Variation-Mitigation for Reliable, Dependable and Energy-Efficient Future System Design

Shidhartha Das

Abstract Integrated circuits in modern SoCs and microprocessors are typically operated with sufficient timing margins to mitigate the impact of rising process, voltage and temperature (PVT) variations at advanced process nodes. The widening margins required for ensuring robust computation inevitably leads to conservative designs with unacceptable energy-efficiency overheads. Reconciling the conflicting objectives imposed by variation-mitigation and energy-efficient computing will require fundamental departures from conventional circuit and system-design practices. We begin by reviewing how energy-efficiency constrains computing across the entire spectrum, from ultra-low-power sensor node systems to high-performance supercomputing systems delivering peta-flop order performance. We discuss how rising variations adversely impact energy-efficient system design in the traditional method of designing for the worst case. We classify various sources of variation and discuss the traditional approaches for variation-mitigation and their limitations. The latter half of the chapter deals with several promising techniques for variation-mitigation. We discuss in situ ageing monitors, error-resilient techniques and adaptive-clocking techniques that aim at improving system-efficiency by actively reducing design guardbands. In particular, we focus on error-resilient techniques that exploit tolerance to timing errors to automatically compensate for variations and dynamically tune a system to its most efficient operating point. We present the Razor approach as a pioneering example of such a technique. We present silicon measurement results from multiple industrial and academic demonstration systems that employ Razor dynamic voltage and frequency management. Finally, we conclude the chapter with few pointers on alternative techniques for variability-mitigation.

S. Das (✉)
ARM Research, Cambridge, UK
e-mail: shidhartha.das@arm.com

© Springer International Publishing AG 2018

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_7

1 Introduction

It is a well-known observation that traditional feature-size scaling is increasingly running into fundamental physical limits. Technology innovations such as FinFETs and 3D stacking continue to deliver increased transistor densities. However, rising PVT variations combined with limited supply-voltage scaling significantly undermine automatic energy-efficiency gains traditionally obtained through process scaling. This has created a design paradox often referred to as "Dark Silicon" [1]: more gates can now fit on a die, but cannot actually be used due to strict power limits. Indeed, energy-efficiency is a first-class design constraint across the entire spectrum of computing.

Figure 1 presents a conceptual representation of the computing spectrum and highlights the importance of energy-efficiency in current-generation computing systems. Efficiency constraints are intuitively simpler to visualize for ultra-low-power computing systems at the lowest end of the power-performance spectrum. Such systems typically find usage in applications such as continuous health and infrastructure monitoring where form-factor constraints restrict the capacity of the battery used to power such systems. Operating under heavily energy-constrained environments often requires resorting to energy-harvesting techniques to make up for the energy shortfall. Mobile computers such as smartphones and laptop computers are also energy-constrained systems since they operate under a fixed power budget and typically have to provide several hours of compute-time for every charge cycle. Such systems also operate under strict thermal constraints, which limit the maximum power dissipation of such systems. At the very high-end of the spectrum are enterprise server systems and supercomputing systems that are wall-powered. Such systems are essentially power-constrained system since the

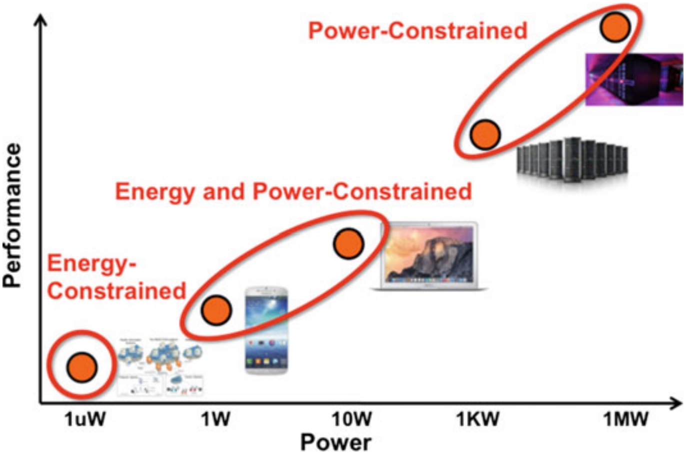


Fig. 1 Conceptual representation of the power-performance spectrum of computing—highlights the importance of energy-efficiency as a first-class design constraint

peak power consumption on these devices limits their Total Cost of Ownership (TCO). Thus, computing is essentially either power- or energy-constrained across the entire spectrum.

Moore's law-based dimensional scaling traditionally provided the necessary efficiency demand. However, sustained process scaling is now no longer economically feasible due to fundamental physical barriers. A direct consequence of smaller geometries and higher integration levels is that the manufacturing process is poorly controlled, leading to large variations in transistor performance. Susceptibility to single-event upsets and ageing-induced reliability issues that are increasingly pronounced at smaller geometries further exacerbate transistor variability. This makes systems susceptible to timing-failures due to gradual slow-down in transistor switching speeds, eventually leading to permanent functional failure.

The traditional approach of robust and reliable computing in the presence of variation relies upon operation at higher supply voltage and/or lower operating frequency. Addition of generous guardbands incurs significant power and performance overheads. Furthermore, operation at higher supply voltages leads to accelerated ageing, thereby impacting long-term system reliability.

