Variation-Tolerant, Ultra-Low-Voltage Microprocessor With a Low-Overhead, Within-a-Cycle In-Situ Timing-Error Detection and Correction Technique

Seongjong Kim and Mingoo Seok

***Abstract*—This paper presents a design approach for upgrading the resiliency of ultra-low-voltage (ULV) microprocessors through a voltage-scalable and low-overhead in-situ error detection and correction (EDAC) technique. Particular efforts are made to overcome the poor voltage scalability and area/energy/throughput overhead of the existing EDAC techniques when applied to ULV designs. The 16 bit microprocessor employing the proposed EDAC and dynamic voltage scaling schemes is demonstrated in a 65 nm. The microprocessor can automatically modulate V_{DD} based on timing error flags across static/slow variations and in-situ detect and correct the timing errors from fast dynamic variations, virtually eliminating timing and voltage margins. At a typical process/voltage/temperature corner, the proposed design improves the minimum energy consumption by 42% with 140 mV additional voltage scaling, as compared to the baseline design. At the same throughput (80 MHz), the proposed design consumes 38% less energy than the baseline operating at its minimum energy point. At the same energy consumption, the proposed design achieves 2.3× higher throughput than the baseline design. The area overhead of the proposed design is 8.3%.**

***Index Terms*—In-situ error detection and correction, microprocessor, near-threshold, sub-threshold, variation tolerance, voltage scaling.**

I. Introduction

ULTRA-LOW-VOLTAGE (ULV) operation is a key technique to achieve extremely energy efficient digital computing hardware by scaling supply voltage (V_{DD}) to the level near or below transistor threshold voltage (V_{TH}) [1]. ULV operation can improve computing energy efficiency, particularly beneficial to prolong battery lifetime of ranges of energy-constrained systems such as biomedical implants, environment sensors, and micro-robots [2], [3].

One of the most critical challenges in designing ULV computing hardware is large delay variability [4]. As shown in Fig. 1, at the ULV regime, circuit delay can radically change across typical and worst-case process, voltage, and temperature (PVT) conditions. At 0.4 V operation, for example, the worst-case clock frequency (F_{CLK}) of a 50-FO4 (Fan-Out of 4) inverter chain is 12× slower than the typical-case. The conventional practice to ensure correct operation at the worst-case condition is to operate a processor at 12× slower F_{CLK} or at 160 mV higher V_{DD}. Given that the worst-case condition rarely occurs, those large timing or voltage margins are wasteful and also limit the useful voltage scaling [5].

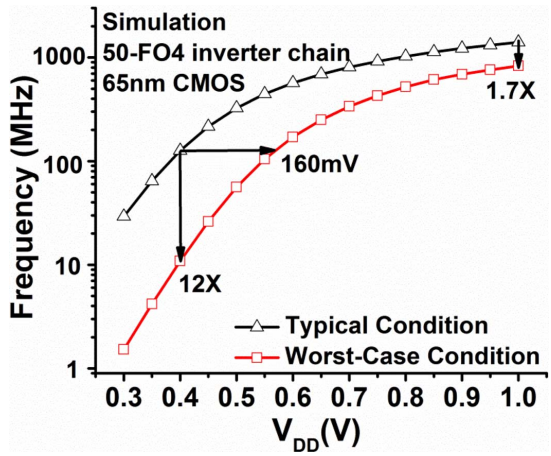


Fig. 1. Simulation using 50-FO4 long inverter chains in a 65 nm CMOS shows that 12× frequency margin or 160 mV supply voltage margin are needed for worst-case condition at 0.4 V.

Several adaptive techniques [6]–[18] have been proposed to reduce the delay and voltage margins. One class of techniques is to use the circuits that replicate the critical paths of a target design. The delay of the replicated circuits can predict timing errors, and if needed inform a dynamic voltage frequency scaling (DVFS) controller to modulate V_{DD} and F_{CLK} [15]–[18]. This, so-called canary technique, can track global and slow-changing variations such as inter-die process variation and ambient temperature fluctuation. However, the canary technique cannot remove the margins for fast dynamic variations due to the limitation in the response time of DVFS systems. It also cannot remove the margins for local (random) variations since the replicated circuits cannot capture them. Those fast-dynamic and local variations include intra-die

Manuscript received December 27, 2014; revised February 22, 2015; accepted March 20, 2015. This work was supported by research grants from the DARPA PERFECT Program and the Catalyst Foundation.

The authors are with Columbia University, New York, NY 10027 USA (e-mail sk3667@columbia.edu; mgseok@ee.columbia.edu).

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.2015.2418713

process variation, local dynamic V_{DD} drop, capacitive coupling, and local hot spots.

In order to remove the margins for fast-dynamic and local variation, in-situ error detection and correction (EDAC) techniques have been proposed [6]–[14]. In these techniques, error-detecting registers are inserted in critical paths to *in-situ* detect and correct timing violations via hardware. This can eliminate the margins for fast-dynamic and local variation. In addition, the error statistics from those registers can also inform a DVFS controller to modulate V_{DD} and F_{CLK}, which can eliminate the margins for global and slowly changing variations. The EDAC, coupled with DVFS schemes, can therefore virtually eliminate all the timing margins.

The conventional EDAC techniques [6]–[14], however, cannot be directly applied to ULV designs for the following critical problems. 1) A significantly larger number of registers need to be replaced with bulky error-detecting ones. Note that the error detecting registers typically have 8 to 44 more transistors than the conventional one per register [6]–[14]. 2) A large amount of short-path padding is needed, incurring large area and energy overhead. 3) The conventional error-detecting register circuits become unreliable. 4) Timing error rates can increase, degrading energy efficiency and throughput.

In this paper, in order to mitigate those problems, we propose a voltage-scalable, low-overhead, and within-a cycle EDAC technique, which consists of the following three sub-techniques. 1) We devise a sparse error detection strategy where errors are detected in every *several* pipeline stages rather than every stage without compromising detection coverage. The benefit is a substantially less number of error-detecting registers inserted, and the elimination of short-path padding requirement. 2) We design an error-detecting latch circuit that can operate reliably at very low voltage. 3) We develop an error-correction scheme where errors are detected and corrected within a cycle, without stalling pipelines. This eliminates the control overhead of the existing multi-cycle detection and correction process.

Based on the proposed EDAC technique, a variation-tolerant Resilient-Processor (R-Processor) is designed and fabricated in a 65 nm CMOS. At a typical PVT corner, R-Processor can reduce the minimum energy consumption (E_{min}) by 42% at a 140 mV lower V_{DD}, as compared to the baseline processor operating with the worst-case voltage margins. At the same $F_{CLK} = 80$ MHz where the baseline processor achieves its E_{min}, R-Processor consumes 38% less energy. Finally, R-Processor can have 2.3× higher throughput at the same energy consumption of the baseline operating at its energy-optimum supply voltage (V_{OPT}). The area overhead of the proposed EDAC technique in R-Processor is only 8.3%.

II. The Conventional EDAC Techniques in ULV Design

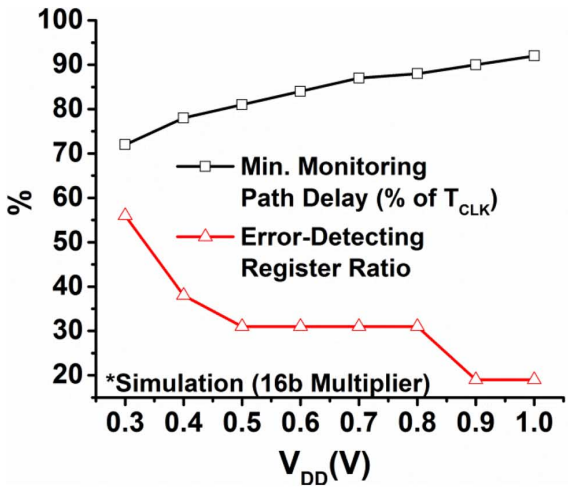
A. Conventional Flip-Flop-Based EDAC Techniques

Several papers have proposed EDAC techniques for edge-triggered pipelines operating at super-V_{TH} V_{DD}s [6]–[13]. In these techniques, the flip-flops which receive signals from critical and near-critical paths are replaced with the error-detecting flip-flops (EDFF). The flip-flops to be replaced are identified during design time using timing analysis tools. Those works have reported the EDFF replacement rate (= the number of EDFFs/the number of original flip-flops) in the range of 7% to 20% (see Table V). However, note that some of the designs have unbalanced pipeline stages which can allow no EDFF insertion into some short stages [12]. The replacement rates can significantly increase for more balanced pipelines.

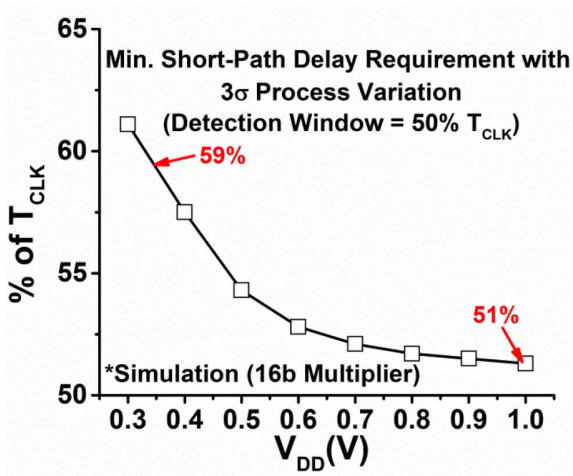
In ULV operation, as the circuit delay is highly sensitive to PVT variations, the EDFF replacement rate becomes larger. In order to analyze the growing overhead, we apply the EDAC technique (similar to [6], [7]) onto a single-stage 16 bit multiplier in a 65 nm CMOS. As shown in Fig. 2(a), whereas at $V_{DD} = 1$ V EDFFs only need to monitor the paths whose delays are longer than 92% of T_{CLK}, at $V_{DD} = 0.35$ V the paths whose delays are longer than 76% of T_{CLK} need to be monitored. As a result, at 0.35 V, 44% of regular flip-flops have to be replaced with EDFFs. The replacement rate at 1 V is 19% at 1 V. In our experiment, this high EDFF replacement rate can cause 12% area and 12% energy overhead (including clock trees and buffers) when the Razor-I flip-flops [6], [7] are used.

