The BubbleWrap Many-Core: Popping Cores for Sequential Acceleration

Ulya R. Karpuzcu, Brian Greskamp, and Josep Torrellas
University of Illinois at Urbana-Champaign
rkarpu2, greskamp, torrella@illinois.edu
http://iacoma.cs.uiuc.edu

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, many of the cores may have to be dormant at any given time to meet the power budget.

To push back the many-core power wall, this paper proposes *Dynamic Voltage Scaling for Aging Management (DVSAM)* — a new scheme for managing processor aging to attain higher performance or lower power consumption. In addition, this paper introduces the *BubbleWrap* many-core, a novel architecture that makes extensive use of DVSAM. BubbleWrap identifies the most power-efficient set of cores in a variation-affected chip — the largest set that can be simultaneously powered-on — and designates them as *Throughput* cores dedicated to parallel-section execution. The rest of the cores are designated as *Expendable* and are dedicated to accelerating sequential sections. BubbleWrap attains maximum sequential acceleration by sacrificing Expendable cores one at a time, running them at elevated supply voltage for a significantly shorter service life each, until they completely wear-out and are discarded — figuratively, as if popping bubbles in bubble wrap that protects Throughput cores. In simulated 32-core chips, BubbleWrap provides substantial improvements over a plain chip. For example, on average, one design runs fully-sequential applications at a 16% higher frequency, and fully-parallel ones with a 30% higher throughput.

Categories and Subject Descriptors

B.8.1 [**Hardware**]: Performance and Reliability. Reliability, Testing and Fault Tolerance.

General Terms

Design, Performance, Reliability.

Keywords

Processor Aging, Voltage Scaling, Process Scaling, Power Wall.

1. INTRODUCTION

Ideal CMOS device scaling [9] relies on scaling voltages down with lithographic dimensions at every technology generation — giving rise to faster circuits due to increased frequency and smaller silicon area for the same functionality. In this model, the dynamic power per unit area stays constant. This is because the energy per switching event decreases enough to compensate for the increased energy due to having more devices in the same area and switching them faster.

In recent generations, however, to keep leakage current under control, the decrease in the transistor's threshold voltage (Vth) has stopped. This, in turn, has prevented the supply voltage (Vdd) from scaling [13, 22]. Given the strong dependence of the dynamic energy on supply voltage, the net result is that the compensation effect explained above does not exist any more. As more transistors are integrated on a fixed-sized chip at every generation, the chip power increases rapidly. Chip power does not scale anymore.

If we fix the chip power budget due to system cooling constraints and the associated costs, we easily realize that there is a growing gap between what can be placed on a chip and what can be powered-on simultaneously. For example, Figure 1 shows data computed from the ITRS 2008 update [16] assuming Intel Core i7-like [15] cores and a constant 100W chip power budget. The figure compares the number of cores that can be placed on a chip at a given year (normalized to year 2008 numbers) and the number of those that can be powered-on simultaneously. The growing gap between the two curves shows the *Many-Core Power Wall*. Soon, many cores may have to remain powered-off.

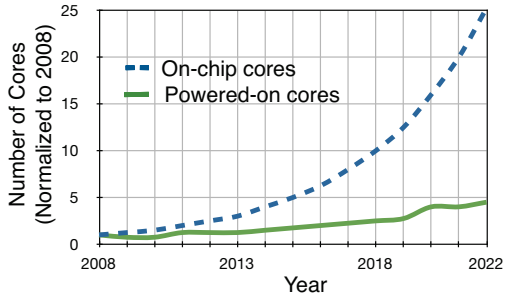

Figure 1: The many-core power wall, based on data from ITRS projections.

Another key effect of aggressive scaling is increasing parameter variation [6]. Variation can manifest as static, spatial variations across the chip, or dynamic, temporal variations as a processor is used. A significant contributor to the latter is device wearout or aging. Aging induces a progressive slowdown in the logic as it is being used.

Recently, processor aging has been the subject of much work (e.g., [1, 2, 7, 27, 28, 29, 30, 34]). It is well accepted that the aging rate is highly impacted by Vdd and temperature (T), where higher values increase the aging rate. Consequently, approaches to slow-down aging by operating at lower Vdd or T have been introduced [34]. We observe that such approaches to change the aging rate typically affect performance and power. Consequently, they could help push back the many-core power wall.

Based on this observation, this paper proposes a novel scheme for managing processor aging that attains higher performance or

MICRO'09, December 12–16, 2009, New York, NY, USA.

$$\Delta Vth_{BTI} = \Delta Vth_{STRESS} \times (1 - \sqrt{\eta \times \frac{t_{RECOVERY}}{t_{RECOVERY} + t_{STRESS}}}) \quad \text{, where}$$
$$\Delta Vth_{STRESS} = A_{BTI} \times \left[\frac{q^3}{Cox^2} \times (Vdd - Vth_{NOM}) \times exp\left(-\frac{Ea}{2kT} + \frac{Vdd - Vth_{NOM}}{tox \times 0.5Eo}\right)\right]^{2a} \times (t_{STRESS})^a \quad (1)$$

lower power consumption. We call our scheme *Dynamic Voltage Scaling for Aging Management (DVSAM)*. The idea is to continuously tune Vdd (but not the frequency), exploiting any currently-left aging guard-band. The goal can be one of the following: consume the least power for the same performance and processor service life; attain the highest performance for the same service life and within power constraints; or attain even higher performance for a shorter service life and within power constraints.

We also propose *BubbleWrap*, a novel many-core architecture that makes extensive use of DVSAM to push back the many-core power wall. BubbleWrap identifies the most power-efficient set of cores in a variation-affected die — the largest set that can be simultaneously powered-on. It designates them as *Throughput* cores dedicated to parallel-section execution. The rest of the cores are designated as *Expendable* and are dedicated to accelerating sequential sections. BubbleWrap attains maximum sequential acceleration by sacrificing Expendable cores one at a time, running them at elevated Vdd for a significantly shorter service life each, until they completely wear-out and are discarded — figuratively, as if popping bubbles in bubble wrap that protects Throughput cores.

In simulated 32-core chips, BubbleWrap provides substantial gains over a plain chip. For example, on average, one design runs fully-sequential applications at a 16% higher frequency, and fully-parallel ones with a 30% higher throughput.

Overall, this paper makes two main contributions:

- We introduce *Dynamic Voltage Scaling for Aging Management (DVSAM)*, a new scheme for managing processor aging to attain higher performance or lower power consumption.
- We present the *BubbleWrap* many-core, a novel architecture that makes use of DVSAM to push back the many-core power wall.

The rest of the paper is organized as follows. Section 2 provides a background; Section 3 introduces DVSAM; Section 4 presents the BubbleWrap many-core; Sections 5 and 6 evaluate BubbleWrap and Section 7 discusses related work.

2. BACKGROUND

2.1 Modeling Aging

Our analysis focuses on aging induced by Bias Temperature Instability (BTI), which causes transistors to become slower in the course of their normal use. BTI-induced degradation leads to increases in the threshold voltage (Vth) of the form $\Delta Vth_{BTI} \propto t^a$, where t is time and a is a time-slope constant. Constant a is strongly related to process characteristics, and generally takes a value between 0 and 0.5 for recent process generations [3, 4, 35].

To model ΔVth_{BTI}, we adopt the framework of Wang *et al* [35]. Vth only increases when the voltage between the gate and the source of the transistor is set to a given logic value, and decreases more slowly when the voltage is set to the opposite logic value. These conditions are called Stress and Recovery conditions, respectively. Equation 1 above shows ΔVth_{BTI} as a function of the time the transistor is under stress (t_{STRESS}) and recovery ($t_{RECOVERY}$). In the equation, A_{BTI}, a, Eo and η are model fitting parameters. Importantly, the equation shows that ΔVth_{BTI} depends exponentially on the supply voltage (Vdd) and the temperature (T). Therefore, high values of Vdd or T will substantially increase the aging rate.

With this effect, Vth at a given Vdd and T is given by Equation 2, where Vth_{NOM}, Vdd_{NOM}, and T_{NOM} are the nominal values of these parameters, and k_{DIBL} and k_T are constants.

$$\begin{aligned} Vth =& Vth_{NOM} + k_{DIBL} \times (Vdd - Vdd_{NOM}) \\ &+ k_T \times (T - T_{NOM}) + \Delta Vth_{BTI} \end{aligned} \quad (2)$$

The increase in Vth translates into an increase in transistor switching delay (τ) as per the alpha-power law [26] (Equation 3). The result is a slowdown in the processor's critical paths and, hence, its operating frequency. In the formula, $\alpha > 1$ and $\mu \propto T^{-1.5}$.

$$\tau \propto \frac{Vdd}{\mu(Vdd - Vth)^\alpha} \quad (3)$$

The aging model from Equation 1 was derived for devices with silicon-based dielectrics and poly gates (*Poly+SiON* devices), and verified against an industrial 65nm node [35]. To reduce gate leakage, starting from 45nm, manufacturers have introduced a new generation of devices with higher gate dielectric constants (high-k) and metal gates (*HK+MG* devices) [3, 4]. However, we argue that the model is still applicable.

To see why, we refer to a recent reliability characterization of Intel's 45nm node [5], one of the most up-to-date HK+MG processes in production. The authors report that (1) PMOS Negative BTI (NBTI) remains a major reliability concern, like it was in the predecessor Poly+SiON 65nm node, and that (2) at higher electric fields (as induced by higher Vdd), NMOS Positive BTI (PBTI) becomes significant. Further, they demonstrate that the BTI characteristics closely follow the BTI behavior of Poly+SiON devices. Specifically, the same physical phenomena cause this behavior [23].

We modify the process parameters and time-slope characteristics of the base model to reflect the new technology node in light of the observations from [23]. In this paper, we assume cores like the Intel Core i7 [15], which are based on Intel's 45nm HK+MG technology. Note also that another finding from [5] is that the Hot-Carrier Injection (HCI) effect is not that significant for any realistic use condition. Hence, we focus on BTI-based aging only.

