View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Illinois Digital Environment for Access to Learning and Scholarship Repository

© 2009 Rahmet Ulya Karpuzcu

MANAGING MANY-CORE AGING

BY

RAHMET ULYA KARPUZCU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Adviser:

Professor Josep Torrellas

ABSTRACT

Many-core scaling now faces a power wall. The gap between the number of cores that fit on a die and the number that can operate simultaneously under the power budget is rapidly increasing with technology scaling. In future designs, the majority of the cores will necessarily have to be dormant at any given time to meet the power budget.

To push back the many-core power wall, this work introduces *Dynamic Voltage Scaling for Aging Management* (DVSAM) — a new scheme for trading off processor aging for performance and power. DVSAM can be used to maximize performance, minimize power, or boost performance for a short life. In addition, this work introduces the *BubbleWrap* many-core, an architecture that makes use of DVSAM. BubbleWrap identifies the most power-efficient cores on a variation-affected chip and designates them as *Throughput* cores dedicated to parallel-section execution; the rest of the cores (*Expendable* cores) are dedicated to sequential sections. In one use of DVSAM, BubbleWrap sacrifices Expendable cores one at a time by running them at elevated Vdd for a month or so each, until they completely wear out. Our simulations show that a 32-core BubbleWrap many-core provides substantial improvements over a plain chip. For example, on average, one design runs fully sequential applications at a 22% higher frequency, and fully parallel applications with a 33% higher throughput.

To His Eminency, Supreme Potentate Pater, Lord High Emperor, for sparing my life in order that I might complete this thesis

ACKNOWLEDGMENTS

My collegue, Brian Greskamp retains at least equal credit for this thesis. Special thanks in addition go to Josep Torrellas, my trusted adviser, for invaluable consultation and direction and for patience with my glacial progress.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION **1**

CHAPTER 2 BACKGROUND **4**
2.1 Modeling Processor Aging 4
2.2 Hiding/Slowing Down Processor Aging 5

CHAPTER 3 DVSAM: DYNAMIC VOLTAGE SCALING FOR AGING MANAGEMENT **8**

CHAPTER 4 THE BUBBLEWRAP MANY-CORE **12**
4.1 Overview of the BubbleWrap Many-Core 12
4.2 Example BubbleWrap Environments 13
4.3 Hardware Support for Aging Management 15

CHAPTER 5 EVALUATION SETUP **17**
5.1 Process Technology 17
5.2 Workload 18
5.3 Many-Core Microarchitecture 18
5.4 BubbleWrap Controller 19

CHAPTER 6 EVALUATION **20**
6.1 Enhancing Throughput: DVSAM-Pow 20
6.2 Enhancing Frequency: DVSAM-Perf 22
6.3 Popping Cores: VSAM-Short, DVSAM-Short 24
6.4 Performance Analysis 24

CHAPTER 7 RELATED WORK **29**

CHAPTER 8 CONCLUSION AND FUTURE WORK **30**

REFERENCES **31**

CHAPTER 1

INTRODUCTION

Ideal CMOS device scaling [1] relies on scaling voltages down with lithographic dimensions at every technology generation. In this case, the power per unit area stays constant, since the energy per switching event decreases enough to compensate for the increased energy of having more gates in the area and having them switch faster. In recent generations, however, to keep leakage current under control, the decrease in the transistor's threshold voltage (Vth) has stopped, which in turn has prevented the supply voltage (Vdd) from scaling [2].

A direct consequence of this fact is that chip power does not scale. As more transistors are integrated on a fixed-sized chip at every generation, the power generated in the chip increases rapidly. If we fix the chip power budget for cost considerations, we easily realize that there is a growing gap between what can be placed on a chip and what can be powered up simultaneously. For example, Figure 1.1 shows data computed from the ITRS 2008 update [3] assuming beefy cores and a 100 W chip power budget. The figure compares the number of cores that can be placed on a chip at a given year and the number of those that can be powered up simultaneously. The growing gap between two curves shows the *many-core power wall*. While we have long been told that cores will come for free, they will in fact come for free and *powered off*.

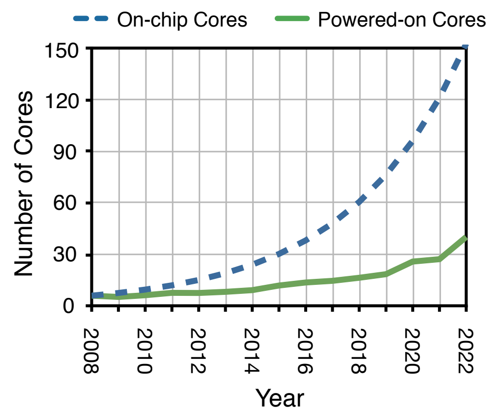


Figure 1.1 The many-core power wall, based on data from ITRS projections

Another key trend in devices is that of increasing parameter variation [4]. Such variation can be spatial across the chip or temporal with use. The latter is often referred to as device wear out or aging. Aging is observed in the progressive slowdown of logic as it is being used. Aging is due to a variety of mechanisms [4].

Recently, processor aging has been the subject of much work [5–11]. It is well accepted that the aging rate is highly impacted by Vdd and temperature — higher values increase the aging rate. Consequently, some researchers have proposed to reduce aging by applying lower Vdd or *T* [11]. We observe that such approaches to reduce aging typically trade off aging for other parameters such as power or performance. If such approaches could be properly generalized, they could provide a new way to use the surplus cores of power-limited future many-cores.

Based on this observation, this work proposes a new way to manage processor aging by trading it off for performance and power. We call our scheme *Dynamic Voltage Scaling for Aging Management* (DVSAM). It involves managing the aging rate by continuously tuning the Vdd (but *not* the frequency) throughout the processor's service-life. The goal can be one of the following: consume less power at the same performance and service-life; attain higher performance at the same service-life for the given power budget; or attain much higher performance with a short service-life for the given power budget.

With the use of DVSAM, we propose a novel many-core architecture called *BubbleWrap* that pushes back the many-core power wall. BubbleWrap identifies the most power-efficient set of cores on a variation-affected die — the largest set that can be simultaneously powered up — and designates them as *Throughput* cores dedicated to parallel section execution; the rest of the cores (*Expendable* cores) are dedicated to sequential sections. BubbleWrap applies DVSAM in several modes to improve performance or power. One mode entails sacrificing Expendable cores one at a time by running them at elevated Vdd for a month or so each until they completely wear out.

We simulate a BubbleWrap many-core with 32 cores with a range of DVSAM environments. BubbleWrap provides substantial improvements over a plain chip with the same number of cores and the same power envelope. For example, on average, our most aggressive BubbleWrap design runs fully sequential applications at a 22% higher frequency, and fully parallel applications with a 33% higher throughput.

Overall, this work makes two main contributions:

- We introduce *Dynamic Voltage Scaling for Aging Management* (DVSAM), a new way to trade processor aging for performance and power: Maximize performance, minimize power, or maximize performance for a short life.

- We introduce the *BubbleWrap* many-core, a new many-core architecture that makes use of DVSAM to push back the many-core power wall. In one environment, to speed up serial sections, it runs cores at elevated Vdd for a month or so each until they completely wear out.

The rest of this thesis is organized as follows. Chapter 2 provides background; Chapter 3 introduces DVSAM; Chapter 4 presents the BubbleWrap many-core; Chapters 5 and 6 evaluate BubbleWrap ; and Chapter 7 discusses related work.

CHAPTER 2

BACKGROUND

2.1 Modeling Processor Aging

Our analysis focuses on the bias temperature instability (BTI) induced aging, leading to dynamic slowdown in transistors over the course of the service-life of a processor. The performance degradation stems from increases in the threshold voltage over time, which in turn increases gate delays according to alpha-power law (Equation (2.1)). For BTI-induced degradation in Vth, $\Delta Vth \propto time^a$ applies, where a represents the time-slope, a constant strongly related to process characteristics.

$$\tau_g \propto \frac{Vdd}{\mu(T)(Vdd - Vth)^{\alpha}} \qquad \mu(T) \propto T^{-1.5} \tag{2.1}$$

In modeling $\Delta Vth \propto time^a$, we adopt the framework from Wang et al. [12]. Since BTI represents a data-dependent failure mode, Vth degradation is only anticipated under specific bias conditions (stress), and partial recovery occurs when the bias is removed. Assuming constant stress over the service-life, for any given time t, the increase in the threshold voltage due to BTI becomes

$$\Delta Vth_{STRESS} = A_{BTI} \times f_{BTI} \times t^a$$
$$f_{BTI} = \left[\left(\frac{q^3}{Cox^2}\right) \times (Vdd - Vth_{NOM}) \times exp\left(-\frac{Ea}{2kT} + \frac{Vdd - Vth_{NOM}}{tox \times 0.5Eo}\right)\right]^{2a} \tag{2.2}$$

The power law reflects process characteristics along with specifics of operating conditions as captured by Vdd and T. For our analysis, we conservatively deploy Equation (2.2) and exclude the impact of recovery. The following summarizes our power/thermal model:

$$P_{DYNAMIC} \propto C \times Vdd^2 \times f$$
$$P_{STATIC} = Vdd \times I_{LEAK} \qquad I_{LEAK} \propto \mu(T) \times T^2 \times e^{-qVth/kTn} \qquad \mu(T) \propto T^{-1.5} \tag{2.3}$$

$$T_J = T_A + \theta_{JA} \times P \qquad P = P_{STATIC} + P_{DYNAMIC} \tag{2.4}$$

where T_J corresponds to the junction temperature, and θ_{JA} the junction-to-ambient thermal resistance, modeled as a lumped equivalent of junction-to-package and heat-sink-to-ambient thermal resistances.