Short-path padding can cause another large overhead when the conventional techniques are used for ULV pipelines. The EDFFs interpret it as timing errors if they observe their inputs making transitions during the error-detection window (typically the clock-high phase). However, the correct signals, propagating through short paths, can arrive to the inputs of EDFFs during the error-detection window, and this can cause EDFFs to make false error detection. False error detection does not affect the functionality of pipelines but can significantly waste energy and throughput due to the unnecessary exercise of error-correction processes (e.g., instruction replay in [8]–[12]). In order to avoid this problem, the existing EDAC techniques apply so-called short-path padding where those short paths are extended with delay elements (e.g., buffers) such that none of the correct signals arrive at the inputs of EDFFs during the error-detection window. This creates a design trade-off between the detection window which dictates the degree of tolerance to dynamic and local variation and the short path constraint dictating the overhead of added delay buffers. To avoid excessive short path padding, some of previous works have proposed the use of $<$50% duty cycle in clock. This reduces the size of detection window but also mitigate the overhead of short-path padding [8], [10]–[12]. In ULV operation, however, the growing variability demands large detection window, rendering this approach less favorable. It is also not trivial to reliably distribute non-50% duty-cycle clock.

To make things worse, in ULV pipelines, short paths need more padding and some other paths that do not require short-path padding at nominal V_{DD} may additionally require padding, all due to the large variability. This significantly increases the overhead to implement the conventional EDAC technique. We again investigate the impact of ULV on short-path padding requirement using the 16 bit multiplier. As shown in Fig. 2(b), at 0.35 V, all the paths arriving at the EDFFs need to be longer than



(a)



(b)

Fig. 2. (a) At lower V_{DD}s, the larger delay variability makes more paths to violate timing requirement, demanding more error-detecting registers to be inserted. (b) At near/sub-threshold regime, *more* paths need *longer* padding due to larger variability. For example, all the paths need to be longer than 59% of T_{CLK} at 0.35 V while 51% at 1 V.

59% of T_{CLK} when considering 3σ delay variation incurred by local process variation whereas 51% at 1 V. Due to the wall of slacks in delay-power optimized design [19], the short path padding at 0.35 V causes 2× area and 69% energy overhead. This high overhead can be a serious limiting factor to use the existing flip-flop-based EDAC techniques in ULV design.

B. Conventional Two-Phase Latch Based EDAC Techniques

EDAC techniques for two-phase latch-based pipelines have been recently proposed [14]. Similar to the flip-flop-based EDAC techniques, the latches that receive data from near-critical and critical paths are replaced with error-detecting latches (EDL). The input change in the transparent phase of latches (i.e., cycle borrowing window) is interpreted as timing error. Since two consecutive latch stages cannot be transparent at the same clock high or low phase, no short-path padding is required.

The EDAC techniques for two-phase latch-based pipelines, however, have their own challenges. First, two-phase latch pipelines have inherently higher sequential overhead than flip-flop pipelines. We find that it increases the sequential area by 77%, the clock switching capacitance by 2.5×, the total area by 13%, and the total energy consumption by 15% in a 16 bit multiplier. Specifically, 16 flip-flops are replaced with 39 latches. In the used standard cell library, a pair of latches have 45% larger area and 2× more clock load than one flip-flop. While the overhead is considerable, it is still worthwhile to note that the inherent overhead of latch-based pipelines is significantly smaller than the overhead of short path padding in flip-flop-based pipeline when we apply the EDAC techniques at 0.35 V.

In addition, the EDAC techniques for ULV operation can suffer from a high replacement rate of EDLs. First of all, latch-based pipelines have roughly twice more critical paths than flip-flop-based pipeline since it has 2× more stages. Furthermore, since the length of one latch stage is 2× shorter than that of a flip-flop stage, each stage has less of averaging effects across logic gates, which can worsen delay variability induced by local variations [20]. Monte Carlo simulations with local process variation in a 16 bit multiplier show that one latch stage has 1.7× worse variability (defined as σ/μ of delay variation) than one flip-flop stage at 0.35 V. As a result, it requires replacing 23 out of 39 latches with EDLs, increasing total area and energy consumption by 24% and 31%, respectively, without counting the inherent overhead of latch-based pipelines.

In order to understand the overhead in larger-scale circuits, we design 3-stage pipeline test circuits (PTC) using the baseline flip-flop sequencing, the conventional flip-flop-based EDAC technique [6], [7], the conventional latch-based EDAC technique [14], and the proposed EDAC technique (see. Section III-A). Each PTC consists of three 16 bit multipliers and operates at 0.35 V. As shown in Fig. 3, the flip-flop-based EDAC technique causes 2.1× area and 81% energy overhead over the baseline. The latch-based EDAC technique causes 40% area and 47% energy overhead over the baseline due to the inherent overhead from latch sequencing and a higher rate of EDL replacement. The large overhead of both of the conventional techniques can significantly undermine the energy and throughput benefits that EDAC and DVFS can achieve by removing margins.

C. Error-Detecting Flip-Flop and Latch

One of the common methods to detect timing errors in EDFF and EDL circuits is double-sampling [6], [7], [10], [11], [13], [14]. Fig. 4(a) shows the schematic of a conventional double-sampling-based EDFF. The data input (D) is sampled by the main positive edge-triggered flip-flop and also by the transparent-high shadow latch. During the clock high phase (i.e., detection window), the output of the shadow latch (Q_{shadow}) directly show the data input while the output of the main flip-flop

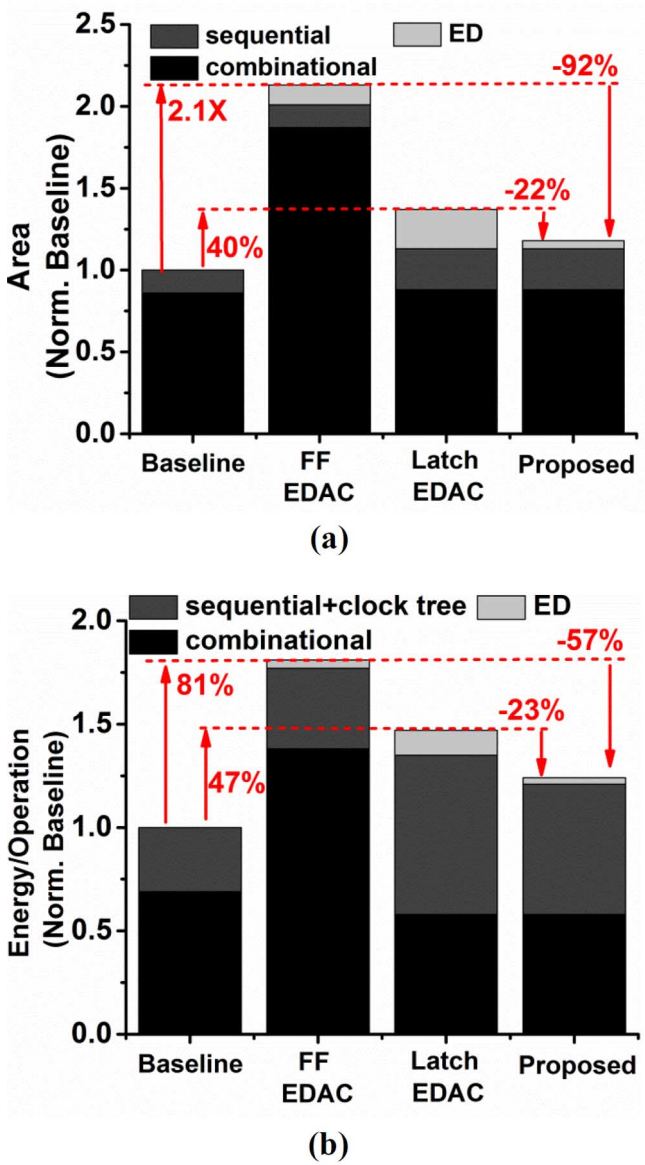


Fig. 3. (a) The experiments using TPCs show that the conventional flip-flop and latch-based EDAC techniques incur 2.1× and 40% area overhead, respectively, as compared to the baseline. The proposed EDAC technique achieves 22 to 92% smaller area overhead than the conventional EDAC techniques. (b) The energy overhead of conventional EDAC techniques are 81% and 47%, respectively. The proposed technique can reduce energy overhead by 23 to 57% over the conventional techniques.

