iRazor: Current-Based Error Detection and Correction Scheme for PVT Variation in 40-nm ARM Cortex-R4 Processor

Yiqun Zhang, *Student Member, IEEE*, Mahmood Khayatzadeh, *Member, IEEE*, Kaiyuan Yang, *Member, IEEE*, Mehdi Saligane, Nathaniel Pinckney, Massimo Alioto, *Fellow, IEEE*, David Blaauw, *Fellow, IEEE*, and Dennis Sylvester, *Fellow, IEEE*

***Abstract*— This paper presents iRazor, a lightweight error detection and correction approach, to suppress the cycle time margin that is traditionally added to very large scale integration systems to tolerate process, voltage, and temperature variations. iRazor is based on a novel current-based detector, which is embedded in flip-flops on potentially critical paths. The proposed iRazor flip-flop requires only three additional transistors, yielding only 4.3% area penalty over a standard D flip-flop. The proposed scheme is implemented in an ARM Cortex-R4 microprocessor in 40 nm through an automated iRazor flip-flop insertion flow. To gain an insight into the effectiveness of the proposed scheme, iRazor is compared to other popular techniques that mitigate the impact of variations, through the analysis of the worst case margin in 40 silicon dies. To the best of the authors' knowledge, this is the first paper that compares the measured cycle time margin and the power efficiency improvements offered by frequency binning and various canary approaches. Results show that iRazor achieves 26%–34% performance gain and 33%–41% energy reduction compared to a baseline design across the 0.6- to 1-V voltage range, at the cost of 13.6% area overhead.**

***Index Terms*— Adaptive circuits, canary circuits, error detection and correction (EDAC), Razor, variation tolerance.**

I. Introduction

PROCESSORS and systems-on-chip (SoC) are traditionally designed to accommodate for worst case variations, with a cycle time target that incorporates process, voltage, temperature, and aging margins, which in turn substantially degrade performance and energy efficiency. Adaptive designs with *in situ* error detection and correction (EDAC) capability have been widely explored to suppress the cycle time margin, using specialized registers on critical paths that perform timing EDAC [1]–[9]. Unfortunately, such specialized registers typically incur a large area overhead compared to conventional registers. For example, Razor requires 44 extra transistors per register [1], double sampling with time borrowing (DSTB) [2] needs 26 extra transistors, and Razor-lite [3] requires eight extra transistors, which is currently the EDAC approach with smallest overhead. The significant area overhead has been an obstacle to the adoption of EDAC approaches in commercial designs, and currently there is no significant commercial processor implementing EDAC approaches [10]. In addition, the performance and energy gains from EDAC approaches have not been thoroughly quantified in relation to competing approaches to mitigate variations at lower overhead, such as frequency binning, critical path monitors [11]–[14], and canary circuits [15].

In this paper, we propose a very lightweight EDAC approach that is based on a novel specialized flip-flop requiring only three additional transistors, compared to a conventional D flip-flop. The iRazor flip-flop [16] leverages a current-based mechanism to detect timing violations at the cost of only 4.3% larger area than a conventional D flip-flop. The iRazor approach is validated through the implementation of an ARM Cortex-R4 processor testchip [17], as representative of designs with non-trivial complexity with eight pipeline stages and a gate count in excess of 1 Mgates. An automated flip-flop insertion flow is adopted to augment the design with iRazor flip-flops, based on a systematic design strategy to ensure timing closure. Measurement results show that iRazor achieves 26%–34% improvement in performance and 33%–41% reduction in energy across the 0.6–1 V voltage range, compared with a baseline design without EDAC capabilities. Such improvements are achieved at the cost of 13.6% area overhead, compared to a conventional design. As further contribution and to the best of the authors' knowledge, this is the first paper that quantitatively compares today's industry-standard methods to mitigate variations (e.g., margining, frequency binning, and different canary approaches), based on silicon measurements on the same processor design. The characterization of 40 silicon dies provides an insight into the design margin required by iRazor and other techniques, quantifying the performance and power improvement and the related area and energy cost.

The remainder of this paper is organized as follows. Section II reviews state-of-the-art circuit techniques to mitigate or suppress the design margin. Section III introduces the proposed iRazor flip-flop and a detailed analysis of its

Manuscript received March 16, 2017; revised July 20, 2017; accepted August 23, 2017. Date of publication October 6, 2017; date of current version January 25, 2018. This paper was approved by Associate Editor Dejan Markovic. This work was supported by the Singapore Ministry of Education under Grant MOE2014-T2-1-161. *(Corresponding author: Yiqun Zhang.)*

Y. Zhang, M. Khayatzadeh, K. Yang, M. Saligane, N. Pinckney, D. Blaauw, and D. Sylvester are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: zhyiqun@umich.edu).

M. Alioto is with the National University of Singapore, Singapore 117583.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2017.2749423

main properties. Section IV describes the architecture of the EDAC scheme. Section V describes the automated iRazor insertion flow, and the fabricated testchip details. In Section VI, the benefits and the cost of various industry-standard methods to mitigate the design margin are evaluated. Section VII presents the overall comparison of iRazor and the schemes in Section VI. Conclusions are drawn in Section VIII.

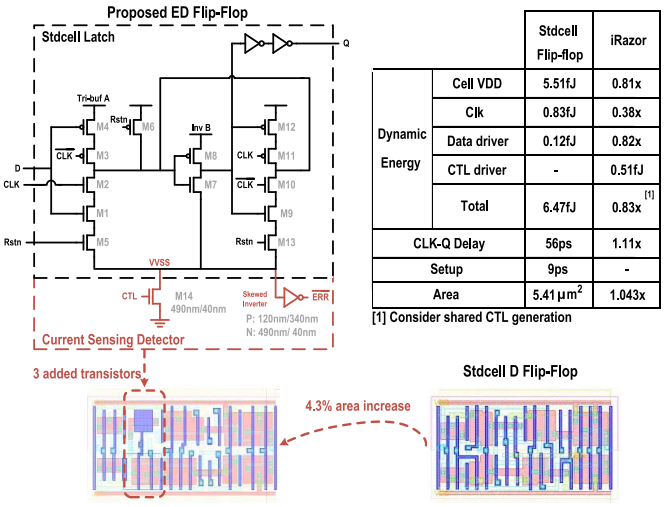
II. Review of Circuit Techniques to Mitigate the Cycle Time Margin

Traditionally, processors are margined to tolerate process, voltage and temperature (PVT) variations. Among the existing techniques to reduce their impact on the related cycle time margin, frequency binning entails the lowest overhead as it relies on additional testing time to perform coarse-grained discrete frequency tuning to mitigate process variations at given environmental conditions.

More sophisticated self-adapting design techniques introduce process and environmental sensors [e.g., ring oscillators (RO)] to further reduce the margin, and are customarily adopted in today's processor and SoC designs. These approaches can adapt to variations to some extent, monitoring them through "canary" circuits that mimic the delay of the critical path(s), and fitting the actual margins. However, the design margin cannot be completely eliminated by these approaches, due to the residual mismatch between the on-chip sensors (e.g., RO frequency) and the actual critical path delay.

EDAC approaches can virtually eliminate the design margin, based on the insertion of specialized registers on critical paths to perform timing EDAC. Among the proposed techniques, output waveform analysis [18], time-redundant latches [19], transition detector with time borrowing [2], DSTB [2], and different Razor latches [3], [4], [6], [20] have been proposed. For example, the Razor approach eliminates the design margin by allowing for reducing the clock cycle until timing constraints are barely met. This occurs right before timing failures are detected by specialized registers, such as Razor-I [1], Razor II [4], Bubble-Razor [20], and Razor-lite latches [3]. The key idea of Razor latches is that the data comes into the main flip-flop and is also tapped off to a shadow latch, which is clocked slightly later. The mismatch between the output of the main flip-flop and the shadow latch reveals the occurrence of the timing error. Once an error is detected, it can be corrected in several manners as proposed in previous work. For example, Fojtik *et al.* [20] uses a bubble propagation algorithm to send stalling signals to neighbors in half a cycle assuming a two-phase latch clocking. As another example, global clock gating and counterflow pipelining were proposed in Razor I [1]. In the former technique, the whole processor is stalled until correct values are reloaded. Through counterflow pipelining, a bubble is sent upstream and downstream pipeline stages at every clock cycle to prevent the propagation of errors and perform their correction.

Although EDAC techniques fundamentally eliminate the design margin, they suffer from relatively large area and energy overhead due to the complexity of the detection mechanism. For example, [3] require eight additional transistors per flip-flop or more. The direct and significant impact on cost has limited the diffusion of prior EDAC techniques, as confirmed by the lack of adoption in any significant commercial design to date, and motivates the introduction of novel lightweight EDAC schemes that can be truly afforded in real designs.



		Stdcell Flip-flop	iRazor
Dynamic Energy	Cell VDD	5.51fJ	0.81x
	Clk	0.83fJ	0.38x
	Data driver	0.12fJ	0.82x
	CTL driver	-	0.51fJ
	Total	6.47fJ	0.83x [1]
CLK-Q Delay		56ps	1.11x
Setup		9ps	-
Area		5.41 μm^2	1.043x

[1] Consider shared CTL generation

Fig. 1. Schematic of the proposed iRazor flip-flop with error detection capability, and its energy, delay, and area compared with conventional flip-flop standard cell (both positive edge triggered).

