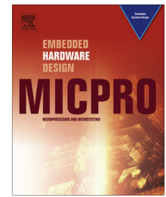
Microprocessors and Microsystems 38 (2014) 22–30

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro



Design configuration selection for hard-error reliable processors via statistical rules

Ying Zhang [a,*], Lide Duan [b], Bin Li [c], Lu Peng [a], Xin Fu [d]

[a] Division of Electrical & Computer Engineering, School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, LA 70803, United States
[b] Advanced Micro Devices, Inc., Austin, TX 78735, United States
[c] Department of Experimental Statistics, Louisiana State University, Baton Rouge, LA 70802, United States
[d] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, United States

ARTICLE INFO

Article history:
Available online 11 November 2013

Keywords:
Hard-error reliability
Modeling
Process variation

ABSTRACT

Lifetime reliability is becoming a first-order concern in processor manufacturing in addition to conventional design goals including performance, power consumption and thermal features since semiconductor technology enters the deep submicron era. This requires computer architects to carefully examine each design option and evaluate its reliability, in order to prolong the lifetime of the target processor. However, the complex wear-out mechanisms which cause processor failure and their interactions with varying microarchitectural configurations are still far from well understood, making the early optimization for chip reliability a challenging problem. To address this issue, we investigate the relationship between processor reliability and the design configuration by exploring a large processor design space in this paper. We employ a rule search strategy to generate a set of rules to identify the optimal configurations for reliability and its tradeoff with other design goals.

In addition to the wear-out effects, the ever-shrinking feature size of modern transistors makes process variation a significant issue in the chip fabrication. Process variation results in unexpected distributions of key design parameters, thus remarkably impacting important features of the target processor. Therefore, we also extend our investigation to identify the optimal configurations in the presence of process variation.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The unceasing downscaling of the semiconductor technology makes hard-error reliability a first-order concern in modern high-performance processor design. Due to the substantial transistors integrated on a chip, the processor power density and runtime temperature keep rising, which tend to largely impair the processor lifetime reliability. For instance, Negative Bias Temperature Instability (NBTI), caused by the continuous increase in processor threshold voltage, becomes a key reliability issue when the manufacturing technology reaches 90 nm. The situation is further exacerbated when the on-die temperature increases.

Although several techniques including power balancing and hotspot elimination have been proposed to improve the lifetime reliability for a given processor [8], designing hard-error resilient processors at the pre-silicon stage remains an open topic. Considering that processors with different microarchitectural configurations are prone to show distinct behavior such as various power dissipations, it is reasonable to infer that microarchitectural configurations selections will heavily impact the hard-error reliability of a processor [10,28]. However, traditional design space explorations mainly concentrate on performance, power, and thermal features of the target processor [17], leaving the relationship between hard-error reliability and the underlying design option far from obvious. This indicates that selecting the most reliable configurations at the early stage of processor manufacturing is of great importance for computer architects.

In this paper, we aim to address this issue by exploring a vast design space. We model four important and well-studied failure mechanisms including electromigration (EM), stress migration (SM), thermal cycling (TC), and the aforementioned NBTI. The processor failure rate (FIT, or failures in 10^9 h) is used to interpret the reliability, where a smaller FIT value indicates a longer MTTF (mean time to failure), i.e., higher reliability. To correlate configuration parameters to the design objective, we employ an advanced statistical technique called Patient Rule Induction Method (PRIM) to facilitate our work. The PRIM model can extract a few simple "rules" or conditions by which the processor satisfies a preset design goal for a certain response. Specifically in our work, the optimal design configurations for improving hard-error reliability and

* Corresponding author.
E-mail addresses: yzhan29@lsu.edu (Y. Zhang), lide.duan@amd.com (L. Duan), bli@lsu.edu (B. Li), lpeng@lsu.edu (L. Peng), xinfu@ittc.ku.edu (X. Fu).

0141-9331/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2013.11.002

its tradeoff with other design goals are generated by training PRIM models.

On the other hand, process variation (PV) is gradually becoming a significant issue during processor manufacturing in the deep submicron era. Process variation manifests itself in a variability of transistor process parameters (e.g., threshold voltage, or V_{th}) from their design specifications, which is caused by the difficulty in consistently controlling the transistor manufacturing in small dimensions. Process variation can impose significant impact on the performance and power consumption of the target processor. For example, the varying V_{th} among integrated transistors results in substantial delay variability in critical paths, thus greatly degrading the processor frequency and hence the overall performance as it is determined by the slowest critical path across the chip. In one word, process variation is becoming a major challenge in the design of future high-performance microprocessors.

Moreover, the interaction between varying microarchitectural configurations and process variation introduces more complexities to the designers. For instance, the design options can impact the number and distribution of critical paths, so processors with various architecture designs exhibit different capability in tolerating the PV effect. Therefore, in the early design stages, it is essential for designers to consider the PV effect when selecting the optimal design parameters. Taking this into consideration, we apply the PRIM based approach to circumstances where process variation is present, in order to seek the optimal configurations in this scenario.

In summary, the main contributions of this paper are as follows:

- *Optimal design configurations for hard-error reliability*: We demonstrate that different design choices will result in distinct reliability behaviors. Based on the investigation, we identify the most reliable microarchitectural configurations.
- *Optimal configurations for other metrics*: We show that different metrics including performance, power, temperature, and hard-error reliability prefer disparate design configurations. Therefore, we also investigate optimal configurations for other metrics.
- *Balancing reliability, power, and performance*: We generate rules to filter out configurations that yield optimal tradeoffs among multiple metrics.
- *Design stage optimization under process variation*: We take the widely-acknowledged process variation effect into consideration, and aim to identify the most promising microarchitectural configurations in the deep submicron era when PV is present.

2. Related work

The hard-error defects that degrade processor reliability [9,22] are first noticed in the circuit design process. Srinivasan et al. propose a reliability-aware microprocessor (RAMP) model [24] to assist overcoming the unnecessary high cost in manufacturing an overestimated processor. The authors further extend their study in [25] and demonstrate the impact of fabrication technology on hard-error reliability. At the architectural level, Bower et al. [5] introduce an online diagnosis technique to detect the hard faults. On the other hand, since the processor hard-error reliability is highly related to its runtime power and temperature, many techniques are proposed to improve the processor reliability by removing the on-die hotspot at runtime. Coskun et al. [8] discuss the impact of different job scheduling algorithms on processor lifetime. They demonstrate that the processor lifetime can be effectively prolonged when smart temperature-aware scheduling mechanisms are engaged. Hsu and Feng [14,15] concentrate on the heating and consequent reliability issues in high performance computing (HPC) systems and propose power management techniques by using dynamic voltage and frequency scaling, in order to reduce the failure rates.

The process variation phenomenon has gained much attention in recent years due to its increasing significance in modern chip fabrication. Borkar et al. [4] investigate the impact of parameter variations on circuits and microarchitecture and propose the Body Bias Control Techniques to reduce the negative impact by PV. Teodorescu et al. [26] propose using dynamic fine-grain body biasing to mitigate the process variation effect. The analysis on leakage power in the face of process variation is given by Chang and Sapatnekar in [7]. Garg et al. [13] develop a model to describe the system performance under PV.