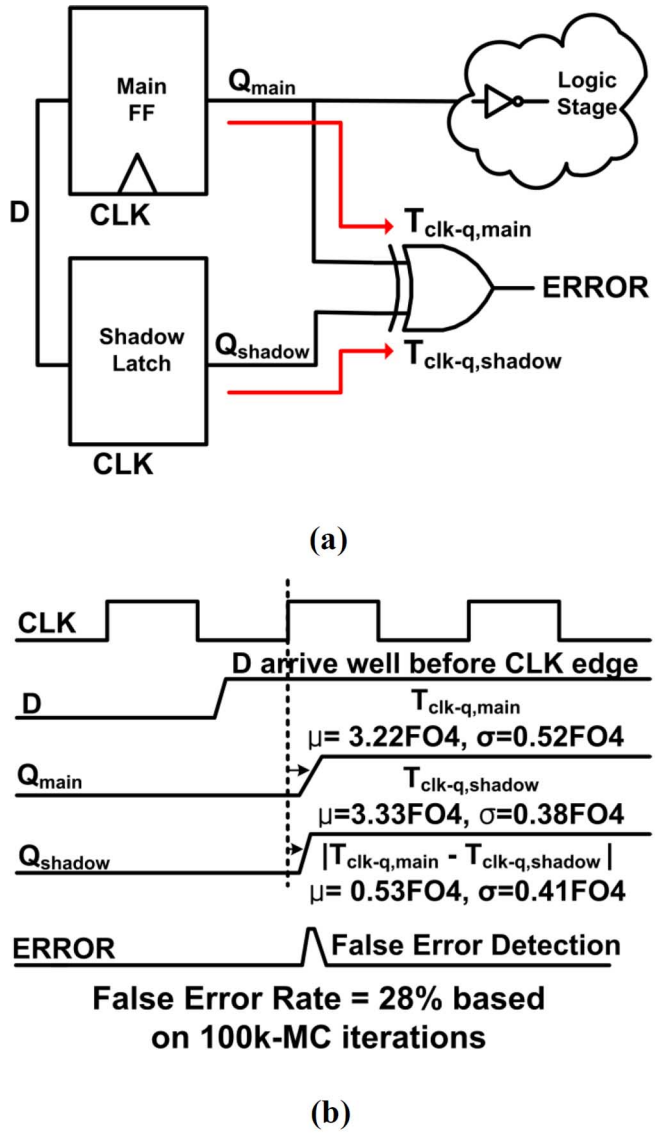


Fig. 4. (a) The conventional double-sampling method suffers from false error detection due to clk-to-q delay mismatch between main and shadow elements at low voltage. (b) The 3σ clk-to-q delay mismatch over 100 k Monte Carlo simulation with random process variation at 0.35 V is 1.8-FO4 delay, which causes the false error rate of 28%.

(Q_{main}) is the input captured at the rising clock edge. Discrepancy between Q_{shadow} and Q_{main} suggests that the input arrives after rising clock edge, i.e., timing violation.

The double-sampling technique becomes unreliable at ULV operation. Particularly, the clk-to-q delay mismatch between the main flip-flop and the shadow latch can become large. As shown in Fig. 4(b), although the data arrives well before the clock rising edge (i.e., error-free operation), the delay difference between Q_{main} and Q_{shadow} can cause a glitch in the ERROR signal, leading to a false error detection. We perform 100 k Monte Carlo simulations with process variation at 0.35 V, and find that the 3σ clk-to-q delay difference is 1.8 FO4 delays, which cause the false error detection rate of 28%.

This large false error rate can be masked in the conventional EDAC techniques having multi-cycle correction schemes, as the error signals are sampled by another registers at the next clock edge. In the proposed within-a-cycle correction scheme (Section III-C), however, the error signals cannot be sampled by another registers due to the stringent timing constraints. This can unnecessarily trigger a correction process, and thereby wasting energy and throughput. While other EDFF circuits without double-sampling have been proposed [8]–[11], those circuits rely on non-static gates and largely skewed transistor sizes, which can become unreliable at ULV regime.

D. Error Correction Scheme

Upon detection, EDAC techniques can correct the errors. Instruction replay is one of the common methods [8]–[13]. It flushes the pipeline and replays the instructions that just caused

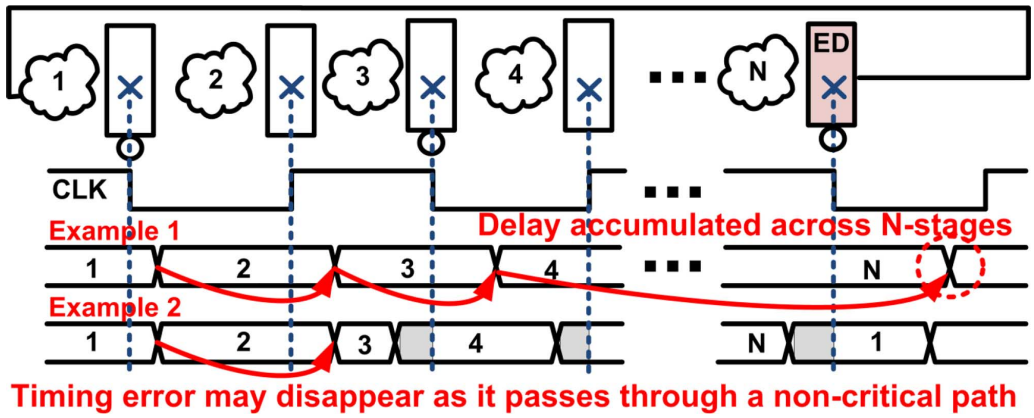


Fig. 5. The proposed sparse error detection technique inserts EDLs every N stage, instead of every stage. The delay increase (timing error) can propagate to next stage via cycle borrowing and may disappear as it passes through a non-critical path or be cycle-borrowed again. We insert error-detecting latches before the delay accumulation can exceed the cycle borrowing window.

errors at a safer clock frequency (e.g., half the original F_{CLK}). The instruction-replay-based correction scheme, however, requires architecture modification and consumes up to 28 cycles per correction, which can severely degrade throughput and energy efficiency [11]. The authors in [11] have proposed a correction method based on multiple-issue instruction replay. In this method, upon detecting errors, the instruction that just caused errors is issued multiple times at the original F_{CLK}. As it requires no change in F_{CLK}, it incurs 15 cycle penalty per correction. In the existing latch-based EDAC technique [14], a correction scheme based on local stalling has been proposed. When error is detected, the pipeline takes one extra cycle during which it sends clock gating signals to the subsequent stages.

III. Proposed EDAC Techniques

A. Sparse Error Detection

In order to minimize the replacement rate of EDFFs or EDLs, we propose a sparse error detection technique where error-detecting registers are sparsely inserted across multiple pipeline stages. The technique is based on two-phase latch-based pipelines for eliminating the short-path padding requirement. In the proposed technique, we do not detect the delay increase (potential timing error) generated in every latch stage. Instead we let it be cycle-borrowed to the subsequent stages. As shown in Fig. 5, the cycle-borrowed delay increase may disappear in the next stage while propagating through non-critical paths, or is cycle-borrowed again to the following stage. The EDL is inserted just before the accumulation of the delay increases across several stages is expected to exceed the size of cycle borrowing window. This sparse detection technique can significantly reduce the overhead incurred by the high EDL replacement rate, even in the pipeline whose stages are balanced. In addition, it can reduce the number of actual errors and the cost involved to correct them.

Sparse insertion reduces the maximum detection capability per stage, however, the technique attempts to maximize the detection window utilization if every stage does not require the largest-possible detection window of 50% of T_{CLK}. This can be the case when EDAC is combined with DVFS. In this combination, the global and static/slow-dynamic variations can be tracked as the DVFS makes a pipeline to operate at points of first failure (PoFF). Only the remaining local (random) and fast-dynamic variations need to be detected by EDAC. When we use the 6σ worst-case delay variation from Monte Carlo simulation with local process variation as a proxy for these fast-dynamic and local variations, we find that the detection window required in each stage is significantly less than 50% of T_{CLK}. Therefore, we can share the detection window across multiple stages. For this reason, in the proposed scheme, most of the cycle-borrowing window is reserved for dynamic-variations induced delay variability, by operating the pipeline at PoFFs.

The benefit of only latch-based pipelines in ULV regime has been demonstrated in some of the previous works [20], [23], [24]. However, the use of only cycle-borrowing cannot completely remove the worst-case margin across ranges of PVT variations as the variability is severe.

The optimal number of latch stages (N_{OPT}) that can be skipped without having EDLs while still maintaining the full detection capability across pipeline stages is a function of V_{DD}, process technology, and stage length. In a 65 nm CMOS, we find the N_{OPT} across different V_{DD}s and stage lengths for inverter chains via simulation. The chain has 20-consecutive latch stages where per-latch-stage length is 25-FO4 delays. We set T_{CLK} to be 50-FO4 delays with 50% duty cycle such that no cycle borrowing is needed under the typical PVT condition. We perform the Monte Carlo simulation with local process variation to find the required detection window. In this experiment, we use the 6σ worst-case delay variation from local process variation as a proxy for all the fast-dynamic and local variations that should be handled by EDAC techniques. While this is less accurate than detailed simulation and measurement, we find that our assumption well agrees with the findings in some of the previous works [6]–[14], [25]. The estimation of the impact of fast-dynamic and local variations can be pursued independently for more accurately guiding the proposed sparse insertion technique.