A Note on the Model Accuracy: The projective nature of our analysis demands proper characterization for future technology generations: (1) We deploy a physically aware version of the basic alpha-power law [13] to capture short channel effects, which get more and more pronounced over technology generations. With corrections incorporated to model undesired shifts in the threshold voltage [14], Vth from Equation (2.1) becomes

$$Vth = Vth_{NOM} + k_{DIBL} \times (Vdd - Vdd_{NOM}) + k_T \times (T - T_{NOM}) + \Delta Vth_{STRESS} \quad (2.5)$$

(2) The aging model from Equation (2.2) is derived for devices with Si-based dielectrics and verified against an industrial 65 nm node [12]. The main issue with this model is that it does not account for transistors with high-k dielectrics and metal gates (HK+MG devices), which have become the standard starting from the 45 nm node onwards.

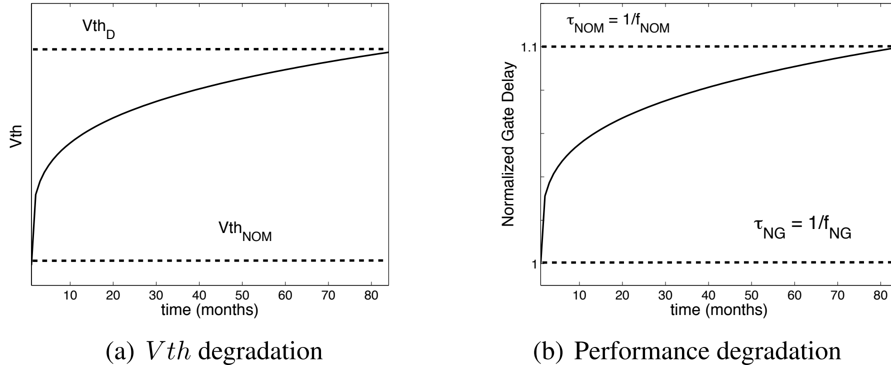
Addressing parasitics as induced by Si-based dielectrics of only a few atomic layers of physical thickness, HK+MG devices should achieve service-lives *at least* as long as their their Si-based counterparts. For Intel's recent 45 nm node [15], it is demonstrated that the BTI characteristics closely follow the aging behavior observed for Si-based devices, specifically, that the same physical phenomena result in this behavior. This justifies the application of the aging model at hand to HK+MG devices. The process parameters and time-slope characteristics of the base model are hence modified to reflect a technology node of interest in light of observations from [15].

Where HK+MG devices have become the state of the art for 45 nm, further radical changes in the device architecture are expected beyond 32 nm. Note that (i) We do not know which device architecture will be accepted for a given technology node. (ii) Even if the device architecture is known, it can be aggressively optimized in various ways. Such device optimizations usually lead to significant changes in device characteristics. The key observation here is that any radical change regarding the device architecture (including optimizations) should lead to devices at least as good as those of preceding generations. In other words, a new technology generation should be (at least) as reliable as its predecessor generations.

2.2 Hiding/Slowing Down Processor Aging

Nominal operating conditions characterize operation at Vdd_{NOM}, f_{NOM} and T_{NOM} throughout the service-life of the core. f_{NOM}, Vdd_{NOM} and T_{NOM} represent the nominal frequency, supply voltage and junction temperature respectively. Vth_{NOM} corresponds to the saturation threshold voltage at T_{NOM}. *Worst-case guard-band*, G, refers to the maximum margin in path delays to mask the aging-induced performance decrease over the service-life of the core. For analysis purposes, other contributors to the guard-band are neglected. We represent the guard-band as a fraction of increase over the *zero-guard-band* period, τ_{NG}. Without aging, no increase in path delays would occur over the service-life. Hence, the core would be able to cycle at the *zero-guard-band* frequency f_{NG}. In other words, $f_{NOM} = f_{NG}/(1 + G)$.

Figure 2.1 shows aging-induced degradation for a traditional design with $G = 10\%$ with gate delays normalized to the *zero-guard-band* period, τ_{NG}, under nominal conditions (specifically, $Vdd = Vdd_{NOM}$). At the beginning of the service-life, $Vth = Vth_{NOM}$, permitting operation at the *zero-guard-band* frequency, f_{NG}. The increase in gate delays at the end of the service-life corresponds to G, where due to the degraded $Vth = Vth_D$, operation at only $f_{NOM} < f_{NG}$ can be afforded. A traditional design thus wastes performance potential at the beginning of the service life by enforcing operation at f_{NOM}. This observation motivates techniques to extract more performance (or save power) by adapting to the dynamic guard-band consumption over the service life.



(a) Vth degradation (b) Performance degradation

Figure 2.1 Aging-induced degradation for $G = 10\%$ and a service-life of 7 years

Facelift [11] provides an adaptive framework for hiding and slowing down aging based on application of two types of techniques: Techniques speeding up path delays at the cost of increased aging rate (*HighSpeed*) and techniques decreasing aging rate at the cost of

increased path delays (*SlowAge*). A key observation here is that *SlowAge* techniques are more effective towards the beginning of the service-life. This is due to the aging rate being higher at the beginning of the service-life, as can be observed from Figures 2.1(a) and 2.1(b). Towards the end of the service-life, the aging rate comes to saturation, and the path delays anticipate very high degradation. At this stage, the negative impact of *High Speed* techniques on the aging rate is minimized (due to the saturation in aging rate), and their boost on path delays is most required (due to the high degradation in path delays). A key question for this framework is when and how often the switch between the techniques should occur over the service-life. The authors introduce a two-step scheme, where the switch occurs at the point minimizing the maximum delay degradation.

Of the techniques proposed within the Facelift framework, we adopt Vdd reduction beyond the nominal at the early stages of the service-life to reduce the aging rate. To speed up paths, we increase Vdd at later stages of the service-life. In contrast to [11], we do not deploy Adaptive Body Biasing (ABB) due to (1) the questionable feasibility of ABB under aggressive scaling [16] and (2) the simplicity and availability of dynamic Vdd management techniques in modern systems.

In the next chapter, we detail the specifics of our aging management scheme. We improve on Facelift [11] by imposing *fine-grain* aging management to trade off performance, power and service-life. Facelift does not provide the power axis. Further, where Facelift proposes a single instantaneous transition from a *SlowAge* technique to a *High-Speed* technique, we introduce a *continuous* fine-grain adaptation over the processor service-life.

CHAPTER 3

DVSAM: DYNAMIC VOLTAGE SCALING FOR AGING MANAGEMENT

We manage the aging rate by continuously tuning the Vdd (but *not* the frequency) throughout the processor's service-life. Table 3.1 summarizes our aging-management tool-set: DVSAM, Dynamic Voltage Scaling for Aging Management. The goal can be one of the following: consume less power at the same performance and service life (DVSAM-Pow); attain higher performance at the same service-life for the given power budget (DVSAM-Perf); or attain much higher performance with a short service-life for the given power budget (DVSAM-Short and VSAM-Short).

Table 3.1 DVSAM Tool-Set

Name	Goal	Method
DVSAM-Pow	Consume minimum power for the same service-life	$Vdd << Vdd_{NOM}$ at the beginning of service-life $Vdd < Vdd_{NOM}$ at the end of service-life $P < P_{NOM}$ $f = f_{NOM}$
DVSAM-Perf	Attain maximum performance for the same service-life	$Vdd < Vdd_{NOM}$ at the beginning of service-life $Vdd > Vdd_{NOM}$ at the end of service-life $P > P_{NOM}$ $f > f_{NOM}$
DVSAM-Short	Attain higher performance at a shorter service-life	$Vdd >> Vdd_{NOM}$ throughout the service-life $P >> P_{NOM}$ $f >> f_{NOM}$
VSAM-Short	Attain higher performance at a shorter service-life (No Vdd changes)	$Vdd >> Vdd_{NOM}$ throughout the service-life $P >> P_{NOM}$ $f >> f_{NOM}$

If we impose a constant $Vdd = Vdd_{NOM}$ over the service-life, a core can potentially operate at a higher frequency than f_{NOM} at the early stages of its service-life (Figure 2.1). The timing guard-band is fully consumed only towards the end. Hence, imposing operation at Vdd_{NOM} to cycle at f_{NOM} over the service-life leads to (1) *excessive power consumption* and (2) *sub-optimal performance*.

Excessive power consumption: To satisfy a frequency target in an environment where Vth increases continuously, Vdd can be conservatively set to the supply voltage that

is required to meet the frequency target for the *maximum* value of Vth. Recall from Equation (2.1) that a higher value of Vdd is required for a higher value of Vth to achieve the same gate delay target, τ_{NG}. Operation at Vdd_{NOM} is only required at the end of the service-life to meet f_{NOM}, where the increase in Vth takes its maximum value (Vth_D) over the service-life. Due to $Vth < Vth_D$, a lower supply voltage than Vdd_{NOM} suffices to cycle at f_{NOM} before the end of the service-life. If, for any point in time over the service-life except the end, we impose operation at a higher supply voltage (Vdd_{NOM}) than would be necessary to cycle at f_{NOM}, we end up wasting power according to Equation (2.3).

