Empirical derivation of upper and lower bounds of NBTI aging for embedded cores

Empirical Derivation of Upper and Lower Bounds of NBTI Aging for Embedded Cores

Yukai Chen Enrico Macii Massimo Poncino
Department of Control and Computer Engineering
Politecnico di Torino
Corso Duca degli Abruzzi, 24 - 10129 Torino, ITALY
Email: yukai.chen@polito.it, enrico.macii@polito.it, massimo.poncino@polito.it

***Abstract*—In deeply scaled CMOS technologies, device aging causes transistor performance parameters to degrade over time. While reliable models to accurately assess these degradations are available for devices and circuits, the extension to these models for estimating the aging of microprocessor cores is not trivial and there is no well accepted model in the literature.**
This work proposes a methodology for deriving an NBTI-induced aging model for embedded cores. Since aging can only be determined on a netlist, we use an empirical approach based on characterizing the model using a set of open *synthesizable embedded cores*, which allows us to establish a link between the aging at the transistor level and the aging from the core perspective in terms of maximum frequency degradation.
Using this approach, we were able to (1) prove the independence of the aging on the workloads which run by the cores, and (2) calculate upper and lower bounds for the "aging factor" that can be used for a generic embedded processor.
Results show that our method yields very good accuracy in predicting the frequency degradation of cores due to NBTI aging effect, and can be used with confidence when the netlist of the cores is not available.

I. Introduction

Aging of CMOS devices has been one of the latest undesired side-effects of technology scaling. Among the many different aging phenomena, negative bias temperature instability (NBTI) has emerged as one of the most crucial factors in shortening the lifetime of devices [1]. A large bulk of research has addressed the issue of NBTI-induced aging from the modeling and optimization perspectives; they have been generally focused on logic blocks and SRAM structures because accurate characterization of NBTI aging *requires the availability of the circuit netlist* in order to extract critical paths and the signal probabilities of the relative cells. These information are easily available for logic circuits during the synthesis phase, and are implicit for SRAM structures, whose topology is well-defined. Extending this analysis to processor cores, however, is a quite different matter. In the typical design scenario, cores are in fact regarded as black-box, third-party IPs whose netlists are obviously not available. As a consequence, the state of the art in the modeling of the aging of a core is limited to very simple approximations based on the power states of the core: the core will age according to some constant *aging factor* when active, and it will not age (or recover) when idle. Aging and recovery are estimated assuming a baseline aging model, which can be either analytical (taken from physics, as in [2]–[6]) or empirical (derived by fitting data as in [7]). While some differences exist among the various approaches ([2]– [7]), this state-based aging model is the underlying common paradigm.

One problem with these approaches is that the assumption of a constant "aging factor" is not motivated nor validated. In fact, as [3] states, "*there is no publicly-available validated information on expected service life and aging rates of processors*". Therefore, previous works provide little or no hint on how this factor can be computed, and how a factor applicable to one core can be applied to a different one. Furthermore, the underlying models used in these works refer to the model for a single logic gate; however, in order to fully characterize the aging rate of a core, a more complex model should be used.

The limitations of these works are a consequence of the fact that an accurate characterization of the value dependency of NBTI is *only possible through an accurate logic simulation of the netlist.* The true NBTI aging of the critical path depends on the signal probabilities of the gates it contains; without these information only coarse approximations are feasible.

In this work, we determine such aging factor by using an empirical analysis on a set of open, synthesizable cores for which a gate-level netlist can be obtained. We target embedded processor cores with typical RISC architectures, for which we can have a better degree of confidence about the generality of the presented results. Our ultimate objective is in fact to derive an aging factor that is as much general as possible and applicable to a generic core, in particular when its netlist is not available.

The core netlists are used to obtain detailed aging data by running a set of application kernels as data for characterization and using a logic simulator augmented with transistor aging models [8]; aging data are then used to fit an underlying aging macromodel. Statistical correctness of the evaluation is guaranteed by using different datasets for the characterization and different cores for the validation.

The following are the main results that we obtained from our analysis:

1) We show that **aging of the core is independent of the workload**. Quite surprisingly, the impact of different applications and different datasets have negligible impact on the aging of the embedded core. As further elaborated in the paper, this is mainly due to the characteristics of the critical paths that determine the maximum frequency of the core; in practice, different executed instructions

and data do not alter significantly the probability values on the critical paths.

2) A direct consequence of the workload independence is that, for a given core, **it is possible to define a fixed aging factor**, to be interpreted as an equivalent signal probability to be used in a traditional transistor-level aging model. This is the most relevant result of our work.
3) Since in general any core will have its own aging factor because of differences in the architecture or different synthesis constraints, deriving a single, "universal" aging factor good for any embedded core is generally not feasible. However, the characterization on different architectures and with different synthesis constraints allows to yield **lower and upper bounds for the aging factor**. They can be used with a high degree of confidence for any embedded core having comparable complexity to those used in our characterization. Notice that these bounds are not related to different workloads but just to different architectures and different synthesis constraints.
4) Based on these outcomes, **we propose a practical NBTI aging macromodel for a core** that is a generalization of the traditional transistor-level one, where aging is expressed as a degradation of the maximum working frequency F_{max} and the best- and worst-case aging scenarios can be inferred based on the aging factor range.

Our validation is simulation-based, and is done on one of the synthesizable cores that was not used for the characterization. In this way we were able to compare the actual aging on the netlist with the bounds provided by the aging factor range applied to the macromodel. The comparison shows extremely good accuracy, with a maximum error of 2.88%.

The manuscript is organized as follows: in Section II we describe the background on NBTI and works related to our contribution. Section III presents the methodology used for characterization and to derive the proposed NBTI aging model; the reference embedded cores platforms are illustrated in Section IV. In section V we present the results of the model characterization phase, while Section VI shows our derived model and its relative validation by comparison of estimated and simulated results. Finally, Section VII discusses a few perspectives and possible enhancements to the proposed model.

II. Background

A. Background

Among the many different device aging mechanisms, NBTI is regarded as the most critical one. It causes a gradual increase in threshold voltage (V_t) and occurs when a pMOS transistor is negatively biased, that is, when $V_{gs} = -V_{dd}$, corresponding to a logic "0" being applied to the gate of a pMOS transistor (the $stress$ state); the increase of V_t absolute value causes then a degradation of the delay of the device. Conversely, when a logic "1" is applied to the pMOS gate, NBTI stress is partially removed (the $recovery$ state), resulting in a decrease V_t absolute value.

A compact and general model of NBTI-induced V_t drift for a transistor can be written as follows:

$$\Delta V_t = \alpha \cdot f(V_{dd}, V_t, T, \mathbf{R}) \cdot g(t). \quad (1)$$

The model has three main factors:

1) A term α denotes the *aging factor* which depends on (i) the actual stress/recovery pattern (i.e., time spent with the two logic values at the inputs) and is also affected by the activity (i.e., the time spent in active model) of the device.
2) A term including all technological and environmental parameters ($f()$): the degradation by NBTI depends on operational parameters, supply voltage V_{dd}, threshold voltage V_t, temperature T, and all the device parameters, lumped here for compactness into set $\mathbf{R}$, comprising for instance oxide geometrical and electrical parameters, activation energy, device size, and load. For the precise mathematical expression of $f()$ the reader is referred to classical NBTI overview paper [1].
3) A function $g(t)$ modeling the dependence over time of the drift. The shape of the function depends on the physical mechanism adopted to explain the NBTI effect [14]. Examples are the reaction-diffusion (R-D) model, for which $g(t) = t^n$, [9] indicates the values of n in the R-D model. Or the hole trapping (HT) one, for which $g(t) = 1 - e^{t^n}$, [10] shows that the reported time exponent n from previous works has a wide spread from 0.1 to 0.3; it also observes that n is around 0.2 can yield 90% accuracy NBTI aging under real use conditions. The work of [11] presents a thorough comparative analysis of the two models at the gate-level, and shows both models match well to obtain the NBTI degradation signature as a function of the *gate type, drive strength, input frequency,* and *the duty factor.* While in terms of *non-periodicity, instant degradation vs. long term degradation, CPU time* and *memory usage*, the two models show the different performance on these aspects. For example, the Atomistic trap based model can target the degradation simulation as fast as *ns* stress time, but cannot target the longer stress simulation; on the other hand, R-D model simplifies the BTI interpretation as a slow mechanism in order to speed-up the long-term stress simulation. The authors of [12] also give a review of R-D model and charge trapping model. Moreover, we refer the reader to [13] for the latest charge-trapping NBTI model. In this work we adopt the R-D model for our analysis since the technology library we used already includes the V_t drift information for a R-D model, and specifically $g(t) = t^{1/4}$ based on the technology library provided by *STMicroelectronics*.

