Design and Test Strategies for Microarchitectural Post-Fabrication Tuning

Xiaoyao Liang
Suzhou University
xliang@suda.edu.cn

Benjamin C. Lee
Stanford University
bclee@post.harvard.edu

Gu-Yeon Wei, David Brooks
Harvard University
guyeon,dbrooks@eecs.harvard.edu

***Abstract*— Process variations are a major hurdle for continued technology scaling. Both systematic and random variations will affect the critical delay of fabricated chips, causing a wide frequency and power distribution. Tuning techniques adapt the microarchitecture to mitigate the impact of variations at post-fabrication testing time. This paper proposes a new post-fabrication testing framework that accounts for testing costs. This framework uses on-chip canary circuits to capture systematic variation while using statistical analysis to estimate random variation. We derive regression models to predict chip performance and power. These techniques comprise an integrated framework that identifies the most energy efficient post-fabrication tuning configuration for each chip.**

I. Introduction

Moore's Law has driven fundamental advances in computing, enabling regular and predictable transistor scaling. Such scaling improves computing capabilities, increases memory capacities, and reduces price-capability ratios. However, process variations, a manufacturing effect of Moore's Law scaling in nanoscale CMOS technologies, jeopardize the significant performance and power advances from scaling. Due to variations in modern, aggressively-scaled technologies of 45nm and beyond, the expected 30 percent performance improvement per process generation is no longer certain [1]. Designers strive to balance delays across paths through combinatorial logic blocks such that no single path defines the critical path delay. With process variations, delay balancing is particularly challenging since one marginal set of transistors might compromise system performance.

Although statistical timing analysis attempts to model process variations at design time, the realized impact of process variations is unknown until the chip returns from fabrication. Thus, strategies to mitigate process variations often include a post-fabrication component. To mitigate variations' impact on critical path delay, post-fabrication tuning configures a chip's voltage, latency, and structure sizes. Such analysis and tuning at the microarchitectural level is necessary to fully account for variations' impact on system performance (instructions per cycle) and power.

These variation mitigation techniques expose a large space of tunable parameters, presenting significant challenges for frameworks that extract performance and power efficiency from a chip using these parameters. A tuning technique that exhaustively searches the space to identify optimal configurations is intractable since the search space size grows combinatorially with the number of tuning techniques, the tunable parameters within each technique, and the number of tuned microarchitectural structures.

Addressing these search and optimization challenges in post-fabrication tuning, we propose an enhanced testing framework. This framework implements the standard test flow, which includes wafer-level test/repair, packaging, and chip-level test/bin. We enhance the standard flow to characterize process variations, to estimate the impact of those variations, and to optimize tunable parameters to best mitigate that impact. Such a framework addresses two challenges in mitigating process variations. First, post-fabrication tuning is required since the realized impact of variations are unknown pre-fabrication. Secondly, intelligent modeling and optimization is required to tractably explore the space of tunable parameters. Collectively, the proposed testing framework effectively mitigates process variations, delivering high performance and power efficiency at low testing cost.

II. Motivation and Background

The techniques described in this paper are at the intersection of microarchitectural design and test. As such, this section provides the necessary background in these subareas.

A. Process Variations

While process variation exists at several scales, within-die (WID) variation is particularly important for nanoscale technologies [1]. Variations impact device feature sizes and threshold voltages, which cause delay variations. To meet speed targets, a typical design must accommodate the worst-case portion of the chip by either reducing frequency or increasing voltage. In contrast to die-to-die (D2D) variation, which might be mitigated with coarse grained strategies (e.g., frequency binning), WID variation requires more localized, die-level microarchitectural solutions.

Process variation includes both systematic variations due to lithographic irregularities and random variations due to varying dopant concentrations. In this work, systematic variations are defined as those that exhibit strong spatial correlation among device features for structures located close together. Canary circuits placed near a circuit of interest exploit these correlations yielding insights into a chip's systematic variations. In contrast, adjacent circuits are uncorrelated for random variations. While it is difficult to directly measure the random variation, statistical timing analysis provides a framework for reasoning about their impact. We combine

canary circuits and statistical timing analysis to enable new capabilities in post-fabrication tuning.

This paper uses delay and power data are derived from Hspice circuit simulations at the 32nm technology node using Predictive Technology Models (PTM) [2]. We rely on a Monte-Carlo simulation framework, which is similar to prior approaches [3], [4]. We model both random and systematic fluctuations at the transistor level. We assume $\sigma L/L_{nominal} = 7\%$ for gate-length variations and $\sigma V_{th}/V_{th_{nominal}} = 15\%$ for threshold voltage variations, which are comparable to data forecasts in prior work [1].

