Optimal Power/Performance Pipelining for Error Resilient Processors

Nicolas Zea, John Sartori, Ben Ahrens, Rakesh Kumar
Electrical & Computer Eng. Dept., Univ. of Illinois, Urbana-Champaign, IL, USA
{*nzea2, sartori2, bahrens2, rakeshk*}*@illinois.edu*

***Abstract*— Timing speculation has been proposed as a technique for maximizing the energy efficiency of processors with minimal loss in performance. A typical implementation of timing speculation involves speculatively reducing the voltage of a processor to a point where errors are possible but rare, and employing an error recovery mechanism to ensure correct functionality. This allows significant energy savings with a small recovery overhead.**

Previous work on timing speculation has either explored the benefits of customizing the design methodology for a particular error resilience mechanism or has attempted to understand the benefits from error resilience for a particular processor design. There is no work, to the best of our knowledge, that attempts to understand the benefits of co-optimizing microarchitecture and error resilience.

In this paper, we present the first study on co-optimizing a processor pipeline and an error resilience mechanism. We develop an analytical model that relates the benefits from error resiliency to the depth of the pipeline as well as its circuit structure. The model is then used to determine the optimal pipeline depth for different energy efficiency metrics for different error resilience overheads. Our results demonstrate that several interesting relationships exist between error resilience and pipeline structure. For example, we show that there are significant energy efficiency benefits to pipelining an architecture for an error resiliency mechanism vs error resiliency-agnostic pipelining. As another example, we show that benefits from error resiliency are greater for short pipelines than long pipelines. We also confirm that the benefits from error resiliency are higher when the circuit structure is such that error rate increases slowly on reducing input voltage vs a circuit optimized for power where a slack wall exists at the nominal operating point. Finally, we quantify the difference in benefits from error resiliency for irregular vs regular workloads and show that benefits from error resiliency are higher for irregular workloads. Our analytical results were validated using a cycle-accurate simulation-based model.

I. Introduction

Timing speculation [1] has been proposed as a technique for maximizing energy efficiency of processors with minimal loss in performance. A typical implementation of timing speculation involves speculatively reducing the voltage of a processor to a point where errors are possible but rare, and employing an error recovery mechanism to ensure correct functionality. This allows significant energy savings with a small recovery overhead.

Previous work on timing speculation has either explored the benefits of customizing the design methodology for a particular error resilience mechanism [2], [3], [4], [5] or has attempted to understand the benefits from error resilience for a particular processor design [1], [6], [7]. There is no work, to the best of our knowledge, that attempts to understand the benefits of co-optimizing microarchitecture and error resilience.

In this paper, we present the first study on co-optimizing a processor pipeline and an error resilience mechanism. We develop an analytical model that relates the benefits from error resiliency to the depth of the pipeline as well as its circuit structure. Our model builds upon Hartstein and Puzak's model for optimizing pipeline depth considering both power and performance [8]. We have added a model for voltage overscaling to enhance power savings and then modeled different relationships between voltage overscaling and timing error rate. Also, we model different overheads for error recovery. The overhead of error recovery may either be fixed or depend on the length of the processor's pipeline. The new model allows us to optimize both the pipeline depth and operating voltage for a given error recovery mechanism.

Our results demonstrate that several interesting relationships exist between error resilience and pipeline structure. We show that not only can the optimal pipeline depth be significantly different when error resilience is taken into account, but that different error resilience mechanisms (as reflected by their recovery overhead) impact the architecture differently. We additionally explore the importance of other architectural and workload parameters on the effects of error resilient designs. Finally, we demonstrate that optimizing an architecture without considering error resiliency results in sub-optimal energy efficiency benefits. We explain why this is the case and show that optimal architectures should take error resilience mechanisms into consideration.

Section II discusses related work. Section III describes our analytical model that relates the impact of error resilience and voltage overscaling to the structure of a processor's pipeline. Section IV discusses our simulation and analytical methodology. Section V presents results and analysis. Section VI summarizes and concludes.

II. Related Work

Timing speculation and voltage overscaling have been studied extensively to improve yield and to reduce processor power consumption. However, previous work focuses on studying the benefits from such approaches for a *given* processor design [1], [6], [9], [10] or understanding the benefits from a custom design methodology for a given error resilience mechanism [2], [3], [4], [5]. We attempt to understand the interaction between a processor's pipeline structure and the effectiveness of an error resilience mechanism when applied to it.

Optimal pipelining has also been studied significantly. Hrishikesh et al. [11] determined that the optimal logic depth per pipeline stage is 6 to 8 FO4 delays when considering performance only. Hartstein and Puzak built on power models from Srinivasan et al. [12] to develop an analytical model that determines the optimal pipeline depth for metrics that consider both power and performance [8]. We build on Harstein and Puzak's model to develop a model that determines the optimal pipeline depth for processors that tolerate voltage overscaling-induced timing errors.

The closest work is by de Kruijf et al. [13] who develop a performance/power model for understanding the effectiveness of timing speculation for different process technologies, power designs, and error recovery techniques. Their work is focused on understanding the efficiency of timing speculation for a *given* architecture. We attempt to understand the benefits of co-optimizing a processor's pipeline and circuit structure and error resilience strategy.

III. Theory

A. Baseline

First, we consider the analytical model developed by Hartstein and Puzak [8] for optimizing a processor pipeline for a metric that considers power and performance ($\text{Metric}_{P/P}$).

$$Metric_{P/P} = 1/((T/N_I)^m P_T) \tag{1}$$

This is composed of the following two parts:

$$T/N_I = 1/(f_s a) + (\gamma_h N_h p)/f_s \tag{2}$$

and

$$P_T = (f_{cg} f_s P_d + P_l) N_L p^\eta \tag{3}$$

where m in Equation 1 is the exponential weighting for delay in the energy efficiency metric, T/N_I, defined in Equation 2, is the average CPI of the system, and P_T, defined in Equation 3, is the average power consumption. Following [8]'s example, we use $m = 3$ for our studies unless mentioned otherwise.

