# A Variation-Resilient Microprocessor With a Two-Level Timing Error Detection and Correction System in 28-nm CMOS

Cheng-Yao Hong and Tsung-Te Liu

***Abstract*— This article presents a variation-resilient microprocessor architecture with a two-level timing error detection and correction (EDAC) system. The proposed EDAC system first performs circuit-level error correction through time borrowing when a timing error occurs and subsequently employs a system-level error correction scheme if the timing error is relatively large and cannot be resolved within a cycle. Therefore, compared with the existing EDAC approaches, a processor using the proposed EDAC system can achieve better performance and energy efficiency under different operating conditions without incurring significant implementation effort and overhead. The proposed EDAC system was designed and implemented on an ARM Cortex-M0 microprocessor using a 28-nm CMOS process. The measurement results show that the proposed processor achieves 60.5% reduction in energy by operating under a 0.36-V lower supply voltage and at the same frequency as the baseline processor. In addition, it achieves a 37.1% reduction in minimum energy consumption compared to the baseline design.**

***Index Terms*— Adaptive design, energy-efficient, microprocessor, timing error detection and correction (EDAC), variation-resilient.**

## I. Introduction

FUTURE intelligent edge devices for Internet-of-things (IoT) applications demand more complex computations and require longer periods before batteries can be replaced. In these devices, the microprocessor must always be in ON state to ensure the reliable operation of the edge devices. Therefore, it must be highly energy efficient under different operating scenarios. Modern processor design traditionally employs the dynamic voltage and frequency scaling (DVFS) technique to improve processor performance by operating it at optimum supply voltage and frequency [1]. However, process, voltage, temperature (PVT) variations, and aging effects can lead to large circuit variabilities and can consequently cause the operation of a processor to deviate from its optimum operating point. Although designers can generally tackle the variability problem by reserving more timing margins to ensure correct circuit functionality, this worst case design approach severely limits circuit capacity and the potential performance gain that could be achieved from DVFS.

Several approaches have been proposed to minimize the performance impact of these variations instead of operating the processor under the worst case scenario [2]–[12]. One popular technique is to employ additional variability monitoring circuitry, the so-called canary circuitry, to detect and estimate circuit variability [8]. On the basis of the variability information collected by the canary circuitry, the processor operating point can be adjusted to reflect actual operating conditions and recover performance loss due to variations. This approach is straightforward, with little implementation overhead, but its effectiveness is fundamentally limited by mismatches between the main processor and the canary circuitry. As a result, this approach is impractical for processors that operate in the near- or sub-threshold regime where circuit mismatches have a high impact on performance, which seriously constrains the voltage scalability and energy efficiency of the processor [13].

Another existing solution is to use a timing error detection and correction (EDAC) system. The EDAC system prevents mismatch issues through *in situ* detection of possible timing violations that occur in the processor [2]–[7], [9]–[12]. Once a timing error event is detected, a correction mechanism at the circuit- and/or system-level can be employed to ensure that the processor reliably operates at the optimum energy-efficiency point. EDAC systems can be classified as either latch-based or flip-flop-based depending on the sequential element that they use. These two kinds of EDAC systems exhibit different performance characteristics in terms of operating speed, latency, energy, and implementation cost. Processors with latch-based EDAC systems are realized by first converting a flip-flop-based design into a two-phase latch-based pipeline system and consequently replacing the sequential latch elements located in the critical paths with error-detecting latches (EDLs) [6], [9], [10]. An EDL is designed to be capable of sending additional alarm signals when timing errors occur. Since a two-phase latch-based pipeline inherently operates with a soft timing edge, it is possible to resolve timing error events with time borrowing after one or several pipeline stages. More aggressive circuit-level correction techniques could further recover timing errors with or within a clock cycle

Manuscript received May 19, 2019; revised August 4, 2019 and September 24, 2019; accepted October 21, 2019. Date of publication November 19, 2019; date of current version July 23, 2020. This article was approved by Associate Editor Vivek De. This work was supported in part by the Ministry of Science and Technology, Taiwan, and in part by the Intelligent and Sustainable Medical Electronics Research Fund in National Taiwan University. *(Corresponding author: Tsung-Te Liu.)*

C.-Y. Hong was with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan. He is now an Independent Researcher (e-mail: cyhong@eecs.ee.ntu.edu.tw).

T.-T. Liu is with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: ttliu@ntu.edu.tw).

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

Digital Object Identifier 10.1109/JSSC.2019.2951692

by adjusting the clocking signal, supply, or body voltage of the next-stage pipeline circuits that follow the EDLs [9], [12]. As a result, processors with latch-based EDAC systems can operate even beyond the point-of-first-failure frequency (PoFF) where the timing errors first occurred. However, the conversion from flip-flop-based design to a latch-based pipeline system introduces the significant area and power overheads due to re-timing [10]. This significantly increases the implementation effort and overhead and severely limits the feasibility of latch-based EDAC systems.

By contrast, processors with flip-flop-based EDAC systems are realized by directly replacing the conventional flip-flops located in the critical paths with error-detecting flip-flops (EDFFs) [2]–[5], [7], [11]. Unlike latch-based EDAC systems, data propagation between each pipeline stage and the corresponding timing constraints in a flip-flop-based EDAC system are still similar to those of the original processor design. As a result, when a timing error event occurs, flip-flop-based EDAC designs rely only on system-level correction mechanisms such as pipeline recovery [2], [3], [7] and instruction replay [4], [5], [11] to recover the correct output data. Launching system-level correction mechanisms could severely degrade processor performance such as throughput and latency [4]. However, one significant advantage of flip-flop-based EDAC systems is that they can directly adapt commercial standard cell-based digital design flows. As a result, designers require no further knowledge of the details of the processor architecture. Therefore, compared to latch-based EDAC approaches, flip-flop-based EDAC systems are usually less invasive solutions with lower implementation effort and cost.

In this article, we propose an EDAC system with a two-level correction mechanism that combines the advantages of latch-based and flip-flop-based EDAC approaches. The proposed system employs an EDFF circuit capable of recovering correct output data to realize an instant circuit-level correction when a timing error occurs. As a result, it achieves immediate error correction and possesses the feature of time borrowing, similar to a latch-based EDAC design. Conversely, if the timing error is relatively large and cannot be resolved within a cycle, the proposed EDAC system launches instruction reply at the system level as a second-level correction mechanism to ensure reliable operation. Therefore, the proposed two-level EDAC system has the following features.

1) It achieves reliable and energy-efficient operation beyond the PoFF (similar to a latch-based EDAC system) while avoiding the high implementation overheads associated with latch-based EDAC designs such as architectural re-timing and adjustments of clocking/voltage signals in pipeline circuits.
2) Since the system-level correction mechanism is launched only when a timing error cannot be resolved within a cycle from the circuit-level correction, the cost of instruction reply operation is substantially lower than that of traditional flip-flop-based EDAC systems that perform error recovery at the PoFF.
3) Commercial standard cell-based digital design flows can be directly adapted for the proposed two-level EDAC system, hence offering high scalability and reduced implementation effort and cost.

