Fine-Grained Aging Prediction Based on the Monitoring of Run-Time Stress Using DfT Infrastructure*

(Invited Paper)

Abhishek Koneru[‡], Arunkumar Vijayan[†], Krishnendu Chakrabarty[‡], and Mehdi B. Tahoori[†]
[‡]Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA
[†]Karlsruhe Institute of Technology, Germany
Email: abhishek.koneru@duke.edu[‡], arun.v@kit.edu[†], krish@duke.edu[‡], mehdi.tahoori@kit.edu[†]

Abstract—**Run-time solutions based on real-time monitoring and adaptation are required for resilience in nanoscale integrated circuits as design-time solutions and guard bands are no longer sufficient. Bias Temperature Instability (BTI)-induced transistor aging, one of the major reliability threats in nanoscale VLSI, degrades path delay over time and may eventually induce circuit failure due to timing violations. Chip health monitoring is, therefore, necessary to track delay changes on a per-chip basis. Chip-monitoring techniques based on actual measurement of path delays can only track a coarse-grained aging trend in a reactive manner. In this paper, we show how the on-chip design for test (DfT) infrastructure can be reused in order to perform fine-grain workload-induced stress monitoring for accurate aging prediction. The captured stress information is fed to a prediction model in real-time. The prediction model is trained offline using support-vector regression and implemented in software. This approach can leverage proactive adaptation techniques to mitigate further aging of the circuit by monitoring aging trends. Simulation results for realistic open-source benchmark circuits highlight the accuracy of the proposed approach.**

I. Introduction

Design-time solutions and guard bands for resilience are no longer sufficient for integrated circuits (ICs) fabricated at nanoscale technology nodes. There is a need for run-time solutions based on real-time monitoring and adaptation. Embedded sensors (thermal, power, performance, delay, etc.) in today's ICs provide an enormous amount of data that can be used for real-time adaptation during system operation. Chip manufacturers incorporate dynamic adaptation strategies such as voltage/frequency scaling, biasing, and thermal management in response to slow-down due to aging, high temperature, current surge, process variations, leakage, etc., but the *adaptation policies* are *static*. The decisions taken in response to system behavior are hard-coded, e.g., in look-up tables, boot ROM, firmware, etc.; hence, todays adaptation methods are more reactive than predictive, and there is no solution available to train the adaptation policy dynamically in response to changes in chip behavior. In many critical applications, it is important to predict system state so that we can take countermeasures before it is too late: a failure occurs and the system crashes. Thus, there is a need for predictive and proactive adaptation methods.

Each chip, due to process variations, is born with a unique personality ("nature"), and because of operating conditions, environment, and workload, grows uniquely ("nurture"). Our proposed System Physician On-Chip (SPOC) infrastructure is illustrated in Fig. 1. SPOC is motivated by the need to guarantee that each system, despite different nature and nurture, has an acceptable behavior ("resilience"). Resilience has been defined as the persistence of performance level that can justifiably be trusted in the presence of change. Hence static solutions that rely on predetermined adaptation strategies cannot provide resilience as systems evolve with time. SPOC is focused on data-driven techniques for guiding dynamic adaptation policies.

Recently, an aging-aware representative path-selection method was presented [1]; this approach uses the delay of a small set of paths to infer the delay of a larger pool of paths that are likely to fail due to transistor aging. Moreover, since aging is affected by process variations and runtime variations in temperature and voltage, machine learning and linear algebra were used to incorporate these variations during representative path selection. In addition, it was shown how delays induced by on-chip voltage droop can be predicted using support-vector machines (SVM) [2]. This prediction technique facilitates effective dynamic frequency scaling based on accurate and real-time voltage-droop prediction.

In this paper, we show how the on-chip design for test (DfT) infrastructure can be reused in the SPOC framework in order to perform fine-grain workload-induced stress monitoring for accurate aging prediction. This approach involves the ranking of workloads or workload phases based on their aging-stress severity. This ranking can then be used to activate aging mitigation techniques proactively. During the lifetime operation of the chip, the DfT infrastructure can be activated periodically to aggregate the system workload information in the form of signatures. The predictor based on machine learning can then map this aggregated information to the amount of aging that has occurred in the circuit. Note that the aging computation and prediction in this work are made on the basis of aging projection. The circuit is simulated with a representative workload for a short time period and the extracted behavior of the circuit is then extrapolated to estimate the aging trends. Then these workloads are ranked based on estimated aging trends.

It has been shown in the literature that the impact of workload on aging can be computed on the basis of the gate-level signal probabilities that can be attributed to it [3]. However, it

*This research was supported in part by the Semiconductor Research Corporation under contracts 2502 and 2503.

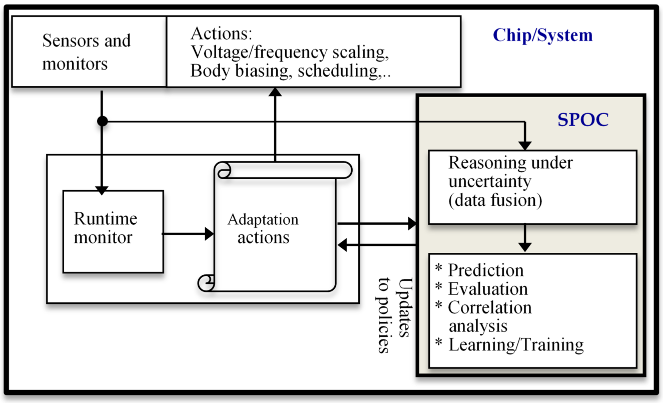


Figure 1. Learning-driven dynamic policy adaptation in SPOC.

