

Reducing Wearout in Embedded Processors Using Proactive Fine-Grain Dynamic Runtime Adaptation

Fabian Oboril and Mehdi B. Tahoori

Chair of Dependable Nano Computing (CDNC), Karlsruhe Institute of Technology (KIT)

Karlsruhe, Germany

Email: {fabian.oboril, mehdi.tahoori}@kit.edu

Abstract—With shrinking feature sizes, transistor aging becomes a reliability challenge for embedded processors. Processes such as NBTI and HCI lead to increasing gate delays and eventually reduced lifetime. Currently, to ensure functionality for a certain lifetime, safety margins are added to the design, which means overdesign and increased costs. To extend lifetime, reduce power and heat, while maintaining the required performance we propose a dynamic runtime adaptation approach, which is based on runtime monitoring of temperature, performance, power and wearout in combination with fine-grained proactive dynamic voltage and frequency scaling. The experimental results presented in this work show lifetime improvements between 63% up to 5x, while the required performance as well as power and temperature constraints are maintained.

I. INTRODUCTION

Nowadays, embedded systems are indispensable in our daily life and can be found in many fields from multimedia devices, smartphones, cars to almost every household appliance. Most of these systems underlie several constraints. First of all in many application areas costs have to be low. Second, many of them have to be power efficient and deliver a high performance. And third, they have to be reliable and have to have a certain minimum lifetime. Due to the first two aspects, many embedded processors use the newest hardware technologies with the smallest possible feature sizes. However, with approaching nanoscale dimensions these processors face various reliability challenges. Thereby faster wearout due to transistor aging, is one major reliability issue [5], [7].

Among several physical effects that cause transistor aging, *Negative Bias Temperature Instability* (NBTI) [20] and *Hot Carrier Injection* (HCI) [18] are the dominant effects [5]. Both phenomena lead to a shift of the threshold voltage (V_{th}) of the impaired transistors, which manifests in increasing switching and path delays. This will eventually lead to timing violations and finally to faster wearout of the system. Thereby, NBTI- and HCI-induced wearout strongly depends on several key factors such as usage (gate bias, number of transitions, etc.), temperature and supply voltage of the affected transistors.

Currently manufacturers deal with aging by adding safety margins so called *guardbands* to their designs, i.e. they reduce the clock frequency to avoid timing violations due to aging in the foreseen lifetime. Recent work has shown, that the necessary frequency reduction has to be more than 10% for a lifetime of 3 years for a 32 nm technology [12]. Additionally, in future the timing margins will increase, since aging effects

are increasing with smaller feature sizes [7], [12]. Hence, adding guardbands requires an overdesign, i.e. to reach the performance goals with guardbands, the product has to achieve a higher performance without guardbands than required. This means higher costs and can also go hand in hand with a higher power consumption due to conservative supply voltages.

Hence, new approaches are necessary to take further advantage of scaled technology nodes in terms of power consumption, performance, costs and in particular reliability. To combine high performance with low power consumption one of the most common techniques is *Dynamic Voltage and Frequency Scaling* (DVFS). In case a microprocessor is executing only a light workload or no workload at all, frequency and voltage are reduced to save energy. Furthermore, DVFS techniques are also often applied to reduce heat [2], [14].

Voltage and frequency scaling also strongly affects transistor aging. Hence, some previous work uses coarse-grained voltage scaling techniques (i.e. time between adaptations is in the order of days or more) [4], [13], [16], [19] to address transistor aging. However, most of them just increase supply voltage stepwise so that frequency and hence performance can be kept on the original level. Also some of these techniques increase frequency in early life (possible due to guardbands) to increase performance. However, after a certain operating time, frequency and performance are back on the original level (i.e. specified frequency). In other words, these techniques try to address transistor aging after it has been accumulated beyond a certain level, i.e. in a *reactive* manner. Furthermore, these coarse-grained techniques are *static* and cannot immediately react on dynamic events due to changing environmental conditions, performance or power demands. This makes them unfeasible for some embedded systems e.g. smartphones, where user/application needs can change in the order of seconds or less. *Hence, a proactive, fine-grained and highly dynamic DVFS approach to extend lifetime, reduce power and heat, while maintaining the required performance is still missing.*

In this paper, we present such a solution, in which voltage and frequency can be adjusted several times per second, as it is in real devices (e.g. in some linux systems the sampling frequency is ~ 1 kHz). After each time frame, an *expert system* determines the voltage-frequency-configuration for the next time period based on the current and former system states, the predicted system behavior and user specific constraints. The results obtained using SPEC2000 benchmarks show that

the *Mean Time to Failure* (MTTF) can be extended by 63% up to 5x, while the performance impact is negligible.