We acknowledge, however, that advances in devices in future nodes may require reconsidering the formulae. In the contemporary era of CMOS scaling, as we face more scaling limits, the pace at which new materials and features are introduced will increase [21, 22], as demonstrated by the introduction of HK+MG devices.

2.2 Impact of Aging

Equation 1 shows that ΔVth_{BTI} follows a power law with time. Since a typical value of the time slope a is between 0 and 0.5, ΔVth_{BTI} increases rapidly first and then more slowly. Let us assume a fixed ratio of stress to recovery time, and that stress and recovery periods are finely interleaved. If Vth_{NOM} is the value of Vth at the beginning of the service life, the curve $Vth = Vth_{NOM} + \Delta Vth_{BTI}$ as a function of time is shown in Figure 2(a). At the end of the service life, which the figure assumes is 7 years, Vth has reached Vth_D.

The switching delay of a transistor is given by Equation 3. As Vth increases with time due to aging, so does the switching delay τ for the same supply voltage and temperature conditions. Both PMOS and NMOS transistors suffer from BTI-induced aging — called NBTI and PBTI, respectively [5].

To determine the delay of a logic path in a processor, we follow the approach of Tiwari and Torrellas [34]. Specifically, when the

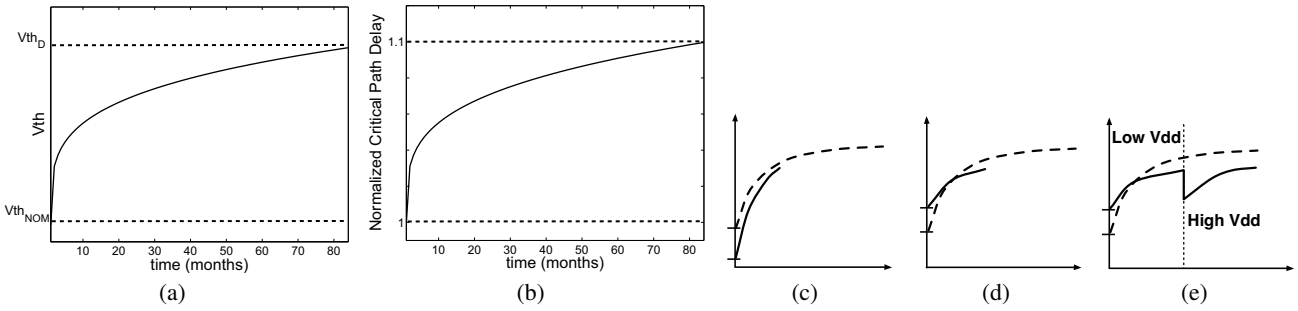

Figure 2: Effects of aging on Vth degradation (a) and on critical path delay degradation (b)-(e).

path is activated, we identify the set of transistors that switch. The path delay is given by the switching delays of such transistors plus the wire delays. As individual transistors age and their ΔVth_{BTI} increases following a power law, the total path delay can be shown to also increase following a similar curve. Specifically, the path delay increases fast at the beginning of the service life and then increases progressively more slowly.

The actual path delay increase depends on many issues, including the path composition in terms of PMOS or NMOS transistors, the amount of wire, the ratio of stress to recovery periods, the T, and the Vdd. However, the increase follows the general shape described. If we assume that the delay of the critical path of the processor increases 10% in a 7-year service life, we attain a normalized curve like the one in Figure 2(b). In the figure, the normalized critical path delay is 1 at the beginning, and 1.1 at the end of the service life.

Let us call the delay of the processor's critical path at the beginning of service τ_{ZG} (where ZG stands for zero guard-band). The same path will have a delay of $\tau_{NOM} = \tau_{ZG} \times (1 + G)$ at the end of the service life, where G is the timing guard-band that the path will consume during its life (10% in our example). Consequently, the processor cannot be clocked at a frequency $f_{ZG} = 1/\tau_{ZG}$ because it would soon suffer timing errors. It is clocked at the lower frequency f_{NOM} during all its service life:

$$f_{NOM} = \frac{1}{\tau_{NOM}} = \frac{1}{\tau_{ZG} \times (1 + G)} = \frac{f_{ZG}}{1 + G} \qquad (4)$$

2.3 Slowing Down Aging

In recent work called Facelift [34], Tiwari and Torrellas attempt to slow down aging by perturbing the curve in Figure 2(b). A major knob they use is to change Vdd. Vdd affects transistor aging as per Equation 1 and transistor delay as per Equation 3. Specifically, if we increase Vdd (without changing any other parameter), transistors become faster (from Equation 3, since α>1) and also age faster (from Equation 1). If, instead, we decrease Vdd, transistors age more slowly but they also become slower.

They then use timely Vdd changes to slow down aging. Graphically, this means forcing the curve in Figure 2(b) to reach a lower Y coordinate value at the end of the service life. To see the impact of timely Vdd changes, we repeat Figure 2(b) in Figures 2(c) and 2(d). Figure 2(c) shows the effect of increasing Vdd: it pushes the curve down (faster critical paths) but increases the slope of the curve (faster aging). Figure 2(d) shows the effect of decreasing Vdd: it pushes the curve up (slower critical paths) but reduces the slope of the curve (slower aging).

They make the observation that changing Vdd impacts (i) the aging rate and (ii) the critical path delay differently within the course of the processor service life. Specifically, it impacts the aging rate strongly (positively or negatively) at the beginning of the service life, and little toward the end. In contrast, it impacts the delay more uniformly across time. Consequently, they propose to *apply a low* Vdd toward the beginning of the service life. At that time, it reduces the aging rate the most and, therefore, slows down aging the most. Moreover, there is still substantial guard-band available to tolerate the lengthening of the critical path delay. They propose to *apply a high* Vdd toward the end of the service life. At that time, it still speeds up the critical path delay, while it increases the aging rate the least. The result of using this strategy is shown in Figure 2(e). At the end of the service life, the critical path delay (and therefore the wearout of the processor) is lower. Consequently, the authors can run the processor at a constant frequency throughout the service life that is higher than f_{NOM}.

Tiwari and Torrellas [34] also use this strategy to configure cores for a shorter service life. They consolidate all the aging into the shorter life and run the processor at an even higher, constant frequency throughout the short life.

3. DVSAM: DYNAMIC VOLTAGE SCALING FOR AGING MANAGEMENT

3.1 Main Idea

While Tiwari and Torrellas [34] change the Vdd of a processor only once or twice in its service life, we observe that we can manage aging better if we continuously tune Vdd in small steps over the whole service life — keeping the frequency constant as these authors do. Moreover, we observe that these changes can be done not only to improve performance, but to reduce power consumption as well.

Based on these two observations, we propose *Dynamic Voltage Scaling for Aging Management (DVSAM)*. The idea is to manage the aging rate by continuously tuning Vdd (but not the frequency), exploiting any currently-left aging guard-band. DVSAM is a novel approach to trade-off processor performance, power consumption, and service life for one another.

We propose the four DVSAM modes of Table 1. *DVSAM-Pow* attempts to consume the minimum power for the same performance and service life. *DVSAM-Perf* tries to attain the maximum performance for the same service life and within power constraints. *DVSAM-Short* tries to attain even higher performance for a shorter service life and within power constraints. Finally, *VSAM-Short* is the same as DVSAM-Short but without changing Vdd with time.

DVSAM operates by aggressively trying to consume, at any given time, all the aging guard-band that would be otherwise available. Consequently, the design assumes the existence of aging sensor circuits that reliably measure the guard-band available at all times [2, 7, 17, 18, 20, 28, 31] (Section 4.3.2). Note that circuits have additional guard-bands to protect themselves against other effects such as thermal and voltage fluctuations.

3.2 Detailed Operation

To understand the DVSAM operation, note that the nominal supply voltage Vdd_{NOM} used in a processor is "over-designed" for

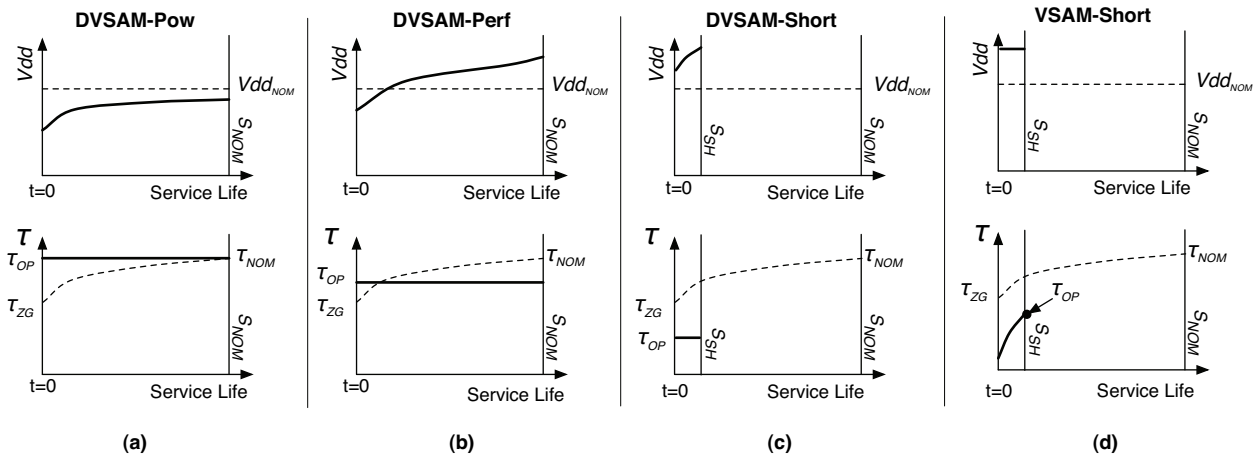

Figure 3: Changes in Vdd (top row) and critical path delay (bottom row) as a function of time for the different DVSAM modes.