The proposed system was verified on an ARM Cortex-M0 microprocessor using 28-nm CMOS technology. The measurement results show that the proposed processor achieves a 60.5% reduction in energy consumed by operating at 0.36-V lower supply voltage, for the same frequency of operation as the baseline processor. In addition, the proposed design achieves 37.1% lower minimum energy point (MEP) or an equivalent of 3.07× higher throughput than that of the baseline design.

The remainder of this article is organized as follows. We introduce the proposed two-level EDAC system in Section II. The operating mechanisms of both circuit- and system-level corrections are also described in detail. Section III covers the implementation details of the proposed EDAC system on an ARM Cortex-M0 MCU. The measurement results are shown in Section IV, which also provides comparisons with the state-of-the-art results, and Section V serves as the conclusion.

## II. Proposed Two-Level EDAC System

### A. Two-Level Timing Error Correction Scheme

In conventional flip-flop-based EDAC systems, EDFF circuits provide only error detection functions. The error recovery mechanism depends on system-level correction schemes such as instruction reply with lower operating frequency. Once a timing error event occurs, the processor immediately launches a system-level correction scheme to fix the error. This error correction process causes extra cycle overhead that can be defined as follows:

$$T_{\text{extra}} = T_{\text{detection}} + T_{\text{interrupt}} + T_{\text{correction}} \tag{1}$$

where $T_{\text{detection}}$ is the latency that the EDFF incurs to inform the processor of an error event, $T_{\text{interrupt}}$ is the latency needed to interrupt and flush the pipeline register, and $T_{\text{correction}}$ is the latency required to install the correct instruction and recover the correct data. For example, the EDAC system in [4] uses $T_{\text{detection}} = 1$, $T_{\text{interrupt}} = 2$, and $T_{\text{correction}} = 8$. The normalized total cycle overhead of executing an instruction set for an $N$-stage pipeline can therefore be defined as

$$T_{\text{extra,total,normalized}} = \frac{T_{\text{extra}} \times \text{ER} \times N_{\text{instruction}}}{\text{CPI} \times N_{\text{instruction}} + N} \cong \frac{T_{\text{extra}} \times \text{ER}}{\text{CPI}} \tag{2}$$

where ER is the instruction error rate, $N_{\text{instruction}}$ is the number of instructions, and CPI is the clock cycle per instruction.

To demonstrate the performance loss caused by the system-level correction, the flip-flop-based EDAC system in [4] was implemented in a five-stage Microprocessor without Interlocked Pipelined Stages (MIPS) processor to simulate the normalized total cycle overhead of a processor operating beyond the PoFF. The processor was designed to operate at a maximum frequency of 200 MHz, with a cyclic redundancy check testing program for 472 instructions. Fig. 1 shows the simulation results of the normalized total cycle overhead

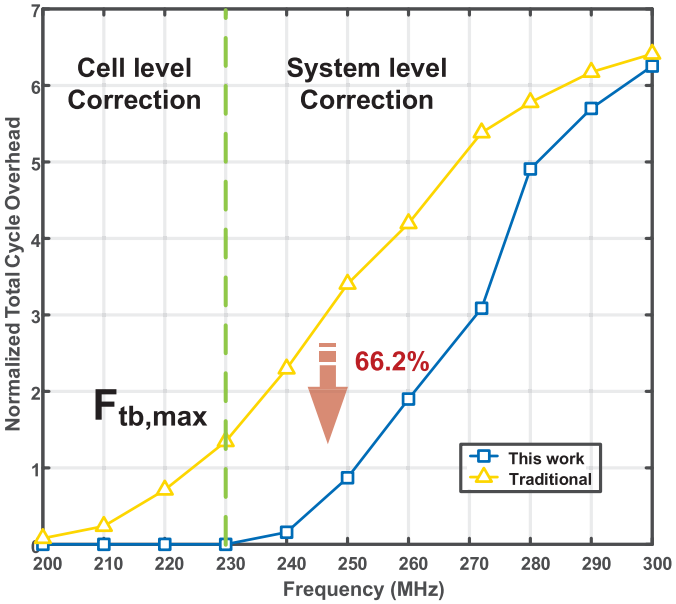


Fig. 1. Simulated total cycle overheads of the conventional flip-flop-based EDAC system and the proposed two-level EDAC system as functions of operating frequency.

of the processor at different operating frequencies using a conventional flip-flop-based EDAC system. As the operating frequency is raised beyond the PoFF, the cycle overhead increases substantially because timing error events occur more frequently. The EDAC system can detect and correct timing error events until its operating frequency reaches 272 MHz, at which point it introduces 5.38× extra cycle overhead. This extra cycle overhead causes a huge latency and energy penalty that can offset the performance gain achieved from operating the processor at a higher frequency beyond the PoFF.

To reduce the cycle overhead of conventional flip-flop-based EDAC systems, we propose a two-level error correction mechanism, as shown in Fig. 2. The corresponding processor implementation using the proposed EDAC system is shown in Fig. 3. In addition to the original system-level error correction scheme, the proposed EDAC system further uses the EDFF circuit with the time-borrowing capability to help in recovering correct data at the circuit level. Once an error event occurs, the proposed EDAC system will first try to resolve it at the circuit level between pipeline stages, similar to a two-phase latch-based pipeline. The system-level correction is launched only when a timing error cannot be eliminated from time borrowing. As shown in Fig. 2(a), if the timing error is relatively small and there is enough timing slack that can be borrowed from the following pipeline stage, the timing error can automatically be resolved between pipeline stages. Consequently, the EDFF circuit in the following pipeline will not generate a timing error signal, and the system-level correction process can be avoided. By contrast, if the timing error is so large that it causes an error event to consecutively occur in two pipeline stages, as shown in Fig. 2(b), the processor would launch a system-level correction mechanism to recover the correct data. Fig. 4 shows the timing diagram of the proposed EDAC system when the system-level correction scheme is launched. If a timing error cannot be resolved from time