The rest of this paper is organized as follows. In Section II the considered aging phenomena are introduced. The framework to analyze aging and fine-grained DVFS is presented in Section III. The expert system and the chosen DVFS policy are provided in Section IV, followed by the experimental results in Section V. Finally, we conclude in Section VI.

II. PRELIMINARIES ON TRANSISTOR AGING

1) NBTI: The NBTI effect consists of two different phases. When a logic '0' is applied at the gate of a PMOS transistor ($V_{gs} = -V_{dd}$), this transistor is under *stress*. During that phase, traps are generated in the interface between gate oxide and channel, which increases $|V_{th}|$. In contrast, when a logic '1' is applied at the gate of the same transistor ($V_{gs} = 0$), some traps are filled, which leads to a decreasing $|V_{th}|$ (*recovery phase*). However, the initial shift cannot be entirely compensated leading to an overall V_{th} drift over time. Thereby, the shift depends on several different aspects, e.g. temperature T , supply voltage V_{dd} and the ratio between the time a transistor is under stress and total time (duty cycle δ). In [20] the following analytical model for the NBTI process is derived:

$$\Delta V_{th}(\delta, T, V_{dd}, t) \leq A_N \cdot u(V_{dd}) \cdot \frac{(v(T) \cdot \delta(t) \cdot t_m)^n}{w(\delta, T, t)^{2n}} \quad (1)$$

with

$$\begin{aligned} u &= (V_{dd} - V_{th}) \cdot \exp((V_{dd} - V_{th})/E_0) \\ v &= \xi_4 \cdot \exp(-E_a/kT) \\ w &= 1 - \left(1 - \frac{\xi_1 + \sqrt{\xi_3 \cdot v(T) \cdot (1 - \delta(t)) \cdot t_m}}{\xi_2 + \sqrt{v(T) \cdot t}} \right)^{\frac{1}{2n}} \end{aligned}$$

where A_N , n , E_0 and ξ_i are technology dependent constants, E_a is the activation energy (positive), k is the Boltzmann constant and t_m is the period between two measurements.

Beside NBTI, *Positive Bias Temperature Instability* (PBTI) is another emerging reliability problem due to the introduction of high-k gate oxides. However, the impact of PBTI on the behavior of NMOS transistors is very similar to the NBTI effect on PMOS transistors and hence our framework can be easily extended to consider PBTI as well.

2) HCI: HCI is mainly affecting NMOS transistors, where accelerated electrons inside the channel can collide with the gate oxide interface and thereby create electron-hole pairs. Thus, free electrons get trapped in the gate oxide layer, which leads to an increasing V_{th} . Since the "hot" energetic electrons are generated when the NMOS transistor is making a transition, the voltage shift is very sensitive to the number of transitions. The authors in [18] have shown, that the relationship between number of transitions and voltage shift is sublinear. Hence, the voltage shift has a sublinear dependency on the clock frequency f , runtime t and the activity factor α , which is the ratio of the cycles the transistor is doing transitions and the total amount of cycles. Furthermore, the HCI effect has an exponential dependency on temperature [8].

Putting all the dependencies together leads to the following model, which describes the HCI effect:

$$\Delta V_{th}(\alpha, T, V_{dd}, t) = A_H \cdot u(V_{dd}) \cdot v(T) \cdot \sqrt{\alpha \cdot f \cdot t} \quad (2)$$

with

$$u(V_{dd}) = \exp((V_{dd} - V_{th})/E_1), \quad v(T) = \exp(-E_a/kT)$$

A_H and E_1 are technology dependent constants and the activation energy E_a is again considered to be positive. Please note that the temperature relation for technologies using feature sizes larger than 100 nm is reversed [8].

3) State of the Art: Beside the voltage and frequency scaling techniques mentioned in the introduction there are several other (orthogonal) techniques that can reduce wearout. To name just a few ones, special NBTI-resilient processors are proposed in [1] and specific input patterns at the primary inputs of a subcircuit can mitigate the NBTI effect during idle periods [21]. Furthermore, power gating [9], adaptive body biasing [19] and enhanced instruction as well as application scheduling techniques [17], [19] have been proposed to mitigate the effect of NBTI and HCI.

III. SIMULATION FRAMEWORK

The goal of this work is to extend lifetime, reduce power and heat, while maintaining the required performance using fine-grained proactive dynamic voltage and frequency scaling. Therefore a suitable microarchitectural framework containing accurate models for power, temperature and aging is necessary.