Recently, error-resilient techniques have been proposed that mitigate the power, performance and reliability impact of excessive design margining. In lieu of margins, such techniques rely upon error-detection and correction techniques to reduce or eliminate voltage guardbands, leading to energy-efficient operation. Razor [2–5] is a specific example of such a technique where error-detecting circuitry at critical-path endpoints flag timing violations. Error-correction is achieved either through correct data substitution or through instruction-replay from a check-pointed state. In situ error-detection circuits and microarchitectural recovery mechanisms eliminate these margins and scale the supply voltage to the Point of First Failure (PoFF) and below. Error-detection and recovery enables Razor systems to survive both fast-moving and transient events, and adapt to the slow-changing prevailing conditions, allowing excess margins to be reclaimed. The reclaimed margins can be traded-off for per-device improvements in energy-efficiency or parametric yield improvement for a batch of devices.

Razor-enabled dynamic adaptation has demonstrated substantial improvements in performance and energy-efficiency in microprocessor pipelines. In [4], we demonstrate 52% energy savings at 1 GHz operation for a Razor-based ARM ISA processor. Related work in [5] shows 32% throughput improvements at iso-voltage and 17% voltage reduction at iso-frequency operation.

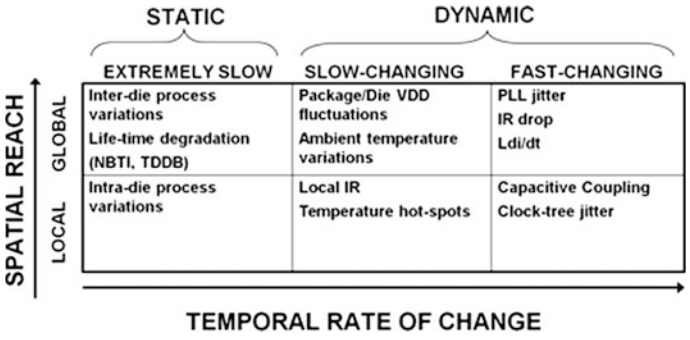
In this chapter, we review how variation-mitigation can be an effective tool for energy-efficient computing. In the following section, we classify the various sources of on-chip variation into their time rate of change and according to their spatial reach. In Sect. 3, we examine tracking circuits as a technique for compensating slow-changing variations. Section 4 examines various flavours of error-resilient computing. Section 5 discusses adaptive-clocking techniques. Finally, we end the chapter in Sect. 6 where we provide concluding remarks.

2 Classification of Variations

Figure 2 classifies the various sources of variations according to their spatial reach and temporal rate-of-change. Based on their spatial reach, variations can be *global* or *local* in extent. Global variations affect all transistors on die such as inter-die process variations and ambient temperature fluctuations. In contrast, local variations affect transistors that are in the immediate vicinity of one another. Examples of local variations are intra-die process variations, local resistive (IR) drops in the power-grid and localized temperature hot spots.

Based on their rate-of-change with time, variations can be classified as being *static* or *dynamic*. Static variations are essentially fixed after fabrication such as process variations, or manifest extremely slowly over processor lifetime such as ageing effects. Dynamic variations affect processor performance at runtime. Slow-changing variations such as temperature hot spots and board-parasitic induced regulator ripple have kilohertz time constants. Fast-changing variations such as inductive undershoots in the supply voltage can develop over a few processor cycles. The rate and the duration of these Ldi/dt droops is a function of package inductance and the on-chip decoupling capacitance. Coupling noise and Phase-Locked Loop (PLL) jitter are examples of local and extremely fast dynamic variations with duration less than a clock-cycle.

In general, slow-changing and global effects such as process variations and ageing effects are relatively easier to predict and compensate for. For instance, binning is a well-known technique to compensate for inter-die process variations. In contrast, fast-changing and local-effects do not provide sufficient temporal and spatial context necessary for compensation. As such, guardbanding is the only way to compensate for such effects. Example for such high-frequency and localized events are capacitive coupling effects and clock-tree jitter.



	STATIC	DYNAMIC	
	EXTREMELY SLOW	SLOW-CHANGING	FAST-CHANGING
GLOBAL	Inter-die process variations Life-time degradation (NBTI, TDDB)	Package/Die VDD fluctuations Ambient temperature variations	PLL jitter IR drop Ldi/dt
LOCAL	Intra-die process variations	Local IR Temperature hot-spots	Capacitive Coupling Clock-tree jitter

Fig. 2 Sources of variations—taxonomy

In the subsequent sections, we provide an analysis of several variation-mitigation techniques including tracking circuits, error-resilient computation and adaptive-clocking techniques.

3 Tracking Circuits for Variation-Mitigation

Traditional adaptive techniques [6–13] based on canary or tracking circuits can compensate for certain manifestations of PVT variations that are global and slow changing. These circuits are used to tune the processor voltage and frequency taking advantage of available slack. Tuning is limited to the point where delay measurements through the tracking circuits predict imminent processor failure.

These circuits are limited by measurement uncertainty, the degree to which current and future events correlate and the latency of adaptation. Substantial margining for fast-moving or localized events, such as Ldi/dt, local IR-drop, capacitive coupling, or PLL jitter must also be present to prevent potential critical-path failures. These types of events are often transient, and while the pathological case of all occurring simultaneously is extremely unlikely in a real system, it is impossible to rule this out. Tracking circuits also incur significant calibration overhead on the tester to ensure critical-path coverage over a wide range of voltage and temperature conditions. The delay impact of local variations and fast-moving transients worsens at advanced process nodes due to aggressive minimum feature lengths and high levels of integration. This undermines the efficacy of tracking circuits.