is computationally impractical to simulate large designs at the gate-level on a cycle-by-cycle basis for realistic workloads. We have shown in this paper that, to rank workloads in terms of severity of aging, we can also use the state of the flip-flops as a surrogate measure of signal activity. For the task of workload ranking (in terms of aging severity), we have obtained high correlation between the gate-level signal activity and the state of the flip-flops. The monitoring of flip-flops is computationally less burdensome since not all gates in the designs need to be monitored. However, a strategy that relies on the monitoring of all the flip-flops is also not scalable for very large designs. Therefore, we have considered a further level of abstraction, where we only monitor a signature derived from the flip-flops by infrequent sampling. We have shown the workload ranking based in these signatures is highly correlated with the ranking based on the state of the flip-flops, as well as the ranking based on signal probability at the gate level.

The key benefits of the proposed aging prediction method are listed below: (i) **Proactive**: the aging trend is predicted before a measurable delay degradation happens, hence countermeasures can be considered in a timely manner. (ii) **Low overhead**: since the proposed technique uses the existing DfT infrastructure, it imposes minimal area and power overhead. (iii) **Accurate**: the simulation results based on benchmark circuits show that the correlation coefficients of workload rankings based on actual aging trends versus predicted aging trends is extremely high (higher than 0.9, where 1.0 indicates perfect correlation).

The rest of the paper is organized as follows. Section II overviews related prior work. Section III describes the overall methodology underlying aging monitoring and prediction. Section IV presents the method used for designing and training the prediction model. Section V explains the online stress-monitoring technique used for aging prediction. Experimental results are presented in Section VI. Finally, Section VII concludes the paper.

II. Related Prior Work

Worst-case guard-bands have been used in industry for many years. Designers use this conservative approach to ensure that circuits will operate under worst-case temperature, voltage, and workload conditions [4][5]. However, a major problem with guard bands is that uncertainties in input signal probabilities may lead to considerable prediction error [6], and worst-case assumptions are too pessimistic [7][8]. In addition, similar devices may age differently even for the same environmental and workload conditions, which makes aging in the field even less predictable. Finally, guard bands cannot keep up with aging challenges in newer technologies [9][10].

An alternative approach for achieving resilience is referred to as on-line circuit failure prediction. This approach predicts the occurrence of a failure before errors actually occur [7]. Prediction requires information collection in real time on temperature, signal activity, signal delay, IR-drop, etc, and analysis of the data. Such information is usually collected through ring oscillators, temperature sensors, delay sensors, and special circuit structures. Circuit failure prediction can be used to take actions to prevent the chip from failing, and these actions are collectively referred to as on-line self-healing [11][12].

Dynamic reliability management (DRM) techniques were proposed in [13][14]. In [15], the processor uses runtime adaptation to respond to changing application behavior to maintain its lifetime reliability target. In [16], architectural-level models were developed for lifetime-reliability-aware analysis of applications and architectures. In [17], selective redundancy is applied at the micro-architectural level. The method proposed in [18] slows aging through application scheduling and voltage changes at key times.

Dynamic adaptation is often implemented through the use of embedded lookup tables (LUTs). The adaptation policies are encoded and stored in the LUTs, but they are predetermined at design time. A representative LUT-based adaptation policy was proposed in [19]. When a macro changes its state from standby to active mode, the power-management unit fetches a codeword from the LUT, which is provided as input to the adaptive body-bias controller. Another LUT-based technique adjusts the power-supply voltage and body bias to compensate for aging [20]. In [21], a LUT stores, on a case-by-case basis, the optimum values of body bias, supply voltage, and clock frequency to compensate for droop and temperature variations. In [22], a built-in proactive tuning (BIPT) system was proposed, based on a canary circuit that generates predictive warning signals. In [23], the authors proposed control policies to achieve better energy efficiency and lower cost than worst-case guard-banding. Dynamic cooling was introduced as an additional tuning parameter.

A number of the above methods have been adopted for industrial circuits. In [24], the design of a Texas Instruments 3.5G baseband and multimedia applications processor is presented. This SoC consists of multiple independently controlled power domains that use dynamic voltage/frequency scaling and adaptive voltage scaling. In addition, it implements adaptive body biasing [21]. In [25], runtime adaptation techniques are described for Intel's Itanium architecture microprocessor. The core supply voltage and clock frequency are dynamically modulated in order to maximize performance within the power envelope [26]. In [27], the SmartReflex power-management techniques implemented on the OMAP3430 Mobile Multimedia Processor are presented. Active power reduction is achieved through aggressive voltage/frequency scaling and process compensation.

To enable adaptive aging mitigation, online monitoring of circuit degradation is required. State-of-the-art monitoring methods include in-situ sensors [28][7], tunable replica circuits [29], and representative critical reliability path based monitoring [30]. However, these monitoring methods suffers from three problems: (i) additional hardware or design modifications, which are undesirable due to their associated area and power overhead, (ii) intrinsic conflict between the accuracy of the method and hardware overhead, and (iii) existing techniques cannot track short-term aging trends, since they are designed to capture an aggregated measure of the degradation when a significant delay increase takes place. Therefore, the mitigation techniques based on these monitoring systems are *reactive* rather than *proactive*.

Recently, proactive mitigation of BTI has been advocated as a promising approach that can be more effective than reactive techniques [2]. However, designing a suitable BTI-aware delay monitoring for this purpose is challenging for two reasons: (i) it is impractical to accurately measure the delay degradation of a circuit over a short time period. This is because traditional delay monitoring sensors track path delays, and the path-delay degradation for a short time period is too small for these sensors to capture; (ii) degradation rate depends on the currently running workload and working conditions.