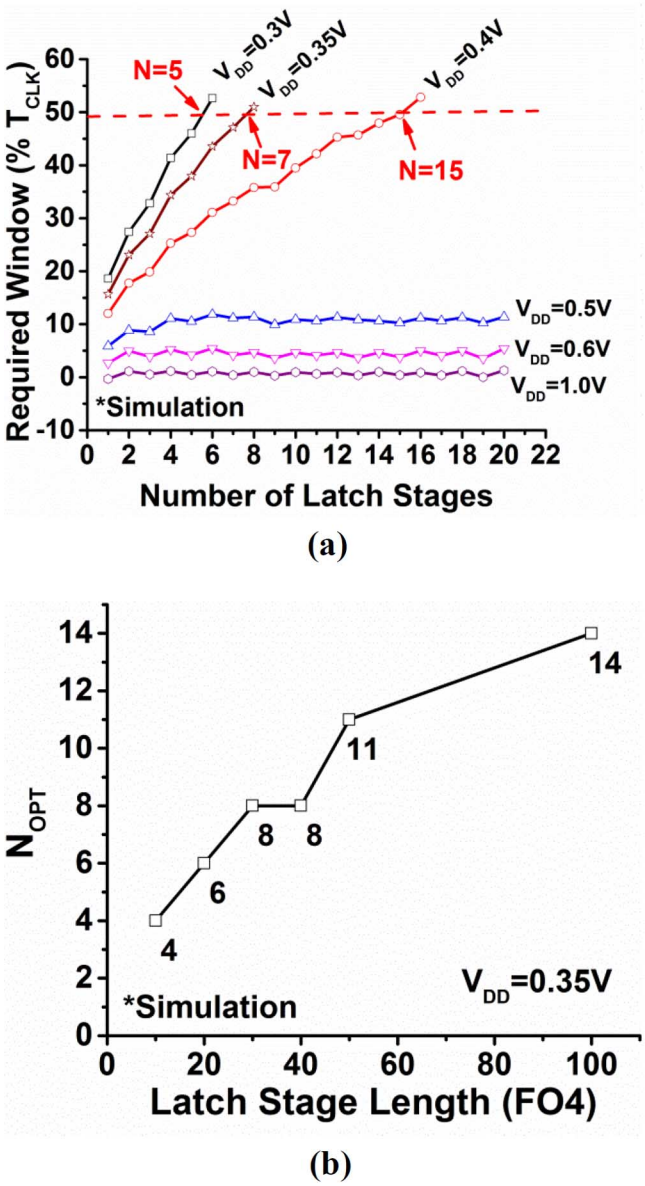


Fig. 6. (a) The maximum number of latch stages (N) that can be skipped before inserting EDLs, defined as N_{OPT}, is a function of V_{DD}, technology, and stage length. Simulation shows more frequent insertion, i.e., smaller N_{OPT}, is needed at lower V_{DD}s. (b) N_{opt} is simulated across different per-latch-stage lengths of pipelines at 0.35 V. A longer per-latch-stage length increases N_{opt} due to more averaging effects of random variation across stages.

As shown in Fig. 6(a), at 0.35 V we can insert EDL in every seven latch stages (i.e., $N_{OPT} = 7$). Fig. 6(b) shows the found N_{OPT} across different per-latch-stage lengths at $V_{DD} = 0.35$ V. The longer per-stage length allows the use of larger N_{OPT} via the averaging effect across more number of gates. Note that, at V_{DD} higher than 0.5 V, the N_{OPT} can be larger than 20 stages due to the smaller delay variability.

Again, we evaluate the overhead of the proposed sparse insertion technique in the context of the PTC. Since $N_{OPT} = 7$ at 0.35 V, the PTC with the proposed techniques replaces only one latch stage among total of 6 latch stages (3 flip-flop stages). As shown in Fig. 3, it incurs 18% area and 24% energy overhead over the baseline. As compared to the conventional EDAC

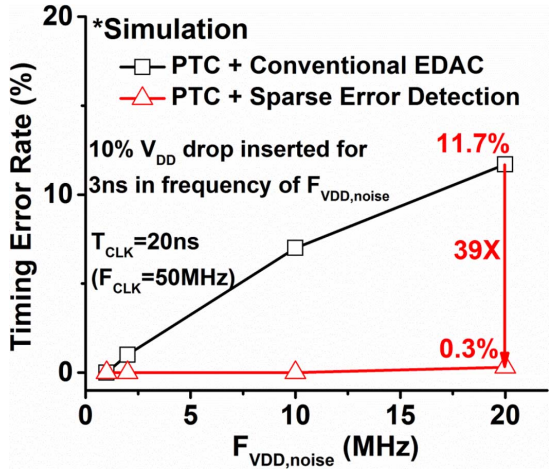


Fig. 7. Timing error rates of the PTC designs with the conventional and proposed EDAC techniques. An artificial square-pulse-shape V_{DD} noise with the -10% V_{DD} magnitude and a duration of 3 ns is injected and repeated at the frequency of $F_{VDD,noise}$. At $F_{VDD,noise} = 20$ MHz, the proposed technique has the 39X less error rate.

techniques, it achieves 22% to 92% smaller area and 23% to 57% lower energy consumption. We also fabricate and measure some of the PTCs. As shown in Table IV, the PTC with the proposed EDAC technique achieves 13%–23% energy savings by removing the worst-case margin over the baseline. The measurement results of the PTC design using the conventional EDAC techniques are not included as it could not achieve any meaningful improvement over the baseline due to the excessive false error detection.

Finally, the proposed sparse error detection technique can substantially reduce error rates. We inject an artificial square-pulse-shape V_{DD} noise with an amplitude of -10% of V_{DD} and a duration of 3 ns, and repeat the injection at a frequency of $F_{VDD,noise}$ into the PTCs with the conventional and proposed EDAC techniques. In this simulation the PTCs operate at the 50 MHz at its PoFF. As shown in Fig. 7, the proposed techniques can achieve a 39$\times$ smaller error rate at $F_{VDD,noise} = 20$ MHz.

B. Voltage-Scalable Error Detecting Latch Circuits

In order to apply EDAC techniques in any ULV designs, EDFF and EDL circuits need to be voltage-scalable, particularly free of false error detection (see Section II-C for details). For this purpose, as shown in Fig. 8(a), we propose voltage-scalable EDL circuits which use only the shadow latch for error detection. The main and shadow latches are transparent high and low, respectively. The shadow latch uses the side-channel timing error detection method adapted from [12] and is designed such that no glitches can be generated in the error detection node though a domino-circuit-like mechanism having precharge and evaluation phases.

More specifically, Fig. 8(b) shows the operation of the EDL During the clock-low phase, the virtual nodes, VVDD and VVSS, in the shadow latch is pre-charged high and pre-discharged low via P1 and N2, respectively. During this phase, the tri-state inverter at the input of the shadow latch is active and the node DN is the inversion of the data input (D). When the

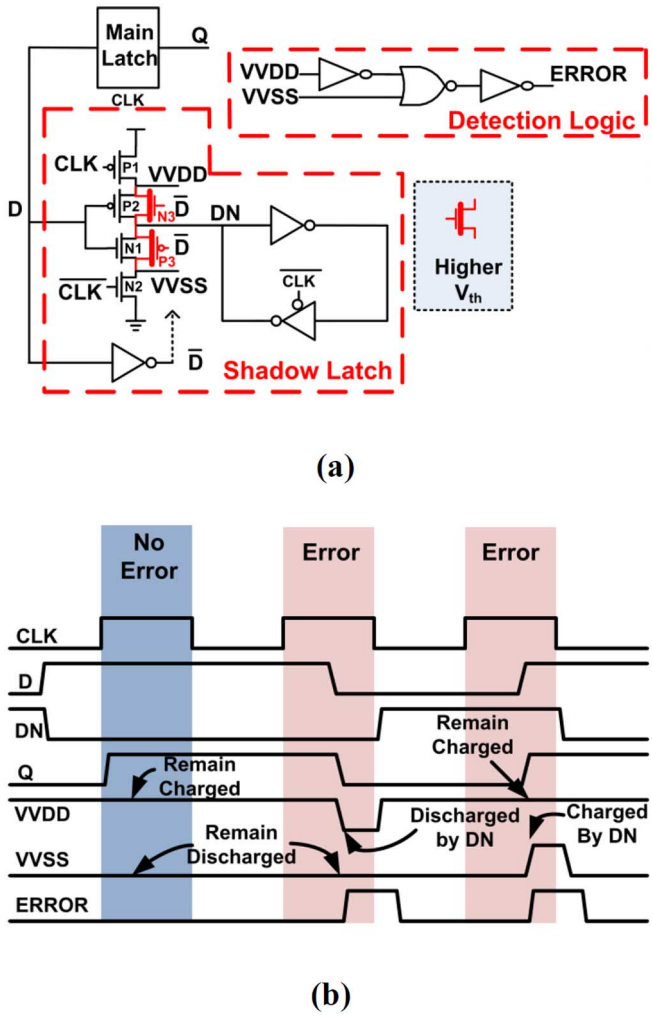


Fig. 8. (a) The schematic and (b) the operational waveform of the proposed EDL. It uses the side-channel error detection method to avoid the clk-to-q mismatch problem. It is also optimized to upgrade voltage scalability down to 0.3 V.

clock becomes high, the devices P1 and N2 are turned off and the state of the node DN is maintained through the back-to-back inverters in the shadow latch. At the absence of errors (i.e., the input D does not change during a clock-high phase), the potentials of the VVDD and VVSS remain high and low, respectively. When D = 1 and DN = 0, for example, the transistor N1 and the feedback inverter keeps the VVSS to be low. The node VVDD is floating, but the V_{GS} of the P2 becomes negative, significantly cutting leakage and helping maintaining the potential of the VVDD. However, when error happens, i.e., the input D changes during a clock-high-phase, either the VVDD becomes low or the VVSS becomes high, which sets the ERROR signal to be high through simple detection logic. In the error detecting mechanism, the states of VVDD/VVSS have no glitch during error-free operation regardless of the state of the new data, which is also confirmed via 100 k Monte Carlo simulation with process variations at 0.3 V.

