 3.1: *Power reduction: 16-core STC chip versus 128-core NTC for both SAMT and MAMT versions of the target applications*

3.3.3 Relaxing the Isofrequency Constraint

Figure 3.2 shows the power/performance impact of the relaxation on the isofrequency constraint. To better evaluate this scenario, we present the experimental data considering only the MAMT version of the AVERAGE case. As stated in the previous section, while the MVMF has ideally an advantage due to the increment of the tile frequency, this can be really exploited only when the application is aware of this performance asymmetry. This is not the case of the SAMT version of our target applications. To have a clear view of the performance improvement we adopted the application throughput concept as the rate of jobs (application instances) completed within a time interval. As expected, the MVMF approach offers a performance speedup due to the frequency increment in the tiles not affected by the critical V_{th}. However, the performance improvement ($\approx$ 27%) is balanced by an increased power overhead ($\approx$ 45%).

Additionally, Figure 3.3 shows the tile frequency distribution across the 100 variation maps by using the MVMF mode. The minimum frequency is $400MHz$ to guarantee the application performance in terms of throughput. As expected, the minimum value is the most probable since there is at least 1 tile per VI (the one that limits the V_{dd} scaling) running at that frequency. Regarding the other values, we can notice that the distribution shows a long tail meaning that there is a large margin that can be used for further speedups.

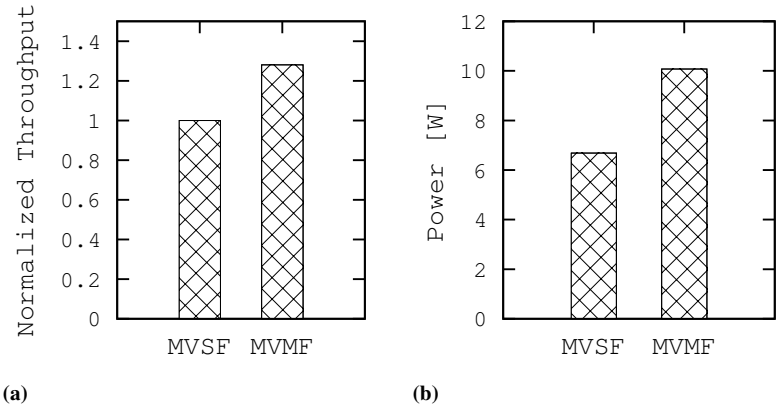


Figure 3.2: *Impact of MVMF vs MVSF in terms of (a) Throughput and (b) Power*

3.3.4 Voltage Regulators Analysis

The analysis conducted so far considers an ideal scenario where we can deliver all the requested on-chip voltage levels. According to state-of-the-art power supply architectures, we want to start including realistic constraints to the results, so in this section we analyze the impact of the on-chip voltage regulator resolution on power efficiency. We analyzed three different voltage regulator resolutions, delivering voltage with a precision of *(i)* 12.5mV, *(ii)* 25mV and *(iii)* 50mV. Figure 3.4 presents: the average power overhead for each voltage regulator precision in Figure 3.4a and the V_{dd} distribution according to each regulator resolution in Figures 3.4b - 3.4d. The power overhead and the V_{dd} distributions have been calculated across the 100 variation maps considering a target frequency of $400MHz$ to be sustained.

In Figure 3.4a we refer to power overhead as the normalized average difference between the power consumed in the ideal case (voltage regulator delivering arbitrary V_{dd} values) and the power corresponding to specific values of voltage precision. As expected, the higher is the resolution the smaller is the overhead since we are closer to the ideal case, passing from a 12% at 50mV to less than 3% at 12.5mV. This limited overhead value is interesting also considering the results shown in Figures 3.4.b-d, where it can be noticed that the V_{dd} distribution is very concentrated, which makes the use of the cost-efficient LDO on-chip regulation [21] schemes feasible to the NTC regime.

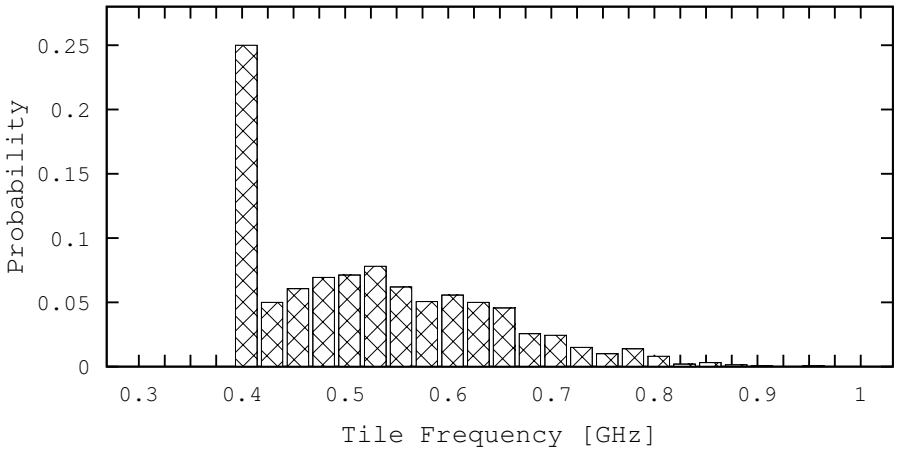


Figure 3.3: *Tile frequency distribution in MVMF mode*

3.4 Conclusion

This chapter focused on the emerging NTC paradigm as a key enabler for the power-efficient scaling of manycore architectures. While power efficiency is guaranteed by definition at the NTC regime, performance guarantee is still an open challenge. Sustaining STC performance figures during NTC operation is a critical issue for the wider adoption of the NTC paradigm. Towards this direction, we presented a set of techniques for variability-aware voltage island formation and voltage/frequency tuning that enable moving to NTC regime while sustaining STC performance guarantees. Extensive experimentation showed the optimization potentials of moving towards near-threshold voltage computing, outlining its high dependency on both workload characteristics and voltage tuning strategy.

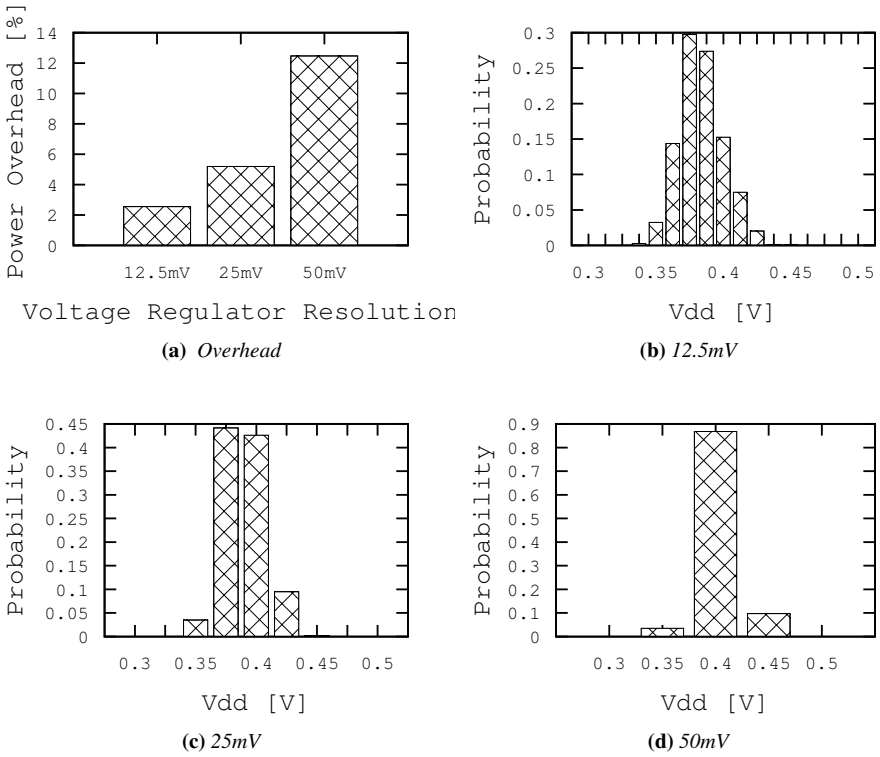


Figure 3.4: *Voltage regulator analysis: Power overhead (a) and V_{dd} probability distribution (b-d) for three voltage regulator resolutions*

Part II

Power Delivery Architecture Exploration and Runtime Optimization

Overview

In this second part, we extend the work and concepts presented in the first one in two directions: 1) power delivery architecture and 2) runtime optimization. Initially, motivated by the observation that there are not many solutions for near-threshold manycore architectures we evaluated the existing ones and showed what are the options and the potential benefits that we can obtain by optimizing the voltage allocation at NTC by considering the workload characteristics. Next, assuming a feasible, low overhead Power Delivery Network (PDN) for NTC, we propose a runtime management scheme for improved NTC manycore energy efficiency. Targeting highly parallel, multithreaded applications, which are ideal candidates for near-threshold computing, we propose an algorithm for balancing throughput under process (and workload) variability that sustains performance while reducing power.

CHAPTER 4

A System-Level Exploration of Power Delivery Architectures for Near-Threshold Manycores Considering Performance Constraints

4.1 Introduction

The power wall imposed by the breakdown of Dennard's scaling combined with the recent shift towards the manycore paradigm [10], [59] as a result of the continuous technology scaling, has brought into the picture the problem commonly referred to as *dark silicon*. The widening gap between the number of cores integrated on a chip and the number of cores that can be powered on simultaneously has been under active research by the community, leading to a number of proposed techniques ranging from micro-architectural level [23], [60] down to physical and device level [53], [49]. One of the promising techniques proposed [11] is Near-Threshold Computing (NTC), where aggressive voltage scaling is performed, enabling a large number of cores to operate simultaneously at the expense of performance degradation and high susceptibility to parametric process variations.