III. Reusing DfT for Fine-Grain Aging Monitoring

The proposed aging prediction method is based on tracking the severity of the workload-induced run-time stress. This information can be used to guide fine-grained proactive aging mitigation policies. Workload, which is the main contributor to stress, is captured during run-time and then fed into the prediction software. The output from the aging prediction software can be used to proactively actuate aging mitigation measures. Details about aging mitigation measures are not presented in this paper since the focus here is on aging prediction.

The proposed aging prediction mechanism can be divided into two parts: (1) run-time stress monitoring, (2) feeding the prediction software with the captured data. In this work, on-chip hardware is used to track run-time stress. However, the prediction software is trained offline at design-time and then deployed in the system. Therefore, the effectiveness of the prediction software depends on the training data and the implementation method. The training data is generated by compacting different representative workloads in the form of signatures and mapping the impact of these workloads on the aging trend. The goal here is to characterize the circuit delay in terms of different workloads and train a prediction model based on the data generated from these simulations. The trained prediction model can be implemented in either hardware or software.

IV. Design-time Characterization

During design time, the effect of workload on aging-induced delay is analyzed. A set of representative workloads are applied to the circuit and the signal probabilities of all nodes are calculated for each workload. The amount of circuit delay is projected by assuming the same amount and type of workload over a fixed period of time. The delay of each gate is updated based on the signal probability values and then an aging-aware static timing analysis is performed [1]. In this way, the aging-induced delay corresponding to each representative workload is obtained. Each representative workload is simultaneously compacted to a signature by a functional simulation of a multiple-input signature register (MISR). These signatures together with the circuit delay values are used to train a support vector regression (SVR) model. The goal here is to capture the impact of workload on circuit delay by constructing an analytical aging-prediction model. The flow-chart in Fig. 2 shows the whole process involved in construction of the aging prediction model. In summary, this phase involves (i) estimation of BTI-induced delay degradation and signature extraction, and (ii) construction of an aging prediction model based on support-vector regression.

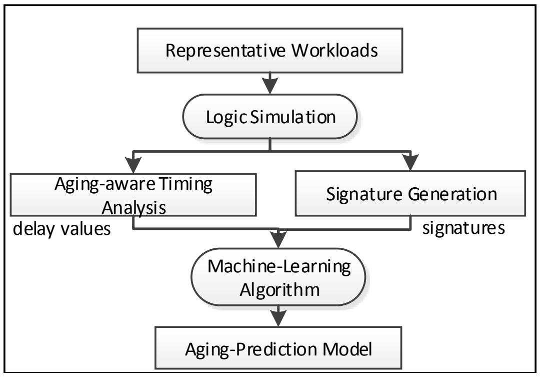


Figure 2. Flow-chart showing the steps involved in training the aging prediction model.

BTI-induced threshold voltage degradation of the transistors in logic gates depends on their input signal-probability values. For each representative workload, signal probabilities of primary inputs are propagated to find the signal probabilities of the internal nodes. This is achieved by annotating the signal probability values of the primary inputs and carrying out a zero delay simulation using Synopsys Power Compiler. The resultant signal probabilities of the internal nodes are extracted from the SAIF file generated by Synopsys Power Compiler. The signal probability values at the inputs of each logic gate are then translated to the threshold-voltage degradation of the transistors within these gates.

A delay look-up table (LUT) is generated by carrying out SPICE characterization of the standard-cell library. For each BTI-induced threshold voltage value, the corresponding delay of the gate is obtained from this LUT. Static timing analysis with the updated gate delays yields an estimate of the aging-induced critical delay of the circuit. Note that STA is block-based and it implicitly considers the delay increase in all possible circuit paths.

In the signature-extraction phase, each representative workload used in the delay-estimation procedure described above is compacted in the form of a MISR signature. The circuit is simulated for each workload and the circuit states are captured by feeding the values in the flip-flops into a MISR using appropriate scan chains.

A. Predictor Training Using Support-Vector Machines

Support-Vector Machine (SVM) is a popular supervised learning algorithm used for pattern recognition [31]-[32]. Though SVM was originally developed for classification problems, it can be easily extended to be applied on regression problems. We have utilized SVM-based regression or SVR because of its high prediction accuracy for a wide range of applications [32].

For training an SVM-based predictor, the input vectors are mapped into a high-dimensional feature space and an optimal hyperplane (prediction function) is constructed in this space. Prediction for new input vectors are made using this function. The SVM-based predictor has been trained using a set of MISR signatures and the corresponding delay values collected from the aging-aware timing analysis framework described in the previous section. This trained predictor is then used for run-time aging prediction. Let $(X_i, Y_i)_{i=1}^{S}$ denote the training set, where $x_i \in \mathbb{R}^d$, and $y_i \in \mathbb{R}$. The training set consists of S input vectors (MISR signatures) $x_1, x_2, ..., x_S$, and each input vector has d features and a corresponding target value (delay) y_i which is a real number. In this work, the ϵ-SVR technique has been used. The objective of ϵ-SVR is to find an optimal hyperplane (i.e. a regression line) that fits most points with in ϵ-margin ($\epsilon > 0$). The regression line has the form: $f(x) = w^T\phi(x) + b$, where $\phi(x)$ is a fixed feature space transformation and b is the bias parameter.