Synthesized and automatically placed-and-routed designs present even greater challenges. Figure 3 highlights critical-paths on a Cortex-A9 core converging on a single critical-path endpoint. There are in excess of 100 paths within 70 ps of the critical-path at 90 nm technology. These paths cover 377 unique instances and 118 unique cell-masters, thereby making the problem of creating tracking paths extremely difficult.

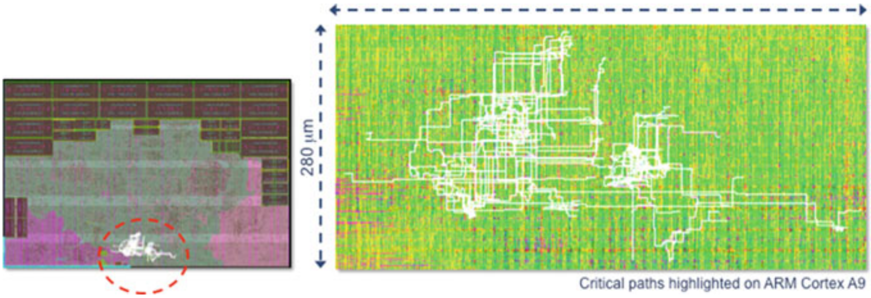


Fig. 3 Critical-paths on the ARM Cortex-A9 illustrating the complexity of creating suitable tracking circuits for synthesized and placed-and-routed designs

4 Error-Resilient for Variation-Mitigation—Razor

In contrast with the traditional adaptive techniques, the Razor approach [2–5] exploits the observation that the pathological combination of worst-case variation conditions occur extremely rarely in practice. Therefore, in Razor, requisite margins are added to the operating point dynamically according to the workload, prevailing environmental and silicon conditions.

In Razor, we exploit the dynamic nature of variations to speculatively operate a processor without statically added timing guardbands. Speculative operation requires efficient circuitry for reliable detection of and subsequent recovery from timing violations. A combination of error-detecting circuits and microarchitectural recovery mechanisms create a system that is robust in the face of timing errors, and can be tuned to an efficient operating point by dynamically eliminating unused guardbands.

The operational principle of Razor is illustrated in Fig. 4 and shows the qualitative relationship between the supply voltage, energy consumption and pipeline throughput of a Razor-enabled processor [3]. The PoFF of the processor (V_{ff}) and the minimum allowable voltage of traditional DVS techniques (V_{margin}) are also labelled in the figure. V_{margin} is much higher than V_{ff} under typical conditions, since safety margins need to be included to accommodate for worst-case operating conditions. Razor relies on in situ error-detection and correction capability to operate at V_{ff}, rather than at V_{margin}. The total energy of the processor (E_{tot}) is the sum of the energy required to perform standard processor operations (E_{proc}) and the energy consumed in recovery from timing errors ($E_{recovery}$). Of course, implementing Razor incurs power overhead such that the nominal processor energy (E_{nom}) *without* Razor technology is slightly less than E_{proc}. This overhead is

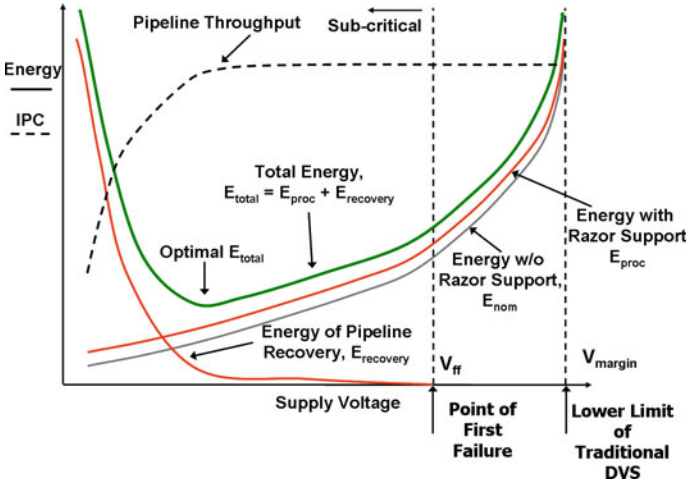


Fig. 4 Razor operational principle

attributed to the use of delay-error tolerant flip-flops on the critical-paths and the additional recovery logic required for Razor. However, since the extra circuitry is deployed only for those flip-flops that have critical-paths terminating in them, the power overhead due to Razor is fairly minimal. Razor-based systems in [3–5, 14] report power overheads that range between 2–8%.

As the supply voltage is scaled, the processor energy (E_{proc}) reduces quadratically with voltage. However, as voltage is scaled below the first failure point (V_{ff}), a significant number of paths fail to meet timing. Hence, the error rate and the recovery energy ($E_{recovery}$) increase exponentially. The processor throughput also reduces due to the increasing error rate because the processor now requires more cycles to complete the instructions. The total processor energy (E_{tot}) shows an optimal point where the rate of change of $E_{recovery}$ and E_{proc} offset each other. Thus, in the context of Razor, a timing error is not a catastrophic failure but a trade-off between the quadratic energy savings due to voltage scaling versus the overhead of recovery due to errors.