The power budget of a manycore platform is a major constraint, thus the design of an efficient power delivery system is crucial. With recent advancements in on-chip voltage regulator (VR) design and implementation [37], [38], [45] it is now possible to design complex delivery networks supporting fine grained voltage domains. Several attempts [26], [38] to support near-threshold voltages (NTV) as well as to design systems for fine grained Dynamic Voltage and Frequency Scaling (DVFS) [39], [57], [22] have been made, but most of them address small multicore platforms, consisting of 16 up to 32 cores, and none of them validates the scalability of those approaches to manycore systems consisting of hundreds of cores. Thus, issues such as: 1) the impact of the applications scalability regarding the increasing number of cores when running in NTC, 2) the increased current and 3) the many different voltage levels needed ideally by the cores because of the increased variability impact, must be considered. The main purpose of this chapter is to evaluate and assess existing power delivery architectures that can be applied to manycore platforms; more specifically we investigate the efficiency and the implications of different power delivery schemes for a near-threshold platform with a predefined voltage range. The main contributions of this chapter are:

- Evaluating the existing power delivery architectures for near-threshold manycores under process variation.
- Optimizing voltage allocation for each application considering its own characteristics.
- Considering the correlation between the PD system and workload characteristics to fully exploit the NTC benefits.

4.2 State of the Art

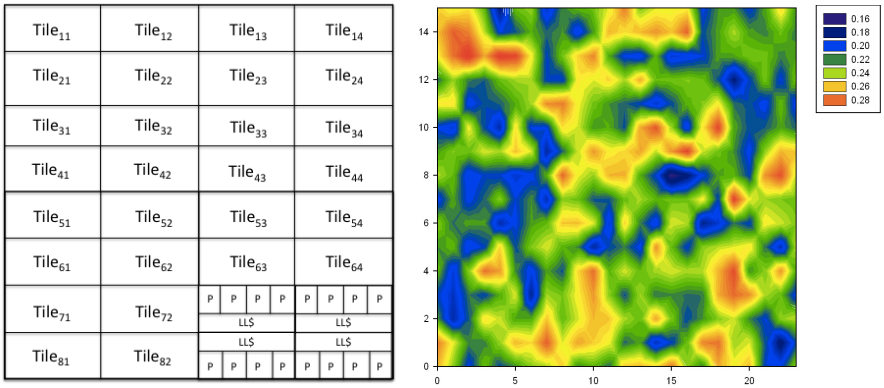
Near-threshold voltage operation relies on the aggressive tuning of the V_{dd} very close to the transistors' threshold voltage V_{th}. This reduction of the supply voltage increases the potential for energy efficient computation, for example by decreasing V_{dd} from the nominal 1.1 V to 500 mV, significant energy gains are reported in [11]. Many researchers have investigated the possibility of delivering near-threshold voltages on-chip. Hsieh et. al [28] designed a linear regulator with a variable output voltage that ranges from 0.5 to 1 V in steps of 0.1 V. Lee et al. [38] evaluated the technique known as "voltage stacking" for near-threshold voltages: multiple low-voltage blocks are powered from a single higher voltage by "stacking" the logic blocks and

recycling charge between the layers. SuperRange [26] is a wide operational range (0.4 - 1.2 V) power delivery scheme that uses an off-chip VR to deliver the Super Threshold (ST) voltages and an on-chip VR for the NT ones with an average of 70% power efficiency. Booster [44] includes two power supply rails at 400mV and 600mV respectively and uses hints provided by synchronization libraries to determine which cores should be "boosted" (run at higher frequency) maintaining an average per-core frequency and reducing the effect of process variation. Interesting power delivery (PD) architectures targeting multicore platforms with per-core DVFS support, that do not operate in the NT region, have been proposed as well. VRCon [39] uses a switching network that allows the cores to share on-chip voltage regulators according to the DVFS intervals and load conditions improving the overall energy efficiency, but it is not scalable since the integration of hundreds of switching (SW) regulators and switches on-chip would impose a significant area overhead. Sinkar et al. [57], based on the observation that a Low-Dropout Regulator (LDO) can share its largest component with the Per-Core Power-Gating (PCPG) device, and thus can be integrated with a minimum area overhead, demonstrated that a multicore platform with per-core voltage domains is implementable, and under certain assumptions, can be as effective as one having per core switching VRs (instead of LDOs). All of those techniques address part of the problem but they do not offer an insight on how near-threshold voltages can be delivered efficiently in a manycore architecture with hundreds of cores with many different voltage levels. In the first part, we investigated the performance sustainability in NTC manycore platforms through Voltage Island (VI) management and here we try to investigate more the problem of power delivery.

Our study mainly focuses on the NTC design space defined by [11] and [34]. Specifically, we target power efficient NTC manycore architectures that sustain ST performance levels by considering their increased sensitivity to process variation and investigate the trade-offs of different power delivery schemes.

4.3 Proposed Methodology

Different power management techniques, such as Single-Voltage/Multiple-Frequencies (SVMF), Multiple-Voltages/Single-Frequency (MVSF) and Multiple-Voltages/Multiple-Frequencies (MVMF), have been proposed and evaluated for manycore architectures in [34], [10]. Motivated by the existing programming models that assume a single platform frequency and aiming for a predictable performance and symmetric computational power we adopt



(a) *Manycore Architecture* **(b)** V_{th} *variation map*

Figure 4.1: *Tile-based architecture (a) and* V_{th} *variation map derived by [32] (b).*

an iso-frequency approach, meaning that all our cores have the same frequency. This also simplifies the design and eliminates any synchronization problem. However, given the underlying process variability (Figure 4.1b), the core voltage has to be tuned accordingly in order to achieve the target frequency leading to the MVSF approach. Figure 4.1a shows the floorplan of the target tile-based manycore architecture, as well as the intra-tile organization that has been used throughout this thesis. In this chapter, the intra-tile architecture is composed of 4 cores per tile and a last level cache (LL$) shared among all the cores in the tile. Each core owns a private instruction and data cache (P$). Intel's Nehalem processor configuration for the core and the P$ has been adopted as reference. In the next section we analyze the frequency and voltage allocation process, as well as the different power delivery schemes.

4.3.1 Workload-Dependent Frequency Calculation for Sustaining Performance and Variability Aware V_{dd} Allocation at NTC

Typically, application workloads are developed and characterized for the ST regime. In order to sustain ST performance figures (i.e. latency or throughput) when moving to the NT regime, the inherent parallelism of the applications should be exploited to alleviate the impact of the reduced clock frequencies at NTC. The procedure for calculating the the frequency for sustaining performance in NTC as well as for allocating the proper voltage is the same with the one that has been described in Part I and more specifically in *Section 2.4*. We describe it briefly below for reasons of consistency

and convenience. Assuming a minimum allowed latency L_{min} and a maximum core count constraint C_{max} for the NTC manycore, we first calculate the clock frequency of the platform at NT regime, f_{NTC}, that satisfies the performance constraint. Let $L_{C_{max}}$ be the performance, in terms of latency, at the ST regime of a manycore architecture with C_{max} number of cores, running at f_{STC}. At STC, $L_{min} - L_{C_{max}} > 0$ is the available latency slack due to the higher degree of parallelism of the architecture, that can be exploited to run the application at lower frequency. Utilizing this positive slack, the f_{NTC} is calculated as follows:

$$f_{NTC} = \frac{L_{C_{max}}}{L_{min}} \times f_{STC} \tag{4.1}$$

The calculated f_{NTC} refers to the target clock frequency of each core at NTC for sustaining ST performance, without considering the spatial effects of process variations. Assuming B as the set of component blocks in the floorplan and D the set of dies, we model variability by using different values for the voltage threshold parameter and we define $V_{th}^{(i,j)}$, $i \in B$, $j \in D$ that corresponds to the V_{th} of the architecture's component i in sample die j. Once extracted, $V_{th}^{(i,j)}$ is used for allocating to each component the lowest possible $V_{dd}^{(i,j)}$ for sustaining the f_{NTC} frequency constraint given that:

$$f_{NTC} \propto \frac{(V_{dd}^{(i,j)} - V_{th}^{(i,j)})^{\beta}}{V_{dd}^{(i,j)}} \tag{4.2}$$

where β is a technology-dependent constant (≈ 1.5). The extraction of the f_{NTC} and the per component $V_{dd}^{(i,j)}$ enables the adoption of different power management schemes for NTC operation with guaranteed performance sustainability. After obtaining the aforementioned parameters we proceed to the calculation of the NTC power by scaling the initial power, both dynamic and leakage.

4.3.2 Power Delivery Schemes

After determining the voltage range of the platform there are several ways of delivering the requested voltages to the cores, differing in both complexity and efficiency:

1. One power supply rail: One rail, powered by an off-chip voltage regulator, provides the appropriate voltage to the cores, meaning that the slowest core determines the voltage value. The power management scheme in this case coincides with the SVSF approach. It is a simple, low cost but very inefficient solution

2. Two power supply rails: In this case, there are 2 rails (2 off-chip VRs), set at two different voltages. Each voltage can be either fixed, or in a more complex scenario it can be adjusted at runtime by the power manager, based on the application demands. Depending on variability, the cores must be able to connect to the rail that satisfies their voltage demand using PMOS power gates as in [59].

3. Per core LDOs: Each core is equipped with a low dropout voltage regulator (LDO) [63]. An LDO is a type of linear regulator and its power efficiency can be calculated as follows:

$$\eta_{ldo} = \frac{P_{out}}{P_{in}} = \frac{I_{out}V_{out}}{I_{in}V_{in}} = \frac{I_{out}V_{out}}{(I_{out} + I_q)V_{in}} \tag{4.3}$$

   where I_q is the quiescent current flowing to the ground:

$$I_{in} = I_{out} + I_q \tag{4.4}$$

   The LDOs exhibit current efficiencies up to 99%, making the difference between the input and the output voltage the dominant factor of Equation 5.1. The lower the difference, the higher the efficiency, however the difference should not drop below a certain threshold because the circuit ceases to regulate its output voltage against any fluctuations in the input voltage. This threshold is defined as the *dropout voltage* and is defined at design time. The main advantages of an LDO are its fast transient response and its low cost, making it a potential candidate for on-chip VR. It can be implemented by adding some extra circuitry in the PCPG system with an area overhead of 2% [57], [21]. The regulators are used as a second level that steps down core voltage even further and are powered by one of the previous schemes (one or two rails).

