Hierarchically Focused Guardbanding: An Adaptive Approach to Mitigate PVT Variations and Aging

Abbas Rahimi
CSE Department, UC San Diego
La Jolla, CA 92093, USA
abbas@cs.ucsd.edu

Luca Benini
DEIS, University of Bologna
40136 Bologna, Italy
luca.benini@unibo.it

Rajesh K. Gupta
CSE Department, UC San Diego
La Jolla, CA 92093, USA
gupta@cs.ucsd.edu

Abstract— **This paper proposes a new model of functional units for variation-induced timing errors due to PVT variations and device Aging (PVTA). The model takes into account PVTA parameter variations, clock frequency, and the physical details of Placed-and-Routed (P&R) functional units in 45nm TSMC analysis flow. Using this model and PVTA monitoring circuits, we propose *Hierarchically Focused Guardbanding* (HFG) as a method to adaptively mitigate PVTA variations. We demonstrate the effectiveness of HFG on GPU architecture at two granularities of observation and adaptation: (i) fine-grained instruction-level; and (ii) coarse-grained kernel-level. Using coarse-grained PVTA monitors with kernel-level adaptation, the throughput increases by 70% on average. By comparison, the instruction-by-instruction monitoring and adaptation enhances throughput by a factor of 1.8×−2.1× depending on the configuration of PVTA monitors and the type of instructions executed in the kernels.**

Keywords-adaptive guardbanding; PVT variation; aging; GPU;

I. INTRODUCTION

Variability in microelectronic circuits and systems stems from different physical sources: (i) Static intrinsic process parameter variations (e.g., effective transistor channel length and threshold voltage) result from device dimension and doping concentration variations that occur during silicon fabrication, and are amplified as device dimensions shrink [1]; (ii) Dynamic environmental variations in ambient condition caused by temperature fluctuations and supply voltage droops; (iii) Aging and temporal degradation in device reliability due to Negative Bias Temperature Instability (NBTI), and Hot Carrier Injection (HCI). These factors will worsen in future technologies due to increased fallibility in the process and increased environmental stresses [2]. Designers commonly handle variability issues in synchronous circuits by adding safety timing margin as guardband, which leads to overly conservative designs. This guardband is computed from a corner-based worst-case analysis during the design phase [3]. However, as the relative variation grows with technology scaling [4], adaptive techniques are needed to ensure continued advantages of technology scaling.

Indeed, several recent efforts have focused on measures to mitigate variability through circuit monitors in two broad categories. First, internal or *in situ* monitors such as Razor [5], and Error-Detection Sequential (EDS) [6] that typically use double sampling with shadow latches to detect Process, Voltage, and Temperature (PVT) variations. Intel resilient core [7] integrates EDS in critical paths to detect late transitions. Recently, in [8] a 45-nm decoupled 10-Lane SIMD (Single Instruction, Multiple Data) processor utilizes Razor error detection in the specific context of GPUs. For slower variations, compact *in situ* aging sensors with digital outputs have been proposed to measure NBTI and oxide degradation [9]. A second approach relies on external or replica monitors that are on the same die, but outside of the functional paths. Compared to *in situ* monitors, replica circuits are less intrusive on system operation. Bowman *et al* [7] place a Tunable Replica Circuit (TRC) [10] per pipeline stage to monitor timing errors. In a similar vein, replica Critical Path Monitor (CPM) [11] measures the timing margin available to circuits. IBM 8-core POWER7 employs five CPMs per each core to capture PVT variations and detect early wearout [12]. High-resolution, digital on-chip voltage droop sensors [13] as well as thermal sensors [14] are widely used to measure distinct dynamic variations.

Recovery from errors can be cycle-by-cycle or over many cycles. Once variation is detected in the current cycle, [5], and [6] try to compensate timing error for the activated critical paths by dynamically switching to a two-cycle operation. Next, instruction-by-instruction clock adjustment technique is proposed to handle dynamic variations in [15] that uses a fast single-cycle adaptive frequency circuit [14]; multiple-issue instruction replay design can also correct errant instructions without requiring clock control [7]. Decoupling SIMD queues in [8] prevent error events in any single lane from stalling all other lanes, thus enables each lane to tolerate errors independently. Lanes are only required to resynchronize when a micro-barrier (e.g., load, store instruction) is reached. A bank of aging sensors also includes one 20-bit counter and storage units that allow quick measurements when a stress cycle is interrupted [9]. Going further up on the hardware-software stack, procedure-level [16] as well as task-level [17] techniques are proposed to guarantee error-free operation in case of variations.

A. Contributions

I. We provide a new high-level model for Timing Error Rate (TER) of various integer as well as floating-point functional units that is derived using accurate industrial-strength tools and calibration flows validated in real silicon. This model yields the TER of microarchitectural functional units as a function of clock frequency and the amount of PVT variations and Aging (PVTA). Section III describes the model that can be used both online and offline. Online, it provides a model-based rule to derive guardband from the PVTA sensor readings. Offline, it enables design time analysis to identify vulnerable functional units at a given amount of PVTA variations. The model is publicly available for download at [18].