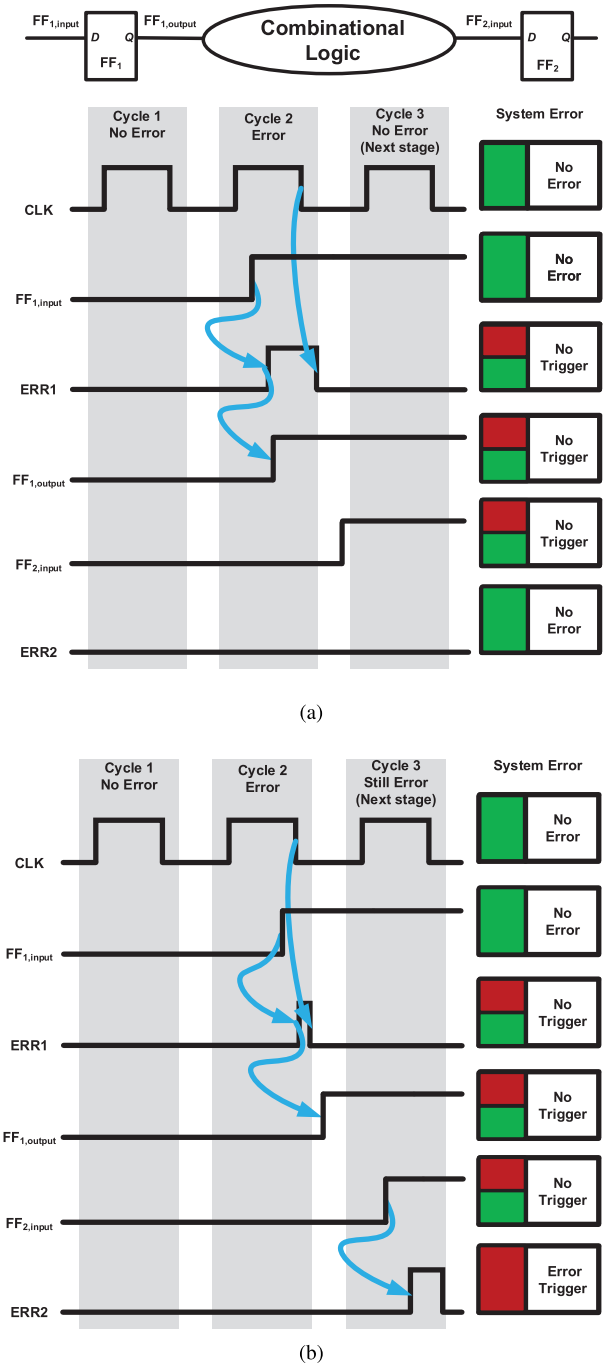


Fig. 2. Timing diagrams of the proposed two-level EDAC system when a timing error (a) can be resolved at circuit level between pipeline stages or (b) needs system-level correction to recover the correct data.

borrowing, it would trigger a system error event and the interrupt controller would issue the signal Interrupt to interrupt and flush pipeline register. The problematic instruction is then re-executed with a half clock frequency for six cycles, as shown in Fig. 4. As a result, $T_{\text{detection}} = 2$, $T_{\text{interrupt}} = 2$, and $T_{\text{correction}} = 6$, which results in $T_{\text{extra}} = 10$.

In the proposed EDAC system, the conventional EDFF circuits with only error detection functions are replaced with ones capable of correct data recovery. The design of the EDFF circuit will be described in detail in Section II-B. Additional EDFF circuits are also placed at pipeline stages after

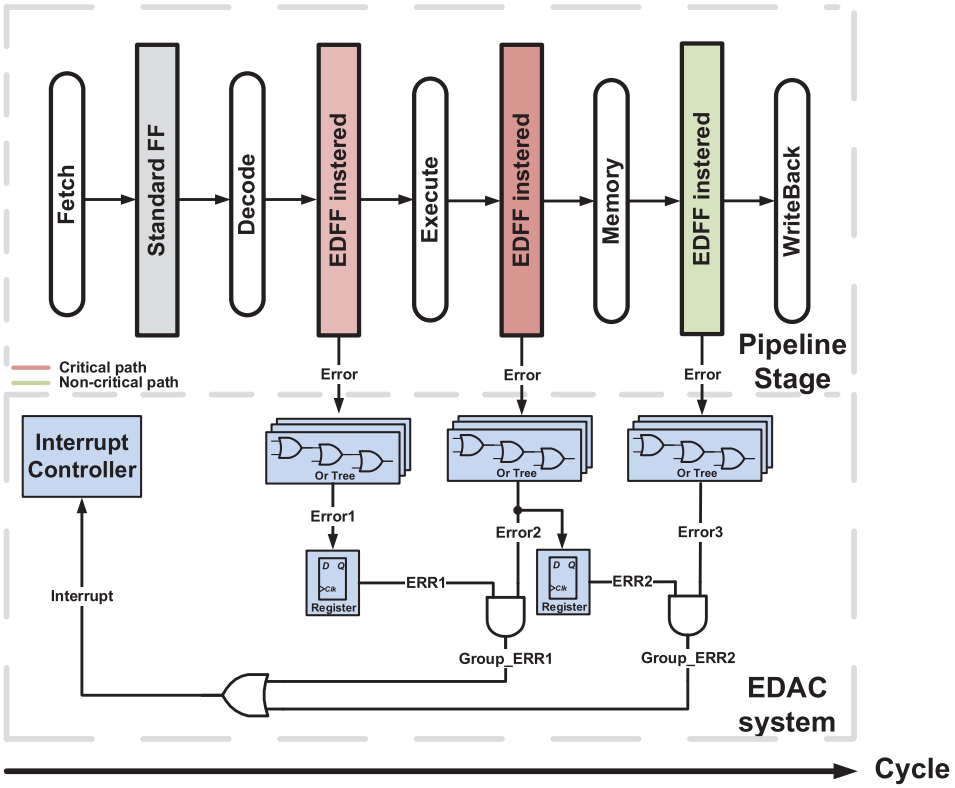


Fig. 3. Proposed EDAC system implemented in a MIPS processor.

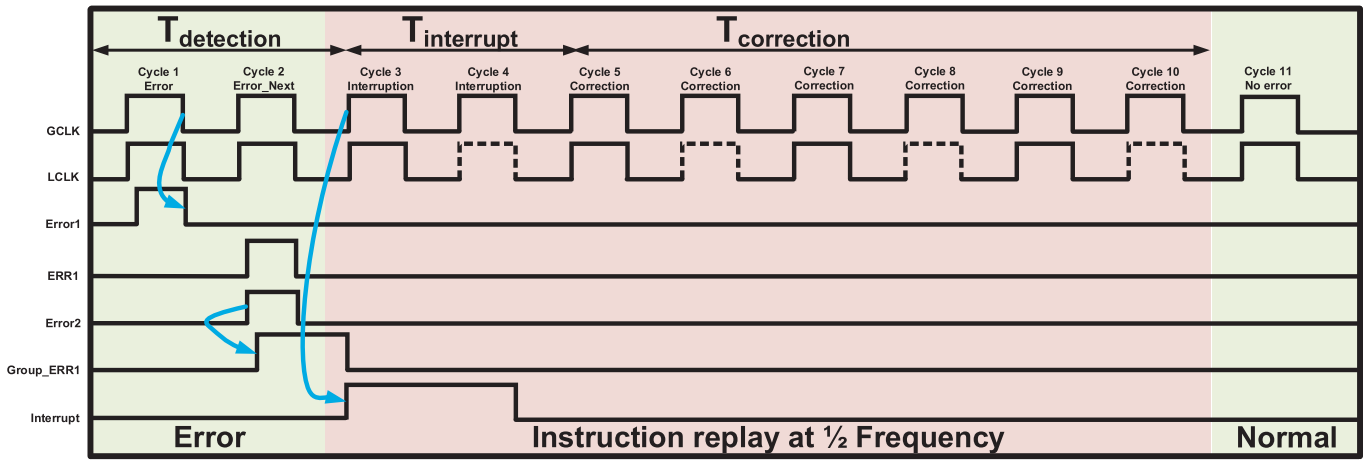


Fig. 4. Timing diagram of the proposed EDAC system when the system-level correction scheme is launched.