B. Post-Fabrication Tuning Techniques

Post-fabrication tuning techniques seek to optimize delay by modifying the supply voltage or the structures of the microarchitecture units. To meet delay targets, we focus on three post-fabrication tuning techniques: variable latency (VL), voltage interpolation (VI), and resizing:

- **Variable Latency (VL):** VL provides two possible latency settings for each architecture unit [5]. If the delay of one unit exceeds the target delay, the latency of that unit can be extended without reducing the global frequency. Extending the latency of architectural blocks often lead to an IPC loss and a detailed study of post-fabrication tuning configurations is required to guarantee a net performance gain when using this technique.
- **Voltage Interpolation (VI):** VI can provide fine-grained voltage tuning for each architecture unit [5]. It can provide effective voltage levels by interpolating two global supply voltages (VddH and VddL). Depending on the number of cuttings of the logic blocks, we can have a different number of "effective" voltage levels, which we refer to the as VI points. These VI points are obtained by assigning each pipeline stage to a low or high voltage. A VI point is a particular combination of low and high voltages across these stages.
- **Resizing:** Resizing adjusts array sizes of key microarchitectural structures (e.g., cache), which significantly impact system performance [6]. If the delay cannot fit into the target frequency, we can reduce the array size by turning off part of the array that operates at a slow speed. This technique should be applied with caution since the size of key architecture queues is very important to the system performance. This technique trades off IPC with target frequency.

A common theme across the space of post-fabrication tuning schemes is an exacerbation of testing challenges. All techniques, in effect, require a per-chip customization of various resources at microarchitectural block granularity.

III. Standard Test Flow

We define the test flow as the sequence of steps from wafer fabrication to product shipment. A variety of tests are conducted at both wafer and packaged-chip levels, stuck-at fault checks, IDDQ measurements, at-speed functional tests, AC scan, etc. [7]. Test time directly translates into cost and, consequently, any post-fabrication tuning technique must minimally impact the test time.

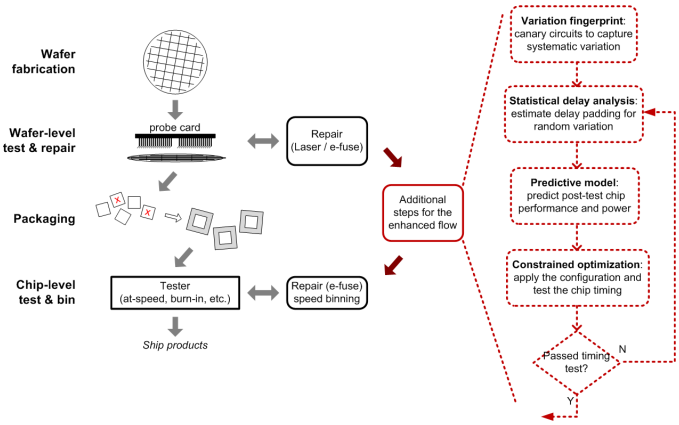

Fig. 1. Illustration of standard microprocessor test flow with additional steps for proposed post-fabrication tuning.

Figure 1 illustrates a standard test flow for microprocessors. Preliminary tests at the wafer level pre-screen defective parts early in the flow. After wafer-level test and sort, wafers are sent to the assembly house to be diced up and only parts that pass wafer-level tests are packaged. We assume a relatively long latency between wafer-level test and the next test phase, especially if the fab and assembly houses are not co-located. Packaged parts are put through another round of rigorous tests and binning before they are shipped. Additional steps in the flow added for post-fab tuning, identified on the right, are described in Section IV.

IV. Enhanced Test Flow

Tuning techniques require low-cost test solutions that efficiently set tuning knobs without incurring large overheads. We propose a generally-applicable testing framework for post-fabrication tuning with minimal impact on testing latency and time. In Figure 1, the right-most sequence of steps illustrate the additions made to the standard test flow. The proposed scheme relies on scan-enabled on-chip process monitoring circuits, also known as canary circuits, scattered throughout the chip to provide a "fingerprint" of on-chip process variations (Section IV-A). Based on the variation fingerprint, we statistically estimate the true critical path delay (Section IV-B). This estimated delay is combined with regression models to predict and to optimize performance and power as a function of tunable parameters (Section IV-C). We then apply parameters identified optimal by regression to chips and test to verify their timing behavior (Section IV-D). If a chip passes the delay test, regression-predicted optima for tuning knobs are applied to the packaged part for shipment.

