ExtraTime: A Framework for Exploration of Clock and Power Gating for BTI and HCI Aging Mitigation

Fabian Oboril, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, Email: fabian.oboril@kit.edu
Mehdi B. Tahoori, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, Email: mehdi.tahoori@kit.edu

Abstract

Bias Temperature Instability (BTI) and Hot Carrier Injection (HCI) are two major causes for transistor aging at nano-scale, leading to slower devices, more failures during runtime, and ultimately reduced lifetime. Typically these issues are handled by adding extra guardbands to the design, i. e. overdesign, which results in lower clock frequencies and hence, performance losses. Alternatively, efficient aging mitigation techniques can be used to relax such guardbands. In this paper we explore various clock and power gating techniques for BTI and HCI aging mitigation at microarchitecture-level for superscalar processors. This is done with the help of our aging-aware microarchitectural framework *ExtraTime*, which includes a cycle-accurate performance simulator together with microarchitectural models to estimate power consumption, temperature, and particularly aging. The simulation results show that using an aging-optimized combination of clock and power gating, aging (delay) of the execution units of a 32 nm superscalar microprocessor due to BTI can be reduced by 30 % while aging due to HCI is mitigated by 70 %. This is achieved with only 2 % reduction in performance (IPC). This gives one the possibility to either extend the lifetime by 3 times, or reduce the guardbands.

1 Introduction

Faster runtime degradation, also known as aging, is one major reliability issue of current and future nano-scale processors. Experiments [1] have thereby shown, that *Negative Bias Temperature Instability* (NBTI) [2] and *Hot Carrier Injection* (HCI) [3] are the most eminent causes for transistor aging. However, with the establishment of high-κ metal gates also *Positive Bias Temperature Instability* (PBTI) [4] becomes a serious reliability problem. All phenomena lead to a shift of the threshold voltage, which manifests in an increasing switching delay of the impaired transistors. this results in increasing path delays, which can lead to timing violations and finally to the death of the system.

Currently manufactures deal with these runtime degradation issues by adding safety margins, so called *guardbands* [5], to their designs. Instead of using the maximum achievable frequency at time $t = 0$ for the final product, they reduce the frequency to make sure that no timing violations due to aging will occur during a specific time. A major problem that goes along with these safety margins is the implied performance loss, that can easily exceed 10 % for a 32 nm technology for a 3 year lifetime [6]. To make it even worse, aging will speed up with future generations, leading to greater margins and thus higher performance losses.

Hence, other techniques are necessary to take further advantage of scaled technology nodes. The possible hardware approaches are thereby multifaceted ranging from delay sensors, that can detect critical timings [7], [5], [8] to processors [9] and memory cells [10] that are almost resilient against NBTI. Even at system and architecture-level, there are techniques to mitigate aging. Thereby, most work focuses on aging-aware job scheduling for multi-core processors or dynamic voltage and frequency scaling [11], [12] and often considers only NBTI or HCI.

Our work, deals also with microarchitectural aging mitigation techniques. However, our focus is not on aging at multi-core-level but instead at unit-level. Thereby, the goal is to find out, how aging of the execution units of a superscalar processor is influenced by BTI and HCI during lifetime and how it can be efficiently mitigated using the microarchitectural techniques clock and power gating. To investigate these questions our self-developed, microarchitectural framework called *ExtraTime* was used. ExtraTime is based on a cycle-accurate performance simulator with extensions to model power consumption and temperature. Moreover, ExtraTime includes aging models for BTI and HCI at microarchitecture-level, derived from transistor-level models. By this means it is possible to evaluate many critical design parameters like temperature, power consumption, performance and aging rates, while various programs are running on the processor. All this can be done in early design phases of a microprocessor enabling design space exploration not only for performance and power but also for aging.

The results obtained with ExtraTime for a 32 nm superscalar processor show that, aging of the execution units induced by BTI and HCI can be mitigated by 30 % respectively 70 % using clock gating together with "aging-optimized" power gating strategies. If a chip would fail because of an increased transistor delay of 10 %, these techniques can hence prolongate the lifetime by a factor of 3. Thereby the performance loss is just 2 %, which can be eliminated using some of the gained headroom.

This paper is organized as follows. In Section 2 the transistor-level models for HCI and BTI are introduced. The ExtraTime framework is presented in Secti-

on 3, followed by the description of the used microarchitectural aging models in Section 4. The proposed aging mitigation techniques are provided in Section 5. The corresponding results can be found in Section 6 and in Section 7 some related work is presented.

2 Aging Effects

2.1 Bias Temperature Instability

The Bias Temperature Instability (BTI) can affect both PMOS and NMOS transistors, using two different mechanisms, called Negative BTI (NBTI) for PMOS transistors and Positive BTI (PBTI) for NMOS transistors. The BTI effect consists thereby of two different phases. When a channel is formed in a transistor ($V_{gs} = -V_{dd}$ for PMOS and $V_{gs} = V_{dd}$ for NMOS), traps are generated in the interface between gate oxide and channel, which increases $|V_{th}|$. In contrast, when the same transistor is off ($V_{gs} = 0$), some traps are filled, which leads to a decreasing $|V_{th}|$. Hence, this period is called *recovery* phase, while the first one is called *stress* phase. However, during the recovery phase the V_{th} shift is not completely eliminated, leading to an overall drift over time.