Mode: Goal	Vdd Values
DVSAM-Pow: Consume minimum power for the same performance and service life. ($f = f_{NOM}, P < P_{NOM}$)	At $t = 0$: $Vdd << Vdd_{NOM}$ At $t = S_{NOM}$: $Vdd < Vdd_{NOM}$
DVSAM-Perf: Attain maximum performance for the same service life and a given power budget. ($f > f_{NOM}, P > P_{NOM}$)	At $t = 0$: $Vdd < Vdd_{NOM}$ At $t = S_{NOM}$: $Vdd > Vdd_{NOM}$
DVSAM-Short: Attain even higher performance for a shorter service life and a given power budget. ($f >> f_{NOM}, P >> P_{NOM}$)	At $t = 0$: $Vdd > Vdd_{NOM}$ At $t = S_{SH}$: $Vdd >> Vdd_{NOM}$
VSAM-Short: Special case: Same as DVSAM-Short but no Vdd changes with time. ($f >> f_{NOM}, P >> P_{NOM}$)	$\forall$ t $\in [0, S_{SH}]$: $Vdd >> Vdd_{NOM}$

Table 1: DVSAM modes. P denotes total power consumption, with P_{NOM} corresponding to the power consumption of a core clocked at f_{NOM} under nominal operating conditions.

the early parts of the processor's service life. It is designed so that, by the end of the service life, the processor's critical path is just fast enough to avert any timing error. However, earlier on in the service life, before that path aged, that path used to take less time than the cycle time — and its speed was enabled by the Vdd_{NOM}. Clearly, at that time, we could have used a lower Vdd.

This effect is seen analytically by assuming a critical path of identical gates and summing up the switching delays of all the transistors in the path using Equation 3. The critical path delay at the end of the service life (τ_{NOM}), when $Vth = Vth_D$, is supported by Vdd_{NOM}:

$$\tau_{NOM} \propto \frac{Vdd_{NOM}}{\mu(Vdd_{NOM} - Vth_D)^{\alpha}} \quad (5)$$

The same Vdd_{NOM} is used at the beginning of the service life when, because $Vth = Vth_{NOM}$, the same path only takes τ_{ZG}:

$$\tau_{ZG} \propto \frac{Vdd_{NOM}}{\mu(Vdd_{NOM} - Vth_{NOM})^{\alpha}} \quad (6)$$

However, since the processor is clocked at the same frequency throughout the whole service life, keeping these paths so fast is unnecessary. Consequently, DVSAM-Pow reduces Vdd in the early stages of the service life, slowing these paths but ensuring that they do not take longer than τ_{NOM}. The result is power savings. Note that the Vdd reduction becomes gradually smaller, as the paths age. It may be that, by the end of the service life, we can still use a lower Vdd than Vdd_{NOM}. The reason is that, thanks to having applied lower-than-usual Vdd to the processor over a long period, its paths have aged less than usual. Table 1 shows these Vdd values. In the table, the nominal service life is denoted as S_{NOM}.

Alternatively, since the paths have timing slack in the early stages of the service life, DVSAM-Perf increases the frequency of the processor, hence delivering higher performance. However, for simplicity in our design, we want to keep the elevated frequency of the processor constant over the whole service life. To do so, DVSAM-Perf also changes Vdd. Toward the early stages of the service life, to slow down aging as Tiwari and Torrellas [34], Vdd will be set slightly below Vdd_{NOM}. Toward the end of the service life, to keep up with the higher frequency of the processor, Vdd will be set above Vdd_{NOM}. Table 1 shows these Vdd values.

Figures 3(a) and 3(b) show the operation of the DVSAM-Pow and DVSAM-Perf modes, respectively. The top row shows the changes in Vdd as a function of time, while the bottom one shows the changes in critical path delay (τ) as a function of time. Time goes from $t = 0$ to the end of the service life S_{NOM} (e.g., 7 years). Each chart also shows, with a dotted line, the evolution of the parameter value if DVSAM was not applied. Specifically, Vdd stays constant at Vdd_{NOM} (top row), and τ changes from τ_{ZG} at $t = 0$ to τ_{NOM} at S_{NOM} — first quickly and then slowly.

Consider the Vdd charts first. As indicated above, in DVSAM-Pow, Vdd starts significantly lower than Vdd_{NOM}, gradually increases, and reaches a value below Vdd_{NOM} at S_{NOM}. In DVSAM-Perf, Vdd starts slightly lower than Vdd_{NOM}, increases faster, and ends up significantly higher than Vdd_{NOM}.

In the critical path delay charts (bottom row), both modes show a constant critical path delay (labeled τ_{OP}). This is because they dynamically tune the Vdd so that the critical path always takes exactly the same time — balancing the natural lengthening of the critical path due to aging with progressively higher Vdd. In both cases, the processor is clocked at constant frequency $f_{OP} = 1/\tau_{OP}$ over the whole service life. In DVSAM-Pow, τ_{OP} is equal to τ_{NOM}; in DVSAM-Perf, τ_{OP} is kept smaller than τ_{NOM} to enable a higher frequency. DVSAM-Perf can keep pushing τ_{OP} lower at progressively higher power costs. However, we will reach a point where constraints in Vdd, T, or service life duration will prevent any further reduction in τ_{OP}.

To attain higher performance beyond such points, we have the last two DVSAM modes. They deliver higher performance than DVSAM-Perf (at higher power) by giving up service life (Table 1).

This means that the processor will become unusable and be discarded at a time S_{SH} (for short service life), much earlier than S_{NOM}.

Figure 3(c) shows the operation of DVSAM-Short. Already at the start of the service life, Vdd is set to a value higher than Vdd_{NOM} (top chart of the figure). This dramatically reduces the delay of the critical path to τ_{OP} (bottom chart), and enables the processor to cycle at a high frequency. To make up for the lengthening of the critical path with time due to aging, Vdd has to continue to increase with time (top chart). The result is that the critical path takes the same time (bottom chart) and, therefore, the high frequency is maintained. However, since aging is exponential on Vdd and T (Equation 1), the high Vdd and resulting high T rapidly age the critical path. Soon, the aging is such that, to keep up with the high frequency required, Vdd (or T) would have to go above allowed values. At that point, shown as S_{SH} in Figure 3(c), the processor is discarded.

Finally, VSAM-Short is a simpler design than DVSAM-Short. Figure 3(d) shows its operation. Vdd is set to a value higher than Vdd_{NOM} (top chart of the figure). However, instead of dynamically compensating the increase in critical path delay due to aging with higher Vdd, we keep Vdd constant (top chart). As a result, the critical path delay increases (bottom chart). After a relatively short duration S_{SH} (different than for DVSAM-Short), the processor has aged too much and is discarded. Note that the critical path delay is not constant, while we want to keep the frequency constant. Consequently, the processor can only be clocked at the frequency allowed by the path delay at S_{SH}, namely τ_{OP} in the figure.

4. THE BUBBLEWRAP MANY-CORE

4.1 Overview

The *BubbleWrap* is a novel many-core that uses DVSAM to address the problem described in Section 1 of not being able to power-on all the on-chip cores simultaneously. While BubbleWrap uses all DVSAM modes, its most novel characteristic is the use of DVSAM-Short and VSAM-Short.

BubbleWrap has N architecturally homogeneous cores, of which only N_T can be powered-on simultaneously. The architecture distinguishes two groups of cores: *Throughput* (T) and *Expendable* (E) cores. In a die affected by process variation, we select N_T cores as the Throughput cores. We choose the ones that consume the least power at the target frequency. They are used to run the parallel sections of applications where, typically, we want throughput. The rest of the cores ($N_E = N - N_T$) are designated as Expendable cores. They are dedicated to run the sequential sections of applications, where we want per-thread performance.

The Expendable cores form a sequential accelerator. In our most novel designs, we attain acceleration by sacrificing one Expendable core at a time. Specifically, each Expendable core runs under DVSAM-Short (or VSAM-Short) mode, at elevated Vdd and frequency. It delivers high performance, but it ages fast until it can no longer sustain such conditions. At that point, we say that the core *pops*. It is discarded and replaced by another core from the Expendable group. This process of *popping* cores by applying DVSAM-Short (or VSAM-Short) modes gives its name to the BubbleWrap many-core.

Figure 4(a) shows a logical view of the BubbleWrap many-core. The figure depicts a mid-life chip, when some of the Expendable cores have already popped (in black). Although the Throughput and Expendable cores form two logical groups, process variation determines their actual location on the die and, therefore, the cores of each group are typically not physically contiguous. All Throughput cores, however, receive the same supply voltage Vdd_T. When active, an Expendable core receives Vdd_E.

To estimate how quickly BubbleWrap can afford to pop Expendable cores, we add up the sequential-execution time of all the applications that are expected to run on the chip over its service life. This number, as a fraction of the nominal service life S_{NOM} of the chip, is called the *Sequential Load* (L_{SEQ}). For example, if we expect to run 21 applications, each of which runs a sequential section for 6 months, and the nominal service life of the chip is 7 years, then $L_{SEQ} = 21 \times 0.5/7 = 1.5$. Knowing L_{SEQ} and the number of Expendable cores N_E, we can conservatively estimate the short service life of each individual Expendable core S_{SH} as follows

$$S_{SH} = \frac{S_{NOM} \times L_{SEQ}}{N_E} \quad (7)$$

In our example, if we have 32 Expendable cores, each has to last $S_{SH} = 7 \times 1.5/32$, which is approximately 4 months. In reality, each core will take less than 4 months to execute its load because, thanks to its higher Vdd, it runs faster.

Figure 4(b) qualitatively shows how an Expendable core's shorter service life (S_{SH}) permits operation at increasingly higher frequencies. The figure plots the frequency gain over the nominal frequency as a function of S_{SH}. The curve is generated with representative parameter values. It can be shown that the frequency gain increases exponentially with decreasing S_{SH}. Consequently, for (D)VSAM-Short to be profitable, it is required that $S_{SH} \ll S_{NOM}$. Fortunately, we expect that S_{SH} will continue to shrink with time, since technology scaling is providing more Expendable cores with each generation.