Sub-optimal performance: Another way of interpreting the worst case margin as imposed by operation at Vdd_{NOM} and f_{NOM} is the following: Vdd_{NOM} affords cycling at a strictly higher frequency than f_{NOM} especially at early stages of the service-life, where Vth degradation remains negligible. Hence, if, for any point in time over the service-life except the end, we impose operation at a lower frequency (f_{NOM}) than can be afforded by Vdd_{NOM}, we end up wasting performance.

DVSAM-Pow addresses (1) by enforcing operation at the *minimum required supply voltage* to cycle at f_{NOM} over the entire service-life. DVSAM-Perf, on the other hand, addresses (2) by enforcing operation at the *minimum required supply voltage* to cycle at the *maximum feasible frequency*, f with $f > f_{NOM}$ over the entire service-life.

Specifics of DVSAM operation are given in Figure 3.1. Dotted lines correspond to the base-line behavior where no aging-management is imposed. Solid lines depict the evolution of Vdd over time (first row) to track the degradation in path delays (second row) for various DVSAM modes throughout the service-life. For each mode, $1/\tau_{OP} = f_{OP}$ specifies the operating frequency. Operating points corresponding to $\tau_{NG} = 1/f_{NG}$, $\tau_{NOM} = 1/f_{NOM}$, Vdd_{NOM} and S_{NOM} are explicitly marked for each plot, which correspond to the zero-guard-band period, worst-case-guard-band (nominal) period, nominal supply voltage and nominal service-life respectively.

Figure 3.1 (a) provides a closer look at DVSAM-Pow: Following an initial decrease to reduce the aging rate, we increase the supply voltage to track and compensate for the aging-induced increases in Vth throughout the service-life. We impose operation at f_{NOM} by making sure that, at any given time over the service-life, the $minimum$ Vdd to achieve f_{NOM} is provided. This results in critical path delays to remain constant at τ_{NOM} throughout the service-life ($\tau_{OP} = \tau_{NOM}$).

Due to the continuous increase in Vth, Vdd increases monotonically to satisfy the frequency target (Equation (2.1)). Under nominal operating conditions, when Vth anticipates its maximum at Vth_D, the *minimum supply voltage required* to cycle at f_{NOM} be-

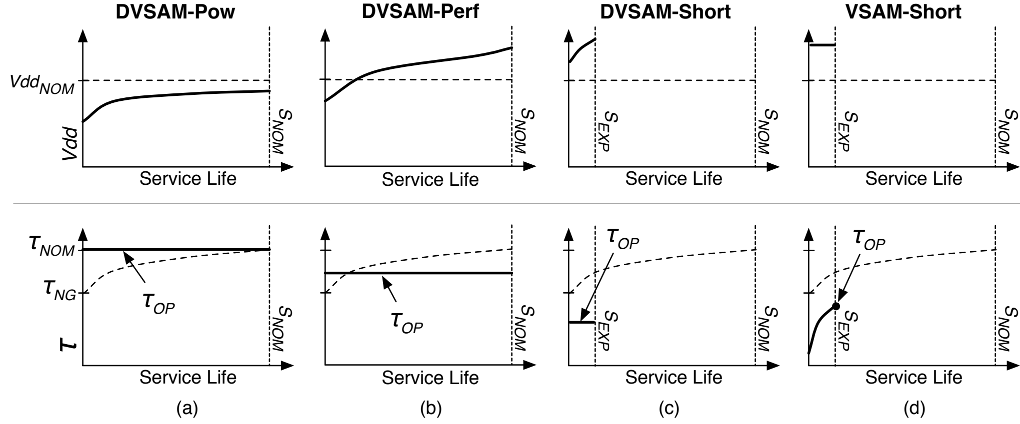


Figure 3.1 Impact of Dynamic Voltage Scaling for Aging-Management (DVSAM)

comes Vdd_{NOM}. However, due to the initial Vdd decrease as imposed by DVSAM-Pow, the aging rate decreases; consequently, Vth at the end of the service-life remains below Vth_D. Hence, the *minimum supply voltage required* to cycle at f_{NOM} under DVSAM-Pow becomes less than Vdd_{NOM} at the end of the service-life when the maximum increase in Vth is anticipated. In this manner, Vdd *always* remains below Vdd_{NOM}, which translates to power-savings according to Equation (2.3).

As depicted in Figure 3.1 (b), DVSAM-Perf shows a similar behavior when compared to DVSAM-Pow: Following an initial decrease to reduce the aging rate, we increase the supply voltage to track and compensate for the aging-induced increases in Vth throughout the service-life.

To cycle at the maximum feasible frequency f_{OP} with $f_{OP} > f_{NOM}$ over the service-life without consuming the slack provided by the worst-case guard-band excessively to result in timing errors, DVSAM-Perf operates at the minimum possible Vdd required at each point in time over the service-life. This results in critical path delays to remain constant at $\tau_{OP} < \tau_{NOM}$ throughout the service-life.

The key difference between DVSAM-Perf and DVSAM-Pow is that for DVSAM-Perf, Vdd remains below Vdd_{NOM} only shortly towards the beginning of the service-life and exceeds Vdd_{NOM} rapidly towards the end. This increases the power-consumption over the nominal, but only *slightly*: For the target frequency assumed, DVSAM-Perf ensures operation at the *minimum supply voltage* by closely tracking the aging-induced degradation in Vth. In other words, we achieve a given performance target by consuming *the minimum required power*.

Note that operation at *increased* f_{OP} ($> f_{NOM}$) and *increasing* Vdd does not lead to a faster than acceptable consumption of G, due to the slowdown achieved in aging rate by the initial Vdd decrease. This observation follows [11].

If more performance is demanded than achievable by DVSAM-Perf, we have options. We can impose operation under DVSAM-Short, which follows DVSAM-Perf semantics for a shorter service-life than the nominal, S_{EXP} (Figure 3.1 (c)). The impact of shorter service-life is strictly higher frequency of operation when compared to DVSAM-Perf. We still operate at the *minimum required supply voltage* to satisfy the frequency target. This results in critical path delays to remain constant at $\tau_{OP} < \tau_{NG}$ throughout the service-life. Since we expect frequencies $f_{OP} > f_{NG}$ for this mode, Vdd remains strictly higher than Vdd_{NOM} even at the initial stages of the service-life.

VSAM-Short represents an alternative to DVSAM-Perf. Instead of dynamically tracking increases in Vth and compensating by increases in Vdd, this mode enforces operation at a strictly higher Vdd than Vdd_{NOM} *constantly* over a shorter service-life than the nominal, S_{EXP} (Figure 3.1 (d)). In this manner, aging rate significantly increases as can be seen from the increased slope of the delay curve when compared to the base-line (Figure 3.1 (d)). However, significantly shorter service-life still permits operation at frequencies higher than the zero-guard-band frequency f_{NG}, corresponding to τ_{OP}. Note that the gate delays do not stay constant for VSAM-Short throughout the service-life, as opposed to other modes of DVSAM. Hence, f_{OP} for this case corresponds to the maximum gate delay observed throughout S_{EXP}, τ_{OP}.

CHAPTER 4

THE BUBBLEWRAP MANY-CORE

We now introduce the BubbleWrap many-core, an architecture that expends or *pops* otherwise unusable cores on the die to maximize performance on sequential sections. For a homogeneous many-core architecture, we show how a BubbleWrap chip applies core popping along with the aging management techniques of Chapter 3.

4.1 Overview of the BubbleWrap Many-Core

Given an N-core homogeneous many-core design where only N_T cores can be powered simultaneously, BubbleWrap distinguishes two groups of cores: *throughput cores* and *expendable cores*. Throughput cores are the the most power-efficient cores (as determined by the chip's process variation profile) and are used to run as many threads as possible under the power constraint during parallel phases. BubbleWrap puts the remaining $N_E = N - N_T$ cores to work as a "sequential accelerator" by running them at elevated voltage and frequency to speed up sequential phases. Such aggressive operation quickly ages or *pops* an Expendable core until it can no longer sustain high performance. At that point, it is simply replaced with another core from the Expendable group.

Figure 4.1 shows a logical overview of the BubbleWrap chip. The figure depicts a chip in mid-life when some of the Expendable cores (black) have already popped. Note that although the Throughput and Expendable cores form two logical groups, process variation determines their actual location on the die so that the cores of each group may not be physically contiguous. All cores in the Throughput group do, however, receive the same supply voltage Vdd_T. When active, Expendable cores run at Vdd_E.

To determine how quickly BubbleWrap can afford to pop Expendable cores, consider that the chip as a whole must last for some nominal service-life S_{NOM} (e.g., 7 years). We define the "sequential load" L_{SEQ} to be the average over time of the number of latency-sensitive sequential threads actively executing in the system. Each such latency-sensitive thread would fully occupy an expendable core throughout its execution time.