All the previous schemes were chosen to be evaluated because they have the advantage of introducing a small but affordable (off-chip: an extra regulator, on-chip: LDOs which have less than a 2% overhead as described above) area and design overhead making them cost effective choices. In this chapter, the proposed methodology has been applied to a 128-core manycore architecture, but it can be easily applied to other manycore platforms in order to evaluate its scalability and efficiency.

Table 4.1: *Experimental Setup (Chapter 4): Platform Parameters*

Parameters	Value
Process Technology	22nm
STC Frequency	3.2GHz
STC Supply Voltage	1.05V
Nominal $V_{th}/\sigma_{V_{th}}$	0.23V/0.025
Number of Cores/Core Area	128/$6mm^2$
Tile/VI Size	4cores/4tiles
Private Cache Size/Area	320KB/$4.14mm^2$
Last Level Cache Size – Area	8 MB / $15.52mm^2$

4.4 Experimental Results

4.4.1 Experimental Setup

The Sniper multicore simulator [5] and the McPAT power modeling framework [40] have been used for the performance and power characterization respectively, while the Varius-NTV microarchitectural model [32] has been employed to capture the process variation at the NT regime. Since McPat is not validated against near-threshold voltages, we use it for obtaining the ST values and then we scale accordingly in order to calculate the near threshold ones, using the models for NTC provided in [43].

A summary of the experimental setup used to evaluate the methodology is presented in Table 5.1. Intel's Nehalem architecture has been adopted for determining the core and cache types and sizes. The target platform is a 128 core manycore chip at NTC (22nm technology node) composed of 32 tiles, each one including 4 cores and a shared last level cache (LL$) of 8MB. The maximum V_{dd} has been set to 1.05V and the frequency to 3.2 GHz for STC, according to parameter values derived from [4] for conservative technology scaling. The performance to be sustained at NTC (L_{min}) corresponds to a 16 core architecture in the ST regime. From Varius-NTV, we extracted 100 different variation maps by using a 24x16 grid based on the core/cache granularity.

Finally, the SPLASH-2 [2] and PARSEC [3] benchmark suites provided the target applications and the "large dataset" workload option in Sniper has been adopted. From the given applications, we experimented with the ones that exhibit adequate performance improvement when going from 16 to 128 cores and thus are good candidates for NTC. After running the simulations, we can categorize the applications in 4 different classes according to their performance scalability: ideal scaling (*BLACKSCHOLES*), good

scaling (*BARNES, WATER-NSQ*), medium scaling (*DEDUP*) and limited scaling (*RAYTRACE, WATER-SP*). We consider as *ideal scaling*, the case where an application's performance increases proportionally with the number of cores (for example, 8 times faster when going from 16 to 128 cores), as *limited scaling* when performance improvement is not proportional but just good enough to run in the near threshold region while the rest of the cases (*good and medium scaling*) exhibit a scaling between the two previous delimiters.

4.4.2 Determining the Voltage Range of the Platform

After running the benchmarks, we proceed to determining the requested voltages under process variation. The applications, based on their workload characteristics, exhibit different behaviors, thus they request different frequencies for sustaining the STC performance and as a consequence different minimum supply voltages are needed in order to achieve them. We calculate the per-core V_{dd} for all the different variation maps generated by Varius-NTV and for all the different applications. The SRAM voltage cannot be scaled down after a certain point because it leads to major defects and errors and we assume it has its separate constant supply : 500mV in standby mode and 700mV when accessing. The procedure is the following: we create 12.5 mV interval "voltage bins" and we assign each voltage generated to the corresponding one. For each application we generate a voltage "profile" and plot the probability density function (PDF) and the cumulative distribution function (CDF). The PDF and CDF of all the applications combined can be seen in Figure 4.2. By observing the figure one can extract valuable information: First, the minimum (0.25V) and the maximum (0.65 V) requested voltages (requested voltage range; in our case is a little bit more than 400mV). Notice that Near-Threshold operation does not refer to a specific voltage value or voltage range, it depends on the threshold voltage of the transistors. In our case the range is not very narrow and the maximum voltage may slightly diverge from the values referred in the bibliography but this is done because in that way it can be ensured that all the performance constraints imposed by the considered applications can be met. As mentioned at the beginning, it is a major challenge to distribute this variety of requested voltages, a problem which is also exacerbated by the increased effects of variability in the investigated voltage range. Additionally, defining the voltage range is a very important task that affects the decisions taken when designing the power delivery system as we are going to demonstrate below. Second, the median of the distribution, which

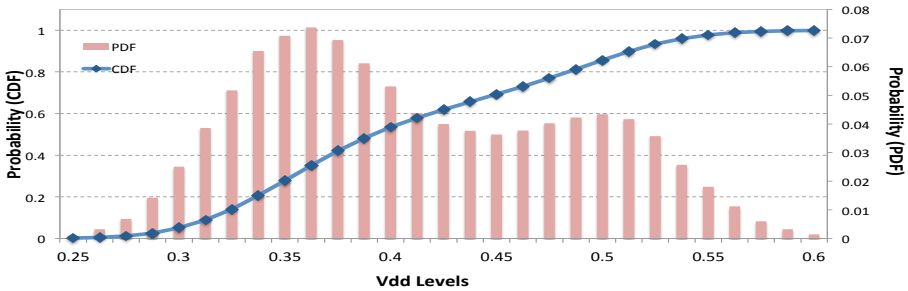


Figure 4.2: *Probability Density and Cumulative Distribution Functions.*

Table 4.2: *Application Frequency and Supply Voltages*

Application	Frequency (GHz)	Low V_{dd} (V)	High V_{dd} (V)
Blackscholes	0.4	0.325	0.45
Barnes	0.535	0.35	0.475
Water.nsq	0.615	0.3625	0.5
Dedup	0.84	0.4125	0.55
Raytrace	1.2	0.4875	0.6375
Water.sp	1.27	0.5	0.65

is a good candidate for the low V_{dd} rail value considering that half of the voltages lay below this values. From our analysis it is shown that the core power efficiency is maximized when a value between the 80th and the 90th percentile (application dependent) is chosen, contributing an extra 8%-12% power saving. This is because more cores are enabled to connect to the lower V_{dd} rail. However, this is true, only if we choose to disregard the PDN power losses. Our experiments show that, depending on the application, there can be a 16% up to 36% increase in the power delivery loss if the current is not balanced between the two rails. Thus, in each case, we choose the voltage that minimizes the current imbalance and not the one that ideally maximizes the power consumption of the cores. In this way, we minimize the IR drop as well as the power delivery loss and we reduce the effect of electromigration, improving reliability. Finally, the maximum voltage should be the value for the high V_{dd} rail. The same procedure has to be followed for each application separately. The results are presented in Table 4.2

4.4.3 Power Efficiency of Different Delivery Schemes

In this section, we evaluate the efficiency of different power delivery architectures. We include two different scenarios: in the first one the supply volt-

ages are fixed and defined at design time (*Fixed Voltages*) and in the second they are adjusted dynamically at runtime (*Custom Voltages*) depending on the current application running. In the *fixed voltages* scenario, every time a new application arrives, due to the different frequency demand, the power management unit (PMU) which is aware of both the chip-variability and the different workload characteristics, is configuring the switching network in order to connect each core to the appropriate supply rail [59], [1]. In some cases, when the requested voltage is at either extreme (low or high), an imbalance occurs where up to 85% of cores can be connected to only one rail. In this case, each rail should be designed to handle the worst case, leading to significant over-design. In the second approach, *fixed voltages*, since one rail is delivering the low voltage and the other one the higher one, as we can see in Table 4.2 by observing the low and high Vdd columns, the range for each rail is limited ($< 200mV$) and it can be delivered with high efficiency (90%) [21] by an off-chip regulator designed and optimized for delivering these values.

In Figure 3, we can see the power reduction for three different applications. All values are normalized to the one rail scheme. One rail means that there is only one voltage supply for the whole chip: In the *fixed voltages* scenario this value is the maximum voltage requested among all the applications, whereas in the *custom voltages* scenario it is "tailor-made", meaning that it is the maximum voltage requested by each specific application. The same holds for the 2 rails: in the first case the values derive from the CDF generated by the full set of applications (Figure 4.2), but in the second case each time the rail voltages change depending on the CDF of the application currently running (Table 4.2). The third approach is the per-core LDOs scheme, which could be applied to either 1-rail or 2-rail scheme. In this approach, each core has its own LDO block which downscales the rail voltage to the exact value needed by that core.