III. Proposed iRazor Circuit and Analysis

A. iRazor Flip-Flop and Circuit Analysis

The iRazor flip-flop supplements a latch circuitry [21] with asynchronous reset (signal *Rstn*) in Fig. 1 (drawn in black) with the lightweight error detection circuit (highlighted in red). The latter consists of a novel three-transistor current detector that reveals whether the latch is transiently drawing any transistor on-current after the clock edge, thus effectively detecting transitions occurring at the input of the iRazor flop. In the following, positive edge-triggered timing is assumed with no loss of generality.

Timing violations are caught within an error detection window during which the first tristate inverter (M1–M5 in Fig. 1) is transparent, and it represents the portion of the clock cycle when the input should not transition to avoid timing violations. The detection window is defined by setting the signal CTL in Fig. 1 as low, and timing violations are signaled by the active-low error signal $\overline{ERR}$ in Fig. 1. As discussed the following, the error detection window starts after the falling edge of CTL, thus enabling some amount of time borrowing at the very beginning of the clock cycle, in addition to the capability of subsequently detecting timing violations.

When the iRazor input correctly transitions before the rising clock edge and after the falling clock edge as in Fig. 2(a), CTL is high and transistor M14 in Fig. 1 is ON, thus tying the virtual ground virtual voltage source source (VVss) to ground. Accordingly, the iRazor circuit in Fig. 1 operates like a conventional flip-flop and updates its output at the rising clock transition, which makes the first tristate inverter transparent. In this case, the active-low error signal $\overline{ERR}$ is deasserted (i.e., $\overline{ERR}$ is set to 1) by the skewed inverter in

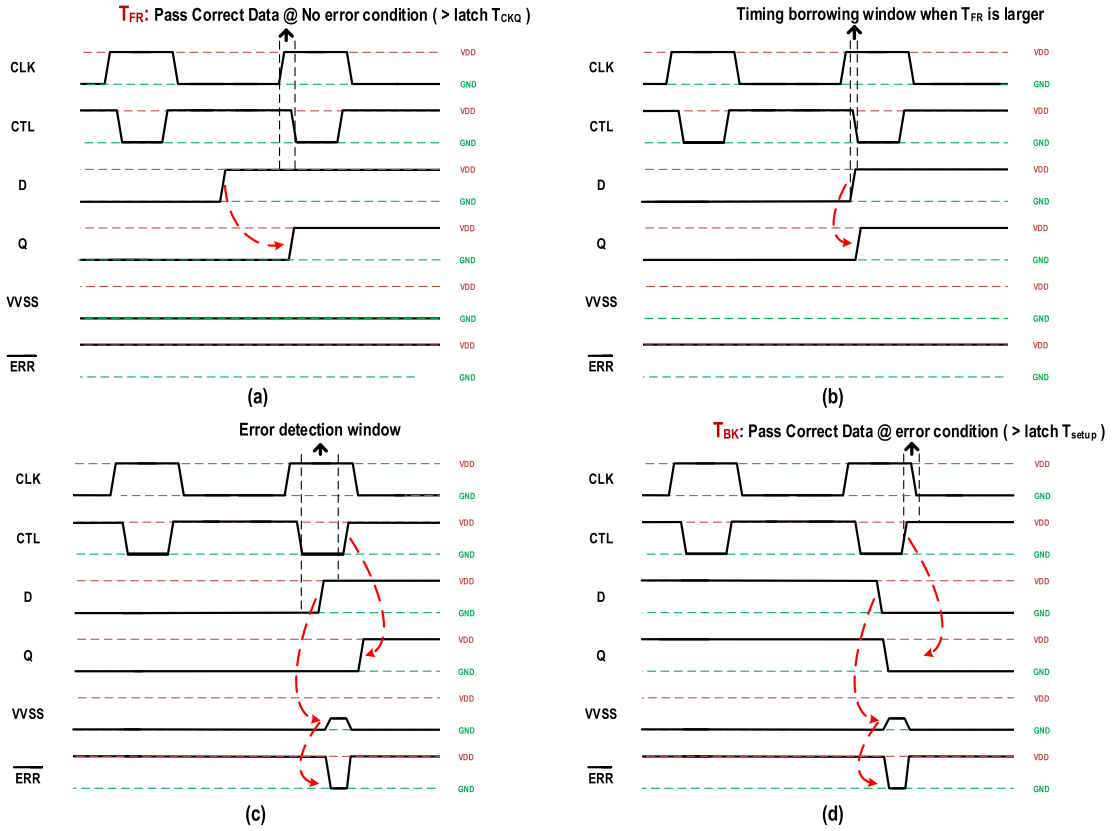


Fig. 2. Waveforms in iRazor flip-flop when (a) input D is correctly switching before the rising clock edge, (b) D is switching within the time borrowing window, (c) error is occurring due to the transition of the input D from 0 to 1 during the error detection window, and (d) error is occurring due to the transition of the input D from 1 to 0 during the error detection window.

red, as required. Instead, when the iRazor input D transitions after the rising clock edge and before the beginning of the error detection window as in Fig. 2(b), the iRazor latch is transparent and allows for timing borrowing. In this case, moderately late arriving inputs are forgiven and no error is flagged (i.e., $\overline{ERR} = 1$).

During the error detection window as in Fig. 2(c) and (d), the CTL signal is set to 0, transistor M1 is turned off, and the virtual ground is disconnected from the ground. If no input transition occurs during the error detection window, the virtual ground is dynamically held at ground, and no error is flagged by the skewed inverter in Fig. 1 (i.e., $\overline{ERR}$ is kept at 1). Instead, if the flip-flop input D performs a transition during the error detection window, the voltage of the floating virtual ground is raised by the charge provided by either the first tristate inverter (M1–M5) or the subsequent inverter (M7–M8), as discussed in the following. The red inverter in Fig. 1 is skewed low so that the raised virtual ground voltage lies beyond the inverter logic threshold, and hence $\overline{ERR}$ is set to 0, thus signaling an error. In particular, if D transitions from 0 to 1 during the error detection window [see Fig. 2(c)], the initially discharged capacitance at the virtual ground node VVss is charged by transistors M1–M2 and M5. This is due to the charge sharing with the capacitance at the output of the tristate inverter M1–M5, which was precharged at V_{DD} by M1–M5 before the input transition, since the input D was initially equal to 0. Similarly, when D transitions from 1 to 0 during the error detection window [see Fig. 2(d)], the capacitance at the virtual ground node is charged by transistor M7 due to the charge sharing with the capacitance at its output. In both cases, the virtual ground voltage VVss is raised and complemented by the skewed inverter in Fig. 1 to flag the error and hence set $\overline{ERR}$ to 0. According to the above considerations, the VVss node is dynamic and its signal integrity needs to be preserved through routine layout strategies, such as shielding or proper spacing of strong aggressors.

To ensure correct error detection, the error detection window has to be correctly aligned with the clock cycle. In particular, from Fig. 2(a), the falling edge of CTL marks the start of the detection window and must occur with sufficient delay after the rising clock edge. Otherwise, correct output transitions right after the clock edge would be incorrectly flagged as errors, due to the subsequent transition in the first tristate inverter (M1–M5) occurring a clock-to-Q delay after the clock edge. This minimum delay from the rising clock edge and between the beginning of the error detection window is here referred to as the front timing constraint T_{FR}, and must certainly exceed the flip-flop clock-to-Q delay to allow the data to pass through the slave latch M9–M13 without triggering an error. Larger values of T_{FR} allow time borrowing as in Fig. 2(b), although at the expense of a shorter error detection window.

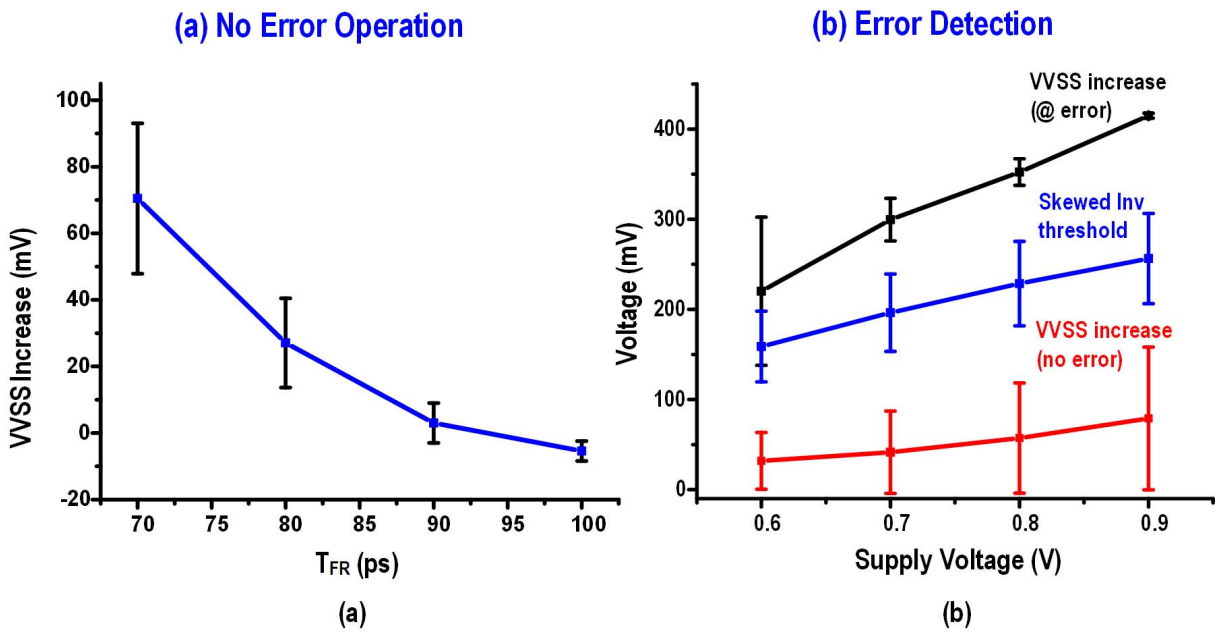


Fig. 3. Statiscal analysis of (a) virtual ground voltage (VVss) versus T_{FR} at no error operation (b) VVss increase with/without error, and Skewed Inverter threshold across supply voltages. 1000 Monte Carlo runs, whiskers indicate three standard deviations around the mean value.