As just described, NBTI and HCI induced aging strongly depends on "usage" and temperature. We have chosen the cycle-accurate microarchitectural performance simulator gem5 [6] and incorporated models for power, temperature and aging, so that these parameters can be observed during the execution of typical applications. For the power model (both dynamic and static power) we use a customized version of *McPAT* [15] and the temperature model is based on *HotSpot* [11]. The temperature information in conjunction with information about the usage/activity of different microarchitectural blocks is used by our microarchitectural aging models for NBTI and HCI. These are based on the transistor-level models explained in Section II. For each microarchitectural block (e.g. ALU, instruction decoder, etc.) it is assumed that all transistors behave similar (i.e. have the same temperature, aging rates, etc.). Hence, a representative transistor can be chosen, for which the current and future V_{th} shift is estimated. Based on that the delay increase using an alpha power law can be calculated and by that means the current and future delay change of the entire block can be determined. We have validated this representative transistor model with accurate circuit-level implementation and the results show a high accuracy (< 3% difference).

We have integrated all models in one common framework, which enables a runtime analysis of power, temperature and wearout every X cycles (online), which makes the investigation of dynamic runtime adaptation techniques possible. Thereby X can be chosen freely, depending on the needs of user/system. However, in order to achieve a highly dynamic

Sampling Period [ms]	500	100	10	1	0.1	0.01	0.005
Performance [%]	100	99	98	98	97	94	91
worst MTTF [%]	100	100	135	163	164	164	164
average MTTF [%]	100	225	246	290	290	290	290

TABLE I

EFFECT OF DIFFERENT SAMPLING PERIODS ON PERFORMANCE AND LIFETIME FOR THE USED SPEC2000 BENCHMARKS

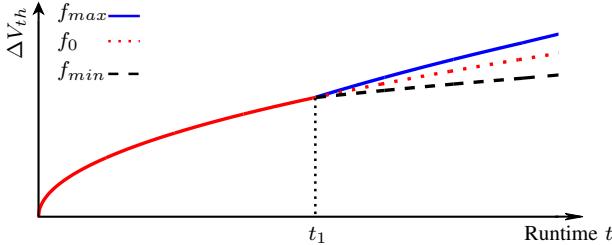
system, the time period between two analysis steps should be in the order of seconds or even less. Most previous approaches that applied aging-aware DVFS use much longer periods (i.e. in the order of days or more), which can lead to huge performance impacts (up to a factor of f_{max}/f_{min}) or can not yield good aging mitigation results (see Table I). In addition, the integration of all models allows a close interaction between them, for example leakage power can be estimated based on the current temperature. Further details on our simulation framework can be found in [17].

To accurately model the impact of fine-grained DVFS on power, temperature and wearout we have also added the support for this technique to the models and to the simulator itself. In our implementation a frequency/voltage change leads to a pipeline stall of a few μs , which is typical for modern processors that support DVFS via digital PLLs [3]. Hence, the performance can be (negatively) affected, which makes it unreasonable to adjust voltage or frequency in the order of μs . In our case a 1 ms sampling interval yields the best compromise between dynamics of the system and performance, which is illustrated in Table I. This underlines that only fine-grained techniques can combine long lifetime and high performance.

Please note that the aging models for NBTI and HCI presented in Section II are only valid if the parameters such as supply voltage or frequency are constant over time. However, due to the dynamic scaling of voltage and frequency during runtime these parameters will change. Hence, the V_{th} shift has to be calculated stepwise. Every time voltage or frequency changes, the aging rates have to be adjusted according to the new parameters. Since $\Delta V_{th}(t)$ is continuous, the parameter change at time t_1 will not lead to a jump of ΔV_{th} , but to a continuous change, i.e.:

$$\Delta V_{th}(t_1, T_1, V_{dd,1}, f_1) = A' \cdot \Delta V_{th}(t_1, T_2, V_{dd,2}, f_2)$$

Using this equation A' can be derived and by that means the V_{th} shift for the following time frame in which the parameters are again constant using the Equations (1) and (2) can be determined. This process is also illustrated in Figure 1 for an example, in which the frequency is changed at time t_1 .

Fig. 1. Effect of parameter change on Wearout due to NBTI/HCI at time t_1

IV. DYNAMIC RUNTIME ADAPTION METHODOLOGY

In this section our dynamic runtime adaptation methodology is presented. First, the monitoring part is introduced, followed by the presentation of the expert system that is used to determine the next (i.e. new) runtime configuration.