II. We introduce the notion of *Hierarchically Focused Guardbanding* (HFG) in Section IV to adaptively mitigate PVTA variations. HFG is guided by online utilization of the model, and enables a *focused* adaptive guardbanding in view of monitors, observation granularity, and reaction times.

978-3-9815370-0-0/DATE13/©2013 EDAA

III. We demonstrate the effectiveness of HFG using the proposed model on GPU pipeline at two distinct granularities. HFG enhances the throughput of kernels, on average by 70%, employing coarse-grained PVTA monitors and applying adaptive guardbanding at granularity of kernel-level. The finer granularity of instruction-level monitoring and adaptation achieves 1.8×−2.1× throughput improvements depending on the PVTA monitors configuration and the type of instructions executed within the kernels. Section V details the results.

II. Related Work

[19]−[23] propose partial solutions to model and mitigate a wide range of variability, including DC component (time-independent), low-frequency components (slow-varying), and high-frequency components (fast-changing). VARIUS [19] proposes a microarchitecture-aware model for timing error caused by process variation. To address variations at near-threshold computing, [20] also proposes a microarchitectural model of process variations. To mitigate circuit aging, [21] presents a framework for optimizing dynamic control of self-tuning parameters (such as supply voltage, operating clock frequency, and dynamic cooling) of a digital system over its lifetime. A thermal-aware frequency scaling [22] as well as architectural event-guided method [23] are proposed to avoid dynamic voltage variation. These techniques either solely consider static process variation or only individual dynamic variations. Although various sensors and policies are proposed, the analysis linking PVTA sensor readings and necessary guardband (for a target TER dictated by applications) is preliminary. Further, [22][23] also use "generic" variability models on high-level architectural simulators that do not take into account the effect of physical implementation on variability.

Razor [5], EDS [6], TRC [10], CMP [11] circuit sensors raise a warning signal when a timing error is detected in case of PVT variations. Their common strategy is to allow the timing errors to happen, and then pay extra cost to compensate errors through expanding the window of recoverability or tuning CMOS knobs such as the supply voltage, frequency, or body bias. Their cost of recovery has shown to be high in SIMD- and GPU-specific extensions [24] because an error in any functional unit stalls the entire pipeline, in effect multiplying the baseline error rate by the SIMD width. Although [8] decouples SIMD lanes for recovery, frequent micro-barriers exhibit throughput penalty. Furthermore, these online detect-then-recover mechanisms do not tie to any characterized modeling, thus suffer from lack of correlation between the occurred errors and the sources of variations. This limits their usage for prediction of the timing errors and their root causes at the upper layers for better decision and appropriate adjustment. Thus, improve modeling is needed to connect timing errors with sources of variability for better prediction. The model should be coupled with adaptive resource management to proactively prevent timing error by applying a *focused* guardbanding.

III. Timing Error Model for PVTA

A. Analysis Flow for Timing Error Extraction

To build a parametric model for timing errors, we rely upon design time analysis that yields the TER of individual Functional Units (FUs) as a function of clock period (t_{clk}) and the amount of PVTA variations. We have analyzed a wide range of FUs, listed in [18], that are being used in a rich GPU pipeline, including 10 32-bit integer FUs as well as 15 single precision floating-point FUs fully compatible with the IEEE 754 floating-point standard. The floating-point FUs also cover the transcendental operations, thus act as the special FUs in the GPU pipeline to support sin, cosine, reciprocal, and square root instructions. FUs are selected from *Synopsys DesignWare*, a library of functions for computational circuits in high end ASICs. The speed optimized architectures have been selected for FUs in conjunction with tight synthesis and physical optimizations for timing closure. FUs have been synthesized for TSMC 45nm target, the general purpose process. The front-end flow with normal V_{th} cells uses *Synopsys DesignCompiler* with the topographical features enabled and *Synopsys IC Compiler* for the backend as shown in Figure 1 & Table I.

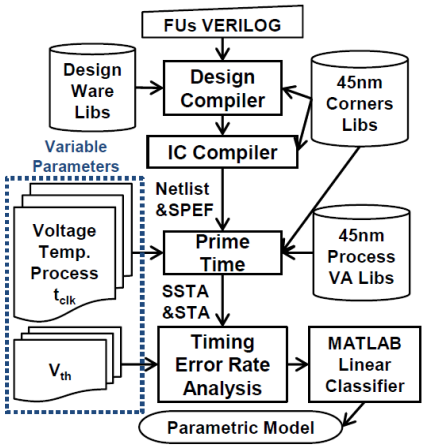

Figure 1. Timing error rate analysis flow for model extraction.