In both cases (NBTI and PBTI) the amount of voltage shift depends on several different aspects, e.g. temperature T and the ratio between the time a transistor is under stress and total time (duty cycle δ).

In [2] an analytical model for the NBTI process is derived. With this model it is possible to make a long term prediction of the V_{th} shift for a couple of years. Thereby, the long term prediction model is of the form:

$$\Delta V_{th}(t) \leq A_{BTI} \left(\frac{\sqrt{\gamma \cdot \delta \cdot t_m}}{1 - \beta^2} \right)^{0.5} \quad (1)$$

with

$$\begin{aligned} \beta &= 1 - \frac{\xi_1 + \sqrt{0.5 \cdot \gamma \cdot (1-\delta) \cdot t_m}}{\xi_2 + \sqrt{\gamma \cdot t}} \\ \gamma &= \xi_3 \cdot exp(-E_a/kT) \end{aligned} \quad (2)$$

where A_{BTI} and ξ_i are technology dependent constants, E_a is the activation energy (positive) and t_m is the period between two measurements.

Please note that this long term degradation model does not describe the underlying physical processes (which is still debated), but rather the effect of these processes over a long period of time. Since the effect of NBTI and PBTI are very similar, we use this model as a basis for the microarchitectural aging models for both, NBTI and PBTI by replacing A_{BTI} with appropriate values for A_{NBTI} and A_{PBTI}, respectively.

2.2 Hot Carrier Injection

Hot Carrier Injection (HCI) is mainly affecting NMOS transistors, where accelerated electrons ("hot") 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, resulting in a V_{th} shift over time.

Since the "hot" energetic electrons are only generated when the NMOS transistor is making a transition [13], the voltage shift is directly proportional to the switching frequency f and the activity factor α, which is the ratio of the cycles in which the transistor is making a transition and the total number of cycles. Furthermore, the HCI effect has an exponential dependency on the temperature T [14] and also a relation to the total runtime [3]. Putting all these factors together leads to the following model for HCI effect:

$$\Delta V_{th}(t) = A_{HCI} \cdot \alpha \cdot f \cdot exp(E_a/2kT) \cdot \sqrt{t} \quad (3)$$

whereby A_{HCI} is a technology dependent constant and the activation energy E_a is again considered to be positive. Please note, that hence an increasing temperature leads to a smaller shift of the threshold voltage.

3 ExtraTime Framework

Our ExtraTime framework is based on the M5 performance simulator [15]. M5 includes a cycle-accurate model for a pipelined, out-of-order, superscalar architecture, which is based on the Alpha 21264 core [16]. In this platform all pipeline stages such as fetch, decode, etc. as well as branch predictors, queues, execution units and caches are modeled and can be configured (size, width, latency, etc.) independently. M5 supports multiple instruction sets like Alpha, ARM, x86 or Power and can model single- as well as multi-core processors. Furthermore it supports full-system simulation, which is used together with the cycle-accurate out-of-order-model for the experiments, that are presented in Section 6.

By executing several different workloads, M5 delivers detailed information regarding the overall performance of the modeled processor as well as the usage of different execution units such as ALUs or FPUs. However, such information is not sufficient to make an accurate aging estimation. Therefore, sophisticated temperature and power models are necessary to be integrated into the performance simulator. Such integration also helps to keep the simulation runtime and the amount of data communicated between sub-models as low as possible. By doing this, ExtraTime models a state-of-the-art microprocessor with on-chip sensors for temperature and power consumption. For the power model we use a customized version of *McPAT* [17] and the temperature model is based on *HotSpot* [18].

McPAT is a power and area modeling framework. It uses technology data (V_{dd}, V_{th}, feature size) based on the ITRS roadmap for technology nodes ranging from 90 nm to 16 nm. To calculate the static and dynamic power consumption the model uses the performance

data delivered by M5 in conjunction with architecture models for the basic components of the processor including out-of-order processor cores, shared caches and integrated memory controllers. In addition to power estimation, McPAT delivers the size for each processor component (microarchitectural block).

HotSpot on the other hand is an accurate thermal model. It is based on an equivalent circuit of thermal resistances and capacitances that correspond to microarchitectural blocks and essential aspects of the thermal package. Based on the area and power information provided by the power model, HotSpot calculates the temperature of each microarchitectural block.

This information is then passed to our microarchitectural aging models for BTI and HCI, which are based on the transistor-level models. The abstraction from transistor to architecture-level for our aging models is explained in the following Section 4. With the help of these models the aging status of the transistors inside each block can be estimated by using just microarchitectural information.

We have integrated all three models (power, temperature, aging) directly into the simulator, which is illustrated in Figure 1. Therefore, it is not only possible to calculate power consumption, temperature or aging at the end of simulation run (offline), but also every X cycles during a simulation run (online), where X can be chosen arbitrarily. This is a great advantage compared to the original, stand-alone McPAT and HotSpot solutions. Since these are offline, one has to run the entire benchmark on M5, save all necessary data and feed that after the simulation to McPAT and HotSpot, which leads to a high data and runtime overhead.