A. Runtime Monitoring

Many embedded systems have tight constraints regarding power, operating temperature and performance. Hence, if dynamic runtime adaptation techniques are used, it is important to monitor these three aspects during runtime and make adaptation decisions dependent on the current state of the system, the history of system states, the predicted system behavior in future and user inputs. Furthermore, for our purpose the wearout status (lifetime) due to transistor aging needs to be monitored. While this is done in our case using the framework with integrated models for power, temperature and aging as detailed in Section III, in real processors information about power and temperature can be obtained by on-chip sensors. The performance information such as IPC (Instructions per Cycle), activity of execution units, etc. of a real processor is delivered by special performance counters and aging can be either measured using special path delay sensors [10] or estimated using analytical approaches similar to our models using other on-chip sensors (power, temperature, performance) if available (otherwise they can be modeled/estimated as well). In our framework, power consumption, temperature and wearout are estimated for each microarchitectural block.

B. Expert System

All monitored data is sent to an expert system similar to the one depicted in Figure 2. It is responsible for making decisions regarding the next runtime configuration and is split into two parts. First, there is one *local* expert for each sensor group. These experts preselect the monitored data, to reduce the data amount that has to be analyzed later on. Nevertheless, the local experts can also initiate an immediate adaptation of the system, for example if critical values, e.g. temperature, are detected. Further details on the preselection process of the local experts and the critical values are given in Table II.

Furthermore, there is one *global* expert used to find the best fitting runtime configuration in case no local expert is detecting a critical status. The inputs of this expert are the

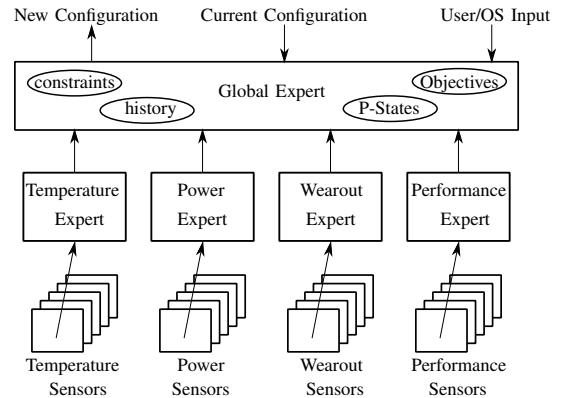


Fig. 2. Organization of the expert system for dynamic runtime adaptation

Local Expert	Preselection Function	Critical if
Temperature	$T_{max} = \max T_i$	$T_{max} \geq 100^\circ\text{C}$
Power	$P_{total} = \sum P_i$	$P_{sum} \geq 25 \text{ Watt}$
Performance	no preselection	none
Wearout	$MTTF = \min MTTF_i$	$MTTF \leq 11 \text{ Days}$

TABLE II
FUNCTIONALITY OF THE LOCAL EXPERTS

current runtime configuration, the preselected data from the local experts, history of the most recent system states (wearout, temperature, etc.) and various objectives for which the optimal runtime configuration for the next time frame has to be found. The process to find this configuration is explained in detail in the following subsection. Furthermore, the global expert should be able to get input from the user, for example to choose special objectives which fit the needs of the user most. By this means both self-adaptation and user-controlled adaptation are possible.

C. Dynamic Voltage and Frequency Scaling Policy

The global expert contains multi-dimensional objectives for runtime adaptation, since simple one-dimensional objectives, such as $\max(\text{lifetime})$, $\max(\text{performance})$, $\min(\text{power})$ or $\min(\text{temperature})$ are not the best choice for embedded systems. For example, neglecting the performance requirements, when maximizing the lifetime would lead to the choice of the lowest possible *P-State* (i.e. combination of lowest frequency and lowest supply voltage), which in turn would lead to a very low performance. Hence, the global expert has to contain complex objectives to optimize the system configuration for a certain goal, but with respect to several constraints. A very important objective for embedded devices is to maximize the lifetime, while the required performance is still ensured (as well as power and temperature constraints). In the following we will explain, how this objective can be achieved using fine-grained and proactive DVFS. For this reason, we focus on the i^{th} analysis step, i.e. what analysis and decisions are made between the DVFS interval $[t_{i-1}, t_i]$ and the next DVFS interval $[t_i, t_{i+1}]$. The length of such an interval is 1 ms, due to the explanations given in Section III about the optimal sampling interval (see also Table I).