Common to both Equations 2 and 3 are the p and f_s variables. p represents the pipeline depth of the processor and is varied in the optimization process. f_s is defined as the operating frequency, and is derived from:

$$f_s = 1/(t_o + t_p/p) \tag{4}$$

where t_o is the latch delay and t_p is the logic delay of the full pipeline.

The CPI equation is composed of two parts, the busy time and the non-busy time. The busy time is simply the frequency weighted by the superscalar width factor, a, representing the average amount of ILP per cycle for a workload. The non-busy time uses a single variable, N_h, defined as the fraction of all instructions which might cause hazards. These hazards include mispredictions, structural hazards, data dependence stalls, etc. γ_h is the average performance penalty factor for hazards. It represents the fraction of pipeline stages which must stall/bubble when a hazard occurs. Because it is a fraction of the pipeline stages, the non-busy time is weighted by p in addition to the clock period $1/f_s$.

The power equation, derived from Srinivasan et al's work [12] includes three components: dynamic power, leakage power, and a latch growth factor. Dynamic power is represented by P_d, the average dynamic energy/cycle per latch (note that these units are not in watts), weighted by the clock gating factor, f_{cg} and the frequency. The clock gating factor is 1 when no clock gating is performed, and less than 1 for different degrees of clock gating. A f_{cg} value of 0.3 is considered to be an aggressively clock gated design. P_l represents the average leakage power per latch in energy/second or watts. Because both these power values are per latch, they are weighted by the average number of latches per stage, N_L. The latch growth component of the system accounts for the superlinear growth in latches as pipeline depth increases, argued by Srinivasan et al in [12]. This is represented by η, the latch growth factor.

By accounting for workload variation in hazards and ILP and architectural variation in delays and power consumption, Equation 1 is able to optimize the number of pipeline stages for particular architectures based on an energy efficiency metric.

B. Modeling Voltage Overscaling and Error Resilience

The key to modeling voltage overscaling and error resilience is accounting for the power and reliability impact of overscaling and the performance impact of error recovery. The magnitude of voltage overscaling directly determines the power savings and the timing error rate. The error rate, given an error recovery mechanism and the associated recovery cost, determines the performance penalty.

The performance cost of error recovery can be modeled as:

$$T_{err}/N_I = \gamma_e e p (T_o/N_I) \tag{5}$$

where γ_e is the average number of pipeline stages delayed by error recovery, p is the number of pipestages for that design, e is the average number of errors per cycle (the error rate), and T_o/N_I is the CPI of the system described in Equation 2. When the cost of error recovery is independent of the total number of pipestages, the performance cost of error recovery can be modeled as:

$$T_{err}/N_I = \gamma_e e c (T_o/N_I) \tag{6}$$

where c is a constant. The overhead of error recovery calculated as above can then be added to the CPI in Equation 2. The new performance (CPI) equation that accounts for the overhead of error recovery is:

$$T/N_I = 1/(f_s a) + (\gamma_h N_h p)/f_s + T_{err}/N_I \tag{7}$$

To model the impact of voltage overscaling on processor power and reliability, we introduce a voltage overscaling factor, f_v. We scale the dynamic power quadratically with

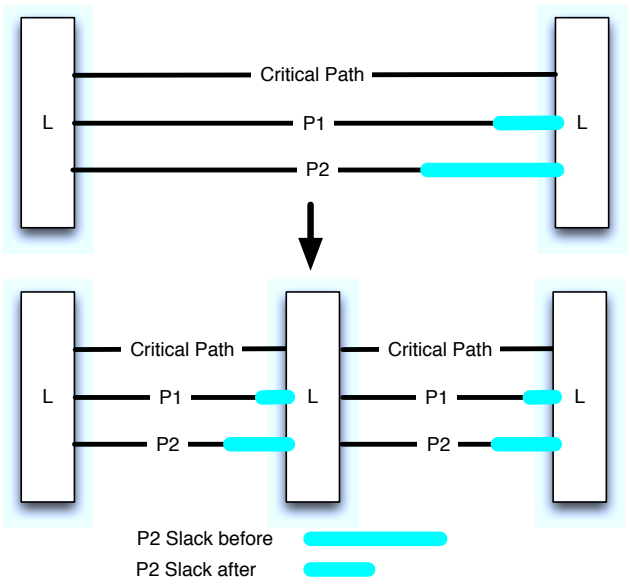

Fig. 1. The effect of pipelining on the slack of a design (the highlighted portion denotes the path slack). When a logic stage is pipelined, the absolute length of the timing paths, and therefore the amount of slack per stage, is reduced. This causes more errors for a given absolute reduction in voltage.

voltage. Leakage power is scaled linearly with voltage. Our new power model is as follows:

$$P_T = (f_{cg} f_s P_d {f_v}^2 + P_l f_v) N_L p^{\eta} \tag{8}$$

For modeling the relationship between error rate and voltage overscaling, we assume that a slack wall exists at which the error rate explodes [14], [3]. The relationship can then be modeled by:

$$e = min(1, ((1 - f_v)/(1 - v_o))^w) \tag{9}$$

where e is the error rate; f_v is the voltage overscaling factor ($0 \leq f_v \leq 1$, $f_v = 1$ corresponds to the nominal voltage); v_o is the normalized voltage at which the slack wall is reached ($0 \leq v_o \leq 1$), and w is the exponential relating how steeply the errors increase on overscaling. A small w value corresponds to a relatively smooth increase in error rate as voltage is reduced.