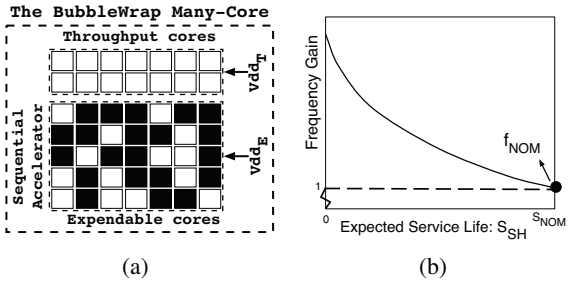

Figure 4: BubbleWrap chip (a) and operation (b).

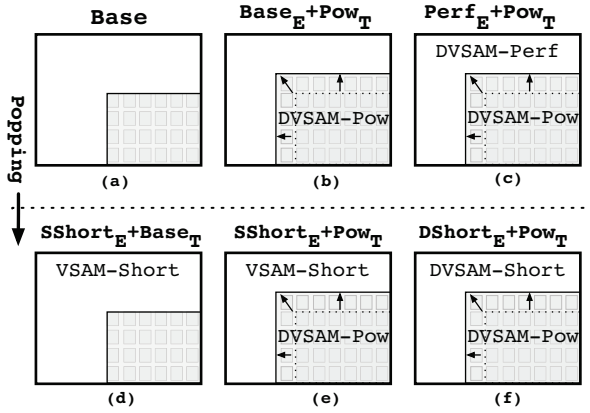

Figure 5: BubbleWrap chips corresponding to different environments. In each environment, Throughput cores are in gray and Expendable cores in white. Recall that *popping* means applying DVSAM-Short or VSAM-Short to the Expendable cores.

4.2 BubbleWrap Environments

The application of the different DVSAM modes of Section 3 to the Throughput or Expendable cores gives rise to six different BubbleWrap environments. The environments are pictorially shown in Figure 5 and described in Table 2.

Chip (a) in Figure 5 shows the *Base* environment, which serves

Environment	Vdd_E	Vdd_T
$Base$	N/A (Cores disabled)	Vdd_{NOM}
$Base_E$+Pow_T	N/A (Cores disabled)	$< Vdd_{NOM}$ (DVSAM-Pow)
$Perf_E$+Pow_T	Variable (DVSAM-Perf)	$< Vdd_{NOM}$ (DVSAM-Pow)
$SShort_E$+$Base_T$	$> Vdd_{NOM}$ (VSAM-Short)	Vdd_{NOM}
$SShort_E$+Pow_T	$> Vdd_{NOM}$ (VSAM-Short)	$< Vdd_{NOM}$ (DVSAM-Pow)
$DShort_E$+Pow_T	$> Vdd_{NOM}$ (DVSAM-Short)	$< Vdd_{NOM}$ (DVSAM-Pow)

Table 2: Voltage applied to the Expendable cores (Vdd_E) and to the Throughput cores (Vdd_T) in each of the BubbleWrap environments.

as the baseline. In *Base*, we disable the Expendable cores and operate the Throughput cores at Vdd_{NOM} and f_{NOM}.

Chip (b) shows the $Base_E$+Pow_T environment, where we keep the Expendable cores disabled and apply DVSAM-Pow to the Throughput cores. Throughput cores operate at f_{NOM} and at a supply voltage less than Vdd_{NOM}. Since each Throughput core now consumes less power, we can *expand* the number of Throughput cores and power-on all of them at the same time. This is shown in Figure 5(b) with the arrows. Overall, this environment increases throughput while clocking all the processors at f_{NOM}.

Chip (c) shows the $Perf_E$+Pow_T environment, where we apply DVSAM-Perf to the Expendable cores (one at a time, during sequential sections) and, as before, DVSAM-Pow to the Throughput cores (to all of them, under expansion, during parallel sections). Expendable cores now deliver higher sequential performance, while Throughput cores deliver higher throughput.

The environments in chips (d), (e), and (f) are the same as the ones in (a), (b), and (c) except that, in addition, we apply core popping. Recall that this means applying VSAM-Short or DVSAM-Short to the Expendable cores. As a result, these environments further increase sequential performance.

The type of popping we perform (VSAM-Short or DVSAM-Short) in each case depends on whether the corresponding chip in Figure 5 already supported DVSAM on Expendable cores to start with. Specifically, if it did not, we apply VSAM-Short; if it did, we apply DVSAM-Short. Consequently, in chip (d), we take *Base* and apply VSAM-Short to the Expendable cores. The result is $SShort_E$+$Base_T$. In chip (e), we take $Base_E$+Pow_T and apply VSAM-Short to the Expendable cores. The result is called $SShort_E$+Pow_T. Finally, in chip (f), we take $Perf_E$+Pow_T and apply DVSAM-Short to the Expendable cores, creating the $DShort_E$+Pow_T environment.

4.3 Hardware Support for BubbleWrap

We describe three hardware components of BubbleWrap, namely the power and clock distribution system, the aging sensors, and the optimizing controller.

4.3.1 Power and Clock Distribution System

Our proposed BubbleWrap design has two voltage and frequency domains in the chip, namely one for the set of Throughput cores and one for the set of Expendable ones. This simple implementation provides enough functionality for our environments of Figure 5. Indeed, in such environments, all the cores in the Throughput set operate under the same conditions, and these conditions are potentially different than those for the Expendable cores. Note that other designs are also possible. For example, in a scenario with high within-die process variation, it may make sense to have one voltage and frequency domain per core. Alternatively, in a scenario where the chip runs a single application at a time, which alternates between parallel and sequential phases, BubbleWrap would only need a single voltage and frequency domain. This simpler design would suffice because we would not have Throughput cores and Expendable cores busy at the same time. However, we do not consider such designs here.

Let us consider the power distribution first. Since the physical location of the Throughput and Expendable cores on chip is unknown until manufacturing test time, we need a design that can supply either Vdd_E or Vdd_T to any core on the die. The most flexible solution is to include two independent supply networks [24], and connect all cores to both grids through power-gating transistors. In this manner, a controller can select, for each core, whether to connect to the Vdd_E grid or the Vdd_T grid by turning on the appropriate transistor.

Figure 6(a) shows such a design. The figure shows a chip with a global controller and one core. The core has a multiplexer that can select one of the two power grids. The controller determines the values of Vdd_E and Vdd_T, along with which grid each core should be connected to. This design has the advantage of allowing cores to move from one grid to the other dynamically, possibly based on their aging conditions.

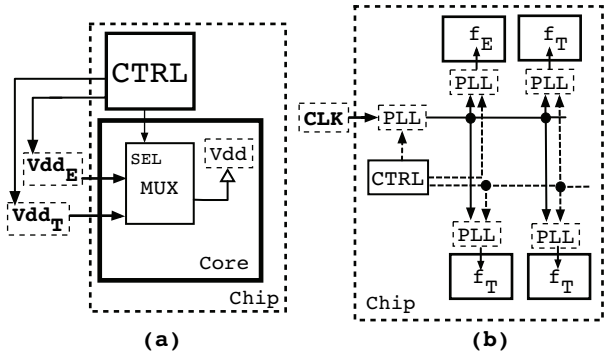

Figure 6: BubbleWrap power (a) and clock (b) distribution.

Figure 6(b) shows the clock distribution network. The figure shows the grid with a global controller and four cores — one Expendable and three Throughput ones. Each core contains a PLL for signal integrity. The controller manages each PLL so that the correct frequency is supplied. This design allows a core to move from one of the core sets to the other dynamically. Moreover, if needed, it can also support per-core frequency domains [10, 15].

4.3.2 Aging Sensors

The effective application of the different DVSAM modes requires that we have a way to reliably estimate the aging that each core experiences. This can be done by using aging sensors that dynamically measure the increase in critical path delays due to aging. The literature proposes a variety of techniques for this purpose, including canary paths distributed throughout the cores or periodic BIST (e.g., [2, 7, 17, 18, 20, 28, 31]). It is claimed that such techniques can measure critical path delays with sub-μs measurement times and sub-*p*s precision [18]. Moreover, Intel Core i7 [15] already includes some of this circuitry to determine the level of Turbo Boost that can be applied.

4.3.3 Optimizing Dynamic Controller

The BubbleWrap controller interacts with the rest of the chip as shown in Figure 7(a). The controller performs several tasks. First, it dynamically adjusts the $Vdd_T(t)$ and $Vdd_E(t)$ supplied to the two sets of cores. Second, it keeps a table of which cores are Throughput, which are Expendable, and which ones are currently running. Based on this information, it sends the core selection signals described in Section 4.3.1. Note that, if it uses any of the BubbleWrap environments with core popping, the controller also determines when an Expendable core can be considered popped

Cox	$2.5 \times 10^{-20} F/nm^2$	n	1.5
k_T	$-1mV/K$	E_0	$0.08V/nm$
T_0	70^oC	Ea	$0.56eV$
T_{MAX}	100^oC	a	0.2
Vdd	$0.8 - 1.3V$	S_{NOM}	7 years
Vdd_{NOM}	$1V$	α	1.3
Vth_{NOM}	$250mV$	G	10% at T_{MAX}
k_{DIBL}	$-150mV/V$	η	0.35

Table 3: Technology parameters.

Technology node: 22nm	f_{NOM}: 4.5GHz
Cores per chip: 16 T + 16 E	P_{MAX}: 80W (all cores), 5W (T core), 10W (E core)
Width: 6-fetch 4-issue 4-retire OoO	L1 D Cache: 16KB WT, 0.44ns round trip, 4 way, 64B line
ROB: 152 entries	L1 I Cache: 16KB, 0.44ns round trip, 2 way, 64B line
Issue window: 40 fp, 80 int	L2 Cache: 2MB WB per core, 2ns round trip,
LSQ Size: 54 LD , 46 ST	8-way, 64B line, has stride prefetcher
Branch pred: 80Kb tournament	Memory: 80ns round trip

Table 4: Microarchitectural parameters.