For Barnes (Figure 4.3a), in the *fixed voltages* scenario, there is a big power reduction even with the transition from a single rail to two. This is because most of the cores are "served" by the low V_{dd} rail, but before, not having an alternative option, they were connected to the high Vdd rail. When we have custom voltages we also have significant savings (compared to *one-rail (fixed))* and when employing two rails the reduction surpasses 60%. In the per-core LDO approach, the reduction becomes a little bit larger except for the case of one rail with fixed voltages. This is because, based on Equation 5.1, the smaller the difference between the input and the output voltage of the regulator is, the higher the efficiency. In our case, the LDOs exhibit a very good average efficiency ($> 85\%$) because the average

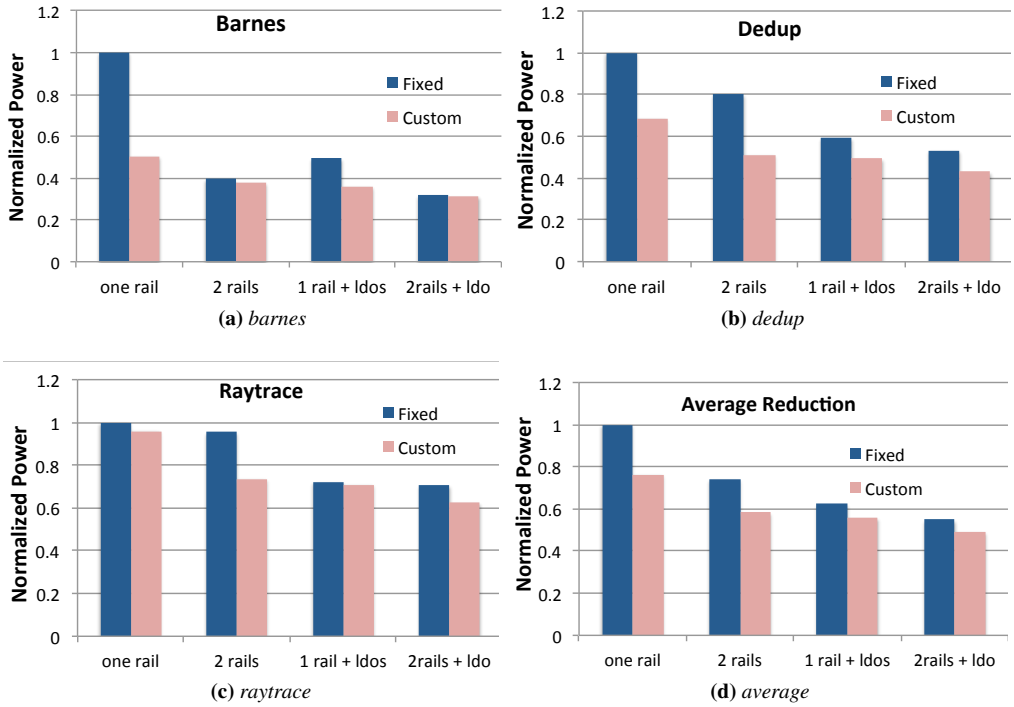


Figure 4.3: *Power Reduction: (a) Barnes (b) Dedup (c) Raytrace (d) Average.*

difference between input and output voltage is less than 70 mV. This also explains why in the one-rail plus LDOs (fixed voltages) scenario, there is a small increase compared to the 2-rail scheme: one single V_{dd} rail, with the maximum voltage requested (worst case) is powering all the regulators and thus their efficiency is decreased due to the large input/output voltage difference. The previous observations show that for applications that demand voltages near the lower end of the voltage range it is necessary to provide a second rail and/or LDOs because they are benefited confoundedly.

In the Dedup application (Figure 4.3b) we observe similar trends, but the reduction is not as significant. Dedup requests higher voltages than Barnes, provided in a more balanced way from the high and the low V_{dd} rail, and so the savings are not as large when compared to the trivial case of the one supply rail. In this case, there is a significant benefit of 20% when adopting the *one-rail + LDOs* scheme compared to the *two-rails (fixed)*. The main insight here is that for applications that request voltages in the middle range an extra fine tuning (such as using the LDOs) provide adequate results in all the cases.

Table 4.3: *Power Savings: Fixed Vs Custom Voltages*

Delivery Scheme	Power Savings
1 Rail	24%
2 Rails	21%
1 Rail+LDOs	11%
2 Rails+LDOs	10%

Finally, we can make very interesting observations concerning the Raytrace application, which does not scale as well as the previous ones. First, there is no significant difference between the fixed and the custom voltage approach when there is one supply voltage. This is because the requested voltages of the specific application are pretty close to the maximum platform voltage. In the 2-rail scheme with fixed voltages, the reduction is no more than 5% because only a few cores benefit from the low V_{dd} rail. We have finally to use to the custom V_{dd} rails in order to get the first notable results (27% reduction) and we can achieve similar results by just adding LDOs to the *one-rail* design. This observation, combined with the previous, is very important because it shows that when designing a platform for running applications with different workload characteristics, we have to take into consideration that the power delivery system has to be optimized for all different application categories.

Figure 4.3d depicts the aggregated power reduction for all applications normalized to the one-rail scheme. Depending on the approach, we can have an average reduction that ranges from 24% (*one rail - custom*) up to 50% (*2 rails + LDOs*). Figure 4.3d is also important because we can deduct the average power savings between the the fixed and custom approaches for each scheme. Table 4.3 shows the savings for each scheme when going from fixed to custom voltages. The one and two rail schemes are benefited the most (24% and 21% respectively) when using custom voltages, tuned at runtime for each application (instead of fixed ones), in contrast to the schemes supporting LDO regulators, which exhibit savings of 11% and 10% respectively.

This is because the approaches with the integrated LDO regulators can deliver a more customized per-core voltage by downscaling the supply provided by the rails even further, and they do not depend solely on the custom rail-voltage tuning for reaping the benefits of fine-grained V_{dd} allocation.

Finally, we also compared the performance of our proposed scheme with that presented in [39]. In [39] the authors propose a dynamically switching network connecting on-chip switching regulators with the cores and

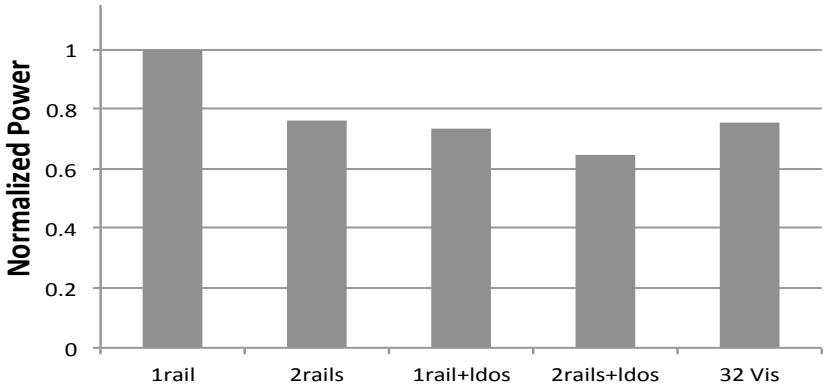


Figure 4.4: *Average power reduction of proposed schemes w.r.t. [39]*

making it possible for the cores to share the VRs (consolidation) using indications provided by the DVFS governor to improve overall efficiency. It is important to note that a switching VR requires at least 4 times more area that an LDO [19]. However at an area disadvantage to the LDO approach we assumed that 32 on-chip SW VRs are integrated, delivering power to 128 cores, meaning that each VR would be shared among 4 cores forming 32 clusters. We further favored this approach by assuming that there is an optimal network configuration leading to an ideal efficiency of 90% for the SW VRs, giving in that way the switching network scheme an extra advantage. In Figure 4.4, we can see that this approach delivers savings that are comparable to the case of having two supply rails but less than the per-core LDO scheme. This is because the slowest core in each VI determines the voltage for all 4 cores, leading in a higher (per-core) voltage for the remaining three cores and consequently to a greater power consumption.

4.4.4 Complexity and power overhead of the PDN

In the previous section, we compared the different power delivery schemes assuming that there is no additional cost when choosing to have two supply rails instead of one. However, there are overheads introduced when more than one voltage rail is considered for the design. For instance, metal utilization for delivering the V_{dd} and V_{ss} will change as more rails are used. This results in higher resistance for each voltage rail compared to that of a single one and therefore in a higher IR drop on each one of them. Higher IR drop demands higher pin voltage in order to maintain the same voltage at the transistors compared to the single rail design. For each design the overhead has to be carefully analyzed and calculated, based on the maxi-

mum current. Our initial calculations show that going from a single-rail to a two-rail design in a technology with 12 metal layers, results in a 10mV - 15mV worst-case IR drop increase that has to be accounted for, leading to a maximum 1.5 - 2.3% of power overhead for the delivery network. Additionally, the bump map and package planning will also get impacted and slightly more complicated. The number of Electro-Static Discharge (ESD) clamps for the design will increase and careful consideration for intentional de-cap is needed since the shared intrinsic de-cap will be smaller than that of the single rail.

4.5 Conclusions

In this chapter, we explored and evaluated different power delivery schemes for near-threshold manycore architectures under process variation. We showed that when the workload characteristics of the applications are analyzed and considered at runtime, significant power savings can be obtained even when using existing, cost-effective power delivery techniques, while meeting application performance constraints.

CHAPTER 5

Throughput Balancing for Energy Efficient Near-Threshold Manycores

5.1 Introduction

Technology scaling has enabled the chip industry to integrate more cores per processor leading to manycore designs. However, the stagnation of voltage scaling, known also as Dennard's law, and its thermal and power budget implications, do not let us exploit the full potential of those architectures. This problem, widely known as Dark Silicon, is expected to be exacerbated in the future.

At the same time the demand for computational power is steadily increasing even though it is hidden from the average user through cloud computing and server applications. Manycore processors can provide high throughput for highly parallel workloads, making them ideal candidates for running multi-threaded parallel applications, but they are constrained by their strict power/thermal budgets. Usually, in highly parallel applications, there is an initial section that spawns the threads that will handle the parallel task. The main workload is split in chunks and parallelized among the available resources. In a homogeneous manycore platform all of the

resources i.e. the cores, have the same characteristics and properties and the workload is distributed as equally as possible. At the end of the data processing, there is typically a barrier to synchronize the cores.

Near Threshold Computing (NTC), a well-known technique for increasing energy efficiency with the drawback of performance degradation, can be used for this kind of applications running on manycore platforms because, as it has been shown [51], it provides operation at the most energy efficient point and it can compensate for the performance loss by squeezing all the available parallelism. The target performance metric for highly parallel applications is throughput and the main goals are maximizing it given a power constraint or minimizing power consumption while meeting a desired throughput constraint.

Several studies have been made in this topic, like in [58] where the authors maximize throughput by balancing power using linear integer programming, or in [30] where throughput maximization is done by creating a model for predicting the power to frequency relation.