1. Analyze recent trend of Temperature, Power, Wearout:

The first step of the DVFS policy is to analyze the history of the last n system states obtained from the last n DVFS intervals. Based on the history and the current state, *linear trend functions (LTF)* are build for wearout (i.e. MTTF), temperature and power. An *LTF* is basically a linear regression of n data points using the least square fitting method. With the obtained trend functions the future values for power P^i , temperature T^i and wearout W^i (i.e. MTTF i) after the next DVFS interval are extrapolated:

$$\begin{aligned} T^i &= LTF_T^i(T^{i-1}, \dots, T^{i-n}), \\ P^i &= LTF_P^i(P^{i-1}, \dots, P^{i-n}), \\ W^i &= LTF_W^i(W^{i-1}, \dots, W^{i-n}). \end{aligned}$$

If one of these extrapolated values is considered to be critical, the P-State in the next DVFS interval $[t_i, t_{i+1}]$ will be the next smallest P-State than the one used in the last DVFS interval

$[t_{i-1}, t_i]$. In other words, voltage and frequency are scaled one level down to reduce wearout, temperature, or power.

The number of investigated system states n has a huge influence on the extrapolated values and by that means on the decision making process. If n is too small, individual events have a too much influence on the trend. If in contrast n is too large, outdated system states still affect the trend function. In our case, it has been shown that $n = 10$ is a good compromise between proactive reliability enhancements and performance.

2.1. Suggest new P-State based on Performance:

If the trend evaluation does not indicate problems, the next step is to find all possible P-States, which guarantee that the performance constraints are fulfilled. Therefore, the global expert accesses various load/performance indicators for the last DVFS interval $[t_{i-1}, t_i]$, such as IPC^{i-1} , number of executed instructions in different execution units (activity $A_{EU_j}^{i-1}$), current frequency F^{i-1} , etc. Based on these parameters all P-States are suggested for a usage in the next DVFS interval, that satisfy the performance constraints, i.e.:

$$\{F_{sug,k}^i\} = \{F | F \geq F_{base} = f(IPC^{i-1}, A_{EU_j}^{i-1}, \dots, F^{i-1})\} \quad (3)$$

where F_{base} is the minimum frequency fulfilling the performance requirements. For each suggested frequency $F_{sug,k}^i$ the supply voltage is given using a fixed one-to-one-mapping, i.e. for each frequency there is a fixed supply voltage.

Please note that the function f has not only a huge impact on performance, but also on wearout, temperature and power consumption. The function f reflects the “aggressiveness” with which the frequency/voltage is scaled up or down. From the wearout perspective, a very aggressive downscaling is desirable, while from the performance point of view an aggressive upscaling is needed. Hence, the function f (and with it the “aggressiveness” of DVFS), which is used to estimate the frequency needed for the next time period, can be used to optimize the DVFS behavior in various ways, i.e. to make the DVFS policy more aging-aware or more performance-aware. Since the global expert is capable of taking inputs from the user or operating system, the function f can be changed during runtime, depending on the current needs.

In this paper the function f is always a polynomial. The simplest case is thereby a linear function with the structure $f(P) = a \cdot P + b$, where P is a vector containing the above mentioned load indicators such as IPC^{i-1} or $A_{EU_j}^{i-1}$. The parameters a and b are set in such a way, that $f(P)$ returns the maximum allowed frequency in case the maximum performance is required and that the minimum allowed frequency is returned if the minimum performance is sufficient. For example, if P just contains the activity $A \in [0, 1]$ of the first execution unit, the maximum frequency is 3 GHz and the minimum is 1 GHz, f has the following form:

$$f(P) = f(A) = (3\text{GHz} - 1\text{GHz}) \cdot A + 1\text{GHz}.$$

2.2. Select new P-State based on Temp., Power, Wearout:

However, the aforementioned way to select the P-State for the next time frame is just one part. While the first step is used to suggest a new P-State according to the performance

needs, the second part takes care of the other parameters such as power P , temperature T and wearout W (i.e. MTTF). For each P-State (F_k, V_k) , power, temperature and wearout after the “next” DVFS interval i is estimated, based on the current values and the chosen P-State:

$$\begin{aligned} P_{(F_k, V_k)}^i &= g_P(P^{i-1}, T^{i-1}, F_k, V_k) \quad \forall(F_k, V_k), \\ T_{(F_k, V_k)}^i &= g_T(T^{i-1}, P^i, F_k, V_k) \quad \forall(F_k, V_k), \\ W_{(F_k, V_k)}^i &= g_W(W^{i-1}, T^i, F_k, V_k) \quad \forall(F_k, V_k). \end{aligned}$$