TABLE I. ANALYSIS FLOW: TOOLS AND PARAMETERS

Stage	Tools/Libs	Version/Details
Front-end	Design Compiler	E-2010.12-SP5
Back-end	IC Compiler	E-2010.12-ICC-SP5
Sign-off	PrimeTime VX	F-2011.06-SP3
Libraries	45nm GS TSMC	Variation Aware (v. 110d)
Linear Classifier	MATLAB	Discriminant Analysis (v. R2011b)

For each FU_i working with t_{clk} and a given PVTA variations, Timing Error Rate (TER) is defined in (1):

$$\text{TER}(FU_i, t_{clk}, V, T, P, A) = \frac{\sum \text{CriticalPaths}(FU_i, t_{clk}, V, T, P, A)}{\sum \text{Paths}(FU_i)} \times 100 \quad (1)$$

where CriticalPaths are those paths with a negative slack that cannot meet the setup-time of flip-flops with the clock period of t_{clk} under certain PVTA variations, and Σ Paths is the total number of paths in FU_i. After the back-end optimizations, during the sign-off, we calculate TER by analysis of FU PVTA parameter variations as follows:

Dynamic variations: The full industrial temperature range of 0°C−120°C, and voltage range of 0.88V−1.1V are considered by utilizing six 45nm TSMC characterized sign-off corners by changing these parameters at the resolution of 10°C and 0.01V respectively. To do this, we use the voltage-temperature scaling features of *Synopsys PrimeTime* for the composite current source approach of modeling cell behavior. Then, at each pair of the voltage and temperature, we use Static Timing Analysis (STA) to analyze the critical paths.

Process variation: The device parameters are varied from die-to-die (D2D) as well as within-die (WID), and then Statistical STA (SSTA) is used to report delay variation of each path.

To perform an accurate design time SSTA, we employ the variation-aware timing analysis engine of *Synopsys PrimeTime VX* [25], using process parameters of 45nm variation-aware TSMC libraries [26] derived from first-level process parameters by Principal Component Analysis (PCA). PCA is a mathematical procedure that simplifies a data set by transforming a number of correlated parameters into a smaller number of uncorrelated parameters. Based on [27], the process parameters are varied as normal distributions with zero mean and standard deviations of σ_{D2D}=5% and $\sigma_{WID}\in$[0%, 9.6%]. Therefore, we change the process variation components and examine its induced delay variation with a given set of accurate variability models from commercial libraries. These are more accurate and realistic than commonly used 'in-house models' extracted from predictive technology models.

Aging: Two major mechanisms that induce progressive slowdown are NBTI and HCI, these effects manifest as voltage threshold (V_{th}) shift and gradually slower the critical paths. The delay of critical paths under various dynamic and process parameter variations is reposted by STA and SSTA. To analyze the effect of aging on those paths, their V_{th} is increased, and then their aging-induced delay variation is calculated using the alpha-power law. The V_{th} is increased with steps of 25mV and up to 100mV which can occur over years of stress [9].

Considering the full permutation of PVTA parameters variations, the effects of variability on the delay of a FU is finely captured for its entire lifetime. To observe how this variability can be compensated by adaptive clocking, the t_{clk} is changed from 0.2ns to 5.0ns. Then, TER Analysis module (Figure 1) calculates TER based on t_{clk} and the amount of PVTA variations using Equation (1). Consequently, the calculated TER function of the five variables (summarized in Table II) is input to a parametric linear classifier for model generation.

TABLE II. PVTA AND CLOCK PARAMETERS.

	Start Point	End Point	Step	# of Points
Voltage	0.88V	1.10V	0.01V	23
Temperature	0°C	120°C	10°C	13
Process (σ_{WID})	0%	9.6%	3.2%	4
Aging (ΔV_{th})	0mV	100mV	25mV	5
t_{clk}	0.2ns	5.0ns	0.2ns	25

B. Parametric Model Fitting

We present a parametric model at the level of FU that relates PVTA parameters variation and t_{clk} to TER, thus enables higher level simulation and adaptiveness. To quantify the impact of timing error on the quality of service at the application-level, we define four classes based on the magnitude of TER shown in Table III. A higher TER implies higher number of violated critical paths, thus lower application-level quality of service. If a TER is classified as C_0, it means that all paths of FU meet the timing requirement; on the contrary, more than 66% of the paths (and up to 100%) are failed if a TER is classified as C_H. Hence, this classification covers various application-specific requirements on computational accuracy: C_0 for error-intolerant applications (e.g., general purpose applications), and C_L, C_M, C_H for error-tolerant applications (e.g., probabilistic applications [28]) where the acceptance threshold of TER is specified according to the target quality of service of applications.

TABLE III. CLASSES OF TER.