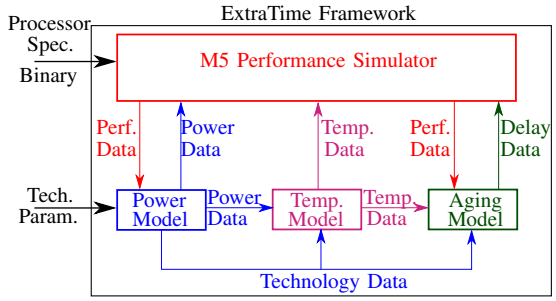

Figure 1 Data flow in the ExtraTime Framework

4 Microarchitectural Aging Models

Since ExtraTime is a microarchitectural framework, transistor-level aging models for BTI and HCI cannot be used in ExtraTime. Thus an important step is the abstraction of the aging models from transistor to microarchitecture-level.

Thereby, the first step is the definition of a useful metric to estimate aging. We use the *relative delay change* ($\Delta^{rel} d_B$) of a microarchitectural block B, which is defined as the fraction between the delay change (Δd_B) and the original delay d_B:

$$\Delta^{rel} d_B = \Delta d_B / d_B.$$

The relative delay change is a good choice as a metric, since we are interested in the lifetime improvements of different aging mitigation techniques and not in the absolute delay itself.

Assuming that all transistors inside one microarchitectural block age at the same rate, $\Delta^{rel} d_B$ of a block can be estimated by the relative delay change of the transistors $\Delta^{rel} d_{T_B}$, inside this block, i.e. $\Delta^{rel} d_B \approx \Delta^{rel} d_{T_B}$. Hence, $\Delta^{rel} d$ at block-level can be calculated by $\Delta^{rel} d$ at transistor-level. Moreover, as it will be explained in the following subsections, $\Delta^{rel} d$ at transistor-level can be estimated using only microarchitectural information and some known constants. Thus, $\Delta^{rel} d$ of each block can be calculated using only data which is known at microarchitecture-level.

Note that the assumption above, that all transistors in one microarchitectural block behave the same, is typical for microarchitectural models. For example the power model based on McPAT uses similar simplifications and hence the same level of accuracy. Of course this means an accuracy loss compared to work at transistor-level, but on the other hand microarchitectural mitigation techniques and real world workloads for microprocessors can be analyzed, which is almost impossible at transistor-level. By this means gaining new information about the aging process of microprocessors is possible. Furthermore this enables designers to tune the architecture of a microprocessor for aging reduction and not only for performance or power.

4.1 Bias Temperature Instability

As said before, the goal of this section is to estimate the relative delay change $\Delta^{rel} d$ at transistor-level due to BTI by using only microarchitectural information.

Since the effects of NBTI and PBTI are very similar, we use the same analytical transistor model described in equation (1) as a base for both. The transistor dependent temperature T and duty cycle δ cannot be obtained directly at microarchitecture-level. Hence, to be able to transfer the model to microarchitecture-level, some abstractions are necessary. Therefore we use again the assumption that all transistors inside one block behave similar, resulting in the same voltage shift for all transistors in this block. Thus, all transistors in a microarchitectural block have the same temperature (as the whole block) T_B, which can be gained from the temperature model of ExtraTime. Another abstraction is regarding the duty cycle δ. For this purpose the stress time $t_{stress,B}$ of a block B is defined as the time in which at least one transistor inside this block is under stress (i. e. the block is not power gated).

$$\begin{aligned} \#cyc_{stess,B} &= \#cyc_{total} - \#cyc_{pg,B} \\ \Rightarrow \quad \delta_B &= \frac{t_{stress,B}}{t_{total}} = 1 - \frac{\#cyc_{pg,B}}{\#cyc_{total}} \geq \delta_{T_i} \end{aligned}$$

This newly-defined microarchitectural duty cycle δ_B

of an entire block can be derived from parameters delivered by a performance simulator (total cycle count $\#cyc_{total}$, number of power gated cycles $\#cyc_{pg,B}$). It should be noted that, δ_B is greater than or equal to the duty cycle of each transistor δ_{T_i} inside block B.

Putting all these together in conjunction with the known constants leads to an estimation of the V_{th} shift due to BTI at microarchitecture-level:

$$\Delta V_{th}(t) \leq A_{BTI} \left(\frac{\sqrt{\gamma_B \cdot \delta_B \cdot t_m}}{1 - \beta_B^2} \right)^{0.5}, \quad (4)$$

whereby γ_B and β_B are defined as in the transistor-level model (see equation (2)), but using the microarchitectural values T_B and δ_B instead of the transistor-level ones (T and δ). A_{BTI} is set according to the assumed worst case conditions [6], i. e. $\delta_B = 1$, $T_B = 90°\text{C}$ and $\Delta^{rel} = 10\%$ after 3 years. Thereby, the same aging rate for NBTI and PBTI is used, which seems to be reasonable for technologies using high-κ dielectrics [4].

In order to obtain the delay change from the V_{th} shift, the delay model of a transistor is used:

$$d = \frac{V_{dd} L_{eff}}{\mu (V_{dd} - V_{th})^{1.3}}, \quad (5)$$

During the simulation run the actual ΔV_{th} is calculated and then used to determine with the help of equation (5) the current Δd and $\Delta^{rel} d$.