The per-core throughput imbalance that can be caused is usually neglected, however, this cannot be overlooked in NTC, because the underlying variability, which is also exacerbated in each subsequent technology node, has a great impact on both performance and power consumption. In [9], a 30% deviation in frequency among the cores of an Intel 80-core manycore platform has been observed in real, on-silicon measurements. The results are referring to a 65nm technology and a much higher voltage than a near-threshold one, thus the variation is expected to be much worse at the NT region. In multithreaded parallel applications, there can even exist an IPC imbalance caused by the access latency irregularities and the per-thread data workload variation leading to different memory access patterns. Given the experimental setup described in Section 5.4.1, the per-core, normalized to the minimum value, throughput is shown in Figure 5.1. The IPCs are obtained from simulations and no process variation is considered, i.e. all the cores run at the same frequency. The standard deviation (σ) is around 11%, but this can be worsened by the actual maximum per-core frequencies caused by process variation. The per-core throughput (normalized to the minimum value), when variability is taken into account, is depicted in Figure 5.2 for the Cholesky benchmark run on 64 cores. Not only the core exhibiting the least throughput is now a different one but also the variation in per-core throughput has increased dramatically, leading to a standard deviation of 38%.

The irregular behavior described above leads us to the conclusion that additional power is wasted from the faster cores even though not needed

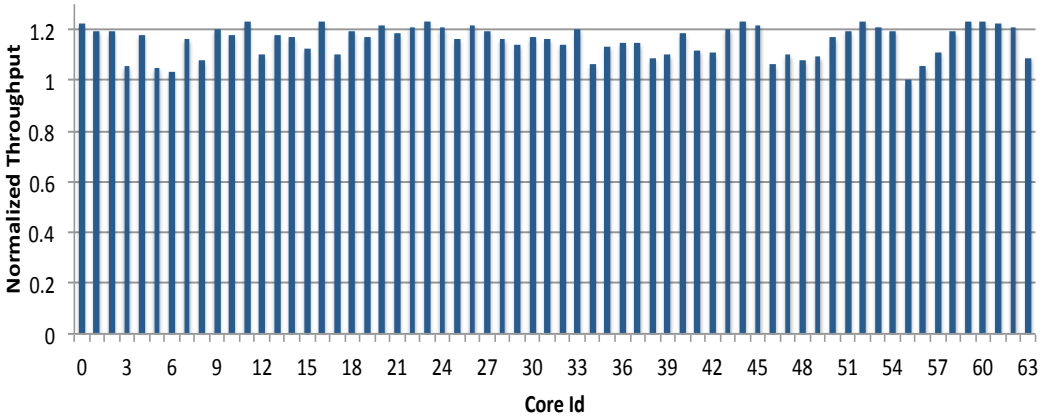


Figure 5.1: *Per-core throughput: imbalance due to IPC variability.*

since they will finish sooner and wait at the synchronization barrier. This observation, creates both a problem that has to be taken care of and an opportunity that should be exploited. Our proposed algorithm manages to mitigate the throughput imbalance and provides up to 43.5% of power savings, on average, for the reported results. Additionally, we are able to identify for each benchmark which is the resource allocation that maximizes energy efficiency, a decision that depends heavily on their unique workload characteristics.

The main contributions of this chapter are:

- Analyzing the per-core throughput imbalance caused by the per-core IPC and process variability.
- Introducing an efficient, low overhead algorithm for sustaining application throughput while improving energy efficiency.
- Investigating the optimal resource allocation and voltage/frequency assignment for maximizing the energy efficiency of highly parallel applications.

5.2 State of the Art

The NTC paradigm has recently emerged as an extremely optimized solution for energy-efficient systems through low-voltage processing. Early studies [11], [51], have shown that near-threshold voltage operation forms an optimal design point in respect to energy and performance efficiency.

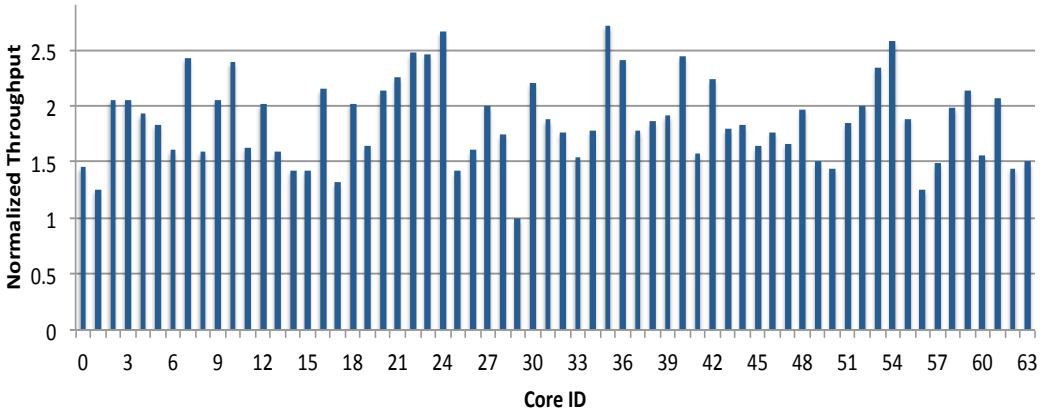


Figure 5.2: *Per-core throughput: imbalance due to process variability.*

Early NT voltage processor prototypes [18], [36] have been recently presented validating the theoretical premises, while several studies show the high efficiency of NTC for cloud- [61] and server-based workloads [48]. More specifically, in [18], the authors present a 128-core manycore design, that combines 3D integration and near-threshold computing and delivers impressive results regarding energy efficiency. In [61], emerging scale-out applications are evaluated and it is demonstrated that building tiled out-of-order chip multiprocessors under NTC is more preferable than conventional designs since it can efficiently utilize the chip level resource and deliver the optimal balance between performance and energy consumption. Finally, the authors in [48] demonstrate that significant improvements in energy efficiency can be achieved, while meeting the strict QoS requirements of scale-out server workloads. They base their work on the novel FD-SOI technology instead of the traditional bulk one, which provides a much better behavior in terms of performance and energy efficiency for transistors operating at low voltage [52], [29] as well as gives a better control and handle over the mitigation of process variation [54]. All the previous are very important observations because they show that NTC has a true potential for driving the future's energy efficient manycore design for emerging parallel workloads and applications.

Targeting mainly manycore architectures, NTC imposes several challenges regarding the application mapping and the resource/power management due to its increased sensitivity to parametric variation [35]. In [33], Karpuzcu et al. proposed a single-voltage multiple-frequency power management scheme for clustered Near-Threshold Voltage (NTV) many-cores,

aiming at the overhead minimization of on-chip power regulation. They explore the efficiency of voltage island domains and propose a multiple-voltage single-frequency power management scheme to mitigate within-die variability at NTC voltages. Recently, the use of machine-learning techniques has been proposed for voltage-frequency allocation in wide-range processors [30], while in [58] the authors proposed a variability aware thread balancing scheme for throughput maximization under power and thermal constraints.

Previous approaches focused on voltage-frequency allocation, without taking into consideration the effects of resource allocation decisions. In this capter, we extend prior-art in NTC by considering a resource aware voltage allocation scheme. More specifically, we model and explore the combined impact that power delivery, variability and allocated resources have on the performance and energy efficiency of NT voltage manycores, proposing a fast runtime algorithm for effective voltage-frequency allocation.

5.3 Throughput Balancing in Near-Threshold Manycores

This section describes a throughput balancing methodology adapted and tuned for near-threshold manycores. It represents a runtime management scheme based on a lightweight optimization algorithm that takes into account realistic models for power delivery, process and workload variability. The following sections describe in more detail the models and algorithms utilized.

5.3.1 Power Delivery System

In most of the research conducted around NTC, the power delivery architecture is not considered and it is assumed that the near-threshold voltages can be delivered without any losses. However, the PDN can have a large overhead on the power consumption if not designed and customized for the specific platform. In [57], the authors, based on the observation that Per-Core Power-Gating (PCPG) devices augmented with feedback control circuitry can serve as low-cost Low-Dropout Regulator (LDO), show that power efficiency as high as that delivered by on-chip switching Voltage Regulators (VRs) can be achieved. The per-core area overhead is only 2%, making the implementation of this design affordable even for a manycore platform. An LDO is a type of linear regulator, where its efficiency is defined as:

$$\eta_{ldo} = \frac{P_{out}}{P_{in}} = \frac{I_{out}V_{out}}{I_{in}V_{in}} = \frac{I_{out}V_{out}}{(I_{out} + I_q)V_{in}} \tag{5.1}$$

where I_q is the quiescent current flowing to the ground:

$$I_{in} = I_{out} + I_q \tag{5.2}$$

The LDOs exhibit current efficiencies up to 99%, making the difference between the input and the output voltage the dominant factor of Equation 5.1. The lower the difference, the higher the efficiency, however the difference should not drop below a certain threshold because the circuit ceases to regulate its output voltage against any fluctuations in the input voltage. This threshold is defined as the *dropout voltage* and is defined at design time. The main advantages of an LDO are its fast transient response and its low cost, making it a potential candidate for on-chip VR. In [47] , the authors design and implement an LDO for near-threshold logic circuits. Its target design characteristics are: 0.5V input voltage, 0.35V - 0.45V output voltage range, 0.05V dropout voltage and 98.7% current efficiency.

5.3.2 Proposed Algorithm

An abstract view of the target tile-based manycore architecture, as well as the intra-tile organization, is depicted in Figure 4.1a: each tile is composed of 4 cores and the last level cache (LL$) is shared among all the cores in the tile. As shown in Figure 5.2, the throughput achieved among the different cores differs significantly. For this reason we designed an algorithm for dealing with this discrepancy (Algorithm 1). The IPC of each core is provided from our simulations in Sniper and can be obtained at runtime by the performance counters provided by the processor. The relationship between frequency and the transistor characteristics is depicted in Equation 5.3, where β is a technology-dependent constant ($\approx$ 1.5), V_{th} and L_{eff} is the transistors' threshold voltage and the effective length respectively.

$$f_i \propto \frac{(V_{dd}^{(i)} - V_{th}^{(i)})^{\beta}}{V_{dd}^{(i)} \times L_{eff}^{(i)}} \tag{5.3}$$

Initially, we calculate the maximum per-core throughput that can be achieved assuming that the maximum voltage and frequency is allocated to them. The throughput of the i_{th} core is calculated as in Equation 5.4.

$$Throughput_i = IPC_i \cdot F_i \tag{5.4}$$

where IPC_i and F_i are the Instructions Per Cycle (IPC) and Frequency of the i_{th} core respectively.