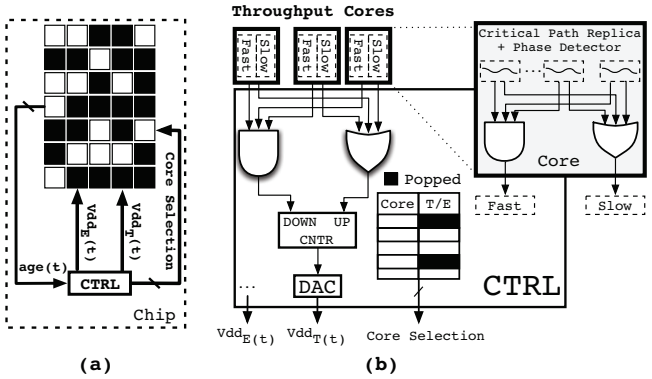

Figure 7: Overview of the BubbleWrap controller.

and keeps track of which cores are already popped. To perform all these tasks, the controller needs age information from all the cores.

We envision a simple hardware-based implementation of the controller. Every time that the operating system (OS) changes the threads running on the chip, it passes information to the controller on which DVSAM mode is most appropriate for each thread, or at least which threads require acceleration. Based on this information and core age information, the controller outputs its initial Vdd_T, Vdd_E, and the core selection signals.

As the threads execute, the controller dynamically tunes the Vdd_T and Vdd_E values. At any time, its goal is to supply the *minimum* Vdd_T (or Vdd_E) value that enables the cores to keep up with the target operational frequency (f_{OP}) for the current DVSAM mode. Note that such frequency is not set by the controller; it is set statically by the manufacturer.

To tune the voltages, the controller relies on the age information provided by the cores in Figure 7(a). To see how this works, Figure 7(b) shows the internals of the controller. The figure also shows 3 Throughput cores, one of which is expanded.

In each core, aging sensors detect aging-induced increases in critical path delays. The figure sketches a design similar to the one by Teodorescu *et al* [32]. It uses multiple critical path replicas distributed across the core along with a phase detector. If each replica satisfies the frequency specification by a certain margin, the core asserts signal *Fast*, to indicate that this core may operate at a lower Vdd and still be clocked at the target frequency. If at least one of the replicas does not satisfy the frequency specification, signal *Slow* is asserted to demand a higher Vdd for this core to support the target frequency. If no signal is set, all of the replicas satisfy the frequency specification with the lowest margin. The combination of the Fast and Slow signals is the age information passed from each core to the controller.

The controller collects all Fast and Slow signals from all Throughput cores and combines them using a similar circuit inside the controller (Figure 7(b)). The circuit asserts signal *DOWN* when a lower Vdd may be feasible to cycle at the target frequency; it asserts signal *UP* if at least one of the cores requires a higher Vdd. If no signal is asserted, all of the cores satisfy the frequency at the lowest possible power budget. Finally, *DOWN* and *UP* form the control inputs to a counter. The counter is then connected to a digital-to-analog converter (DAC), which converts the counter output to Vdd_T.

Although not shown in the figure, an analogous circuit exists for Expendable cores. In addition, the figure shows the table that records the mapping of cores to Throughput or Expendable type, and which cores are already popped.

The characteristics of the DAC, together with the magnitude of the voltage noise determine the grain size at which the DVSAM modes can tune Vdd.

5. EVALUATION SETUP

In this section, we present the evaluation methodology that we use to characterize a BubbleWrap chip in a near-future process technology.

5.1 Power and Thermal Model

We use a simple model to estimate power and temperature values in a BubbleWrap chip. We estimate the dynamic power consumed by a core using Wattch [8]; for the static power, we use the following equations:

$$P_{STA} = Vdd \times I_{LEAK}$$
$$I_{LEAK} \propto \mu \times T^2 \times e^{-qVth/kTn} \quad , where \ \ \mu \propto T^{-1.5} \tag{8}$$

Adding up the power consumed by all of the on-chip cores in P_{TOT}, we use the following equations to estimate the (junction) temperature T_J of a core:

$$T_J = T_S + \theta_{JS} \times (P_{STA} + P_{DYN}) \ \ \textit{(per core)}$$
$$T_S = T_A + \theta_{SA} \times P_{TOT} \tag{9}$$

In the equations, θ_{SA} is the spreader to ambient thermal resistance, which is modeled as a lumped equivalent of spreader to heat-sink and heat-sink to ambient thermal resistances. Moreover, T_A is the ambient temperature, T_S is the spreader temperature, P_{STA} and P_{DYN} are the static and dynamic power consumptions of the core, and θ_{JS} is the junction to spreader thermal resistance. We assume an ambient temperature T_A of 45°C and a θ_{SA} of 0.222K/W [15], while θ_{JS} is calibrated for the worst case operating conditions.

For the near-future technology node we are assuming, the core area is already so small that intra-core spatial thermal variation becomes negligible. Therefore, we model temperature at core granularity, which offers reasonable fidelity for many-core designs [14]. Moreover, since we assume a checkerboard core-cache layout, which reduces core-to-core thermal coupling significantly, we neglect inter-core lateral conduction. The cache power density is significantly lower than the core power density. Hence, caches placed between cores act as virtual lateral heat-sinks.

We impose a maximum chip power in the cores of 80W, and a maximum power of 5W per Throughput core. A core is designed to support a power limit of 10W; going above that could damage its power-distribution system. Consequently, 10W is the effective maximum power for an Expandable core.

5.2 Process Technology

We assume a 22nm technology node based on the Predictive Technology Model's bulk HK-MG process [37], which incorporates recent corrections [19]. The technology parameters are given in Table 3.

We use a very simple model to estimate the effects of Vth process variation. Power and performance asymmetry due to process variation at the core granularity stems mainly from systematic variation, which shows strong spatial dependence [33]. Hence, given the small area taken by a core, we neglect any intra-core Vth variation. Moreover, we assume a normally-distributed core-to-core variation in Vth with $\sigma/\mu = 5\%$. In our evaluation, we perform some experiments on cores with $[-3\sigma, +3\sigma]$ deviation from Vth_{NOM}. In our experiments, such range causes a variation of approximately 12% in power consumption between the most power consuming core and the least consuming one. Moreover, it leads to a variation of approximately 14% in frequency between the fastest core and the slowest one.

In our experiments, we assume that processors have a nominal service life S_{NOM} of 7 years. The constant of proportionality of Equation 1, A_{BTI}, is calibrated so that a core with a $+3\sigma$ Vth value (i.e., the slow corner) slows down by $G = 10\%$ at the end of S_{NOM} if operated continuously at Vdd_{NOM} and T_{MAX}. The constant of proportionality for the alpha-power law (Equation 3) is calibrated to guarantee operation at f_{ZG} for a core with a $+3\sigma$ Vth value at the beginning of the service life, at Vdd_{NOM} and T_0. Finally, the constant of proportionality for leakage current (Equation 8) is set so that leakage accounts for 25% of the total power consumption for a core with a -3σ Vth value (i.e., the leaky corner) at Vdd_{NOM} and T_0.

5.3 Workload

Each non-idle core repeatedly cycles through the SPECint2000 applications in a round-robin fashion (switching every $\approx$ 45minutes of simulated time), therefore experiencing a diverse range of work over its service life. We create parallel sections, where all Throughput cores are busy running this load (or an expanded number of Thoughput cores). We also create sequential sections, where only one Expendable core is running. To assess how BubbleWrap performs with different proportions of parallel and sequential sections, we parameterize the workload with the fraction W_{SEQ} of time spent in the sequential section — for the default number of Throughput cores, without expansion. The evaluation characterizes BubbleWrap for $W_{SEQ} = [0, 1]$.

5.4 Many-Core Microarchitecture

We model a near-future 32-core many-core microarchitecture as described in Table 4. Based on ITRS projections [16] (along with corrections based on current industry status), we use 16 Throughput and 16 Expendable cores as the default (not counting expansion for the set of Throughput cores). We simulate the core microarchitecture with the SESC simulator [25] instrumented with Wattch models [8].

6. EVALUATION

In this section, we assess the impact of BubbleWrap in terms of power and performance. The evaluation optimistically assumes that the aging sensors always give correct information on the critical path length and that the BubbleWrap controller adds no overhead. In the following, we analyze DVSAM-Pow, DVSAM-Perf, DVSAM-Short and VSAM-Short, and finally the different BubbleWrap environments.

6.1 Enhancing Throughput: DVSAM-Pow

For the evaluation, we consider three cores to cover the whole Vth spectrum: one with Vth_{NOM}, one with Vth at -3σ, and one with Vth at $+3\sigma$. They are all clocked at f_{NOM} and work at workload temperature conditions. To each of these cores, we apply DVSAM-Pow. The second column of Table 5 shows, for each of the cores, the energy reductions attained with DVSAM-Pow over the entire service life of the core. From the table, we see that DVSAM-Pow reduces the energy consumption by 13–31%. The savings are more pronounced for the cores with low Vth, which suffer from higher static energy consumption. For the core with Vth_{NOM}, the savings are a substantial 23%.

Deviation in Vth	Energy Savings due to DVSAM-Pow (%)	Frequency Increases due to DVSAM-Perf (%)
-3σ	31	18
0	23	14
$+3\sigma$	13	10

Table 5: Benefits of DVSAM-Pow and DVSAM-Perf.

We now take the core with Vth_{NOM} before and after the optimization and plot its temporal evolution over the whole service life. We call the two resulting cores *Base* and *DVSAM-Pow*, respectively, and show the Vdd evolution (Figure 8(a)), normalized critical path delay τ/τ_{ZG} evolution (Figure 8(b)), power evolution (Figure 8(c)), and temperature evolution (Figure 8(d)). The plots show banded structures for the curves. They are due to temporal variations in the workload, as we execute the SPECint2000 applications in a round-robin manner.

