A metastability-immune error-resilient flip-flop for near-threshold variation-tolerant designs

Sheng Wang¹, Chen Chen², Xiaoyan Xiang^{2a)}, and Jianyi Meng²

¹ Institute of VLSI Design, Zhejiang University, Hangzhou, 310027, China

² State Key Laboratory of ASIC and System, Fudan University, Shanghai, 201203, China

a) xiangxy@fudan.edu.cn

Abstract: A metastability-immune error-resilient flip-flop (MIERFF) is proposed to eliminate timing margins. It detects timing errors by generating and capturing a pulse that is wide enough to avoid metastability, in response to the data input transition. Timing errors are immediately corrected by dynamically making the master latch transparent to resample the late-arriving data. The MIERFF improves the system reliability and reduces the correction performance penalty. We apply the MIERFF to a 32-bit embedded processor in a 40 nm CMOS technology. Simulation results show that the proposed design under 0.6 V consumes 47% less energy than the traditional worst case design and achieves 6%–38% energy benefits over previous error detection and correction designs.

Keywords: energy efficiency, metastability, near threshold, timing error resilience

Classification: Integrated circuits

References

- [1] X. Lian, *et al.*: “Dynamic-static hybrid near-threshold-voltage adder design for ultra-low power applications,” *IEICE Electron. Express* **12** (2015) 20141122 (DOI: [10.1587/elex.12.20141122](https://doi.org/10.1587/elex.12.20141122)).
- [2] W. Shan, *et al.*: “Timing monitoring paths selection for wide voltage IC,” *IEICE Electron. Express* **13** (2016) 20160095 (DOI: [10.1587/elex.13.20160095](https://doi.org/10.1587/elex.13.20160095)).
- [3] 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* **50** (2015) 1478 (DOI: [10.1109/JSSC.2015.2418713](https://doi.org/10.1109/JSSC.2015.2418713)).
- [4] S. Das, *et al.*: “A self-tuning DVS processor using delay-error detection and correction,” *IEEE J. Solid-State Circuits* **41** (2006) 792 (DOI: [10.1109/JSSC.2006.870912](https://doi.org/10.1109/JSSC.2006.870912)).
- [5] M. R. Choudhury, *et al.*: “Time-borrowing circuit designs and hardware prototyping for timing error resilience,” *IEEE Trans. Comput.* **63** (2014) 497 (DOI: [10.1109/TC.2012.190](https://doi.org/10.1109/TC.2012.190)).
- [6] M. Nejat, *et al.*: “Dynamic flip-flop conversion: A time-borrowing method for

- performance improvement of low-power digital circuits prone to variations,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **23** (2015) 2724 (DOI: [10.1109/TVLSI.2014.2366918](https://doi.org/10.1109/TVLSI.2014.2366918)).
- [7] S. Valadimas, *et al.*: “Timing error tolerance in small core designs for SoC applications,” IEEE Trans. Comput. **65** (2016) 654 (DOI: [10.1109/TC.2015.2420562](https://doi.org/10.1109/TC.2015.2420562)).
- [8] I. Kwon, *et al.*: “Razor-lite: A light-weight register for error detection by observing virtual supply rails,” IEEE J. Solid-State Circuits **49** (2014) 2054 (DOI: [10.1109/JSSC.2014.2328658](https://doi.org/10.1109/JSSC.2014.2328658)).
- [9] S. Das, *et al.*: “RazorII: In situ error detection and correction for PVT and SER tolerance,” IEEE J. Solid-State Circuits **44** (2009) 32 (DOI: [10.1109/JSSC.2008.2007145](https://doi.org/10.1109/JSSC.2008.2007145)).
- [10] K. A. Bowman, *et al.*: “Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance,” IEEE J. Solid-State Circuits **44** (2009) 49 (DOI: [10.1109/JSSC.2008.2007148](https://doi.org/10.1109/JSSC.2008.2007148)).
- [11] 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 **48** (2013) 66 (DOI: [10.1109/JSSC.2012.2220912](https://doi.org/10.1109/JSSC.2012.2220912)).
- [12] S. Valadimas, *et al.*: “Cost and power efficient timing error tolerance in flip-flop based microprocessor cores,” Test Symposium (ETS) (2012) 1 (DOI: [10.1109/ETS.2012.6233002](https://doi.org/10.1109/ETS.2012.6233002)).
- [13] G. Sannena and B. P. Das: “A metastability immune timing error masking flip-flop for dynamic variation tolerance,” Great Lakes Symposium on VLSI (GLSVLSI) (2016) 151 (DOI: [10.1145/2902961.2902976](https://doi.org/10.1145/2902961.2902976)).