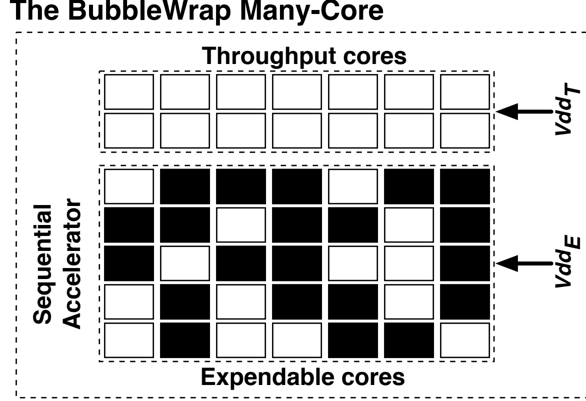


Figure 4.1 Logical organization of the BubbleWrap many-core

Then, the service-life of an individual Expendable core S_{EXP} is given by

$$S_{EXP} = \frac{S_{NOM} \times L_{SEQ}}{N_E} \tag{4.1}$$

For example, consider an BubbleWrap many-core with 32 Expendable cores that spends its entire lifetime concurrently executing two applications A_0 and A_1. Assume that A_0 has a sequential thread that runs 20% of the time ($L_{SEQ0} = 0.2$) and A_1 comprises a sequential thread that runs half of the time ($L_{SEQ1} = 0.5$). Then, the total sequential load becomes $L_{SEQ} = 0.7$. Given that $N_E = 32$ and assuming $S_{NOM} = 7$ years, Equation (4.1) gives $S_{EXP} \approx 0.15$ years, meaning that each expendable core has to last less than 1.8 months.

For core popping to be profitable requires that $S_{EXP} \ll S_{NOM}$ as in the preceding example. Figure 4.2 shows how shorter effective service-lives S_{EXP} permit increasingly higher frequencies. Fortunately, we expect that S_{EXP} will continue to shrink with time: First, technology scaling is providing more Expendable cores with each generation, increasing the denominator of Equation (4.1). Second, we expect the sequential load L_{SEQ} to remain small because applications designed to exploit many-cores will spend most of their time in parallel sections.

4.2 Example BubbleWrap Environments

The aging management techniques of Chapter 3, together with core popping, can combine in different ways. Figure 4.3 shows three example many-core environments with the level

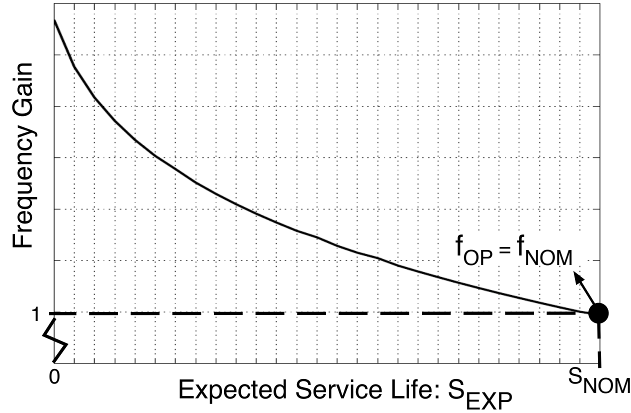


Figure 4.2 Frequency gain due to shorter service-life

of DVSAM involvement increasing from left to right. For each environment, the top row of the figure shows a version of the chip, called *NoPop*, that does not pop its Expendable cores. The configurations in the bottom row are identical except that they pop their Expendable cores to achieve sequential acceleration. Table 4.1 shows which DVSAM techniques the environments provide for each group of cores.

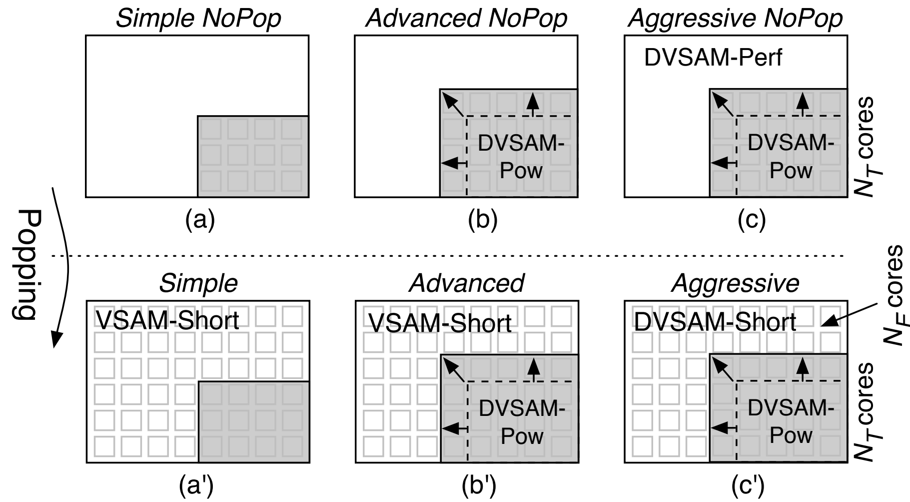


Figure 4.3 Example many-core environments before (top) and after (bottom) adding core popping (Throughput cores are highlighted in gray)

The *SimpleNoPop* environment (a) serves as a baseline where all Throughput cores operate at a constant $Vdd = Vdd_{NOM}$ at f_{NOM}. Environments (b – c) introduce the ability to scale the supply voltage dynamically over the service-life to avoid sub-optimal operation.

Table 4.1 DVSAM Support for Example Environments

Environment	$\mathbf{Vdd_E}$	$\mathbf{Vdd_T}$
Simple NoPop	NA (Cores disabled)	$= Vdd_{NOM}$ (constant)
Advanced NoPop	NA (Cores disabled)	$< Vdd_{NOM}$ (DVSAM-Pow)
Aggressive NoPop	Variable (DVSAM-Perf)	$< Vdd_{NOM}$ (DVSAM-Pow)
Simple	$> Vdd_{NOM}$ (VSAM-Short)	$= Vdd_{NOM}$ (constant)
Advanced	$> Vdd_{NOM}$ (VSAM-Short)	$< Vdd_{NOM}$ (DVSAM-Pow)
Aggressive	$> Vdd_{NOM}$ (DVSAM-Short)	$< Vdd_{NOM}$ (DVSAM-Pow)

With this capability, the *AdvancedNoPop* environment (b) applies DVSAM-Pow (Chapter 3). Power-savings on Throughput cores as achieved by DVSAM-Pow can be exploited in two different ways: (i) We can impose a closer to optimal operation in a power-critical environment, characterized by a lower-power consumption for the same level of performance. (ii) We can translate the power-savings on Throughput cores to *throughput* performance by expending the set of throughput cores. Due to DVSAM-Pow resulting in a per-core power consumption strictly less than the nominal, the number of cores that can be simultaneously powered for the same budget increases. Note that each core still cycles at f_{NOM}. In the following, we assume one such performance-critical scenario, favoring (ii).

Finally, *AggressiveNoPop* (c) adds DVSAM-Perf (Chapter 3) support for Expendable cores to achieve sequential acceleration, on top of DVSAM-Pow for throughput cores.

Designs (a' – c') augment the above environments with core popping to reach higher levels of sequential acceleration: Where environments (a' – b') make use of VSAM-Short on Expendable cores to this end, environment (c') uses DVSAM-Short.

4.3 Hardware Support for Aging Management

All BubbleWrap environments require modifications to the power grid and addition of a simple controller to implement the aging management. Additionally, some environments may require aging sensors. The following describes the specific requirements of each environment and enumerates some of the available design alternatives.

Controller: All of the aging management techniques, including core popping, require dynamic adjustment of the core configuration. For DVSAM-Pow and DVSAM-Perf, this

means gradually ramping the Expendable core voltages over time. For core popping, it means selecting the next core to pop, identifying when it has been expended, and keeping track of which cores are popped and which are not. Furthermore, the controller must be architecturally exposed so that the operating system can inform it about which threads require acceleration. A logical place to implement such a controller is in the System Management firmware.

Power Distribution System: All environments except *SimpleNoPop* and *AdvancedNoPop* require the ability to supply different voltages to the Expendable and Throughput cores. Since the physical location of the Throughput and Expendable cores is not known until manufacturing test time, we require the ability to dynamically supply either Vdd_E or Vdd_T to any core on the die. The most flexible solution is to include two independent supply networks [17] and connect each core to both grids through power-gating transistors. Then, each core can dynamically select either voltage by turning on the appropriate transistor.

A simpler design is possible if we impose the constraint that all Throughput cores must be idle when an Expendable core is active, which is reasonable if the system will run a single application (i.e., not multiprogrammed) workload with alternating parallel and sequential phases. In that case, it suffices to include only one supply grid and externally apply either Vdd_E or Vdd_T. All cores again include power-gating transistors so that Throughput cores disconnect from the grid during sequential phases and vice-versa.

Aging Measurement: The DVSAM-Pow and DVSAM-Perf aging management schemes included in all environments except *Simple* and *SimpleNoPop* require a way of determining the current aging state of a core in order to choose the minimum supply voltage that guarantees safe operation at a target frequency. One way to measure the aging state is to embed "aging sensors" in each core. Typically, these are implemented as test paths connected to ring oscillators [6], but other approaches are also possible [18].

CHAPTER 5

EVALUATION SETUP

We evaluate each of the environments of Section 4.2 in a near-future process technology using a synthetic workload to perform a limit study of the sequential, parallel, and overall performance gains of an BubbleWrap system.

5.1 Process Technology

We assume a 22 nm node based on the Predictive Technology Model's bulk HK+MG CMOS scaling [19] incorporating recent corrections [20]. Table 5.1 shows the basic technology parameters, and Table 5.2 shows the parameters that determine the aging rate (via Equation (2.2)). The constant of proportionality of Equation (2.2), A_{BTI}, is calibrated so that a core with $+3\sigma$ Vth variation (slow corner) slows down by $G = 10\%$ at the end of the service life S_{NOM} if operated continuously at (Vdd_{NOM}, T_{MAX}). The constant of proportionality for the alpha-power law (Equation (2.1)) is set to guarantee operation at f_{NG} for Vdd_{NOM} under the same $+3\sigma$ Vth conditions. Finally, the constant of proportionality for leakage current is set so that leakage accounts for 20% of the average power consumption for a core with -3σ Vth variation (the leaky corner) at Vdd_{NOM}.

