Adaptive Design for Nanometer Technology

Shidhartha Das[1] and David Blaauw[2]
[1]ARM Ltd., Cambridge, U.K
[2]Department of EECS, University of Michigan, Ann Arbor, U.S.A

Abstract—**Rising design uncertainties at advanced process nodes place conflicting demands on todays engineers. The widening safety-margins required to ensure robust designs in the face of such uncertainties lead to conservative designs with unacceptable power and performance overheads. On the other hand, low-power techniques such as Dynamic Voltage Scaling (DVS) and clock-gating adversely affect circuit robustness. In this paper, we will review a number of techniques for adaptive design which mitigate the impact of margins by tuning system parameters (voltage and frequency) according to variations in runtime workload, environmental and process conditions. We will evaluate the margins that each of the different approaches require to guarantee correctness in the worst-case and typical operating point. We also propose a technique called Razor which is an aggressive method for eliminating all safety-margins through in situ error-detection and correction. Error-detection is achieved by a new type of Razor flip-flop that monitors critical path endpoints and flags timing errors upon detecting spurious transitions. Recovery is achieved through replay from a check-pointed state. We show the design of the transition-detecting Razor flip-flop and illustrate how it naturally detects SEU in combinational logic and inside latches. We present a 64bit processor implementing Razor and show, on an average, 33% energy savings from our measurements on silicon.**

Index Terms— **Adaptive circuits, Dynamic Voltage Scaling, Design Margins, Soft Error Rate, Self-tuning**

I. Introduction

In the last few years, the computational capability of mobile and hand-held devices has witnessed phenomenal improvements. Desktop applications such as 3-D graphics, audio/video, internet access and gaming are now available for mobile platforms as well. A key technique that has led to such performance improvements has been technology scaling at the rate dictated by the Moore's Law. By shrinking transistor dimensions, designers can deliver consistent improvements in computational capability of processors through higher integration levels and faster switching times. However, starting with the 65nm node, higher transistor integration levels, combined with almost constant supply voltages and stagnation of energy efficiency, has caused power consumption of processors to actually worsen at aggressive process nodes. This has created a design paradox: more transistors can now be fitted on a die, however, they cannot be used due to strict power limits.

Power consumption is especially relevant for battery-operated mobile processors as they increasingly handle computationally demanding applications under stringent power budgets. Rising process, voltage and temperature (PVT) variations further exacerbate power consumption issues at aggressive geometries. Designing robust circuits that can cope with such variations requires operation at a higher supply voltage. This ensures that any unforeseen slow-down because of voltage glitches, high temperature conditions and process variations does not cause computing errors due to processor timing violations.

Voltage margining leads to robust circuit operation in presence of variations, although at the expense of higher power consumption. As variations worsen even wider margins are required. However, safety margins are not needed for all chips or for the entire duration of their operational lifetime. Only a small percentage of the manufactured chips are inherently slow. Even for these slow chips, it is highly unlikely that they will exhibit worst-case temperature and voltage conditions for significant periods during their operation. Thus, the fundamental issue with margining is that it seeks to budget for worst-case conditions that occur extremely rarely in practice. This adversely impacts the power budgets of processors that are already stressed due to performance demands.

The increasingly large energy wasted due to margins has lead to significant interest in a new approach to chip design called adaptive design. The key idea of this approach is to tune the voltage and frequency specific to the native speed of each individual chip and its dynamic computational requirements. Thus, if the transistors are inherently faster, then the chip detects this and lowers or scales the voltage of operation. Of course, voltage scaling needs to be within safe limits; otherwise, the consequent slow-down of the processor can result in timing failures.

The traditional approach of voltage scaling using adaptive design techniques is called the "always-correct" approach. This approach seeks to predict the failure limit of a chip and to tune system parameters to operate near this predicted point. Typically, safety margins are added to the predicted failure point to guarantee computational correctness. The second approach called "let fail and correct" approach allows a chip to incur timing errors and then recover to achieve correct operation. In the next two sections, we survey these approaches in detail. Section 4 deals with a specific "let fail and correct" technique called Razor. We show the design of a delay-error tolerant Razor flip-flop and demonstrate SER tolerance with it. Finally, we give concluding remarks in Section 5.