To assure that the input data is correctly latched into the cross-coupled inverter pair (M7–M13 in Fig. 1) during the error detection window, the latter needs to end before the falling clock edge by an appropriate back time constraint T_{BK} as in Fig. 2(c) and (d). Quantitatively, T_{BK} needs to be greater than (or equal to) the latch setup time T_{setup}, so that metastability is prevented during the error detection window.

B. Analysis of the iRazor Flip-Flop Robustness, Area, and Energy

In general, increasing T_{FR} leads to a wider time borrowing window at the expense of a shorter error detection window. Also, larger T_{FR} reduces the probability of false positive errors due to the transition in the output of the first tristate inverter M1–M5 right after the rising clock edge, and ending sometime after a clock-to-Q delay (i.e., when the output of the tristate inverter is close to the steady state). More quantitatively, T_{FR} needs to be large enough to give transistor M14 enough time to bring the virtual ground VVss back to the ground voltage (since CTL = 1), after its temporary increase due to the above transition in M1–M5.

Monte Carlo simulations in Fig. 3(a) illustrate the relationship between T_{FR} and the VVss increase during time borrowing (i.e., when no error occurs), including variations. From Fig. 3(a), large enough T_{FR} values keep the VVss upward transition small when no error occurs. As shown in Fig. 3(b), large enough values of T_{FR} make the temporary VVss increase caused by data transitions in the time borrowing window smaller than the skewed inverter threshold voltage, and avoid false error triggering (shown by the blue line). In case of timing error occurrence [see black line in Fig. 3(b)], the VVss increase exceeds the skewed inverter threshold voltage to trigger an error. However, at low voltages the ability to detect an error is potentially compromised at very low voltages, for a given T_{FR}. For example, Fig. 3(b) shows that some error may not be occasionally flagged at 0.6 V and below, as the VVss increase might be higher than the skewed inverter logic threshold in some rare cases. Indeed, the whiskers of VVss and the threshold of the skewed inverter start overlapping at 0.6 V in Fig. 3(b).

Results of post-layout analysis of the iRazor flip-flop relative to a standard flip-flop[1] are reported in Fig. 1. The added three transistors in red in Fig. 1 increase the area by 4.3%, due to the large gate length of the PMOS transistor in the skewed inverter, as required to make its logic threshold closer to ground to better capture the VVss increase. In the adopted technology, increasing the gate length of PMOS to reduce the logic threshold is preferable to stacking, as the latter would entail a larger area penalty of 11.8%. The total dynamic energy of the iRazor flip-flop is decreased by 17% compared to the conventional flip-flop, when sharing the CTL generation circuitry, as discussed in the final chip implementation in Section IV. Fig. 1 also gives the breakdown of the energy across cell V_{DD}, clock, input driver and CTL driver. The iRazor clock-to-Q delay increases by 11% compared with the conventional flip-flop.

IV. iRazor Error Detection and Correction Scheme

This section describes the global EDAC scheme for iRazor, as shown in Fig. 4. This is similar to the global clock gating scheme mentioned in Section II. Local clock generators are used as the last level of the clock tree to generate the clock and the CTL signals in iRazor, as shown in Fig. 4. These generators are shared between registers to minimize the area

[1]The baseline flip-flop was taken from the same standard cell library in 40 nm that was adopted for the design of the test chip described in Section V.

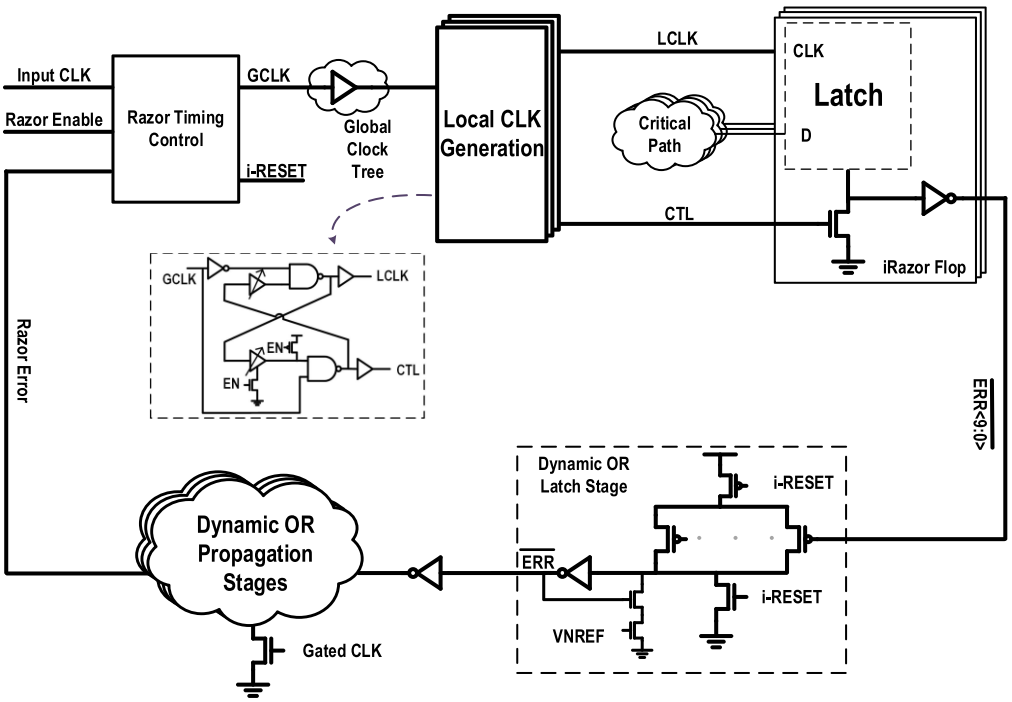


Fig. 4. Overall iRazor EDAC scheme diagram.

and energy overhead, and control the T_{FR} and T_{BK} windows in Section III to avoid the power overhead and the inter-clock skew that would be needed by two clock distribution networks.

Under normal operation when no error occurs, data arrives before the rising clock edge and the iRazor output Q latches the value after the clock rises, with $\overline{ERR}$ staying high. When an error occurs due to a data transition within the detection window, the $\overline{ERR}$ signal is pulled low by the skewed inverter in Fig. 1 of the relevant flip-flop. The resulting $\overline{ERR}$ signal experiences a negative pulse, which is captured by a PMOS-based dynamic OR-latch, which is shared by up to 10 iRazor flip-flops, as shown in Fig. 4. The aggregate output of the OR-latch is then ORed together with all other aggregate error signals within the processor by using conventional dynamic OR gates, thus generating the global Razor error signal shown in Fig. 4. This global Razor error signal then propagates through the Razor timing control as shown in Fig. 4. Razor timing control skips the clock edge following the occurrence of an error, providing the pipeline with a further cycle to resolve the error, as shown in the third cycle at the left of Fig. 5. (The error occurs in the second cycle.) Following error resolution, the dynamic OR-latches are reset using the *i-RESET* signal. Normal operation resumes in the next cycle (fourth cycle in Fig. 5), as the global razor error signal is reset to 0 when clock gating is released. The dynamic OR latch stages (Fig. 4) are reset through the *i-RESET* signal, which can catch the $\overline{ERR}$ signal generated by the iRazor flip-flop when the clock is either low or high. The dynamic OR propagation stages are reset using the gated CLK signal to keep the global Razor error signal to be high within the error recovery stage (in Fig. 5) to avoid glitches of the gated local clocks.

Using local detection and clock stalling, the pipeline is halted within one cycle of a detected error, allowing the EDAC technique to be integrated into the processor without requiring rollback or architectural changes. To accomplish this, the error signal must propagate through the above logic within one clock cycle. As shown at the right of Fig. 5, the error critical path includes: the clock tree delay to reach the clock tree leaves from the clock root first; the T_{FR} delay, the detection window itself; the error detection delay, the dynamic OR latch stage and three dynamic OR propagation stages; and finally the Razor timing control to ultimately generate the clock gating signal.

V. Automated iRazor Design Flow and Testchip Design

The automated and architecture-independent iRazor flow in Fig. 6 was developed and adopted to design an ARM Cortex-R4 processor, which is used as reference design example in the following.