For a given combination of environmental conditions, technological values, and $g(t)$, it is the value of α determines the actual V_t drift. Thanks to some mathematical properties of NBTI aging mentioned in [15], it can be shown that it is possible to use *signal probabilities* instead of actual signal values for the evaluation of the effective stress. It is worth remarking that the model of Equation 1 technically applies to an individual transistor and, with minor adaptation, to a logic

gate. The translation of the V_t drift on a more macroscopic performance metric such as circuit delay or processor maximum frequency still requires the availability of a netlist to determine the actual critical paths.

B. Previous Work

In the last few years, many studies have addressed NBTI aging from the design and EDA perspective, focusing on modeling first and then on various aging mitigation strategies. In this section we will focus our review on aging models that can be applied to processors; for a more in-depth overview of NBTI modeling solutions for generic logic circuits and memories we refer the reader to [14]–[17].

The work of [2] presents a NBTI-aware processor (Penelope) in which several aging mitigation strategies are proposed. For evaluation of these strategies, the authors do not use a true aging model but rather use a *NBTI efficiency* metric that combines the nominal delay, the TDP (Thermal Design Power) limit, and the NBTI guard-band of the processor. The latter is obtained by choosing the maximum guard-band required by any block, assuming that all paths of the different blocks have been adjusted to fit the cycle time to save power. In some sense this work relies a "static" NBTI model, where aging is not truly evaluated but statically defined in terms of a guard-band for each sub-block.

The authors of [3] also adopt a sort of "a priori" model. They assume target processor lifetime of 7 years, and evaluate two different aging rates, called *Low Wear-out* and *High Wear-out* life. They increase the delay of the critical paths by 10% and by 25%, respectively, in 7 years. They use an explorative approach also for other parameters related to NBTI aging, namely the average fraction of stress time of pMOS transistors, and the average ratio of pMOS to nMOS transistors in critical paths. Since the authors do not assume availability of the processor netlist, the critical paths are statistically estimated by using inverter chains for pipeline stages, while for memory stages they are estimated based on their circuit-level structure.

The authors of [4] present an analysis of workload-dependent aging effects in a large microprocessor. The authors propose a cell-level timing degradation model that is progressively extended to the path-level, block-level, and processor-level. They claim that while the timing degradation for different blocks (block-level) in the processor can vary significantly, however, such degradation of the whole processor (processor-level) roughly independent of the applications that are running on it, which is based on the assumption that active-state at all times (without considering power-state of processor). They use an architectural simulator that relies on an instance-level aging model but do not propose a true processor-level applicable as is. Furthermore, because they only refer one processor (and do not specify which one) the generality of results cannot be demonstrated.

The work of [5] aims at balancing workload in multicores using an aging-related metric. The aging model for a core relies on a traditional gate-level model for a critical path; the authors provide no insight on (i) how the critical path(s) of the processor is detected, and (ii) how the percentage of stress on these critical paths is computed. The resulting model, although approximate, works well because the objective is mainly to determine whether the load of a processor should be increased or decreased.

Another aging model was proposed in [6], they adopt the model of Equation 1 as a reference and extend it at the core level; $\mathcal{A}$ is assumed to be available either by direct characterization on the core or by using specific aging monitors (e.g., [18]–[20]). For the first option, they also suggest an approach similar to ours, based on the collection of statistics about delay and core activity at different operating conditions (i.e., V_{dd}, T) running benchmarks with different activity levels (i.e.,CPU bound and memory bound). A relationship between delay and core activity can be finally established for example using regression analysis. However, the paper does not specify further details about its implementation nor results.

III. Methodology

Figure 1 pictorially shows the three steps of our methodology, which are detailed in the following subsections.

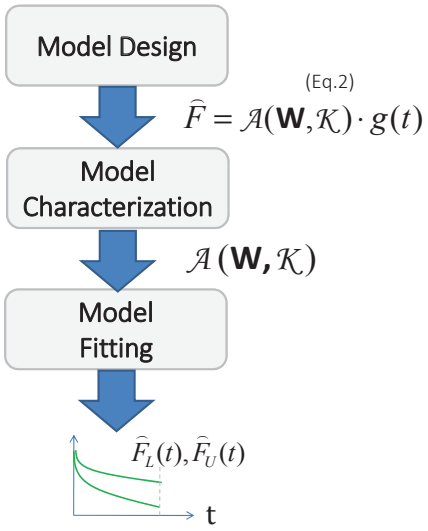

Figure 1. The three phases of the proposed methodology.

A. Model Design

The first step consists of the adaptation of the general transistor-level model of Equation 1 to the context of an entire processors core. Since Equation 1 expresses a drift in threshold voltage and technically refers to a single gate, our first task in order to model the aging of a processor is to devise a macro-model that (i) tracks a quantity related to the system-level performance of a embedded core, and (ii) uses a "core-level" activity factor (as opposed as a gate-level one).

The first objective can be met by modeling *the degradation of maximum operating frequency of the core* instead of that of the V_t degradation. This is done by transposing the V_t drift of equation 1 into a delay degradation using a classical alpha-power law [21], whose inverse determines the maximum operating frequency. Notice that this is possible since our

methodology relies on the availability of a core netlist, thus it is possible to accurately extract the critical path by simulation and compute the maximum frequency.
The second requirement implies that we should use an *aging factor* $\mathcal{A}$ that is some function of the *workload* $\mathbf{W}$. From the core perspective the workload is a mix of applications, whereas at the gate-level this will translate into some signal probability pattern in the circuit. This "core-level" activity factor will therefore represent a mapping of the core workload onto signal probability values. Equation 2 shows the expression of the proposed core-level NBTI aging model.

$$\hat{F}(t) = \frac{F_{max}^{aged}}{F_{max}^{nom}} = \mathcal{A}(\mathbf{W}, \mathcal{K}(V_{dd}, V_t, T, \mathbf{R})) \cdot g(t) \qquad (2)$$

where $\hat{F}$ is the normalized maximum operating frequency; $\hat{F}$ is 1 ($F_{max}^{aged} \equiv F_{max}^{nom}$) at time 0 and will progressively decrease over time.

- The function describing the evolution vs. time is a generic $g(t)$ **that has to be empirically derived from simulated data**. As a matter of fact, Equation 2 expresses the drift of frequency (and therefore delay), not of threshold voltage as in Equation 1. As such, the dependency will have a similar shape but not necessarily described by the same t^n function.
- The model has two main differences with respect to the one of Equation 1. The term $\mathcal{K}(V_{dd}, V_t, T, \mathbf{R})$, which conceptually maps to $f()$ in Equation 1, is incorporated into $\mathcal{A}$ because the model is empirical; what will be derived during the characterization is a factor that correlates $\hat{F}$ with the $g(t)$ term. In a single characterization run (for a given netlist and the relative synthesis constraints), the terms represented by $\mathcal{K}$ can be considered constant and are therefore included in the activity factor are $\mathcal{A}$.

The derivation of the actual model is carried out in the two other phases of the methodology.

B. Model Characterization

The second step is the most articulated one and concerns the empirical characterization of the model template obtained in the first phase by running simulations on the netlists of a set of synthesizable cores. In particular, this step empirically determines the value of the aging factor $\mathcal{A}()$.
Figure 2 shows the flow of the characterization phase in terms of tools, files, and formats.
Initially, the core is synthesized using Synopsys Design Compiler. The post-synthesis netlist is then simulated using a set of testbenches using Mentor's Modelsim. The testbenches are obtained by cross-compiling the source code of the test applications into executable codes through tool-chain correspond to the specific core. Modelsim pre-loads these executable codes to the processor memory, then read SDF file from previous step to perform a full timing simulation. Finally, Modelsim dumps the signal change profile of each node in the netlist by generating VCD (value change dump) files correspond to each run, from which we generated SAIF (switching activity interchange format) files to get static probability of each node from VCD files.

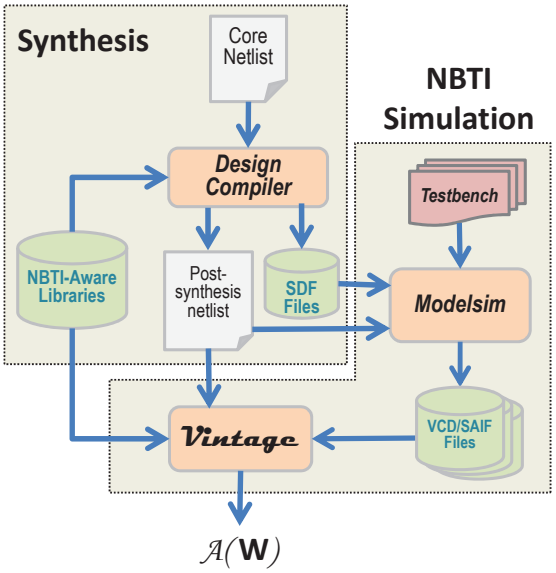

Figure 2. Flow of the Aging Factor Characterization Step.