Differently from the prior design for super-V_{TH} operation [12], we augment the devices N1 and P2 with the devices N3 and P3. In the prior design, the node VVDD is discharged through

TABLE I
COMPARISONS OF THE CONVENTIONAL LATCH AND THE PROPOSED EDL CIRCUITS AT 0.35 V

	Conventional Latch	Proposed EDL
CLK-Q Delay	3-FO4	3.7-FO4 (25%)
D-Q Delay	3.6-FO4	4.1-FO4 (14%)
D-DETECT Delay	-	3.5-FO4
Switching Energy	0.68fJ	1.44fJ (2.1×)
Leakage Power	7nW	20nW (2.9×)
# of Transistor	19	43 (2.3×)
Area	6.84μm^2	25.2μm^2 (3.7×)

the PMOS (P2) and the node VVSS is charged through the NMOS (N1). In ULV operation, however, the threshold-voltage drop and the resultant noise-margin and delay penalties are intolerable. A downside is that the added devices N3 and P3 can contribute leakage current which can affect the floating virtual rails (i.e., VVDD or VVSS) and cause false error detection. Particularly, the devices N3 and P3 do not experience the negative V_{GS} as do the devices P2 and N1 (the N3 and P3 have $V_{GS} = 0$). In order to minimize the leakage, we selectively use higher V_{TH} transistors for the N3 and P3 at about 30% longer detection delay (i.e., delay from D to DETECT). Note that, the VVDD and VVSS nodes are refreshed every cycle, which can relax leakage reduction requirement. Again, we perform 100 k Monte Carlo simulations for the proposed EDL circuits and find that the circuits cause no leakage-induced false error detection down to $V_{DD} = 0.3$ V and the worst-case $F_{CLK} > 10$ KHz.

Table I summarizes the comparisons of the proposed EDL and the regular latch circuits at 0.35 V. The D-Q delay is increased by 14% due to the extra capacitance from driving the shadow latch. The CLK-Q delay is increased by 25% as the internal clock buffers of main latch were shared with shadow latch to reduce the energy overhead from extra clock loading at the cost of increased internal clock delay. Although the overhead of the individual EDL is considerable, the proposed sparse error detection strategy can amortize the overall overhead by enabling the EDL insertion only in a single pipeline stage.

C. Non-Stall Error Correction

In the conventional EDAC techniques, the process to detect and correct an error can take multiple clock cycles, undermining throughput and energy efficiency [8]–[13]. Furthermore, the detection and correction process needs to be controlled across modules connected together for multiple clock cycles. This requires architectural modifications and the consequent hardware overhead [8]–[13].

In order to reduce the overhead associated with the detection and correction process, we propose a highly localized technique, called the non-stall error correction method. This can detect and correct error within a single cycle, obviating the need to control detection and correction processes across pipelines

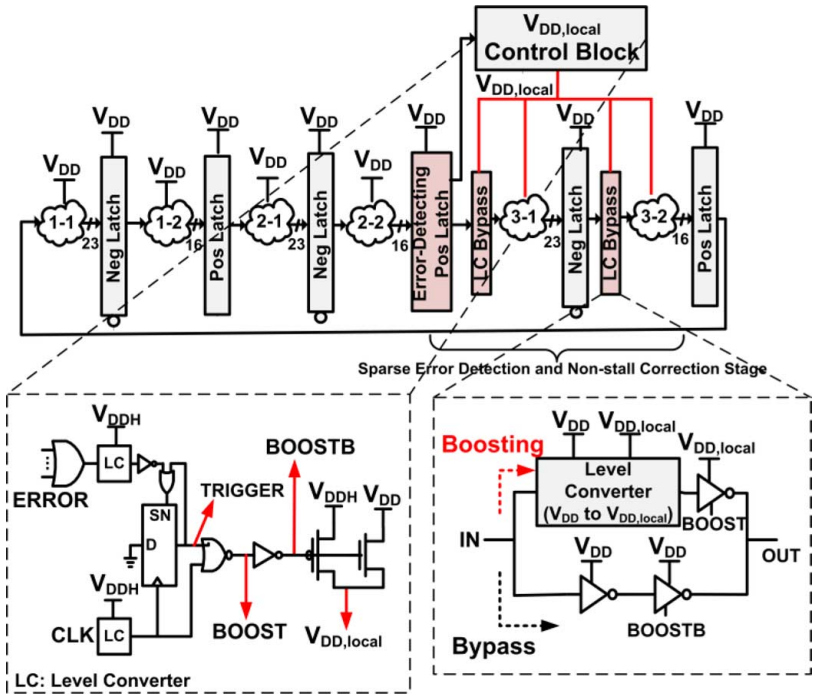
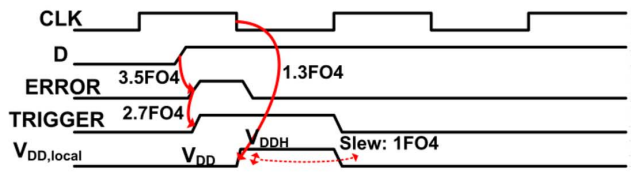


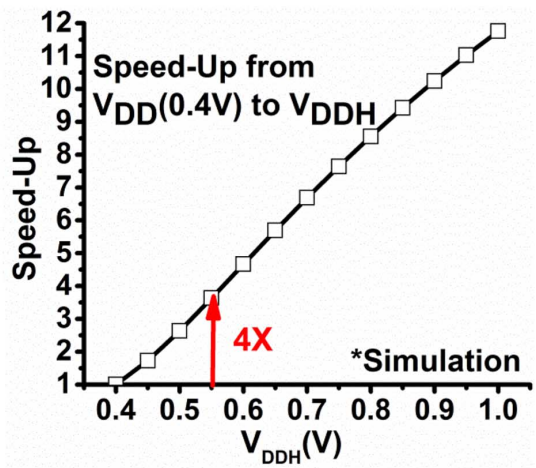
Fig. 9. We propose a non-stall error correction scheme which utilizes local and temporal V_{DD} boosting. When timing error is flagged in the detection stage and while the late arriving signals still propagate via cycle borrowing, the $V_{DD,local}$-control block changes the supply voltage ($V_{DD,local}$) of the next stage (correction stage) to higher V_{DD}(V_{DDH}) to accelerate signal propagation. The latches in the correction stage are not boosted to avoid accidental state loss. Level converters are bypassed at the absence of errors.

and cycles. Fig. 9 shows the block diagram of the proposed correction technique and Fig. 10(a) shows the waveforms during correction. As shown in Fig. 9, the ERROR signals from the EDLs in the detection stage are collected via an OR tree by the $V_{DD,local}$ Control Block. This block then controls the supply voltage of the following correction stage ($V_{DD,local}$). While the timing error is detected, the late arriving correct data still propagate to the correction stage via cycle borrowing. $V_{DD,local}$ is then boosted from nominal (V_{DD}) to boosting voltage (V_{DDH}) by switching the header devices. The headers are sized to enable sufficient switching speed (i.e., switching slew of 1-FO4 delay). The boosting is only performed during the negative phase of the clock for isolating the correction stage from other adjacent stages. Also, the latches in the correction stage are not boosted to avoid accidental state loss. Level-converters are instead inserted after the latches which can be bypassed at the absence of errors for avoiding the delay overhead.

In this project we set V_{DDH} to be 0.55 V to allow a speed-up of 4× in the correction stage (i.e., the parts denoted as 3–1 and 3–2 in the Fig. 9) from the pipeline operating at the nominal V_{DD} of 0.4 V (Fig. 10(b)). The required speed-up is estimated as following. In the targeted pipeline having the stage length of 50 FO4 delays, the correction scheme need to detect the worst-case timing violation (i.e., signals arrive at the EDL input at 25 FO4 delays after the rising clock edge) and produce the error-free result before the following rising clock edge. Under this worst-case scenario, the time budget for boosting V_{DD} and re-computing the results is 25 FO4 delays. As shown in Fig. 10(a), it takes 7.5 FO4 delays ($= 3.5 + 2.7 + 1.3$) to



(a)



(b)

Fig. 10. (a) Operational waveform of the non-stall correction technique. The $V_{DD,local}$ is boosted during the negative clock phase for isolating the correction stage from the next stage. The supply voltage headers are sized to meet the 1-FO4 boosting slew. (b) Boosting the voltage at ULV can allow sufficient speed-up to correct error without stalling pipelines. For example, boosting from 0.4 V to 0.55 V can give 4× speed-up which is sufficient to produce error-free results in less than a half cycle.