4.2 Hot Carrier Injection

The approach to transfer the transistor-level model for HCI in equation (3) to microarchitecture-level is quite similar. Again the problem is that the temperature T in the original model corresponds to the temperature of one transistor and the activity factor α is also transistor dependent. Hence, again the temperature of an entire microarchitectural block is used for all transistors inside this block, which corresponds again to the assumption that all transistors in one block behave the same. Since the activity factor of a transistor is the product of the activity factor of this transistor while the complete block B is active (effective activity factor α_e) and the activity factor α_B of the block, equation (3) can be written as follows:

$$\begin{aligned} \Delta V_{th}(t) &= \underbrace{A_{HCI} \cdot \alpha_e}_{=:\tilde{A}_{HCI}} \cdot \alpha_B \cdot f \cdot e^{E_a/2kT} \cdot \sqrt{t} \\ &= \tilde{A}_{HCI} \cdot \alpha_B \cdot f \cdot e^{E_a/2kT} \cdot \sqrt{t} \end{aligned}$$

Thereby a block is active, when at least one transistor inside this block is active (i. e. the block is not clock/power gated).

$$\begin{aligned} \#cyc_{active,B} &= \#cyc_{total} - \#cyc_{pg,B} - \#cyc_{cg,B} \\ \Rightarrow \quad \alpha_B &= 1 - \frac{\#cyc_{pg,B} - \#cyc_{cg,B}}{\#cyc_{total}} \end{aligned}$$

As the equations above illustrate, the activity factor α_B of a block, can thereby be calculated using only parameters delivered by the performance simulator (total cycle count $\#cyc_{total}$, number of clock gated cycles $\#cyc_{pg,B}$, number of power gated cycles $\#cyc_{cg,B}$).

Thus, the microarchitectural HCI model has the form:

$$\Delta V_{th}(t) = \tilde{A}_{HCI} \cdot \alpha_B \cdot f \cdot e^{E_a/2kT_B} \cdot \sqrt{t} \quad (6)$$

$\tilde{A}_{HCI} = A_{HCI} \cdot \alpha_{eff}$ is set in a way, that the delay increases by 10 % in 3 years under the same conditions that are used for BTI. Since the real activity factor of a transistor is included in $\tilde{A}_{HCI}$, by this means it is also included in the microarchitectural model for HCI.

Similar to BTI induced aging, equation (5) and (6) enable us to also calculate the $\Delta^{rel} d$ due to HCI during runtime to determine the actual aging status.

5 Techniques for Aging Mitigation

With the help of the features implemented in the ExtraTime framework, we can explore many different aging mitigation techniques. Here in this work clock and power gating are investigated.

Clock gating is a well known and efficient technique to reduce the power consumption of a microprocessor. By switching off the clock signal of a logic block, the latches inside this block are no longer toggling. Hence the overall switching activity can be reduced, leading to a lower power consumption [19]. Furthermore, clock gating can be used to mitigate the effects of BTI and HCI. By reducing the power consumption, the temperature of the clock gated block decreases and hence V_{th} shift due to BTI is reduced. Please note, that thereby the voltage drift due to HCI is increased. However, the V_{th} shift due to HCI happens only during transitions, which are reduced by clock gating (reduced α_B). Hence, to mitigate the HCI effect, it is important that the reduction of the activity factor compensates the trend induced by the decreasing temperature. Since the clock signal can be (de)activated every cycle without any delay, clock gating does not come along with performance losses, which makes it very attractive. For this reason, it is already applied in many state-of-the-art microprocessor in order to reduce the power consumption.

Power gating is nowadays not as widespread as clock gating. Currently it is mainly used in multi-core processors to switch off unused cores to reduce power consumption [20]. Similar to clock gating, power gating can be used for aging mitigation. Since it reduces the power consumption, it in turn lowers the temperatures which mitigates the BTI effect. Furthermore, the stress time of PMOS and NMOS transistors, and hence δ_B, is reduced since a power gated transistor is in recovery mode. In addition power gating also lowers the number of transitions and thereby α_B. By this means the BTI and HCI effect are mitigated. Thereby, the reduction of α_B needs to compensate the trend induced by lower temperatures in order to mitigate aging due to HCI. However, it takes some time to power down

Chip: Single-core @ 3 GHz	Expected wearout: $\Delta^{rel}d$ of 10 % in 3 years
Core: out-of-order, 4-issue, like Alpha 21264	T_{start} =57 °C
L1-Cache: 64 KByte, 2-way, 64 Byte line, 3 cyc latency	$V_{dd} = 1.0$ V, $V_{th} = 0.21$ V
L2-Cache: 2 MByte, 16-way, 64 Byte line, 15 cyc latency	$t_m = 100\ \mu s$
Execution Units: 2x ALUs, 2x FPUs with similar delay	Power gating: power down 7 ns, power up 3 ns

Table 1 Configuration details for the experiments

or up a processor block, whereby the time periods depend on the size and number of power-gate transistors as well as the size of the power-gated block. Recent work has shown that an ALU can have a wake up time of 3 ns to 10 ns [21]. Hence, the power gated block is not available for a certain amount of time, which can lead to performance losses. In this work a wakeup time of 7 ns for the execution units and a power down period of 3 ns is used. The time period t_{idle} after which a unit can be power gated and the minimum duration t_{dur} of power gating can be chosen freely. Please note, that the time a unit is power gated t_{sleep} can be much longer than t_{dur}. The effects of various power gating parameters and strategies on aging mitigation as well as power and performance impacts are detailed in Section 6.