II. "Always Correct" Approaches

The key idea in the "always correct" techniques is to predict the operational point where the critical-path fails to meet timing and to guarantee correctness by adding safety margins to the predicted failure point. The conventional approach of predicting this point of failure is to use either a look-up table or so-called "canary" circuits.

A. Look-up table based approach

In the look-up table based approach, the processor is pre-characterized during design-time to obtain its maximum obtainable frequency for a given supply voltage. The safe voltage-frequency pairs are obtained by performing conventional timing analysis on the processor. Typically, the operating frequency is decided based on the deadline under which a given computational task needs to be completed. Accordingly, the supply voltage corresponding to the frequency requirement is "dialed in". The table look-up approach exploits periods of low CPU utilization by dynamically scaling voltage and frequency, thereby leading to energy savings. However, its reliance on conventional timing analysis performed at the combination of worst-case process, voltage and temperature corners implies that none of the safety margins are eliminated at a particular

978-1-4244-3828-0/09/$25.00 ©2009 IEEE

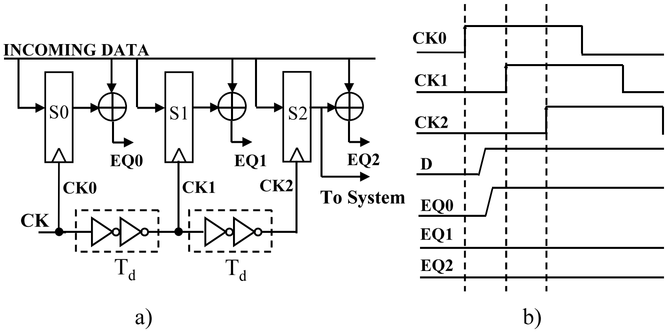


Figure 1. Kehl's Triple Latch Monitor. Figure a) shows the system configuration. Figure b) shows the timing diagrams when the system is tuned.

operating point.

B. *Canary-circuits based approach*

An alternative approach relies on the use of the so-called "canary" circuits to predict the failure point [1-4]. Canary circuits are typically implemented as delay chains which approximate the critical path of the processor. Voltage and frequency are scaled to the extent that this replica-delay path fails to meet timing. The replica-path tracks the critical-path delay across inter-die process variations and global fluctuations in supply voltage and temperature, thereby eliminating margins due to global PVT variations. However, the on-die location of the critical-path and its replica differs. Consequently, margins are added to the replica-path in order to budget for delay mismatches due to intra-die process and local variations in temperature and supply voltage. Margins are also required to address fast-changing transient effects, such as coupling noise, which are difficult to respond to in time using this approach. Furthermore, mismatches in the scaling characteristics of both paths require additional safety margins. These margins ensure that the processor still operates correctly even at the point of failure of the replica-path.

C. *In situ triple-latch monitor*

Kehl’s Triple-Latch Monitor [5] is similar to the canary-circuits based techniques, but utilizes *in situ* monitoring of circuit delay. Using this approach, all monitored system state is sampled at three different latches with a small delay interval between each sampling point, as shown in Figure 1(a). The value in the latest-clocked latch which is allowed the most time is assumed correct and is always forwarded to later logic. The system is considered “tuned” (Figure 1ub) when the first latch does not match the second and third latch values, meaning that the logic transition was very near the critical speed, but not dangerously close. If all latches see the same value, the system is running too slowly and frequency should be increased. If the first two latches see different values than the last, then the system is running dangerously fast and should be slowed down.

Because of the *in situ* nature of this approach, it can adjust to local variations such as intra-die process and temperature variations. However, it still cannot track fast-changing conditions such as cross-coupling and voltage noise events. In addition, to avoid overly aggressive clocking, evaluations of the latch values must be limited to tests using worst-cast latency vectors. Kehl suggests that the system should periodically stop and test worst-case vectors to determine if the system requires tuning. This requirement severely limits the general applicability of this approach since vectors that account for the worst-case delay and coupling noise scenario are difficult to generate, and exercise, for general-purpose processors.