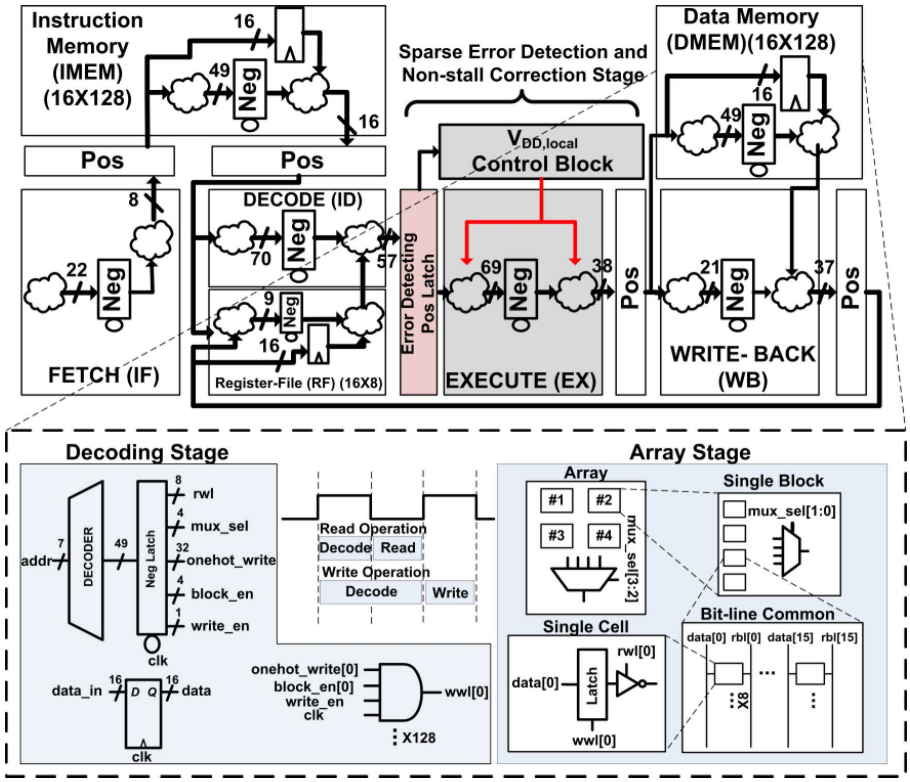


Fig. 11. R-Processor: a 16-bit 5-stage microprocessor employing the proposed EDAC techniques. The memory blocks (DMEM, IMEM, and RF) are also pipelined with 2-phase latches to continue cycle-borrowing across the entire pipeline.

boost V_{DD}. If we can achieve 4× shorter circuit delay via the boosting, re-computation takes 12.5 FO4 delays ($= 50/4$). The sum of those two delays is 20 FO4 delays, which is smaller than the aforementioned time budget of 25 FO4 delays. For the pipelines having shorter T_{CLK}s, we can increase the V_{DDH} for a larger speed-up to a certain extent, but have to pay a higher energy cost per correction.

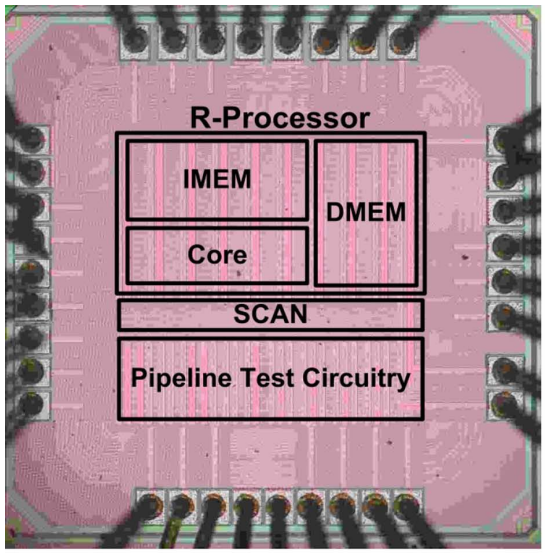
IV. R-Processor Design and Implementation

As shown in Fig. 11, the proposed EDAC techniques are applied to the design of a 5-stage, 16-bit microprocessor, which we call R-Processor. We first replace flip-flops with two-phase latches using industrial retiming tools (Section II-B for details). R-Processor is retimed with T_{CLK} of 50 FO4 delays where per-latch-stage is 25-FO4 long. As shown in Fig. 6(a), for $V_{DD} = 0.4$ V, N_{OPT} is found to be 15 (equivalent to about 7 flip-flop stages) for the per-latch-stage length of 25 FO4 delays. Therefore, we replace the positive-phase latches only in the ID stage with the proposed EDLs, resulting in very low EDL replacement rate of 13% (57 out of 445). The proposed non-stall correction technique is applied in the EX stage. EX stage is chosen for avoiding boosting memory circuits (DMEM, IMEM, and RF). Timing errors in other stages (e.g., IF, EX, MEM, and WB) can disappear when cycle-borrowed to subsequent non-critical paths, or be carried on to the end of ID stage for detection and be corrected in the EX stage. Therefore, the seemingly localized detection can actually cover the entire pipeline of the R-Processor.

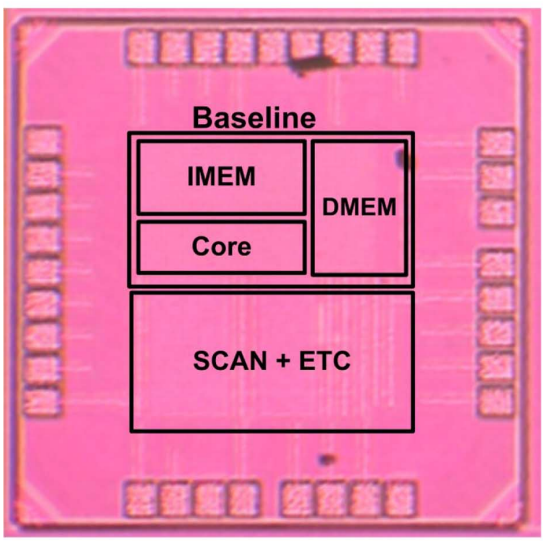
In order to allow cycle borrowing to continue across memory structures, all the memory blocks (DMEM, IMEM, and RF) are also pipelined with 2-phase latches. For example, in DMEM, the negative latches are inserted between address decoders and arrays of bitcells. For read operation, the first half clock cycle (positive clock phase) is used to decode the address and the next half clock cycle (negative clock phase) is used for accessing bitcells. In order to guarantee sufficient writing time even in the worst-case delay increase, writing to arrays is delayed by half clock cycle. Note that this delaying does not interfere with the next instruction that might access memory since the next instruction still decodes the address during the first half cycle.

The use of 2-phase latches in memory blocks allows the timing errors occurred in them, like other pipeline stages of the R-Processor, to either disappear when propagating through the subsequent non-critical paths or be eventually detected at the end of ID stage and corrected in the EX stage. To reliably explore the proposed EDAC techniques at ULV operation, we use the memory cell based on a regular 12-transistor (12T) latch with the single-ended and static readout path. A single V_{DD} as low as 0.26 V is used for both pipeline and memories. The use of the compact bit-cells is under investigation.

Fig. 12 shows the die photograph of the R-Processor and the baseline processor with flip-flop sequencing and no EDAC technique, both of which are fabricated in a 65 nm CMOS technology. The absolute layout area overhead of R-Processor was



(a)



(b)

Fig. 12. Die photograph of (a) the R-Processor, (b) the baseline processor.

1.8% compared to baseline chip. When considering the utilization statistics from automatic placement and routing (APR) tool, the area overhead was 8.3%.

V. Measurement Results

We measure and compare the baseline processor having no adaptive techniques and the R-Processor having the proposed EDAC and DVFS combination. The operating condition of the baseline is set for guaranteeing correct computation across all PVT conditions. As the baseline is forced to operate with a single V_{DD}/F_{CLK} pair, the chosen pair has a large amount of margins. On the other hand, the R-Processor can dynamically choose the optimal V_{DD}/F_{CLK} pair for the given PVT condition. In this process, the EDAC technique can inform the DVFS controller to ensure the R-Processor operate at its PoFFs. Note that the existing EDAC works have demonstrated the aggressively scaling of V_{DD}/F_{CLK} beyond PoFFs for additional improvement [6]–[9], [12], [14], whereas the R-Processor limits such scaling for maintaining a sufficient size of error detection window. The savings by operating beyond PoFFs are also small as the R-Processor already operates in very low V_{DD} (Fig. 15).

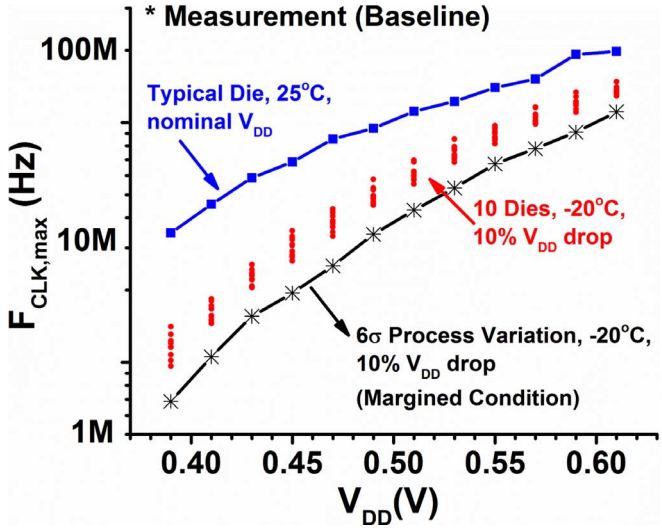


Fig. 13. The $F_{CLK,max}$ of the baseline processor is measured based on the worst-case PVT condition. First, the $F_{CLK,max}$ at the worst-case voltage and temperature condition (10% V_{DD} drop and $-20°C$) is measured over 10 chips. Then, in order to account for process variation we find the 6σ worst-case $F_{CLK,max}$ out of the 10 chip measurements, which is used for the margined F_{CLK} of the baseline processor. If considering the variation across wafers and lots, the worst-case $F_{CLK,max}$ can be even worse than our estimation.

Based on the conventional worst-case design practice we determine the V_{DD} and F_{CLK} of the baseline processor. As shown in Fig. 13, at each V_{DD}, the maximum F_{CLK} for correct operation is measured for 10 dies under the worst-case voltage and temperature condition. We assume 10% V_{DD} drop and the temperature of $-20°C$ as the worst-case condition. To account for the worst-case process variation, we use the -6σ value of the measured maximum F_{CLK} for the margined F_{CLK}. We did not use the worst-case measured performance since the number of the samples (i.e., 10 dies) is too small to represent the worst-case process variation.