The iRazor design flow starts with a placed and routed baseline design. Then, flip-flops to be razorized are selected, based on the tradeoff between the path coverage and the area overhead due to iRazor flip-flops, the transistor upsizing to meet timing, and the additional hold buffers, which are required to make the min-delay larger than the transparency window in the covered paths. As shown in Fig. 7, iRazor flip-flops are progressively inserted to cover paths with increasing timing slack (i.e., from the most to the least critical one), and higher path coverage entails a larger number of iRazor flip-flops and area. A high path coverage also makes the design hard to route. In the considered ARM Cortex-R4 design, from Fig. 7 a reasonable compromise between path

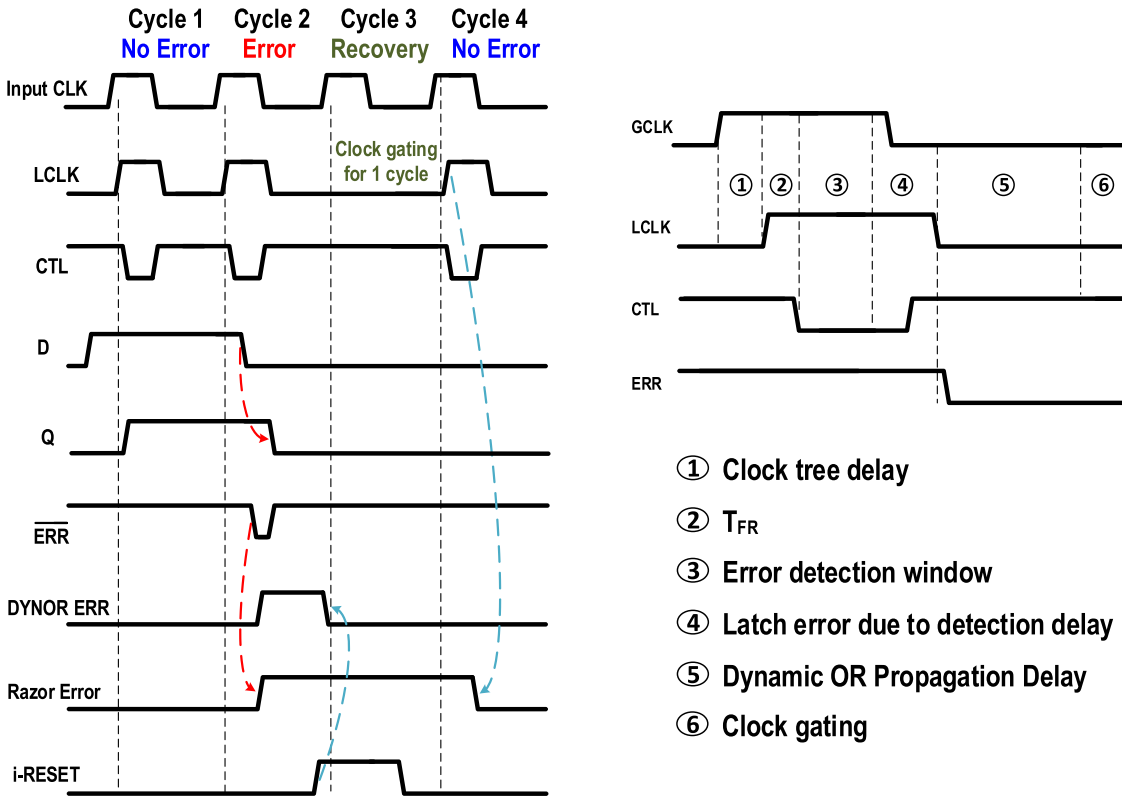


Fig. 5. iRazor timing diagram (left) and timing analysis of the error critical path (right).

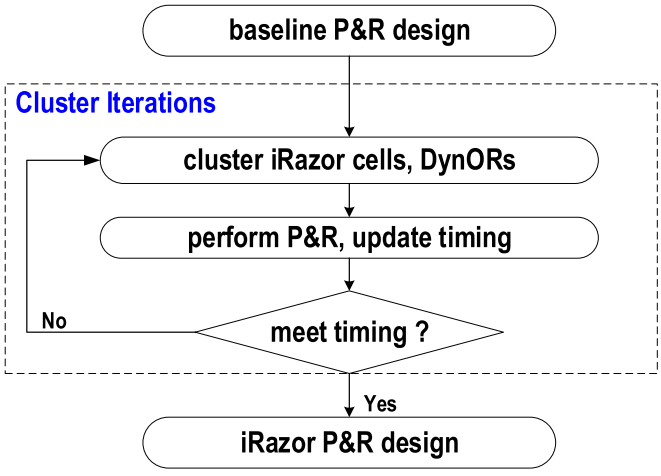


Fig. 6. Architecture independent automated flow for iRazor flip-flop replacement and clustering.

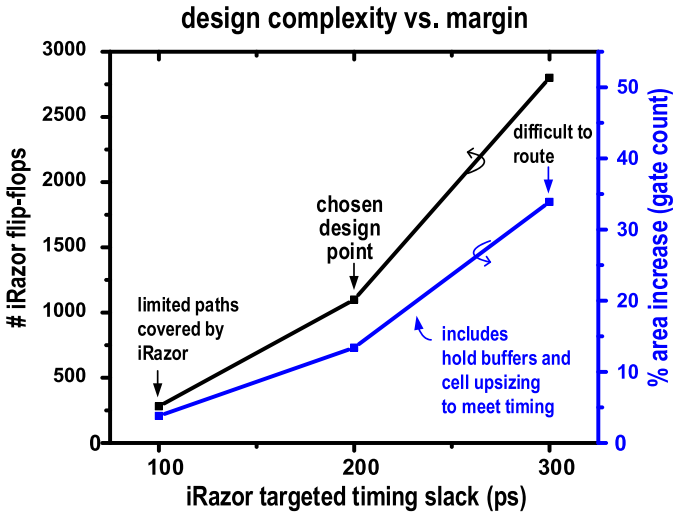


Fig. 7. Design complexity (in number of iRazor flip-flops) versus targeted timing slack of iRazor.

coverage and overhead is to cover paths with 200-ps timing slack or lower, replacing the corresponding conventional flip-flops with iRazor flip-flops. This leads to the replacement of 8.7% of the total flip-flop count. The resulting datapath delay histogram after razorizing is shown in Fig. 8 together with the baseline histogram. Overall, path delays in iRazor are pushed to the right because of the addition of hold buffers. The last two columns represent paths with iRazor flops.

After iRazor insertion, placement of dynamic ORs needs to be optimized. According to the initial placement of the baseline design, automated clustering of iRazor cells is performed to share the local clock generator and the different levels of dynamic OR trees. Both the physical locations and the loading in each stage are key factors for clustering. A threshold distance is set first for the iRazor flip-flop clustering into the same group, creating a new group once the threshold is exceeded. In this design, the distance threshold is set to 60, 300, and 1000 μm for the first, the second, and the third stage. Fig. 9 shows the resulting placement of iRazor flip-flops, dynamic OR latches, and subsequent stages.

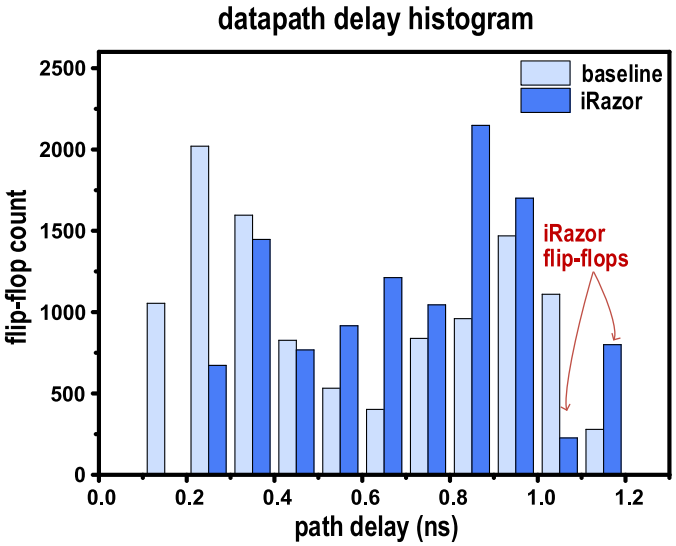


Fig. 8. Path delay histogram of baseline and iRazor design.

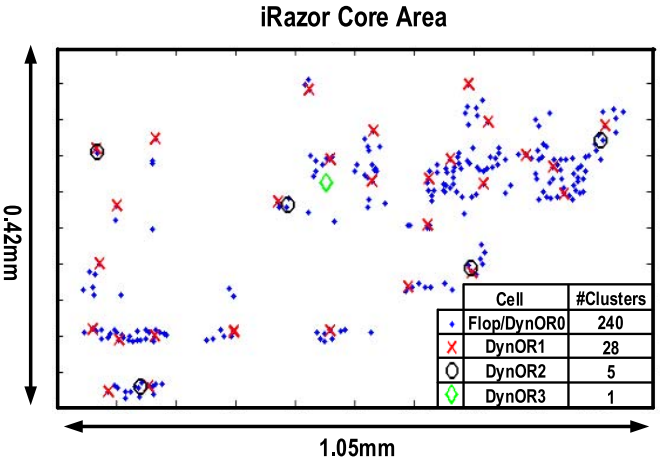


Fig. 9. iRazor cluster spatial position within the on-die processor footprint.

Then, place and route is performed, checking the timing of the overall error control feedback loop since the wirelength from the skewed inverter output to the dynamic OR-latch is critical for timing closure (see Fig. 5 right). If timing is not met, hierarchical iterations of clustering are performed followed by a new placement, while freezing the original iRazor flip-flop locations to facilitate convergence. Further iterations of clustering/placement are performed until the timing is closed. Then, a final iRazor place and routed design is achieved, with all prior steps performed in a fully automated fashion. As well known for all EDAC approaches, timing closure might not be guaranteed in very large designs, although iRazor is demonstrated to work in a microprocessor core that is an order of magnitude more complex than prior demonstrations (see Table I).

The effective overhead of the iRazor scheme relative to a conventional flip-flop-based design is shown in Fig. 10. First, three additional transistors are included in each latch, although the latch itself has eight fewer transistors than a conventional flip-flop. Then, 240 local clock generation blocks are used in the final design, each comprising 30 transistors. The additional

	3	Additional transistors over latch
-	8	Latch compared with flip-flop
+	30*240/1115	240 blocks of local clock generation over 1115 iRazor flops
=	1.46	

Fig. 10. iRazor effective overhead explicit calculation.

transistors are amortized across the 1115 iRazor flip-flops, resulting to an effective overhead of only 1.46 transistors per flip-flop.