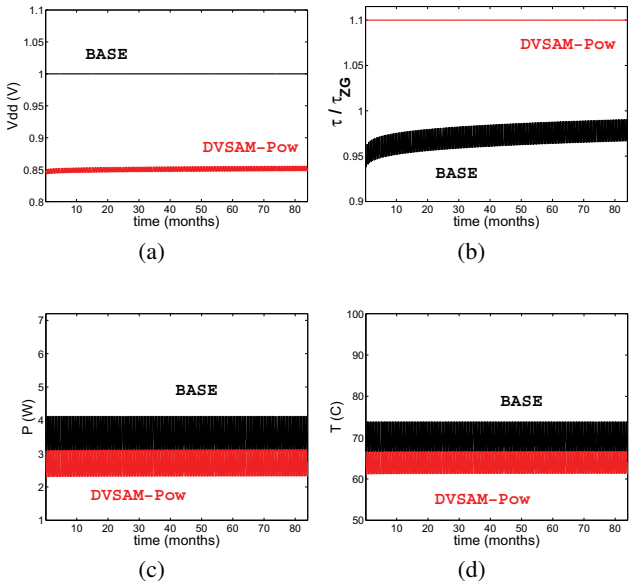

Figure 8: Temporal evolution of the effects of DVSAM-Pow.

Figure 8(a) corresponds to the top row of Figure 3(a). We see that DVSAM-Pow keeps Vdd about 0.15V lower than Base. Moreover, the difference does not decrease much over the whole lifetime. To understand why, consider Figure 8(b), which corresponds to the second row of Figure 3(a). While DVSAM-Pow's critical path consumes all the guard-band from the beginning (by design), Base's critical path delay increases only a little, never consuming much of the guard-band. The reason is because the guard-band is dimensioned for the worst case conditions, namely a core with Vth at $+3\sigma$ operating at T_{MAX}. In our case, Base has Vth_{NOM} and operates at workload temperatures. As a result, it does not age as much and does not use most of the guard-band. Overall, the across-the-board gap in Figure 8(b) induces the resulting gap in Figure 8(a).

Note that DVSAM-Pow only consumes the guard-band set aside for aging (and variation). There is additional guard-banding present for other reasons, such as voltage noise. That one remains.

Figure 8(c) shows that the lower operating voltage of DVSAM-Pow saves significant power compared to the core operating continuously at Vdd_{NOM}. Finally, Figure 8(d) shows that the temper-

ature also decreases due to the lower operating voltage.

6.2 Enhancing Frequency: DVSAM-Perf

We now take the three cores covering the Vth spectrum as explained in the beginning of Section 6.1 and apply DVSAM-Perf. The third column of Table 5 shows, for each of the cores, the frequency increases attained with DVSAM-Perf over f_{NOM}. From the table, we see that DVSAM-Perf increases the frequency by 10–18%. The increase is larger for the core with low Vth, since this core can cycle faster for any given Vdd. For the core with Vth_{NOM}, the increase is a significant 14%.

Using the core with Vth_{NOM} before and after the optimization, we plot its temporal evolution over the whole service life. We call the resulting two cores *Base* and *DVSAM-Perf*, respectively, and show the evolution of Vdd (Figure 9(a)), normalized critical path delay τ/τ_{ZG} (Figure 9(b)), power (Figure 9(c)), and temperature (Figure 9(d)).

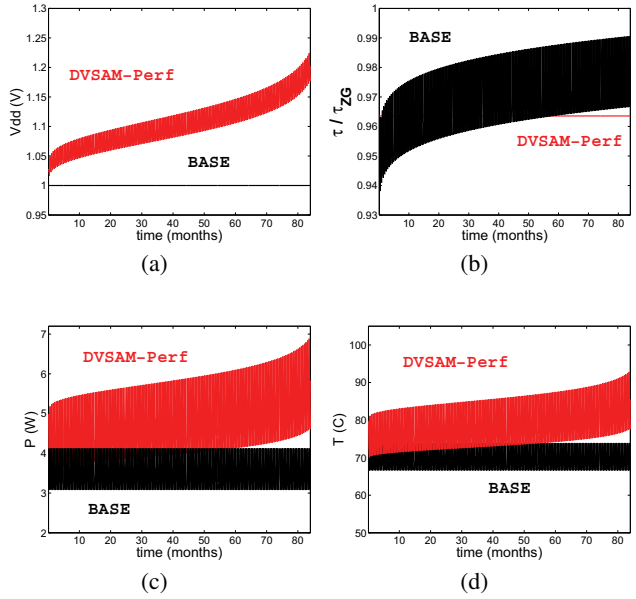

Figure 9: Temporal evolution of the effects of DVSAM-Perf.

Figure 9(a) corresponds to the top row of Figure 3(b). We see that, with DVSAM-Perf, Vdd starts around Vdd_{NOM} at the beginning of the service life and increases beyond Vdd_{NOM} thereafter. At the end of the service life, it reaches around 1.25V. Consider now Figure 9(b), which corresponds to the second row of Figure 3(b). Thanks to DVSAM-Perf's high Vdd operation, DVSAM-Perf keeps the critical path delay around $0.96\times\tau_{ZG}$. As a result, DVSAM-Perf operates at a constant frequency that is 14% higher than f_{NOM} over the whole service life. Recall that f_{NOM} is the frequency of the Base core. It has a period of $\tau_{ZG} \times (1 + G) = \tau_{ZG} \times 1.1$. This substantial frequency increase, even over the frequency of no guard-band ($1/\tau_{ZG}$) is possible because the guard-band is dimensioned for a core with Vth at $+3\sigma$ operating at T_{MAX}.

Figure 9(c) shows that the higher voltages of DVSAM-Perf cause a continuous increase in core power. By the end of the service life, the power consumed by a core is 7W. This is a high value, but still less than the 10W reserved to Expendable cores. As shown in Figure 9(d), the junction temperature goes above 90°C at the end of the service life.

6.3 Popping: DVSAM-Short & VSAM-Short

We now consider the effect of popping Expendable cores, first with DVSAM-Short and then with VSAM-Short. As before, we use a core with Vth_{NOM}. We are interested in the evolution of the core at elevated voltage and frequency conditions for a period S_{SH} that ranges from 0 to S_{NOM}.

We consider DVSAM-Short first. Figure 10 considers all possible values of S_{SH} and shows: The maximum Vdd that gets applied to the core (Chart (a)), the frequency of the core relative to f_{NOM} (Chart (b)), the maximum power consumed by the core (Chart (c)), and the maximum temperature attained by the core (Chart (d)). Each chart highlights two data points, namely one for a one-month service life (labeled *1mo*) and one for a nominal service life (labeled S_{NOM}). The latter corresponds to the DVSAM-Perf mode.

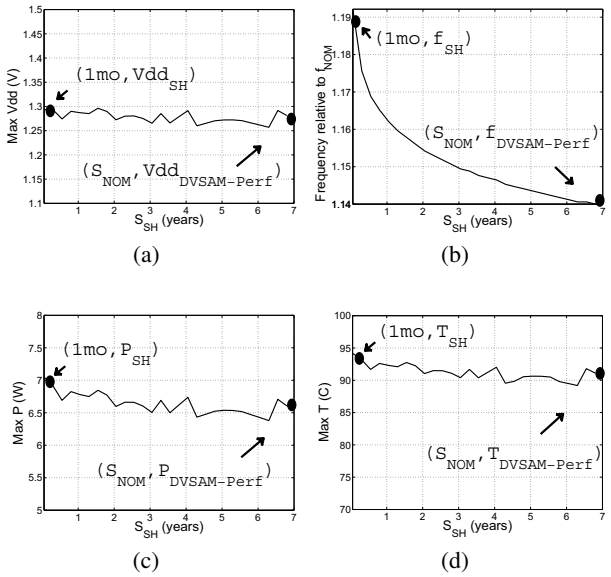

Figure 10: Impact of core popping with DVSAM-Short.

Recall from the top row of Figure 3(c) that, under DVSAM-Short, Vdd starts-off elevated and continues to increase until the core pops (i.e., we need to violate Vdd_{MAX}, P_{MAX}, or T_{MAX} to maintain the frequency). Figure 10(a) shows that, in all cases of S_{SH}, Vdd practically reaches Vdd_{MAX}, which is 1.3V. Figures 10(c) and 10(d) show that the maximum core power and temperature, respectively, are fairly similar for different S_{SH}, and have not reached their allowed limit — core power is 6.5–7W and temperature is 90–95°C.

Finally, Figure 10(b) shows that the frequency at which we can clock the core increases as S_{SH} becomes smaller. For a service life of 1 month, the core can be clocked at a frequency of 1.19 relative to f_{NOM}. However, as S_{SH} increases, the frequency quickly goes down. Very soon, we attain frequencies not much higher than the one reached by DVSAM-Perf, namely 1.14 relative to f_{NOM}. This is because the Vdd conditions required to deliver high frequency quickly age the core, limiting its service life.

We now consider VSAM-Short. This mode represents a simpler approach than DVSAM-Short because we do not need to repeatedly adjust Vdd based on measurements of the critical path delays. We simply set a constant, elevated Vdd_{SH} for the duration of the shorter service life S_{SH}. This was shown in the top row of Figure 3(d). However, this mode is less effective than DVSAM-Short because we need to set Vdd_{SH} conservatively — with the same conservative assumptions as we set Vdd_{NOM} when we want the processor to last for S_{NOM}. Specifically, Vdd_{SH} should guarantee that, if operated continuously at T_{MAX}, the core with $+3\sigma$ Vth deviation (the slow corner) consumes the entire guard-band only at the end of its presumed short service life (S_{SH}).

With these assumptions, we generate Figure 11(a), which shows

for each short service life S_{SH}, the elevated Vdd_{SH} that we need to apply. The figure highlights two data points, namely one for a one-month service life (labeled *1mo*) and one for the nominal service life (labeled S_{NOM}). The latter corresponds to a core operating at Vdd_{NOM}. We can see that, for $S_{SH} = 1$ month, we get a Vdd_{SH} of only 1.1V.