1 Introduction

Near-threshold computing is a promising technique to achieve high energy efficiency by scaling the supply voltage [1]. However, a serious challenge is the significant effect of process variations, voltage disturbances, temperature fluctuations (PVT) on the circuit delay [2]. Traditionally, sufficient timing margins are required to ensure correct operations across the worst case conditions. Given that the worst case conditions rarely occur, these margins lead to unwanted losses in energy efficiency.

In-situ error detection and correction (EDAC) techniques have been proposed to eliminate PVT margins [3]. These techniques monitor critical paths for late-arriving data transitions and correct timing violations after occurring. Error detection methods can be broadly classified into three categories: double sampling, transition detecting, and virtual supply rails observation. Error correction methods include instruction replay, local stalling, and time borrowing. However, the previously proposed EDAC techniques are limited by the following two critical problems: the unreliable metastability caused by timing violations and the performance penalty incurred by error corrections.

In this paper, in order to mitigate these two problems, a metastability-immune error-resilient flip-flop (MIERFF) is proposed. It detects timing errors by generating and capturing a pulse that is wide enough to avoid metastability, in response to the data input transition. Timing errors are immediately corrected by dynamically

making the master latch transparent to resample the late-arriving data. The MIERFF improves the system reliability and reduces the correction performance penalty. It is implemented in a 32-bit embedded processor to verify functionality and illustrate benefits.

The rest of this paper is organized as follows. We describe related works about EDAC in Section 2. Section 3 presents the circuit design of the MIERFF. Section 4 demonstrates the system-level design of the MIERFF. Implementation details and experimental results are provided in Section 5. Finally, the conclusion is drawn in Section 6.

2 Related works

2.1 Error detection

A representative error detection method is double sampling [4, 5]. The data input is sampled by the main flip-flop, and also by the shadow latch that provides additional time to always capture the correct data. A comparator flags a timing error when it detects a discrepancy between the shadow latch and the main flip-flop (or the master latch) output. Another common error detection method is transition detecting [6, 7]. This method compares the data input with the main flip-flop (or the master latch) during the detection window. Virtual supply rails observation [8] achieves light-weight detection by using charge sharing at the master latch internal node.

However, the state of the main flip-flop or the master latch becomes metastable when the data input transition happens during the setup time window. This is a critical problem, since the metastability can potentially lead to incorrect execution due to inconsistent interpretation and increase in propagation delay. The metastability problem gets worse at near-threshold regime because of longer resolution time, which is confirmed in Section 5.1.

2.2 Error correction

Instruction replay [9, 10] is a representative error correction method. It flushes the pipeline and re-executes the erroneous instruction at half the clock frequency. However, this method incurs a large performance penalty. To reduce the correction penalty, local stalling [11] corrects timing errors by communicating a stall to neighboring stages. But it significantly increases the design complexity and the interconnection overhead because of the point-to-point path network.

An alternative correction method is borrowing time from next pipeline stages. There are several ways to implement time borrowing. In TIMBER [5], the slave latch is driven by the master latch or the shadow latch in different time intervals to achieve time borrowing. However, the clock control and the error relay are complex. In [12], timing errors are masked by using asynchronous preset and clear inputs to flip the output. But, they have large area overhead. In [13], a multiplexer is used to pass the data input instead of the master latch output to the slave latch when an error is detected. But, the extra multiplexer between the master latch and the slave latch increases the propagation delay and the data-path loading. Moreover, this work does not compensate for the time borrowed from next stages.

3 MIERFF circuit design

The proposed MIERFF detects timing errors by generating a pulse in response to the data input transition and capturing the pulse during the high clock phase. The pulse width is larger than the setup time window to handle the metastability as a timing error. To immediately correct timing errors, the late-arriving data is resampled by dynamically making the master latch transparent.

Fig. 1. Schematic of the proposed MIERFF.

The schematic of the MIERFF is shown in Fig. 1. Based on the conventional master-slave flip-flop (CFF), it adds two parts: error detector and error corrector. (a) The error detector uses an XOR transmission gate for generating a pulse and a dynamic AND gate for capturing the pulse. Any data input transition (rising or falling) will produce a pulse, and the pulse width is determined by delay buffers. If the high pulse phase covers the high clock phase, the pre-charged node in the dynamic gate is discharged, and thus the error signal is activated to indicate a timing violation. In contrast, if the high pulse phase occurs during the low clock phase, the error signal is deactivated to indicate a normal operation. (b) The error corrector adds four transistors in the master latch to achieve time borrowing by reusing the existing data path. The MIERFF operates identically to the CFF when no error occurs since M5 and M6 are turned off, and M7 and M8 are turned on. Once the error signal is activated, M5 and M6 are turned on, and M7 and M8 are turned off. Consequently, the master latch is transparent although the clock is high (M1 and M2 are turned off). Hence, the late-arriving data is passed to the slave latch through the existing data path.