A. Variation Fingerprint

Canary circuits measurements characterize a chip's systematic variations, providing a variation fingerprint. Canary circuits are on-chip process monitoring circuits, commonly

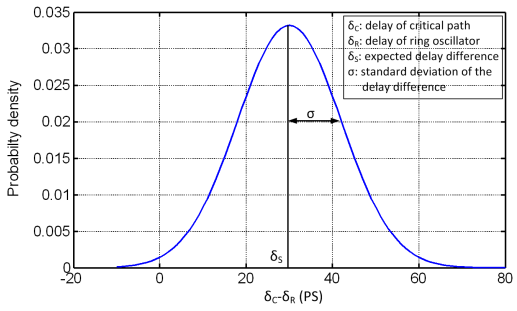

Fig. 2. The probability density function of difference between ring oscillator and critical path delays for the representative instruction decoder. $\delta_R - \delta_C \sim N(\delta_S \approx 30ps, \sigma \approx 12ps)$.

used to profile chip characteristics [8]. Since canary circuits are in physical proximity to the critical paths of an architectural block, the effects of systematic variations on canary circuits and the average path are highly correlated. Ideally, delay measurements from canary circuits will be an accurate proxy for path delay. Without loss of generality, we assume ring-oscillators are used as canary circuits in this paper both other canary circuits are equally applicable.

In practice, canary delays are imperfect proxies of path delays since canary circuits only capture systematic variations and cannot accurately model random variations. These random variations will lead to differences between the canary circuit and the average path. Thus, canary circuits are representative of the *average* path and not the *critical* path. To illustrate this difference between paths, we consider a hypothetical scenario with known delay for both ring oscillators and critical paths. To quantify delay differences, we implement a Monte Carlo experiment with 1,000 instances of the instruction decoder, a logic-dominated architectural block. Each Monte Carlo instance has a slightly different oscillator and critical path delay due to process variations.

Figure 2 illustrates the distribution of delay differences between the oscillator delay (δ_R) and the worst-case critical path delay (δ_C) in the decoder block from Monte Carlo simulations. The delay difference follows approximately a Normal distribution with mean δ_S and standard deviation σ. Intuitively, δ_S quantifies the expected difference between the measured ring oscillator delay and true critical path delay ($\delta_S = E[\delta_R - \delta_c]$). If we consider a large number of paths through a logic block, the expected difference between measured ring oscillator delay and true *average* path delay is zero. But for a fully synchronous digital system, the true *critical path* delay determines the operating frequency. Noting that ring oscillators are designed to capture only average delays, critical path delays are estimated from ring oscillators with an additional delay shift δ_S to account for this difference between average and critical path delays.

The standard deviation σ captures the spread in differences between measured oscillator delay and true worst-case critical path delay. This σ is primarily due to variance on the critical paths, which produces a delay distribution for fabricated chips. Parameters δ_S and σ constitute our variation fingerprint. In practice, we would fully pre-characterize δ_S and σ by measuring an initial lot of fabricated chips during wafer-level test and repair as described in Figure 1. In this work, we assume N=1,000 chips are pre-characterized early in the production cycle as the product ramps up. The overhead of measuring N chips is low compared with subsequent chip manufacturing volumes.

B. Statistical Delay Analysis

In contrast to statistical timing analysis at design time, our testing framework implements post-fabrication statistical delay analysis. A post-fabrication test defines a target for critical path delay and identifies all chips that satisfy that target. Because worst-case critical path delay is unknown, the testing methodology must rely on a combination of a measured ring oscillator delay (δ_R), the expected difference between measured oscillator and true critical path delays ($\delta_S = E[\delta_R - \delta_C]$), and extra delay padding (δ_P) that provides an error margin to the estimate of critical path delay. Padding δ_P is needed since tests are evaluated with measured oscillator delays that may not accurately capture true critical path delay. Thus, the estimated critical path delay for a microarchitectural block is given by $\hat{\delta}_C = \delta_R + \delta_S + \delta_P$ and the estimate is made more conservative by increasing δ_P. $\hat{\delta}_C$ is an estimate for the true delay δ_C.

Post-fabrication delay tests are evaluated with estimated worst-case critical path delay $\hat{\delta}_C$ but results may differ if tests were evaluated with true critical path delay δ_C. We define the *block pass rate* (PR_{block}) as the number of blocks that pass the same test under $\hat{\delta}_C$. Intuitively, the pass rate is a measure of confidence in the estimate $\hat{\delta}_C$. A high pass rate means $\hat{\delta}_C$ is a conservative estimate and is likely at least as large as δ_C. If $\hat{\delta}_C \geq \delta_C$, then any tuning configuration that satisfies the delay test for $\hat{\delta}_C$ (a more difficult constraint) will also satisfy the delay test for δ_C (a less difficult constraint) leading to a high pass rate. For example, logic that is pipelined assuming a critical path delay of $\hat{\delta}_C$ will still meet timing constraints if the actual critical path delay δ_C is shorter.