The regression function is written in terms of the kernel function and the Lagrange multipliers as follows:

$$f(x) = \sum_{i=1}^{S} \beta_i k(x, x_i) + b,$$
$$b = \frac{1}{S}\sum_{i=1}^{S}(y_i - \frac{\beta_i}{|\beta_i|}\epsilon - (\sum_{j=1}^{S}\beta_i k(x_i, x_j))) \qquad (1)$$

Using data consisting of MISR signatures and the corresponding delay values, a regression function similar to the one shown in (1) can be trained. This trained model is used for run-time prediction in our work. The run time of the prediction software is dependent on the choice of the kernel. Using simple kernels such as the linear kernel and the polynomial kernel makes the prediction process faster but they tend to under-fit complex data sets. On the other hand, non-linear kernels such as the *radial basis function* (RBF) kernel and the sigmoid kernel fit complex data sets very well but require higher run times. In our method, the polynomial kernel has been used because the prediction accuracy was found to be sufficiently high and computation times were very low.

To illustrate the SVR methodology, consider a hypothetical case with five training samples, and let each sample consist of a signature and the corresponding circuit delay. We form the training set as a matrix $\mathcal{A} = [\mathcal{B}|\mathcal{C}]$ as shown below:

$$\mathcal{A} = [\mathcal{B}|\mathcal{C}] = \left[\begin{array}{ccccc:c} 0 & 1 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0 & 1.0 \\ 1 & 0 & 1 & 0 & 0 & 0.1 \\ 0 & 1 & 1 & 0 & 1 & 0.7 \\ 0 & 1 & 1 & 0 & 1 & 0.7 \end{array}\right], \qquad (2)$$

where the left part ($\mathcal{B}$) in each row corresponds to signature and the value in ($\mathcal{C}$) corresponds to the circuit delay.

We obtain the Lagrange multipliers $\beta_1 = -0.149$, $\beta_2 = 0.287$, $\beta_3 = -0.275$, $\beta_5 = 0.137$ for 4 support vectors, by solving the optimization function using a linear kernel. We get $b = 0.613$ by substituting the Lagrange multipliers in (1). Therefore, the regression model for predicting the delay for a given signature is generated as follows:

$$\begin{aligned} f(\boldsymbol{x}) &= -0.275 \cdot x_1 - 0.012 \cdot x_2 - 0.138 \cdot x_3 \\ &\quad +0.287 \cdot x_4 + 0.137 \cdot x_5 + 0.613 \end{aligned} \qquad (3)$$

Suppose we have an input signature $[01101]$, which is the fifth row in (2). The circuit delay is evaluated to be $y = 0.60$ using (3). Let us consider a second signature $[01010]$. In this case, the regression function evaluates to $y = 0.888$.

V. Run-time Stress Monitoring

The existing DfT infrastructure on chip can be used to dynamically capture the system workload. The predictor implemented on chip can translate workloads signatures of recent workload to corresponding aging stress. We prefer a software implementation of the predictor for the following reasons: (1) hardware implementation results in area overhead, (2) aging monitoring does not require cycle-by-cycle observation using on-chip hardware.

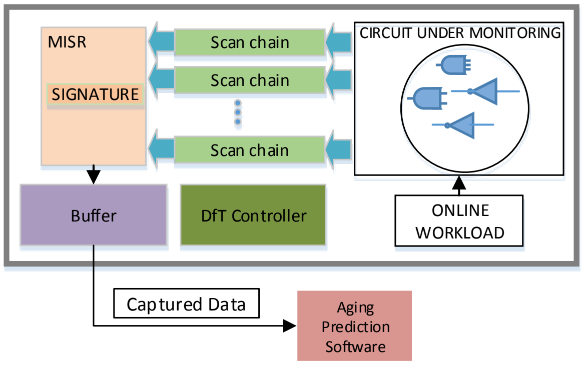


Figure 3. DfT infrastructure used to capture the chip workload.

A. Hardware Architecture

The DfT infrastructure employed to aggregate the system workload in the form of signatures is shown in Fig. 3; it includes the scan chain framework and a MISR. Circuits that employ logic built-in self test (BIST) or test compression are likely to have an on-chip MISR, which can be reused for generating signatures in the field at no additional hardware overhead. The DfT controller, which is a simple finite-state machine (FSM), periodically switches the circuit into scan mode to capture the circuit state. The contents of the scan chains are then shifted out to the MISR and compacted as signatures. However, this operation would overwrite the state of the flip-flops and the circuit cannot resume normal operation after it is completed. Therefore, to maintain the state of the flip-flops, the bits that are being shifted out from the scan chain are fed back into it through a multiplexer, as shown in Fig. 4. This multiplexer, which is also controlled by the DfT controller, is used to switch between the normal scan mode and the scan mode in order to maintain the circuit state during run-time monitoring.

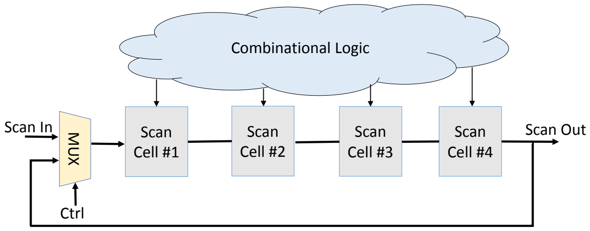


Figure 4. Modified scan chain architecture to preserve the state of flip-flops after the shift-out operation.