Both the baseline and the iRazor designs of the targeted processor were implemented on a testchip, whose micrograph is shown in Fig. 11. The ARM Cortex-R4 processor was implemented in 40-nm CMOS, with a total number of flip-flops of approximately 13 000, of which 8.7% were razorized. The total number of gates increased by 13.4% when applying iRazor, due to the addition of minimum-sized hold time buffers, iRazor flip-flops, the OR tree, and the CTL tree, which respectively contributed by 10.06%, 0.95%, 0.27%, and 0.36% to the overall area increase, while the remaining 1.76% is due to signal routing. The total iRazor core area includes 8-kB instruction/data cache and 12-kB memory, and increased by about 13.6% compared to the baseline. Note that buffer insertion takes most of the area in logic in this specific design, although the memory size can be much larger in many other modern processors, in which case the percentage overhead is expected to be significantly reduced. Compared with previous EDAC testchips, this design marks a significantly more complex processor implementation, particularly in terms of the number of total and replaced flip-flops, other than gate count.

VI. Comparative Evaluation of iRazor and Previous Variation-Aware Techniques

Based upon the techniques discussed in Section II, 40 baseline chips were measured to gain an insight into the effectiveness of iRazor, compared to a baseline margined design, frequency binning and RO-based canary methods.

The worst case margining of 85 °C temperature, 10% supply drop, and 3σ process variation is used to define the baseline. As shown in Fig. 12(a), the histogram in red is the maximum operating frequency of 40 baseline chips at 1 V and room temperature, whereas the margined frequency able to work across all PVT variations is plotted in green. The detailed margin histogram of baseline at 1 V and room temperature is shown in Fig. 17(a). The margined frequency is typically 25% lower and up to 32% than the maximum frequency allowed by the measured chips. The detailed margin breakdown into PVT across 0.6–1 V is plotted in Fig. 12(b), which shows that voltage margin gives the largest contribution. As the processor voltage approaches the threshold voltage, the margin contributions increase substantially (i.e., $2\times$ or more).

Let us now consider the case of frequency binning, with dies being divided into three bins based on their process corner labeled as slow, typical, and fast in Fig. 13. Then, each bin is margined for worst case temperature and voltage (85 °C, 10% supply drop). The frequency histogram under frequency binning for the 40 chips at 1 V is shown in Fig. 13, whose comparison with Fig. 12(a) clearly shows that some

TABLE I
COMPARISON TABLE OF EDAC APPROACHES AND iRAZOR

		Razor II JSSC'09 [4]	TDTB JSSC'09 [2]	DSTB JSSC'11 [2]	ARM JSSC'11 [6]	Razor-lite ISSCC'13 [3]	iRazor
	Type	Latch	Latch	Latch	Flip-Flop	Flip-Flop	Latch
EDAC Level	Extra # of Transistor (I)	31 (8 Shared)	15	26	28+ delay chain	8	1.46 (II)
	Possible Datapath Metastability	No	No	No	Yes	Yes	No
	FF Power Overhead	28.5%	-9%~ -13% (III)	14%~ 34%	Not Reported	2.7%	12.5%
	FF Area Overhead	Not Reported	Not Reported	Not Reported	Not Reported	33%	**4.3%**
Test Chip	Total Area Overhead	Not Reported	Not Reported	3.8%	6.9%	4.42%	13.6%
	# Razor cell/ # Total FF	121/ 826	Not Reported	12%	503/ 2976	492/ 2482	1115/ 12875
	# Gate Count	65K (IV)	123K (IV)	Not Reported	Not Reported	Not Reported	1040K
	Technology	130 nm	65 nm	45 nm	32 nm	45 nm	40 nm
	Max. Perf. (V) Improvement	Not Reported	40% @ 0.7V	28% @ 0.8V	50% @ 0.96V	54% @ 0.86V	34% @ 0.6V
	Max. Energy (VI) Improvement	35% @ 185MHz	37% @ ~3BIPS	22% @ ~0.74BIPS	52% @ 1GHz	45.4% @ ~1.2GHz	41% @ 843MHz

(I) Compared to standard 24T DFF
(II) Listed extra # of Transistor includes transistor count in local clock generation (30 Trs, Fig. 2), amortized over average # of latches per cluster and compared to 24T FF. 3 transistors added to each latch making -5 transistors compared to 24T FF.
(III) Only clock overhead compared to standard flip-flop (IV) Transistor counts divided by 4
(V) Iso-voltage comparison, [3,6] Compare PoFF with Margined Baseline; iRazor: Compare Margined Razor with Margined Baseline
(VI) Iso performance comparison

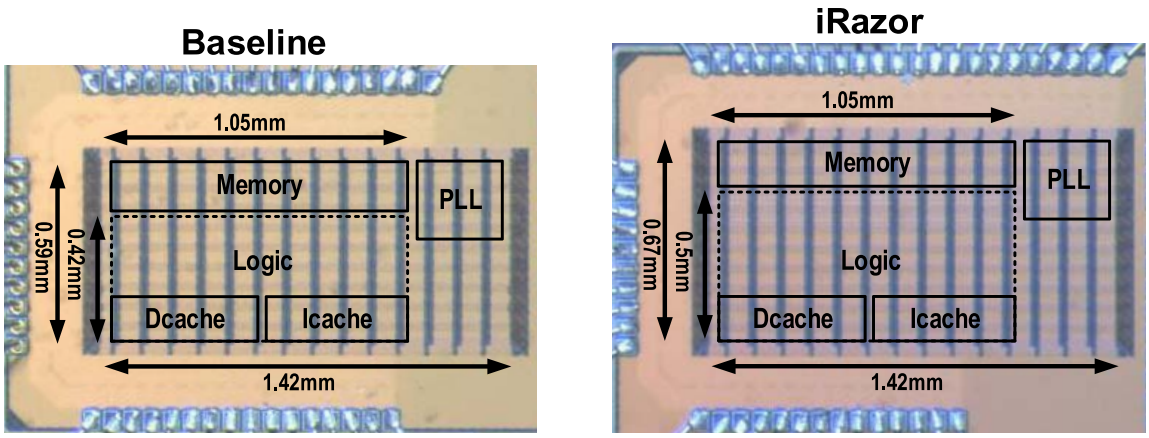


Fig. 11. Die photograph of baseline and iRazor Cortex-R4 processor in 40-nm CMOS.

margin is removed from the baseline approach. For completeness, the detailed margin histogram of the frequency binning is shown in Fig. 17(b).

As third variation-aware mainstream design approach, let us consider the "simple canary" method, under which the baseline processor is equipped with a RO used as processor

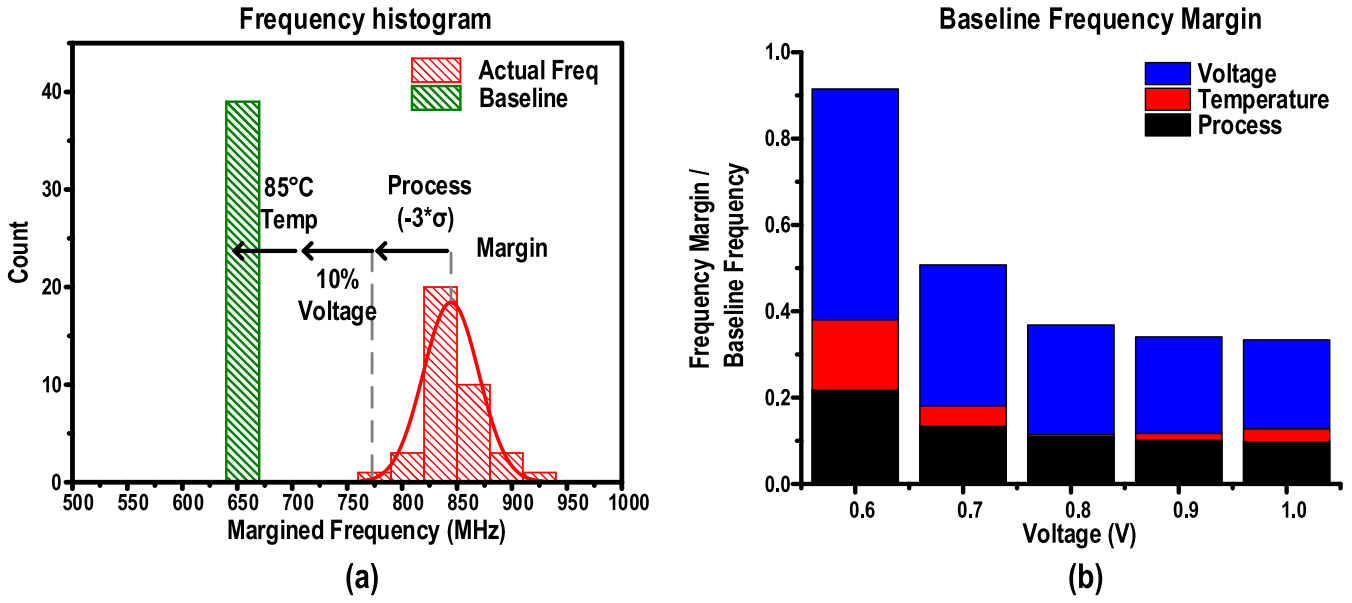


Fig. 12. (a) Detailed frequency histogram and margin analysis of baseline at 1 V. (b) Baseline frequency margin across 0.6–1 V voltage range including 10% voltage margin, 60 °C temperature margin, and three sigma process margin. The frequency margin is normalized to the average across dice of its actual frequency at nominal voltage/temperature conditions.