Note that v_o depends on the length of the pipeline. This is because the amount of available voltage slack decreases as the length of the pipeline is increased. Figure 1 illustrates this effect. We model the dependence of v_o on the length of the pipeline using the following equation:

$$v_o = 1 - (1 - v_{ob}) * (p_b/p)^k \tag{10}$$

where v_{ob} is the normalized slack wall voltage for the base pipeline, p_b is the base pipeline depth (we assume the traditional 5 stage pipeline as the baseline in our experiments), k controls how quickly the error rate grows with the number of pipestages, and p is the current pipeline depth. Effectively, as the pipeline depth gets deeper than the base pipeline depth, the amount of available voltage slack decreases proportionally. Note that the equation assumes that all timing paths can be equally divided when pipelining (all previous works on optimal pipelining depth make the same assumption).

IV. Methodology

Our analytical model requires data on dynamic and static power per latch (note that we make the assumption that all power is consumed in latches, the same assumption made in all previous work on optimal pipelining). Because we do not have actual gate-level data available to use as parameters in our model, we rely on data from an architecture-level power simulator (Wattch [15]) that is coupled with a cycle-accurate processor simulator (SMTSIM [16]) simulating an Alpha core. The dynamic power estimates are derived as an average over 8 randomly-chosen SPEC2000 benchmarks [17], listed in Table I when run for 100 million instructions after fast-forwarding them to the Early Simpoints [18]. We assume that leakage power is 30% of the total power at the nominal voltage. We do not consider clock gating, and we assume $\eta = 1.3$, based on [12]. Our power formula, therefore, is the following

$$P_T = (f_s(P_{sim}/f_{sim}) {f_v}^2 + (.3P_{sim}/.7) f_v) p^{1.3} \tag{11}$$

where P_{sim} is the dynamic power reported by the simulator at the nominal voltage and f_{sim} is the frequency at which that power was reported.

For validating our analytical model and confirming the conclusions we drew from the analytical model, we performed further experiments using a modified version of SMTSIM [16] coupled with power estimates from Wattch [15]. Our modifications allowed us to vary the frequency, operating voltage (V_{dd}), insert errors at a particular rate per cycle, and control the error recovery penalty. To model error recovery, we simply penalize the system for $\gamma_e \times p$ cycles (or $\gamma_e \times c$ cycles when the recovery penalty is fixed). To change the length of the simulated pipeline, we added extra stages to the front end of the simulated processor. This ensures that the increased length of the pipeline affects the overhead of hazards. In addition, Wattch does not account for power growth due to pipeline depth. We assumed the same latch growth exponent of $\eta = 1.3$ as in our analytical model, and scaled our power accordingly. Our validation experiments were run using the same SPEC2000 binaries in Table I. We fast-forwarded to the Early SimPoint [18] of each benchmark before beginning error injection simulations.

Table I describes the benchmarks we used in our simulations. The benchmarks were chosen randomly, with five floating point and three integer benchmarks. The Base IPC is the IPC of the benchmark when simulated on the minimal 8 stage pipeline supported by the simulator without considering errors (no timing speculation). Table II presents our SMTSIM settings, while Lastly, Table III presents our power settings for Wattch.

V. Results and Analysis

In this section, we analyze the relationship between the benefits from error resilience and pipeline, circuit, and

TABLE I

SPEC2000 Benchmarks Employed

Benchmark	Description	Base IPC
SPECFP		
applu	Parabolic / Elliptic Partial Differential Equations	0.307
art	Image Recognition / Neural Networks	0.44
equake	Seismic Wave Propagation Simulation	0.331
swim	Shallow Water Modeling	0.302
wupwise	Physics/Quantum Chromodynamics	0.649
SPECINT		
bzip	Compression	0.837
crafty	Game Playing: Chess	0.719
vpr	FPGA Circuit Placement and Routing	0.293

TABLE II

SMTSIM Parameters

Core	
Number of instructions simulated	100 Million
Instruction order	in-order
Number of threads	Single Threaded
Number of stages	8+
L1 Split I/D Cache	
Size	32KB
Assoc	4-Way
Miss Penalty	8 cycles
L2 Cache	
Size	2MB
Assoc	4-way
Miss penalty	40 cycles
L3 cache	
Size	4MB
Assoc	4-way
Miss penalty (to memory)	255 ps

workload characteristics. We also present results from our validation experiments.

A. Exploring the Interaction between Error Resilience, Pipelining, Circuit Structure, and the Metric for Energy Efficiency

We begin by exploring the benefits from error resilience when voltage is overscaled to allow errors which are then assumed to be tolerated using suitable error tolerance mechanisms (the recovery penalty is considered while evaluating energy efficiency). Figure 2 illustrates the benefits of error resilience for pipelines of different lengths and for different error rates. The figures also illustrate the sensitivity to the voltage vs error rate relationship. From top to bottom, the figures correspond to a more gradual voltage vs error rate relationship (the voltage can be reduced further before reaching the same error rate).

Figure 2 confirms the conclusion from the previous studies that there can indeed be significant error efficiency benefits from introducing error resilience into a design. We observe up to 30% benefits relative to a processor that is not allowed to produce errors ($e = 0$).