those located on the critical paths to enable time borrowing, as shown in Fig. 3. This causes extra area overhead of 3.2% when compared with the traditional flip-flop-based EDAC system that replaces flip-flops located only in the critical paths. Since the proposed two-level EDAC system will first try to resolve timing error between pipeline stages with time borrowing, its normalized total cycle overhead can be represented as

$$T_{\text{extra,total,normalized,two-level}} \cong \begin{cases} \dfrac{T_{\text{extra}} \times \text{ER}_{\text{two-level}}}{\text{CPI}}, & F \geq F_{\text{tb,max}}, \\ 0, & F < F_{\text{tb,max}} \end{cases} \quad (3)$$

where $\text{ER}_{\text{two-level}}$ is the instruction error rate of the proposed two-level EDAC system and $F_{\text{tb,max}}$ is the maximum operating frequency at which time borrowing can successfully eliminate a timing error. $T_{\text{extra}}$ depends on the system-level correction scheme and processor architecture. The proposed EDAC system focuses on the reduction of ER by further employing time borrowing at the circuit level to resolve the timing error first and minimize the timing error events that cause instruction error at the system level. The proposed two-level EDAC system has zero cycle overhead when the operating frequency is less than $F_{\text{tb,max}}$. It is only when the operating frequency is greater than $F_{\text{tb,max}}$ that system-level correction is launched to recover the error event that cannot be

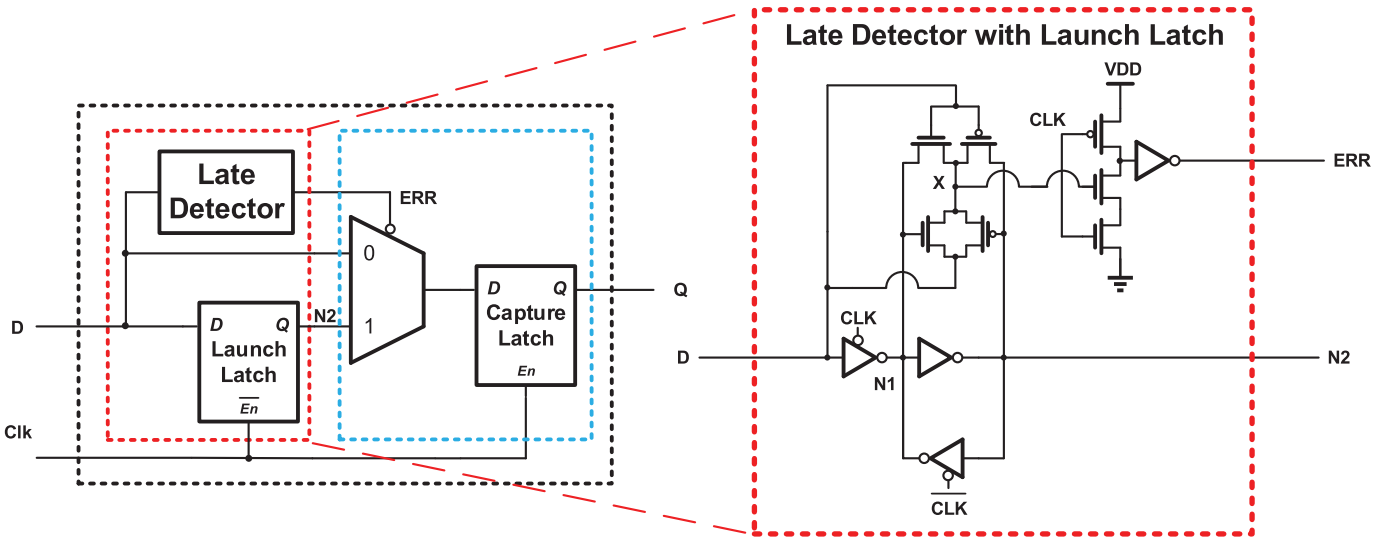


Fig. 5. Schematics of the proposed EDFF circuit.

eliminated from time borrowing. Hence, compared to conventional flip-flop-based EDAC systems with system-level correction schemes [2], [4], [5], the proposed approach achieves a much lower ER across the entire operating frequency, thereby substantially reducing the cycle overhead due to system-level correction. The simulation results shown in Fig. 1 demonstrate that the proposed EDAC system has zero cycle overhead until the system reaches $F_{\mathrm{tb,max}}$ at 230 MHz. When operating at the frequency range from 240 to 272 MHz, it achieves an average of 66.2% cycle overhead reduction with the help of circuit-level correction, when compared to the conventional flip-flop-based EDAC system. Since most of the timing errors in this processor implementation can be corrected with time borrowing between two consecutive pipeline stages, increasing the level of EDFF insertion from two-level to three-level offers very limited performance gain, but substantially increases the implementation overhead by 6.5%.

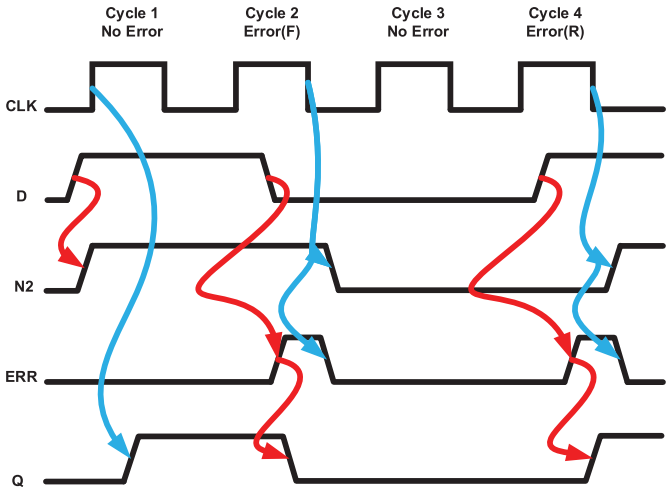


Fig. 6. Operation waveforms of the proposed EDFF circuit.

### B. EDFF Design

To support time borrowing at the circuit level as mentioned in Section II-A, the proposed EDAC system employs an EDFF circuit that not only detects timing errors but also recovers the correct output data. Fig. 5 shows the schematic of the proposed EDFF circuit. It comprises a launch latch, a capture latch, a multiplexer, and a late detector. The late detector employs the timing error detection method adapted from [14] and is designed to further minimize transistor count and energy consumption. Fig. 6 provides a detailed illustration of the operation of the EDFF. When the clock goes low, the error alarm signal ERR is reset to low. When input data $D$ change before the rising edge of the clock, the sampled data in the inner node, N2, after the clock transition are the same as $D$. As a result, there is no timing error, and the signal X and ERR stay at logical low. The sampled data in the launch latch are then passed to the capture latch. This operation is similar to a normal flip-flop, as illustrated in cycles 1 and 3 in Fig. 6. If the input data arrive after the rising edge, the sampled value in N2 is different from $D$. As a result, X will become logical high, and ERR will be raised. The multiplexer will instead select $D$ as the input to the capture latch. The output data $Q$ will thus simply follow $D$ until the clock goes low, as illustrated in cycles 2 and 4. Hence, the proposed EDFF can detect and correct timing error events with a transparency window equal to the period for which the clock is high. The correction mechanism can therefore achieve time borrowing, similar to a two-phase latch-based pipeline.