Our work deviates from the above studies in that we investigate the relationship between microarchitectural configurations and hard-error reliability and extract the promising configurations at early design stages.

3. Methodology

3.1. Patient Rule Induction Method (PRIM)

PRIM is an advanced statistical model [12], the objective of which is to find a region in the input space (composed of configuration parameters in this work) that gives relatively low values for the output response, e.g. the FIT value. The selected region (or "box") is described in an interpretable form involving a set of "rules" depicted as $B = \cap_{j=1}^{p}(x_j \in s_j)$, where x_j represents the jth input variable and s_j is a subset of all possible values of the jth variable. In other words, the identified region B is the intersection of p subsets, each of which is from one of the p input variables.

Fig. 1 illustrates the construction of the "optimal" region, which is composed of two phases: (1) patient successive top-down peeling process and (2) bottom-up recursive pasting process. The top-down peeling starts from the entire space (box B) that covers all the data. In each iteration, a small subbox b within the current box B is removed; we calculate the output mean for the elements remaining in $B - b = \{x \in B \ \& \ x \notin b\}$, performing this operation in each dimension (i.e., try removing a different subbox from each input variable); finally we choose the one which yields the smallest output mean value for the next box $B - b$. This procedure is applied iteratively until the proportion of the data points remaining in the current box (termed the support) is below a preset threshold β. Note that for a categorical variable, an eligible subbox b contains only one element of the possible values of the variable in the current box B. To give an example, let us assume that three values for the L1 data cache size, which is an architectural parameter in the design space, exist in the current box, i.e., s_1 = {8 KB, 16 KB, 32 KB}.

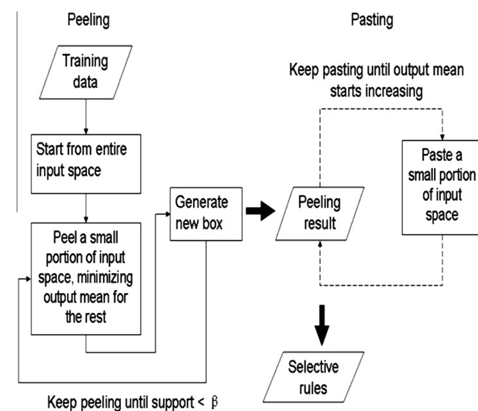


Fig. 1. PRIM training procedure, including peeling and pasting.

With this assumption, there are three eligible subboxes: {x_1 = 8 KB}, {x_1 = 16 KB}, and {x_1 = 32 KB} in this dimension. Therefore, each of these values will be a candidate to be removed in the next iteration.

The pasting algorithm works inversely from the peeling results and the final box can be improved by readjusting its boundaries. The reason for including this step is that we only look one step ahead in each peeling iteration; thus the box boundary is determined without knowledge of later iterations. This implies that we may peel too much from the input space and eliminate many design options unintentionally. In specific, the pasting phase works as follows. Starting with the peeling solution, the current box B is iteratively enlarged by pasting onto it a small subbox that minimizes the output mean in the new larger box. We iteratively apply this process and successively enlarge the current box, until the addition of the next subbox causes the output mean to increase.

According to the above description, it is straightforward to derive that the first peel stage introduces at most $n \times \sum_{j=1}^{p} C_j$ comparisons, where n denotes the number of observations, p is the number of input variables, and C_j indicates the amount of values that the jth variable can take. Each peeling iteration conducts a gradually decreasing number of operations since there are fewer samples left after a peeling iteration. PRIM performs approximately $-\log(n)/\log(1-\alpha)$ peeling steps where α denotes the percentage of points removed at each iteration.

PRIM outstrips many widely-adopted strategies including greedy methods by providing much more stable solutions (hyper-boxes). For instance, a binary tree partitions the data quickly because of its binary splits, while with PRIM only a small portion of data is removed at each iteration. As a consequence, in the case where the training data is slightly changed, a tree structure may change drastically but the PRIM solution is less affected. On the other hand, PRIM is also capable of identifying the optimal region even if it is inconsecutive. In this situation, PRIM will generate a sequence of hyper-boxes instead of only one. That is, the PRIM algorithm can be repeated on the remaining dataset after generating the first hyper-box. By doing so, the disconnected subspace can also be covered. In this work, we found that the leading box often covers most of the points with the small response values, thus we only identify the first hyper-box. Also, recall that the threshold β indicates the percentage of data points that remain in the final hyper-box. We set β to 0.05 in this work. As can be seen from later sections, PRIM is very effective in identifying optimal points. The extracted 5% design space points are usually within top 10–20% optima of the entire design space. The identified points are not necessarily the top 5% optima due to two reasons: (1) the optimal subspace may not be connected, and we only extract the leading one box in this work; (2) our model is evaluated across different programs which could demonstrate different behavior for the metrics in analysis.

3.2. Reliability modeling

The processor hard-error reliability issue is a general term that refers to the hardware degradation or failure caused by different mechanisms during the operations. Existing literature demonstrates that phenomena including electromigration, stress migration, thermal cycling, etc., tend to pose significant threats to processors' reliability. In this work, we pay attention to the following four important failure mechanisms.

3.2.1. Electromigration (EM)

The phenomenon electromigration manifests itself in progressive displacement of metal atoms in a semiconductor device, which gradually leads to opens and shorts in metal lines. The meantime to failure due to EM is usually modeled based on Jim Black's equation [1] as follows:

$$MTTF_{EM} \propto (J - J_{crit})^{-n} e^{\frac{E_a}{kT}} \quad (1)$$

where J and J_{crit} are the current density and the threshold current density, E_a is the activation energy, k is Boltzmann's constant, and T is the temperature in Kelvin.

3.2.2. Stress Migration (SM)

This phenomenon, also termed stress induced voiding, is caused by the hydrostatic stress gradient in the device. Similar to electromigration, SM will cause metal atoms to migrate, forming opens or extremely high resistance in the circuit. The processor degradation caused by stress migration can be approximated through the following model:

$$MTTF_{SM} \propto |T_0 - T|^{-n} e^{\frac{E_a}{kT}} \quad (2)$$

In this expression, T_0 denotes the metal deposition temperature and is set to 500 K according to recent semiconductor studies [1]. Other parameters have the same meaning as used in Eq. (1).

3.2.3. Thermal cycling (TC)

TC degrades the hardware due to the device fatigue. Specifically, a processor inevitably goes through thermal cycles including powering up and down or workload behavior changes. This repetitive temperature variations cause fatigue failures every time, which gradually accumulate throughout the processor's lifespan and result in permanent failures at last. The effect of thermal cycling is usually analyzed using the Coffin–Manson equation [1]:

$$MTTF_{TC} \propto \left(\frac{1}{T - T_{ambient}}\right)^q \quad (3)$$

where $T_{ambient}$ is the ambient temperature and q is the Coffin–Manson exponent, which is set to 2.35.