TABLE III

Wattch Parameters

Wattch Parameter	**Value**
Process Technology	65nm
Vdd (nominal)	1.5V
Vth	.7V
Dynamic Power vs Voltage relationship	$v^2 f$

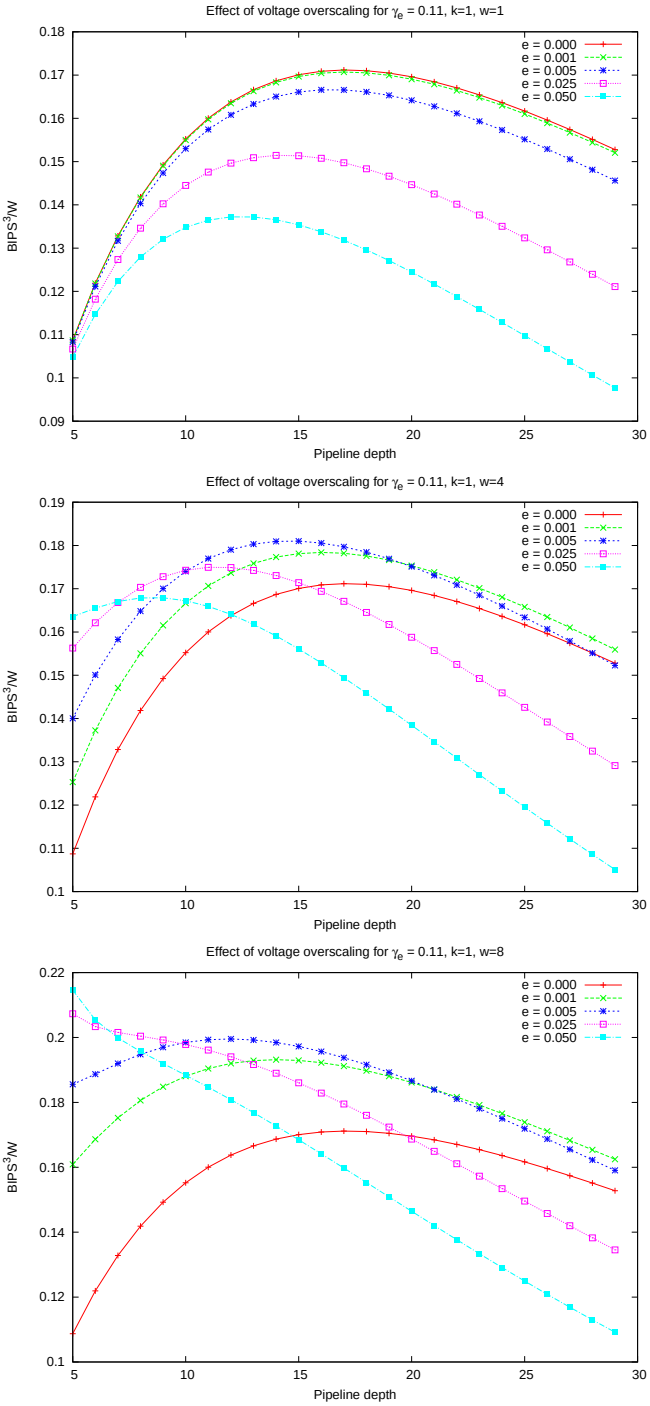

Fig. 2. Error resiliency benefits can be substantial, and are closely tied to both the length of the pipeline and the relationship between error rate and voltage scaling. The figures show error resiliency benefit for different error rate vs voltage scaling relationships. From top to bottom, w=1,4,8.

We also observe that the benefits of error resilience are strongly dependent on the relationship between voltage and error rate. When the voltage vs error rate relationship is steep, the benefits diminish as the error recovery time starts outweighing the power benefits of voltage overscaling. Note that the voltage vs error rate relationship is largely dictated by the timing slack distribution of the design, which in turn, is affected by microarchitectural choices as well as the design

methodology.

Figure 2 also demonstrates that the benefits of error resilience are strongly tied to the number of pipestages. The figure shows that the optimal length of the pipeline (i.e., the one that maximizes energy efficiency) when errors are allowed is shorter than the optimal length of the pipeline when no errors are allowed. This relates to two aspects of error resilience: the time spent recovering from errors, and the relationship between path slack and the number of pipestages. For error recovery mechanisms in which recovery time is proportional to the length of the pipeline, shorter pipelines see reduced recovery time than longer pipelines for the same error rate. Similarly, for architectures whose available path slack is strongly dependent on the length of the pipeline, as modeled by Equation 10, shorter pipelines allow greater voltage overscaling before hitting the slack wall.

To further confirm the dependence of error resiliency benefits on the slack distribution and the number of pipestages, we studied the impact on energy efficiency benefits of pushing the slack wall closer to the nominal voltage at different rates when the number of pipestages is increased. Figure 3 shows the results. The topmost figure, $k = 0$, represents an architecture in which path slack is independent of the number of pipestages. As the length of the pipeline is increased, the performance improves proportionally with the frequency change, increasing the energy efficiency until the point where the hazard and error recovery time, in addition to the power increase from latch growth, outweigh the performance improvement. For architectures in which path slacks are tightly coupled with the length of the pipeline ($k = 1$ or $k = 2$) the slack wall is hit sooner as the length of the pipeline increases, decreasing the energy efficiency benefits.

Finally, we observed the benefits from error resiliency for other energy efficiency metrics. As expected, the greatest error resiliency benefits are seen for the energy efficiency metrics dominated by power (lower values of m). The $m = 1$ curve sees the greatest error resiliency benefit and has the shortest optimal pipeline (pipelining only improves the performance portion of the metric, not the power). For performance-dominated energy efficiency metrics, the optimal pipelines are long, and therefore, the power benefits from voltage overscaling are outweighed by the error recovery overheads. Long pipelines also have reduced path slack, further reducing the benefits of error resilience. Figure 4 demonstrates the benefits of error resiliency for the BIPS^m/W metric as m is varied.