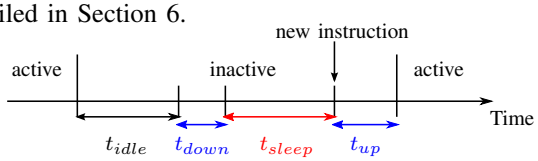

Figure 2 Time flow of power gating periods of an execution unit

6 Results

6.1 Evaluation Setup

The investigated configuration is based on a single-core processor running at 3 GHz. Since the focus of this work is on finding out, how aging affects the execution units, the evaluation of a single-core processor is sufficient. The modeled core is similar to the Alpha 21264 and accommodates 2 ALUs for logic operations and fix-point instructions and 2 FPUs for floating-point operations. Further details of the processor configuration can be found in Table 1. The fabrication technology is a 32 nm node with a supply voltage of 1.0 V and the initial temperature is 57 °C. Furthermore a delay degradation due to HCI and due to BTI of 10 % in 3 years ($\delta_B = 1$, $T_B = 90$°C) is assumed. According to [1], [6] this is reasonable.

Workload	Instructions	ALU [%]	FPU [%]
164.gzip	1422755075	54.4	0.0
176.gcc	1180138007	48.5	0.0
181.mcf	446565422	37.3	0.0
197.parser	931573758	44.4	1.5
256.bzip2	1066656694	45.0	0.0
300.twolf	952478447	40.1	2.8
168.wupwise	1416484719	48.4	22.4
171.swim	506189529	36.7	23.1
172.mgrid	1137745758	19.1	55.1
173.applu	1243856036	26.0	37.0
177.mesa	1538639402	42.6	16.5
183.equake	1267909605	45.7	19.5
189.lucas	1220409145	40.1	30.9

Table 2 Workloads and their instruction counts

The workloads are part of the SPEC2000 benchmark suite. Overall 6 integer and 7 floating point benchmarks, with a runtime of 0.5 seconds each, are used. Thereby, the runtime does not include the initialization phase of each benchmark, which is executed but not included in the measurements. The number of instructions for all workloads and their distribution among the different execution units are listed in Table 2.

6.2 Effect of Clock/Power Gating

Usually clock and power gating are used to reduce the power consumption of a microprocessor. However, as explained in Section 5, both techniques can be used also to mitigate aging induced by BTI and HCI. In the first set of experiments, a power gating strategy, where power gating can be activated after 40 idle cycles with a minimum power gating duration of 0 cycles, is used. The presented aging values are estimated using the aging model of ExtraTime for $t = 3$ years and represent the worst case, if not said otherwise.

Clock gating (CG) is extremely helpful to mitigate the HCI effect on NMOS transistors. The results illustrated in Figure 3 show that the worst $\Delta^{rel}d$ (relative delay change) over all execution units and all workloads is just 31 % of the original value without using any optimization techniques. However, the benefit for the entire processor is much smaller. Since clock gating has only a second order mitigation effect on BTI (reduced temperature), $\Delta^{rel}d$ of affected transistors is just reduced by 3 %. Hence, BTI induces faster aging ($\Delta^{rel}d_{HCI}$ = 3.8 % after 3 years, $\Delta^{rel}d_{BTI}$ = 8.9 % after 3 years).

To reduce the wearout due to BTI, power gating (PG) is much better suited. The used power gating strategy yields a reduction of $\Delta^{rel}d$ of more than 30 %. However, the benefit for HCI is much smaller and almost

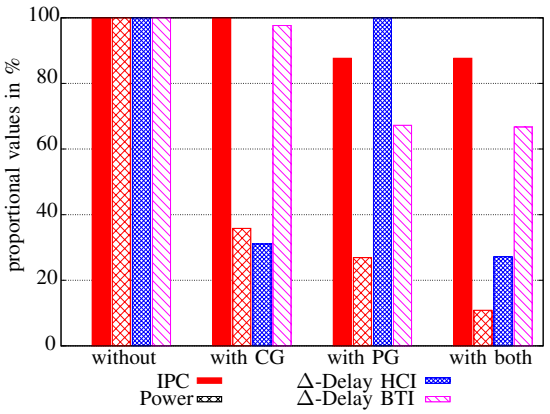

Figure 3 Effect of clock gating (CG) and power gating (PG) on performance (avg. IPC), power (avg. over all workloads, total over all ex. units) and aging (worst over all workloads and ex. units)

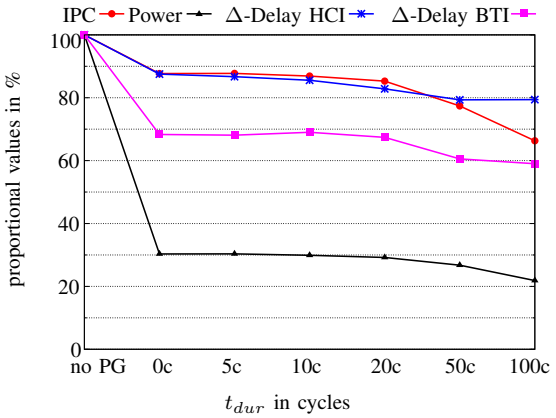

Figure 4 Effect of diff. PG policies for t_{dur} (with CG) on performance (avg. IPC), power (avg. over all workloads, total over all ex. units) and aging (worst over all workloads and ex. units)