This information is used by the Vintage tool [8], a plug-in of Synopsys PrimeTime for calculating the NBTI-induced aging. Vintage uses aging-characterized libraries to calculate gate-by-gate aging on a critical path. Since most conventional design kits do not provide designers with aging library that account for time-dependent variations, Vintage implements a SPICE-based flow for the analysis of the aging of CMOS library cells to achieve such aging library. The limitation in our experiments is the aging library is based on only one temperature since it based on the standard library provided, we only have 25 degree and 125 degree standard library provided by STMicroelectronics. In order to avoid aggressive results, we only chose 25 degree temperature NBTI-aware library in our experiment. Characterizing temperature dependence will be considered in our future work. [8] indicates the detailed process of characterization of NBTI-aware library; it implements a SPICE-based flow for the analysis of the aging of CMOS library cells. Depending on the statistics of the input signals of a cell, the simulations compute the aging of the pMOS based on HSPICE built-in aging models in the technology library and technology parameters provided by the library provider, then the stress information is integrated into the pMOS device parameters and the delay degradation of the cell is measured and stored in a dedicated LUT, ultimately the NBTI-aware library is derived.

NBTI-aware static timing analysis conducted by Vintage consists of two phases. First, the extraction of the stress in the netlist is carried out, which takes the output of probabilistic simulation of the design from Modelsim as one input, others inputs are the aging-aware library obtained from characterization describes above and the circuit description that is post-synthesis netlist in our methodology. In this step, the statistics of each signals of the design are annotated, and for each cell the corresponding delay degradation is computed using the aging models of the cells contained in the library. The second

phase is the actual static timing analysis on the annotated netlist. However, the objective of STA in here is not only the calculation of the critical path and the critical cells, because our target is the delay from an aging point of view, and the aging is value dependent, such analysis must identify a larger set of paths and cells since the paths that can become critical due to aging, the method in Vintage to solve such issue is calculating the Potentially Critical Paths(PCPs) [15], which those paths whose delay is within a given percentage from the nominal critical path. It is noticed that Vintage encompasses the analysis of the idleness periods of the circuits and the extraction of the sleep signal temporal distribution(e.g. busy or free), but we assume the core always in the active state in our experiments which is an aggressive assumption lead us get the aggressive aging degradation of the design. Finally, Vintage carries out the NBTI-aware static timing analysis with each SAIF file related to the different testbenches to derive the timing degradation of the critical path(s), from which the corresponding F_{max}^{aged} is obtained. Since the model of Equation 2 keeps the aging factor $\mathcal{A}$ encapsulates the physical term $\mathcal{K}$, and $\mathcal{K}$ is fixed in each run (an instance of a workload and a given dataset) according to the physical and environmental parameters limited by the standard library provided by the manufacture, $\mathcal{A}$ is directly obtained by the measured F_{max}^{aged} in different time points.

The process depicted in Figure 2 is executed for different synthesis options (e.g., high/nominal/low effort) and applied to different synthesizable core RTL descriptions in order to use a larger characterization sample. Each run will obviously yield a different $\mathcal{A}(\mathbf{W})$, which could impair the possibility of deriving an activity factor $\mathcal{A}$ that is as much general as possible. However, as the results will show, the characterization runs do exhibit an important feature, i.e., the rough independence of $\mathcal{A}$ from the workload, that will allow to derive a value of $\mathcal{A}$ that can be used with a good degree of generality.

C. Model Fitting

The last step concerns the translation of the $\mathcal{A}(\mathbf{W})$ derived in the previous phase into a true model of $\hat{F}(t)$ over time. Thanks to some specific peculiarities of the critical paths in the cores that are shown in the following, it is possible to determine an equivalent aging factor $\mathcal{A}$ that applies to the entire core (details for this phase are provided in Section V). However, in order to account for different synthesis constraints, rather than a single aging factor this phase yields a range $[\mathcal{A}_{min}, \mathcal{A}_{max}]$. This representing an upper and lower bound of the frequency degradation usable for any core. Therefore this step is simply an empirical fitting of the model of Equation 2 using the values of $\mathcal{A}$ and the measured F_{max}^{aged} at each timing point, we use curving fitting function in Matlab derive our NBTI aging model in our proposed methodology. It will derive thus the function $g(t)$ illustrate the aging maximum frequency evolution with time goes on, allowing to obtained a model of $\hat{F}(t)$ over time usable for any embedded core without the knowledge of the netlist or even of the internals.

IV. Experimental Setup

A. Reference Cores Platform

For our characterization we selected four widely used open-source embedded cores, with a synthesizable RTL HDL description and a full tool-chain for cross-compilation of applications sources. We chose these four cores as our target platforms because they have relatively general RISC-based architectures which cover most of traditional embedded cores; in this way, we can have a good degree of confidence about the generality of our methodology and results. The rest of the section lists the basic information of these four cores.

- The Plasma 3 CPU [23] is a synthesizable 32-bit RISC microprocessor implemented in VHDL, which executes all MIPS-I user mode instructions except unaligned load and store operations. The main memory communicates with the core and contains both instruction and data. It features an interrupt controller, UART, SRAM or DDR SDRAM controller, and Ethernet controller. Our version is implemented with three stage pipelines and an additional stage for memory reads and writes.
- The OR1200 [24] is a synthesizable 32-bit RISC CPU core with Harvard micro-architecture implemented in the Verilog HDL. It is maintained by developers at OpenCores.org. OR1200 is intended for embedded, portable and networking applications. The implementation features a power management unit, debug unit, tick timer, programmable interrupt controller, central processing unit, and memory management hardware. Our version is implemented with four-stage integer pipeline.
- The OR10N core [25] is a synthesizable 32-bit RISC processor with four pipeline stages. It is the improved version of OR1200 developed by ETH Zurich, it is redesigned the micro-architecture from scratch to achieve high IPC values, but maintained the four stage pipeline, the designers also balance the pipeline stages, which allows the core can run at a higher frequency and lower voltage for better energy efficiency. OR10N core processes 67% more instructions per second than the OR1200. Our version is implemented with its four-stage integer pipeline, the core is attached to an instruction RAM and a data RAM. The instruction memory interface is implemented such that the instruction RAM can be replaced with a cache, the data memory interface grants incoming requests in the same cycle.
- The RI5CY core [26] is a synthesizable 32-bit RISC-V processor core with four stages developed by ETH Zurich and University of Bologna. The ISA of RI5CY was extended to support multiple additional instructions including hardware loops, post-increment load and store instructions and additional ALU instructions that are not part of the standard RISC-V ISA. Our version is implement with its four-stage pipeline, it is attached to an instruction RAM and a data RAM through the instruction interface and data interface. The instruction interface connects the prefetch buffer, the data interface receives data from execution stage and sends data to write back stage.

V. Experimental Model Characterization

The model characterization phase of Figure 1 is the most important of the methodology; this section will describe the results obtained by running the flow of Figure 2 to our set of test cores.

A. Choice of the Cores for the Characterization

One important issue is to select which cores to use for the characterization and which one(s) to leave out and use for the validation. In order to make the most appropriate choice, we use the results of the synthesis process on the cores so as to avoid too evident correlations in the set used for the characterization.

We synthesized these four cores listed in Section IV on an industrial 45nm standard cell library by STMicroelectronics using a supply voltage of 1.1V and a temperature of 25°. Table I lists the synthesis results of each core under two different timing constraints. The reason for this is that different timing constraints may yield different critical paths and arrangements of the gates in the path, resulting in different aging factors. As noted in [4] and [22], the worst aging factor strongly depends on what kind of gates exist in the path and how these gates are arranged in the path.

	Synthesis Report		
Core	**Frequency (MHz)**	**Tot. Gates**	**# Gates CP**
Plasma	354.58	16.2k	81
	161.06	15.3k	82
OR1200	346.01	41.7k	66
	132.67	37.3k	91
OR10N	352.88	38.9k	62
	153.73	36.6k	92
RI5CY	352.47	40.3k	47
	137.99	38.4k	118

Table I
Synthesis Results of Each Core with Different Timing Constraints.

The two timing constraints chosen were 3 and 10 ns, corresponding to high and low frequency bounds. These values are selected arbitrarily suggested by common sense, but any two frequency bounds (or even more values) could be used; the objective is essentially that of generating different post-synthesis netlist to determine the sensitivity to frequency options by the aging factor. For each core, the table reports two rows with (i) the frequencies resulting from the two timing constraints (top row 3 ns, bottom row 10 ns), (ii) the number of total gates in the post-synthesis netlist, and (iii) the number of gates in the critical path.