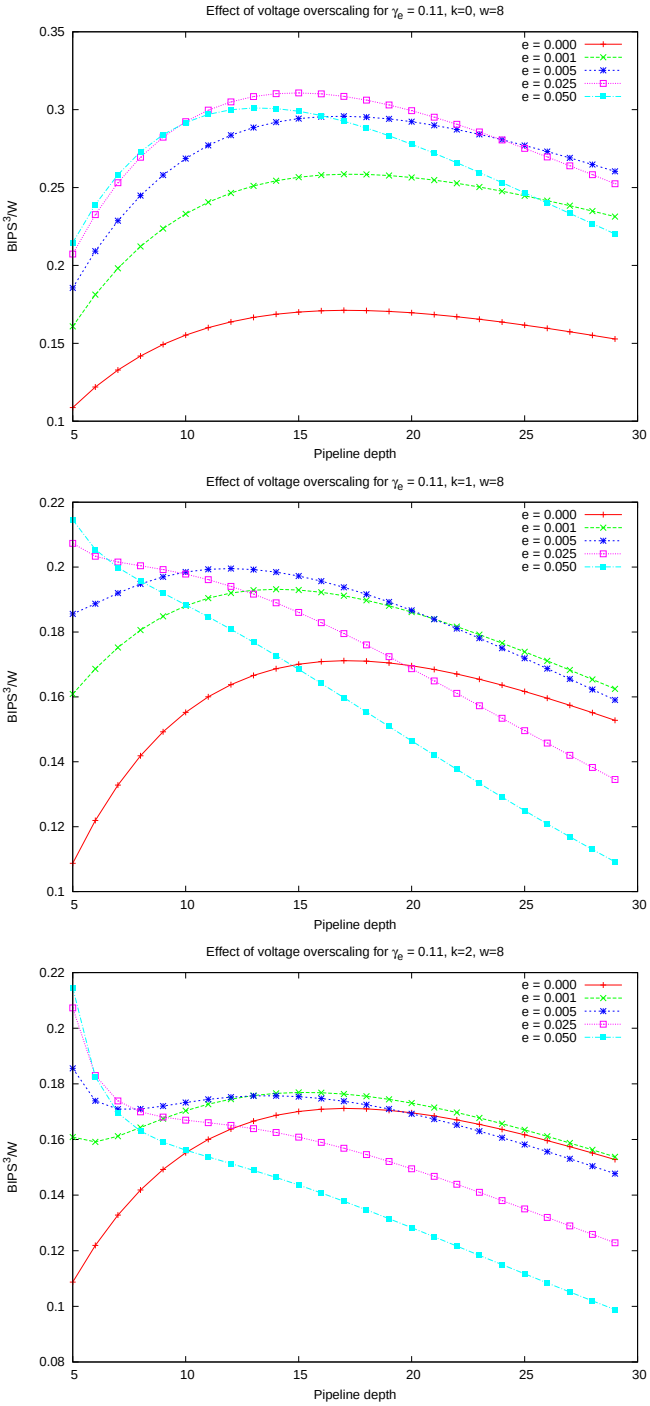

Fig. 3. Error resilient designs see greater benefits from shorter pipelines as the available path slack decreases faster due to pipelining. From top to bottom, k=0,1,2

B. Exploring the Benefits of Co-optimization

The previous results show the sensitivity of energy efficiency of error resilient designs to various architectural, circuit, and modeling parameters. We now consider the following question: how important is it to reconsider the architecture when introducing an error resilience mechanism into a design?

Figures 5 and 6 compare the benefits of error resiliency for an architecture that was optimized without error resiliency in mind against an architecture designed with error resiliency in mind. These figures illustrate the energy efficiency gains that can be had from co-optimizing the architecture with error resiliency. Note that co-optimization, in this case, simply corresponds to identifying the optimal pipeline depth and the corresponding operating voltage for a *given* error resilience mechanism.

For small error recovery penalties, where the largest gains from error resiliency are achieved, we observe significant benefits from re-architecting the processor with error re-

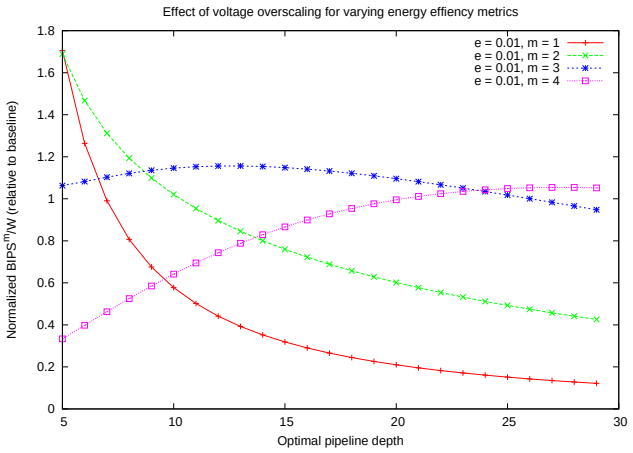

Fig. 4. Benefits of error resiliency improve for energy efficiency metrics dominated by power. The figure shows the benefit of error resiliency for m=1,2,3,4 (BIPS^m/W).

siliency in mind. In fact, we observe gains greater than 15%. The gains from co-optimization diminish as the optimal error rate decreases, which has the effect of moving the optimal pipeline lengths closer to that of the baseline (i.e., the optimal pipeline when no errors are allowed).

We also observe that the benefits of co-optimization are strongly dependent on the relationship between error rate and voltage. From top to bottom, Figure 5 shows decreasing steepness of the voltage vs error rate curve. The lower the voltage before hitting a certain error rate, the higher the optimal error rate, and therefore the greater the benefits from co-optimization.

The benefits of co-optimization are also closely linked to the sensitivity of path slack to pipeline length. Figure 6 illustrates the advantages of co-optimization as the path slack moves from being independent of pipeline length to decreasing rapidly as the length of the pipeline increases ($k = 0$ to $k = 2$). The increased benefit can be attributed to the path slack's sensitivity to the pipeline length causing the optimal architectures to have shallower pipelines. In general, the greater the reduction in the optimal pipeline length when error resiliency is considered, the greater the benefit from co-optimization.