To further explore this relationship between *block pass rate* and average chip $BIPS^3/W$ efficiency, we consider a range of *chip pass rates* (PR_{chip}) and define delay tests such that the pass rate is achieved. We translate chip pass rate to block pass rate using elementary probability theory for B blocks: $(PR_{block})^{\alpha} = PR_{chip}$. If B microarchitectural block delays are perfectly correlated, the chip-level pass rate must be satisfied by every block and $\alpha = 1$. If the B microarchitectural block delays are completely independent, each of the B blocks must have pass rate of $(PR_{chip})^{1/B}$ and $\alpha = B$. In practice, blocks are neither fully correlated nor fully independent, implying $1 \leq \alpha \leq B$. In the absence of any variation, all blocks are perfectly correlated by design. Systematic variations reduce this correlation at coarse granularity and random variations reduce this correlation at fine granularity. The exact impact on correlation must be quantified by measurement or simulation. As described in

Arch unit	Latency choices	Array size choices	VI points
DEC	3,4-cycle	-	0.8:0.02:1.2
MAP	3,4-cycle	32, 64, 96, 128	0.8:0.02:1.2
RF	3,4-cycle	32, 64, 96, 128	0.8:0.02:1.2
IQ	3,4-cycle	10, 20, 30, 40	0.8:0.02:1.2
FXU	3,4-cycle	-	0.8:0.02:1.2
FPU	4,5-cycle	-	0.8:0.02:1.2

TABLE I

POST-FABRICATION TUNING AND CONFIGURATION KNOBS.

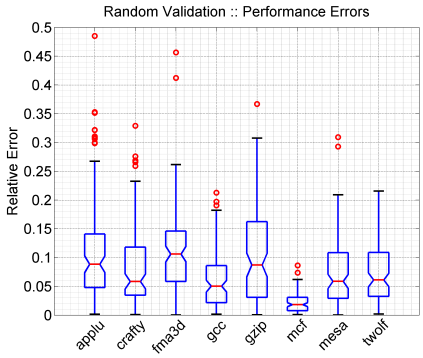

Fig. 3. Distribution of prediction errors for 100 random validation configurations.

Section IV-D, we empirically determine the proper value $\hat{\alpha}$, our best estimate of the true α.

$$PR_{block} = (PR_{chip})^{1/\hat{\alpha}} \quad (1)$$

$$\delta_P = Q(PR_{block}) \times \sigma \quad (2)$$

Given $\hat{\alpha}$, we compute the block pass rate from the desired chip pass rate as shown in Equation 1. The block pass rate, in turn, defines the block-level delay padding as show in Equation 2. Specifically, we compute a particular quantile of the block's oscillator delay distribution and multiply by its standard deviation. For example, if we wish to achieve a block pass rate $PR_{block} = 0.997$, then $Q(PR_{block}) = 3$, since 99.7% of the probability distribution for estimates of critical path delay is located below $\delta_S + 3\sigma$ (shown in the normal distribution of Figure 2). Both the quantile function Q and the standard deviation σ are known once we pre-characterize N=1,000 chips to get a variation fingerprint as described in Section IV-A.

C. Predictive Model

Provided with a characterization and delay analysis of chips' realized variations, the tuning framework must configure the parameters in our post-fabrication tuning techniques. We consider an optimization space with hundreds of billions of possible configurations, which exhibit significant diversity in performance and power. To tractably identify optimal configurations for each chip, we must construct predictive models to capture the relationship between performance, power, and tuning parameters. Computationally efficient predictive models enable comprehensive optimization across the large configuration space.

Techniques in statistical inference reveal performance and power trends from sparsely measured configuration samples, enabling for much larger, comprehensive tuning spaces. In particular, we apply spline-based regression models, which predict a performance or power response as a function of design parameter values [9]. Interactions between predictors are captured by products terms specified in the models' functional form using domain-specific knowledge. For example, cache sizes for adjacent levels in the memory hierarchy should interact (i.e., the optimal L1 cache size depends on the L2 cache size and vice versa, thereby requiring joint optimization). Non-linearity is captured by cubic spline (i.e., piecewise polynomial) transformations on the predictors. Given sparsely measured configurations from the space, a multi-dimensional curve fit is performed to capture relationships between design metrics of interest and tunable parameters. Model construction is computationally efficient and may be reduced to a series of cubic transformations followed by a linear solve (highly-optimized matrix operations). Model evaluation, expressed as matrix multiplication, is also highly efficient. Hundreds or thousands of predictions per second are possible. This computational efficiency allows tractable exploration for a large space of microarchitectural structural, voltage, and latency configurations.