3.2.4. Negative Bias Temperature Instability (NBTI)

The NBTI mechanism eventually causes processor failure due to timing constraint violations. The silicon-hydrogen bonds within a p-type metal–oxide–semiconductor field-effect transistor (MOSFET) progressively dissociate when a negative voltage is imposed to the device. As a consequence, substantial traps, or holes, are emerging on the gate channel interface, which gradually upshifts the transistor threshold voltage and slows down its switching speed. The degradation due to NBTI is expressed as follows [27]:

$$MTTF_{NBTI} \propto \left\{\left[\ln\left(\frac{A}{1+2e^{B/kT}}\right) - \ln\left(\frac{A}{1+2e^{B/kT}} - C\right)\right] \times \frac{T}{e^{-D/kT}}\right\}^{\frac{1}{\beta}} \quad (4)$$

where A, B, C, D, and β are fitting parameters. Their values are respectively 1.63, 0.074, 0.01, −0.069, and 0.3.

In this work, we adopt the sum-of-failure-rates methodology proposed in [24] to compute the aggregated processor failures due to the aforementioned degradation mechanisms. Specifically, the overall failure rates of a processor is calculated by summing up the failures on each architectural component (e.g., the re-order buffer [ROB], load-store queue [LSQ], and L1 cache) due to individual failure mechanisms.

3.3. Overview of the proposed method

As shown in Fig. 2, our modeling work consists of three major steps: First, a group of representative benchmarks (benchmark 1 through benchmark m) are selected. For each of them, we simulate n configurations (*cfg.1* through *cfg.n*) randomly and uniformly sam-

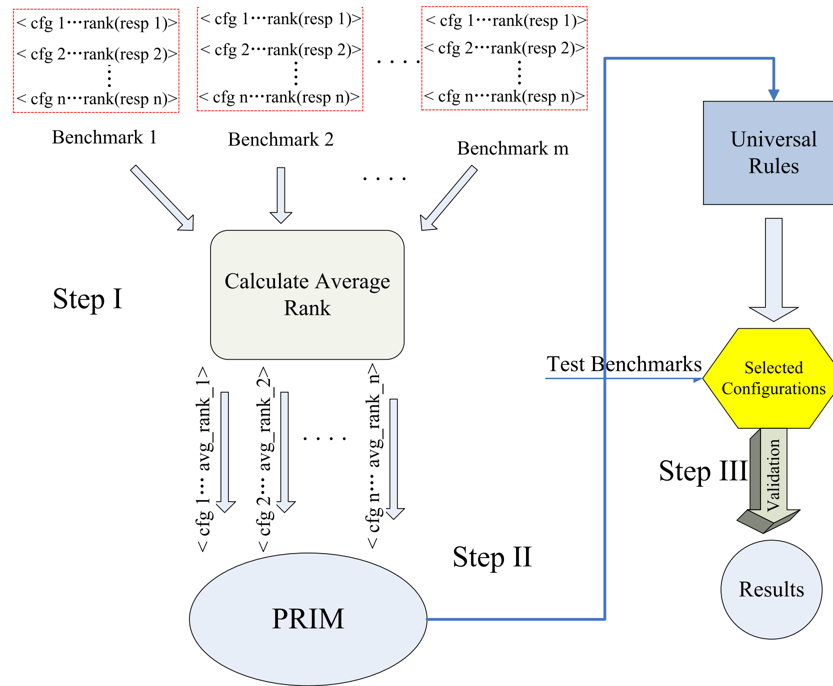


Fig. 2. Procedure of the generation and validation of universal rules for a specific metric.

pled from the entire design space. *Res.1* through *Res.n* are the measured responses corresponding to the sampled *n* configurations. Note that the absolute value ranges of the responses for one benchmark may be different from those of another benchmark. Therefore, it is not reasonable to directly use the configurations and their response values collected from different benchmarks to train a general model. Taking this into consideration, we rank the configurations in each benchmark. Specifically, for the *n* configurations in Fig. 2, the one with the smallest output values, e.g., the shortest execution time or the lowest FIT, is ranked *No. 1* while the one with largest output is ranked *No. n*. We then calculate the average rank for each configuration and use it in the model training. Second, we train PRIM models for optimizing different metrics. Specifically, we generate a set of rules which filter out a design subspace within which each configuration has the optimal output value (performance, power, temperature, or FIT) for different benchmarks. Third, we validate the effectiveness of the universal rules by applying them to benchmarks not used in the training set.

4. Experimental setup

We conduct our evaluation on a modified cycle-accurate simulator SESC [20], which is a widely used tool in computer architecture research community. Wattch [6] and Hotspot [23] are integrated into the simulator for dynamic power and temperature computation, respectively. We implement the failure models described in Section 3.2 for reliability estimation. Those models take as input the runtime power and temperature information collected from the architectural simulation and compute the corresponding processor reliability. Also recall that the overall failure rate is calculated with the sum-of-failure-rates model [24]. Furthermore, considering that our target processor is manufactured with deep submicron technology (i.e., 22 nm in this study), we also derive the leakage power consumption according to the methodology presented in [3]. In general, the leakage power of a processor component is determined by the runtime temperature and its area. The floorplan of the target processor is set based on an Alpha 21264 processor.

We choose 19 benchmarks covering most of the clusters [18,19] in SPEC CPU2000 and SPEC CPU2006 Suite and run them on 2000 configurations randomly and uniformly sampled from the design space for the study. 12 of the programs (*art.SP00*, *bzip2.SP00*, *crafty.SP00*, *equake.SP00*, *parser.SP00*, *dealII.SP06*, *h264.SP06*, *lbm.SP06*, *mcf.SP06*, *milc.SP06*, *sjeng.SP06*, *omnetpp.SP06*) are selected for training the PRIM models while the remaining 7 benchmarks (*applu.SP00*, *gzip.SP00*, *mesa.SP00*, *mgrid.SP00*, *libquantum.SP06*, *namd.SP06*, *soplex.SP06*) are used for the validation.

Table 1
Processor design space in our study.

Parameters	Selected values
Fetch/issue/commit width	2, 4, 6, 8
#ALU/FPU (dependent on width)	1/1, 2/1, associated with 2 2/1, 4/2, associated with 4 3/1, 6/3, associated with 6 4/2, 8/4, associated with 8
Instruction Queue size (IQ)	24, 32, 40, 48, 56, 64
L1D cache size	8, 16, 32, 64 KB
L1D Cache Associativity (L1DA)	1, 2, 4
Reorder Buffer size (ROB)	64–152, with a step of 8
Register File (RF) (dependent on ROB size)	48–144, with a step of 8 RF = ROB size − 8 (or 16)
Load Store Queue size (LSQ)	16–64, with a step of 8
Load/store unit	1/1, 2/2
Branch Target Buffer size (BTB)	1024, 2048
Max branches	8, 16,32
Fixed parameters	**Value**
L1 Instruction cache size	32 KB
L1 Instruction cache associativity	4
L1 Cache block size	64 B
L2 Cache size	2 MB
L2 Cache associativity	4
L2 Cache block size	64 B
Technology	22 nm
Frequency	2 GHz

Table 1 lists the parameters included in our design space. Note that the size of the L2 cache is fixed in this exploration. This is a rational setting for a hard-error reliability study because the L2 cache usually occupies a large portion of the chip area for heat spreading and thus results in much lower temperature compared to the processor core. As a consequence, the L2 cache is likely to show very low thermal–intrinsic failure rates [24,25]. Also note that the operating point (i.e., voltage/frequency) is set as a constant in this work. This is because changing voltage and frequency would be more effective for reliability optimization while explored at runtime [8]. In total, our exploration space contains 1,161,216 points.