As an example, the synthesis for OR10N with 3 ns constraint (fifth line in Table I) yields the following values; the critical path occurs **within the execution stage** of the core, and specifically from signal id_stage_i/alu_operator_ex_o_reg[2] and ends with signal id_stage_i/alu_operand_b_ex_o_reg[31]; it includes 62 gates, resulting into a critical delay of 2.83ns, which corresponds to a nominal frequency $f_{max,nom} = 352.88$ MHz. The synthesis with the 10 ns constraint yields a longer critical path and a reduced number of gates of netlist decrease. The critical path occurs **within the instruction decode stage**, from signal if_stage_i/instr_rdata_id_reg[2] to signal id_stage_i/alu_operand_b_ex_o_reg[31] (92 gates), leading to a critical delay of 6.505 ns, corresponding to a frequency $f_{max,nom} = 153.73$ MHz.

We then used the information in the table to determine the three cores used for characterization and the one left out and used for validation. The table shows that the Plasma core is less complex than the others cores, so we included it in the characterization set. Then, since OR10N is an improved version of OR1200, we chose to include OR1200 and OR10N for the characterization in order to avoid using one core for characterization that was too similar to one used for validation. Therefore, we left RI5CY for the validation of our results.

B. Workload Definition and Analysis

In terms of workloads for the characterization, we chose a set of application kernels that exercise in different ways the various components of the core and induce diverse static probabilities of signals in the core. We used different sets for each core because of their complexity. Plasma for instance is relative simple than other two, and supports only the compilation of code of moderate size. Table II lists the set of applications used as input testbenches for the three netlists;

Application	**Plasma**	**OR1200**	**OR10N**
Bubble Sort	•	•	•
Heap Sort	•	•	•
Quick Sort	•	•	•
Hello World	•	•	•
Count Number	•		
Calculate Pi	•		
Matrix Mul16	•		
Matrix Mul32		•	•
FFT	•	•	•
RGB convert	•	•	•
Gauss-2D filter		•	•
Non separable 2D filter		•	•

Table II
Application list we tested in our experiment

the bullet in each entry indicates if the application is used for a given core.

We ran these selected applications on these three cores to derived the frequency degradation profile over a 10-year interval according to our methodology as shown in Figure 2. Figures 3–5 show the aging frequency degradation curves for Plasma, OR1200, and OR10N, respectively, and using the 10ns constraint (low frequency). It is immediately visible from the plots that the difference among the curves is almost negligible. For Plasma, the maximum difference of aging frequency among two applications is 1.45 MHz (about 0.90%), for OR1200 is 1.57 MHz (about 1.18%), and for OR10N is 1.21 MHz (about 0.79%). The overall degradation over 10 years is approximately 25% for Plasma and OR1200, and 15% for OR10N. We omit the aging degradation results of the three cores under 3ns constraint (high frequency) because they are very similar to the results under 10ns constraint.

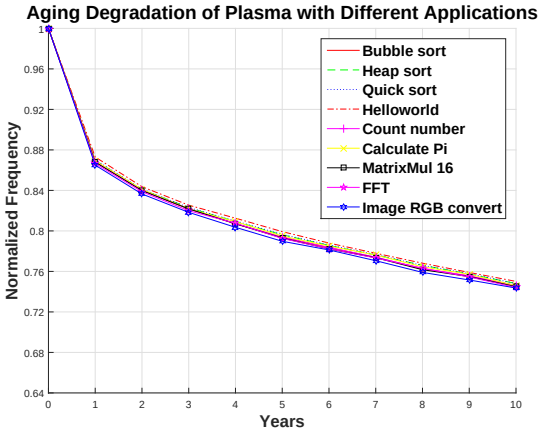

Figure 3. Aging Degradation of Plasma for the Test Applications.

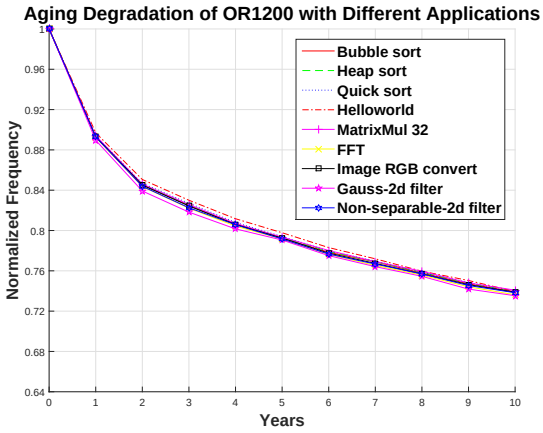

Figure 4. Aging Degradation of OR1200 for the Test Applications.

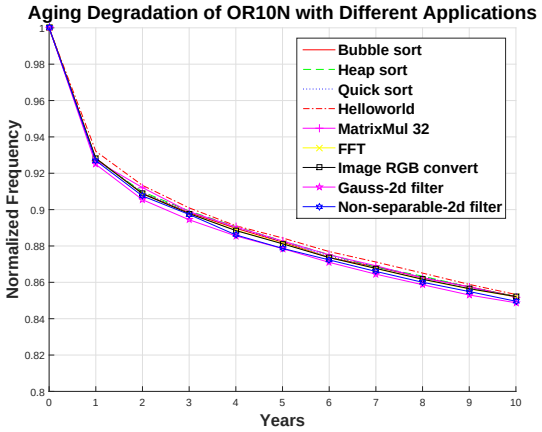

Figure 5. Aging Degradation of OR10N for the Test Applications.

The results of Figures 3–5 refer to a single dataset for each application. In order to further exercise diverse signal probability values, we artificially design multiple datasets for the applications with the objective of achieving as much different as possible static probabilities in the netlist. This has been achieved by defining for each application a 0-dominated (90% of the bits are randomly set at '0' in the data) and a 1-dominated (90% of the bits at '1') datasets. For applications that are strongly data-dependent such as the sorting ones, we further used different initial datasets (sorted, random, inverse sorted) so that the corresponding algorithms will result in different numbers of executed instructions.

Figures 6–8 show a sample of the results by running Bubble-Sort and Matrix Mul32 with the 0- and 1-dominated datasets, and the sorted/random/inverse sorted datasets for QuickSort and HeapSort programs.

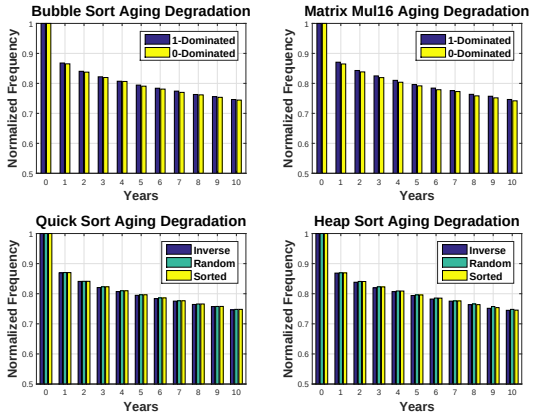

Figure 6. Aging Degradation of Plasma with Different Input Datasets.

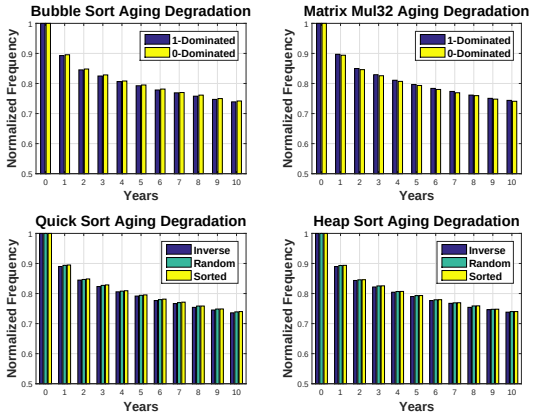

Figure 7. Aging Degradation of OR1200 with Different Input Datasets.

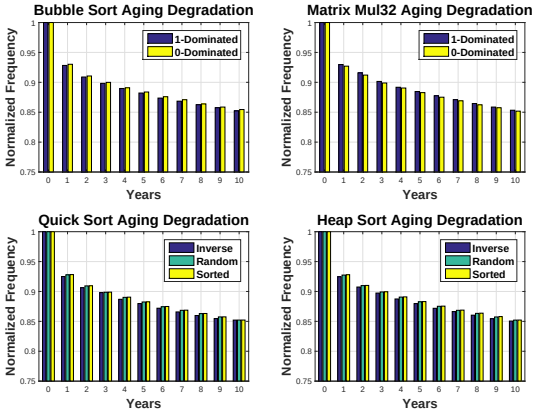

Figure 8. Aging Degradation of OR10N with Different Input Datasets.