We model a baseline processor comparable to the Alpha 21264. The tunable parameters are listed in Table I. With voltage interpolation, every unit can have 20 effective voltage settings. With variable latency, every unit can have two latency choices. Array structures take one of four sizes. We allow eight frequency choices for each chip. Table I yields a large design space of 282 billion post-fabrication configurations. Simulations indicate different configurations produce very different power and performance values. Performance ranges between 0.39 and 1.39 BIPS, a factor of 3.6×. Similarly, power ranges between 0.53 and 0.98, a factor of 1.86×.

We train regression models with 500 configurations sampled uniformly at random from the space of 282 billion points. Such a sparse sampling is used to construct unbiased models that weight every part of the configuration space equally. These training configurations are measured at the beginning of a production cycle, incurring the one time cost of constructing these models.

Figure 3 illustrates model accuracy when validated against simulation for 100 randomly selected and separate validation points, demonstrating median errors of 7.4 percent for performance. Our enhanced test flow relies on these efficient predictive models to capture the relationship between $BIPS^3/W$ efficiency and post-fab tuning parameters. Since test time and costs have a direct impact on profit margins, statistical inference and other modeling methodologies are imperative. The computational efficiency of regression models enables previously intractable modeling and optimization

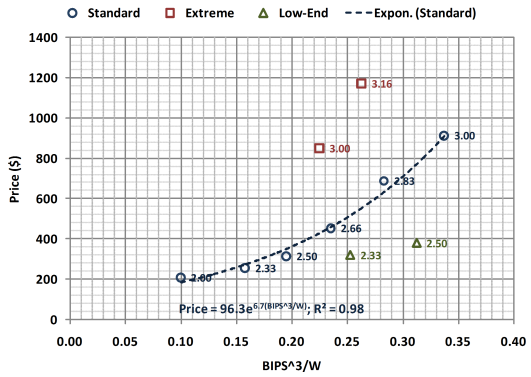

Fig. 4. $BIPS^3/W$ and price for representative Intel Xeon processors (12MB L2, 1333MHz FSB, 45nm) [11]. Data series annotated with products' frequencies in GHz.

for the space of structure sizes, latencies, and voltages. The results of regression model optimization determine the configuration applied in post-fabrication tuning.

D. Constrained Optimization

Constrained $BIPS^3/W$ optimization identifies optimal latencies, voltages, and structure sizes for the post-fabrication tuning techniques of Section II-B. Delay constraints are provided from analysis of Section IV-B. $BIPS^3/W$ estimates are provided by predictive models of Section IV-C.

Optimization Objective. Current microprocessors are binned with respect to maximum achievable clock speed or power efficiency as power often constrains performance in modern designs. Given the direct trade-off between frequency and power, we differentiate chips via the $BIPS^3/W$ metric [10]. Derived from the cubic relationship between power and voltage/frequency, this metric is voltage and frequency invariant. However, $BIPS^3/W$ is sensitive to parameters in the post-fabrication tuning techniques studied throughout this work. In the absence of detailed cost and pricing models, which are typically closely guarded industrial secrets, we use $BIPS^3/W$ as a proxy for price and evaluate the proposed framework with respect to trade-offs between $BIPS^3/W$, yield and test time. Published product data sets, although lacking detailed price and performance components, suggest a strong relationship between $BIPS^3/W$ and price.

For example, Figure 4 plots price against $BIPS^3/W$ as reported by data sheets for server class processors. The figure illustrates ten frequency bins for a particular server product and the price differentiation across these bins. This data captures only the frequency contribution to $BIPS$ and its associated power cost. Despite analyzing this subset of $BIPS$ contributors (other contributors include latency, structure sizes), we observe material relationships between $BIPS^3/W$ and price. An exponential relationship is fit very closely to (R^2=0.98) to six of ten *standard* products. The four outliers are specified as *extreme* or *low-end* processors targeting at special markets. We would expect similar relationships if latency and size contributions to $BIPS$ were included. Thus, we take $BIPS^3/W$ as our proxy for price without loss of generality. Price is likely a function of $BIPS^3/W$ and, although we illustrate an exponential function, the exact function is orthogonal and independent of the proposed methodology. Any other price function might be used in our testing framework.