C. Validation

We used the cycle accurate simulation-based methodology described in Section IV to validate our analytical model from Section III. Our validation experiments were performed using 8 randomly selected SPEC2000 benchmarks from both the integer and floating point suites. Here, we focus on two benchmarks that illustrate the accuracy of our results and show how optimizing for two different workloads affects error resiliency benefits. These results assume the following parameters: $\gamma_e = 0.11$, $k = 1$, and $w = 8$.

Figure 7 shows the error resiliency benefits for the SWIM and CRAFTY benchmarks.

The results confirm that significant energy efficiency benefits are indeed possible from error resiliency. SWIM sees up to 171% improvement in energy efficiency, while CRAFTY sees up to 80% gain. Also, we observe that error resiliency

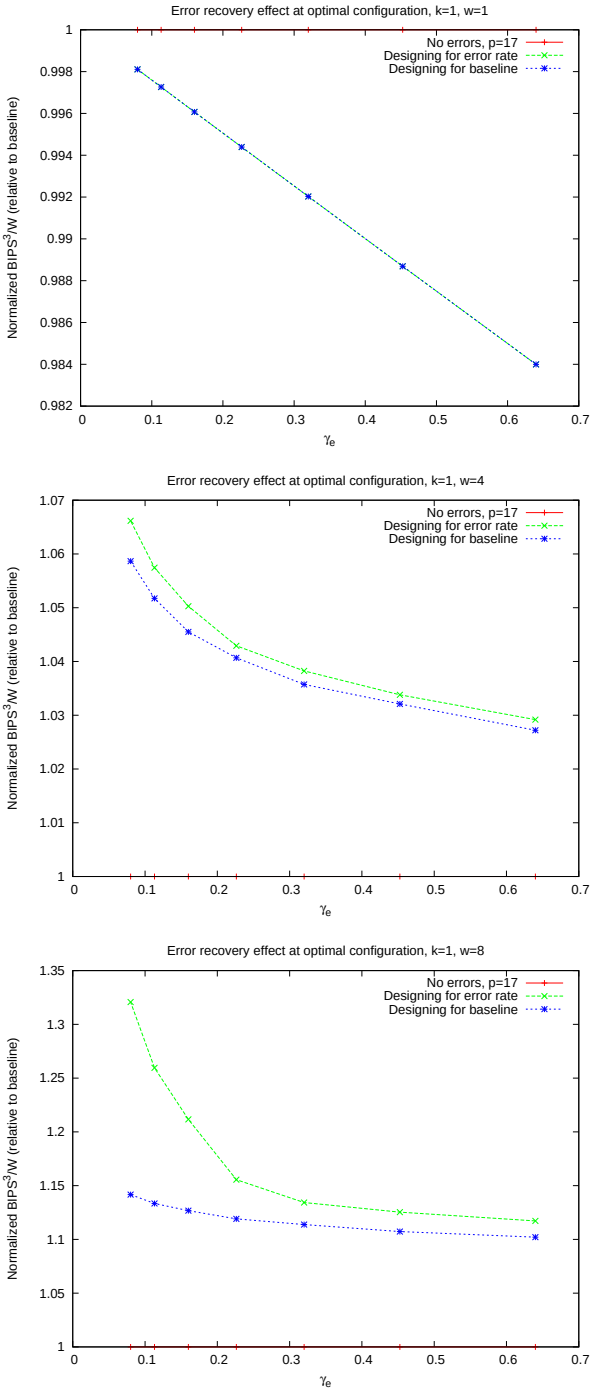

Fig. 5. The benefit of re-optimizing an architecture depends strongly on the error rate vs voltage relationship. Architectures consisting of circuits seeing fewer errors at a particular voltage (bottommost figure) will see the most benefits from re-optimization.

benefits have a strong dependence on the pipeline length for CRAFTY. The error resiliency benefits are maximized when the pipeline has 8 stages, the minimum number of stages supported by the simulator. This is significantly different from the optimal pipeline length of 14 when no errors are allowed.

The SWIM benchmark is significantly more memory sensitive, and therefore has a shorter optimal pipeline than

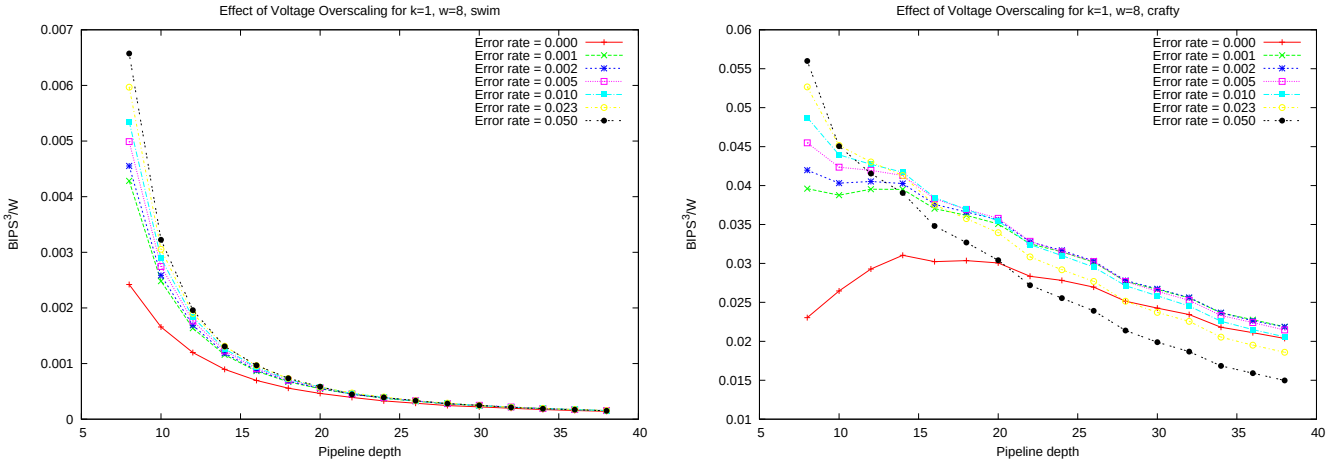

Fig. 7. Simulated results demonstrating the benefits of error resiliency for two benchmarks, SWIM (left) and CRAFTY (right).