The concept of error-detection and correction has traditionally been widely employed in communications and signal-processing applications. In such applications, it is well known to trade-off transmitter power for heavyweight error-correction at the receiver end. In Razor, the concept of error-detection and correction are applied to general-purpose computing.

The RazorI approach relies upon temporal redundancy for timing-error detection. Robustness to Single-Event Upset failures through temporal redundancy has been shown to be particularly effective in a technique first pioneered in [15]. In the RazorI scheme (shown in Fig. 5), every rising-edge triggered critical-path flip-flop is augmented with a so-called shadow latch that samples at the falling edge of the clock. An error signal is flagged when the early speculative sample differs from the correct sample at the shadow latch. In the event of a timing error, a pipeline “restore” signal overwrites the potentially incorrect data in the main flip-flop with correct data from the shadow latch, thereby restoring correct state with a single cycle penalty.

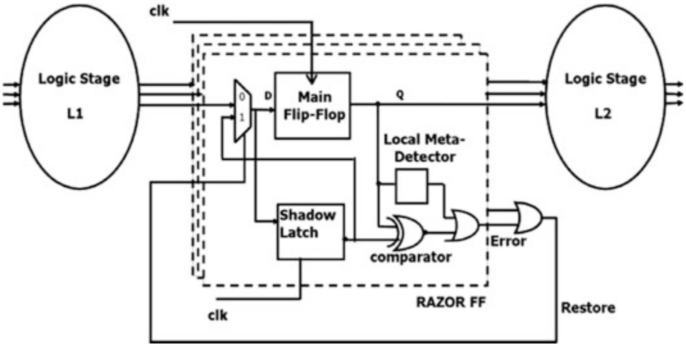


Fig. 5 RazorI flip-flop—conceptual design

The scheme relies upon a pipeline recovery mechanism based on the counter-flow microarchitecture design and requires specialized circuitry for metastability detection and recovery. Process-variability and margining requirements on the metastability detector complicate its deployment in high-performance microprocessors.

The RazorII approach eliminates the need for such a detector by splitting error-detection and correction between circuits and microarchitecture domains. Error-detection occurs exclusively in the RazorII flip-flop and recovery relies upon a conventional check-pointing and replay mechanism that is typically used in most high-performance microprocessors in order to support speculation mechanisms. The schematic for the RazorII flip-flop using a transition-detector is shown in Fig. 6.

Error-detection in the RazorII approach uses a transition detector to generate a pulse out of a transition on the data input. This pulse is then captured within an error-detection window to flag a timing error. The RazorII approach was integrated within an ARM processor implementing a subset of the ARM instruction set architecture. The pipeline design is shown in Fig. 7. Every critical-path endpoint is protected using Razor flip-flops (RFF). The error signal of individual flip-flops is combined together to create the pipeline error signal. The pipeline error signal engages a replay mechanism that restores correct state within the pipeline (Fig. 7).

A Razor dynamic voltage controller compensates for variations by monitoring error rates within the system and adjusting the supply voltage accordingly. The