Optimization Constraints. The framework maximizes $BIPS^3/W$ of the overall chip. The estimated critical path delay consists of three components. Ring oscillator δ_R is measured and known for each block. The delay shift δ_S is pre-characterized for a small number of chips and is included in the variation fingerprint. Lastly, δ_P is delay padding computed for a desired block pass rate from Equation 2. For each block, the estimated critical path delay $\hat{\delta}_C$ is constrained to be less than the delay of the tuning configuration (Equation 4). A block's delay is a function of its configuration, which includes configured voltage V_{cfg} relative to some nominal voltage V_{nom}, configured latency L_{cfg}, and configured frequency f_{cfg}. V_{cfg} affects the constraint as a higher configured voltage V_{cfg} reduces critical path delay and allows the constraint to be more easily satisfied.

$$\hat{\delta}_C = \delta_R + \delta_S + \delta_P \quad (3)$$

$$\hat{\delta_C} \times \frac{V_{nom}}{V_{cfg}} \leq \frac{L_{cfg}}{f_{cfg}} \quad (4)$$

The space of post-fabrication tuning configurations is defined by combinations of V_{cfg}, L_{cfg}, f_{cfg}. The optimization relies heavily on the computational efficiency of our predictive regression models. We exhaustively evaluate regression equations to predict the performance and power efficiency of every configuration. We repeat this optimization for every chip, obtaining chip-specific measurements for δ_R and identifying chip-specific optimal values for V_{cfg}, L_{cfg}, f_{cfg}.

Calibrated Optimization. Recall the analysis of Section IV-B assumes an empirically derived $\hat{\alpha}$. To empirically determine the measure of block-level correlation $\hat{\alpha}$, we repeat the above optimization for varying values of α, $1 \leq \alpha \leq B$. For each value of α, we characterize pass rates using the estimated critical path delay $\hat{\delta}_C$ and the true critical path delay δ_C. The empirically derived $\hat{\alpha}$ is chosen such that pass rates are equal for both analyses. This empirical calibration effectively identifies the degree to which blocks are correlated or independent. The calibration of $\hat{\alpha}$ is a one-time cost, requiring detailed measurements of true critical paths N sample chips early in the manufacturing process, where N is chosen to be very small relative to total volumes. Without loss of generality, we consider N=1,000 chips.

The chip pass rate directly influences the performance of chips passing the delay test. Figure 5 quantifies this relationship, plotting $BIPS^3/W$ efficiency against chip pass rate. A high pass rate leads to lower average efficiency since the high pass rate is achieved by more delay padding δ_P and conservative estimates of $\hat{\delta}_C$, which effectively increase the delay a chip can deliver and still pass. Searching the space of post-fabrication tuning configurations under such conservative delay estimates will produce configurations with

higher latencies, lower frequencies, and higher voltages. The net effect is lower efficiency. In contrast, if we consider a low pass rate, less delay padding δ_P is required and more $BIPS^3/W$ efficient configurations are identified. However, $\hat{\delta}_C$ is a less conservative estimate, which increases estimation error ($\hat{\delta}_C$-δ_C) and causes more chips fail delay tests than when evaluated under the true critical path delay δ_C.

V. Analysis and Evaluation

We evaluate the proposed testing framework by assessing trade-offs between the delivered $BIPS^3/W$ and measures of testing cost: number of tests and canary circuit density.

A. Tuning with Multiple Tests

There is an inherent trade-off between $BIPS^3/W$ and pass rate for a single test iteration. As shown in Figure 5, the pass rate directly influences the average $BIPS^3/W$ efficiency of chips passing the test. As the framework targets higher pass rates, average efficiency decreases. Further exploring the relationship between pass rate and average efficiency, we consider multiple tests and their ability to deliver greater efficiency. Multiple tests stratify chips by their $BIPS^3/W$ efficiency to improve average efficiency. Suppose, for example, we implement two tests. The first test is defined to achieve a low pass rate. Although a small fraction of chips pass this first test, each passing chip will achieve high $BIPS^3/W$ efficiency as shown in Figure 5. The second test, effectively a catch-all for chips that fail the first test, is defined to achieve a high pass rate. Due to this higher second pass rate, chips that fail the first but pass the second test will achieve on average lower $BIPS^3/W$ than those that pass the first test.