Fig. 14 shows the measured energy per cycle of a typical chip of the baseline processor operating at the margined F_{CLK} across V_{DD}s at 25°C. The E_{min} is found to be 5.29 pJ when operating at $F_{CLK} = 80$ MHz and $V_{DD} = 0.57$ V. Fig. 14 also shows the energy consumption per cycle of a typical chip of the R-Processor. Across V_{DD}s, F_{CLK} is selected at the point of the first failure (PoFF). The E_{min} for R-Processor is measured to be 3.13 pJ at $F_{CLK} = 60$ MHz and $V_{DD} = 0.43$ V [Fig. 14, (1)]. The E_{min} of the R-Processor was 42% lower than baseline chip. At the same F_{CLK} of 80 MHz, the R-Processor can operate at 110 mV lower V_{DD}, achieving 38% energy reduction [Fig. 14, (2)]. Also, the R-Processor can have $\sim 2.3\times$ higher throughputs ($F_{CLK} = 180$ MHz) while consuming the same energy with the baseline [Fig. 14, (3)].

Fig. 15 shows the measured energy consumption and error rate of the typical chip of the R-Processor across V_{DD}s at

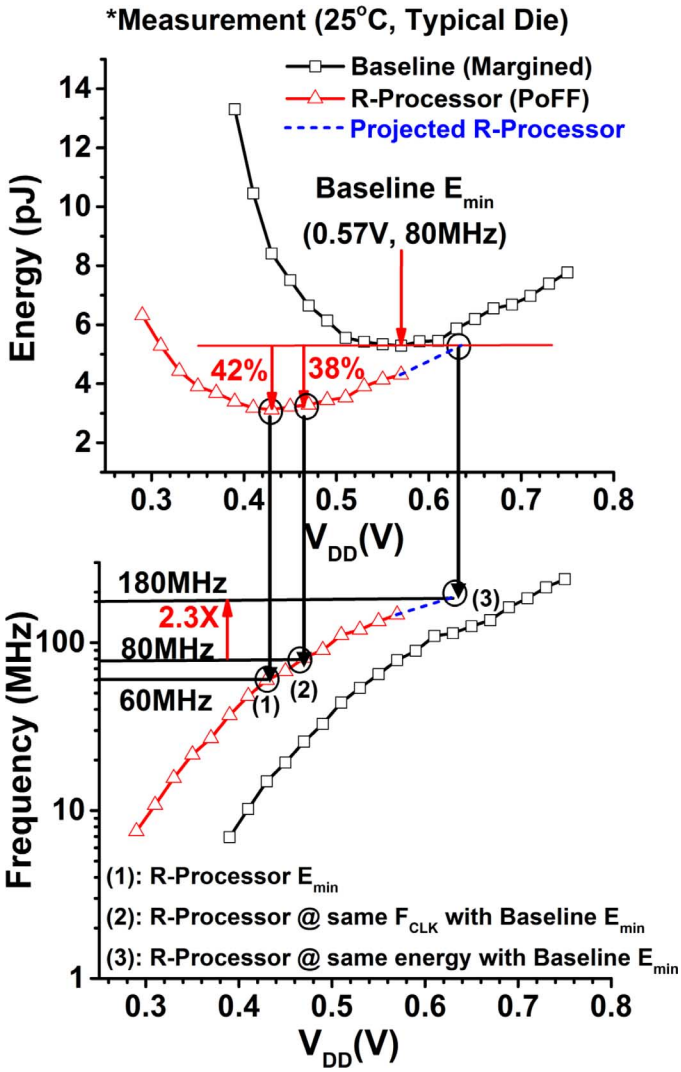


Fig. 14. The R-Processor achieves energy efficiency and performance improvement over the baseline design. (1) The R-Processor can scale V_{opt} by 140 mV as compared to the baseline, where the R-Processor consumes 42% smaller energy per cycle at $F_{CLK} = 60$ MHz; (2) At the same performance (80 MHz that the baseline achieves at its V_{opt}), the R-Processor exhibit 38% lower energy consumption; (3) At the same energy consumption, the R-Processor is estimated to be 2.3× faster than the baseline.

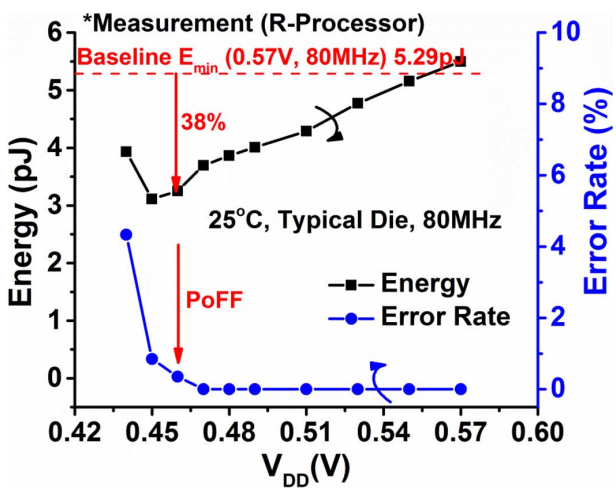


Fig. 15. The energy savings and error rates of R-Processor. The R-Processor can use 110 mV lower V_{DD} and consume 38% less energy when it reaches the point of the first failure (PoFF), i.e., detecting and correcting the first error.

TABLE II
THE SUMMARY OF THE R-PROCESSOR AND THE BASELINE PROCESSOR CHIPS IN THE TYPICAL PVT CORNER. UTILIZATION IS DEFINED AS TOTAL AREA DIVIDED BY GATE AREA

	R-Processor	Baseline
Technology	CMOS 65nm	CMOS 65nm
Clock Frequency	80MHz	80MHz
Supply Voltage	0.46V (R-Processor PoFF)	0.57V (V_{opt})
Sequencing	2-phase latch with EDAC	Flip-flop
E/cycle	3.25pJ	5.29pJ (E_{min})
EDL insertion rate	13% (57/445)	N/A
Total area	96706μm^2	94996μm^2
Total area × Utilization	87916μm^2	81200μm^2
Area Overhead	8.3%	0%
Energy overhead at the same F_{CLK} and V_{DD}	4%	0%

TABLE III
SUMMARY BASELINE PROCESSOR AND R-PROCESSOR AT SLOW, TYPICAL, AND FAST CORNER

	Baseline Processor			R-Processor (@ PoFF)		
	V_{DD}	F_{CLK}	Energy	V_{DD}	F_{CLK}	Energy (savings)
Slow Die, -20°C	0.57V	80MHz	3.27pJ	0.51V	80MHz	2.79pJ (33%)
Typical Die, 25°C	0.57V	80MHz	5.29pJ	0.46V	80MHz	3.25pJ (38%)
Fast Die, 70°C	0.57V	80MHz	15.7pJ	0.43V	80MHz	7.66pJ (51%)

TABLE IV
SUMMARY OF THE BASELINE PTC AND THE PTC WITH THE PROPOSED EDAC TECHNIQUE IN THE SLOW, TYPICAL, AND FAST CORNER

	Baseline PTC			PTC + Proposed EDAC (@ PoFF)		
	V_{DD}	F_{CLK}	Energy	V_{DD}	F_{CLK}	Energy (savings)
Slow Die	0.39V	10MHz	0.84pJ	0.30V	10MHz	0.73pJ (13%)
Typical Die	0.39V	10MHz	0.86pJ	0.29V	10MHz	0.71pJ (17%)
Fast Die	0.39V	10MHz	0.94pJ	0.28V	10MHz	0.72pJ (23%)

$F_{CLK} = 80$ MHz and 25°C. As the V_{DD} is lowered, it reaches to the PoFF at 0.46 V and the error rate increases sharply beyond the PoFF. We can further scale the V_{DD} down to 0.45 V to achieve more energy savings (41% reduction over the baseline). However, it reduces the size of detection window. Also, energy savings by operating the R-Processor beyond PoFFs are small since R-Processor is already in very low voltage regime. Table II summarizes the measurement results of the R-Processor and the baseline processor.

Table III summarizes the measured energy consumption of the baseline and the R-Processor running at the PoFFs at three different process and temperature corners. Energy savings increase to 51% in the fast corner (fast process, 70°C) since the amount of wasted margins of the baseline processor becomes larger. In the slow corner (slow process, −20°C), on the other

TABLE V
COMPARISON OF R-PROCESSOR AND PREVIOUS EDAC WORKS

	[7]	[8]	[9]	[12]	[14]	[11] EDS	[13]	This work
Technology	0.18µm	0.13µm	65nm	45nm SOI	45nm SOI	45nm	45nm SOI	65nm
Target V_{DD}	Super-Vt	Super-Vt	Super-Vt	Super-Vt	Super-Vt	Super-Vt	Super/Near-Vt	Near/Sub-Vt
Architecture	64 bit, 5 stage, Alpha Processor	64 bit, 7 stage, Alpha Processor	32 bit, 6 stage, ARM Processor	64 bit, 7 stage, Alpha Processor	ARM Cortex-M3	32 bit, 7 stage, LEON3	10-lane Synctium-I SIMD	16bit, 5 stage
ICache, Dcache	8KB, 8KB	-	2KB, 2KB	2048KB, 2048KB	16KB (I$+D$)	16KB, 16KB	-	256B, 256B
Sequencing	Flip-Flop	Flip-Flop	Flip-Flop	Flip-Flop	2-Phase Latch	Flip-Flop	Flip-Flop	2-Phase Latch
Duty-cycle Control	No	Yes	No	Yes	No	Yes	No	No
ED Insertion Rate	(7.4%)207/2801	(15%)121/826	(17%)503/2976	(20%)492/2482	100%	12%	-	13%
Typical Energy Saving	50% (at 0.1% error rate)	33% (PoFF)	43% (PoFF)	45.4%(Max)	54% (PoFF) 62% (Max)	22% (iso-throughput, Max)	-	38% (iso-throughput, PoFF)
Typical Throughput Saving	-	-	-	-	67% (PoFF) 103% (Max)	16% (1V, Max) 28% (0.8V, Max) 28% (0.6V, Max) 41% (iso-energy, Max)	9.8% (1V) 11.2% (0.53V) (at 0.5% error rate)	2.3× (iso-energy, PoFF)
Power Overhead	ED: 0.31% Padding: 2.6%	ED: 1.2% Padding: 1.3%	ED: 5.7% Padding: 1.3% Stab: 2.4%	ED: 0.3% Padding: N/A	-	Total: 0.9%	-	Total: 4%
Area Overhead	N/A	N/A	6.9% of the total area	4.42% of the datapath	103% of the total area	3.8% of the total area	-	8.3% of the total area