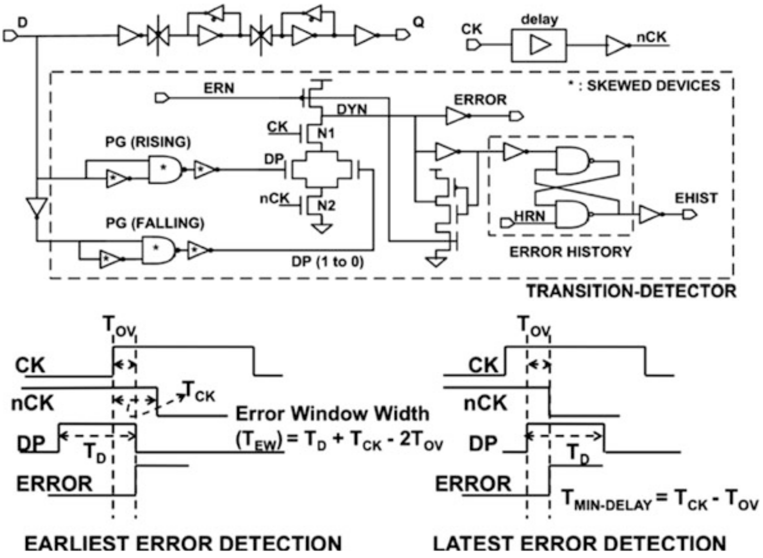


Fig. 6 Transition-detector circuit schematic

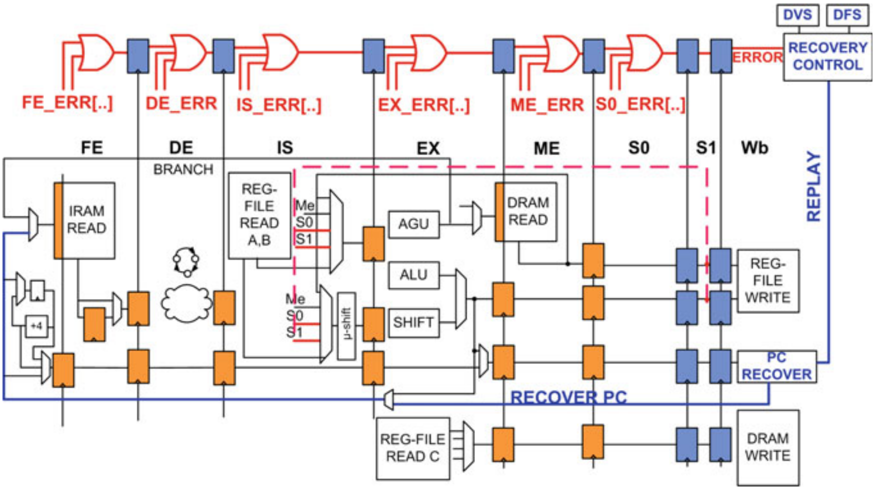
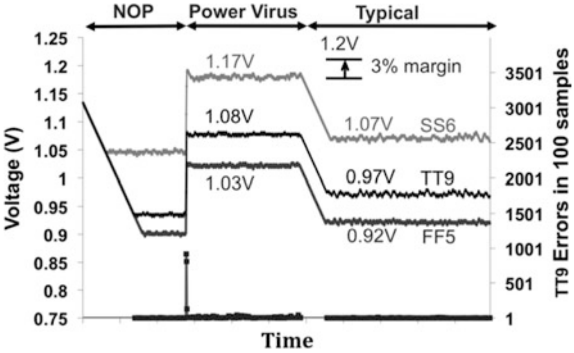


Fig. 7 Pipeline design of an error-resilient microarchitecture

Fig. 8 Response of the razor DVS controller—supply voltage is modulated depending upon the code being executed in the processor. The controller seeks the point of first failure for each code phase



voltage controller automatically tunes the system to operate at the PoFF of the system. The Razor voltage controller response for a code with three distinct phases is shown in Fig. 8 where a 30% energy saving is obtained on a per-die basis by automatically eliminating margins through error-detection and correction.

The Razor approach incurs verification and validation challenges due to the complexities of embedding fine-grained error-detection and recovery within the processing pipeline. On the other hand, data-processing systems such as DSP accelerators are particularly suitable towards error-detection and correction since a data-path dominated pipeline with simplified control typically characterizes them.

In the following, we illustrate how Razor concepts are equally applicable towards DSP accelerators.

5 Error-Resilient DSP Accelerators

Modern mobile and multimedia System-on-Chip (SoC) designs are rapidly evolving into complex, heterogeneous systems. In addition to high-performance application processors, such SoCs rely upon dedicated accelerators to deliver high-performance under stringent power budgets. Unlike microprocessors, DSP accelerators are data-path-dominated with relatively simplified control-plane logic. Such applications are often dominated by tight loops processing large amounts of streaming data, so it is natural to implement these loops as hardware Loop Accelerators (LA). Hardware LAs favourably trade-off surplus transistors to deliver order-of-magnitude higher efficiency compared to the software-only solution in programmable processors, although at the expense of limited or no flexibility.

In [16], we describe the first application of Razor to hardware loop accelerators (RZLA). In contrast with microprocessors, LAs are a class of coprocessors that accelerate a particular function and as such do not need to maintain an internal architectural state. Instead, queues are used in a dataflow-like manner to transfer transient data between functional units. This makes the LAs extremely amenable for implementing Razor recovery, as simply extending existing queues provide the necessary storage for the speculative state in flight, until it is validated using Razor.

Das et al. [16] shows the baseline microarchitecture of the RZLA (Fig. 9). The RZLA is a hardware realization of a modulo scheduled loop. Modulo scheduling is a software pipelining technique that achieves high parallelism by overlapping successive iterations of a loop. The RZLA microarchitecture exploits this parallelism obtained using modulo scheduling through the use of multiple functional units (FUs), each dedicated to a specific operation in the loop. The FUs are labelled ADD (adder), MULT (multiplier), BR (branch unit) and MEM (memory access unit). Unlike microprocessors, the RZLA does not require explicit support for mechanisms such as exception handling. Therefore, state is primarily maintained in Shift-Register Files (SRF) to be consumed when required and then immediately discarded. Wires from the SRF back to the FU inputs allow data transfer from producer to consumer Fig. 9.

The RZLA is architected such that exact recovery is achieved in the event of a timing error. However, most DSP algorithms allow inexactness in the final computational output as long as the algorithmic performance metrics are met. These metrics could be stop-band attenuation in a Finite Impulse Response (FIR) filtre or Peak Signal-to-Noise Ratio (PSNR) in an image compression algorithm. We take advantage of this in wherein we rely upon an approximate error-correction (AEC) algorithm in conjunction with controlled time-borrowing to achieve 37% energy saving in a FIR pipeline [17, 21].