CRAFTY. In fact, the optimal pipeline depth is the minimum of 8 even when no errors are allowed. We observe that the benefits from error resilience are indeed higher for SWIM than CRAFTY (171% versus 80%), despite the fact that all other architectural parameters are the same. This confirms our previous conclusion that error resiliency benefits increase when the optimal architecture is a shorter pipeline, as is the case when designing for irregular workloads.

Lastly, the CRAFTY results illustrate the need for co-optimization. As can be seen, the energy efficiency gains from error resilience are only 34% over the baseline when operating at the optimal non-error resilient pipeline depth. If the architect were to co-optimize the architecture with the error resilience mechanism, therefore reconsidering the pipeline depth, the energy efficiency could be as high as 80% over the baseline. Note that in both these results, the optimal pipeline depth is the minimal one. This is not always the case, and depends on the systems, particularly the error recovery penalty (γ_e), sensitivity of slack to pipeline depth (k). Due to space restrictions we only present one case. Furthermore, due to the limitations of our simulator, we were not able to evaluate systems with less than 8 pipeline stages. Our future work will involve a more adaptable simulation framework.

Figure 8 summarizes the results for all 8 SPEC benchmarks investigated and compares benefits to the pipeline depth. On average, we see a 136% energy efficiency gain from error resiliency, 25% of which is due to co-optimizing the pipeline depth and error resiliency mechanism. In addition, we confirm that those systems designed for the shortest pipeline depths (those points highest on the pipeline depth scale), see the largest benefits from voltage overscaling-based error resiliency. These are the benchmarks that correspond to the lowest base IPC from Table I, and are those that are most irregular in nature, confirming that systems designed for irregular workloads see larger timing speculation-based energy efficiency gains.

VI. Conclusions

Previous work on timing speculation has either explored the benefits of customizing the design methodology for a particular error resilience mechanism [2], [3], [4], [5] or has attempted to understand the benefits from error resilience for a particular processor design [1], [6], [7]. There is no work, to the best of our knowledge, that attempts to understand the benefits of co-optimizing microarchitecture and error resilience.

In this paper, we presented the first study on co-optimizing a processor pipeline and an error resilience mechanism. We developed an analytical model that relates the benefits from error resiliency to the depth of the pipeline as well as its circuit structure. The model was used to determine the optimal pipeline depth for different energy efficiency metrics for different error resilience overheads.

Our results demonstrated that several interesting relationships exist between error resilience and pipeline structure. For example, we showed that there are significant energy efficiency benefits to pipelining an architecture for an error resiliency mechanism vs error resiliency-agnostic pipelining. As another example, we show that benefits from error resiliency are greater for short pipelines than long pipelines. We also confirmed that the benefits from error resiliency are higher when the circuit structure is such that error rate increases slowly on reducing input voltage vs a circuit optimized for power where a slack wall exists at the nominal operating point [14], [3]. Finally, we quantified the difference in benefits from error resiliency for irregular vs regular workloads and showed that benefits from error resiliency are higher for irregular workloads.

Our study demonstrates considerable promise for an approach to processor architecture that considers the error resilience mechanism.

Acknowledgements

The authors would like to thank our anonymous referees for the valuable feedback. This work was supported in part by grants from Intel, NSF, GSRC, SRC, and an Arnold O. Beckman Research Award.