5. Result analysis

5.1. Optimizing hard-error reliability and other metrics

We first train a PRIM model to identify the design patterns that benefit the processor lifetime reliability. In order to gain insights into the interaction between microarchitectural configurations and the reliability of the target processor, we also generate a set of "individual" rules for each training benchmark in addition to the universal rule set. Note that the absolute FIT values collected from a benchmark can be used to extract the individual rules. Table 2 lists the rules for optimizing hard-error reliability for the 12 training benchmarks. The universal rule set which is summarized by following the steps described in Section 3.3 is shown at the bottom. For example, the rule set of *art* from SPEC2000 (i.e., *art.SP00*) suggests that a processor should be configured with fetch/issue/commit width of 8 and the number of ALU/FPU at either 4/2 or 8/4, while the ROB/RF size should not exceed 128/112.

In general, we observe that processors equipped with intermediate size structures demonstrate the optimal reliability. The reason is twofold. Let us use the selection of ROB size as an example for illustration. As can be noted from Table 2 and 11 out of 12 training benchmarks eliminate fairly large ROBs (i.e., with size from 128 to the largest) from the identified subregion, with an exception *parser.SP00* filtering out the extremely small ROBs as well.

To understand the reason behind this phenomenon, we conduct a sensitivity study to investigate the correlation between ROB size and FIT by running *art.SP00*, which is selected as the representative of the majority (i.e., the 11 benchmarks), and *parser.SP00*. Specifically, we fix all architectural parameters but the ROB size, and observe how the processor reliability would change with the ROB size. Fig. 3(a) and (b) respectively demonstrate the results including FIT, temperature and ROB occupancy rate for these two benchmarks. As can be seen from the plots, in the case of *art.SP00*, the FIT gradually increases when the ROB is enlarged, meaning that smaller sized reorder buffer would be more desirable for endurable operation; on the contrary, processors running *parser.SP00* show a reverse trend that high failure rates manifest on relatively small ROBs. The reason is as follows. The processor reliability is a complex function of multiple factors including power, temperature, area, voltage/frequency (i.e., the operating point), and device features, while the operating point and device are not changed in this

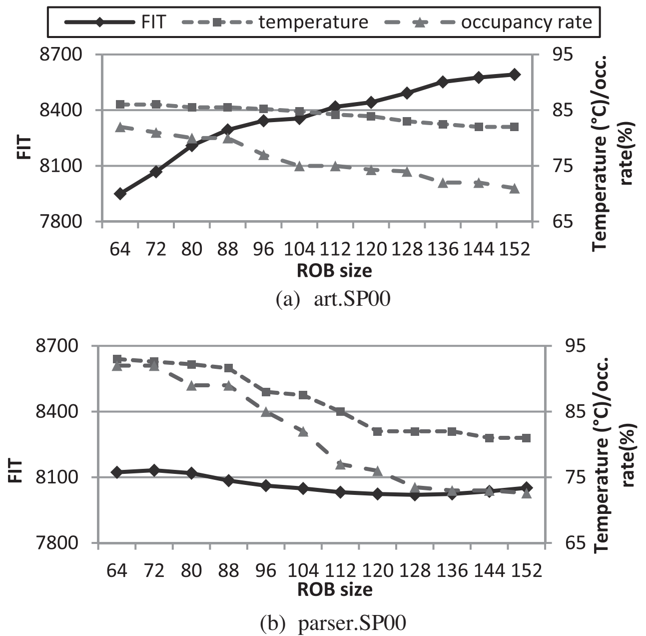


Fig. 3. Variation of FIT, ROB temperature and occupancy rate with ROB size changing.

work. For *art.SP00*, the variation of ROB occupancy rates is moderate, resulting in mild power density and temperature variation. In this situation, the increasing structure area corresponding to large ROB plays a more important role in determining the overall failure rate, since the FIT of a structure is positively correlated to its area [24]. However, running *parser.SP00* with small ROBs leads to occupancy rates higher than 90%. This implies that substantial power tends to be consumed on small area, resulting in extremely high power density and temperature, which essentially translates to a soar of failure rates. Note that many failure mechanisms have exponential dependence on the temperature. Examples include the reliability issue caused by electromigration (EM), which can be estimated via Eq. (1) given in Section 3.2. Due to the exponential relationship, the impact of high temperature outweighs that caused by area when small-ROB processors are running *parser.SP00*. In other words, extremely small ROBs lead to even higher failure rates than large ROBs. Therefore, the intermediate size ROBs are the most promising configuration from the perspective of reliability optimization.

The generated rule set also indicates that large load-store queues are not beneficial for endurable and reliable operations on the processor. The essential reason of this is similar to the description of the ROB selection for *art.SP00* (and most other benchmarks) given in the previous paragraph. Specifically, the LSQ, which is used to store the on-the-fly memory operations during program execution, is a highly utilized structure on a modern out-of-order processor, making the LSQ is a noticeable hot spot on the chip. In other words, the LSQ tends to show high tempera-

Table 2
Rules for optimizing reliability.

App.	Rules	App.	Rules
art.SP00	Width/ALU ⩽ 4/4/2 && ROB/RF ⩽ 144/136	*h264.SP06*	Width/ALU ⩽ 4/4/2 && L1D ⩽ 16 KB && ROB/RF ⩽ 144/136
bzip2.SP00	Width/ALU ⩽ 4/4/2 && ROB/RF ⩽ 136/120 && LSQ < 56	*lbm.SP06*	Width/ALU ⩽ 6/6/3 && L1DA < 4 && ROB/RF ⩽ 136/128
crafty.SP00	Width/ALU ⩽ 4/4/2 && ROB/RF ⩽ 120/112 && LSQ < 64	*mcf.SP06*	Width/ALU ⩽ 6/3/1 && ROB/RF ⩽ 144/136
equake.SP00	Width/ALU ⩽ 4/2/1 && L1D ⩽ 32 KB && ROB/RF ⩽ 136/120	*milc.SP06*	Width/ALU ⩽ 4/4/2 && L1DA < 4 && ROB/RF ⩽ 128/120
parser.SP00	Width/ALU ⩽ 6/6/3 && 88/80 ⩽ ROB/RF ⩽ 136/128	*sjeng.SP06*	Width/ALU ⩽ 4/4/2 && ROB/RF ⩽ 120/112 && LSQ < 56
dealII.SP06	Width/ALU ⩽ 4/2/1 && ROB/RF ⩽ 128/120 && LSQ < 56	*omnetpp.SP06*	Width/ALU ⩽ 4/2/1 && L1D ⩽ 32 KB && ROB/RF ⩽ 136/128
Universal rules	(Width/ALU ⩽ 4/4/2) && (L1D ⩽ 16 KB) && (L1DA ⩾ 2) && (72/56 ⩽ ROB/RF ⩽ 112/104) && (LSQ ⩽ 48)		