Figure 10 shows the pipeline diagram of 16-tap Razor FIR filter. This design utilizes RFFs at critical-path endpoints to monitor for timing errors. Similar to the RazorII design, the RFF uses a pulsed-latch architecture that is transparent in the high-phase of the clock. Error-detection is postponed to the negative clock-phase. This allows late-arriving transitions to opportunistically time-borrow from the

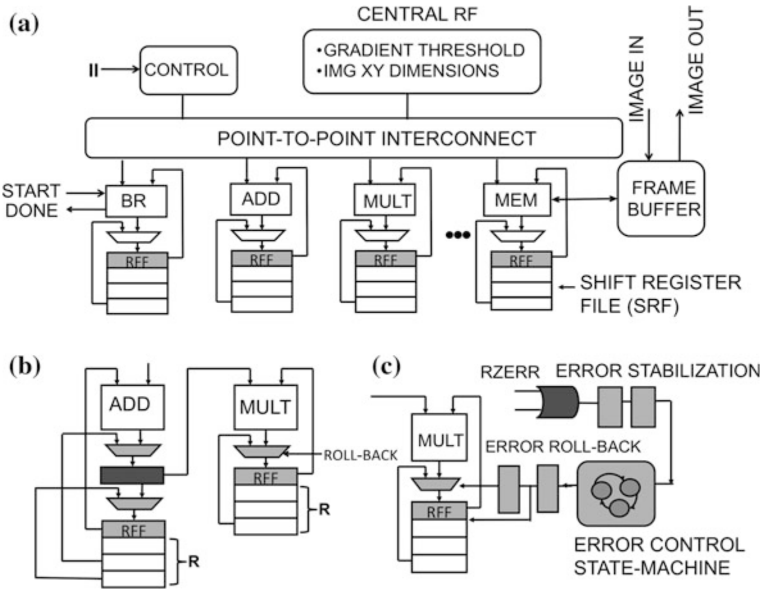


Fig. 9 **a** Architecture of the loop-accelerator implementing the inner kernel of sobel edge-detection algorithm. **b** Extension of shift-register files to incorporate error-correction by keeping **c** Error-control state machine

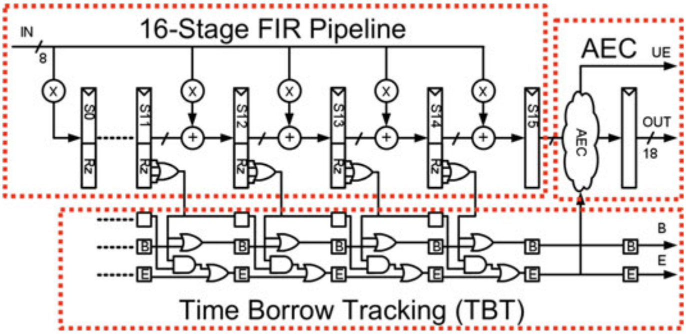


Fig. 10 FIR pipeline implementing a combination of exact and approximate error-correction. Each stage of the FIR pipeline has critical-paths that are protected using pulsed-latch-based razor flip-flops that correct timing errors on the fly. Approximate error-correction is deployed when successive cycles of timing-borrowing is detected

succeeding clock-cycle. Automatic time-borrowing achieves two critical objectives: (1) it leads to a high-performance design enabling 1 GHz operation and (2) it enables on-the-fly error-correction in the event of a timing error.

Thus, error recovery in our scheme is exact in the event of rare timing errors. However, if the error-rate pattern is bursty, it leads to multiple cycles of successive time-borrowing that can potentially exhaust available timing margin. In order to limit the error-magnitude induced due to excessive time-borrowing, we implement an approximate error-recovery scheme that augments exact recovery through time-borrowing. In this scheme, we track successive cycles of time-borrowing. When two cycles of time-borrowing is detected, the AEC block is engaged that replaces the final computational output with a spline-interpolated estimation.

The AEC algorithm uses four neighbouring correct samples (two backward and two forward samples) to generate an estimate of the erroneous sample. Consequently, the algorithmic performance of AEC is impacted as the number of available correct samples reduces. However, the performance with AEC is still significantly better than no error-correction at all.

6 Adaptive Clocking for Supply-Voltage Variations

Typically, error-resilient techniques are robust against all sources of variations. However, they add significant computational resources to implement detection and correction. Several architectural, algorithmic and circuit techniques have to be undertaken in order to limit the resulting overheads of error-correction. Alternative techniques have been pursued to address the overheads and computational complexity of error-resilient techniques. Tracking circuits, ageing monitors and process-binning are examples of such techniques that are comparatively simple in design and implementation. However, these techniques are ineffectual against fast-changing variations such as supply-voltage fluctuations.

Supply-voltage variations are one of the strongest determinants of guardbands in design due to strong correlation of transistor propagation delay with supply voltage. Adaptive-clocking techniques have been developed to particularly address the effect of supply-voltage variations. The key idea in adaptive clocking is to stretch the system-clock frequency in response to supply-voltage variations. Thus, the system clock slows down during periods of supply-voltage droops and increases again when the supply-voltage rises again.

Before we discuss adaptive-clocking techniques, it is important to understand the frequency- and time-domain behaviour of on-chip power-supply networks.