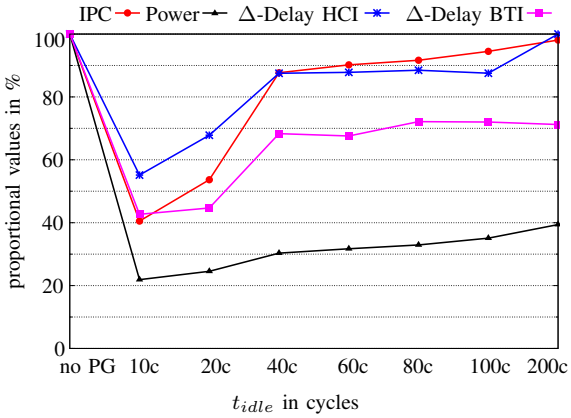

Figure 5 Effect of diff. PG policies for t_{idle} (with CG) on performance (avg. IPC), power (avg. over all workloads, total over all ex. units) and aging (worst over all workloads and ex. units)

negligible. This is due to the fact that clock gating can be activated much more often than power gating and thus reduces the activity factor more. In case of power gating, the decreased temperature (negative) compensates the decreased activity factor (positive), so that finally in this configuration no benefits for HCI can be obtained. In addition HCI is in that case worse than BTI ($\Delta^{rel}d_{HCI}$ = 12.1 % after 3 years, $\Delta^{rel}d_{BTI}$ = 6.1 % after 3 years), because only the latter can recover during power gating periods.

Hence, for an efficient mitigation of HCI and BTI, the combination of clock and power gating is the best choice. However, in this scenario BTI leads again to faster aging than HCI ($\Delta^{rel}d_{HCI}$ = 3.3 % after 3 years, $\Delta^{rel}d_{BTI}$ = 6.1 % after 3 years) due to the already mentioned fact, that power gating cannot be applied that often compared to clock gating and only the first one has a first order influence on BTI (see Section 5). This means, that even after more than 49 years $\Delta^{rel}d$ due to HCI does not exceed 10 % of the original value compared to less than 3 years without aging optimizations. $\Delta^{rel}d$ due to BTI passes the 10 % threshold after 19 years, compared to less than 5 years without aging optimizations.

Looking at the results from the perspective of power consumption, it shows that the average consumption is already heavily reduced by clock gating (just 35 % of the original consumption). Hence, power gating does not provide further noticeable power reduction. However, the results clearly indicate that power gating as an aging mitigation technique is very effective.

6.3 Optimized Power Gating Strategies

The average performance loss of the power gating strategy used in Section 6.2 and depicted in Figure 3 is about 13 %. To optimize power gating for aging mitigation with minimum performance overhead, two basic parameters can be modified. First, the (idle) time t_{idle} of a unit, before the unit can be power gated, is adjustable. The second parameter is the minimum power gating duration t_{dur}, i. e. the time a unit is at least power gated, every time power gating is activated for this unit. In this section the effects of different settings for both parameters are investigated, whereby clock gating is always applied.

In Figure 4 and Figure 5 the results for different settings are illustrated. Thereby $t_{idle} = 40$ cyc. for the different t_{dur} settings and $t_{dur} = 0$ cyc. for the different t_{idle} settings.

Since an increasing t_{idle} or a decreasing t_{dur} means, that power gating is activated less often, the performance is increasing. On the other hand this leads also to faster aging. However, the performance benefits are bigger than the aging deficits. Hence, the best choice by looking at aging and performance at the same time is a high t_{idle} and a small t_{dur}. The benefits for HCI are thereby indeed smaller than those for BTI, but since the latter leads to faster aging in all configurations, the advantages for these are more important.

In the "preferred" configuration ($t_{dur} = 200$ cyc., $t_{idle} = 0$ cyc.), where preferred is a combination of high performance and slow aging, power gating yields a reduction of 28 % for $\Delta^{rel}d$ due to BTI. Thereby the performance loss is only 2 % compared to a configuration without power gating. However, this configuration does not provide additional benefits to mitigate aging induced by HCI. In absolute terms this means, that after 3 years $\Delta^{rel}d$ due to BTI is 6.3 % and due to HCI 3.8 %. Therefore, the lifetime of the execution units is prolongated by more than 3 times compared to a solution without power gating, i. e. the critical $\Delta^{rel}d$ due to BTI exceeds 10 % after 17 years, while without power gating this takes less than 5 years, which can be seen also in Figure 6.

In case that aging induced by HCI is more critical, also the power gating configuration with $t_{dur} = 100$ cyc. might be an option. Furthermore one should note, that the average power consumption for all the shown strategies is very low, i. e. less than 0.5 Watt for all execution units together. However, the "preferred" configuration has the highest power consumption, which

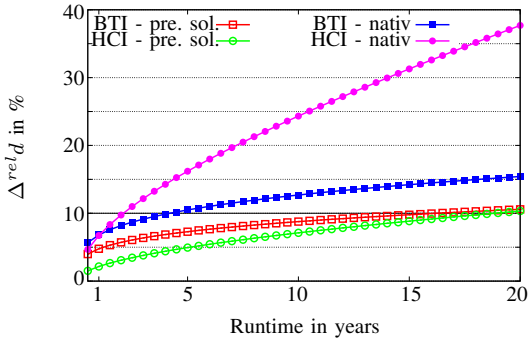

Figure 6 BTI and HCI induced aging during lifetime for the native technique (without clock/power gating) and the "preferred" solution

shows that optimizing for power consumption and optimizing for aging without impacting performance too much goes not always hand in hand.