The number of clock cycles required for shifting out the state of the flip-flops depends on the number of flip-flops in the scan chain. For these clock cycles, the normal execution of the system is interrupted and this can lead to significant performance penalty. However, the performance penalty incurred during this signature-generation phase can be reduced by time-sampling the workload instead of capturing it in every clock cycle. In time-sampling, uniformly distributed samples of the workload are used to generate the signatures. For example, x% time-sampling implies that workload is sampled at x% of the total number of clock cycles that are uniformly chosen. In this way, the impact on the normal functionality of the system becomes negligible. Moreover, time-sampling also reduces the power overhead incurred during signature generation. Our results show that low rates of time sampling do not affect the accuracy of aging prediction. The above steps constitute the signature generation phase, and after this, the system can continue its normal execution. The extracted signatures are stored in a buffer and then fed to the prediction software.

B. Hardware-Software Interface

There are two methods that can potentially be used to implement the prediction software. In the first method, the predictor is executed as a thread on any idle core on-chip. The role of this software thread is to collect signature and temperature sensor data from every core on-chip. These data, which are stored in a buffer on every core (Fig. 3), are transferred from one core to another based on a *handshake* mechanism. The core on which the thread is executing broadcasts a read signal to all the other cores. The core that is ready for this read operation sends back an acknowledgement (permission to read). Once the core executing the thread receives an acknowledgement from another core, it starts reading data from the buffer on that core and stores them in its buffer. In this way, data from all the cores are collected before the aging prediction process is started. This method of executing the prediction software does not require any additional hardware. However, an idle core may not always be available, in which case the system operation has to be interrupted to execute the prediction software. Moreover, migrating the predictor between different idle cores also involves overhead.

The other method is to execute the prediction software on a dedicated programmable microcontroller. The microcontroller communicates with every core on-chip to obtain signature and temperature sensor data. Therefore, it is necessary to define the interface between the microcontroller and the on-chip hardware. In [33], the communication between the energy management microcontroller and the processor core occurs through through the industry-standard Inter-Integrated Circuit (I^2C) interface. The microcontroller can read and write to the registers on every on-chip core through this interface. Therefore, the signature and the sensor data can be accessed by the microcontroller through the I^2C interface using buffer read operations. This method of implementation of the prediction software does not interrupt the processes running on any core, and therefore, has minimal performance overhead. However, additional hardware cost is incurred to implement the dedicated microcontroller on-chip.

VI. Results

A. Experimental Setup

Experiments were performed on two open source processor benchmarks, namely OpenRISC 1200 (OR1200) and Leon3, and on four ISCAS'89 benchmarks, to evaluate the accuracy of the proposed technique. OR1200 is five-stage pipeline embedded processor based on the 32-bit ORBIS32 instruction set architecture (ISA). Leon3 is a 32-bit processor based on the SPARC-V8 RISC ISA.

The benchmarks were synthesized using Synopsys Design Compiler with Nangate 45nm library [35]. Six programs from the MiBench embedded systems benchmark suite [34], shown in Table I, were executed on these processors using Mentor Graphics ModelSim. Each MiBench workload used was divided into several smaller workloads to collect required number of workloads for training. The state of the flip-flops was captured at regular but infrequent intervals; we considered time-sampling at a frequency of 0.1%, 0.01%, and 0.001% of the total number of cycles in a workload. Workloads constituting the training set are used to construct a prediction model and workloads constituting the validation set are used to evaluate the accuracy of the prediction model. The training and validation set of a benchmark together constitutes the total data points used in aging estimation.

The workload-signature corresponding to each workload was extracted using a MISR implemented in C++. The size of the MISR was chosen to be 64 bits. The flip-flops in the benchmarks can be divided into 64 scan chains such that there are not too many flip-flops in one scan chain. If the number of flip-flops in one scan chain is high, then the number of clock cycles required to shift-out the state of the flip-flops will also be high. On the other hand, increasing the number of scan-chains increases the size of the MISR, and therefore, more area is required to implement it.

An in-house aging-aware static timing analysis framework [1], was used to estimate the impact of workload on the benchmark processor delay after a fixed period of time. SVR algorithms used to train and validate the aging predictor were implemented using the MATLAB interface for the LibSVM software package [36]. Experiments were run on a 64 bit Windows machine with 12 GB of RAM and quad-core Intel i7 processors running at 2.67 GHz.

Table I
DETAILS OF THE MIBENCH BENCHMARKS USED IN THIS WORK [34]

Benchmark Name	Bitcount	Qsort	CRC32	Stringsearch	Basicmath	Susan (smoothing)
Application	Industrial	Industrial	Network	Office	Industrial	Industrial
Total number of simulation cycles on OR1200	10.6 million	55.5 million	17.0 million	4.2 million	10 million	60 million
Total number of simulation cycles on Leon3	7.7 million	23.6 million	3.8 million	2.5 million	6.6 million	22.2 million

B. Validation Experiments

A prediction model was trained and validated for each benchmark processor based on a data set consisting of workload-signatures and the projected delay values obtained after executing these workloads at a fixed point of time in the future. The training data set for each processor consisted of 1,000 workload-signatures and the corresponding projected delay values. The remaining 1,000 workload-signatures and the corresponding projected delay values were used to validate the prediction model.

The best value for the regularization parameter and kernel parameters for the kernel used for SVR were determined using a five-fold cross-validation approach. In this approach, the training set was divided into five equal subsets and each subset was validated using a model trained on the remaining four subsets. A grid search was carried out on the parameters and the parameter values corresponding to the highest cross-validation accuracy were chosen. An SVR model was then trained using these best parameter values and the complete training set.

C. Correlation Results

The accuracy of the proposed method is evaluated using the Kendall's rank-correlation coefficient, defined by (4).