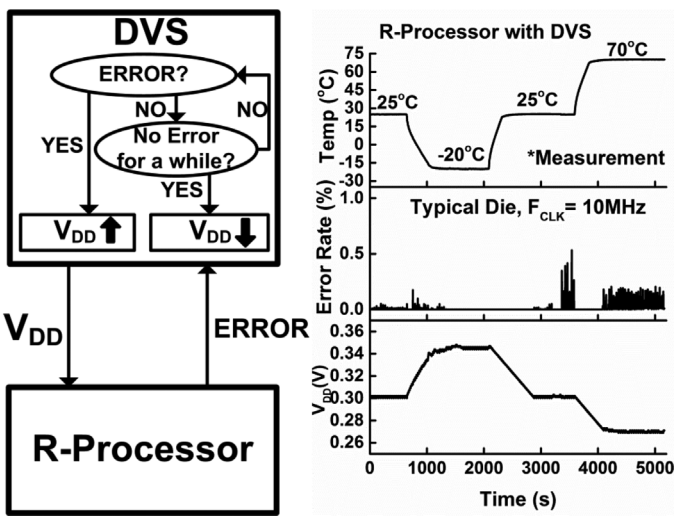


Fig. 16. Experiment results of R-Processor running at 10 MHz with an off-chip DVS system while ambient temperature is varying from -20°C to 70°C. The R-Processor can operate well down to the deep sub-threshold regime of 0.26 V.

hand, the energy savings decrease to 33% as compared to 38% in the typical corner (typical process, 25°C). The energy savings of R-Processor are generally higher than those shown in the PTCs (Table IV) due to its more practical size which amortizes the overhead of the EDAC technique.

The R-Processor can automatically tune the V_{DD} to the PoFF using the error statistics from EDLs. The R-Processor has an on-chip counter that counts the number of timing errors. The number of errors (particularly their LSBs) is sent to an off-chip DVS system consisting of a programmable DC power supply and an NI LabView system. When the DVS system observes new errors, it marginally increases V_{DD}. In addition, if it observes no errors for a predefined period, it reduces V_{DD} for reducing energy consumption.

The R-Processor and the proposed EDAC technique can also reliably operate at V_{DD}s down to deep sub-threshold regime. In order to experiment the functionality, as shown in Fig. 16, we modulate the ambient temperature as rapidly as possible $(0.25^{\circ}\mathrm{C/s})$ between -20°C and 70°C. In this experiment, we use an F_{CLK} of 10 MHz and employ the off-chip DVS system. The R-Processor can reliably detect and correct timing errors while the DVS automatically tunes, based on the error statistics from the R-Processor, V_{DD} from 0.26 V to 0.35 V such that the R-Processor can operate at its PoFFs over the temperature changes.

VI. CONCLUSION

In this paper, in order to remove the worst-case margins and extend the voltage scalability under the extreme variability of ULV design, we propose an EDAC technique that consist of sparse error detection, voltage-scalable error detection latch circuits, and local non-stall error correction. We design and prototype the R-Processor using the proposed technique in a 65 nm CMOS. At a typical PVT corner, R-Processor improves the minimum energy consumption by 42% via 140 mV additional voltage scaling over the baseline processor margined for the worst-case condition. At the same throughput of the baseline processor which operates at its minimum energy point, the R-Processor achieves 38% energy efficiency improvement. At the same energy consumption, the R-Processor achieves 2.3× throughput improvement. The area overhead of R-Processor is 8.3% over the baseline design.

REFERENCES

[1] B. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, Apr. 2005.

[2] M. Seok *et al.*, "The Phoenix processor: A 20 pW platform for sensor applications," in *IEEE Symp. VLSI Circuits*, 2008, pp. 188–189.

[3] Y. Lee *et al.*, "A modular 1 mm3 die-stacked sensing platform with low power I2C inter-die communication and multi-modal energy harvesting," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 229–243, Apr. 2013.

[4] M. Seok, G. Chen, S. Hanson, M. Wieckowski, D. Blaauw, and D. Sylvester, "Mitigating variability in near threshold computing," *IEEE J. Emerg. Special Topics Circuits Syst.*, vol. 1, no. 1, pp. 42–49, Mar. 2011.

[5] M. Seok, D. Jeon, C. Chakrabarti, D. Blaauw, and D. Sylvester, "Extending energy-saving voltage scaling in ultra voltage integrated circuit designs," presented at the IEEE Int. Conf. IC Design and Technology, Austin, TX, USA, Jun. 2012.

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

[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] S. Das *et al.*, "Razor II: 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.

[9] 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.

[10] 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.

[11] 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.

[12] I. Kwon, S. Kim, D. Fick, M. Kim, Y. Chen, and D. Sylvester, "Razor-lite: A light-weight register for error detection by observing virtual supply rails," *IEEE J. Solid-State Circuits*, vol. 49, no. 9, pp. 2054–2066, Sep. 2014.

[13] R. Pawlowski *et al.*, "A 530 mV 10-lane SIMD processor with variation resiliency in 45 nm SOI," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2012, pp. 492–493.

[14] 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.

[15] 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*, 2007, pp. 398–399.

[16] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, Nov. 2000.

[17] 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.

[18] R. Wilson *et al.*, "A 460 MHz at 397 mV, 2.6 GHz at 1.3 V, 32 b VLIW DSP, embedding FMAX tracking," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 452–453.

[19] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *Proc. Asia and South Pacific Design Automation Conf.*, 2010, pp. 825–831.

[20] M. Seok, D. Jeon, C. Chakrabarti, D. Blaauw, and D. Sylvester, "Pipeline strategy for improving optimal energy efficiency in ultra-low voltage design," in *Proc. ACM/IEEE Design Automation Conf.*, 2011, pp. 990–995.

[21] S. Kim and M. Seok, "R-Processor: 0.4 V resilient processor with a voltage-scalable and low-overhead in-situ error detection and correction technique in 65 nm CMOS," in *IEEE Symp. VLSI Circuits Dig.*, 2014, pp. 1–2.

[22] S. Kim and M. Seok, "Analysis and optimization of in-situ error detection techniques in ultra-low-voltage pipeline," in *Proc. IEEE/ACM Int. Symp. Low Power Electronics and Design*, 2014, pp. 291–294.

[23] M. Seok, D. Jeon, C. Chakrabarti, D. Blaauw, and D. Sylvester, "A 0.27 V, 30 MHz, 17.7 nJ-transform 1024-pt complex FFT core with super-pipelining," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2011, pp. 342–344.

[24] D. Jeon, M. Seok, C. Chakrabarti, D. Blaauw, and D. Sylvester, "A super-pipelined energy emerging client subthreshold 240 MSs FFT core in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 1, pp. 23–34, Jan. 2012.

[25] M. Nauna and D. Blaauw, "Investigating crosstalk in sub-threshold circuits," in *Proc. IEEE Int. Symp. Quality Electronic Design*, 2007, pp. 639–646.

Seongjong Kim received the B.S. degree in electrical engineering from Hanyang University, Korea, in 2010, and the M.S. degree from the University of Michigan, Ann Arbor, MI, USA, in 2012. He is currently pursuing the Ph.D. degree in electrical engineering at Columbia University, New York, NY, USA.

His research interests include low-power and variation-tolerant VLSI circuit and system design. He is a student fellow of the Catalyst Foundation.

Mingoo Seok received the B.S. degree (*summa cum laude*) in electrical engineering from Seoul National University, Seoul, Korea, in 2005, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2007 and 2011, respectively, all in electrical engineering.

He was a member of technical staff with Texas Instruments Inc., Dallas, TX, USA, in 2011. Since 2012, he has been an Assistant Professor in the Department of Electrical Engineering at Columbia University, New York, NY, USA. His research interests include variation/aging/thermal-adaptive low-power VLSI circuits and architecture, ultra-low-power SoC design for emerging embedded systems, machine-learning VLSI hardware, and non-conventional computing hardware design.

Prof. Seok received the 1999 Distinguished Undergraduate Scholarship from the Korea Foundation for Advanced Studies, the 2005 Doctoral Fellowship from the same organization, and the 2008 Rackham Pre-Doctoral Fellowship from the University of Michigan. He also won the 2009 AMD/CICC Scholarship Award for picowatt voltage reference work, and the 2009 DAC/ISSCC Design Contest for the 35 pW sensor platform design. He recently won the 2015 NSF CAREER award. He has been serving as an associate editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I since 2013 and IEEE TRANSACTIONS ON VLSI SYSTEMS since 2015.