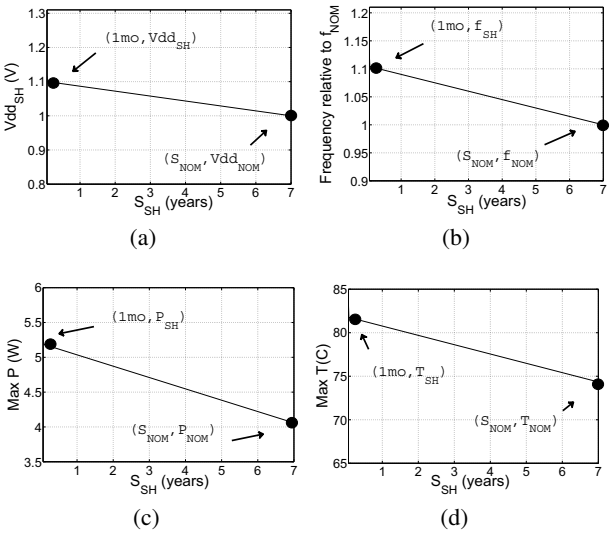

Figure 11: Impact of core popping with VSAM-Short.

Figures 11(b), (c), and (d) repeat Figures 10(b), (c), and (d) for VSAM-Short. As usual, we characterize a core with Vth_{NOM}. From Figure 11(b), we see that the frequency at which we can clock the core increases as S_{SH} becomes smaller. However, VSAM-Short does not reach the frequency values that DVSAM-Short attains in Figure 10(b). At a service life of 1 month, VSAM-Short clocks the core at a frequency of 1.1 relative to f_{NOM}, in contrast to DVSAM-Short's 1.19.

Figures 11(c) and 11(d) show that the maximum power and temperature reached by an Expendable core with a short service life of 1 month are around 5.2W and around 82°C, respectively. This is in contrast to the higher values attained with DVSAM-Short, namely 7W and 93°C (Figures 10(c) and 10(d)).

6.4 BubbleWrap Environments

We now estimate the performance and power impact of each of the BubbleWrap environments described in Figure 5 and Table 2. BubbleWrap increases the frequency of sequential sections by popping Expendable cores, and the throughput of parallel sections by expanding the number of Throughput cores. Next, we examine the frequency of sequential sections, the throughput of parallel sections, and finally the performance and power of the application as a whole.

6.4.1 *Sequential Section Frequency*

The frequency increase achievable by popping cores is a function of the service life per Expendable core (S_{SH}). From Equation 7, S_{SH} depends on the sequential load (L_{SEQ}). The smaller L_{SEQ} is, the shorter is S_{SH}, and the larger is the frequency boost provided by each Expendable core.

To study a range of S_{SH}, we take our workload from Section 5.3 and vary the fraction of its original execution time in the sequential section (W_{SEQ}), from 1 to 0. Note that, for our workload, $L_{SEQ} = W_{SEQ}$. Then, for different values of W_{SEQ}, Figure 12 shows the frequency attained by the sequential section relative to f_{NOM} in all the BubbleWrap environments of Figure 5.

There are three groups of environments. First, there are the environments that do not use Expendable cores ($Base$ and $Base_E + Pow_T$); these cannot get any increase in frequency. The second group is the environment that uses Expendable cores but does not pop them ($Perf_E + Pow_T$). Expendable cores are expected to last for S_{NOM} and, for performance, are run in DVSAM-Perf mode. As a result, $Perf_E + Pow_T$ increases the frequency of the sequential section by a fixed amount irrespective of W_{SEQ}. We see that this increase is 14%, and was already shown in Table 5.

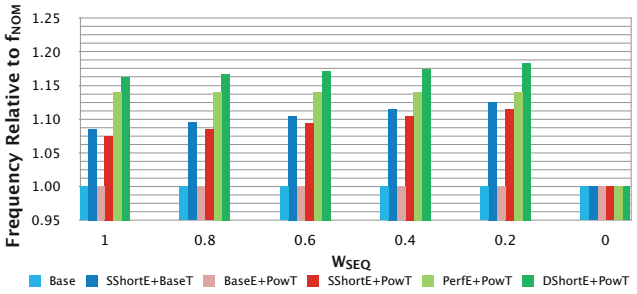

Figure 12: Frequency of the sequential section for each environment.

Finally, there are the environments that pop Expendable cores ($SShort_E + Base_T$, $SShort_E + Pow_T$ and $DShort_E + Pow_T$). These environments increase the sequential section frequency. The lower W_{SEQ} is, the higher their impact is, except for $W_{SEQ} = 0$, where they have no impact. Among the three environments, the one that uses DVSAM-Short (rather than VSAM-Short), is the one with the highest impact : $DShort_E + Pow_T$. For a fully sequential execution ($W_{SEQ} = 1$), we see that this environment provides a 16% frequency increase. Finally, $SShort_E + Pow_T$ increases the frequency less than $SShort_E + Base_T$ because, in $SShort_E + Pow_T$, some Expendable cores are turned into Throughput ones during the parallel section when the set of Throughput cores expands.

6.4.2 *Parallel Section Throughput*

The throughput increase achievable by BubbleWrap environments during parallel sections depends on whether or not they can expand the set of Throughput cores. There are two groups of environments. First, there are the environments that do not expand the set of Throughput cores ($Base$ and $SShort_E + Base_T$); these cannot get any increase in throughput. The other group is the rest of environments ($Base_E + Pow_T$, $SShort_E + Pow_T$, $Perf_E + Pow_T$ and $DShort_E + Pow_T$); they all use the DVSAM-Pow mode on Throughput cores and, therefore, expand them equally during parallel sections. Assuming that the parallel section has enough parallelism, we can increase the number of Throughput cores to make up for the power saved by DVSAM-Pow. As shown in Figure 8(c), DVSAM-Pow reduces the power of a core from 4.1W to 3.1W on average. This is a reduction of 24%. Consequently, for constant power, the number of Throughput cores to execute the parallel section can be increased from the original 16 cores to $16 \times 4.1/3.1 \approx 21$ cores. This represents a throughput increase of $s_t \approx 30\%$ for these BubbleWrap environments.

6.4.3 *Estimated Overall Speedup and Power Cost*

From the previous two sections, we can now roughly estimate the overall speedup and the power cost of each BubbleWrap environment. We consider a workload that has a fraction of sequential time W_{SEQ}. For each BubbleWrap environment, we assume that the time of the parallel section scales down perfectly with the corresponding throughput increase s_t of Section 6.4.2; similarly, we assume that the time of the sequential section scales down perfectly with the relative frequency increase f_r of Figure 12. With these optimistic assumptions, we obtain the following execution time speedup over *Base*

$$Speedup = \frac{Time_{unoptimized}}{Time_{optimized}} = \frac{1}{W_{SEQ}/f_r + (1 - W_{SEQ})/s_t} \quad (10)$$

Figure 13 shows these speedups for all the BubbleWrap environments and different W_{SEQ} values. The speedups are normalized to $Base$. There are three groups of environments. First, $SShort_E + Base_T$ only speeds-up sequential sections; its impact decreases with W_{SEQ}. Second, $Base_E + Pow_T$ only speeds-up parallel sections and, therefore, its effect goes down as W_{SEQ} increases. Finally, the remaining three environments speed-up both sequential and parallel sections. They are the environments that show the highest speedups across most of the W_{SEQ} range. Their relative speedups follow their relative increase in sequential section frequency in Figure 12. Overall, $DShort_E + Pow_T$ gives the highest speedups, which range from 1.16 to 1.30.

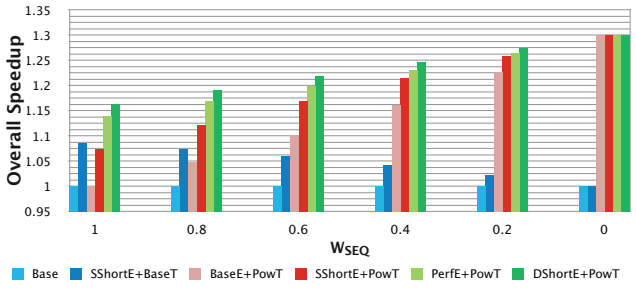

Figure 13: Speedup of BubbleWrap environments.

Note that the best performing environment ($DShort_E + Pow_T$, which supports core popping) is not any more complex than the next best-performing one ($Perf_E + Pow_T$, which does not support core popping). Both environments use two supply networks, tune Vdd_E and Vdd_T, and have aging sensors in both Expendable and Throughput cores. However, the third best-performing environment ($SShort_E + Pow_T$) is significantly simpler, since it does not require tuning Vdd_E or keeping aging sensors for Expendable cores.

Finally, Figure 14 shows the power consumption of the different BubbleWrap environments normalized to the power of *Base*. The power cost of BubbleWrap operation stems only from sequential section acceleration — since the expansion in the number of Throughput cores has no power cost. Consequently, the normalized power increases with W_{SEQ}. Moreover, the figure shows that the most power-consuming environments are those that use the DVSAM-Short (or DVSAM-Perf) modes — namely, $DShort_E + Pow_T$ and $Perf_E + Pow_T$. This is because these modes raise Vdd_E to high values. On the other hand, the environments that use the VSAM-Short mode ($SShort_E + Base_T$ and $SShort_E + Pow_T$) consume much less power. This is because VSAM-Short does not tune Vdd_E and, therefore, has to set Vdd_E to conservatively low values.

6.5 Discussion

All the data shown in our evaluation except for Table 5 corresponds to processors with Vth_{NOM}. In reality, the cores in a BubbleWrap chip will exhibit process variation, which will affect how they respond to the BubbleWrap modes as shown in Table 5. This makes some of our results slightly optimistic while others slightly pessimistic. Specifically, the best cores will be chosen for the Throughput set. This is likely to result in higher parallel section speedup than reported here. However, it will also leave the worst cores for the Expendable set, which will result in lower sequential section speedup (or higher power consumption) than reported here. A further study should be done to address this issue.