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{4}$$

where n_c is the number of concordant pairs, n_d is the number of discordant pairs, and n is total number of workloads.

Let (a_1, p_1), (a_2, p_2), ...,(a_n, p_n) be a set of actual and predicted delay values for n workloads. Any pair of observations (a_i, p_i) and (a_j, p_j) are said to be *concordant* if the ranks for both elements agree: that is, if both $a_i > a_j$ and $p_i > p_j$ or if both $a_i < a_j$ and $p_i < p_j$. They are said to be *discordant*, if $a_i > a_j$ and $p_i < p_j$ or if $a_i < a_j$ and $p_i > p_j$.

Workloads are ranked based on their impact on aging. The workload that results in a higher delay degradation of the circuit has a lower integer value as rank than a workload causing a lower delay degradation of the circuit. We obtain two sets of ranks, the first set from the actual delay degradation values extracted from aging-aware STA and the second set from the predicted delay degradation values using the SVR prediction model. A value of 1 for τ shows a perfect correlation between the workload rankings obtained from actual delay values and predicted delay values. Higher the value of τ ($-1 \leq \tau \leq 1$), the higher is the correlation.

The aging prediction accuracy of the prediction model varies with training and validation set size for OR1200 and Leon3 as shown in Fig. 5. The results are presented for 0.1%, 0.01%, and 0.001% time-sampling. The Kendall's rank-correlation coefficients (τ) for OR1200 and Leon3 for 800 data points exceed 0.9. Note that τ nearly remains constant with a further increase in the number of data points. Hence we use a total of 2000 data points in our experiments. In this scenario, we obtain $\tau \approx 1$ for OR1200. Note that the values of τ are rather small when the number of data points used for training is less than 400 (600) for OR1200 (Leon3). The problem of determining a sufficient number of data points for training remains an interesting open problem. At this time, we advocate the use of as many data points as is computationally realistic.

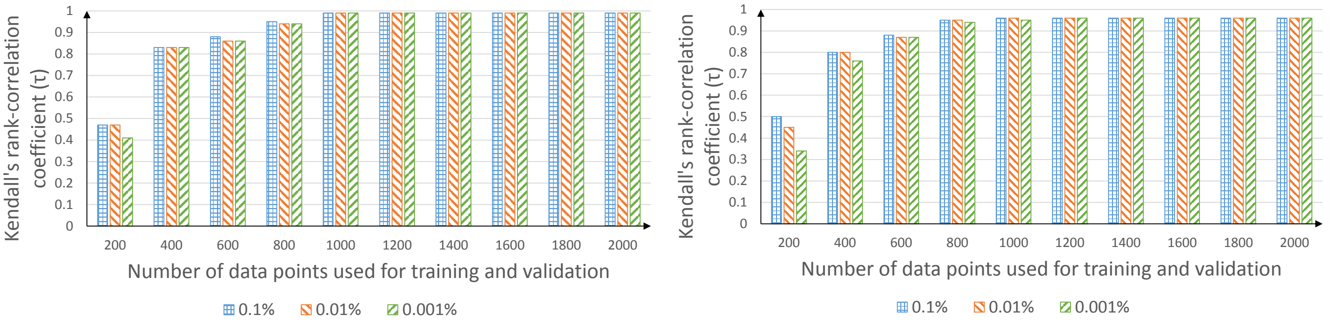
We further analyze the effect of time-sampling rates on prediction accuracy corresponding to 2000 data points in Fig. 5. The prediction model trained with signatures captured from 0.1% time-sampling was used to predict aging trends from signatures captured from 0.1% time-sampling. For both OR1200 and Leon3, 0.1% time-sampling leads to a highly accurate prediction ($\tau = 0.9901$ and 0.9652, respectively). As the sampling rate decreases to 0.001%, prediction accuracy decreases only slightly ($\tau = 0.9897$ and $\tau = 0.9500$, respectively). This shows that accuracy is not significantly affected with lower time-sampling rates.

D. Step-by-step correlation

The increase in threshold voltage of a transistor due to BTI depends on the signal probability at the gate terminal of that transistor. For accurate aging calculation, we require signal probability values for all internal nodes of the circuit netlist. Instead of monitoring the signal probability values of the internal nodes directly, we use MISR signatures to predict the aging trend. MISR signatures are generated by capturing flip-flop states at a particular time sampling rate. The MISR signature is a function of flip-flop states across clock cycles and the flip-flop values across clock cycles is a function of the signal probability values of internal nodes. There are different stages of compaction, from internal signal probability values to flip-flop states and from flip-flop states to MISR signatures. Our objective is to show that, even after two stages of lossy compaction of aging data, we are still able to predict the aging trend with significant accuracy.

Table II
KENDALL'S RANK-CORRELATION COEFFICIENTS (τ) CORRESPONDING TO THE PREDICTIONS BASED ON NODE-LEVEL, FLIP-FLOP-LEVEL AND MISR-LEVEL DATA

Benchmark	SP	FF	MISR
s953	0.9868	0.9868	0.9868
s1196	1.0000	1.0000	1.0000
s838	0.8578	0.8556	0.8578
s1238	1.0000	1.0000	1.0000



(a) OR1200

(b) Leon3

Figure 5. Kendall's rank-correlation coefficient (τ) for OR1200 and Leon3 processors for different training and validation set sizes for 0.1%, 0.01% and 0.001% time-sampling rates.