6.1 Power-Delivery Network Basics

Figure 11 [21] shows a simplistic representation of the power-delivery network (PDN) composed of a die-package-PCB system [18]. The switching transistors on the die are lumped together and modelled as a current source, I_{DIE}. Explicit on-die decoupling capacitors and non-switching transistors act as local charge reservoirs that are modelled by a capacitor, C_{DIE}. The power-line traces on the package and board are represented using R-L networks. Discrete decoupling capacitors

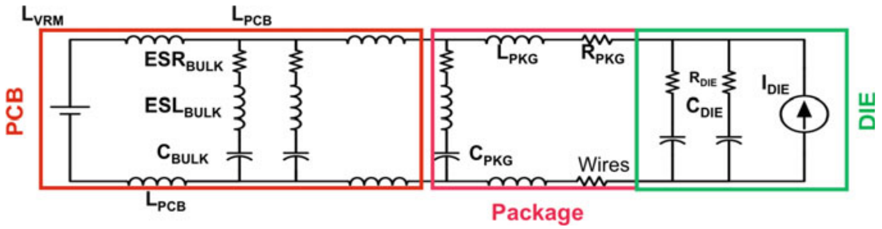


Fig. 11 Simplified representation of a power-supply network

(henceforth, referred to as decaps) on the package (C_{PKG}) and the bulk capacitors on the PCB (C_{BULK}) are modelled by capacitors in series with their effective series resistance (ESR) and inductance (ESL).

$$\Delta V_{\mathrm{DIE}}(t) \cong 2\mathrm{I}_{\mathrm{max}}R + \mathrm{I}_{\mathrm{max}}\sqrt{\frac{2L_{\mathrm{PKG}}}{C_{\mathrm{DIE}}}} \cdot \mathrm{e}^{-\frac{R}{2L_{\mathrm{PKG}}}t}\,\sin(\omega r - \theta) \tag{1}$$

Equation 1 shows the analytical solution for the voltage droop seen at the die supply rails for such a simplified model of the PDN. The voltage droop can be decomposed into a DC IR-drop term and an AC Ldi/dt term. The resistive component of the droop is addressed by increasing the metallization resources in the PDN. The inductive component is a complex trade-off between the package and the die and far exceeds the resistive droop magnitude in modern computing systems.

Figure 12 shows the PDN input impedance (as seen from the die) as a function of frequency for the simplified PDN in Fig. 1. The impedance spectrum shows three distinct impedance peaks due to each capacitor resonating with its counterpart inductor. The highest impedance peak, referred to as the *first-order resonance,* also occurs at the highest frequency (~100 MHz) and is due to the resonance between the die capacitance and the package inductance. The *second-* and *third-order* resonances are due to downstream capacitor networks, and occur at relatively lower frequencies (~1 MHz and ~10 kHz for the 2nd and 3rd-order resonances, respectively) Fig. 12.

Microarchitectural events such as pipeline interlocks cause current-step excitations that exercise the three prominent system resonance frequencies in the PDN (Fig. 12). The maximum magnitude of the voltage droop is caused due to the first-order resonance, which as such dominates the total timing margin.

6.2 Adaptive-Clocking Approaches

Adaptive-clocking techniques slow-down the system clock to mitigate the effect of the first-order supply droop. There are two major categories of adaptive supply techniques. The so-called “analogue” approach provides a continuous modulation

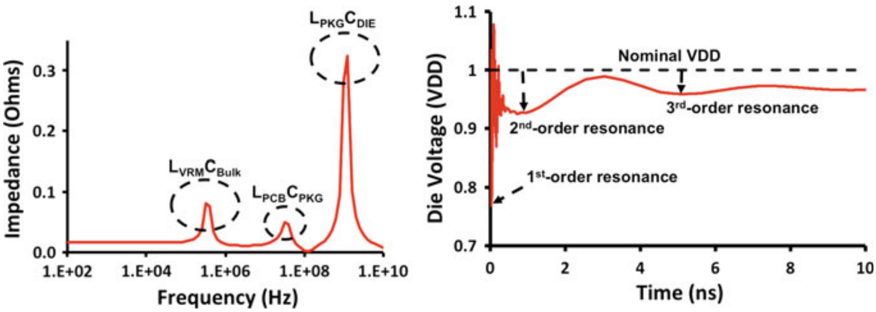


Fig. 12 Frequency- and time-domain response of a Power-Delivery Network (PDN) to a step-current excitation. The PDN response shows the presence of multiple resonance frequencies in the frequency-domain. The time-domain response also shows the same frequencies. In particular, the first-order resonance shows the highest supply-voltage droop at the highest frequency [21].

of the system clock in response to supply-voltage droops. “Digital” techniques, on the other hand, provide thresholded clock-adaptation, i.e. the supply voltage is compared against a certain threshold and modulation is undertaken only when the supply-voltage droops below the threshold.

Kurd et al. describe the analogue approach in [19] wherein power-supply noise is directly mixed into the voltage-controlled oscillator (VCO) of a PLL. The low-bandwidth of the PLL filtres out the high-frequency modulation of the VCO frequency, thereby mitigating concerns against loop stability.

Grenat et al. [20] present a “digital” modulation technique wherein the supply voltage is compared against a programmable threshold. A threshold-crossing initiates a clock-modulation scheme wherein cycle stretching is achieved by choosing successive phases out of the output of a Delay-Locked Loop.

Thus, adaptive clocking enables elimination of a subset of guardbands due to high-frequency supply-voltage fluctuations. Nominal operation of the system occurs at a higher frequency and the system slows down only when a droop-condition is encountered. Adaptive clocking is limited by the response latency that is limited by the clock tree depth and timing delay incurred in detection and initiated response in the event of a voltage droop [22]. Ensuring robustness when operating under reduced guardbands is a key challenge for adaptive technique approaches.