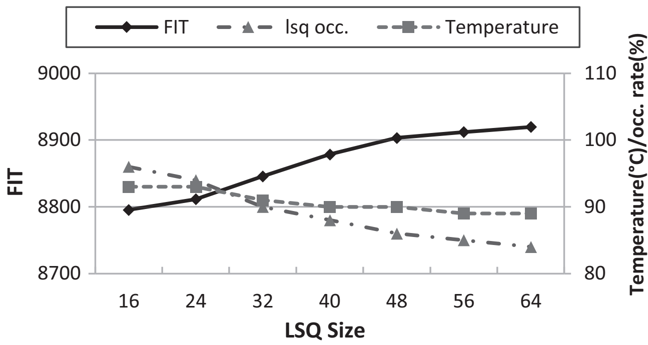


Fig. 4. Variation of FIT, LSQ temperature and occupancy rate with LSQ size changing while running *bzip2.SP00*.

ture in many execution scenarios. As a consequence, the structure area stands as the determinative factor to FIT, which implies that extremely large load-store queues will degrade the overall reliability. In order to illustrate this trend, we collect the variation of FIT with gradually increasing LSQ size while running *bzip2.SP00* and show it along with the changes of temperature and LSQ occupancy rates in Fig. 4. Note that all other architectural parameters are identical across these configurations. As can be observed, the temperature is persistently high in all cases because of the high utilizations on the load-store queue. Therefore, building relatively smaller LSQ is effective in lowering down the failure rate, i.e., improving the reliability. Similar observations can be made from the executions of other benchmarks.

We also generate universal rules to optimize the performance and power consumption. In general, the largest structures (e.g., larger ROB/RF, cache, LSQ, etc.) are selected for a high-performance processor while more conservative configurations (smaller structures) are identified when low-power is the most important design goal. We visualize the configuration selection for these individual metrics in Fig. 5(a).

5.2. Optimizing different tradeoffs

The tradeoff among performance, power dissipation and reliability is becoming increasingly attractive. In this work, we define the following measurement to evaluate the balance among multiple design goals, where a smaller value implies a better tradeoff,

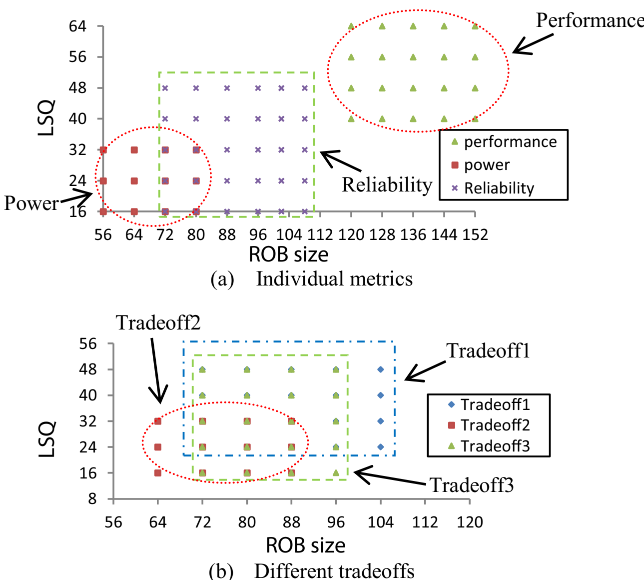


Fig. 5. ROB and LSQ selections for individual metrics and different tradeoffs.

$$(execution\ times^a) \times (power^b) \times (FIT^c) \quad (5)$$

where $a + b + c = 1$. We first focus on the tradeoff by which high performance is emphasized. For simplicity, we fix a to 0.5 and set the values of both b and c to 0.25. As illustrated in Fig. 5(b), larger ROBs and LSQs are suggested in this circumstance. However, these choices differ from those selected when performance is the exclusive design goal. In particular, moderate sized ROBs and LSQs are preferred by this tradeoff instead of extremely aggressive configurations.

In the second case, b is fixed to 0.5 while a and c both take the values of 0.25. By doing this, we try to extract the optimal configurations when low power is more important. Similarly, a, b and c are respectively set to 0.25, 0.25 and 0.5 in the third study, which is conducted when high reliability is preferred. Fig. 5(b) clearly demonstrates that the optimal subspace for the second and the third tradeoffs contain more conservative configurations compared to those for the first one. Again, this is because larger components incline to be more power-hungry and become vulnerable blocks due to large area occupation.

5.3. Rules validation

In this subsection, we validate the universal rule sets for optimizing the four individual metrics and the tradeoffs.

We adopt the bootstrapping method [11] to estimate where the selected points are actually located in the entire design space (not just the 2000 points used for training) and use boxplot to demonstrate the validation results. In a boxplot, the lower and upper boundaries of the central gray box correspond to the 25% and 75% quantiles; the bold line within the box is at the median; the vertical dotted line drawn from the boundaries extend to the minimum and maximum. For instance, for *gzip.SP00* in the validation of reliability rules in Fig. 6 (the third plot), the maximum of points selected by the universal rules corresponds to a value of 7.3% in the vertical axis, meaning that for this benchmark all selected points are within the top 7.3% optima of the entire design space. We observe that the design points filtered by the universal rules remain within the top 10–15% optima for the three individual metrics on average, which justifies the effectiveness of the selected rules. For the three tradeoffs, Fig. 7 illustrates that the approximately top 10% optima are extracted by the corresponding rule sets.

6. Optimal configuration selection under process variation

As described in Section 1, the ever-increasing process variation results in wide distributions of key design parameters. This largely impacts the performance, power, and reliability features of the target processor. Considering the significance of process variation in modern chip fabrication, we extend our studies to evaluate the design configurations in the presence of PV.

We adopt the VARIUS tool [21] to model typical within-die PV effects encountered in modern processor manufacturing. VARIUS assumes that the design parameters follow a normal distribution determined by a mean value μ and a standard deviation σ. In this work, we focus on the variation of threshold voltage V_{th} because of its significant impact on the processor frequency and power consumption. We set μ to 150 mV and evaluate 5 representative σ values ranging from 0.03 to 0.15 [21]. To model the process variation, we generate the V_{th} variation map using the VARIUS model, and map the V_{th} into each critical path based on the Alpha 21264 floorplan. We generate 400 chips for each set of configurations for statistical analysis. Finally, for each chip, the frequency is determined by the critical path with the highest threshold voltage.