TER=0%	33%>= TER >0%	66%>= TER >33%	100%>= TER >66%
$Class_0$ (C_0)	$Class_{Low}$ (C_L)	$Class_{Medium}$ (C_M)	$Class_{High}$ (C_H)

We define X as a matrix of numeric predictor values [t_{clk} V T P A]. Each column of X represents one variable, and each row represents one observation. Y is defined as a numeric vector, and each row of Y represents the classification of the corresponding row of X. A linear parametric classifier, called discriminant analysis [29], is used to create a discriminant classification based on the input variables (predictors) X and output (response) Y. Thus, the model enables mapping of the five input variables to one of the four defined classes. The discriminant analysis assumes X has a Gaussian mixture distribution. To train the classifier, the fitting function estimates the parameters of a multivariate Gaussian normal distribution for each class. After training, the classifier produces the following:

- $\mathbf{M_\mu}$ is a matrix of class means of size K-by-P, where K is the number of classes, and P is the number of predictors. Each row of M_μ represents the mean of the multivariate normal distribution of the corresponding class.
- $\mathbf{M_\sigma}$ is a P-by-P matrix, the between-class covariance, where P is the number of predictors.
- $\mathbf{M_p}$ represents the prior probabilities for each class. M_p is a numeric positive vector of size 1-by-K representing the frequency with which each element occurs.

For each FU, the matrix of numeric predictor values, X, has a size of 149,500 (25×23×13×4×5)-by-5, as each row represents one permutation of the parameters summarized in Table II. Every row of Y depicts the characterized class of the corresponding row of X, determined by the TER Analysis module. The space of X values divides into regions where a classification Y is a particular value. The regions are separated by straight lines for the linear discriminant analysis. Feeding X and Y to the classifier M_μ, M_σ, and M_p are generated. The matrices for the floating-point adder (FP_add) are shown below:

$$M_\mu = \begin{pmatrix} 1.15E{+}00 & 9.97E{-}01 & 5.85E{+}01 & 4.67E{+}00 & 3.48E{+}01 \\ 8.38E{-}01 & 9.84E{-}01 & 6.49E{+}01 & 5.04E{+}00 & 4.09E{+}01 \\ 8.36E{-}01 & 9.71E{-}01 & 6.15E{+}01 & 4.85E{+}00 & 3.89E{+}01 \\ 4.65E{-}01 & 9.83E{-}01 & 6.13E{+}01 & 4.92E{+}00 & 4.00E{+}01 \end{pmatrix}$$

$$M_\sigma = \begin{pmatrix} 4.31E{-}02 & -2.37E{-}03 & 4.83E{-}01 & 4.37E{-}02 & 8.81E{-}01 \\ -2.37E{-}03 & 4.35E{-}03 & 1.03E{-}02 & 9.07E{-}04 & 1.83E{-}02 \\ 4.83E{-}01 & 1.03E{-}02 & 1.60E{+}03 & -1.91E{-}01 & -3.80E{-}00 \\ 4.37E{-}02 & 9.07E{-}04 & -1.91E{-}01 & 1.28E{+}01 & -3.37E{-}01 \\ 8.81E{-}01 & 1.83E{-}02 & -3.80E{+}00 & -3.37E{-}01 & 7.75E{+}02 \end{pmatrix}$$

$$M_p = [4.80E{-}01 \quad 8.10E{-}03 \quad 5.27E{-}03 \quad 5.07E{-}01]$$

Providing these parametric matrices, a prediction method discussed in the next section can accurately classify a given set of variations and a t_{clk} value to the corresponding class of timing error rate. The parametric models for the rest of FUs are detailed in [18] due to the lack space; the prefix 'FP_' stands for floating-point FUs and 'INT_' stands for integer FUs.

C. TER Classification

A classification algorithm seeks to minimize the expected classification cost:

$$\hat{y} = \arg\min_{y=1,\ldots,K} \sum_{k=1}^{K} P'(k \mid x)\, C(y \mid k) \qquad (2)$$

$\hat{y}$ is the predicted classification; K is the number of classes; $\hat{P}(k|x)$ is the posterior probability of class k for observation x; $C(y|k)$ is the cost of classifying an observation as y when its

true class is k. By default, $C(y|k)=1$ if y~=k, and $C(y|k)=0$ if y=k: the cost is 0 for correct classification, else it is 1.

The posterior probability that a point x belongs to class k is the product of the prior probability and the multivariate normal density. The density function of the multivariate normal with mean μ_k (k-th row of M_μ) and covariance M_σ at a point x is

$$P(x \mid k) = \frac{1}{(2\pi |M_\sigma|)^{0.5}} \exp\left(-\frac{1}{2}(x-\mu_k)^T M_\sigma^{-1}(x-\mu_k)\right) \quad (3)$$

where $|M_\sigma|$ is the determinant of M_σ, and M_σ^{-1} is the inverse matrix. Let $P(k)$ represent the prior probability of class k (k-th element of $\mathbf{M_p}$ vector). Then the posterior probability that an observation x is of class k is

$$P'(k \mid x) = \frac{P(k \mid x)P(k)}{P(x)} \quad (4)$$

where $P(x)$ is a normalization constant, the sum over k of $P(x|k)P(k)$. Therefore, we can quantify the expected misclassification cost per observation. Suppose we have an observation, $x=[t_{clk}'\ V'\ T'\ P'\ A']$, to classify with the trained discriminant analysis classifier. The expected (average) cost of classifying the observation into class k of K classes is

$$\text{cost}(k) = \sum_{i=1}^{K} P'(i \mid x) C(k \mid i) \quad (5)$$

$\acute{P}(i|x)$ is the posterior probability defined in Equation (4); and $C(k|i)$ is the cost of classification as described in Equation (2). Therefore, x belongs to the class k that has the lowest cost (k).