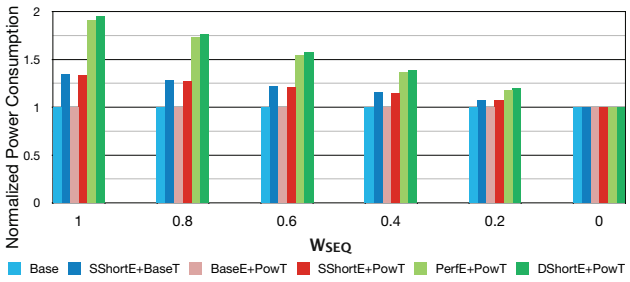

Figure 14: Power consumption of BubbleWrap environments.

We constrained the operation of all cores, including those that pop, to be within the T_{MAX} and Vdd_{MAX} envelopes. Respecting T_{MAX} and Vdd_{MAX} ensures a reliable service life of S_{NOM}. In reality, for a core that will have a short life, it may be possible to go above T_{MAX} or Vdd_{MAX}. This could allow core popping to deliver higher performance than reported here — perhaps at the cost of risking new types of hard or soft failures. This topic will be studied in future work.

7. RELATED WORK

The impending many-core power wall is well-known in industry and reflected in recent ITRS projections [16]. However, our suggestion to expend (or pop) the excess cores that cannot be powered-on is novel as far as we know. Another proposal to address the many-core power wall is to use extremely low supply voltages, possibly close to the threshold voltage [12, 36]. This environment would allow all cores to operate simultaneously, albeit at severely reduced frequencies. Yet another alternative is a design comprising power-efficient, heterogeneous application-specific accelerators [11]. BubbleWrap is unique in extending the scaling of homogeneous many-cores without requiring core modifications.

Recently, processor aging has been the subject of interest in the computer architecture community (e.g., [1, 2, 7, 27, 28, 29, 30, 34]). Some of this work attempts to reduce aging by setting the logic to recovery mode [1, 27]. However, the aging work most closely related to ours is Facelift [34]. Facelift proposed applying a few discrete voltage levels to minimize aging and showed how shorter service lifes could be exploited to increase core frequency. BubbleWrap's DVSAM framework improves on these techniques with continuous voltage tuning and with the power-saving DVSAM-Pow mode. Additionally, BubbleWrap proposes a set of novel architectures (or environments) that use DVSAM. Finally, several authors have designed circuits that detect when a critical path has slowed down [2, 7, 17, 18, 20, 28, 31]. Some of the BubbleWrap environments use these aging sensors.

8. CONCLUSION

To push back the many-core power wall, this paper made two main contributions. First, it introduced *Dynamic Voltage Scaling for Aging Management* (DVSAM) — a new scheme for managing processor aging by tuning Vdd (but not the frequency), exploiting any currently-left aging guard-band. The goal can be one of the following: consume the least power for the same performance and service life; attain the highest performance for the same service life and within power constraints; or attain even higher performance for a shorter service life and within power constraints.

Second, this paper presented *BubbleWrap*, a novel many-core architecture that makes extensive use of DVSAM to push back the many-core power wall. BubbleWrap selects the most power-efficient set of cores in the die — the largest set that can be simultaneously powered-on — and designates them as Throughput cores. They are dedicated to parallel-section execution. The rest of the cores are designated as Expendable. They are dedicated to accelerating sequential sections. BubbleWrap applies DVSAM in several environments. In some of them, it attains maximum sequential ac-

celeration by sacrificing Expendable cores one at a time, running them at elevated Vdd for a significantly shorter service life each, until they completely wear-out.

In simulated 32-core chips, BubbleWrap provides substantial gains over a plain chip with the same power envelope. On average, our most aggressive design runs fully-sequential applications at a 16% higher frequency, and fully-parallel ones with a 30% higher throughput. We are now extending DVSAM to also include changes in processor frequency with time. This improvement should deliver better design points. We are also working on a model to accurately capture the deviations from the nominal behavior within a logical set of cores.

9. ACKNOWLEDGMENTS

We thank the anonymous reviewers and the I-ACOMA group members for their comments. This work was supported by Sun Microsystems under the UIUC OpenSPARC Center of Excellence, NSF under grant CPA-0702501, and SRC GRC under grant 2007-HJ-1592.

10. REFERENCES

[1] J. Abella, X. Vera, and A. González. Penelope: The NBTI-Aware Processor. In *Int. Symp. on Microarchitecture*, December 2007.

[2] M. Agarwal et al. Circuit Failure Prediction and Its Application to Transistor Aging. In *VLSI Test Symp.*, May 2007.

[3] F. Arnaud et al. 32nm General Purpose Bulk CMOS Technology for High Performance Applications at Low Voltage. In *Electron Devices Meeting*, December 2008.

[4] C. Auth. 45nm High-k + Metal Gate Strain-Enhanced CMOS Transistors. In *Custom Integrated Circuits Conference*, September 2008.

[5] D. Bergstrom. 45nm Transistor Reliability. In *Intel Technology Journal*, June 2008.

[6] K. Bernstein et al. High-Performance CMOS Variability in the 65-nm Regime and Beyond. In *IBM Journal of Research and Development*, July/September 2006.

[7] J. Blome et al. Self-Calibrating Online Wearout Detection. In *Int. Symp. on Microarchitecture*, December 2007.

[8] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. In *Int. Symp. on Computer Architecture*, June 2000.

[9] R. Dennard et al. Design of Ion-Implanted MOSFETs with Very Small Physical Dimensions. In *Journal of Solid-State Circuits*, October 1974.

[10] J. Dorsey et al. An Integrated Quad-Core Opteron Processor. In *Int. Solid-State Circuits Conference*, February 2007.

[11] K. Fan et al. Bridging the Computation Gap between Programmable Processors and Hardwired Accelerators. In *Int. Symp. on High Performance Computer Architecture*, February 2009.

[12] R. Gonzalez, B. Gordon, and M. Horowitz. Supply and Threshold Voltage Scaling for Low Power CMOS. In *Journal of Solid-State Circuits*, August 1997.

[13] M. Horowitz et al. Scaling, Power, and the Future of CMOS. In *Int. Electron Devices Meeting*, December 2005.

[14] W. Huang et al. Many-Core Design from a Thermal Perspective. In *Design Automation Conference*, July 2008.

[15] Intel Corporation. Intel Core i7 Processor Extreme Edition and Intel Core i7 Processor Datasheet, November 2008.

[16] International Technology Roadmap for Semiconductors. 2008 Update.

[17] E. Karl et al. Compact In-Situ Sensors for Monitoring Negative-Bias-Temperature-Instability Effect and Oxide Degradation. In *Int. Solid-State Circuits Conference*, February 2008.

[18] J. Keane, D. Persaud, and C. H. Kim. An All-in-one Silicon Odometer for Separately Monitoring HCI, BTI, and TDDB. In *Symp. on VLSI Circuits*, June 2009.

[19] A. Khakifirooz and D. Antoniadis. MOSFET Performance Scaling Part II: Future Directions. In *Transactions on Electron Devices*, June 2008.

[20] T.-H. Kim, R. Persaud, and C. Kim. Silicon Odometer: An On-Chip Reliability Monitor for Measuring Frequency Degradation of Digital Circuits. In *Journal of Solid-State Circuits*, April 2008.

[21] J. W. McPherson. Reliability Challenges for 45nm and Beyond. In *Design Automation Conference*, July 2006.

[22] E. J. Nowak. Maintaining the Benefits of CMOS Scaling When Scaling Bogs Down. In *IBM Journal of Research and Development*, March/May 2002.

[23] S. Pae et al. BTI Reliability of 45 nm High-k + Metal-Gate Process Technology. In *Int. Reliability Physics Symp.*, May 2008.

[24] M. Popovich, A. Mezhiba, and E. Friedman. *Power Distribution Networks with On-Chip Decoupling Capacitors*, chapter 15. Springer, 2008.

[25] J. Renau et al. SESC simulator, January 2005. http://sesc.sourceforge.net.

[26] T. Sakurai and A. Newton. Alpha-Power Law MOSFET Model and Its Applications to CMOS Inverter Delay and Other Formulas. In *Journal of Solid-State Circuits*, April 1990.

[27] J. Shin et al. A Proactive Wearout Recovery Approach for Exploiting Microarchitectural Redundancy to Extend Cache SRAM Lifetime. In *Int. Symp. on Computer Architecture*, June 2008.

[28] J. C. Smolens et al. Detecting Emerging Wearout Faults. In *Workshop on Silicon Errors in Logic - System Effects*, 2007.

[29] J. Srinivasan et al. The Case for Lifetime Reliability-Aware Microprocessors. In *Int. Symp. on Computer Architecture*, June 2004.

[30] J. Srinivasan et al. Exploiting Structural Duplication for Lifetime Reliability Enhancement. In *Int. Symp. on Computer Architecture*, June 2005.

[31] K. Stawiasz, K. Jenkins, and P.-F. Lu. On-Chip Circuit for Monitoring Frequency Degradation Due to NBTI. In *Int. Reliability Physics Symp.*, May 2008.

[32] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas. Mitigating Parameter Variation with Dynamic Fine-Grain Body Biasing. In *Int. Symp. on Microarchitecture*, December 2007.

[33] R. Teodorescu and J. Torrellas. Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors. In *Int. Symp. on Computer Architecture*, June 2008.

[34] A. Tiwari and J. Torrellas. Facelift: Hiding and Slowing Down Aging in Multicores. In *Int. Symp. on Microarchitecture*, November 2008.

[35] W. Wang et al. Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology. In *Transactions on Device and Materials Reliability*, December 2007.

[36] B. Zhai et al. Energy Efficient Near-Threshold Chip Multi-Processing. In *Int. Symp. on Low power Electronics and Design*, 2007.

[37] W. Zhao and Y. Cao. New Generation of Predictive Technology Model for Sub-45nm Design Exploration. In *Int. Symp. on Quality Electronic Design*, 2006.