Table 5.1 General Process Parameters

Parameter	Value
α	1.3
n	1.5
k_{DIBL}	-150 mV/V
k_T	-1 mV/K
T_{NOM}	85 °C
T_{MAX}	100 °C
Vdd	0.8 – 1.3 V ($Vdd_{NOM} = 1$ V)
Vth_{NOM}	250 mV

Table 5.2 Process Aging Parameters

Parameter	Value
tox	1.4 nm
Cox	2.5×10^{-20} F/nm^2
Eox	3.5×10^{-20} F/nm
E_0	0.08 V/nm
Ea	0.56 eV
a	0.2
S_{NOM}	7 years
G	10%

To capture the effects of process variation, we consider normally distributed core-to-core variation in Vth with a standard deviation $\sigma = 12.5$ mV. We then bound the impact of BubbleWrap on variation-afflicted cores by performing experiments on cores with [-3σ, $+3\sigma$] deviation from the nominal Vth.

5.2 Workload

We model a workload consisting of a single application with a parallel and a sequential section. The application is synthetic — designed to explore how BubbleWrap performs on workloads with different proportions of parallel and sequential work. Specifically, it is parameterized by the fraction q of time spent in the sequential section when executing in the *Simple* environment; the evaluation characterizes BubbleWrap for $q = [0, 1]$. The performance of the parallel section is assumed to scale ideally as more cores are added to the Throughput group, and the performance of the sequential section is assumed to scale ideally with frequency. These optimistic scaling assumptions mean that the following evaluation is a limit study presenting the *best possible* performance gains an BubbleWrap system can achieve.

Although the workload is synthetic, it is important to model realistic power and temperature behavior for each thread in order to accurately quantify aging. For this purpose, each "thread" of the workload is modeled by a SPECint2000 benchmark. Every non-idle core cycles through the SPECint benchmarks in round-robin fashion, switching every 45 minutes of simulated time so that each core experiences a diverse range of work over its lifetime.

5.3 Many-Core Microarchitecture

We model a near-future 32-core many-core microarchitecture as described in Table 5.3. Based on ITRS data [3], we predict that such a chip has 17 Throughput cores and 15 Expendable cores when run in a *Simple* (Section 4.2) environment. We simulate the core microarchitecture on a cycle-accurate simulator instrumented with Wattch [21]. The simplified temperature model (Equation (2.4)) assumes an ambient temperature T_A of 45 °C and a lumped device-to-ambient thermal resistance $\theta_{JA} = 5.22$ K/W.

Table 5.3 Microarchitecture Parameters

Technology: 22 nm	Zero-guard-band frequency f_{NG}: 5 GHz
Cores: 15 Expendable, 17 Throughput	Per-core max. power: 6.7 W
Width: 6-fetch 4-issue 4-retire OoO	L1 D Cache: 16 KB WT, 2 cyc round trip, 4 way, 64 B line
ROB: 152-entries	L1 I Cache: 16 KB, 2 cyc round trip, 2 way, 64 B line
Issue Window: 40 fp, 80 int	L2 Cache: 2 MB WB, 10 cyc round trip (at 5 GHz), 8 way, 64 B line, has stride prefetcher
LSQ Size: 54 LD, 46 ST	
Branch pred: 80 KB tournament	Memory: 80 ns round trip

5.4 BubbleWrap Controller

In keeping with the limit study theme, we assume perfect aging sensors (i.e., no measurement error) and an oracular controller (Section 4.3) for the aging management schemes. The controller knows *a priori* the future workload for each core in the system and exploits this information to make an ideal choice of operating frequency for DVSAM-Short, VSAM-Short, and DVSAM-Perf. Similarly, the ideal aging sensors allow DVSAM-Perf, DVSAM-Pow, and DVSAM-Short to operate at the minimum voltage at each instant. This idealized control system provides the maximum possible power savings and performance gains.

CHAPTER 6

EVALUATION

This chapter provides a limit study to assess benefits from BubbleWrap operation regarding power and performance, assuming an ideally *controllable* and *observable* system. We start with a quantitative analysis of DVSAM-Pow and DVSAM-Perf, follow with characterization of core popping using VSAM-Short and DVSAM-Short, and conclude the chapter with an overall performance analysis of different BubbleWrap environments.

6.1 Enhancing Throughput: DVSAM-Pow

Table 6.1 summarizes power and frequency benefits from DVSAM-Pow and DVSAM-Perf, respectively, on a per-core basis for cores with differing levels of Vth due to process variation. For DVSAM-Pow, the BubbleWrap controller's goal is to reduce power as much as possible at f_{NOM}. The second column of Table 6.1 shows that the energy savings range from 1.15x (for cores of $+3\sigma$ Vth) to 1.31x (for cores of -3σ Vth). The power savings are most pronounced for cores of -3σ variation, which suffer from excessive static power consumption. The lower operating voltages of DVSAM-Pow reduce both dynamic and static power (Equation (2.3)).

Table 6.1 Impact of DVSAM-Pow and DVSAM-Perf

Deviation in Vth	**DVSAM-Pow Energy Savings**	**DVSAM-Perf Frequency Gain**
-3σ	1.31x	1.22x
0	1.25x	1.18x
$+3\sigma$	1.15x	1.14x

Figures 6.1(a) – 6.1(d) show how the operating point of a core under DVSAM-Pow varies over the service-life for the nominal core (with no variation in Vth). Each plot shows two curves: *Base* represents a core running continuously at Vdd_{NOM}, where *DVSAM-Pow* corresponds to the same core under aging management. Note that the banded structure of the curves is due to temporal variation in the workload. As shown in Figure 6.1(a),

DVSAM-Pow keeps Vdd below $Vdd_{NOM} = 1$ V over the entire service-life, reaching a maximum of only 0.85 V at the end of the service-life. The trend of aging-induced Vth-degradation is given in Figure 6.1(b). Although Vth for DVSAM-Pow starts at a higher value (due to the DIBL impact of the reduced Vdd), it degrades less over the service-life. Most importantly, Figure 6.1(c) shows that the lower operating voltages of DVSAM-Pow save significant power compared to a core operating continuously at Vdd_{NOM}. Figure 6.1(d) shows a corresponding decrease in temperature due to lower operating power.[1]

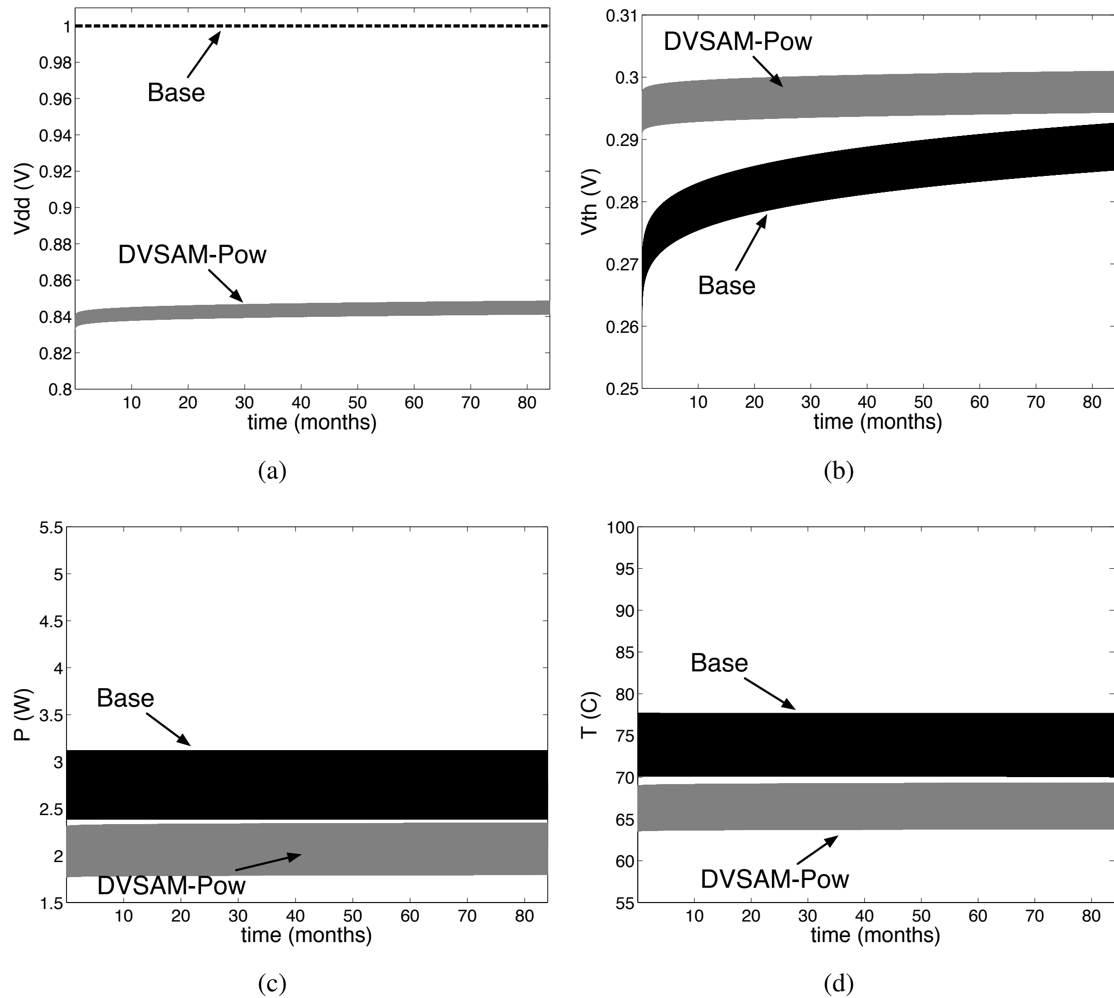


Figure 6.1 Evolution of DVSAM-Pow operating point over $S_{NOM} = 7$ years

[1]The relatively small range on the y-axis hides the increasing trend of power and temperature over the service-life for Figures 6.1(c) and 6.1(d).