D. Robustness of Classification

To ensure the robustness of our method, we calculate resubstitution error as the difference between the response training data and the predictions the classifier makes of the response based on the input training data. If the resubstitution error is high, we cannot expect the predictions of the classifier to be good. The resubstitution error is 0.02 (the fraction of the training data X that classifier misclassifies) for the FP_add. On average, for all FUs the resubstitution error is 0.036 which is very low, meaning the models classify nearly all data correctly.

The sampling data for prediction is almost always a subset of the training data set, since the resolution of the training data, depicted in Table II, is much finer than the resolution of sampling sensors. In case of any out-of-sample data, for instance a temperature sensor with resolution of 1°C, the data can be conservatively matched to a surrounding point. However, we have obtained a full range of extra characterization points for temperature which are not used for training the model, and use these points to check if the model makes reasonable estimates for out-of-sample data. For extra characterization points with temperature range of 1°C–120°C (steps of 1°C) and with two distinct operating voltages (1.0V, 1.1V), the model makes correct estimates for 97% of out-of-sample data. The remaining 3% is misclassified to the high-error rate class (thus will have safe guardband). Note that we cannot go beyond the min/max range of the characterized points in the provided libraries [25].

IV. Runtime Hierarchically Focused Guardbanding

We now describe how this model for TER can guide a control system for runtime variation-aware resource management. At design time, to ensure numerical correctness for the computed result, we need to take the worst-case variations that could display for any combination of values of PVTA parameters. Thus, TER can be conservatively computed with significant uncertainty over the big cloud of possible post-silicon results. With the support of variability measurements at post-silicon fabrication, the PVTA parameters can be continuously monitored during the lifetime of the device, and consequently eliminate the conservativeness. For instance, the table in Figure 2 shows that during design time the delay of the FP_add has a large uncertainty of [0.73ns,1.32ns], since the actual values of PVTA parameters are unknown. But, immediately after fabrication this delay uncertainty is reduced to [0.73ns,1.25ns] if a process sensor reports that the adder is fabricated in a part of die with negligible WID variations. Even more, if the adder is monitored by an aging sensor, the delay uncertainty is further reduced to [0.73ns,1.07ns] when the device is fresh (ΔV_{th}=0mV). Having set the t_{clk}=0.8ns, each curve in Figure 2 shows how TER can change when voltage and temperature are varying at minimum/maximum process and aging conditions.

Thus, *Hierarchically Focused Guardbanding (HFG)* adaptively eliminates the conservative guardband due to PVTA variations during lifetime of device. It finely focuses on a FU and reduces its timing guardband depending upon the availability of distinct observers, in a hierarchical manner, started immediately after post-silicon fabrication (to compensate P), to during runtime execution (to compensate VT), and finally the entire of lifetime (to compensate A). This model-based use of PVTA readings provides a systematic way to reduce guardbands.

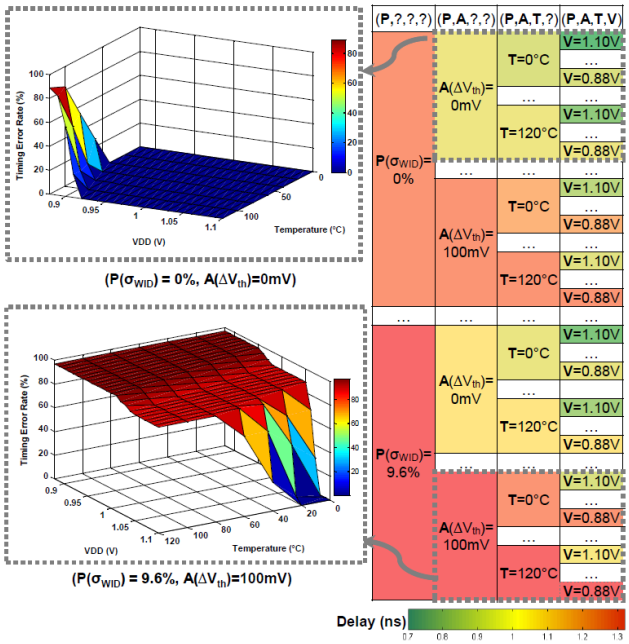

Figure 2. Delay variation and TER across extreme corners of PVTA.