III. “LET FAIL AND CORRECT” APPROACHES

The key concept of these schemes is to scale the system parameters (e.g. voltage and frequency) till the point where the processor fails to meet timing, thereby leading to an timing error. An error-detection block flags the occurrence of the error, upon which a recovery infrastructure is engaged to achieve correct state. Allowing the processor to fail and then recover eliminates worst-case safety margins. This enables significantly greater performance and energy efficiency over “always-correct” techniques. Furthermore, such techniques naturally exploit input-vector dependence of delay by relying on the error-rate for voltage and frequency tuning. The net energy consumption of the system is essentially a trade-off between the increased efficiency afforded by the elimination of margins and the additional overhead of recovery. Of course, the overhead of recovery can make sustaining a high error-rate counterproductive. Hence, these systems typically rely on restricting operation to low error-rate regimes to maximize energy efficiency.

Their relative complexity makes the general applicability of such systems difficult. However, they are naturally amenable for certain applications areas such as communications and signal processing. Communication systems require error correction to reliably transfer information across a noisy channel. Therefore, it is relatively easier to overload the existing error correction infrastructure to enable adaptivity to variable silicon and ambient conditions..

A. *Techniques for communication and signal processing*

Self-calibrating interconnects [6] address the problem of reliable on-chip communication in aggressively scaled technologies. Signal integrity concerns require on-chip busses to be strongly buffered which consumes a significant portion of the total chip power. Hence, it is desirable to transfer bits at the lowest possible operating voltage while still guaranteeing the required performance and the targeted bit-error-rate (BER). Worm addresses this issue by encoding the data words with so-called self synchronizing codes before transmission.

The receiver is augmented with a checker unit that decodes the received code word and flags timing errors. Correction occurs by requesting re-transmission through an Automatic Repeat Request (ARQ) block, as shown in figure 2. Furthermore, an additional controller obtains feedback from the checker and accordingly adjusts the voltage and the frequency of the transmission. By reacting to the error-rates, the controller is able to adapt to the operating conditions and thus eliminate worst-case safety margins. This improves the energy efficiency of the on-chip busses with negligible BER degradation.

Algorithmic Noise Tolerance (ANT) [7] by Shanbhag et al. uses a similar concept for low-power VLSI signal processing architectures. As conceptually illustrated in, the main processor block is augmented with an estimator block. The main block is voltage scaled beyond the point of failure, thereby leading to intermittent timing errors. The

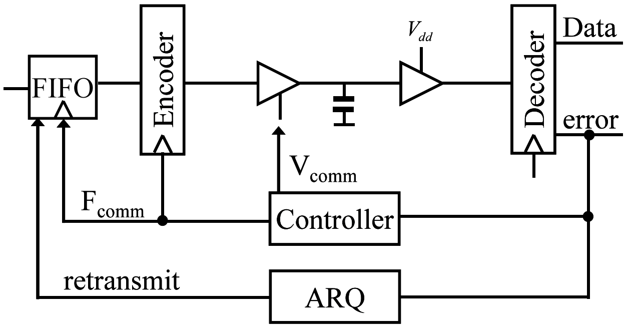


Figure 2 Self-calibrating interconnects

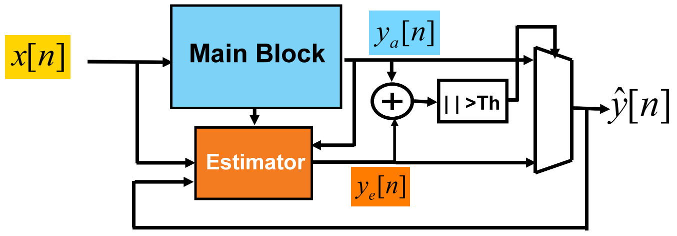


Figure 3 Algorithmic Noise Tolerance

result of the main block is validated against the result of the estimator block which computes correct result, based on the previous history.