6.2 Enhancing Frequency: DVSAM-Perf

The third column of Table 6.1 summarizes the per-core performance benefits from DVSAM-Perf. Recall that DVSAM-Perf is characterized by operation at the maximum frequency f_{MAX} for which S_{NOM} can be achieved by dynamically varying Vdd. According to the third column of Table 6.1, f_{MAX} ranges from $1.22 \times f_{NOM}$ (for cores of -3σ Vth variation) to $1.14 \times f_{NOM}$ (for cores of $+3\sigma$ Vth variation) after applying DVSAM-Perf. The improvement comes from operation at the minimum required voltage Vdd for the frequency target $f_{MAX} > f_{NOM}$, instead of imposing operation at Vdd_{NOM} to cycle at f_{NOM} throughout the service-life.

Figure 6.2 characterizes the response of the nominal core (with no variation in Vth) to DVSAM-Perf. Again, the plots show the trends for a normal core operating at a constant $Vdd = 1$ V (*Base*) for comparison. As shown in Figure 6.2(a), Vdd starts slightly below $Vdd_{NOM} = 1$ V at the beginning of the service-life and increases beyond Vdd_{NOM} thereafter. The supply voltage remains strictly greater than the supply voltage for DVSAM-Pow. Hence, due to exponential dependence of the aging-induced increase in Vth on the supply voltage (Equation (2.2)), the degradation in Vth in this case is larger than the degradation under DVSAM-Pow or *Base*. At the end of the service-life, we observe $Vth = 0.34$ V (Figure 6.2(b)) with $Vdd = 1.25$ V (Figure 6.2(a)) almost hitting the maximum supply voltage permitted by the technology (1.3 V). Due to stressed operation, per-core power consumption reaches 5.26 W at the end of the service-life (Figure 6.2(c)) — higher than a core running at a constant (f_{NOM}, Vdd_{NOM}). However, even at the end of life, power headroom is still available to DVSAM-Perf (the limit on maximum per-core power consumption is 6.7 W). Note that, by definition, DVSAM-Perf accommodates operation at the *minimum required supply voltage*, and therefore at the *minimum required power consumption*, for a given frequency target. The limiting factor is the maximum allowable junction temperature of 100 °C, which is just reached at the end of the service-life (Figure 6.2(d)).

DVSAM-Perf is able to increase f_{MAX} beyond f_{NG}, which might be counterintuitive. Operation above f_{NG} is possible because f_{NG} reflects the maximum frequency only if the chip spends its entire service-life at the maximum allowable temperature $T_{MAX} = 100$ °C. When running a typical workload like ours, the processor spends most of its time at much lower temperatures (Figure 6.2(d)). The oracular BubbleWrap controller is able to exploit this temperature and aging headroom to deliver additional performance.

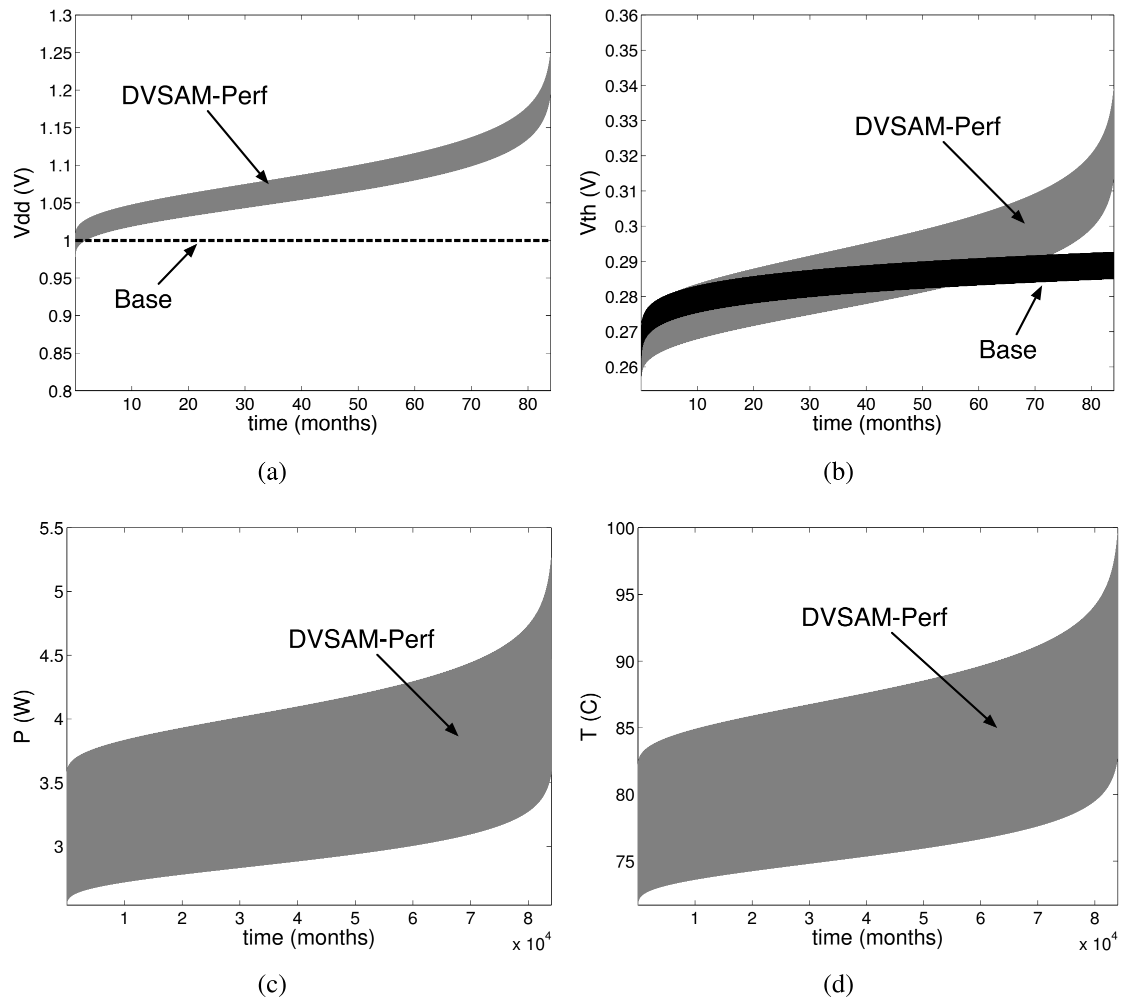


Figure 6.2 Evolution of DVSAM-Perf operating point over $S_{NOM} = 7$ years

6.3 Popping Cores: VSAM-Short, DVSAM-Short

An BubbleWrap many-core can pop Expendable cores at stressed operating conditions to extract the maximum possible performance. Figure 6.3 provides characterization for shorter service-life operation at a typical temperature $T_{NOM} = 85$ °C and constant voltage Vdd_{POP} for VSAM-Short. The decrease in service-life as imposed by operation at elevated supply voltages beyond the nominal is given in Figure 6.3(a).[2] Assume an aggressive scenario where each Expendable core is expected to operate for half a month. The corresponding point in the design space is depicted as (Vdd_{POP}, S_{POP}) in Figure 6.3(a), with $Vdd_{POP} \approx 1.24$ V. Imposing operation at Vdd_{POP} over the service-life translates to a frequency-gain of $\approx 1.24\times$ (Figure 6.3(b)) along with a power cost of $1.9\times$ (Figure 6.3(c)).

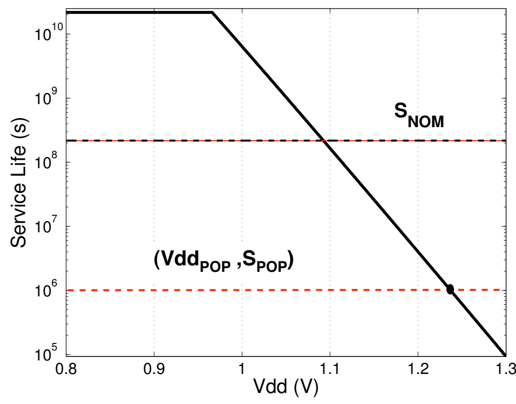
DVSAM-Short represents another, more aggressive approach for core-popping. We observe that operation at ≈ 0.5 months leads to an additional 5% increase in frequency over DVSAM-Perf in this case, at a power cost of $1.1\times$ when compared to DVSAM-Perf. Note that DVSAM-Short represents a DVSAM-Perf equivalent covering a shorter service-life; both schemes follow the same semantics, with service-life being the sole distinguishing parameter.