The functions g_P , g_T , g_W are basically power, temperature, and wearout models based on various voltage and frequency combinations. Note that the temperature model requires power information, and the wearout model is also dependent on temperature. Afterwards all P-States are removed from $\{(F_{sug}^i, V_{sug}^i)\}$, that lead to critical values of $P_{(F_k, V_k)}^i$, $T_{(F_k, V_k)}^i$ or $W_{(F_k, V_k)}^i$. If the set is empty afterwards, the largest, non-critical pair (F_k, V_k) is chosen as next P-State. Otherwise the smallest pair out of the set is chosen, since this P-State will cause the lowest wearout rates of all in the set.

Summing it up, the three steps to determine the P-State for the next time frame allow a proactive lifetime extension (due to trend analysis, the function f , and the prediction in 2.2) and, as we will show in Section V, the performance will be maintained. Please note that in this work we used linear trend functions and a one-to-one-mapping between frequency and voltage. However, both approaches can be extended to support nonlinear trend functions and various supply voltages per frequency grade depending on wearout status, respectively.

Please note that there is also a chance that the prediction fails, i.e. a wrong behavior is predicted. This can lead to a P-State lowering (e.g. critical value is predicted based on the trend analysis) or a frequency increase (e.g. prediction indicates that more performance is necessary). In both cases, the very fine-grained approach allows almost immediate corrections (i.e. P-State adaptation), in case the real behavior differs from the predicted one. However, this phenomenon can lead to a negative performance impact and it can also lower the benefits in terms of lifetime extension. Nevertheless, the chosen techniques still provide very good results as shown in the following section.

D. Possible implementation of the Expert System

In modern systems the DVFS implementation is split into two parts, which can be reused by our proposed expert system. The performance monitors, temperature and power sensors are implemented in hardware as well as the functionality to handle critical temperature or power states. Our local experts can use these functionalities to detect and treat critical system states. In contrast, delay sensors are not as widespread. If they are implemented in hardware as well, they will increase the transistor count and die size. However, these sensors are very small. In case eight of these are build into an ARM Cortex v8 the die size would increase by less than 1% [10]. However, in case transistor budget is limited the local experts can be also implemented in software as a part of the operating system.

Processor	Single-core@3 GHz, out-of-order, 4-issue
L1-Cache / L2-Cache	64 KB, 3 cyc latency / 2 MB, 15 cyc latency
Expected wearout	MTTF = 2.5 years
DVFS parameters	DVFS interval = 1 ms, stall latency = 1 μ s
SPEC2000 benchmarks	applu, bzip2, equake, gcc, gzip, lucas, mesa, parser, twolf, wupwise
P-States (F-V-States)	1.0 GHz/0.6 V, 1.5 GHz/0.7 V, 2.0 GHz/0.8 V, 2.5 GHz/0.9 V, 3.0 GHz/1.0 V

TABLE III
CONFIGURATION DETAILS FOR THE EXPERIMENTS

The more sophisticated analysis and decision making parts of current DVFS solutions are embedded into the kernel of the operating system. Our proposed global expert can use these parts. However, the available routines need to be extended by the proactive, wearout-aware parts. A negative performance impact is thereby not to be expected, since the software routines do not use computationally expensive operations.

V. RESULTS

A. Evaluation Setup

The evaluated 32 nm processor runs at 3 GHz and has one super-scalar, out-of-order core. Further details of the processor configuration can be found in Table III. Furthermore, a delay degradation of 10% in 30 months ($T = 100$ °C) due to HCI and NBTI is assumed [5], [12]. Since the critical delay degradation is set to be 10%, MTTF under this conditions is 2.5 years. Please note that under typical applications the temperature never reaches 100 °C. Hence, MTTF is longer even if no aging mitigation techniques are applied. The following evaluations are based on the execution of 10^9 instructions of various SPEC2000 benchmarks. Thereby the simulations do not include the initialization phase of each benchmark, which is executed but not included in the measurements.