The estimator block is significantly cheaper in terms of area and power as compared to the main block which is being voltage-scaled. At low error-rates, the benefits of aggressive scaling on the main block compensates for the overhead of correction, leading to significant energy savings. Error detection occurs when the difference in results of the main block and the estimator block exceeds a certain threshold. Error correction occurs by overwriting the result of the main block with that of the estimator block. Since the estimator block depends upon past history of correct results to make its prediction, its accuracy reduces as more errors are experienced. Hence, it is desirable to keep the rate of timing errors low for maintaining a low BER and high energy efficiency.

B. *Techniques for general-purpose computing*

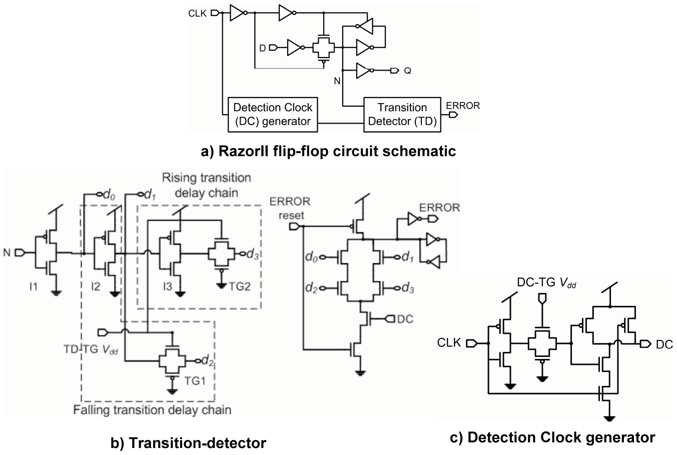
In general-purpose computing the committed architectural state necessarily has to be always correct. Therefore, all timing errors that can alter the architectural state need to be flagged and corrected. Unlike in communication and signal processing applications, corruption of the architectural state in general-purpose computing leads to system failure and needs to be avoided at all costs.

We proposed Razor [8] as the first application of a low-overhead, “let fail and correct” adaptive technique to general-purpose computing. Razor eliminates worst-case safety margin through *in situ* error detection and correction of timing errors. In this technique, we use a delay-error tolerant flip-flop on critical paths to scale the supply voltage to the point of first failure (PoFF) of a die for a given frequency. Thus, all margins due to global and local PVT variations are eliminated, resulting in significant energy savings. In addition, the supply voltage can be scaled even lower than the first failure point into the sub-critical region, deliberately tolerating a targeted error rate, thereby providing additional energy savings.

From our initial Razor experiments [8] (henceforth referred to as RazorI) implemented on a 64bit processor with 0.18 micron technology, we obtained an average energy savings of 50% over the worst-case by operating at the optimal voltage point. A key finding from our measurements is that the error rate at the PoFF is extremely low, ~1 error in 10 million cycles. However, beyond the PoFF, the error rate increases exponentially at one decade per 10mV supply voltage increase. Hence the energy gain from operating substantially below the PoFF was small (~10%) compared to the energy gain from eliminating the PVT margins (~35 to 45%). We took advantage of these findings and proposed a new technique called RazorII [9], wherein the processor is intended to operate near the PoFF and recovery from a timing error occurs by a conventional architectural replay mechanism.

IV. RazorII-based PVT and SER Tolerance

The RazorII approach that introduces two novel components: 1) RazorII performs only error-detection in the FF, while correction is performed through architectural replay. This allows significant reduction in the complexity and size of the Razor FF. Since RazorII is intended to operate near the PoFF, the increased overhead from using architectural correction has a negligible impact on the energy efficiency. 2) The RazorII FF naturally detects SER in the logic and registers without additional overhead. Hence, RazorII provides both low energy operation through dynamic supply adaptation as well as SER tolerance, as demonstrated by radiation tests.



a) RazorII flip-flop circuit schematic

b) Transition-detector

c) Detection Clock generator

Figure 4 RazorII flip-flop circuit schematic

The RazorII FF (Figure 4) uses a single latch combined with a transition detector (TD) controlled by a detection clock (DC). While the implementation uses a latch, it operates as a positive edge triggered FF. If the data transitions before the rising clock edge, the short negative pulse on DC suppresses the TD and no error is registered (Figure 5). However, if the input data transitions after the rising clock edge, during transparency, the transition of latch node N occurs when TD is enabled and results in assertion of the error signal and instruction roll-back. Hence, late arriving signals are flagged as an error which enforces FF based operation of the design. By using a latch instead of a FF, the RazorII FF has a slightly improved Clk-q delay compared to RazorI and 0ps setup time at the positive edge. It uses 47 transistors (RazorI FF used 76) if the DC is generated internally and 39 if the DC generation is shared between several FFs. The power overhead for a RazorII FF as compared to a conventional FF for a 10% activity factor is 28.5%. The total power overhead due to inserting RazorII FFs in the processor was 1.2%.