The difference in the aging within each set of bars is barely distinguishable; the max difference of bars in Plasma core is 0.89 MHz (about 0.5%), in OR1200 core is 0.73 MHz (about 0.6%) and in OR10N is 0.96 MHz (about 0.6%). Such difference is even smaller than the difference resulting by different applications; therefore, data values seem to affect the aging in a negligible way for all the three cores.

C. Forcing Input Probabilities

The previous experiments have shown the aging frequency of the cores is weakly affected by (i) the applications run by the cores and (ii) the data used by these programs. The largest deviation of aging frequency degradation among all the experiments we have run is only about 1.18%.

The above data seems to suggest that aging degradation is independent of the workload in terms of executed instructions and input datasets. However, this observation could be affected by the limited sample used or by a poor choice of the application mix, which might possibly exercise only a limited range of signal probabilities in the netlists. Therefore, to further verify this claim, we *forced a number of combinations of static probability values at the inputs of the pipeline stage that contains the critical path*. The signal probabilities in the netlist are then determined by propagating the input probabilities, and the NBTI degradation for each gate in the critical path is then computed with the NBTI-aware library according to the annotated signal probabilities annotated.

An exhaustive simulation of all probability values is clearly unfeasible. Even assuming a coarse grain quantization of the probabilities (e.g., in steps of 0.1), the number of combinations is prohibitive. If the target pipeline stage has n inputs, and D is the number of probability values (e.g., 11 for a step of 0.1, from 0 to 1), that exhaustive enumeration would be 11^n. Considering the the target cores are 32-bit cores, n is at least 64, i.e., the size of two registers, but is in general much larger. For example, in the Plasma core the inputs of the critical pipeline stage are a_bus[32], b_bus[32], alu_func[4], branch_func[3], c_source[3], imm[16], mult_func[4], rd_index[6] and shift_func[2], for a total of n=102 bits [27].

We therefore sampled the space by applying a small subset of $N = 60$ distinct probability patterns; notice that each simulation and aging extraction run takes approximately 45 minutes. In order to increase the meaningfulness of the sample we have applied random probabilities to the input selectively, in order to avoid exploring probability patterns that are unlikely to occur:

- Inputs relative to vectors carrying data values are assigned as follows:
  - Since very large integers are seldom used in programs, we roughly assume that the 12 MSBs of data are unused. Assuming then that positive and negative values are equally probable, bits from 20 to 31 are fixed at 0.5 probability. Due to sign extension (data are represented in 2's complement), all unused bits will be either 0 (positive values) or 1 (negative values);
  - Bits from 0 to 19 are sampled randomly using discretized probability values from 0.0 to 0.99 in steps of 0.1.

  Notice that this two-region model of data is widely used in power macromodeling of arithmetic operators, and is known as the *dual-bit type model* [27].
- Inputs relative to vectors carrying addresses are considered as fully random. Notice that in principle we could restrict further the space since in the pipeline stage containing the critical path addresses are *data*a addresses which are likely to be limited in one specific portion of the address space (i.e., some bits could be fixed at 0 or 1 probability).
- Inputs relative to control bits are considered as fully random.

Although the sampling is partial, it explores values in a more randomized way than what can be obtained by running an application mix with multiple datasets. This strategy shares some similarity with classical approaches used to generate power macromodels for RTL power estimation [28], [29].

Core	Timing Constraints	Min Aging 10yrs	Max Aging 10yrs	Δ [%]
Plasma	354.58 MHz	265.59 MHz	269.81 MHz	1.2%
	161.06 MHz	121.27 MHz	118.68 MHz	1.6%
OR1200	346.01 MHz	240.85 MHz	244.68 MHz	1.1%
	132.67 MHz	97.23MHz	100.12 MHz	2.2%
OR10N	352.88 MHz	238.62 MHz	241.91 MHz	0.9%
	153.73 MHz	129.89 MHz	131.97 MHz	1.4%

Table III
AGING RANGES WHEN FORCING PROBABILITIES OF THE PIPELINE STAGE INPUTS.

Results are reported in Table III, which reports the worst- and best-case aging after 10 years. Since the curves tend to diverge as the time horizon increases the point at 10 years represents the largest difference between best and worst cases. We ran the experiment for the three cores and for both timing constraints (high and low frequency). Results are even more surprising, since the simulations confirm the previous results, showing that different static input probabilities (at the inputs of the critical pipeline stage) affect only marginally the frequency degradation of the cores.

The largest difference among all points is 1.6% for Plasma, 2.2% for OR1200 and 1.4% for OR10N within different timing constraint situations. The ranges are a bit larger than those resulting from real applications; however, the difference is quite limited. Although a sample of very large space, this further strengthens the claim that the results from running application are not particularly "fortunate" cases for aging, but they are consistent with a larger exploration of the possible input values.

D. Removing the Dependency of Workload

The results shown in the previous section empirically demonstrate one fundamental outcome of the characterization phase: *since aging is not affected by the input static probability values, it is possible to drop the dependency of the workload* $\mathbf{W}$ *from* $\mathcal{A}$ *in Equation 2*. This allows using, for a given instance of the flow of Figure 2, i.e. a core netlist with its synthesis constraint, **a constant aging factor $\mathcal{A}$, regardless of what application is executed.** While this is exactly the assumption adopted by most of the previous works, this feature was assumed without a motivated evidence and no clear indication was given about the actual value of this constant aging factor.

So far, we have empirically shown that the use of this constant factor is justified. However, this property alone does not provide a hint on which value of $\mathcal{A}$ should be used. What is needed is in fact an "equivalent" fixed aging factor that can be applied to all the critical gates and that can be characterized once and for all.

In order to obtain this, we artificially *set different static probability values to* **all signals in the core netlist.** Notice that these values are not logically feasible and can only be forced by writing a corresponding VCD file. Specifically we forced all static probability values of all signals in the core netlist by varying them between 0.0 and 0.99 in steps of 0.1. For each configuration we ran the usual flow to derive the aging curves; this was repeated for each characterization point (core and synthesis constraint). Figure 9 conceptually depicts the two extremes of the range (all signal assigned to 0's or to 1's), which determine the worst and best case for aging.

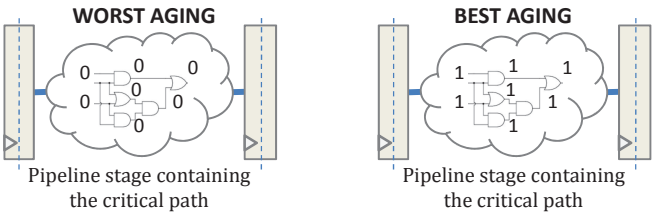

Figure 9. Artificially Setting Signal Probability Values in the Netlist.

The gate-wise assignment of probabilities finally results in different aging profiles for different probabilities. Figure 10 reports the frequency degradation over time for the various probability values assigned to the internal signals, for the three cores and for the two frequency values. As we described in the model characterization section, we do not only consider the critical path of the fresh core, but also set the guardband to catch the critical path evolved from those potentially critical paths. We mark the time point when the critical path changes as time advance in Figure 10. We do not mark all of them since there are too many of them (For instance, we do not mark first year aging frequency points because the critical path always changes during the first year). For instance of OR10N core under high frequency with all signals have 0.7 static probability, the fresh critical path is from signal id_stage_i/alu_operator_ex_o_reg[2] and ends with signal id_stage_i/alu_operand_b_ex_o_reg[31]; then the critical path changes to one from signal id_stage_i/mult_signed_mode_dot_ex_o_reg[0] and ends with signal id_stage_i/mult_operand_b_dot_ex_o_reg[25] at first year; it changes again to the one from signal id_stage_i/mult_signed_mode_dot_ex_o_reg[0] and ends with signal id_stage_i/mult_operand_b_dot_ex_o_reg[28] after two years. Aging effects of all three cores are now sizable, the difference 10-year aging between the worst-case (all signals at 0.0 probability) and the case with all signals at 0.99 probability is very obvious. As a side result, the figure provides upper and lower bounds on the aging frequency. Clearly, all 0's will age all p-transistors (worst case) and 0.99 static probability will allow them to recover (best case). Obviously, aging monotonically decreases for increasing static probability.

In order to obtain the fixed gate-level aging factor, for a given core we superimpose the aging curve obtained by setting input probabilities as described in Section V-C (Table III) on the corresponding plot of Figure 10. The aging curve will lie between two curves of Figure 10, by linear interpolation of these two values we obtain the fixed aging factor. Figure 11 shows an example of the process; curves are in this case relative to the Plasma core.