B. Results

We have investigated different f -functions (polynomials according to Equation (3)) to find out, how aggressive the up/down-scaling of frequency/voltage should be. The results summarized over all executed applications can be found in Table IV. As it can be seen, the difference in terms of performance and aging mitigation of the presented techniques is huge. While the choice of a linear f -function leads to a performance loss of 26% compared to the non-DVFS case, it can extend lifetime (MTTF) by more than 3x in the worst case. This is due to the fact that the P-States are often strongly reduced. In contrast, in case a polynomial of degree 5 is used as f -function (i.e. hexa policy) the downscaling of frequency/voltage is much more conservative and hence the performance impact is minimized. However, the lifetime in the worst case is only extended by 4%. If in addition the proactive features such as trend analysis are enabled, the same policy can achieve a lifetime prolongation of 63%, while the performance impact remains negligible. Compared to the standard hexa policy the huge lifetime improvements are due to the proactive, slight P-State adaptations that avoid critical situations in advance and hence no strong “emergency” adaptations have to be applied (which reduce performance but do not extend lifetime a lot).

Objective	no DVFS			linear f		cubic f		hexa f		hexa f & trend analysis	
	@3 GHz	@2.5 GHz		0.86	26%	0.75	10%	0.70	2%	0.70	2% (overhead)
Avg. Runtime [s]	0.68	0.72	6 %	0.86	26%	0.75	10%	0.70	2%	0.70	2% (overhead)
Max. Temperature [$^{\circ}$ C]	93.96	81.18	14%	81.11	14%	93.82	0%	93.86	0%	89.63	5% (improvement)
Total Power [Ws]	14.24	12.94	26%	7.38	48%	11.03	23%	12.35	13%	11.23	21% (improvement)
worst MTTF [years]	2.75	5.44	98%	9.07	230%	3.06	11%	2.78	1%	4.49	63% (improvement)
avg. MTTF [years]	2.77	7.62	175%	39.00	1408%	7.87	284%	6.95	250%	8.00	189% (improvement)

TABLE IV

EFFECT OF DIFFERENT DVFS POLICIES ON WEAROUT (WORST MTTF OVER ALL BENCHMARKS), POWER (AVG. OVER ALL BENCHMARKS), TEMPERATURE (MAX. OVER ALL BENCHMARKS) AND RUNTIME/PERFORMANCE (AVG. OVER ALL BENCHMARKS)

Furthermore, we evaluated a static approach with 2.5 GHz, which can also improve lifetime a lot. However, the average performance loss is 6% (worst case: 11%). This underlines one major disadvantage of static techniques: for some applications they are suitable, for others they are not. “Static DVFS”, where an application is either executed with 2.5 GHz or 3 GHz will minimize this performance penalty, however MTTF will also be heavily reduced. In contrast our proposed dynamic approach has at most a performance loss of 3%.

Since scaling of voltage/frequency and its efficiency strongly depends on the executed workload, the improvements can vary a lot. In Table V the results for several applications for the proactive “hexa policy” are given, in which the MTTF ranges between 4.49 and 14.75 years, i.e. the chosen DVFS policy can extend lifetime by up to 5x. Furthermore, it is important to note that for different applications and different policies the dominance of NBTI and HCI on wearout are different. Hence, neglecting one of these effects can lead to wrong predictions and hence adaptations.

VI. CONCLUSION

Embedded microprocessors at nano-scale are exposed to various reliability issues, which include a more rapid aging of all components. In this work we have presented a *fine-grained, proactive dynamic runtime adaptation approach* using voltage and frequency scaling, that addresses transistor aging due to NBTI and HCI. Using our technique it is possible to increase lifetime in a range between 63% and 5x, while performance is in average only reduced by 2%. Also the power and temperature constraints are met. Hence, our fine-grained approach combines high performance and long lifetime.