Algorithm 1: Algorithm for Balancing Throughput

Input : Per-core IPC (IPC_i), LDO voltage levels (LDO^j_{out}), Variability modeling parameters (V^i_{th}, L^i_{eff})
Output: Per-core $Voltage$ and $Frequency$ allocation for balancing throughput

1 $V_{max} \longleftarrow max(V^j_{LDOout})$
2 **foreach** $i \in Cores$ **do**
3 $\quad F_i \longleftarrow f(V_{max}, V^i_{th}, L^i_{eff})$
4 $\quad T_i \longleftarrow F_i \cdot IPC_i$
5 **end**
6 $T_{min} \longleftarrow min(T_i)$
7 **foreach** $i \in Cores$ **do**
8 $\quad F_i \longleftarrow \frac{T_{min}}{IPC_i}$
9 $\quad Vdd_i \longleftarrow f^{-1}(F_i, V^i_{th}, L^i_{eff})$
10 **end**

In the pseudocode, this first phase is shown in lines 1-4: given the maximum output voltage of the LDOs $max(V^j_{LDOout})$ (line 1), the maximum achievable frequency is assigned (line 3) and the equivalent throughput is calculated (line 4) for each core. The core exhibiting the minimum throughput is now found and becomes the reference value and bottleneck (line 6). For the rest of the cores, the minimum frequency (and hence voltage) for decreasing their throughput as close as possible to the minimum (found in the previous step), is calculated using Equations 5.4 and 5.3 respectively (lines 8-9). Assuming that the LDOs have discrete output voltage levels, we select the lowest level possible that is providing a higher value than the ideal V_{dd} calculated, ensuring in that way that the desired frequency can be reached. For our experiments we assumed that the LDOs have a 12.5 mV output voltage resolution, meaning that they can provide the desired voltage in steps of the aforementioned voltage value. As mentioned in the previous subsection (5.3.1), the LDOs provide an output voltage range of 100mV (0.35V - 0.45V), offering, as a consequence, 8 discrete voltage levels.

The algorithm described above sustains and guarantees the throughput exhibited from each application (in the sense that its goal is not to maximize it like in other research attempts), while minimizing the overall power consumption. It introduces a low overhead and can be run at runtime while it scales linearly with the number of cores, i.e. $\mathcal{O}(n)$ complexity. Additionally, each per-core computation (throughput calculation and voltage/frequency allocation) is orthogonal to each other and can be computed fully in parallel. In this chapter, it is applied to homogeneous and symmetrical workloads but it can be applied to unbalanced ones considering each

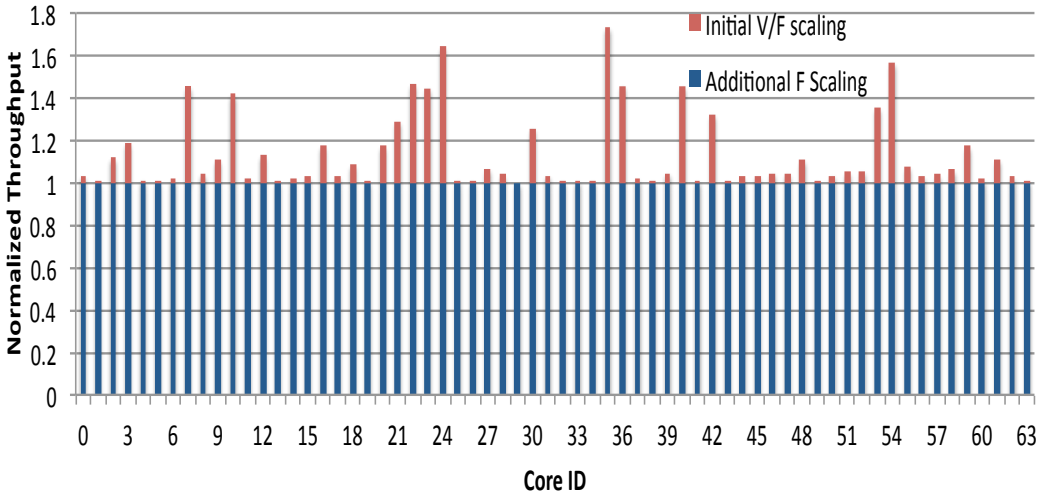


Figure 5.3: *Per-core throughput: balancing obtained by the proposed algorithm.*

time the target platform (e.g. cloud, server etc.), the application scenario (e..g single/multiple applications/instances, number of threads etc.) and the designer's goals, but this is left as future work.

5.4 Experimental Results

5.4.1 Experimental Setup

The experimental setup is similar to the one described in Section 4.4. We use the Sniper manycore simulator [5] for obtaining the performance counters and McPAT [40] for the initial power estimation. Since McPat is not validated against near-threshold voltages, we use it for obtaining the ST values and then we scale accordingly in order to calculate the near threshold ones, using the models for NTC provided in [43]. In order to characterize the process variation at the NT regime we deployed the Varius-NTV microarchitectural model [32]. The applications simulated are taken from the Splash-2 [2] and Parsec [3] benchmark suites, which provide high performance parallel workloads. Each application has different characteristics and exhibits a different behavior, i.e. performance, depending on the number of allocated resources, i.e.cores. Since we are simulating homogeneous multi-threaded applications we consider a discrete number of configurations with the following number of cores: 4, 8, 16, 24, 32, 64.

A summary of the experimental setup used to evaluate the methodology

Table 5.1: *Experimental Setup (Chapter 5): Platform Parameters*

Parameters	Value
Process Technology	22nm
NTC Supply Voltage	0.45V
Nominal $V_{th}/\sigma_{V_{th}}$	0.23V/0.025
Number of Cores/Core Area	64/6mm^2
Tile/	4cores
Private Cache Size/Area	320KB/4.14mm^2
Last Level Cache Size – Area	8 MB / 15.52mm^2

is presented in Table 5.1. Intel's Nehalem architecture (22nm technology node) has been adopted for determining the core and cache types and sizes. The maximum number of cores is 64, divided in tiles consisting of 4 cores and a shared last level cache (LL$) of 8MB and each core owns a private instruction and data cache (P$).

5.4.2 Throughput Balancing

Figure 5.3, depicts the per-core throughput, normalized to the minimum value, after running the algorithm proposed in the previous section. In order to evaluate its efficiency we define a measure similar to standard deviation (σ) by substituting the mean with the minimum. Its purpose is to measure the distance of the throughput values from the minimum for each core. This is because our goal is to balance per-core throughput as much as possible w.r.t. the core with the lowest throughput. We define the following measure:

$$\zeta = \sqrt{\sum_i \frac{(T_i - T_{min})^2}{\#cores}} \tag{5.5}$$

Whereas the throughput ζ-deviation, as defined in Equation 5.5, for the 16-threaded Cholesky application is 77.8% before balancing, it drops to 10.5% after applying our algorithm. For 64 cores, the same measure is increased to 95.8% before balancing but it is reduced down to 30% after applying our technique (69% reduction). As we can see in Figure 5.3, most of the cores exhibit a similar throughput, except for some of them, represented by the peaks in red color in the graph, e.g. 24, 35, 54, that are much faster and with the specific voltage range provided by the regulators it is impossible to reduce their throughput any further. For each application the ζ-deviation is reduced by 70% on average. The previous results are obtained assuming that, given the allocated voltage, the maximum achievable frequency is assigned to each core. As mentioned before, throughput in

Figure 5.3 is normalized w.r.t. minimum value, therefore, the faster cores can exhibit a quite higher throughput (represented as red colored peaks in the graph) due to the higher frequency that can be assigned to them. As an extra, fine tuning optimization, we can slow down those cores, by reducing the frequency in order to perfectly balance the throughput and eliminate the variation completely, obtaining some extra power savings (reported in section 5.4.3).

5.4.3 Energy Efficient Configurations

In this section, for each application, we characterize each configuration in terms of energy efficiency and we find the most efficient one. Energy efficiency, as shown in Equation 5.6, is defined as Throughput over Power and is measured in Instructions/Joule or MIPS/Watt equivalently. After running our algorithm, we can determine the most energy efficient configuration for each application.

$$\textit{Energy Efficiency} = \frac{Throughput}{Power} \tag{5.6}$$

Table 5.2 summarizes the results. Applications have different optimal configuration points (#cores) depending on their workload characteristics, i.e. after a certain point there is no sense in allocating more resources because the performance-throughput is not increased significantly in order to compensate for the increased power consumed by the extra cores. For each optimal configuration, we calculate the power savings achieved by applying our optimization algorithm. The power savings are quite high, 39% on average, (varying from 34% up to 44%) and they become even more significant when considering that the overall performance was not impacted. As mentioned in the previous subsection, by performing an extra frequency scaling for the faster cores, we can eliminate the throughput variation completely, increasing ideally the average power savings from 39% to 43.5% (varying from 37% up to 51%).