The σ on V_{th} gradually increases as the feature size scales down [2]; thus the configuration selection rules obtained under a certain

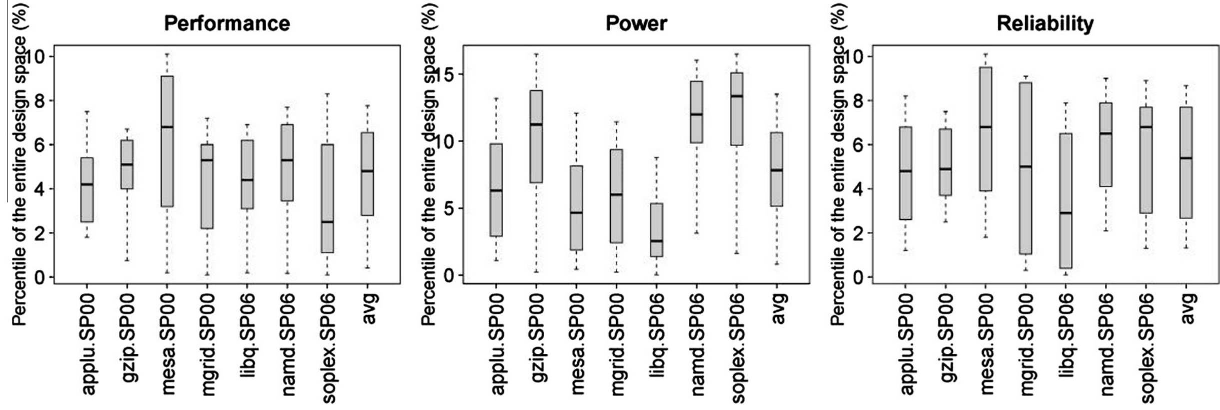


Fig. 6. Validation results of rules for individual metrics.

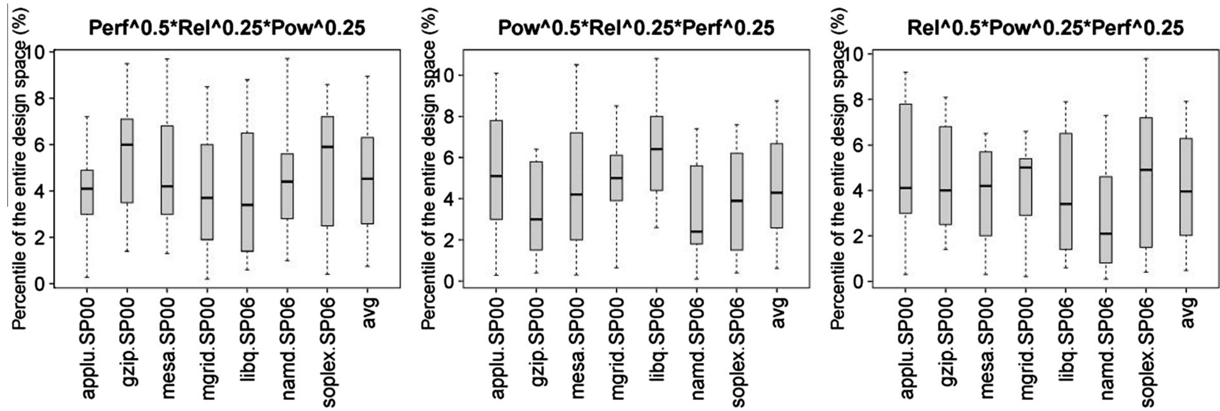


Fig. 7. Validation results of rules for different tradeoffs.

Table 3
Universal rules for optimizing power under different standard deviations.

σ value	Rules
0.03	(width/ALU ⩽ 4/2/1) && (L1D ⩽ 16 KB) && (ROB/RF ⩽ 88/80) && (LSQ ⩽ 24)
0.06	(width/ALU ⩽ 4/2/1) && (L1D ⩽ 16 KB) && (ROB/RF ⩽ 88/80) && (LSQ ⩽ 24)
0.09	(width/ALU ⩽ 4/4/2) && (L1D ⩽ 16 KB) && (ROB/RF ⩽ 88/72) && (LSQ ⩽ 16)
0.12	(width/ALU ⩽ 4/4/2) && (L1D ⩽ 8 KB) && (ROB/RF ⩽ 80/72) && (LSQ ⩽ 16)
0.15	(width/ALU ⩽ 4/4/2) && (L1D ⩽ 8 KB) && (ROB/RF ⩽ 72/64) && (LSQ ⩽ 16)

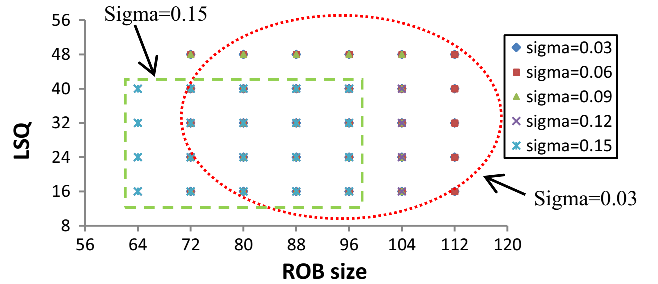


Fig. 8. ROB and LSQ selection for optimal reliability under different sigma values.

σ may not be applicable to other cases since the V_{th} distribution tend to vary significantly on a given architectural configuration. One possible solution is to explore the rules fitting to all the cases with different σ. However, in that case, the generated rules will be extremely strict on the configuration selection which should be avoided during the design process. Considering that the σ is usually known at the design time, we will extract separate rules for each case individually.

6.1. Optimal design options for processor power and reliability

Process variation tends to significantly impact the power consumption of a processor, especially for the leakage power. The dynamic power is less sensitive to the variation because of its approximate linear relation to the process parameters [16]. Therefore, in this section, we mainly concentrate on the change of leakage power and the resultant varying total power. A previous work [26] demonstrates that the post-variation leakage power can be calculated with the following equation,

$$\frac{P_{leak}}{P^0_{leak}} = e^{\frac{\left(\frac{q\sigma}{\eta KT}\right)^2}{2}} \tag{6}$$

In the above formula, P_{leak} and P^0_{leak} denote the leakage consumed by a processor with and without PV, respectively, while q, η, and K are constants. T is the chip temperature. Table 3 lists the universal rules for power optimization under different σ (sigma) values. As can be noted, different distributions of the threshold voltage lead to disparate preferences on the configuration selection. In general, more conservative design options are preferred when the variation is large (i.e., higher σ value). For example, the most suitable ROB/RF size should not exceed 88/80 when σ is

Table 4
Universal rules for optimizing performance under different standard deviations.