To evaluate the correlation between node signal probabilities, flip-flop states, and MISR signatures, we have to carry out detailed gate-level simulation and record signal values for each clock cycle. The training time for an SVR model scales between quadratic and cubic with respect to the number of training samples and the number of input features [37]. We determined the actual run time for training for 10 input features to be 0.0625 seconds. For OR1200 (Leon3), having 37276 (35399) gates, the number of input features is 456150 (390080) for 0.001% time-sampling and the estimated run time is 4.12 (3.01) years, assuming time complexity of O(n^2) for an SVR model [37]. This is computationally impractical, hence we compute these correlations for smaller ISCAS'89 benchmarks. If consistently high correlations are obtained for the smaller designs, we can expect similar high correlations for OR1200 and Leon3. Note that these correlations, which we are calculating here to highlight accuracy, do not need to be computed during workload ranking using the proposed method.

The correlation results in terms of workload ranking are shown in Table II. The three columns show correlation coefficients (τ) for different benchmarks based on aging-stress severity calculated from: (i) signal probability values of internal nodes (SP); (ii) flip-flop values for clock cycles at 0.001% time-sampling (FF); (iii) MISR signatures (MISR). The ranking of workloads based on actual delay values from aging-aware STA and the ranking of workloads based on predicted delay values from SVR prediction model are correlated here. The high correlation values show that MISR signatures contain significant aging information to predict aging even after these two stages of compaction.

E. Characteristics of signatures

MISR signatures across workloads were evaluated for their uniqueness. Signatures across workloads used in learning and prediction for the two processor benchmarks were analyzed. We found that the 64-bit signatures were all mutually distinct; in other words, the signatures were unique. Hence, we conclude that no two workloads for a benchmark used in learning and prediction have the same signature, and there is negligible bias in the results that we are reporting.

The variation in prediction accuracy with signature size, i.e; the width of the MISR was analyzed. The signature size was varied to capture the aging impact of the same set of workloads. For this experiment, we used four different MISR sizes: 64 bits, 32 bits, 16 bits, and 8 bits. As signature size was decreased from 16 bits to 8 bits, prediction accuracy was found to decrease significantly for both benchmarks. This is expected because a smaller MISR leads to more information loss. The results are shown in Table III. In other words, when the signature is large enough, the MISR signature can accurately capture aging information. Thus we conclude that larger MISRs are desirable.

Table III
KENDALL'S RANK-CORRELATION COEFFICIENTS (τ) SHOWING THE CORRELATION BETWEEN PREDICTED AND ACTUAL AGING-SEVERITY RANKING OF WORKLOADS WITH DIFFERENT MISR SIZES

Benchmarks	MISR size			
	64 bits	32 bits	16 bits	8 bits
OR1200	0.9901	0.9836	0.9819	0.7630
Leon3	0.9652	0.9547	0.9471	0.7739

The aliasing probability of MISR signatures that are obtained from flip-flop-state vectors was analyzed. Aliasing occurs when two or more flip-flop-state vectors are mapped to the same MISR signature. Suppose there are M state vectors mapped to N MISR signatures ($N < M$). Let the signatures be denoted as S_1, S_2, S_3, ..., S_N. The aliasing probability can be defined as the probability of any two randomly selected state vectors getting mapped to the same MISR signature and can be expressed as in (5):

$$P_A = \frac{\sum_{i=1}^{N} \binom{q_i}{2}}{\binom{M}{2}}, \tag{5}$$

where P_A is the aliasing probability and q_i is the number of state vectors that map to signature S_i. The change in aliasing probability with the MISR signature size is shown in Table IV. The aliasing probability increases by two orders of magnitude when the MISR size is decreased from 16 bits to 8 bits. In Table III, the effect of aliasing can be observed as the significant drop in τ when the MISR size is reduced from 16 bits to 8 bits. For larger MISR sizes, the aliasing probability is zero and very accurate workload rankings are obtained.

Table IV
ALIASING PROBABILITY (P_A) IN THE MAPPING OF FLIP-FLOP-STATE VECTORS TO DIFFERENT MISR SIZES

Benchmarks	MISR size			
	64 bits	32 bits	16 bits	8 bits
OR1200	0.000000	0.000000	0.000019	0.003932
Leon3	0.000000	0.000000	0.000014	0.003861

F. Run-time and Overheads

Since the proposed technique is based on reusing the existing DfT infrastructure on-chip, it introduces minimal area overhead. Power calculations were carried out using Synopsys Design Compiler. The power overhead of the proposed method was 0.0048% and 0.0008%, for OR1200 and Leon3, respectively. The results were obtained with 0.01% time-sampling. We also note that, as expected, the power overhead was found to increase in a linear fashion with time-sampling rate.

As discussed in Section III, the proposed technique consists of the design-time construction of the aging prediction model using machine learning, and run-time evaluation. The CPU time for the design-time phase is only couple of hours for Leon3 or OR1200 processor benchmarks. Note that the design-time phase is only performed once for each circuit. SVR model construction only involves simple software-based vector multiplication and summation. Note also that the status of flip-flops are pushed through scan chains to the MISR at fixed sampling intervals. The performance overhead depends on the number of clock cycles required for each sampling moment, which in turn depends on the size of each scan chain. We can reduce the performance overhead by tracking the aging rate less frequently. For example, while a typical processor runs at GHz frequency, aging occurs far more slowly, and sampling of the flip-flops can be done at rates that are several orders of magnitude smaller.

VII. CONCLUSION

We have proposed an approach for aging-induced delay prediction based on reusing the existing DfT infrastructure. Unlike today's state-of-the-art based on hardware sensors, our method imposes minimal area and power overhead since we leverage on-chip hardware and use a software implementation of the prediction function. This method also makes it possible to capture fine-grained aging trends that can support proactive aging mitigation techniques. Simulation results for open-source processor designs demonstrate that the proposed approach can accurately predict workload-induced aging trends.