Table 5.2: *Energy Efficient Optimal Configuration*

App	#cores	App	#cores
blackscholes	48	canneal	48
fmm	32	ocean	32
fluidanimate	24	raytrace	24
streamcluster	48	cholesky	64
swaptions	8	water.nsq	8
radix	8	barnes	8

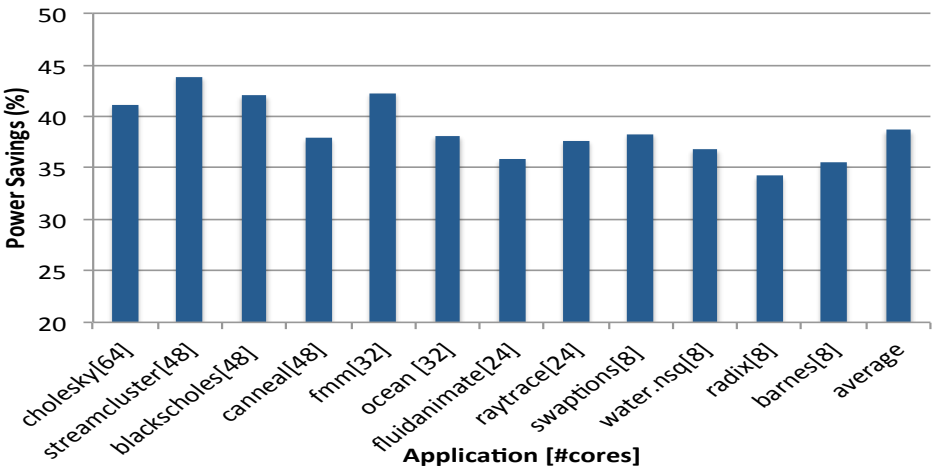


Figure 5.4: *Power savings obtained by the proposed technique.*

5.5 Conclusion

In this chapter, we tackled the problem of throughput imbalance of homogeneous workloads in manycore processors, caused by the excessive impact of process variability at near-threshold (NT) voltages. Considering the power inefficiencies of a scalable and low overhead NT power delivery network, we designed and implemented a runtime algorithm that manages to reduce both the variation of the per-core throughput w.r.t. the minimum and the power consumption, on average, by 70% and 43.5% respectively for the reported results, while not impacting the overall performance.

CHAPTER 6

Conclusions

This thesis has addressed the problem of sustaining the performance of manycore architectures when operating at Near-Threshold voltage. The effects of process variation are exacerbated in NTC and the applications suffer significant losses in terms of performance. Thus, in Part I, we introduced variability-aware framework for exploring the power-efficiency of Near Threshold Computing (NTC) while meeting performance constraints and in Part II we explored the design and optimization of power delivery architectures and runtime optimizations. All of our experiments have been conducted by using the state of art tools and simulators and evaluated on highly parallel, scalable applications. Below, we wrap-up and summarize the conclusions, referencing the publications that each chapter is based on.

In **Chapter 2**, we explored the concept of voltage island formation in order tackle the variability problem in the NT region while not impacting performance. The work is automated by the framework developed for this purpose which allows extensive parametrization and a lot of different configurations to be explored. The main insight here is that NTC can deliver significant gains for multithreaded applications, but its high dependency on both workload characteristics and underlying architectural organization has to be considered carefully.

- ***I. Stamelakos**, S. Xydis, G. Palermo, C. Silvano. "Variation aware voltage island formation for power efficient near-threshold manycore architectures." In Proceedings of the ASP-DAC, ASP-DAC '14, 2014.*

- ***I. Stamelakos**, S. Xydis, G. Palermo, C. Silvano. "Variability-Aware Voltage Island Management for Near-Threshold Computing with Performance Guarantees". Springer Book: Near Threshold Computing - Technology, Methods and Applications (2015)*

In **Chapter 3**, we extended our set of voltage tuning and allocation techniques and strategies, to both thread-parallel and process-parallel workloads, while always meeting the performance constraints and we showed that super-threshold performance can be efficiently sustained or even surpassed at the NT regime. This is a very important observation, because it proves that NTC when combined with the appropriate design strategies can be used to efficiently alleviate the dark silicon problem.

- *C. Silvano, G. Palermo, S. Xydis, **I. Stamelakos**, "Voltage Island Management in Near Threshold Many-core Architectures to Mitigate Dark Silicon". In Proceedings of DATE '14, 2014.*

- ***I. Stamelakos**, S. Xydis, G. Palermo, C. Silvano. "Variability-Aware Voltage Island Management for Near-Threshold Computing with Performance Guarantees". Springer Book: Near Threshold Computing - Technology, Methods and Applications (2015)*

In **Chapter 4**, motivated by the lack of research and novel ideas targeting power delivery in NTC, we evaluated the existing power delivery architectures for near-threshold manycores under process variation and we showed that not only is it possible to deliver efficiently the near-threshold voltages in manycore architectures, but there is plenty of space for optimizations when taking into account the workload characteristics of the target applications at design time.

- ***I. Stamelakos**, A. Khajeh, G. Palermo, C. Silvano, F. Kurdahi. "A System-Level Exploration of Power Delivery Architectures for Near-Threshold Manycores Considering Performance Constraints". IEEE Computer Society Annual Symposium on VLSI, Pittsburgh, Pennsylvania, U.S.A., July 11-13, 2016*

Finally, in **Chapter 4**, we addressed the problem of throughput imbalance of homogeneous workloads in manycore processors. We propose a

runtime management scheme for improving NTC manycore energy efficiency and by considering the power inefficiencies of a scalable and low overhead NT power delivery network, we managed to significantly reduce both the variation of the per-core throughput w.r.t. the minimum and the power consumption, while not impacting the overall performance. The main conclusion and future motivation is that there is still a lack of research in runtime management and optimizations in Near-Threshold Computing, which leaves a great potential unexploited.

- ***I. Stamelakos**, S. Xydis, G. Palermo, C. Silvano. Throughput Balancing for Energy Efficient Near-Threshold Manycores. PATMOS 2016, 26th International Workshop on Power and Timing Modeling, Optimization and Simulation, Bremen, Germany, 2016*

Bibliography

[1] Kanak Agarwal and Kevin Nowka. Dynamic power management by combination of dual static supply voltages. In *Quality Electronic Design, 2007. ISQED'07. 8th International Symposium on*, pages 85–92. IEEE, 2007.

[2] Jeffrey M Arnold, Duncan A Buell, and Elaine G Davis. Splash 2. In *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*, pages 316–322. ACM, 1992.

[3] Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li. The parsec benchmark suite: Characterization and architectural implications. In *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, pages 72–81. ACM, 2008.

[4] S. Borkar. The exascale challenge. In *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*, pages 2–3, 2010.

[5] Trevor E. Carlson, Wim Heirman, and Lieven Eeckhout. Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2011.

[6] L. Chang, R.K. Montoye, Y. Nakamura, K.A. Batson, R.J. Eickemeyer, R.H. Dennard, W. Haensch, and D. Jamsek. An 8t-sram for variability tolerance and low-voltage operation in high-performance caches. *Solid-State Circuits, IEEE Journal of*, 43(4):956–963, 2008.

[7] A. Das, S. Ozdemir, G. Memik, and A. Choudhary. Evaluating voltage islands in cmps under process variations. In *Computer Design, 2007. ICCD 2007. 25th International Conference on*, pages 129–136, 2007.

[8] R.H. Dennard, F.H. Gaensslen, V.L. Rideout, E. Bassous, and A.R. LeBlanc. Design of ion-implanted mosfet's with very small physical dimensions. *Solid-State Circuits, IEEE Journal of*, 9(5):256–268, 1974.

[9] S Dighe, Sriram Vangal, Paolo Aseron, Sudhakar Kumar, Tony Jacob, Keith Bowman, John Howard, James Tschanz, V Erraguntla, N Borkar, et al. Within-die variation-aware dynamic-voltage-frequency scaling core mapping and thread hopping for an 80-core processor. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pages 174–175. IEEE, 2010.

[10] 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.

[11] Ronald G. Dreslinski, Michael Wieckowski, David Blaauw, Dennis Sylvester, and Trevor N. Mudge. Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266, 2010.

[12] Ronald G. Dreslinski, Bo Zhai, Trevor N. Mudge, David Blaauw, and Dennis Sylvester. An energy efficient parallel architecture using near threshold operation. In *PACT*, pages 175–188, 2007.

[13] M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf. The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 5(4):360–368, 1997.

[14] Hadi Esmaeilzadeh, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th annual international symposium on Computer architecture*, ISCA '11, pages 365–376, 2011.

[15] J. Howard et al. A 48-core ia-32 message-passing processor with dvfs in 45nm cmos. In *ISSCC*, pages 108–109. IEEE, 2010.

[16] S. Dighe et al. Within-die variation-aware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-core teraflops processor. *J. Solid-State Circuits*, 46(1):184–193, 2011.

[17] Gregory G. Faust, Runjie Zhang, Kevin Skadron, Mircea R. Stan, and Brett H. Meyer. Archfp: Rapid prototyping of pre-rtl floorplans. In Srinivas Katkoori, Matthew R. Guthaus, Ayse Kivilcim Coskun, Andreas Burg, and Ricardo Reis, editors, *VLSI-SoC*, pages 183–188. IEEE, 2012.

[18] David Fick, Ronald G Dreslinski, Bharan Giridhar, Gyouho Kim, Sangwon Seo, Matthew Fojtik, Sudhir Satpathy, Yoonmyung Lee, Daeyeon Kim, Nurrachman Liu, et al. Centip3de: A 3930dmips/w configurable near-threshold 3d stacked system with 64 arm cortex-m3 cores. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pages 190–192. IEEE, 2012.

[19] Wei Fu and Ayman Fayed. A feasibility study of high-frequency buck regulators in nanometer cmos technologies. In *Circuits and Systems Workshop,(DCAS), 2009 IEEE Dallas*, pages 1–4. IEEE, 2009.

[20] Tobias Gemmeke, Mohamed M. Sabry, Jan Stuijt, Praveen Raghavan, Francky Catthoor, and David Atienza. Resolving the memory bottleneck for single supply near-threshold computing. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '14, 2014.

[21] Hamid Reza Ghasemi, Abhishek A. Sinkar, Michael J. Schulte, and Nam Sung Kim. Cost-effective power delivery to support per-core voltage domains for power-constrained processors. In *Proceedings of the 49th Annual Design Automation Conference*, DAC '12, pages 56–61, 2012.

[22] Waclaw Godycki, Christopher Torng, Ivan Bukreyev, Alyssa Apsel, and Christopher Batten. Enabling realistic fine-grain voltage scaling with reconfigurable power distribution networks. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 381–393. IEEE Computer Society, 2014.