However, the additional transparency window exacerbates the short-path timing issue and results in a stricter hold-time constraint as

$$t_{\mathrm{ccq}} + t_{\mathrm{cd}} \geq t_{\mathrm{hold,EDFF}} \tag{4}$$

where $t_{\mathrm{ccq}}$ is the contamination delay of the clock to $Q$, $t_{\mathrm{cd}}$ is the contamination delay of the shortest data path, and $t_{\mathrm{hold,EDFF}}$ is the equivalent hold time of EDFF that can be

TABLE I

PERFORMANCE COMPARISONS OF THE CONVENTIONAL FLIP-FLOP AND THE PROPOSED EDFF CIRCUITS AT 0.4 V

| Architecture | Conventional Flip-lop † | Proposed Error-Detecting Flip-Flop † |
|---|---|---|
| $T_{setup}$ | -0.8 FO4 | 1.02 FO4 |
| $T_{hold}$ | 1.5 FO4 | 1.02 FO4 |
| $T_{cq}$ | 11.5 FO4 | 1.2 × |
| $T_{dq}$ | N/A | 18.7 FO4 |
| *Normalized Dynamic Energy* | 1 | 1.09 × |
| *Normalized Static Energy* | 1 | 1.1 × |
| *Normalized Area* | 1 | 1.42 × * |

† Spice simulation results including layout parasitics.
* Extra 13 transistors

expressed as

$$t_{\text{hold,EDFF}} = t_{\text{hold}} + t_w \tag{5}$$

where $t_{\text{hold}}$ is the hold time of the original flip-flop and $t_w$ is the size of the transparency window. Extra delay buffers must be inserted to maintain the functionality of the processor that employs the proposed EDAC system. This is handled by the automated design flow described in Section III.

Table I provides a performance comparison between the proposed EDFF circuit and the conventional flip-flop circuit at 0.4 V. Compared to conventional flip-flop circuits, the additional late detector and multiplexer circuit in the proposed design increase the area, switching, and leakage energy by 42%, 9%, and 10%, respectively. The proposed EDFF circuit achieves both EDAC functions using five extra transistors with an additional area overhead of 9% when compared to the state-of-the-art EDFF design in [4] which offers only error detection function. Compared to the EDFF design in [2], the proposed EDFF does not need an additional shadow latch, which substantially reduces the implementation overhead by 20.6% and significantly enhances the reliability of the EDAC system especially for near-/subthreshold voltage operation.

## III. AUTOMATED DESIGN FLOW AND CHIP IMPLEMENTATION

Existing commercial automated design flows can be adapted for the proposed two-level EDAC system to facilitate processor implementation and optimization. Fig. 7 shows an automated design flow that incorporates EDFF circuit descriptions including behavior model, timing model, and layout data at different design stages to realize a processor with the proposed EDAC system. There are two major differences between an EDFF and a standard DFF description: 1) EDFF requires additional timing parameters, $T_{\text{de}}$ and $T_{\text{eq}}$, where $T_{\text{de}}$ is the propagation delay from the data input $D$ to the output error signal ERR and

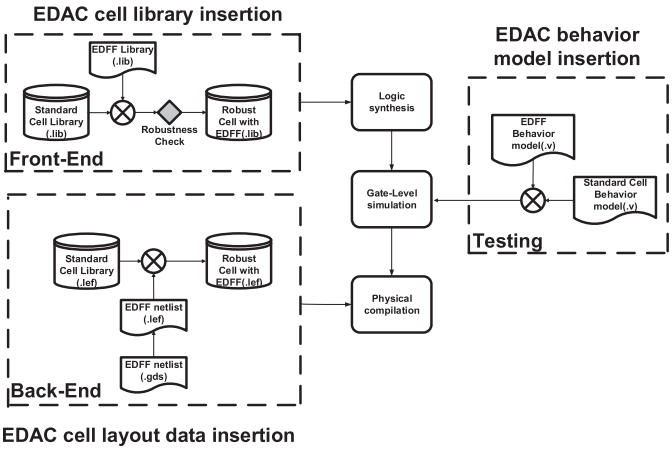


Fig. 7. Automated design flow of the processor with the proposed EDAC system.

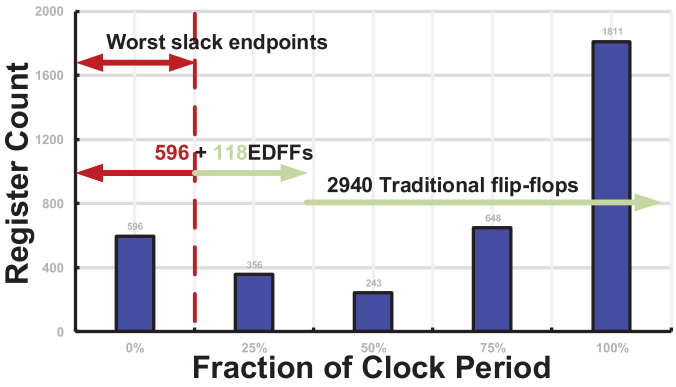


Fig. 8. Clock period histogram of an ARM Cortex-M0 microprocessor.

$T_{\text{eq}}$ is the propagation delay from the error signal ERR to the data output $Q$ and 2) EDFF has a stricter hold-time constraint. The equivalent hold time for EDFF increases in proportion to its transparency window, as shown in (5). As a result, the hold time for EDFF increases proportionally to its transparency window. With these updated timing definitions, the EDFF circuit was characterized and included in the standard cell library for processor synthesis and physical implementation. Moreover, to guarantee processor functionality at low supply voltages, static noise margin simulations were performed on each cell to check their robustness, following the method used in [15]. The standard cell library was pruned to filter out unreliable cells for low-voltage operation.