A. Observability

The sensor instrumentation is required as delay variation changes across extreme corners of PVTA parameters. The question is that mix of monitors that would be useful? External non-intrusive monitors reside on the same die can measure distinct parameters like voltage droop [13], and temperature fluctuation [14]. In a similar vein, CPM [11] and TRC [10] monitors whole PVT variations. On the other hand, internal *in situ* monitors like EDS [6], Razor [5], and NBTI sensors [9] can measure the actual delay variation of device due to PVT

and aging. Figure 3 shows the minimum affordable t_{clk} (i.e., $1/Frequency_{Max}$) in presence/absence of various sensors for three FUs with a TER target of 0%. The sensors are sorted based on the time constant of the measured parameter, PATV: from DC component to high-frequency components. For instance, t_{clk} of FP_add can be reduced from 1.32ns to 1.26ns (a 0.06ns guardband reduction) depends to the actual value of WID process variation reported by a process monitor (P_sensor). It can be further reduced to 1.08ns if FP_add is equipped with the aging as well as the process sensor (PA_sensors). Adding thermal sensor enables even 0.06ns more reduction to 1.02ns (PAT_sensors). Finally, considering the full set of sensors enables decreasing t_{clk} from 1.32ns to 0.74ns (a great guardband reduction of 0.58ns) based on the measured values of variations reported by PATV_sensors. The more sensors we provide for a FU, the better conservative guardband reduction for that FU: the guardband can be reduced up to 8%, 24%, 28%, 44%, if we equip FP_add only with P_sensor, PA_sensors, PAT_sensors, and PATV_sensors, respectively.

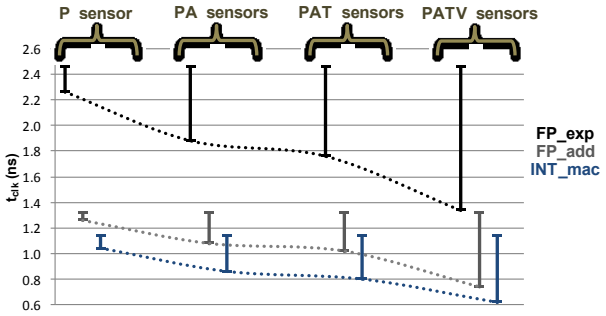

Figure 3. Hierarchical sensors for reducing guardband on t_{clk}.

As shown, this benefit is consistent across different FUs – with a shift in the worst-case guardband – even with better reduction for FP FUs (e.g., up to 47% for FP_exp with PATV_sensor case) due to the higher complexity of the circuit topology. Internal PVT sensors impose 1–3% area overhead [6], whereas five replica PVT sensors increase area of each POWER7 core by 0.2% [11],[12]. The banks of 96 NBTI aging sensors occupy less than 0.01% of the core's area [9].

B. Controllability

Employing any combination of PATV sensors provides online measurement of the actual parameters variations, and thus a control system can adaptively apply an appropriate guardbanding utilizing the characterized models for FUs. Among available control knobs, adaptive clock scaling using Phase-Locked Loop (PLL) is widely utilized in resilient implementations [7][12],[14]. Therefore, the control system tunes the clock frequency through an online model-based rule. To support fast controller's computation, the parametric model (as the outcome of the analysis flow in Figure 1) generates distinct LookUp Tables (LUTs) for every FUs. LUTs are generated during design time for specific configuration of sensors, their resolution, and the desire target TER for FUs (target_TER). Figure 4 shows a full configuration of PATV_sensors with resolutions of (3.2%, 25mV, 20°C, 0.04V) that support the range of variations summarized in Table I. Therefore, in total 980 (4×5×7×7) rows are required within a LUT. The parametric model fills every row of a LUT for FU_i with the minimum t_{clk} such that TER (FU_i, t_{clk}, V_{row}, T_{row}, P_{row}, A_{row}) < target_TER. Every LUT is stored in a dedicated 1KB SRAM to enable fast return of the 5-bit t_{clk} for the corresponding values of PATV_sensors. The clock control changes the frequency based on the returned t_{clk}, thus reduces the guardbanding. Note that, since TER characterization in Equation (1) considers the static critical paths (which might not be activated during execution of certain dynamic inputs), the model always returns *an upper bound* of the actual TER, thus returned t_{clk} of LUTs guarantees the target_TER.

The next question to address is what type of monitoring observation granularity and what type of reacting time we need, e.g., cycle-by-cycle or tens of cycles or hundred of cycles? To analyze the effect of this choice of granularity, we apply HFG to GPU architecture at two granularities:

I. Fine-grained granularity of instruction-by-instruction monitoring and adaptation that signals of PATV sensors come from individual FUs that reside in the execution stage of GPU.

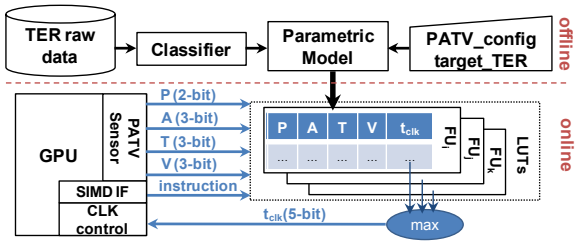

Figure 4. Online utilization of models through HFG.