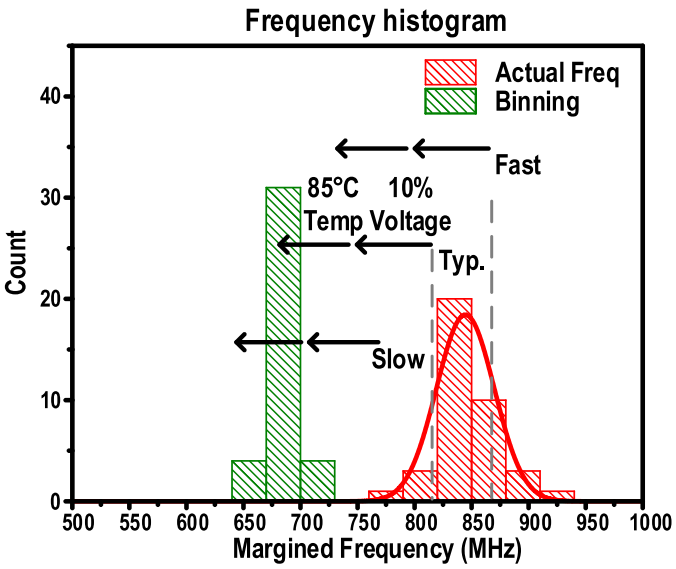


Fig. 13. Detailed frequency histogram and margin analysis of frequency binning method at 1 V.

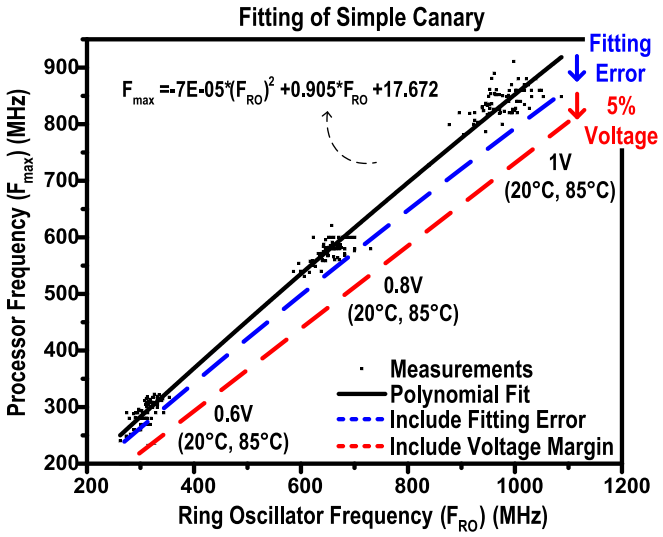


Fig. 14. Fitting of operating frequency versus RO frequency in simple canary fitting method.

frequency predictor. Fig. 14 shows measured processor frequency versus RO frequency across 0.6–1 V and 20 °C–85 °C. Exploiting the correlation between the processor frequency and the RO across voltages and temperatures in the available 40 dice in Fig. 14, the processor frequency is obtained by fitting the RO frequency data points. 2σ fitting error calculated across dies and PVT conditions is applied to evaluate the RO-processor mistracking. In addition, the fitting is de-rated by a 5% voltage margin to account for fast transient voltage excursions that the canary cannot capture. The final frequency histogram of simple canary after including fitting error and the 5% voltage margin is shown in Fig. 15. The margin histogram of the simple canary approach is also shown in Fig. 17(c).

A further comparison, a less simplistic canary approach is considered where each data point is treated as a temperature/voltage-specific canary, to suppress the margin due to temperature and voltage. This is customarily achieved by introducing on-die temperature and voltage sensors, which quantify temperature and voltage of each data point. In this approach, the linear correlation between processor and RO frequency is determined for each temperature and voltage condition. The measurements of 0.6, 0.8, and 1 V and the fitting to the RO frequency are shown in Fig. 16, where blue dots refer to 25 °C and the red ones refer to 85 °C. The linear fit is again de-rated with 5% voltage margin and 2% fitting error, but here the latter is computed only

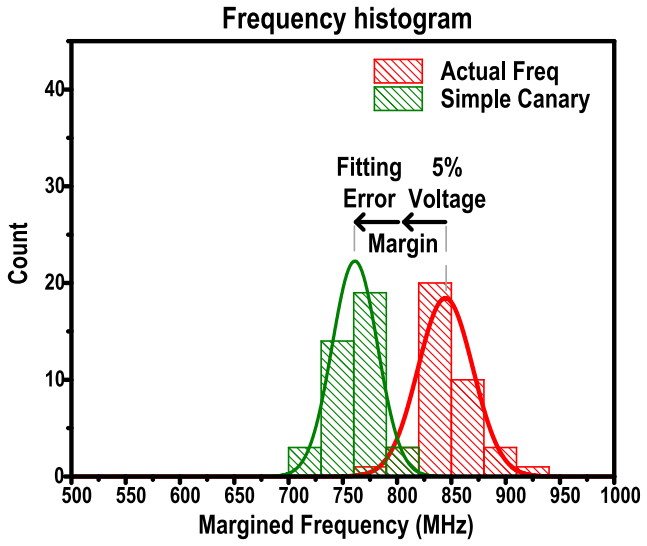


Fig. 15. Detailed frequency histogram and margin analysis of simple canary method at 1 V.

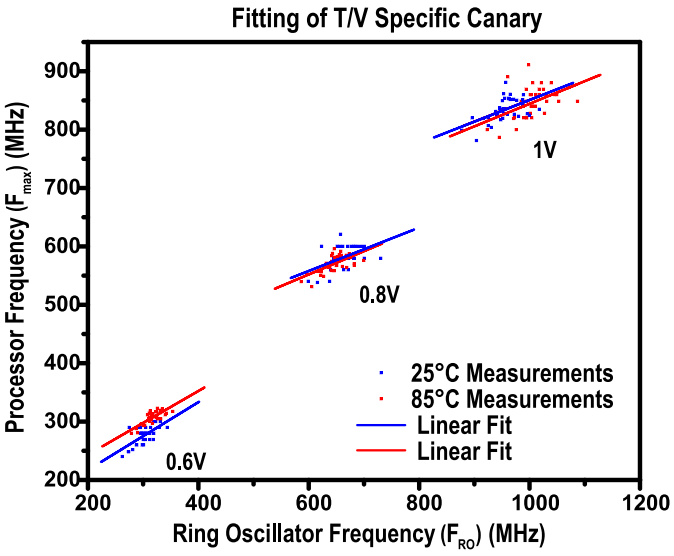


Fig. 16. Fitting of processor frequency versus RO frequency for T/V-specific canary at 25 °C and 85 °C.

across dies (i.e., without considering voltage and temperature margins). The resulting margin histogram of temperature/voltage-specific canary is shown in Fig. 17(d), which clearly shows a further margin reduction compared to the above variation-aware approaches.

VII. Experimental Results and Overall Comparison

Forty dies of the iRazor design of the ARM Cortex-R4 processor were characterized and compared to the above mainstream variation-aware design methods. The Razor point-of-first-failure (PoFF) frequency is the operating frequency beyond which errors occur (see [4] for the details on its measurement). Since iRazor is able to correct errors lying in the transparency window, it can work in a performance-optimal mode where the frequency is pushed beyond the PoFF to allow errors, which are then corrected through

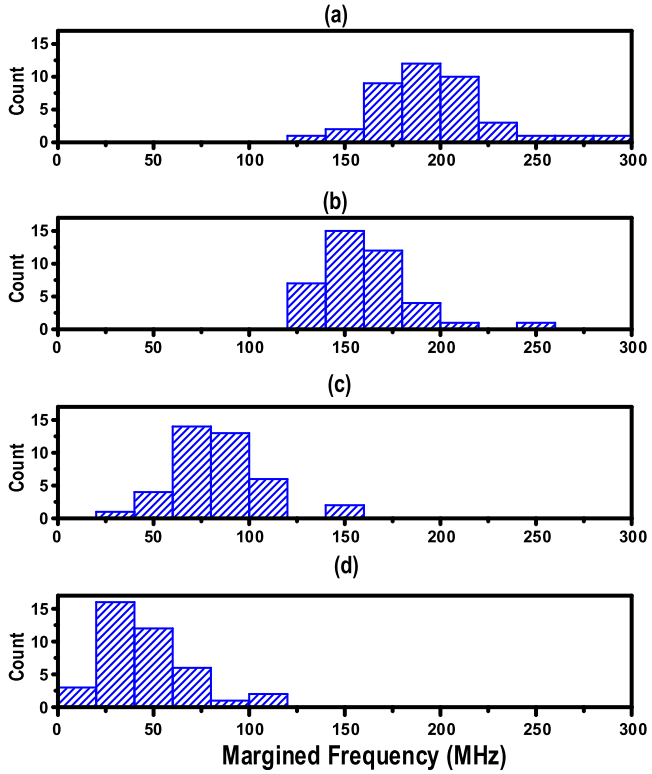


Fig. 17. Margin histogram for different methods (1 V, room temperature). (a) Baseline (process, 10% voltage, temperature). (b). Binning (10% voltage, temperature). (c) Simple canary (5% voltage, process, temperature). (d). Canary T/V spec (5% voltage, process).