7 Conclusion

In this chapter, we reviewed various technological challenges for variation-tolerant computing. We reviewed three classes of techniques, namely tracking circuits, error-resilient computing and adaptive clocking. In particular, we focused on a particular flavour of error-resilient technique called Razor that enables energy-efficient operation by actively allowing timing errors to occur. We reviewed various approaches applied to university and industrial processors that demonstrate reliable energy-efficient operation using Razor-based dynamic adaptation. We showed measurement results where Razor error-correction enables robust operation in the presence of radiation-induced SER failures. Variation-tolerance remains a key design challenge, particularly as process technology scales to sub-10 nm critical dimensions. None of the techniques described in the chapter are perfect silver bullets due to the trade-offs between complexity, efficiency and engineering applicability that are involved. Hence, there is an urgent requirement for continued research investment in this area, both in academia and in industry. As process technology reaches fundamental physical limits, such techniques will prove to be an effective recourse to reliable computation in presence of failure-prone transistors.

References

1. H. Esmaeilzadeh et al., Dark silicon and the end of multicore scaling. Micro IEEE **32**(3), 122, 134 (2012)
2. D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N.S. Kim, K. Flautner, Razor: circuit-level correction of timing errors for low-power operation. IEEE Micro **24**(6), 10–20 (2004)
3. S. Das et al., A self-tuning DVS processor using delay-error detection and correction. J. Solid-State Circ. (2006)
4. S. Das et al., RazorII: in situ error detection and correction for PVT and SER tolerance. IEEE J. Solid-State Circ. **44**(1), 32–48 (2009)
5. D. Bull, S. Das, K. Shivashankar, G. Dasika, K. Flautner, D. Blaauw, A power-efficient 32 bit arm processor using timing-error detection and correction for transient-error tolerance and adaptation to PVT variation. IEEE J. Solid-State Circ. **46**(1), 18–31 (2011)
6. J. Tschanz et al., Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging, in *2007 IEEE International Solid-State Circuit Conference* (2007), pp. 292–293
7. K.J. Nowka et al., A 32-bit POWERPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling. IEEE J. Solid-State Circ. **37**(11), 1441–1447 (2002)
8. A. Drake et al., A distributed critical-path timing monitor for a 65 nm high-performance microprocessor, in *IEEE International Solid-State Circuit Conference* (February 2007), pp. 398–399
9. T. Fischer et al., "A 90-nm variable frequency clock system for a power-managed itanium architecture processor. IEEE J. Solid-State Circ. 218–228 (2006)
10. R. McGowen et al., Power and temperature control on a 90-nm itanium family processor. *IEEE J. Solid-State Circ.* 229–237 (2006)
11. S. Das D. Blaauw, Adaptive design for nanometer technology, in *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009* (2009), pp. 77–80
12. S. Youngmin et al., 28 nm high-metal-gate heterogeneous quad-core CPUs for high-performance and energy-efficient mobile application processor, in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers* (*ISSCC*) (17–21 February 2013), pp. 154, 155
13. F. Masaki et al., A 28 nm high κ-metal-gate single-chip communications processor with 1.5 GHz dual-core application processor and LTE/HSPA + -capable baseband processor, in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers* (*ISSCC*) (17–21 February 2013), pp. 156, 157

14. K. Bowman et al., A 45 nm resilient microprocessor core for dynamic variation tolerance. IEEE J. Solid-State Cir. **46**(1), 194–208 (2010)
15. M. Nicolaidis, Time redundancy based soft-error tolerance to rescue nanometer technologies, in *Proceedings of the IEEE VLSI Test Symposium* (April 1999), pp. 86–94
16. S. Das, G. Dasika, K. Shivashankar, D. Bull, A 1 GHz hardware loop-accelerator with razor-based dynamic adaptation for energy-efficient operation, in *IEEE Custom Integrated Circuits Conference* (September 2013)
17. P. Whatmough, S. Das, D. Bull, A low-power 1 GHz razor FIR accelerator with time-borrow tracking pipeline and approximate error correction in 65 nm CMOS, in *IEEE International Solid-State Circuits Conference* (February 2013), pp. 428–429
18. J. Tschanz et al., Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance, in *2009 Symposium on VLSI Circuits* (2009) pp. 112–113
19. N. Kurd et al., A Family of 32 nm IA processors. IEEE J. Solid-State Circ. **46** (1), 119–130 (2011)
20. A. Grenat, S. Pant, R. Rachala, S. Naffziger, Adaptive clocking system for improved power efficiency in a 28 nm x86-64 microprocessor, in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers* (*ISSCC*) pp. 106–107
21. S. Das, P.N. Whatmough, D. Bull, Modeling and characterization of the system-level power delivery network for a dual-core ARM cortex-A57 cluster in 28 nm CMOS. ISLPED (2015)
22. P.N. Whatmough, S. Das, D. Bull, Analysis of adaptive clocking technique for resonant supply voltage noise mitigation. ISLPED (2015)
23. M. Gupta et al., Cross-layer system resilience at affordable power, in *2014 IEEE International Reliability Physics Symposium* (June 2014)
24. P.N. Whatmough, S. Das, S.D.M. Bull, I. Darwazeh, Circuit-level timing error tolerance for low-power DSP filters and transforms. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on **21**(6), 989–999 (2013)