An ARM Cortex-M0 microprocessor, a widely used commercial processor for IoT applications, was implemented using the proposed EDAC system to demonstrate its effectiveness. The implementation was based on an obfuscated register-transfer level (RTL) netlist using existing digital automated design flows shown in Fig. 7 with Design Complier, Innvous, and PrimeTime. Endpoint slack analysis was first performed to determine the critical paths of the processor, as shown in Fig. 8. A total of 714 out of 3654 flip-flops were replaced with the proposed EDFFs to monitor the upper 25% of the most critical paths, representing an insertion rate

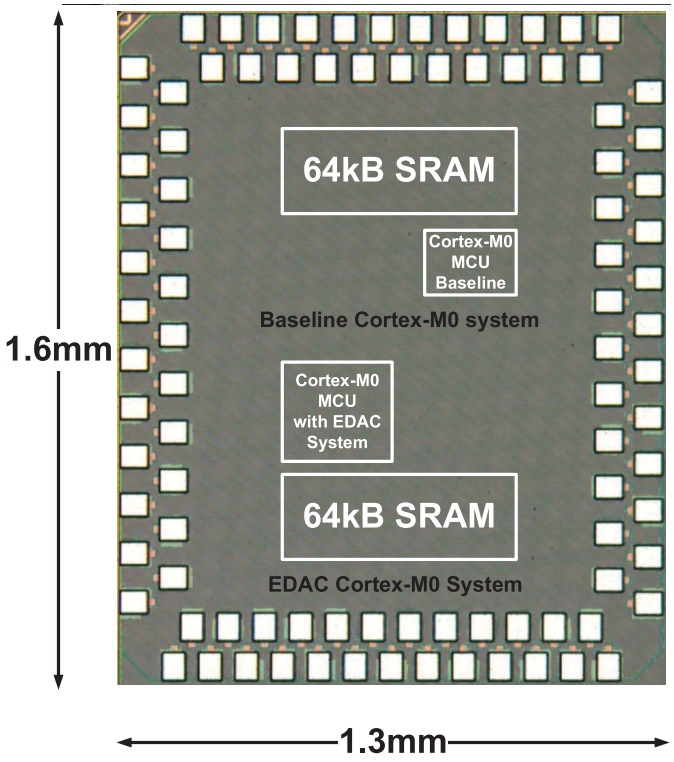


Fig. 9. Die photograph of the test chip.

of 19.5%. Fig. 9 shows a die photograph of the test chip, which was implemented using 28-nm CMOS technology. The test chip includes a processor with the proposed EDAC system and a baseline processor without any adaptive technique. The area of the proposed design including static random-access memory (SRAM) is 4.17% greater than the baseline design. The area overhead is mostly due to the insertion of hold-time buffers for short-path padding. Synthesis results show that the delay buffers used for short-path padding increases by 42.5%, which causes an additional 2.26% energy consumption. For the Cortex-M0 ARM microprocessor implementation with the proposed EDAC system, $T_{\text{detection}} = 2$, $T_{\text{interrupt}} = 2$, and $T_{\text{correction}} = 4$, which results in $T_{\text{extra}} = 8$.

## IV. Measurement Results

The performances of both the baseline processor and the proposed processor were characterized using the Dhrystone benchmark program. Figs. 10 and 11 show the measured maximum operating frequency of the baseline and the proposed processors for ten chips at nominal supply voltage and room temperature, along with the maximum frequency change that occurs when the processor operates under the worst case supply voltage and temperature conditions (10% supply voltage drop at −20 °C), respectively. Since the baseline processor does not employ any adaptive technique, the measured maximum operating frequency shown in Fig. 10 is the maximum frequency without causing any instruction error. On the other hand, the measured maximum operating frequency of the proposed processor shown in Fig. 11 is the maximum frequency that the proposed error correction

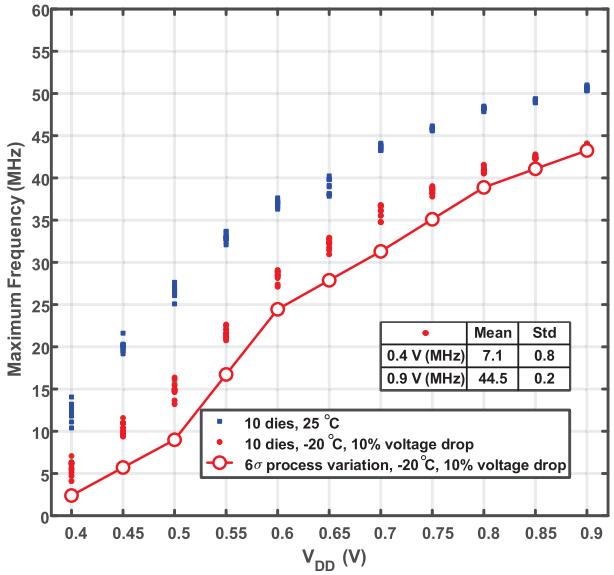


Fig. 10. Measured operating frequency of baseline processor under different operating conditions, and the extrapolated 6$\sigma$ worst case frequency.

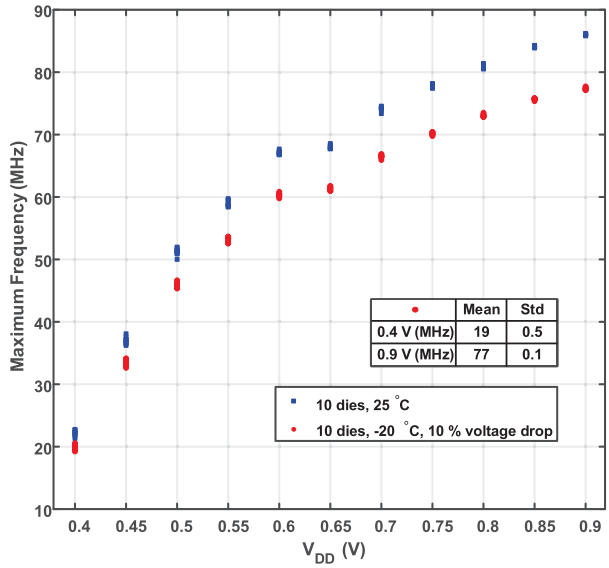


Fig. 11. Measured operating frequency of proposed processor under different operating conditions.

schemes can still handle without causing any instruction error. The 6$\sigma$ worst case frequency from these ten measurements was then used as the marginal operating frequency for the baseline processor, considering the impact of PVT variations [9].

Fig. 12 shows the measured cycle overhead and the corresponding energy per instruction of the proposed EDAC system at 0.4 V as a function of operating frequency. The measurement results show that the proposed processor has zero cycle overhead and continues to reduce its energy consumption until the operating frequency reaches $F_{\text{tb,max}}$ at 16 MHz. When the processor operates at above $F_{\text{tb,max}}$, both cycle overhead and energy consumption start to increase. These measurement results shown in Fig. 12 are consistent with (3) and the simulation results shown in Fig. 1.