6.4 Application Dependencies

Besides investigating different microarchitectural techniques to mitigate aging, ExtraTime can be also used as a tool to find out, how software can influence the aging behavior. In this work we use this capability to explore the dependencies between aging of the execution units and the executed application.

In Figure 7, the influence of several workloads on BTI and HCI induced aging is depicted, whereby the "preferred" aging mitigation technique of Section 6.3 is applied. As one can see, HCI and BTI do not always follow the same trend. Indeed if the BTI effect in application B is smaller than in application A, it has not to be the same for HCI. This is thereby mainly due to the chosen mitigation techniques. If the number of power gated cycles in application B is higher than in application A, the BTI effect can be smaller in B than in A. If at the same time the total amount of idle cycles (power gated or clock gated) in B is lower than in A, the HCI effect in B can be higher than in A. Hence the amount of idle time and its distribution during the execution of the workload has a high influence not only on BTI and HCI, but also on their ratio.

This effect can be seen also in Figure 8, in which the aging for the ALU and FPU induced by HCI and

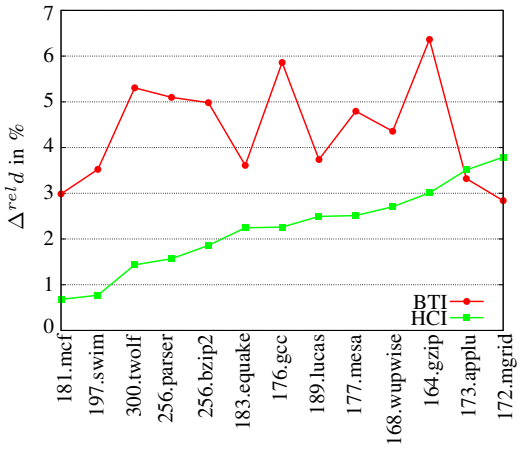

Figure 7 Worst $\Delta^{rel}d$ for BTI and HCI for diff. workloads

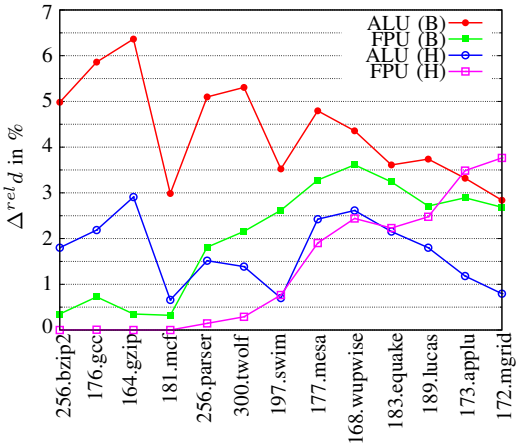

Figure 8 $\Delta^{rel}d$ for ALU and FPU (BTI and HCI) for diff. workloads

BTI is illustrated. Furthermore, the obtained results show another interesting phenomena. Although in the 173.applu and 172.mgrid benchmarks, the instruction ratio for the FPU is much higher (see Table 2) than for the ALU, BTI induced aging of the ALU is higher, than those of the FPU. This is due to the fact, that the ALU can be power gated less often. In contrast the expected aging behavior for those two applications is shown by HCI. Since HCI can be mitigated using clock gating, and clock gating can be applied every time the unit is idle, HCI induces slower aging of the ALU in this benchmarks than for the FPU. For all types of execution units both units, e. g. ALU0 and ALU1, age at the same rate, because of similar load and temperatures.

7 Related Work

In [22] power gating during idle periods is used to shut down unused transistors. If these are PMOS transistors, they are automatically in recovery mode, and hence the NBTI effect is reduced. We use basically the same idea in our work, but since our work makes use of a performance simulator, we can additionally consider the performance impacts of power gating.

A promising arcitecture-level approach called Facelift was presented in [11]. Facelift slows down aging induced by NBTI and HCI by aging-driven application scheduling for multi-core processors. In addition special dynamic voltage scaling techniques are explored. However, the focus of Facelift is on aging mitigation at core-level unlike our work, which concentrates on unit-level. In other words, these two techniques can be combined for further aging mitigation.

Another high-level framework is "New-Age" introduced in [23]. The framework combines different simulators (architectural and RTL) for different abstraction levels, in order to make an accurate estimation of NBTI induced aging. The work shows how different pipeline stages and different components of an ALU are effected by performance degradation due to NBTI. The authors found, that the path delay for some ALU

components increased by 20 % for a runtime of 10 years using a 32 nm technology. Furthermore, input vector control as a mitigation technique is investigated. However, aging analysis is done at RTL using netlists and input probabilities to determine the degradation of gates. In contrast our framework, does not focus on gates but units and hence does not need RTL information. Thus, ExtraTime can be used earlier in the design phase.

8 Conclusion & Future Work

Microprocessors at nano-scale are exposed to various reliability issues, which include a more rapid aging of all components. To model and analyze aging due to BTI and HCI at microarchitecture-level, this paper presented the microarchitectural framework *ExtraTime* and its integrated tool set containing a performance simulator combined with microarchitectural models for power consumption, temperature and aging. With this setup ExtraTime enables the user to investigate aging mitigation techniques not only at hardware-level, but also at software-level (application). Furthermore, ExtraTime can be used very early in the design phase of a microprocessor, enabling design space exploration for performance, power, temperature and aging.