[23] N. Goulding-Hotta, J. Sampson, G. Venkatesh, S. Garcia, J. Auricchio, P. Huang, M. Arora, S. Nath, V. Bhatt, J. Babb, S. Swanson, and M.B. Taylor. The greendroid mobile application processor: An architecture for silicon's dark future. *Micro, IEEE*, 31(2):86–95, 2011.

[24] V. Govindaraju, Chen-Han Ho, and K. Sankaralingam. Dynamically specialized datapaths for energy efficient computing. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 503–514, 2011.

[25] Intel fourth generaton Core CPU Haswell. FIVR Fully Integrated Voltage Regulator, http://www.intel.com, 2013.

[26] Xin He, Guihai Yan, Yinhe Han, and Xiaowei Li. Superrange: wide operational range power delivery design for both stv and ntv computing. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–6. IEEE, 2014.

[27] Sebastian Herbert, Siddharth Garg, and Diana Marculescu. Exploiting process variability in voltage/frequency control. *IEEE Trans. VLSI Syst.*, 20(8):1392–1404, 2012.

[28] Wei-Chih Hsieh and Wei Hwang. All digital linear voltage regulator for super-to near-threshold operation. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 20(6):989–1001, 2012.

[29] David Jacquet, Frederic Hasbani, Philippe Flatresse, Robin Wilson, Franck Arnaud, Giorgio Cesana, Thierry Di Gilio, Christophe Lecocq, Tanmoy Roy, Amit Chhabra, et al. A 3 ghz dual core processor arm cortex tm-a9 in 28 nm utbb fd-soi cmos with ultra-wide voltage range and energy efficiency optimization. *IEEE Journal of Solid-State Circuits*, 49(4):812–826, 2014.

[30] Da-Cheng Juan, Shelly Garg, Jinpyo Park, and Diana Marculescu. Learning the optimal operating point for many-core systems with extended range voltage/frequency scaling. In *Hardware/-Software Codesign and System Synthesis (CODES+ ISSS), 2013 International Conference on*, pages 1–10. IEEE, 2013.

[31] D. Kanter. Inside nehalem: Intel future processor and system. http://www.realworldtech.com, 2008.

[32] Ulya R. Karpuzcu, Krishna B. Kolluru, Nam Sung Kim, and Josep Torrellas. Varius-ntv: A microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages. In *IEEE/IFIP International Conference on Dependable Systems and Networks, DSN*, pages 1–11, 2012.

[33] Ulya R Karpuzcu, Abhishek Sinkar, Nam Sung Kim, and Josep Torrellas. Energysmart: Toward energy-efficient manycores for near-threshold computing. In *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pages 542–553. IEEE, 2013.

[34] Ulya R. Karpuzcu, Abhishek A. Sinkar, Nam Sung Kim, and Josep Torrellas. Energysmart: Toward energy-efficient manycores for near-threshold computing. In *HPCA*, pages 542–553, 2013.

[35] 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.

[36] Surhud Khare and Shailendra Jain. Prospects of near-threshold voltage design for green computing. In *2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems*, pages 120–124. IEEE, 2013.

[37] Wonyoung Kim, David M Brooks, and Gu-Yeon Wei. A fully-integrated 3-level dc/dc converter for nanosecond-scale dvs with fast shunt regulation. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 268–270. IEEE, 2011.

[38] Sae Kyu Lee, David Brooks, and Gu-Yeon Wei. Evaluation of voltage stacking for near-threshold multicore computing. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, pages 373–378. ACM, 2012.

[39] Woojoo Lee, Yanzhi Wang, and Massoud Pedram. Vrcon: dynamic reconfiguration of voltage regulators in a multicore platform. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 365. European Design and Automation Association, 2014.

[40] Sheng Li, Jung Ho Ahn, Richard D. Strong, Jay B. Brockman, Dean M. Tullsen, and Norman P. Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, pages 469–480, 2009.

[41] Sohaib S. Majzoub, Resve A. Saleh, Steven J. E. Wilton, and Rabab K. Ward. Energy optimization for many-core platforms: communication and pvt aware voltage-island formation and voltage selection algorithm. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 29(5):816–829, May 2010.

[42] D. Markovic, C.C. Wang, L.P. Alarcon, Tsung-Te Liu, and J.M. Rabaey. Ultralow-power design in near-threshold region. *Proceedings of the IEEE*, 98(2):237–252, 2010.

[43] Dejan Marković, Cheng C Wang, Louis P Alarcon, Tsung-Te Liu, and Jan M Rabaey. Ultralow-power design in near-threshold region. *Proceedings of the IEEE*, 98(2):237–252, 2010.

[44] Timothy N Miller, Xiang Pan, Renji Thomas, Naser Sedaghati, and Radu Teodorescu. Booster: Reactive core acceleration for mitigating the effects of process variation and application imbalance in low-voltage chips. In *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, pages 1–12. IEEE, 2012.

[45] Robert J Milliken, Jose Silva-Martínez, and Edgar Sánchez-Sinencio. Full on-chip cmos low-dropout voltage regulator. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(9):1879–1890, 2007.

[46] Gordon E Moore et al. Cramming more components onto integrated circuits, 1965.

[47] Yasuyuki Okuma, Koichi Ishida, Yoshikatsu Ryu, Xin Zhang, Po-Hung Chen, Kazunori Watanabe, Makoto Takamiya, and Takayasu Sakurai. 0.5-v input digital ldo with 98.7% current efficiency and 2.7-μa quiescent current in 65nm cmos. In *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, pages 1–4. IEEE, 2010.

[48] Ali Pahlevan, Javier Picorel, Arash Pourhabibi Zarandi, Davide Rossi, Marina Zapater, Andrea Bartolini, Pablo G Del Valle, David Atienza, Luca Benini, and Babak Falsafi. Towards near-threshold server processors. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 7–12. IEEE, 2016.

[49] Francesco Paterna and Sherief Reda. Mitigating dark-silicon problems using superlattice-based thermoelectric coolers. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, pages 1391–1394, San Jose, CA, USA, 2013. EDA Consortium.

[50] N. Pinckney, K. Sewell, R. G. Dreslinski, D. Fick, T. Mudge, D. Sylvester, and D. Blaauw. Assessing the performance limits of parallelized near-threshold computing. In *Proceedings of the 49th Design Automation Conference*, pages 1147–1152, 2012.

[51] Nathaniel Pinckney, Korey Sewell, Ronald G Dreslinski, David Fick, Trevor Mudge, Dennis Sylvester, and David Blaauw. Assessing the performance limits of parallelized near-threshold computing. In *Proceedings of the 49th Annual Design Automation Conference*, pages 1147–1152. ACM, 2012.

[52] Nicolas Planes, Oliver Weber, V Barral, S Haendler, D Noblet, D Croain, M Bocat, P-O Sassoulas, X Federspiel, A Cros, et al. 28nm fdsoi technology platform for high-speed low-voltage digital applications. In *VLSI Technology (VLSIT), 2012 Symposium on*, pages 133–134. IEEE, 2012.

[53] Arun Raghavan, Yixin Luo, Anuj Chandawalla, Marios C. Papaefthymiou, Kevin P. Pipe, Thomas F. Wenisch, and Milo M. K. Martin. Computational sprinting. In *HPCA*, pages 249–260. IEEE, 2012.

[54] Davide Rossi, Antonio Pullini, Igor Loi, Michael Gautschi, Frank K Gürkaynak, Andrea Bartolini, Philippe Flatresse, and Luca Benini. A 60 gops/w,- 1.8 v to 0.9 v body bias ulp cluster in 28nm utbb fd-soi technology. *Solid-State Electronics*, 117:170–184, 2016.

[55] S.R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas. Varius: A model of process variation and resulting timing errors for microarchitects. *Semiconductor Manufacturing, IEEE Transactions on*, 21(1):3–13, 2008.

[56] A. Sasan, H. Homayoun, A. M. Eltawil, and F. J. Kurdahi. Inquisitive defect cache: A means of combating manufacturing induced process variation. *IEEE Trans. VLSI Syst.*, 19(9):1597–1609, 2011.

[57] Abhishek A Sinkar, Hamid Reza Ghasemi, Michael J Schulte, Ulya R Karpuzcu, and Nam Sung Kim. Low-cost per-core voltage domain support for power-constrained high-performance processors. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 22(4):747–758, 2014.

[58] Abhishek A Sinkar, Hao Wang, and Nam Sung Kim. Maximizing throughput of power/thermal-constrained processors by balancing power consumption of cores. In *Quality Electronic Design (ISQED), 2014 15th International Symposium on*, pages 633–638. IEEE, 2014.

[59] Dean N Truong, Wayne H Cheng, Tinoosh Mohsenin, Zhiyi Yu, Anthony T Jacobson, Gouri Landge, Michael J Meeuwsen, Christine Watnik, Anh T Tran, Zhibin Xiao, et al. A 167-processor computational platform in 65 nm cmos. *Solid-State Circuits, IEEE Journal of*, 44(4):1130–1144, 2009.

[60] Yatish Turakhia, Bharathwaj Raghunathan, Siddharth Garg, and Diana Marculescu. Hades: architectural synthesis for heterogeneous dark silicon chip multi-processors. In *DAC*, page 173. ACM, 2013.

[61] Jing Wang, Junwei Zhang, Weigong Zhang, Keni Qiu, Tao Li, and Minhua Wu. Near threshold cloud processors for dark silicon mitigation: the impact on emerging scale-out workloads. In *Proceedings of the 12th ACM International Conference on Computing Frontiers*, page 4. ACM, 2015.

[62] Steven Cameron Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Pal Singh, and Anoop Gupta. The splash-2 programs: characterization and methodological considerations. *SIGARCH Comput. Archit. News*, 23(2):24–36, May 1995.

[63] Zhiyu Zeng, Xiaoji Ye, Zhuo Feng, and Peng Li. Tradeoff analysis and optimization of power delivery networks with on-chip voltage regulation. In *Proceedings of the 47th Design Automation Conference*, pages 831–836. ACM, 2010.