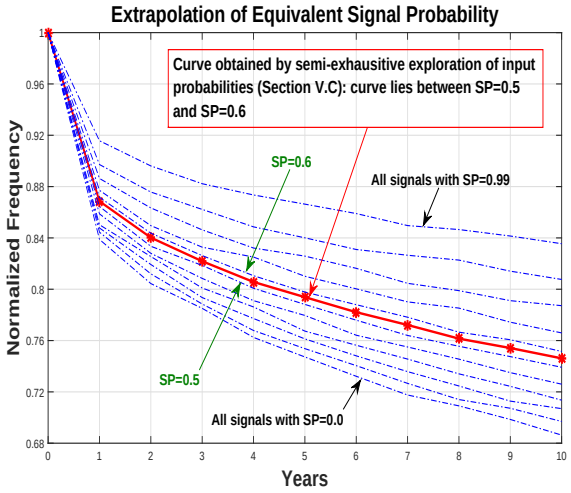

Figure 11. Extrapolation of the Equivalent Signal Probability.

The solid red curve is the curve that represents the sheaf of aging curves obtained for various input probabilities points, as described in Section V-C. Since the sheaf is quite concentrated, it is a reasonable approximation to take the average of all the curves and use it as a representative of the entire sheaf. The dashed curves are those obtained by setting all signal probabilities to the various values from 0 to 0.99 (they correspond to the top right plot of Figure 10). We can see that the red curve lies between the 0.5 and 0.6 probability, being closer to the latter value. Using the curve interpolation with Matlab, we obtain a value of 0.56 as the aging factor for this instance.

The outcome of this operation is particularly relevant; since the actual aging of an instance (core/synthesis constraints) does not depend on the workload, **we can associate an equivalent aging factor to the entire core as if all gates in the critical path were aging of the same amount**.

The important consequence of this is that this process allows us to use a "gate-level" model of aging for the whole core similar to Equation 1, where the activity factor is constant for a given instance. Obviously, different netlists will have a different equivalent aging factor. The values of the latter for the three cores used for characterization under the two timing constraint are reported in Table IV.

As a general observation, we can notice that more stringent constraints (higher frequency) yield smaller and closer values in the rage (0.30–0.32); as we move to slower implementations (lower frequency), the range gets larger (0.42–0.56) and also shifted towards larger values. Notice also that simpler netlists

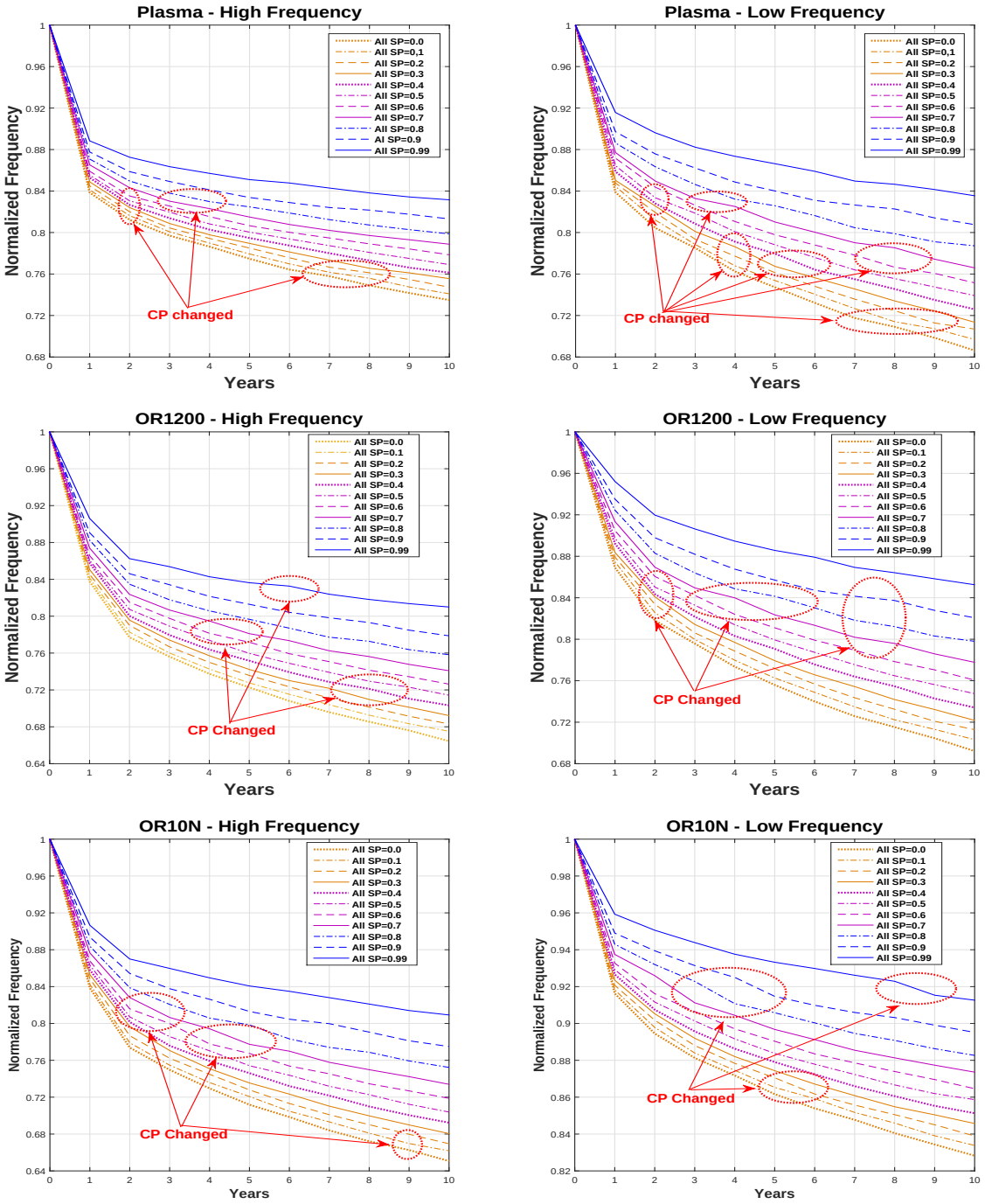

Figure 10. Aging Degradation of Cores with Different Static Probability for All Signals in the Core Netlist.

Core	F_{max} [MHz]	$\mathcal{A}$
Plasma	354.58	0.32
	161.06	0.56
OR1200	346.01	0.31
	132.67	0.46
OR10N	352.88	0.30
	153.73	0.42

Table IV
EQUIVALENT AGING FACTORS FOR THE THREE CORES AND THE TWO TIMING CONSTRAINTS.

like the Plasma have larger range between low and high frequency than more complex core such as the OR1200 or OR10N, denoting a stronger sensitivity to the synthesis process for the given timing constraints. It is also shown that ranges of Plasma are wider because the netlist gets more optimized given the relatively low constraints compared to more complex cores. From the other side, in a more optimized network, the critical path becomes then more sensitive to the aging and the two extremes become wider. The actual and final output of the characterization is therefore a range of values for $\mathcal{A}$

determined by the largest upper and smallest lower bounds over all the characterization runs. Our simulation yields the range $\mathcal{A} = [0.30, 0.56]$. The flow of Figure 1 simply yields in this case the above range rather than a true function as initially described, thanks to the special property of workload insensitivity.

E. Model Fitting

Thanks to the workload independence observed during the characterization phase, the last step of the methodology of Figure 1 becomes relatively straightforward. Since $\mathcal{A}$ is simply a range, we can derive the frequency degradation $\hat{F}(t)$ as a pair of lower and upper bound function $[\hat{F_L}(t), \hat{F_U}(t)]$. Notice that the values of $\mathcal{A}$ derived empirically include the technology-dependent factor $\mathcal{K}$ of Equation 2.

The fitting process consists of finding the best approximation to the upper and lower bound curves by determining the function $g(t)$ of Equation 2. Using Matlab curve fitting for the two curves yields the following functions:

$$\hat{F_L}(t) = \mathbf{0.30} \times (-0.5175 \times t^{0.3259} + 3.3269) \quad (3)$$

$$\hat{F_U}(t) = \mathbf{0.56} \times (-0.1384 \times t^{0.2936} + 1.8068) \quad (4)$$

The SSE and RMSE of the above functions generated by Matlab curving fitting are $8.47 \cdot 10^{-5}$ and $3.26 \cdot 10^{-3}$.