Using this framework, we investigated various clock and power gating strategies for aging mitigation with minimal performance and power impacts.

The simulation results show that using clock gating together with "aging-optimized" power gating, aging of the execution units of a 32 nm superscalar microprocessor due to BTI can be reduced by 30 % while aging due to HCI is mitigated by 70 %. At the same time performance is only decreased about 2 %. Thus, lifetime of the execution units can be extended by a factor of 3, or instead some of the gained headroom can be used to increase the frequency to compensate this performance loss by relaxing the guardbands.

In our future work we will compare the accuracy of ExtraTime with respect to aging and power consumption with accurate RTL and transistor-level models of microprocessors supported by M5 such as IVM and thereby investigate further improvements.

9 References

[1] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, "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.

[2] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao, "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.

[3] E. Takeda, Y. Nakagome, H. Kume, and S. Asai, "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.

[4] S. Pae, M. Agostinelli, M. Brazier, R. Chau, G. Dewey, T. Ghani, M. Hattendorf, J. Hicks, J. Kavalieros, K. Kuhn, M. Kuhn, J. Maiz, M. Metz, K. Mistry, C. Prasad, S. Ramey, A. Roskowski, J. Sandford, C. Thomas, J. Thomas, C. Wiegand, and J. Wiedemer, "BTI reliability of 45 nm high-K + metal-gate process technology," in *IEEE Int'l Reliability Physics Symp. 2008*, May 2008, pp. 352–357.

[5] M. Agarwal, B. C. Paul, M. Zhang, and S. Mitra, "Circuit Failure Prediction and Its Application to Transistor Aging," in *Proc. of the VLSI Test Symp.*, May 2007, pp. 277–286.

[6] K. Kang, S. P. Park, K. Roy, and M. A. Alam, "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.

[7] J. Blome, S. Feng, S. Gupta, and S. Mahlke, "Self-calibrating Online Wearout Detection," in *Proc. of the Int'l Symp. on Microarchitecture*, Dec 2007, pp. 109–122.

[8] J. Keane, X. Wang, D. Persaud, and C. Kim, "An All-In-One Silicon Odometer for Separately Monitoring HCI, BTI, and TDDB," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 817–829, April 2010.

[9] J. Abella, X. Vera, and A. Gonzalez, "Penelope: The NBTI-Aware Processor," in *Proc. of the Int'l Symp. on Microarchitecture*, Dec 2007, pp. 85–96.

[10] J. Abella, X. Vera, O. Unsal, and A. Gonzalez, "NBTI-resilient memory cells with NAND gates for highly-ported structures," in *Workshop on Dependable and Secure Nanocomputing*, Dec 2007.

[11] A. Tiwari and J. Torrellas, "Facelift: Hiding and slowing down aging in multicores," in *Proc. of the Int'l Symp. on Microarchitecture*, Nov 2008, pp. 129–140.

[12] M. Basoglu, M. Orshansky, and M. Erez, "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*, Aug 2010, pp. 253–258.

[13] K.-L. Chen, S. Saller, and R. Shah, "The case of AC stress in the hot-carrier effect," *IEEE Trans. on Electron Devices*, vol. 33, no. 3, pp. 424–426, March 1986.

[14] X. Li, J. Qin, and J. B. Bernstein, "Compact Modeling of MOSFET Wearout Mechanisms for Circuit-Reliability Simulation," *IEEE Trans. on Device and Materials Reliability*, vol. 8, no. 1, pp. 98–121, March 2008.

[15] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt, "The M5 Simulator: Modeling Networked Systems," *IEEE Micro*, no. 4, pp. 52–60, July 2006.

[16] R. E. Kessler, "The Alpha 21264 microprocessor," *IEEE Micro*, vol. 19, no. 2, pp. 24–36, March 1999.

[17] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," in *Proc. of the Int'l Symp. on Microarchitecture*, Dec 2009, pp. 469–480.

[18] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design," *IEEE Trans. on VLSI Systems*, no. 5, pp. 501–513, May 2006.

[19] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F. Baez, "Reducing Power in High-performance Microprocessors," in *Proc. of DAC*, 1998, pp. 732–737.

[20] R. Kumar and G. Hinton, "A Family of 45nm IA Processors," in *Int'l Solid-State Circuits Conf. - Digest of Technical Papers*, Feb 2009, pp. 58 –59.

[21] K. Usami, T. Shirai, T. Hashida, H. Masuda, S. Takeda, M. Nakata, N. Seki, H. Amano, M. Namiki, M. Imai, M. Kondo, and H. Nakamura, "Design and Implementation of Fine-Grain Power Gating with Ground Bounce Suppression," in *Int'l Conf. on VLSI Design*, Jan 2009, pp. 381–386.

[22] A. Calimera, E. Macii, and M. Poncino, "NBTI-Aware Power Gating for Concurrent Leakage and Aging Optimization," in *Proc. of the Int'l Symp. on Low power electronics and design*, 2009, pp. 127–132.

[23] M. DeBole, R. Krishnan, V. Balakrishnan, W. Wang, H. Luo, Y. Wang, Y. Xie, Y. Cao, and N. Vijaykrishnan, "New-Age: A Negative Bias Temperature Instability-Estimation Framework for Microarchitectural Components," *Int'l Journal of Parallel Programming*, vol. 37, pp. 417–431, Aug 2009.