6.4 Performance Analysis

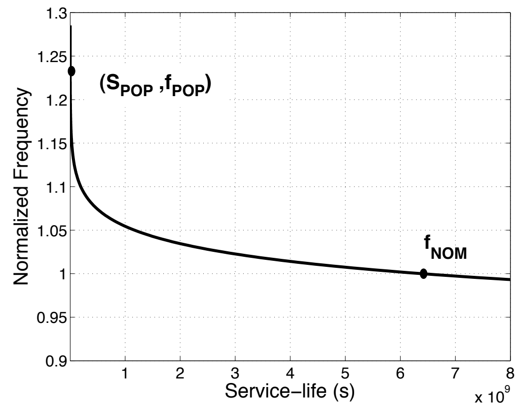
Performance benefits from BubbleWrap operation stem from (1) sequential acceleration by popping Expendable cores; and (2) throughput boost by growing the set of Throughput cores. Performance benefits achievable by popping cores are determined by the service-life per Expendable core, S_{EXP} (Equation (4.1)), which is a function of the sequential load, L_{SEQ}. The smaller L_{SEQ}, the shorter S_{EXP} and the larger the frequency gain. For the synthetic workload introduced in Chapter 5, which spends a fraction q of its execution time in sequential phases before BubbleWrap optimization, $L_{SEQ} \approx q$.

Frequency gain on sequential sections is depicted in Figure 6.4. The x-axis sweeps the load condition q from 100% (a fully serial application) to 0% (a fully parallel application). All frequencies are normalized to the nominal frequency, f_{NOM}. For environments that do not provide sequential acceleration, the frequency of any core remains equal to f_{NOM}. This applies to the baseline (*SimpleNoPop*) and *AdvancedNoPop* for any load condition.

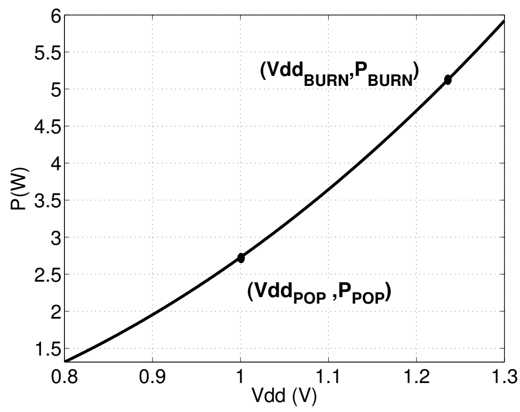
[2]Note that operation at $Vdd_{NOM} = 1$ V does not correspond to $S_{NOM} = 7$ years for nominal temperature conditions; that would only be the case if the core were at $T_{MAX} = 100$ °C.



(a)



(b)



(c)

Figure 6.3 Impact of Popping cores: VSAM-Short

Any environment where Expendable cores are popped gets a higher frequency boost with lower values of q because, in accord with Equation (4.1), the lower sequential load translates to shorter service-life per Expendable core. This trend is observed for *Simple*, *Advanced* and *Aggressive*. The most powerful scheme, *Aggressive*, always performs best, but its relative advantage becomes more pronounced for larger values of q.

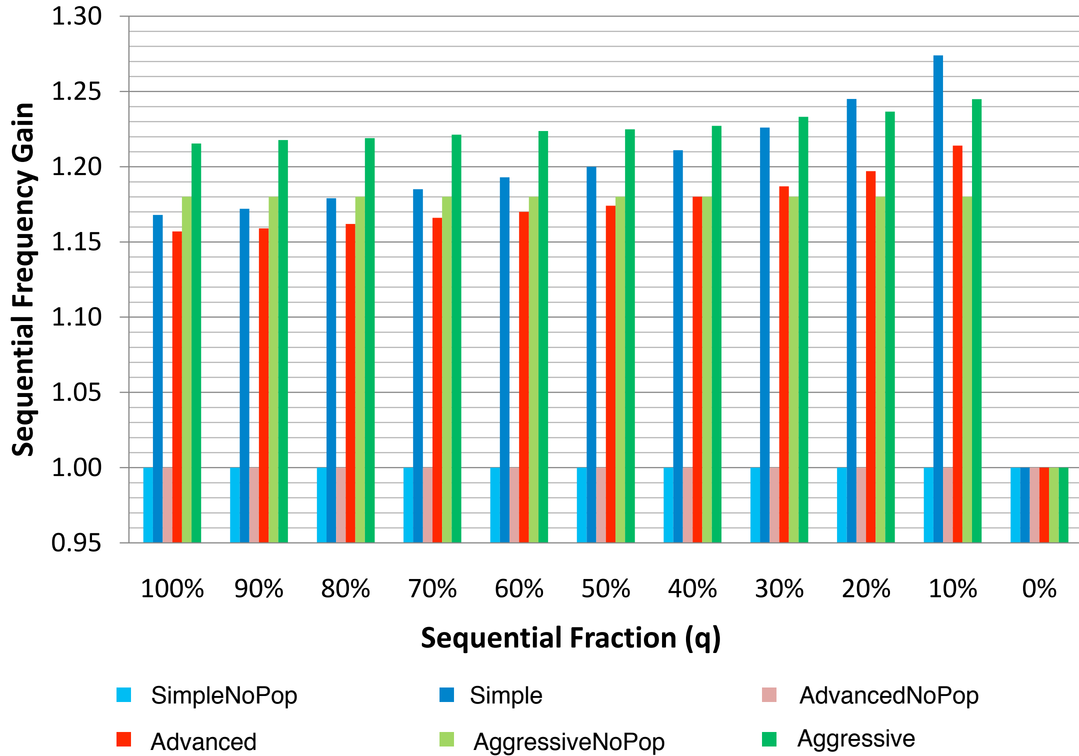


Figure 6.4 Sequential frequency gain normalized to *SimpleNoPop*

Simple and *Advanced* deploy the same mechanism to pop cores: Operate at a strictly higher supply voltage than the nominal throughout the service-life (as imposed by VSAM-Short). However, *Advanced* provides fewer Expendable cores than *Simple* due to expansion of the set of Throughput cores. This results in *Simple* achieving more frequency boost on sequential sections when compared to *Advanced*. Finally, *AggressiveNoPop* imposes operation on Expendable cores at the frequency as set by DVSAM-Perf. Since no popping takes place, the frequency gain on sequential sections is independent of q.

Recall that we assume a base-line system of 32 cores with 17 cores assigned as Throughput cores (Table 5.3). Only environments supporting DVSAM-Pow are able to increase the number of Throughput cores.

For these environments, the additional number of Throughput cores is proportional to the power savings. For *Advanced*, *AdvancedNoPop*, *Aggressive* and *AgressiveNoPop*, the $1.33\times$ nominal power savings from Table 6.1 enables $17 \times 1.33 \approx 22$ Throughput cores.

The overall performance benefit per environment is characterized in Figure 6.5, which combines the data from Figures 6.4 and the expansion in the set of throughput cores to arrive at the overall application speedup. All speedups are normalized to *SimpleNoPop*. With decreasing q, *Simple* converges to *SimpleNoPop* because the performance becomes less and less sensitive to sequential. Since *Simple* does not provide any throughput benefits, the overall performance benefit comes only from sequential acceleration and decreases with q. On the other extreme, we have *AdvancedNoPop* providing throughput boost only. In this environment the performance benefits increase with decreasing q. All environments except *Simple* and *SimpleNoPop* provide throughput benefits by expanding the set of throughput cores as enabled by DVSAM-Pow. Generally, we observe increasing overall performance boost with decreasing q for these cases.

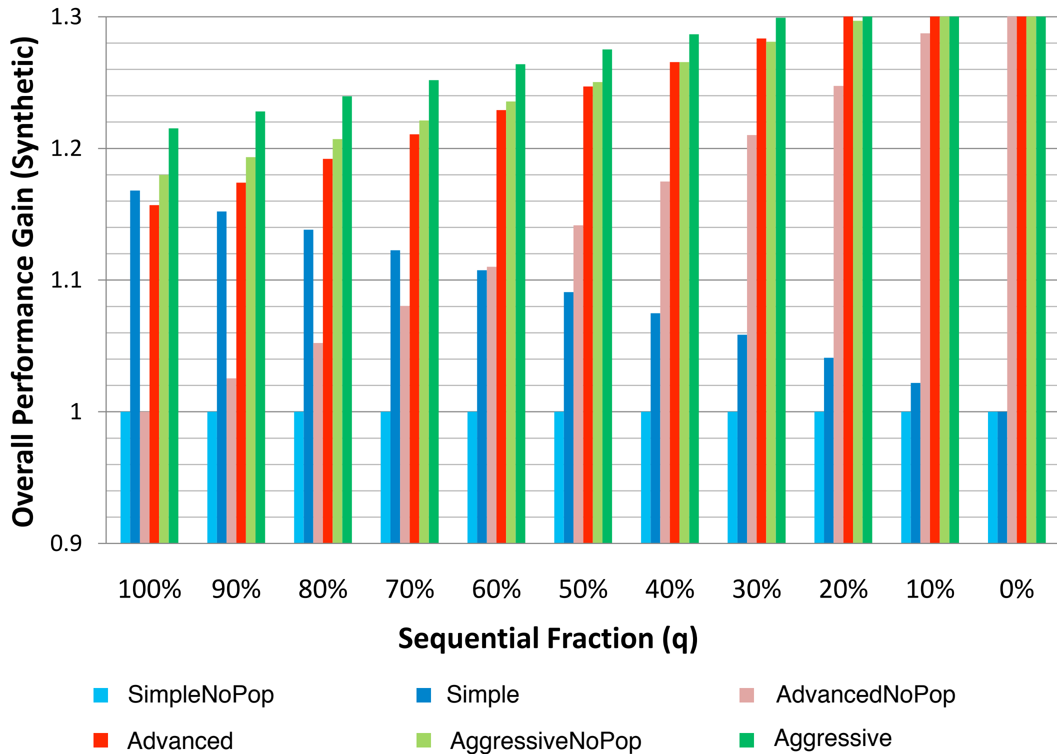


Figure 6.5 Overall performance gain for different BubbleWrap environments

Simple and *Advanced* provide the same mechanism for sequential acceleration, while *Advanced* is enhanced by throughput boost in addition. For higher values of q where the performance is more sensitive to sequential acceleration, *Simple* provides more boost since *Advanced* has fewer Expendable cores due to the expansion in the set of Throughput cores. With decreasing q, the impact of sequential acceleration decreases, and with the throughput boost provided, *Advanced* performs significantly better.