Since the latch node is monitored by the TD during both clock phases, SER strikes at the latch node or propagated from the logic to the latch are automatically detected without additional overhead

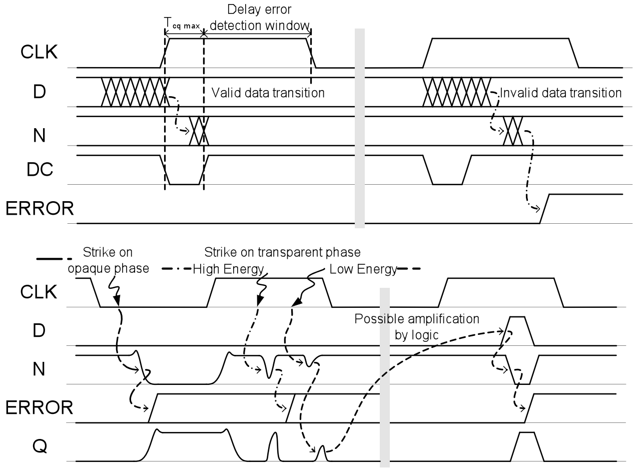


Figure 5 RazorII timing diagrams for PVT variations and SER

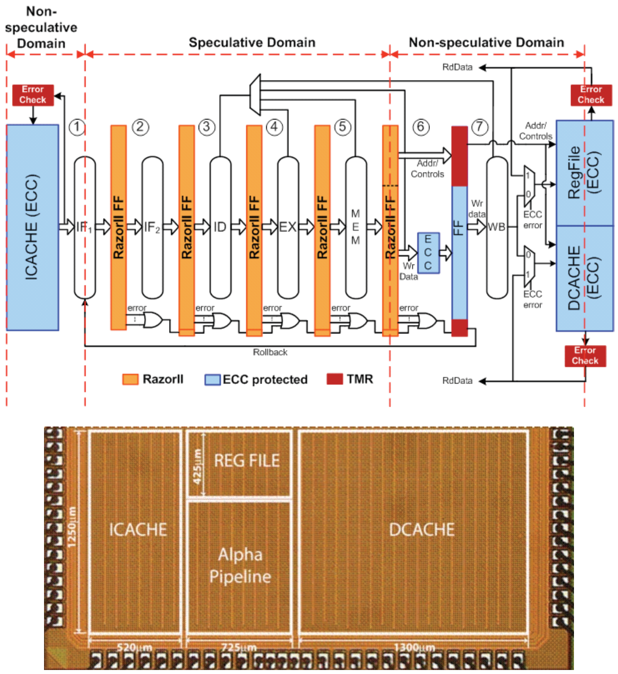


Figure 6 RazorII pipeline architecture and die-photo

(Figure 5). A strike during the low phase of DC when the TD is disabled is either benign, if the signal returns to its valid value before TD is re-enabled, or is detected as an error. Hence, the transparency window must extent beyond the low phase of DC to avoid latching of SER strikes before they are detected by the re-enabled TD.

V. RazorII Prototype Processor

RazorII was incorporated in a 64-bit, 7 stage Alpha processor in a 0.13µm technology. The architecture (Figure 6) is divided into a pipeline with speculative state protected using RazorII FFs, and a non-speculative memory and register file protected by ECC or triple-module redundancy (TMR). The error signals of all RazorII FFs in each pipeline stage are ORed together and the result is propagated and ORed with that of the next stage. The 7th stage was designed to be non-timing critical to stabilize the pipeline state. It also encodes the speculative state before it is passed to the RF or SRAM. In the event of an error, the pipeline is flushed and the failing instruction is re-executed. In case of repeatedly failing instructions, the error controller switches the clock frequency by half for 8 cycles. The error rate is kept at 0.04% using an off-chip controller. However, since failing instructions are guaranteed to complete, control can also be performed in software on the chip itself.