REFERENCES

[1] F. Firouzi, F. Ye, K. Chakrabarty, and M. B. Tahoori, "Aging- and variation-aware delay monitoring using representative critical path selection," *TODAES*, vol. 20, no. 3, pp. 39:1–39:23, 2015.

[2] F. Ye *et al.*, "On-chip voltage-droop prediction using support-vector machines," in *VTS*, 2014, pp. 1–6.

[3] D. Lorenz, G. Georgakos, and U. Schlichtmann, "Aging analysis of circuit timing considering NBTI and HCI," in *IOLTS*, 2009, pp. 3–8.

[4] K. Kang *et al.*, "Efficient transistor-level sizing technique under temporal performance degradation due to NBTI," in *ICCD*, 2007, pp. 216–221.

[5] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Temporal performance degradation under NBTI: estimation and design for improved reliability of nanoscale circuits," in *DATE*, 2006, pp. 780–785.

[6] W. Wang *et al.*, "The impact of NBTI on the performance of combinational and sequential circuits," in *DAC*, 2007, pp. 364–369.

[7] M. Agarwal *et al.*, "Circuit failure prediction and its application to transistor aging," in *VTS*, 2007, pp. 277–286.

[8] D. Sylvester *et al.*, "Elastic: An adaptive self-healing architecture for unpredictable silicon," *Design & Test*, vol. 23, no. 6, pp. 484–490, 2006.

[9] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *Micro, IEEE*, vol. 25, no. 6, pp. 10–16, 2005.

[10] J. W. McPherson, "Reliability challenges for 45 nm and beyond," in *DAC*, 2006, pp. 176–181.

[11] B. Kapoor *et al.*, "Impact of SoC power management techniques on verification and testing," in *ISQED*, 2009, pp. 692–695.

[12] J. W. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *JSSC*, vol. 37, no. 11, pp. 1396–1402, 2002.

[13] E. Karl, D. Blaauw, D. Sylvester, and T. Mudge, "Multi-mechanism reliability modeling and management in dynamic systems," *Transactions on VLSI Systems*, vol. 16, no. 4, pp. 476–487, 2008.

[14] A. Urmanov *et al.*, "A new sensor validation technique for the enhanced RAS of high end servers," in *MLMTA*, 2004.

[15] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," in *SIGARCH*, vol. 32, 2004, p. 276.

[16] J. Srinivasan, S. V. Adve, P. Bose, J. Rivers *et al.*, "Lifetime reliability: Toward an architectural solution," *MICRO*, vol. 25, pp. 70–80, 2005.

[17] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "Exploiting structural duplication for lifetime reliability enhancement," in *SIGARCH*, vol. 33, 2005, pp. 520–531.

[18] A. Tiwari and J. Torrellas, "Facelift: Hiding and slowing down aging in multicores," in *MICRO*, 2008, pp. 129–140.

[19] B. Choi and Y. Shin, "Lookup table-based adaptive body biasing of multiple macros," in *ISQED*, 2007, pp. 533–538.

[20] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "NBTI-aware synthesis of digital circuits," in *DAC*, 2007, pp. 370–375.

[21] J. Tschanz, *et al.*, "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," in *ISSCC*, 2007, pp. 292–604.

[22] N. Shah *et al.*, "Built-in proactive tuning system for circuit aging resilience," in *DFT*, 2008, pp. 96–104.

[23] E. Mintarno *et al.*, "Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging," *TCAD*, vol. 30, no. 5, pp. 760–773, 2011.

[24] G. Gammie, *et al.*, "A 45 nm 3.5G baseband-and-multimedia application processor using adaptive body-bias and ultra-low-power techniques," in *ISSCC*, 2008, pp. 258–611.

[25] T. Fischer *et al.*, "A 90-nm variable frequency clock system for a power-managed Itanium architecture processor," *JSSC*, vol. 41, no. 1, pp. 218–228, 2006.

[26] R. McGowen *et al.*, "Power and temperature control on a 90-nm Itanium family processor," *JSSC*, vol. 41, no. 1, pp. 229–237, 2006.

[27] H. Mair *et al.*, "A 65-nm mobile multimedia applications processor with an adaptive power management scheme to compensate for variations," in *IEEE Symposium on VLSI Circuits*, 2007.

[28] D. Ernst *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *MICRO*, 2003, pp. 7–18.

[29] K. Bowman *et al.*, "Circuit techniques for dynamic variation tolerance," in *DAC*, 2009, pp. 4–7.

[30] S. Wang, J. Chen, and M. Tehranipoor, "Representative critical reliability paths for low-cost and accurate on-chip aging evaluation," in *ICCAD*, 2012, pp. 736–741.

[31] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[33] M. Floyd *et al.*, "Introducing the adaptive energy management features of the POWER7 chip," *MICRO*, vol. 31, no. 2, pp. 60–75, March 2011.

[34] M. R. Guthaus *et al.*, "MiBench: A free, commercially representative embedded benchmark suite," in *WWC*, 2001, pp. 3–14.

[35] "Nangate 45 nm open cell library v1.3," http://www.nangate.com.

[36] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. 27, 2011.

[37] L.-J. Cao *et al.*, "Support vector machine with adaptive parameters in financial time series forecasting," *Transactions on Neural Networks*, vol. 14, no. 6, pp. 1506–1518, 2003.