The LUTs return the minimum t_{clk} depending on the actual value of PATV sensors and the chain of FUs that will be activated by the fetched instruction. To support single-cycle adaptation, a fast adaptive clocking circuit [14] consisting of three PLLs is use. Each PLL is running at independent frequencies, and a multiplexer quickly switches between them in a single-cycle. Therefore, the clock controller selects the highest t_{clk} (safe across all activated FUs) and reduces guardband that is compatible with PATV parameters and the demands of instructions, as shown in the following algorithm:

$$\forall \text{fetched instruction}_k$$
$$N = \#\text{of activated FUs by instruction}_k$$
$$\text{for } i=1 \text{ to } N$$
$$t_{clk-i}=\text{LUTs}(FU_i, V, T, P, A)$$
$$\text{set\_clock max}\{t_{clk-1}, t_{clk-1}, \ldots, t_{clk-N}\}$$

II. Coarse-grained granularity of kernel-level monitoring uses a representative PATV sensors for the entire execution stage of GPU pipeline. The clock adaptation is applied periodically before kernel execution. The controller selects t_{clk} based on current value of PATV sensors of the execution units and the chain of FUs that potentially will be activated during kernel execution (in a static sense). Since the adaptation of clock during kernel execution is prohibited, the controller considers a 5% extra margin on the reported voltage and temperature values to recover intra-kernel dynamic variations.

V. A CASE STUDY OF HFG ON GPU

We examine the effectiveness HFG on GPU architecture with the fine-grained instruction-by-instruction as well as the coarse-grained kernel-level monitoring and adaptation. We demonstrate our approach in an Evergreen-like GPU pipeline where our FUs reside in the execution stages of a Processing

Element (PE) and benefit from the adaptive clock scaling decided by the controller of HFG. The rest of pipeline stages are assumed to support resilient circuit techniques, as both resilient processor [7] and relaxed-reliability cores [28] consider sufficient guardband in the register stage, the memory management unit, L1 instruction cache, and the interconnect. We note that the instruction fetch and decode stages are not strongly vulnerable to variations [15], thus low-cost to protect.

For GPU kernel benchmarks, we use AMD APP SDK 2.5 [30] kernels suitable for stream applications written in OpenCL. Their device-specific assembly code is extracted by AMD APP KernelAnalyzer tool for applying the instruction-by-instruction and kernel-level HFG. Figure 5 (right) shows the maximum throughput (GIPS for a PE) of each kernel, when applying the coarse-grained kernel-level monitoring and adaptation with support of the four scenarios of PATV sensors. The results highlight two points: (a) more sensors in a PE result in a greater reduction in the guardband, and thus higher throughput for all kernels. On average, the throughput increases from 1.04 GIPS to 1.77 GIPS (70%), when the PE moves from only P_sensor to PATV_sensors scenario; (b) the throughput of kernel-level adaptation is limited by the slowest FU activated during the execution of the kernel. For instance, the throughput of MatrixMult, DCT, and EigenValue kernels is limited to 1.2 GIPS (with PATV_sensors), since those kernels activate FP_mac as the slowest FU.

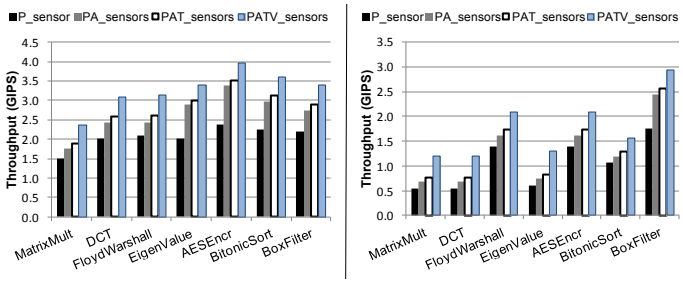

Figure 5. Maximum throughput benefit of HFG: (i) at instruction-level monitoring, the left figure; (i) at kernel-level monitoring, the right figure;

Figure 5 (left) shows the maximum throughput improvement in the instruction-by-instruction method. This method not only benefits from more sensors (60% in average), but also exploits the within-kernel opportunities for further reduction of inter-FU guardband. For example in PA_sensor case, the throughput of AESEncr kernel is increased up to 3.4 GIPS (93% higher than MatrixMult), thanks to all its integer instructions that only activate fast INT FUs. In comparison with the kernel-level method, the instruction-by-instruction monitoring and adaptation improves the throughput by a factor of 1.8×−2.1× depends to the PATV sensors configuration and kernel's instructions. Of course, this fine-grained instrumentation and adaptation has a higher cost in the area.

VI. CONCLUSION

This paper presents a model and its usage for runtime variation-aware resource management as well as design time analysis of vulnerable functional units. The model takes into account process parameters, temperature and voltage operating conditions, aging, and the physical details of P&R functional units using an accurate 45nm TSMC design and analysis flow. The model is used in a guardbanding scheme as an adaptive resource management technique to proactively prevent timing error by applying a *focused* guardbanding. HFG enhances the throughput of GPU kernels by 70% employing coarse-grained PVTA monitors and by applying adaptive guardbands at kernel-level. The finer granularity of instruction-by-instruction monitoring and adaptation achieves 1.8×−2.1× throughput improvements depends to the PVTA monitors configuration and the type of instructions executed within the kernels.