Given this intuitive understanding interactions between multiple tests, we consider a range of test counts and assess its impact on delivered efficiency. Figure 6 illustrates efficiency trends as the number of tests increases. We consider a continuum between *no-tuning* and *oracle-tuning*. No-tuning is the baseline method which does not implement post-fabrication tuning techniques for architecture flexibility, voltage interpolation, and variable latency. Thus, in the no-tuning case, the operating frequency is defined by the slowest critical path on the chip. In contrast, oracle-tuning assumes the true critical path is estimated perfectly with no error. Given this perfect estimate where $\hat{\delta}_C = \delta_C$, this ideal test flow fully utilizes architectural flexibility, voltage interpolation, and variable latency to maximize $BIPS^3/W$ efficiency while guaranteeing constraints for the true critical path delay are satisfied (Equation 4). As shown in Figure 6, no-tuning is 30 percent less efficient than oracle-tuning and significant efficiency is possible from post-fabrication tuning.

To determine the number of required tests, we vary test counts between one and five. We assume a total pass rate of 100 percent, which means all chips must eventually pass a delay test and the total pass rate across T tests must sum to 100 percent: $\sum_{t=1}^{T} PR_t = 100$. If only one test is used ($T = 1$), $PR_1 = 100$ and passing chips will achieve low average efficiency. If multiple tests are used ($T > 1$), we explore all

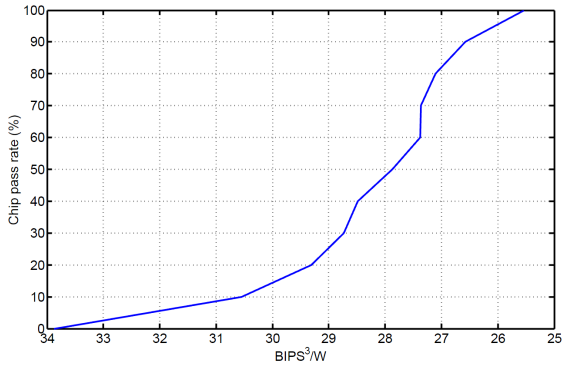

Fig. 5. Average $BIPS^3/W$ versus chip pass rate.

combinations of pass rates that satisfy $\sum_{t=1}^{T} PR_t = 100$ to identify the combination that maximizes average efficiency. For example, if $T = 3$, we examine all combinations of (PR_1,PR_2,PR_3) to maximize efficiency. Thus, our analysis considers the best achievable efficiency for each test count.

The diamond line (canary-per-block) of Figure 6 illustrates efficiency trends as the number of tests varies. With tuning techniques, even a modest number of tests drastically improves delivered efficiency. A single test improves efficiency by 1.27x, increasing normalized efficiency from 70 to 89 percent of oracle-tuning. Efficiency increases monotonically with the number of tests. However, we observe diminishing marginal returns in efficiency. Two post-fabrication tests are sufficient to achieve 93 percent of oracle-tuning whereas five post-fabrication tests achieve 96 percent. Overall, this analysis shows the effectiveness of using canary circuits to guide post-fabrication tuning.

B. Canary Circuit Density

The efficiency gains from our enhanced test flow are driven by canary circuits and their characterization of variation fingerprints. The effectiveness of this characterization depends on the density of canary circuits. Figure 6 illustrates efficiency trends under different canary circuit (e.g., ring oscillator) densities. We consider three scenarios in order of decreasing canary density: (1) canary-per-block, (2) canary-per-cluster, and (3) canary-per-chip. The canary-per-block scenario is the baseline scenario considered in the previous analysis, illustrating trends for the greatest canary density where each of the six architectural blocks contains a ring-oscillator. For comparison, we define a cluster of three architectural blocks and consider a canary-per-cluster scenario where the three blocks share a single ring oscillator. In this scenario, the test flow attempts to capture systematic variations for the three blocks using a single measurement. Similarly, we also consider a canary-per-chip where all six architectural blocks share a single-ring oscillator.

As shown in Figure 6, canary density significantly impacts achieved efficiency from our test flow. Post-fabrication tuning is less effective when fewer ring oscillators are available. In particular, one post-fabrication test is worse than no-tuning under canary-per-cluster or canary-per-chip scenarios.

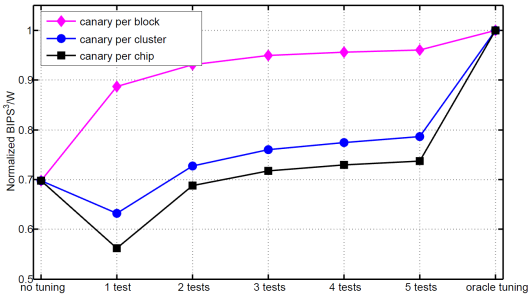

Fig. 6. Average $BIPS^3/W$ versus number of tests.