REFERENCES

- [1] J. Abella *et al.*, “Penelope: The NBTI-Aware Processor,” in *Proc. of the Int'l Symp. on Microarchitecture*, Dec. 2007, pp. 85–96.
- [2] M. Bao *et al.*, “On-line thermal aware dynamic voltage scaling for energy optimization with frequency/temperature dependency consideration,” in *Proc. of the Design Automation Conf.*, 2009, pp. 490–495.
- [3] A. Bashir *et al.*, “Fast Lock Scheme for Phase-Locked Loops,” in *Proc. of the Custom Integrated Circuits Conf.*, Sep. 2009, pp. 319–322.
- [4] M. Basoglu *et al.*, “NBTI-Aware DVFS: A New Approach to Saving Energy and Increasing Processor Lifetime,” in *Proc. of the Int'l Symp. on Low Power Electronics and Design*. ACM, Aug. 2010, pp. 253–258.
- [5] K. Bernstein *et al.*, “High-performance CMOS variability in the 65-nm regime and beyond,” *IBM Journal of Research and Development - Advanced silicon technology*, vol. 50, pp. 433–449, July 2006.
- [6] N. L. Binkert *et al.*, “The M5 Simulator: Modeling Networked Systems,” *IEEE Micro*, vol. 26, no. 4, pp. 52–60, July 2006.
- [7] S. Borkar, “Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation,” *IEEE Micro*, vol. 25, no. 6, pp. 10–16, Nov.-Dec. 2005.
- [8] A. Bravaix *et al.*, “Hot-Carrier Acceleration Factors for Low Power Management in DC-AC stressed 40nm NMOS node at High Temperature,” in *Int'l Reliability Physics Symposium*, April 2009, pp. 531–548.
- [9] A. Calimera *et al.*, “NBTI-Aware Power Gating for Concurrent Leakage and Aging Optimization,” in *Proc. of the Int'l Symp. on Low Power Electronics and Design*. ACM, Aug. 2009, pp. 127–132.
- [10] A. Drake *et al.*, “A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor,” in *IEEE Int'l Solid-State Circuits Conf. - Digest of Technical Papers*, Feb. 2007, pp. 398–399.
- [11] W. Huang *et al.*, “HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design,” *IEEE Trans. on VLSI Systems*, vol. 14, no. 5, pp. 501–513, May 2006.
- [12] K. Kang *et al.*, “Estimation of statistical variation in temporal NBTI degradation and its impact on lifetime circuit performance,” in *Proc. of the Int'l Conf. on Computer-Aided Design*, Nov. 2007, pp. 730–734.
- [13] O. Khan *et al.*, “A Self-Adaptive System Architecture to Address Transistor Aging,” in *Proc. of the Conf. on Design, Automation and Test in Europe*. Euro. Design and Automation Ass., 2009, pp. 81–86.
- [14] J. Lee *et al.*, “Predictive Temperature-Aware DVFS,” *IEEE Trans. Comput.*, vol. 59, pp. 127–133, Jan. 2010.
- [15] S. Li *et al.*, “McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures,” in *Proc. of the Int'l Symp. on Microarchitecture*. ACM, Dec. 2009, pp. 469–480.
- [16] E. Mintarno *et al.*, “Self-Tuning for Maximized Lifetime Energy-Efficiency in the Presence of Circuit Aging,” *IEEE Trans. on Comp.-Aided Design of Int. Circuits and Systems*, vol. 30, no. 5, pp. 760–773, May 2011.
- [17] F. Oboril *et al.*, “ExtraTime: Modeling and Analysis of Wearout due to Transistor Aging at Microarchitecture-Level,” in *Proc. of Dependable Systems and Networks*, June 2012.
- [18] E. Takeda *et al.*, “New hot-carrier injection and device degradation in submicron MOSFETs,” *IEEE Proc. I, Solid-State and Electron Devices*, vol. 130, no. 3, pp. 144–150, June 1983.
- [19] A. Tiwari *et al.*, “Facelift: Hiding and slowing down aging in multicores,” in *Proc. of the Int'l Symp. on Microarchitecture*. IEEE Computer Society, Nov. 2008, pp. 129–140.
- [20] W. Wang *et al.*, “The Impact of NBTI Effect on Combinational Circuit: Modeling, Simulation, and Analysis,” *IEEE Trans. on VLSI Systems*, vol. 18, no. 2, pp. 173–183, Feb. 2010.
- [21] Y. Wang *et al.*, “On the efficiency of Input Vector Control to mitigate NBTI effects and leakage power,” in *Proc. of the Int'l Symp. on Quality of Electronic Design*. IEEE Computer Society, Mar. 2009, pp. 19–26.

	Policy	applu	bzip2	equake	gcc	gzip	lucas	mesa	parser	twolf	wupwise
MTTF-NBTI [years]	no DVFS (@3 GHz) hexa & trend	2.76 4.66	2.78 7.18	2.77 5.32	2.78 27.72	2.77 5.66	2.76 4.82	2.80 22.69	2.78 16.07	2.77 55.33	2.78 4.49
MTTF-HCI [years]	no DVFS (@3 GHz) hexa & trend	3.12 4.97	4.55 8.56	3.11 5.69	4.49 11.17	3.14 5.82	3.50 5.64	3.41 7.92	3.23 14.75	3.01 13.98	3.02 5.46
Runtime [sec.]	no DVFS (@3 GHz) hexa & trend	0.95 0.97	0.89 0.92	0.61 0.62	0.49 0.50	0.61 0.63	1.07 1.09	0.37 0.38	0.63 0.64	0.59 0.61	0.59 0.60

TABLE V
INFLUENCE OF APPLICATION ON THE EFFICIENCY OF DVFS