Fig. 13 shows the measured average operating frequency and energy consumption of the proposed processor at different

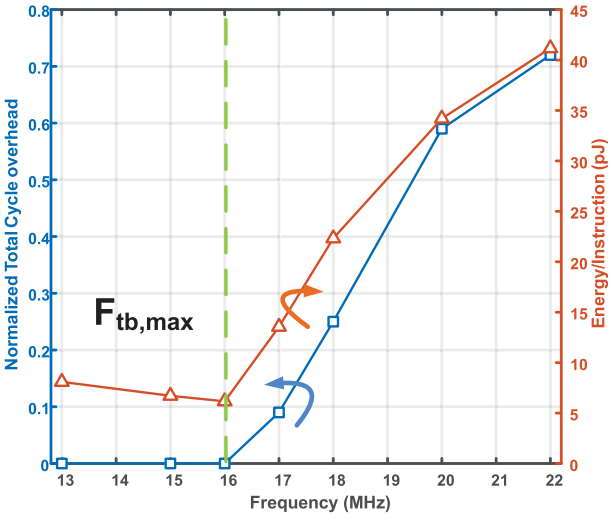


Fig. 12. Measured cycle overhead and the corresponding energy per instruction of the proposed EDAC system at 0.4 V as a function of operating frequency.

TABLE II
MEASURED PERFORMANCE COMPARISONS OF THE PROPOSED AND THE BASELINE PROCESSORS

| | Basline processor | Proposed processor |
|---|---|---|
| **Technology** | CMOS 28 $nm$ | CMOS 28 $nm$ |
| **Operating frequency** | 43 MHz | 43 MHz |
| **Supply voltage** | 0.9 V | 0.54 V |
| **Energy** | 13.14 $pJ$ | 5.19 $pJ$ |
| **Power** | 0.56 $mW$ | 0.22 $mW$ |
| **Sequential type** | Flip-Flop | Flip-Flop |
| **EDAC insertion rate** | N/A | 19.5 % |
| **Total area** | 147192 $um^2$ | 153323 $um^2$ |
| **Area overhead** | 0 % | 4.17 % |
| **Energy overhead** † | 0 % | 4.3 % |

† at the same operating frequency and supply voltage

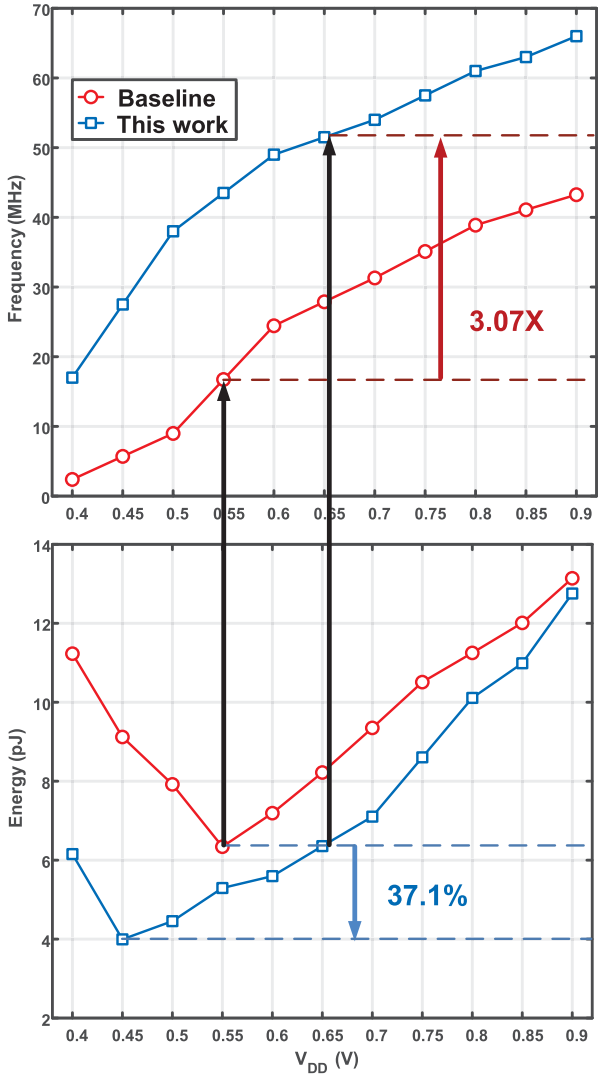


Fig. 13. Measured frequency and energy performances of the proposed and the baseline processors at different supply voltages.

supply voltages, along with the measured energy performance of the baseline processor operating at the marginal frequency extracted from the measurement data in Fig. 10. The operating frequency of the proposed processor is the frequency where the processor consumes minimum energy consumption (roughly equal to $F_{\text{tb,max}}$ as shown in Fig. 12). The energy consumption is the energy per cycle consumed by running Dhrystone benchmark program. The proposed processor can operate at 1.79× and 8.28× higher frequency than the baseline processor at 0.9 and 0.4 V, respectively. As a result, the static energy of the proposed processor is significantly lower than that of the baseline processor especially at low supply voltages. On the other hand, the proposed processor has an additional energy overhead of 4.3% due to extra circuitry. Therefore, the total energy consumption of the proposed processor is only 2.6% lower than that of the baseline processor at 0.9 V, but is 44.8% lower than that of the baseline design at 0.4 V due to the significantly less contribution of the static energy. This also explains that the proposed design can realize 37.1% lower MEP (3.99 pJ at 0.45 V) than that of the baseline design (6.34 pJ at 0.55 V). This energy-saving can also be converted into a speed performance gain. With an energy consumption specification of 6.34 pJ, the proposed processor can operate at a higher supply voltage of 0.66 V with 3.07× higher operating frequency than that of the baseline design. Table II summarizes the measured performance results for the proposed and baseline processors. The baseline processor has a maximum operating frequency of 43 MHz when PVT variations are considered. For the same operating frequency, the proposed design can operate at a lower supply voltage of 0.36 V than that in the baseline, thereby achieving a 60.5% reduction in energy consumption. The proposed processor has an extra energy overhead of 4.3% when operating at the same operating frequency and supply voltage as the baseline design, which is consistent with the area overhead of 4.17%.

Table III compares the performance of the proposed design with that of the state-of-the-art processor designs. When compared to the state-of-the-art latch-based EDAC designs [9], [10], [12], the proposed EDAC system has

TABLE III
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART DESIGNS

| | JSSC'13 [10] | JSSC'14 [4] | JSSC'15 [9] | JSSC'17 [12] | JSSC'18 [16] | This Work |
|---|---|---|---|---|---|---|
| **EDAC system** | | | | | | |
| **Sequential type** | Latch | Flip-flop | Latch | Latch | Flip-flop | Flip-flop |
| **Extra transistors** | 20 | 8 | 24 | 3 | 46 | 13 |
| **Area overhead** | N/A | 33 % | 268 % | 4.3 % | 92 % | 42 % |
| **Correction mechanism** | Bubble stall | Instruction replay | $V_{DD}$ adjustment | Clock adjustment | Time borrowing | Time borrowing + Instruction replay |
| **Prototype processor** | | | | | | |
| **Technology** | $45nm$ SOI | $45nm$ SOI | $65nm$ | $40nm$ | $40nm$ | $28nm$ |
| **Architecture** | ARM Cortex-M3 Processor | 64 bit, 7 stage Alpha Processor | 16 bit, 5 stage Processor | ARM Cortex-R4 Processor | ARM Cortex-M0 Processor | ARM Cortex-M0 Processor |
| **Target $V_{DD}$** | Super-$V_t$ | Super-$V_t$ | Near/Sub-$V_t$ | Super-$V_t$ | Near/Sub-$V_t$ | Near/Sub-$V_t$ |
| **Insertion rate** | 100 % | 20 % | 13 % | 8.7 % | 5.7 % | 19.5 % |
| **Energy saving †** | 62 % | 45.4 % | 38 % | 41 % | N/A | 60.5 % |
| **Performance improvement ‡** | 2.03 × | N/A | 2.3 × | 1.34 × | N/A | 3.07 × |
| **$E_{min}$@supply voltage** | N/A | N/A | 3.13 pJ @0.43 V | N/A | 11.12 pJ @0.31 V | 3.99 pJ @0.45 V |
| **Frequency @$E_{min}$** | N/A | N/A | 60 MHz | N/A | 7.5 MHz | 27.6 MHz |
| **Area overhead** | 103 % | 4.42 % | 8.3 % | 13.6 % | 7 % | 4.17 % |