These two equations fit the template of Equation 2 by separating the empirically-derived and the technology-dependent factor $\mathcal{K}$. However, if we re-arrange $[\hat{F_L}(t)$ and $\hat{F_U}(t)]$ we get:

$$\hat{F_L}(t) = 1 - 0.1465 \times t^{0.3205} \quad (5)$$

$$\hat{F_U}(t) = 1 - 0.0719 \times t^{0.3023} \quad (6)$$

These two functions are reminiscent of the more traditional "power-law" aging formula of threshold voltage (Equation 1), yet are in the form $1 - t^n$ since $\hat{F}$ expresses a quantity that decreases over time due to aging, as opposed to threshold voltage.

The functions of Equations 5 and 6 exhibit an interesting feature. The empirically derived $\mathcal{A}$ is now modulated by the interpolation to yield a sort of "effective" aging factor (0.1465 for $\hat{F_L}$, 0.0719 for $\hat{F_U}$), smaller than $\mathcal{A}$. This is however a result of the empirical fitting process, while $\mathcal{A}$ truly represents the actual aging factor derived from circuit-level analysis.

VI. Model Usage and Validation

Our claim is that the two functions $\hat{F_L}(t)$ and $\hat{F_U}(t)$ denote the aging bounds (in terms of maximum frequency degradation) of any core (of comparable complexity to the ones used in characterization, and under the same environmental and technological conditions).

For the assessment of the model, we ran the same methodology for the core left out from the characterization, namely, the RI5CY one [26]. Table V shows results of the core RI5CY. The aging factors (obtained using the method described in Section V-D under the two timing constraints are respectively 0.32 and 0.50, which are both inside the region of $\mathcal{A}$ found empirically.

Core	Frequency(MHz)	$\mathcal{A}$
RI5CY	352.47	0.32
	137.99	0.50

Table V
AGING FACTORS OF RI5CY CORE

In order to quantify the error caused by different range values ($[0.32 - 0.50]$ for the RI5CY vs. the $[0.30 - 0.56]$ range used in Equation 3 and 4) at different time points, we compare simulated and estimated maximum frequency values for two implementations (low- and high-frequency) of the RI5CY core. Tables VI and VII report the relative results.

Years	F_{max} *(Simulated)*	F_{max} *(Estimated, Worst)*	Error [%]
0	352.4692 MHz	352.4692 MHz	-
1	301.6320 MHz	297.0681 MHz	1.513%
2	275.2635 MHz	283.1993 MHz	2.883%
3	267.6943 MHz	273.5097 MHz	2.173%
4	262.0570 MHz	265.8155 MHz	1.434%
5	257.7382 MHz	259.3304 MHz	0.618%
6	252.6876 MHz	253.6701 MHz	0.388%
7	247.4047 MHz	248.6749 MHz	0.489%
8	244.5705 MHz	244.0259 MHz	0.223%
9	242.2467 MHz	239.1769 MHz	1.006%
10	237.6745 MHz	235.8971 MHz	0.748%

Table VI
ACCURACY OF PROPOSED NBTI MODEL FOR RI5CY CORE (HIGH FREQUENCY IMPLEMENTATION).

Years	F_{max} *(Simulated)*	F_{max} *(Estimated, Best)*	Error [%]
0	137.9890 MHz	137.9890 MHz	-
1	128.1659 MHz	128.9237 MHz	0.591%
2	125.9604 MHz	126.5099 MHz	0.436%
3	124.5290 MHz	124.8528 MHz	0.260%
4	123.3782 MHz	123.5514 MHz	0.140%
5	122.3700 MHz	122.4636 MHz	0.077%
6	121.5128 MHz	121.5202 MHz	0.006%
7	120.7220 MHz	120.6823 MHz	0.033%
8	120.0146 MHz	119.9252 MHz	0.075%
9	119.4058 MHz	119.2323 MHz	0.145%
10	118.8340 MHz	118.592 MHz	0.204%

Table VII
ACCURACY OF PROPOSED NBTI MODEL FOR RI5CY CORE (LOW FREQUENCY IMPLEMENTATION)

Values in columns F_{max} *(Estimated, Worst)* and F_{max} *(Estimated, Best)* are obtained using the lower and upper bound Equations 3 and 4 respectively. The frequency degradation for the high frequency implementation of the core (Table VI) exhibits a maximum error of 2.883%, when comparing against the worst-case aging; when considering the low frequency implementation the maximum error is even smaller (0.591%), in this case when compared against the best case aging. Notice that, due to the empirical curve fitting, the error is not monotonic over the various time points; in any case, the variance of the error remains quite limited.

It is evident from these numbers that a difference in the actual aging bounds actually results in lower errors in the final estimate. For instance, for the RI5CY core the 0.32 simulated

factor vs. the 0.30 of the model (6.6% difference) results only in an average error of about 1.1% for the worst case; the best-case factors (0.5 vs. 0.56, 10.7% difference) yields an average error around 0.17%.

It is worth observing that the comparison in Tables VI and VII refers to a scenario in which the designer knows whether the core be analyzed has been synthesized for a fast or slow synthesis corner. In general, however, the designer is unaware of this details; in this case he might decide to use the bounds and obtain a range of aging rather than a single value.

Conversely, if the designer (erroneously) chooses to pick one of the two extremes and incidentally the target core has been designed using the opposite synthesis corner. Specifically, designers use the bound for a $slow$ corner when the characterization is done for the $fast$ one. An interesting figure could be what is the maximum error than one can expected by the mismatch of the two extremes. Tables VIII and IX provide the answer, for the cases of a fast corner compared against the aging lower bound, and a slow corner compared against the upper bound, respectively. This is clearly a wrong design choice that would not be done by a designer, in spite of such a wrong match of corners, the maximum error is 27%. On the other hand, the true errors for a matched corner analysis are shown in tables VI and VII, in this case errors are truly small.

Years	F_{max} *(Simulated)*	F_{max} *(Estimated, Best)*	Error [%]
0	352.4692 MHz	352.4692 MHz	-
1	301.6320 MHz	329.3134 MHz	9.177%
2	275.2635 MHz	323.1479 MHz	17.396%
3	267.6943 MHz	318.9149 MHz	19.134%
4	262.0570 MHz	315.5909 MHz	20.428%
5	257.7382 MHz	312.8121 MHz	21.368%
6	252.6876 MHz	310.4026 MHz	22.840%
7	247.4047 MHz	308.2623 MHz	24.598%
8	244.5705 MHz	306.3283 MHz	25.251%
9	242.2467 MHz	304.5584 MHz	25.722%
10	237.6745 MHz	302.9224 MHz	27.452%

Table VIII
WORST-CASE ERROR FOR RI5CY CORE (CASE OF HIGH FREQUENCY IMPLEMENTATION) USING THE WRONG CORNER.

Years	F_{max} *(Simulated)*	F_{max} *(Estimated,Worst)*	Error [%]
0	137.9890 MHz	137.9890 MHz	-
1	128.1659 MHz	116.2999 MHz	9.258%
2	125.9604 MHz	110.8704 MHz	11.980%
3	124.5290 MHz	107.0770 MHz	14.015%
4	123.3782 MHz	104.0647 MHz	15.654%
5	122.3700 MHz	101.5259 MHz	17.034%
6	121.5128 MHz	99.3099 MHz	18.272%
7	120.7220 MHz	97.3309 MHz	19.376%
8	120.0146 MHz	95.5343 MHz	20.398%
9	119.4058 MHz	93.8834 MHz	21.375%
10	118.8340 MHz	92.3519 MHz	22.285%

Table IX
WORST-CASE ERROR FOR RI5CY CORE (CASE OF LOW FREQUENCY IMPLEMENTATION) USING THE WRONG CORNER.

We notice that the errors are not so dramatic; the largest error is 27.452%. Moreover, unlike the case of matched corners (Tables VI and VII) where the errors are roughly constant, here they increase as the time horizon gets larger, and the errors under the case of high frequency implementation increase more rapid than ones under the case of low frequency implementation, although both of them have similar values (around 9.2%) at first year.

Again, the purpose of this analysis is just for assessing the extreme case of an incorrect use of the model by the designer. The right use of the model in absence information about the implementation of the core would be to keep the two bounds to get a range of aging degradation at different time points instead of a single value.

VII. CONCLUSIONS

In this paper we have presented a NBTI aging model for embedded processor cores. The proposed methodology results into two major conclusions: (1) the aging degradation is independent of the workload, i.e., the switching activity in the netlist, and, based on this first observation, (2) it possible to identify an "effective" static probability, called *aging factor*, that can be used as a core-wide stress probability, which allows one to use a gate-level aging model for the entire core. This value has been empirically derived as fixed stress probability correspond to different cores based on a set of synthesis data, which shows that the aging factor is different for different embedded cores, and for different synthesis constraints. However, by taking the extreme values of the aging over all design points it is possible to derive upper and lower bounds to the aging as a function of time.