Fig. 2 shows the timing diagram of the MIERFF. In cycle 3, an input data transition during the setup time window is flagged as a timing error. Therefore, the output metastability is avoided by passing the late-arriving data to the output. In order to make the flip-flop metastability-immune, the error detection window

Fig. 2. Timing diagram of the proposed MIERFF.

(T_{EDW}) should cover the setup window (T_{setup}) across all PVT conditions. This is guaranteed by suitably determining a sufficient pulse width (T_{pulse}) to meet the timing constraint in Eq. (1). T_{EG} is the minimum overlap time required for discharging the dynamic node (error generation). T_{PG} is the data to pulse propagation delay. For a given pulse width, we selectively use higher threshold voltage (V_{TH}) transistors in delay buffers to minimize area and power overheads. In addition, the keeper circuit in the dynamic gate needs to be weak considering the contention during pre-charge and evaluation. It also uses higher V_{TH} transistors. Due to the added margin on the setup window, the error signal may be activated even when the data arrives just before the setup window. Besides, the error signal may become metastable due to a partial discharge when the data arrives slightly before the error detection window. However, since the error detection window covers the setup window, the output can correctly sample the input data without any impact on the output timing. Therefore, the metastable error signal does not pollute the pipeline function.

$$T_{setup} + T_{EG} < T_{PG} + T_{pulse}. \quad (1)$$

$$T_{EDW} = T_{setup_margin} + T_{high} - T_{ECD}. \quad (2)$$

$$T_{max} < T_{CLK} + T_{high} - T_{ECD}. \quad (3)$$

$$T_{min} > T_{high} - T_{ECD}. \quad (4)$$

An error correction delay (T_{ECD}) is needed to pass the late-arriving data to the output. It is approximately equal to the sum of the error detection delay and the data input to output delay. T_{ECD} should be deducted from the high clock phase (T_{high}) to ensure correct functionality. Hence, the T_{EDW} is defined in Eq. (2). In addition, the maximum and the minimum path delay constraints are defined in Eq. (3) and Eq. (4). T_{CLK} is the clock period.

4 MIERFF system design

As the MIERFF corrects timing errors based on time borrowing, next stages may fail to capture the correct data. In order to compensate for the time borrowed from next stages, one additional clock cycle is provided for error resolution by clock gating. Fig. 3 shows the system design of the MIERFF. Error signals of individual

MIERFFs are collected to generate a final error signal that is used to stall the next clock cycle. To reduce the error propagation delay, dynamic OR gates are used for collection. In the normal operation, the D-RESET_B signal is high and dynamic OR gates are evaluated according to error signals. Once the final error is activated, it holds the value until D-RESET_B becomes low in the next cycle. On the one hand, the final error signal is used to control a clock gate cell. On the other hand, the final error is sampled by a flip-flop to generate the dynamic OR reset signal (D-RESET_B). This flip-flop should be driven by the non-gated clock. After skipping the next clock pulse, dynamic OR gates are reset by D-RESET_B, and the normal operation resumes. In order to avoid unpredictable results, dynamic OR gates should be reset when the system is reset (IRST_B is low).

Fig. 3. System design of the proposed MIERFF.

Fig. 4 shows the timing diagram of the MIERFF system. In order to achieve in-cycle clock gating, the timing constraint in Eq. (5) should be satisfied. T_{CTD} stands for the clock tree delay. T_{EDDAPD} stands for the error detection delay and prop-

Fig. 4. Timing diagram of the MIERFF system.

agation delay. T_{SOG} stands for the setup time of the clock gate cell. Dynamic OR gates are helpful to meet this constraint.

$$T_{CTD} + T_{EDW} + T_{EDDAPD} + T_{SOG} < T_{CLK} \quad (5)$$

5 Implementation details and experimental results

5.1 Flip-flop level evaluation

The proposed MIERFF is implemented in a SMIC 40-nm CMOS technology. Based on the parasitic parameters extracted from the layout, HSPICE simulations are performed to verify functionality and confirm immunity to the metastability. In order to illustrate the metastability problem at near-threshold regime, virtual supply rails observation [8] is used for comparison under 0.4 V, as shown in Fig. 5(a). Cycle 3 shows that the CLK-Q delay is significantly increased due to metastability, which may cause the next pipeline stage to capture erroneous data. Cycle 4 shows that the timing error cannot be detected if the metastability resolution time is longer than the error detection window. At near-threshold regime, even a metastability detector becomes unreliable due to largely skewed transistor sizes. As shown in Fig. 5(b), the MIERFF can successfully detect and correct timing errors down to 0.4 V. In 100k Monte Carlo simulations with process variations, the MIERFF cause no metastable states.