The difference between *Aggressive* and *AggressiveNoPop* shows that even for the most aggressive architecture without core popping support, adding core popping yields an additional 5% frequency increase on fully sequential codes. Moreover, this increase comes essentially for free since *AggressiveNoPop* already includes the power grid, aging sensor, and controller support required for core popping.

Overall, *Aggressive* provides the best performance, closely followed by *AggressiveNoPop* and *Advanced*. *Aggressive* and *AggressiveNoPop* represent the most complex schemes, demanding DVS aging management support on both Throughput and Expendable cores. From a performance-complexity perspective, *Advanced* turns out to be the most feasible scheme.

CHAPTER 7

RELATED WORK

The impending multi-core power wall is well-known in industry and reflected in recent ITRS projections [3]. However, our suggestion to expend or "pop" the excess cores that cannot be powered is novel as far as we know. To avoid expending cores, others have proposed extremely low-voltage designs that allow all cores to operate simultaneously, albeit at severely reduced frequency [22, 23]. Others have developed aging-aware designs that reduce the guard-band required for NBTI, resulting in power and performance gains [5]. Another alternative is a system-on-chip style design comprising power-efficient heterogeneous application-specific accelerators [24]. BubbleWrap is unique in extending the scaling of homogeneous CMPs without requiring core modifications.

BubbleWrap's use of dynamic voltage scaling and shorter service-life to improve performance is not entirely new. Facelift [11] proposed applying two discrete voltage levels to minimize aging and showed how shorter service lives could be exploited to increase core frequency. BubbleWrap's DVSAM framework improves on these techniques with continuous voltage scaling and the power saving DVSAM-Pow mode. Additionally, it describes a set of novel architectures with Expendable cores where the short-lifetime optimizations are especially effective. Several other authors have considered processor slowdown with time [6, 24, 25]. They are interested in designing circuits that detect when a critical path has slowed down. These "aging sensors" are a part of some BubbleWrap environments, which use them to support the novel DVSAM framework.

CHAPTER 8
CONCLUSION AND FUTURE WORK

We are already facing the many-core power wall, where the gap between the number of cores that fit on a die and the number that can operate simultaneously under the power budget is rapidly increasing with technology scaling. To push back the many-core power wall, this thesis made two main contributions.

First, it introduced *Dynamic Voltage Scaling for Aging Management* (DVSAM) — a new scheme for trading off processor aging for performance and power. DVSAM can be used with one of the following goals: consume less power at the same performance and service life; attain higher performance at the same service life and only slightly higher power; or attain much higher performance with a short service life and higher power.

Second, this work presented the *BubbleWrap* many-core, an architecture that makes use of DVSAM. BubbleWrap identifies the most power-efficient set of cores on a variation-affected die — the largest set that can be simultaneously powered up — and designates them as *Throughput* cores dedicated to parallel section execution; the rest of the cores (*Expendable* cores) are dedicated to sequential sections. BubbleWrap applies DVSAM in several environments. In one of them, BubbleWrap sacrifices Expendable cores one at a time by running them at elevated Vdd for a month or so each, until they completely wear out.

Our simulations showed that a 32-core BubbleWrap many-core provides substantial improvements over a plain chip with the same number of cores and the same power envelope. For example, on average, our most aggressive BubbleWrap design ran fully-sequential applications at a 22% higher frequency, and fully-parallel applications with a 33% higher throughput. We are now extending DVSAM to also include changes in processor frequency with time. This improvement should deliver better performance/power design points.

REFERENCES

[1] R. Dennard, F. Gaensslen, V. Rideout, E. Bassous, and A. LeBlanc, “Design of ion-implanted MOSFET’s with very small physical dimensions,” *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, October 1974.

[2] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, “Scaling, power, and the future of CMOS,” in *IEEE International Electron Devices Meeting*, December 2005, pp. 7–15.

[3] ITRS, “International Technology Roadmap for Semiconductors, 2008 Update,” 2008. [Online]. Available: http://www.itrs.net/Links/2008ITRS/Home2008.htm.

[4] K. Bernstein et al., “High-performance CMOS variability in the 65-nm regime and beyond,” *IBM Journal of Research and Development*, vol. 50, no. 4/5, pp. 433–449, July/September 2006.

[5] J. Abella, X. Vera, and A. Gonzalez, “Penelope: The NBTI-aware processor,” in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, 2007, pp. 85–96.

[6] M. Agarwal, B. Paul, M. Zhang, and S. Mitra, “Circuit failure prediction and its application to transistor aging,” in *25th IEEE VLSI Test Symposium*, May 2007, pp. 277–286.

[7] J. Blome, S. Feng, S. Gupta, and S. Mahlke, “Self-calibrating online wearout detection,” in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, 2007, pp. 109–122.

[8] J. Shin, V. Zyuban, P. Bose, and T. M. Pinkston, “A proactive wearout recovery approach for exploiting microarchitectural redundancy to extend cache SRAM lifetime,” in *Proceedings of the 35th International Symposium on Computer Architecture*, 2008, pp. 353–362.

[9] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, “The case for lifetime reliability-aware microprocessors,” in *Proceedings of the 31st Annual International Symposium on Computer Architecture*, 2004, pp. 276–287.

[10] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, “Exploiting structural duplication for lifetime reliability enhancement,” in *Proceedings of the 32nd Annual International Symposium on Computer Architecture*, 2005, pp. 520–531.

[11] A. Tiwari and J. Torrellas, “Facelift: Hiding and slowing down aging in multicores,” in *Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture*, 2008, pp. 129–140.

[12] W. Wang, V. Reddy, A. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, “Compact modeling and simulation of circuit reliability for 65-nm CMOS technology,” *IEEE Transactions on Device and Materials Reliability*, vol. 7, no. 4, pp. 509–517, December 2007.

[13] T. Sakurai and A. Newton, “Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas,” *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, April 1990.

[14] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, Inc., 2003.

[15] S. Pae et al., “BTI reliability of 45 nm high-K + metal-gate process technology,” in *Proceedings of the IEEE International Reliability Physics Symposium*, 2008, pp. 352–357.

[16] A. Montree et al., “Limitations to adaptive back bias approach for standby power reduction in deep sub-micron CMOS,” in *Proceedings of the 29th European Solid-State Device Research Conference*, September 1999, pp. 580–583.

[17] M. Popovich, A. V. Mezhiba, and E. G. Friedman, “On-chip power distribution grids with multiple supply voltages,” in *Power Distribution Networks with On-Chip Decoupling Capacitors*. New York, NY: Springer Publishing Company, Incorporated, 2007, pp. 323–359.

[18] T.-H. Kim, R. Persaud, and C. Kim, “Silicon odometer: An on-chip reliability monitor for measuring frequency degradation of digital circuits,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 874–880, April 2008.

[19] W. Zhao and Y. Cao, “New generation of predictive technology model for sub-45nm design exploration,” in *Proceedings of the 7th International Symposium on Quality Electronic Design*, 2006, pp. 585–590.

[20] A. Khakifirooz and D. Antoniadis, “MOSFET performance scaling – part II: Future directions,” *IEEE Transactions on Electron Devices*, vol. 55, no. 6, pp. 1401–1408, June 2008.

[21] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A framework for architectural-level power analysis and optimizations,” *SIGARCH Computer Architecture News*, vol. 28, no. 2, pp. 83–94, 2000.

[22] B. Zhai, R. G. Dreslinski, D. Blaauw, T. Mudge, and D. Sylvester, “Energy efficient near-threshold chip multi-processing,” in *Proceedings of the 2007 International Symposium on Low Power Electronics and Design*, 2007, pp. 32–37.

[23] R. Gonzalez, B. Gordon, and M. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, pp. 1210–1216, August 1997.

[24] K. Fan, M. Kudlur, G. Dasika, and S. Mahlke, "Bridging the computation gap between programmable processors and hardwired accelerators," in *IEEE 15th International Symposium on High Performance Computer Architecture*, February 2009, pp. 313–322.

[25] J. Smolens, B. T. Gold, J. C. Hoe, B. Falsafi, and K. Mai, "Detecting emerging wearout faults," in *The Third IEEE Workshop on Silicon Errors in Logic - System Effects (SELSE-3)*, April 2007.