References

[1] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, “Razor: A low-power

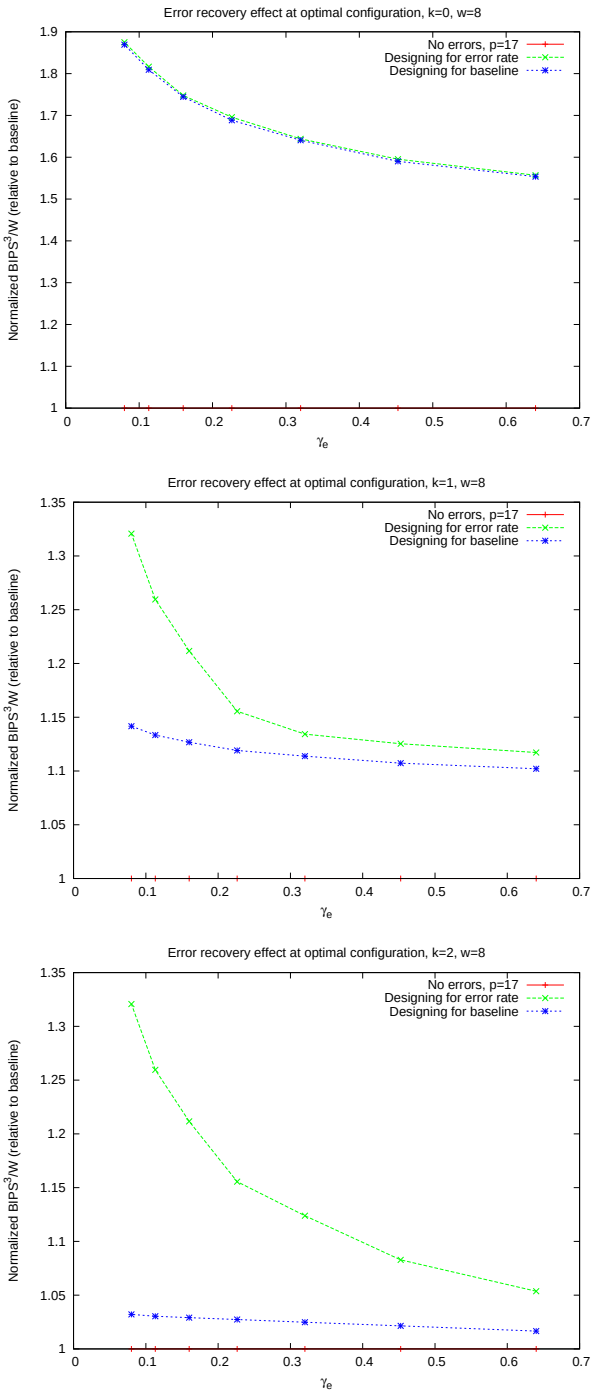

Fig. 6. Architectures with path slacks strongly sensitive to pipeline depths (bottommost figure, k=2) see the most benefit from pipeline co-optimization with error resiliency. As path slack sensitivity increases, the optimal pipeline depth's deviation from the baseline's optimal pipeline depth increases, resulting in a a greater need for co-optimization.

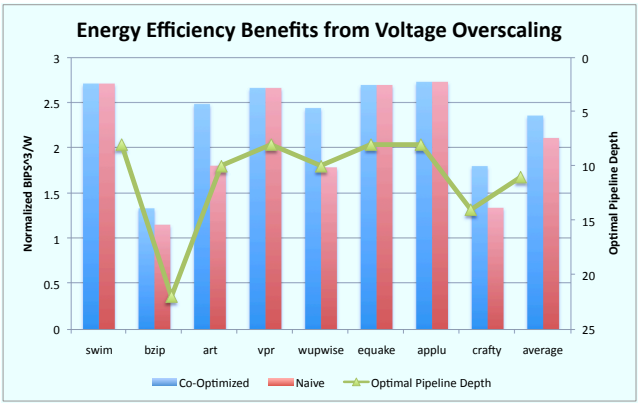

Fig. 8. Simulated results illustrating energy-delay^2 when operating at multiple error rates and pipeline depths. Results are normalized to the energy-efficiency of the optimal non-error resilient design.

pipeline based on circuit-level timing speculation," in *MICRO 36: Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, 2003, p. 7.

[2] B. Greskamp, L. Wan, W. R. Karpuzcu, J. J. Cook, J. Torrellas, D. Chen, and C. Zilles, "Blueshift: Designing processors for timing speculation from the ground up," in *International Symposium on High Performance Computer Architecture*, 2009, pp. 213–224.

[3] A. B. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2010, pp. 825–831.

[4] ——, "Designing a processor from the ground up to allow voltage/reliability tradeoffs," in *International Symposium on High Performance Computer Architecture*, 2010, pp. 1–11.

[5] ——, "Recovery-driven design: A methodology for power minimization for error tolerant processor modules," in *DAC '10: Proceedings of the 47th Annual Design Automation Conference*, 2010, pp. 825–830.

[6] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *ISLPED '99: Proceedings of the 1999 International Symposium on Low Power Electronics and Design*, 1999, pp. 30–35.

[7] J. C. Smolens, B. T. Gold, B. Falsafi, and J. C. Hoe, "Reunion: Complexity-effective multicore redundancy," in *MICRO 39: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, 2006, pp. 223–234.

[8] A. Hartstein and T. R. Puzak, "Optimum power/performance pipeline depth," in *MICRO 36: Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, 2003, p. 117.

[9] S. Dhar, D. Maksimović, and B. Kranzen, "Closed-loop adaptive voltage scaling controller for standard-cell asics," in *Proceedings of the 2002 International Symposium on Low Power Electronics and Design*, 2002, p. 107.

[10] T. Kehl, "Hardware self-tuning and circuit performance monitoring," in *IEEE International Conference on Computer Design*, 1993, pp. 188–192.

[11] M. S. Hrishikesh, D. Burger, N. P. Jouppi, S. W. Keckler, K. I. Farkas, and P. Shivakumar, "The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays," in *ISCA '02: Proceedings of the 29th Annual International Symposium on Computer Architecture*, 2002, pp. 14–24.

[12] V. Srinivasan, D. Brooks, M. Gschwind, P. Bose, V. Zyuban, P. N. Strenski, and P. G. Emma, "Optimizing pipelines for power and performance," in *Proceedings of the 35th International Symposium on Microarchitecture*, 2002, pp. 333–344.

[13] M. de Kruijf, S. Nomura, and K. Sankaralingam, "A unified model for timing speculation: Evaluating the impact of technology scaling, cmos design style, and fault recovery mechanism," 2010.

[14] J. Patel, "Cmos process variations: A critical operation point hypothesis," Online Presentation, 2008. [Online]. Available: http://www.stanford.edu/class/ee380/Abstracts/080402-jhpatel.pdf

[15] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *ISCA '00: Proceedings of the 27th annual international symposium on Computer architecture*, 2000, pp. 83–94.

[16] D. Tullsen, "The SMTSIM multithreading simulator," 2010. [Online]. Available: http://cseweb.ucsd.edu/users/tullsen/smtsim.html

[17] "SPEC CPU2000," 2000. [Online]. Available: http://www.spec.org/cpu2000/

[18] E. Perelman, G. Hamerly, M. Van Biesbrouck, T. Sherwood, and B. Calder, "Using simpoint for accurate and efficient simulation," in *SIGMETRICS '03: Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2003, pp. 318–319.