σ value	Rules
0.03	(width/ALU ⩾ 6/6/3) && (L1D ⩾ 32 KB) && (L1DA ⩾ 2) && (ROB/RF ⩾ 128/112) && (LSQ ⩾ 32) && (LSUnit = 2/2)
0.06	(width/ALU ⩾ 6/6/3) && (L1D ⩾ 32 KB) && (L1DA ⩾ 2) && (ROB/RF ⩾ 120/104) && (LSQ ⩾ 32) && (LSUnit = 2/2)
0.09	(width/ALU ⩾ 6/6/3) && (L1D ⩾ 32 KB) && (ROB/RF ⩾ 120/104) && (24 ⩽ LSQ ⩽ 56)
0.12	(width/ALU ⩾ 6/6/3) && (L1D ⩾ 32 KB) && (ROB/RF ⩾ 120/104) && (24 ⩽ LSQ ⩽ 56)
0.15	(width/ALU ⩾ 6/6/3) && (L1D ⩾ 32 KB) && (120/104 ⩽ ROB/RF ⩾ 144/128) && (24 ⩽ LSQ ⩽ 56)

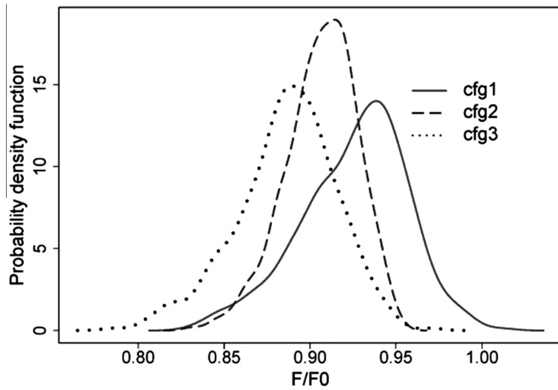


Fig. 9. Probability of frequency distribution under PV (σ = 0.12).

0.03; however, the ROB/RF size is confined within 72/64 while σ becomes 0.15. This is reasonable according to Eq. (3). As the formula indicates, the leakage power will be exponentially increased while the V_{th} variation is getting larger. Therefore, architectural components including large reorder buffer and physical register file, which occupies considerable chip area, will consume substantial static power when the σ value is large. As a consequence, smaller sized ROB and RF should be configured in order to enclose the total power consumption within a reasonable envelope.

The hard-error reliability of a processor is essentially determined by its runtime power consumption and chip temperature. Therefore, the universal rules for hard-error resilient processors demonstrate a similar trend to those for low-power designs. Specifically, higher σ values require a processor to be configured with smaller structures for durable running, because such a configuration is effective to decrease the power consumption and enhance the reliability consequently. We visualize this trend of configuration selection in Fig. 8.

6.2. Optimal design options for system performance

We now shift our concentration to the optimal configurations for system performance. As the delays of critical paths are non-equivalent across the whole chip, the ultimate performance is limited by the slowest critical path (or the lowest frequency). We use the integration *instructions per cycle (IPC)* $\times$ *frequency* to assess the performance delivered by each design configuration, where the frequency depends on the critical path with the longest delay. Table 4 lists the universal rules to guide the design of high-performance processors. As shown in the table, increasing the σ value poses noticeable impact on the configuration selection. In the face of small variation, the most aggressive design options are selected. An example is that the ROB/RF size of the processor should be no smaller than 128/112 (up to the largest 152/144 in our design space) to ensure high performance when the standard deviation is 0.03. On the contrary, the most appropriate ROB/RF size is confined within 120/104 and 144/128 when σ becomes 0.15. The reason is as follows. A processor with the largest structures does not necessarily lead to the optimal performance in the presence of high parameter variation. Aggressive microarchitectural designs increase the number of critical paths; therefore, the mean frequency of the chip tends to decrease as the probability that a critical path suffers from long delay (i.e., high V_{th}) is increasing. That is, extremely large architectural components are more vulnerable to the process variation since the critical paths are highly probable to be slowed down by a high V_{th}. To visualize this trend, we choose three configurations *cfg1* $\sim$ *cfg3* with gradually increasing structure size (e.g., increasing ROB/RF, LSQ, etc.) and plot the frequency distribution of the corresponding processor in Fig. 9. As can be observed, the mean value of the frequency obviously decreases while we increase the structure size. If this frequency degradation outweighs the benefit obtained from the improved IPC, the overall performance may still be impaired. On the other hand, when the V_{th} variation is mild (e.g., σ = 0.03), high IPC still translates to shorter execution time in most cases; thus the largest ROB/RFs are chosen for a high-performance processor in that scenario.

Finally, we use the boxplot to validate the rules for performance, power and reliability, respectively, and exemplify it with the validation of rules for reliability in Fig. 10. We observe that the top 20% optima can be identified under all different σ values.

7. Conclusion

In this paper, we use an advanced statistical technique to bridge the gap between processor hard-error reliability and its microarchitectural configuration. We find that performance, power consumption, and reliability favor different configurations. We also investigate the optimal balance among individual metrics. Moreover, we take

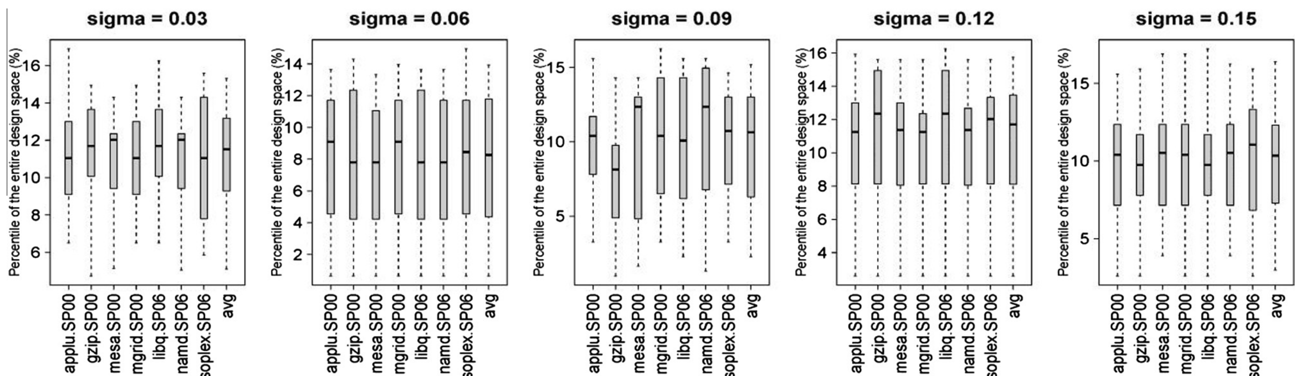


Fig. 10. Validation results of rules for reliability under different sigma values.

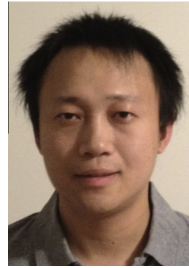
the process variation effect into consideration, in order to identify the most promising architectural configurations for different design goals in a non-ideal manufacturing environment. The evaluation results demonstrate that our strategy is effective for generating rules to assist the design of future processors.