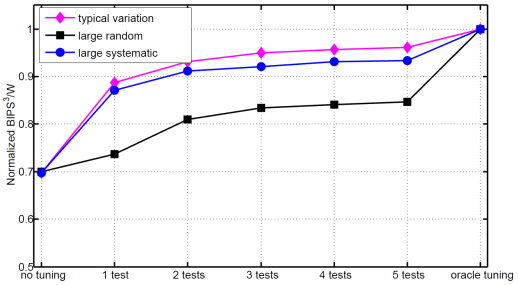

Fig. 7. Average $BIPS^3/W$ versus variations.

These scenarios deliver only 63 and 71 percent of canary-per-block efficiency. Canary circuits are designed to capture the impact of process variations for a small, localized on-chip area, but the canary-per-cluster and canary-per-chip scenarios attempt to generalize variation estimates from these localized measurements to much larger cluster or chip areas. These coarse-grained canary measurements provide a misleading variation fingerprint that leads to a failure of testing and optimization schemes. Although efficiency increases with more tests, canary-per-cluster and canary-per-chip scenarios are disadvantaged because they perform these additional tests with an incomplete characterization of systematic variation that also leads to inaccurate estimates of random variations.

C. Sensitivity to Process Variation

Figure 7 illustrates the impact of process variations on our enhanced test flow. We consider three scenarios: (1) typical variation, (2) large random, and (3) large systematic. Typical variation assumes gate length coefficient of variation $\sigma L/L_{nom} = 7\%$ and threshold voltage coefficient of variation $\sigma V_{th}/V_{th_{nom}} = 15\%$. Large random variation considers greater gate length variation with $\sigma L/L_{nom} = 14\%$ and large systematic variation considers greater threshold voltage variation with $\sigma V_{th}/V_{th_{nom}} = 30\%$. This analysis considers canary-per-block measurements and assesses the impact of greater variations on our test flow.

As shown in Figure 7, the enhanced test flow is more sensitive to random variations. Under a canary-per-block scenario, ring oscillators effectively capture the effects of large systematic variations and our test flow delivers efficiency comparable to that delivered under typical variation. We observe negligible efficiency differences between two and three percent when we consider greater systematic variation. In contrast, we observe significant efficiency losses under large random variation. Ring oscillators cannot capture increased random variation, which increases errors in estimates of critical path delay by increasing the spread σ in Figure 2. As the spread between canary-derived estimates and true critical path delays increases, our test flow requires greater delay padding δ_P to guarantee desired chip-level pass rates, which reduces average $BIPS^3/W$ and hinders tuning.

VI. Conclusion

Process variations has become an increasingly important issue for future microprocessor designs in nanoscale technologies. Various post-fabrication tuning techniques have been proposed recently to adapt the microarchitecture and circuit to different degrees of variations. This paper proposes the use of on-chip canary circuits to capture the correlated systematic variation, combined with statistical analysis and regression models to estimate the random variation and find the best post-fabrication settings for all chips. Experiments show the testing cost for the proposed approach is low and can fit well into existing approaches with minimal overheads.

References

[1] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *Journal of Solid-State Circuits*, vol. 37, no. 2, February 2002.

[2] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm design exploration," in *IEEE International Symposium on Quality Electronic Design*, 2006.

[3] X. Liang and D. Brooks, "Mitigating the impact of process variations on processor register files and execution units," in *39th IEEE International Symposium on Microarchitecture*, December 2006.

[4] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," in *International Conference on Computer-Aided Design*, November 2003.

[5] X. Liang, G. Wei, and D. Brooks, "ReVIVaL: Variation tolerant architecture using voltage interpolation and variable latency," in *International Symposium on Computer Architecture*, June 2008.

[6] A. Agarwal, B. C. Paul, H. Mahmoodi, A. Datta, and K. Roy, "A process-tolerant cache architecture for improved yield in nanoscale technologies," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 13, no. 1, January 2005.

[7] P. C. Maxwell, "Wafer-package test mix for optimal defect detection and test time savings," in *Design and Test of Computers, IEEE vol. 20 (5) pp. 84 - 89*, 2003.

[8] M. Hatzilambrou, A. Neureuther, and C. Spanos, "Ring oscillator sensitivity to spatial process variation," in *Proc. 1st Int'l Workshop Statistical Metrology (IWSM 96)*, 1996.

[9] B. Lee and D. Brooks, "Accurate and efficient regression modeling for microarchitectural performance and power prediction," in *ASPLOS: International Conference on Architectural Support for Programming Languages and Operating Systems*, October 2006.

[10] D. Brooks et al., "Power-aware microarchitecture: Design and modeling challenges for next-generation microprocessors," *IEEE Micro*, vol. 20, no. 6, pp. 26–44, Nov/Dec 2000.

[11] "Intel processor pricing. Effective July 20, 2008."