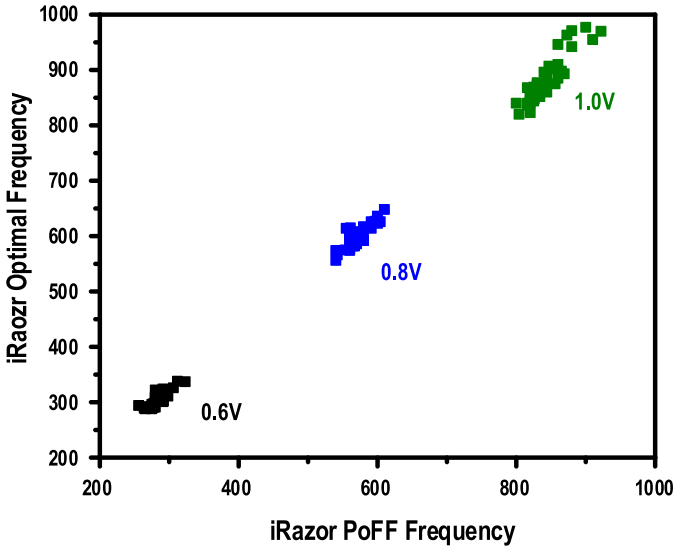


Fig. 18. iRazor frequency at PoFF versus optimal frequency across voltages.

the stalling mechanism in Section IV. In the performance-optimal mode, the resulting performance includes the effect of both the overscaled frequency and the corresponding stalling cycles due to the resulting errors. The results of the iRazor PoFF frequency and the performance-optimal frequency across 0.6, 0.8, and 1 V is shown in Fig. 18. The PoFF represents a conservative 4.4%–6.9% timing margin, compared to the performance-optimal iRazor frequency, which corresponds to a 2.4%–3% voltage margin. As a comparison, the simple

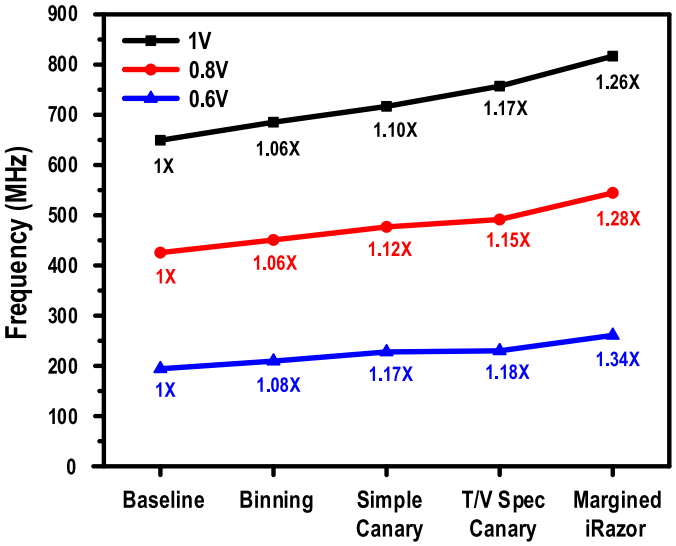


Fig. 19. Performance comparison between the margined iRazor and other methods across 0.6–1 V voltage range.

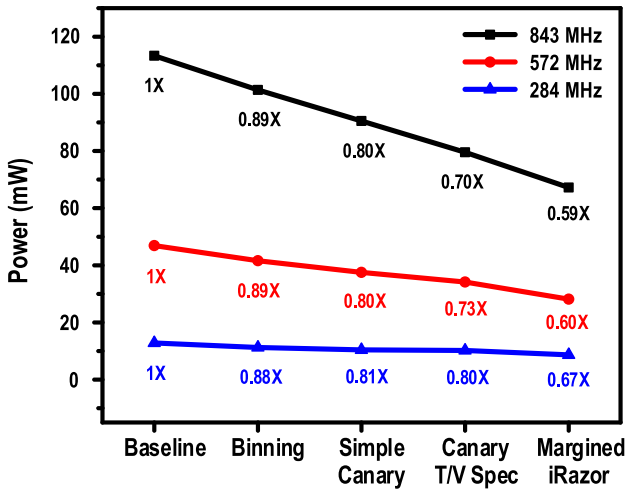


Fig. 20. Power comparison with the margined iRazor across 0.6–1 V voltage range.

canary approach adds 5% voltage margin to iRazor performance-optimal operating voltage.

The previous Razor papers assume that the detection window will surely cover all the PVT variation margins, which is, however, not always the case. Indeed, the transparency window size depends on the hold margin achieved at design time through the inserted hold buffers, hence practical constraints on the overhead due to the inserted hold buffers may prevent the designer from achieving a detection window that fully covers PVT variations. Therefore, this paper enhances the comparison by considering the margined iRazor frequency, rather than the iRazor PoFF frequency. The maximum frequency allowed by the margined iRazor and all the methods discussed in Section VI is summarized in Fig. 19. As shown in Fig. 19, a simple canary approach is about twice as effective as binning. The T/V specific canary offers ∼15%–18% performance increase over the margined baseline across 0.6–1 V, while the margined iRazor shows 26%–34% performance increase, when considering the same voltage margin as canary methods. This translates into a performance gains of 26%, 19%, and 15% compared to standard, binned, and canary-equipped versions of the Cortex-R4 processor, respectively.

The power consumption at a fixed frequency is compared in Fig. 20. In this comparison, we first select the margined baseline frequency at 0.6, 0.8, and 1 V as the target, and then we find the required supply voltage to meet this frequency using other techniques. The resulting power for each case is shown in this plot. Simple canary provides a power benefit of ∼20% over baseline across voltage, and the margined iRazor improves power by another 17%–26% over simple canary from 0.6 to 1 V.

As reported in Table I, iRazor is able to improve the performance by 34% at nominal voltage, and the energy by up to 41% when running at the same performance as the baseline design, thanks to the voltage scaling that it enables.

VIII. Conclusion

The iRazor technique has been proposed as very lightweight technique to enable EDAC, with only three additional transistors per flip-flop. An automated design flow assuring time closure has been introduced and applied to implement an ARM Cortex-R4 microprocessor in 40 nm. The resulting number of additional transistors compared to a baseline design is 1.54 transistors per flip-flop, which is the lowest reported to date. iRazor has been compared to industry-standard techniques to address variations. iRazor achieves 26%–34% performance (power) gain (33%–41%) compared to a baseline design across the 0.6- to 1-V voltage range. Power reduction becomes 17%–26% when comparing to the popular canary approach, at the cost of 13.6% area overhead.

Acknowledgment

The authors would like to thank TSMC University Shuttle Program for chip fabrication.

References

[1] D. Ernst *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. 36th IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2003, pp. 7–18.

[2] K. A. Bowman *et al.*, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, Jan. 2009.

[3] S. Kim, I. Kwon, D. Fick, M. Kim, Y.-P. Chen, and D. Sylvester, "Razor-lite: A side-channel error-detection register for timing-margin recovery in 45 nm SOI CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013, pp. 264–265.

[4] S. Das *et al.*, "RazorII: *In situ* error detection and correction for PVT and SER tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan. 2009.

[5] K. A. Bowman *et al.*, "A 45 nm resilient microprocessor core for dynamic variation tolerance," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 194–208, Jan. 2011.

[6] D. Bull, S. Das, K. Shivashankar, G. S. Dasika, K. Flautner, and 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 Circuits*, vol. 46, no. 1, pp. 18–31, Jan. 2011.

[7] S. Das *et al.*, "A self-tuning DVS processor using delay-error detection and correction," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, Apr. 2006.

[8] M. Nakai *et al.*, "Dynamic voltage and frequency management for a low-power embedded microprocessor," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 28–35, Jan. 2005.
[9] 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 Circuits*, vol. 37, no. 11, pp. 1441–1447, Nov. 2002.
[10] M. Alioto, Ed., *Enabling the Internet of Things—From Integrated Circuits to Integrated Systems*. Springer, 2017.
[11] K. A. Bowman, C. Tokunaga, T. Karnik, V. K. De, and J. W. Tschanz, "A 22 nm dynamically adaptive clock distribution for voltage droop tolerance," in *Proc. Symp. VLSI Circuits (VLIC)*, Jun. 2012, pp. 94–95.
[12] J. L. Shin *et al.*, "The next generation 64b SPARC core in a T4 SoC processor," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 82–90, Jan. 2013.
[13] A. Drake *et al.*, "A distributed critical-path timing monitor for a 65 nm high-performance microprocessor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2007, pp. 398–399.
[14] K. Hirairi *et al.*, "13% Power reduction in 16b integer unit in 40 nm CMOS by adaptive power supply voltage control with parity-based error prediction and detection (PEPD) and fully integrated digital LDO," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2012, pp. 486–488.
[15] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik, and V. De, "Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance," in *Proc. Symp. VLSI Circuits*, Kyoto, Japan, 2009, pp. 112–113.
[16] Y. Zhang *et al.*, "iRazor: 3-transistor current-based error detection and correction in an ARM Cortex-R4 processor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Jan. 2016, pp. 160–162.
[17] *ARM Cortex-R4*. Accessed: Aug. 24, 2016. [Online]. Available: https://www.arm.com/products/processors/cortex-r/cortex-r4.php
[18] P. Franco and E. J. McCluskey, "Delay testing of digital circuits by output waveform analysis," in *Proc. IEEE Int. Test Conf.*, Oct. 1991, pp. 798–807.
[19] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *Proc. 17th IEEE VLSI Test Symp.*, Apr. 1999, pp. 86–94.
[20] M. Fojtik *et al.*, "Bubble Razor: Eliminating timing margins in an ARM Cortex-M3 processor in 45 nm CMOS using architecturally independent error detection and correction," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 66–81, Jan. 2013.
[21] Y. Suzuki, K. Odagawa, and T. Abe, "Clocked CMOS calculator circuitry," *IEEE J. Solid-State Circuits*, vol. SSC-8, no. 6, pp. 462–469, Dec. 1973.