It is worth emphasizing that the proposed methodology does not claim to propose an universal model for all types of processors and under all environmental, technological, and synthesis conditions. The following are possible limitations of the proposed method, some of which represent possible directions for future research:

- The characterization procedure could not be very extensive due to the limited number of synthesizable core publicly available; therefore, the statistical strength of the methodology is somehow limited.
- The models derived are applicable to processors belonging to the class of "embedded" cores or micro-controllers, with relatively simple RISC-like architectures, and we cannot claim it may apply also to the case of high-end super-scalar cores with aggressive out-of-order execution mechanisms.
- The proposed model focuses only on the true aging effects on circuit delay (and hence of processor frequency) *for a specific PVT corner*. As a matter of fact, we are not considering here neither variability nor temperature effects on the aging process.
- The models do not consider the effect of power management knobs such as voltage scaling, which is known to affect the NBTI aging. This is not however a limitation per se, since it has been shown that is possible to calculate total aging by accruing the aging in multiple time slots each one using a different supply voltage value [30].

Nevertheless, the proposed solution provides a method for the construction of aging bounds that is in principle applicable

to any embedded core. The models expose some limitation of previous works that assumed a fixed aging factor for a given core under any operational conditions and without any convincing motivation.

The proposed aging model shows good accuracy when predicting the maximum frequency degradation of a core, and our methodology enable a statistical aging analysis in a standard design flow, improving design predictability and helping to avoid pessimistic guardbanding under the NBTI aging effect.

REFERENCES

[1] Alam, M. "Reliability-and process-variation aware design of integrated circuits." *Microelectronics Reliability 48*, no. 8 (2008): 1114-1122.

[2] Abella, Jaume, Xavier Vera, and Antonio Gonzalez. "Penelope: The NBTI-aware processor." In *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 85-96. IEEE Computer Society, 2007.

[3] Tiwari, Abhishek, and Josep Torrellas. "Facelift: Hiding and slowing down aging in multicores." *Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, pp. 129-140. IEEE, 2008.

[4] Mintarno, Evelyn, Vikas Chandra, David Pietromonaco, Robert Aitken, and Robert W. Dutton. "Workload dependent NBTI and PBTI analysis for a sub-45nm commercial microprocessor." *Reliability Physics Symposium (IRPS), 2013 IEEE International*, pp. 3A-1. IEEE, 2013.

[5] Sun, Jin, Avinash Kodi, Ahmed Louri, and Janet M. Wang. "NBTI aware workload balancing in multi-core systems." *Quality of Electronic Design, 2009. ISQED 2009. Quality Electronic Design*, pp. 833-838. IEEE, 2009.

[6] Paterna, Francesco, Andrea Acquaviva, and Luca Benini. "Aging-aware energy-efficient workload allocation for mobile multimedia platforms." *IEEE Transactions on Parallel and Distributed Systems 24*, no. 8 (2013): 1489-1499.

[7] Srinivasan, Jayanth, Sarita V. Adve, Pradip Bose, and Jude A. Rivers. "Lifetime reliability: Toward an architectural solution." *IEEE Micro 25*, no. 3 (2005): 70-80.

[8] Calimera, Andrea, Enrico Macii, and Massimo Poncino. "NBTI-aware clustered power gating." *ACM Transactions on Design Automation of Electronic Systems (TODAES) 16*, no. 1 (2010): 3.

[9] Mahapatra, S., Goel, N., Desai, S., Gupta, S., Jose, B., Mukhopadhyay, S., Joshi, K., Jain, A., Islam, A.E. and Alam, M.A., " A comparative study of different physics-based NBTI models." *IEEE Transactions on Electron Devices*, 60(3), pp.901-916, 2013

[10] Gao, R., Manut, A.B., Ji, Z., Ma, J., Duan, M., Zhang, J.F., Franco, J., Hatta, S.W.M., Zhang, W.D., Kaczer, B. and Vigar, D., "Reliable Time Exponents for Long Term Prediction of Negative Bias Temperature Instability by Extrapolation". *IEEE Transactions on Electron Devices*, 64(4), pp.1467-1473, 2017

[11] Kukner, H., Khan, S., Weckx, P., Raghavan, P., Hamdioui, S., Kaczer, B., Catthoor, F., Van der Perre, L., Lauwereins, R. and Groeseneken, G., 2014. "Comparison of reaction-diffusion and atomistic trap-based BTI models for logic gates". *IEEE transactions on device and materials reliability*, 14(1), pp.182-193.

[12] Grasser, Tibor, et al. "The paradigm shift in understanding the bias temperature instability: From reactiondiffusion to switching oxide traps." *IEEE Transactions on Electron Devices* 58.11 (2011): 3652-3666.

[13] Wimmer, Y., El-Sayed, A.M., Gs, W., Grasser, T. and Shluger, A.L., 2016, June. "Role of hydrogen in volatile behaviour of defects in SiO2-based electronic devices". In *Proc. R. Soc*. A (Vol. 472, No. 2190, p. 20160009). The Royal Society.

[14] Mahapatra, S., N. Goel, S. Desai, S. Gupta, B. Jose, S. Mukhopadhyay, K. Joshi, A. Jain, A. E. Islam, and M. A. Alam. "A comparative study of different physics-based NBTI models." *IEEE Transactions on Electron Devices 60*, no. 3 (2013): 901-916.

[15] Kumar, Sanjay V., Chris H. Kim, and Sachin S. Sapatnekar. "An analytical model for negative bias temperature instability." In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pp. 493-496. ACM, 2006.

[16] Paul, Bipul C., Kunhyuk Kang, Haldun Kufluoglu, Muhammad Ashraful Alam, and Kaushik Roy. "Temporal performance degradation under NBTI: Estimation and design for improved reliability of nanoscale circuits." In *Proceedings of the conference on Design, automation and test in Europe: Proceedings*, pp. 780-785. European Design and Automation Association, 2006.

[17] Vattikonda, Rakesh, Wenping Wang, and Yu Cao. "Modeling and minimization of PMOS NBTI effect for robust nanometer design." In *Proceedings of the 43rd annual Design Automation Conference*, pp. 1047-1052. ACM, 2006.

[18] Mahapatra, Nihar R., SRIRAM V. Garimella, and A. L. W. I. N. Tareen. "An empirical and analytical comparison of delay elements and a new delay element design." *VLSI, 2000. Proceedings.*, IEEE Computer Society Workshop on VLSI. IEEE, 2000.

[19] Keane, John, Tae-Hyoung Kim, and Chris H. Kim. "An on-chip NBTI sensor for measuring PMOS threshold voltage degradation." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 18*, no. 6 (2010): 947-956.

[20] Singh, Prashant, Eric Karl, Dennis Sylvester, and David Blaauw. "Dynamic nbti management using a 45 nm multi-degradation sensor." *IEEE Transactions on Circuits and Systems I: Regular Papers 58*, no. 9 (2011): 2026-2037.

[21] Sakurai, Takayasu, and A. Richard Newton. "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas." *IEEE Journal of solid-state circuits 25*, no. 2 (1990): 584-594.

[22] Wang, Wenping, Zile Wei, Shengqi Yang, and Yu Cao. "An efficient method to identify critical gates under circuit aging." In *2007 IEEE/ACM International Conference on Computer-Aided Design*, pp. 735-740. IEEE, 2007.

[23] Plasma Project, `http://opencores.org/project,plasma`.

[24] OpenRISC Project, `http://http://openrisc.io/`.

[25] Gautschi, Michael, Michael Muehlberghuber, Andreas Traber, Sven Stucki, Matthias Baer, Renzo Andri, Luca Benini, Beat Muheim, and Hubert Kaeslin. "SIR10US: A tightly coupled elliptic-curve cryptography co-processor for the OpenRISC." In *2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors*, pp. 25-29. IEEE, 2014.

[26] PULP platform Project, `http://www.pulp-platform.org/`.

[27] Landman, Paul E., and Jan M. Rabaey. "Architectural power analysis: The dual bit type method." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 3*, no. 2 (1995): 173-187.

[28] Subodh Gupta, Farid N. Najm, "Power macromodeling for high level power estimation", *DAC-34: 34th ACM Desing Automation Conference*, pp. 365-370, 1997.

[29] Bogliolo, Alessandro, Roberto Corgnati, Enrico Macii, and Massimo Poncino. "Parameterized RTL power models for soft macros." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 9*, no. 6 (2001): 880-887.

[30] L. Zhang, R. P. Dick, "Scheduled Voltage Scaling for Increasing Lifetime in the Presence of NBTI," *ASPDAC'09*, Jan. 2009, pp. 492–497.