VII. ACKNOWLEDGMENTS

This research was supported by NSF Variability Expeditions under award n. 1029783, ERC-AdG MULTITHERMAN GA n. 291125, and Virtical GA n. 288574.

REFERENCES

[1] A. Borkar, et al., "Parameter variations and impact on circuits and microarchitecture," Proc. *DAC*, pp. 338- 342, 2003.
[2] ITRS [Online]. Available: http://public.itrs.net
[3] K.A. Bowman, et al., "Circuit techniques for dynamic variation tolerance," Proc. *DAC*, 2009.
[4] S. Nassif, "Delay variability: sources, impacts and trends," Proc. *ISSCC*, 2000.
[5] D. Ernst, et al., "Razor: circuit-level correction of timing errors for low-power operation," IEEE *MICRO*, pp. 10–20, 2004.
[6] K.A. Bowman, et al., "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," IEEE J. of Solid-State Circuits, 2009.
[7] K.A. Bowman, et al., "A 45 nm resilient microprocessor core for dynamic variation tolerance," IEEE J. of Solid-State Circuits, Jan. 2011.
[8] R. Pawlowski, et al., "A 530mV 10-lane SIMD processor with variation resiliency in 45nm SOI," Proc. *ISSCC*, pp. 492-494, 2012.
[9] P. Singh, E. Karl, D. Blaauw, D. Sylvester, "Compact degradation sensors for monitoring nbti and oxide degradation," IEEE Trans. on VLSI Systems, 2011.
[10] J. Tschanz, et al., "Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and Aging Variation Tolerance," Proc. *Symp. on VLSI Circuits*, pp. 112-113, 2009.
[11] A. Drake, et al., "A distributed critical-path timing moni-tor for a 65nm high-performance microprocessor," Proc. *ISSCC*, pp. 398-399, 2007.
[12] C. R. Lefurgy, et al., "Active management of timing guardband to save energy in POWER7," Proc. *MICRO*, pp. 1-11, 2011.
[13] S. Pant, D. Blaauw, "Circuit techniques for suppression and measurement of on-chip inductive supply noise," Proc. *ESSCIRC*, pp.134-137, 2008.
[14] J. Tschanz, et al., "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," Proc. *ISSCC*, 2007.
[15] A. Rahimi, L. Benini, R. K. Gupta, "Analysis of instruction-level vul-nerability to dynamic voltage and temperature variations," Proc. *DATE*, pp. 1102-1105, 2012.
[16] A. Rahimi, L. Benini, R. K. Gupta, "Procedure hopping: a low overhead solution to mitigate variability in shared-L1 processor clusters," Proc. *ISLPED*, 2012.
[17] A. Rahimi, A. Marongiu, P. Burgio, R. K. Gupta, L. Benini, "Variation-tolerant openMP tasking on tightly-coupled processor clusters," Peoc. *DATE*, 2013.
[18] PVTA Models for Hierarchically Focused Guardbanding [Online]. Available: http://mesl.ucsd.edu/site/PVTA_MODELS/models.htm
[19] S.R. Sarangi, et al., "VARIUS: A model of process variation and resulting timing errors for microarchitects," IEEE Tran. on Semiconductor Manufacturing, Vol.21, pp.3-13, Feb. 2008.
[20] U. R. Karpuzcu, et al., "VARIUS-NTV: A microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages," Proc. *Dependable Systems and Networks (DSN)*, pp.1-11, 2012.
[21] E. Mintarno, et al., "Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging," IEEE Tran. on CAD (TCAD), May 2011.
[22] J. Zhao, B. Datta, W. Burleson, R. Tessier, "Thermal-aware voltage droop compensation for multi-core architectures," Proc. *GLSVLSI*, pp. 335-340, 2010.
[23] M.S. Gupta, V.J. Reddi, G. Holloway, W. Gu-Yeon, D.M. Brooks, "An event-guided approach to reducing voltage noise in processors," Proc. *DATE*, 2009.
[24] E. Krimer, P. Chiang, M. Erez, "Lane decoupling for improving the timing-error resiliency of wide-SIMD architectures," Proc. *ISCA*, pp. 237-248, 2012.
[25] PrimeTime® VX User Guide, June 2011.
[26] TSMC 45nm standard cell library release note, TCBN45GS, Nov. 2009.
[27] S . Herbert, and D. Marculescu, "Characterizing chip-multiprocessor variability-tolerance," Proc. *DAC*, pp.313-318, 2008.
[28] L. Leem, H. Cho; J. Bau, Q.A. Jacobson, S. Mitra, "ERSA: error resilient system architecture for probabilistic applications," Proc. *DATE*, pp.1560-1565, 2010.
[29] MATLAB [Online]. Available: http://www.mathworks.com/
[30] AMD APP SDK 2.5 [Online]. Available: http://www.amd.com/stream