Figure 7 shows the measured energy dissipation for 3 die when operating at 0.04% error rate. Gains were 33.1 to 37.5% compared to the energy when the supply voltage is elevated to ensure correct operation for all 31 fabricated die at 85C with 10% margin for wearout, supply fluctuation and safety. Table 1 shows the radiation setup and the different test cases. In Test 1, the test-chip is exposed to SER with error detection disabled and as expected the final program result is incorrect. When error detection is enabled (Test 2) the processor is able to detect and correct the SER induced errors. The test-chip continues to operate correctly when the frequency of operation is increased beyond PoFF causing delay errors in addition to SER (Test 3).

VI. Conclusion

In this paper, we surveyed different adaptive techniques presented in literature and discussed the margins eliminated by each of them.

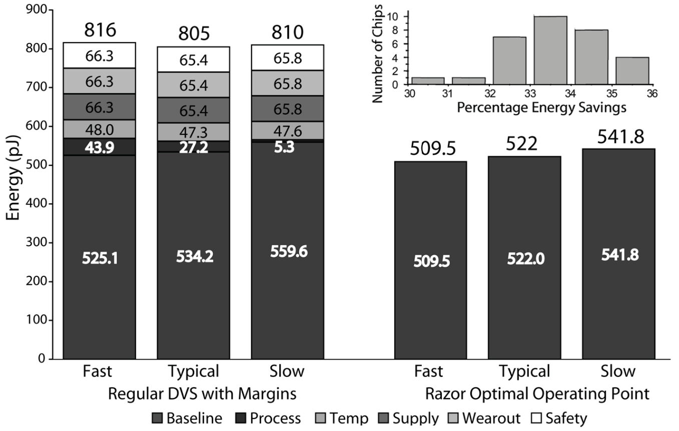


Figure 7 RazorII energy savings

We presented a "let fail and correct" approach for general-purpose computing called RazorII. We demonstrated 33% energy savings and correct functionality in the presence of SER errors from a 64bit RazorII processor.

Table 1 SER Radiation Tests

Radiation Test Description	Error Detection (TD enabled)	Vdd (V)	Razor Errors	ECC Errors	Delay error rate > 0	Correct program execution
1 – No timing errors and error detection off	Off	0.8	NA*	NA*	No	No
2 – No timing errors with error detection	On	0.8	4	5	No	Yes
	On	0.9	6	4		
	On	1.0	4	6		
3 – Both SER and timing errors with error detection	On	0.8	33M **		Yes	Yes

References

[1] A. Drake, et al., "A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor", ISSCC, 2007

[2] T. Burd, et al. "A Dynamic Voltage Scaled Microprocessor System", JSSC, Vol. 35, No. 11, 2000

[3] M. Nakai, et al., "Dynamic Voltage and Frequency Management for a Low-Power Embedded Microprocessor", JSSC, Vol. 40, No. 1, 2005

[4] K. Nowka, et al., "A 32-bit PowerPC System-on-a-chip With Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling", JSSC, Vol. 37, No. 11, 2002

[5] T. Kehl, "Hardware Self-Tuning and Circuit Performance Monitoring," *1993 Int'l Conference on Computer Design (ICCD-93)*, October 1993.

[6] Worm et al., "A Robust Self-Calibrating Transmission Scheme for On-Chip Networks", *TVLSI*, Vol. 13, No. 1, January 2005.

[7] R. Hegde and N. R. Shanbhag, "A Voltage Overscaled Low-Power Digital Filter IC", *JSSC,* Vol.39, No. 2, February 2004.

[8] S. Das, et al., "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction", JSSC, Vol. 41, No. 4, 2006

[9] Blaauw et al. "RazorII: In situ error-detection and correction for PVT and SER tolerance", ISSCC 2008