† Iso-performance comparison
‡ Iso-energy comparison

advantages of high scalability and low implementation effort while achieving comparable performances. On the other hand, when compared to the state-of-the-art flip-flop-based EDAC designs with system-level correction schemes [2], [4], [5], the proposed EDAC system further enables time borrowing to perform instant timing error correction at the circuit level. As a result, the proposed design minimizes the cycle overhead due to system-level correction, and thus achieves better throughput and energy performances than that proposed in [2], [4], and [5] while maintaining similar implementation effort and overhead. Finally, compared to the latest EDAC work [16] that is also implemented in an ARM Cortex-M0 processor, the proposed EDAC system achieves 64.1% lower MEP with 3.68× higher operating frequency. The proposed EDAC system also has a substantially lower design complexity and overhead, and realizes a more robust operation with an additional system-level correction scheme.

## V. CONCLUSION

In this article, we presented an EDAC system with a two-level timing error correction mechanism to realize an energy-efficient, variation-resilient processor architecture. The proposed EDAC system employs an EDFF circuit capable of correcting timing error through time borrowing. If the error is relatively large and could not be resolved within a cycle, a second system-level correction scheme will be launched to ensure reliable operation. As a result, the proposed EDAC system offers similar performance advantages as latch-based EDAC designs while overcoming the large implementation effort and cost. Existing commercial automated design flows can be adapted for the proposed EDAC system with high design scalability and flexibility. An ARM Cortex-M0 microprocessor using the proposed EDAC system and design flow was implemented using a 28-nm CMOS process. The measurement results show that when operating at the same frequency as the baseline processor, the proposed processor achieves a 60.5% reduction in energy consumption by operating at 0.36 V lower supply voltage. The measured MEP is 37.1% less than the baseline design.

## ACKNOWLEDGMENT

The authors would like to thank B.-C. Wu for valuable comments, and TSMC and the Taiwan Semiconductor Research Institute (TSRI), Taiwan, for providing chip fabrication and technical support.

## REFERENCES

[1] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," *ACM SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 89–102, 2001.
[2] D. Ernst *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. 36th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2003, pp. 7–18, doi: 10.1109/MICRO.2003.1253179.
[3] S. J. Dillen, D. A. Priore, A. K. Horiuchi, and S. D. Naffziger, "Design and implementation of soft-edge flip-flops for x86–64 AMD microprocessor modules," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2012, pp. 1–4, doi: 10.1109/CICC.2012.6330707.
[4] I. Kwon, S. Kim, D. Fick, M. Kim, Y.-P. 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, doi: 10.1109/JSSC.2014.2328658.
[5] 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, doi: 10.1109/JSSC.2008.2007145.

[6] 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, doi: 10.1109/JSSC.2008.2007148.

[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, doi: 10.1109/JSSC.2006.870912.

[8] B. H. Calhoun and A. P. Chandrakasan, "Standby power reduction using dynamic voltage scaling and canary flip-flop structures," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1504–1511, Sep. 2004, doi: 10.1109/JSSC.2004.831432.

[9] S. Kim and M. Seok, "Variation-tolerant, ultra-low-voltage microprocessor with a low-overhead, within-a-cycle in-situ timing-error detection and correction technique," *IEEE J. Solid-State Circuits*, vol. 50, no. 6, pp. 1478–1490, Jun. 2015, doi: 10.1109/JSSC.2015.2418713.

[10] 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, doi: 10.1109/JSSC.2012.2220912.

[11] 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, doi: 10.1109/JSSC.2010.2079410.

[12] Y. Zhang *et al.*, "iRazor: Current-based error detection and correction scheme for PVT variation in 40-nm ARM cortex-R4 PROCESSOR," *IEEE J. Solid-State Circuits*, vol. 53, no. 2, pp. 619–631, Feb. 2018, doi: 10.1109/JSSC.2017.2749423.

[13] E. Beigné, "A 460 MHz at 397 mV, 2.6 GHz at 1.3 V, 32 bits VLIW DSP embedding F MAX tracking," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 125–136, Jan. 2015, doi: 10.1109/JSSC.2014.2369503.

[14] C.-M. Huang, T.-T. Liu, and T.-D. Chiueh, "An energy-efficient resilient flip-flop circuit with built-in timing-error detection and correction," in *Proc. VLSI Design, Autom. Test(VLSI-DAT)*, Apr. 2015, pp. 1–4, doi: 10.1109/VLSI-DAT.2015.7114574.

[15] N. Ickes *et al.*, "A 28 nm 0.6 V low power DSP for mobile applications," *IEEE J. Solid-State Circuits*, vol. 47, no. 1, pp. 35–46, Jan. 2012, doi: 10.1109/JSSC.2011.2169689.

[16] H. Reyserhove and W. Dehaene, "Margin elimination through timing error detection in a near-threshold enabled 32-bit microcontroller in 40-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 53, no. 7, pp. 2101–2113, Jul. 2018, doi: 10.1109/JSSC.2018.2821121.

**Cheng-Yao Hong** received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 2012, and the M.S. degree from National Taiwan University, Taipei, Taiwan, in 2017, all in electrical engineering.

His research interests involve energy-efficient circuit and system designs, deep learning model, hardware joint optimization, and neural architecture search algorithm.

**Tsung-Te Liu** received the B.S. and M.S. degrees from National Taiwan University, Taipei, Taiwan, in 2002 and 2004, respectively, and the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2012, all in electrical engineering.

From 2004 to 2005, he was with MediaTek Inc., Hsinchu, Taiwan, where he was involved in circuit and system design for wireless communications. From 2005 to 2012, he was a member of the Berkeley Wireless Research Center (BWRC), University of California at Berkeley. From 2012 to 2014, he was with Interuniversity Microelectronics Centre (IMEC), Leuven, Belgium, where he conducted research on circuit development for advanced CMOS technology. In 2014, he joined the faculty of National Taiwan University, where he is currently an Assistant Professor with the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering. His research interests involve energy-efficient circuit and system designs.

Dr. Liu was a recipient of several design and teaching awards.