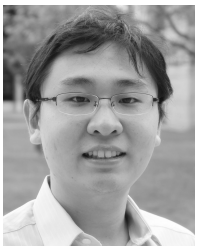
Yiqun Zhang (S'14) received the B.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2013, and the B.S. degree in electrical and computer science from Shanghai Jiaotong University, Shanghai, China, in 2013, and the M.S. degree from the University of Michigan in 2016, where she is currently pursuing the Ph.D. degree.

Her current research interests include security system, fault tolerance circuits, and error-resilient systems.

Mahmood Khayatzadeh (M'15) received the B.S. and M.S. degrees in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2000 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2013.

In 2000, he joined Emad Semiconductor Company, Tehran. From 2006 to 2008, he was with KavoshCom Research and Development Group, Tehran, where he was involved in UHF radio frequency identification reader. In 2008, he was with Delphi Automotive Systems, Singapore Design Engineering Center, Singapore. In 2013, he joined Michigan Integrated Circuit Laboratory at University of Michigan, Ann Arbor, MI, USA, as Research Investigator, where he was involved in various energy-efficient variability-tolerant VLSI designs. Since 2014, he has been a Principal Design Engineer with Oracle, Santa Clara, CA, USA. His current research interests include power-efficient, variability-tolerant VLSI circuits and systems.

Dr. Khayatzadeh has served on the technical program committees and as a reviewer for several conferences and journals.



Kaiyuan Yang (S'13–M'17) received the B.S. degree in electronics engineering from Tsinghua University, Beijing, China, in 2012, and the Ph.D. degree in electrical Engineering from the University of Michigan, Ann Arbor, MI, USA, in 2017.

He is currently an Assistant Professor with Rice University, Houston, TX, USA. His current research interests include digital and mixed-signal circuits for secure and low-power systems, hardware security, and circuit/system design with emerging devices.

Dr. Yang was a recipient of the Distinguished Paper Award at the 2016 IEEE International Symposium on Security and Privacy the Best Student Paper Award (1st place) at the 2015 IEEE International Symposium on Circuits and Systems, and the 2016 Pwnie Most Innovative Research Award Finalist. His Ph.D. work was recognized with the 2016–2017 IEEE Solid-State Circuits Society Predoctoral Achievement Award.

Mehdi Saligane received the B.S. and M.S. degrees in electrical engineering systems and control from the Ecole Polytechnique de Grenoble, Grenoble, France, in 2009, the M.S. degree in electrical engineering from Grenoble University, Grenoble, France, in 2011, and the Ph.D. degree in electrical engineering and computer science from the University of Aix-Marseille, Marseille, France, in 2016.

He was a Visiting Researcher with the Michigan Integrated Circuit Laboratory (MICL), University of Michigan, Ann Arbor, MI, USA. From 2010 to 2015, he was with STMicroelectronics Central Research and Development, Crolles, France, as a Research Engineer where he was involved in the development of new adaptive solutions and ultra-low power digital design. In 2015, he joined MICL, as a Research Investigator, and has been a Research Fellow since 2017. His current research interests include on-chip monitoring, adaptive techniques for variability tolerant designs, and near/sub-threshold energy efficient systems.

Nathaniel Pinckney received the B.S. degree from Harvey Mudd College, Claremont, CA, USA, in 2008, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2012 and 2015, respectively.

He was with Sun Microsystems' VLSI Research Group, Menlo Park, CA, USA. He is currently with NVIDIA, Austin, TX, USA. He has authored or co-authored over 30 publications in the areas of low-power VLSI design and cryptographic accelerators.

Massimo Alioto (M'01–SM'07–F'16) received the Laurea (M.Sc.) degree in electronics engineering and the Ph.D. degree in electrical engineering from the University of Catania, Catania, Italy, in 1997 and 2001, respectively.

He was an Associate Professor with the Department of Information Engineering, University of Siena, Siena, Italy. In 2013, he was a Visiting Scientist at Intel Labs–CRL, Hillsboro, OR, USA. He was a Visiting Professor at EPFL, Lausanne, Switzerland, in 2007; at BWRC–University of California, Berkeley, CA, USA, from 2009 to 2011; at the University of Michigan, Ann Arbor, MI, USA, from 2011 to 2012. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, where he leads the Green IC Group and is the Director of the Integrated Circuits and Embedded Systems area. He has authored or co-authored more than 240 publications in journals (80+, mostly IEEE Transactions) and conference proceedings. One of them is the second most downloaded TCAS-I paper in 2013. He has co-authored three books: *Enabling the Internet of Things—From Circuits to Systems* (Springer, 2017); *Flip-Flop Design in Nanometer CMOS–From High Speed to Low Energy* (Springer, 2015); and *Model and Design of Bipolar and MOS Current-Mode Logic: CML, ECL and SCL Digital Circuits* (Springer, 2005). His current research interests include ultra-low power VLSI circuits, self-powered and wireless nodes, near-threshold circuits for green computing, widely energy-scalable VLSI circuits, circuit techniques for emerging technologies, and hardware-level security, among the others.

Dr. Alioto was a Distinguished Lecturer of the IEEE Circuits and Systems Society, from 2009 to 2010, for which he was also a member of the Board of Governors from 2015 to 2017, and the Chair of the VLSI Systems and ApplicationsTechnical Committee from 2010 to 2012. In the last five years, he has given 50+ invited talks in top universities and leading semiconductor companies. He currently serves as an Associate Editor-in-Chief of the IEEE TRANSACTIONS ON VLSI SYSTEMS, and served as Guest Editor of various journal special issues. He also serves or has served as an Associate Editor of a number of journals (e.g., the IEEE TRANSACTIONS ON VLSI SYSTEMS, the *ACM Transactions on Design Automation of Electronic Systems*, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–PART I. He was the Technical Program Chair (ICECS, VARI, NEWCAS, ICM, PRIME, and SOCC) and the Track Chair in a number of conferences (ICCD, ISCAS, ICECS, VLSI-SoC, APCCAS, and ICM).

David Blaauw (M'94–SM'07–F'12) received the B.S. degree in physics and computer science from Duke University, Durham, NC, USA, in 1986, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1991.

He was with Motorola, Inc. in Austin, TX, USA, as the Manager of the High Performance Design Technology Group. Since 2001, he has been with the faculty of the University of Michigan, where he is currently a Professor. He has authored or co-authored over 500 papers, and holds 60 patents. His research interests included adaptive computing to reduce margins and improve energy efficiency. He has investigated adaptive computing to reduce margins and improve energy efficiency using a new approach he pioneered, called Razor. He has extensively researched in ultra-low-power computing using subthreshold computing and analog circuits for millimeter sensor systems and for high-end servers, his research group and collaborators introduced so-called near-threshold computing, which has become a common concept in semiconductor design. This work led to a complete sensor node design with record low-power consumption, which was selected by the MIT Technology Review as one of the year's most significant innovations. His current research interests include cognitive computing using analog, in-memory neural-networks.

Dr. Blaauw was the General Chair of the IEEE International Symposium on Low Power, the Technical Program Chair for the ACM/IEEE Design Automation Conference, and serves on the IEEE International Solid-State Circuits Conference's technical program committee. He was a recipient of Motorola Innovation Award, the Richard Newton GSRC Industrial Impact Award and IEEE Micro annual Top-Picks award for a new approach he pioneered called Razor, and the 2016 SIA-SRC faculty award for lifetime research contributions to the U.S. semiconductor industry. He has also received numerous best paper awards and nominations.

Dennis Sylvester (S'95–M'00–SM'04–F'11) received the Ph.D. degree in electrical engineering from the University of California, Berkeley, CA, USA, in 1999.

He is a Professor of electrical engineering and computer science with the University of Michigan, Ann Arbor, MA, USA, and the Director of the Michigan Integrated Circuits Laboratory, a group of 10 faculty and more than 70 graduate students. He has held research staff positions with the Advanced Technology Group, Synopsys, Mountain View, CA, USA, Hewlett-Packard Laboratories, Palo Alto, CA, USA, and visiting professorships at the National University of Singapore, Singapore, and Nanyang Technological University, Singapore. He is the Co-Founder of Ambiq Micro, Austin, TX, USA, a fabless semiconductor company developing ultralow-power mixed-signal solutions for compact wireless devices. He has authored or co-authored over 375 articles along with one book and several book chapters. He holds 20 U.S. patents. His current research interests include the design of millimeter-scale computing systems and energy-efficient near-threshold computing.

Dr. Sylvester serves on the Technical Program Committee of the IEEE International Solid-State Circuits Conference and previously served on the Executive Committee of the ACM/IEEE Design Automation Conference. He also serves as a Consultant and Technical Advisory Board Member for electronic design automation and semiconductor firms in his research areas. He has served as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, and the Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II. He was a recipient of the NSF CAREER Award, the Beatrice Winner Award at ISSCC, the IBM Faculty Award, the SRC Inventor Recognition Award, the ACM SIGDA Outstanding New Faculty Award, the University of Michigan Henry Russel Award for distinguished scholarship, and eight best paper awards and nominations. His dissertation was recognized with the David J. Sakrison Memorial Prize as the most outstanding research in the UC-Berkeley EECS Department.