Acknowledgement

This work was supported in part by a US National Science Foundation (NSF) grant CCF-1017961, the Louisiana Board of Regents grant NASA/LEQSF (2005-2010)-LaSPACE and NASA grant number NNG05GH22H, NASA(2011)-DART-46, LQESF(2011)-PFUND-238 and the Chevron Innovative Research Support (CIRS) Fund. The authors acknowledge the computing resources provided by the Louisiana Optical Network Initiative (LONI) HPC team. Finally, the authors appreciate invaluable comments from anonymous reviewers.

References

[1] Failure mechanisms and models for semiconductor devices, in: JEDEC, Publication JEP122-A, 2002.
[2] M. Bennaser, Y. Guo, C.A. Moritz, Data memory subsystem resilient to process variations, in: IEEE TVLSI, vol. 16, December 2008.
[3] S. Biswas, M. Tiwari, T. Sherwood, L. Theogarajan, F.T. Chong, Fighting fire with fire: modeling the datacenter-scale effects of targeted superlattice thermal management, in: ISCA' 11.
[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, V. De, Parameter variations and impact on circuits and microarchitecture, in: DAC' 03.
[5] F.A. Bower, D.J. Sorin, S. Ozev, A mechanism for online diagnosis of hard faults in microprocessors, in: MICRO' 05.
[6] D. Brooks, V. Tiwari, M. Martonosi, Wattch: a framework for architectural-level power analysis and optimizations, in: ISCA' 00.
[7] H. Chang, S.S. Sapatnekar, Full-chip analysis of leakage power under process variations, including spatial correlations, in: DAC'05.
[8] A.K. Coskun, R. Strong, D.M. Tullsen, T.S. Rosing, Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors, in: SIGMETRICS' 09.
[9] A. Dasgupta, R. Karri, Electromigration reliability enhancement via bus activity distribution, in: DAC' 96.
[10] L. Duan, Y. Zhang, B. Li, L. Peng, Universal rules guided design parameter selection for soft error resilient processors, in: ISPASS' 11.
[11] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall/CRC, 1994.
[12] J. Friedman, N. Fisher, Bump hunting in high-dimensional data, Statistics and Computing 9 (1999) 123–143.
[13] S. Garg, D. Marculescu, S.X. Herbert, Process variation aware performance modeling and dynamic power management for multi-core systems, in: ICCAD' 10.
[14] C.-H. Hsu, W.-C. Feng, Reducing overheating-induced failures via performance-aware CPU power management, in: The LCI International Conference on Linux Clusters: The HPC Revolution, April 2005.
[15] C.-H. Hsu, W.-C. Feng, A power-aware run-time system for high-performance, computing, in: SC'05.
[16] Y. Lu, V.D. Agrawal, Total power minimization in glitch-free CMOS circuits considering process variation, in: VLSI' 08.
[17] M. Mochiero, R. Canal, A. Gonzalez, Design space exploration for multicore architectures: a power/performance/thermal view, in: ICS' 06.
[18] A. Phansalkar, A. Joshi, L.K. John, Analysis of redundancy and application balance in the SPEC CPU2006 benchmark suite, in: ISCA'07.
[19] A. Phansalkar, A. Joshi, L. Eeckhout, L.K. John, Measuring program similarity: experiments with SPEC CPU benchmark suites, in: ISPASS' 05.
[20] J. Renau et al., SESC Simulator, 2005. <http://sesc.sourceforge.net/>.
[21] S.R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, J. Torrelas, VARIUS: a model of process variation and resulting timing errors for microarchitectures, in: IEEE Transactions on Semiconductor Manufacturing, vol. 21, February 2008.
[22] D.K. Schroder, J.A. Babcock, Negative bias temperature instability: road to cross in deep submicron silicon semiconductor manufacturing, Journal of Applied Physics 94 (July) (2003) 1–18.
[23] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, D. Tarjan, Temperature-aware microarchitecture, in: ISCA' 03.
[24] J. Srinivasan, S.V. Adve, P. Bose, J.A. Rivers, The case for lifetime reliability-aware microprocessors, in: ISCA' 04.
[25] J. Srinivasan, S.V. Adve, P. Bose, J.A. Rivers, The impact of technology scaling on lifetime reliability, in: DSN' 04.
[26] R. Teodorescu, J. Nakano, A. Tiwari, J. Torrellas, Mitigating parameter variation with dynamic fine-grain body biasing, in: MICRO' 07.
[27] S. Zafer et al., A model for negative bias temperature instability (NBTI) in oxide and high K PFETs, in: Symposia VLSI Technology and Circuits, 2004.
[28] Y. Zhang, L. Duan, B. Li, L. Peng, Optimal microarchitectural design configuration selection for processor hard-error reliability, in: ISQED' 12.



Ying Zhang received the Bachelor's and Master's degree in Electronics and Information Engineering from Huazhong University of Science and Technology, China, in June 2006 and 2008. He is currently a Ph.D. student in Electrical and Computer Engineering, Louisiana State University. His research interests include GPU performance characterization and hard-error reliable processor design. He also has interests in energy-efficiency optimization for heterogeneous architectures.

Lide Duan received the BS degree in Computer Science and Engineering from Shanghai Jiao Tong University, China, and the Ph.D. degree in Electrical and Computer Engineering from Louisiana State University. He is currently working as a senior design engineer in core modeling for AMD's x86 architectures. His research interests include computer architecture, soft error reliability analysis and prediction, and application-level error propagation prediction. He received a Graduate Fellowship from Louisiana Optical Network Initiative (LONI) and the Dissertation Year Fellowship from LSU Graduate School during his Ph.D. study.

Bin Li received his Bachelor's degree in Biophysics from Fudan University, China. He obtained his Master's degree in Biometrics (08/2002) and Ph.D. degree in Statistics (08/2006) from The Ohio State University. He is an Associate Professor with the Experimental Statistics department at Louisiana State University. His research interests include statistical learning & data mining, statistical modeling on massive and complex data, and Bayesian statistics. He received the Ransom Marian Whitney Research Award in 2006 and a Student Paper Competition Award from ASA on Bayesian Statistical Science in 2005. He is a member of the Institute of Mathematical Statistics (IMS) and American Statistical Association (ASA).

Lu Peng is currently an Associate Professor with the Division of Electrical and Computer Engineering at Louisiana State University. He received the Bachelor's and Master's degrees in Computer Science and Engineering from Shanghai Jiao Tong University, China. He obtained his Ph.D. degree in Computer Engineering from the University of Florida in Gainesville in April 2005. His research focus on memory hierarchy system, reliability, power efficiency and other issues in CPU design. He also has interests in Network Processors. He received an ORAU Ralph E. Powe Junior Faculty Enhancement Awards in 2007 and a Best Paper Award from IEEE International Conference on Computer Design in 2001. He is a member of the ACM and the IEEE Computer Society.

Xin Fu is an assistant professor at the Department of Electrical Engineering and Computer Science, University of Kansas. She received Bachelor in Computer Science and Engineering from Central South University, China in 2003; and obtained Ph.D. in Computer Engineering from the Department of Electrical and Computer Engineering at University of Florida in 2009. She was named as 2009 Computing Innovation Fellow, and 2012 Kansas NSF EPSCoR First Award.