Fig. 5. HSPICE waveforms under 0.4 V in a SMIC 40-nm CMOS technology. (a) Razor-Lite. (b) MIERFF.

Table I summarizes the comparisons of the proposed MIERFF and other error-resilient flip-flops at 0.4 V. As mentioned in Section 2.2, MUX-FF [13] corrects timing errors by adding a multiplexer and TET-FF [12] corrects timing errors by using extra asynchronous preset and clear inputs. The CLK-Q delay of the MIERFF is smaller than that of other two flip-flops, because the MIERFF corrects timing errors by reusing the existing data path. The D-ERROR delay of the MIERFF is also smallest, since the multiplexer increases the loading on data input and the TET-FF delay buffer located before the transition detector increases the D-ERROR delay. The MIERFF incurs the smallest energy overhead due to less data-path

loading and transistors. In addition, the MIERFF adopts higher V_{TH} transistors in delay buffers to minimize area and power overheads. The error correction delay of the MIERFF is larger than that of MUX-FF, because the multiplexer directly passes the data input to the slave latch. The difference is the propagation delay of the master latch, which is very small compared to the clock period. This can be compensated by clock gating.

Table I. Comparisons of the proposed MIERFF and other error-resilient flip-flops at 0.4 V

	CFE	MIERFF	MUX-FF	TET-FF
CLK-Q delay	3.86 ns	1.05×	1.15×	1.19×
D-ERROR delay	N/A	5.78 ns (1×	1.12×	3.33×
Error correction delay	N/A	10.38 ns (1×	0.85×	2.29×
Switching energy	0.74 fJ	2.16×	2.78×	3.50×
Correction energy	N/A	2.44 fJ (1×	1.30×	1.55×
Leakage power	0.32 nW	2.24×	2.36×	4.33×
# of transistor	29	52 (1.79×	60 (2.07×	77 (2.66×

5.2 System setup and implementation details

We apply the proposed timing error resilience approach to a three-stage, 32-bit CK802 processor. The CK802 is an energy-efficient commercial processor, which mainly focuses on cost and power-sensitive embedded applications, such as IoT and smart card.

This processor is physically designed in the SMIC 40-nm CMOS technology. The tools include Synopsys Design Compiler, IC Compiler, and PrimeTime. First, the processor is designed in a standard flow from register transfer level code to placed-and-routed netlist. Based on the static timing analysis (STA), the list of critical flip-flops is determined by the error detection window. Second, the critical flip-flops are replaced by the MIERFFs. Third, the placement and the routing are performed again to fix min-delay violations. At last, the STA is checked again to ensure no additional critical path is introduced by the added two steps. If the timing check failed, this flow jumps to the second step and further iterations are performed to achieve timing closure.

In order to evaluate performance, energy, and error rate, simulation and power analysis are performed with the final netlist, the standard delay format file, and the standard parasitic exchange format file. The tools include Synopsys VCS and PrimeTime. The benchmarks are Dhrystone-2.1 and EEMBC-1.1.

Table II provides the implementation details of this variation-tolerant MIERFF processor. The error detection window for this design is chosen to be 20% of the clock cycle. As a result, 163 flip-flops (14.2%) are replaced by the MIERFFs and 348 min-delay cells are inserted. The error signals of 163 MIERFFs are consolidated by 18 10-input dynamic OR gates. As compared with the baseline processor having no EDAC techniques, the total core area overhead due to error resilience

logics is 10.8%. The clock tree delay is 10.4% of the clock cycle. Thus, about 69.6% of the clock cycle is enough for the detected error signals to be collected and stabilized through dynamic OR gates. As reported by the post-simulation results, the variation-toleration processor can operate at 27.5 MHz under 0.6 V, typical process, and 25°C.

Table II. Processor implementation details

Technology	SMIC 40 nm
Total area	28489 μm^2
Total number of gates	44625
Supply voltage	0.6 V
Clock frequency	27.5 MHz (PoFF)
Error detection window	20% of clock cycle
Total number of flip-flops	1148
Number of MIERFFs	163 (14.2%)
Number of 10-input dynamic OR gates	18
Total number of min-delay cells	348
Clock tree delay	10.4% of clock cycle
Total core area overhead	10.8%

5.3 Energy efficiency comparison with the baseline processor

The comparison point of the baseline processor without EDAC is determined by the conventional worst-case design margins. The worst-case PVT conditions are assumed to be 10% supply voltage drop, -20°C , and three-sigma process variation. The variation-tolerant MIERFF processor operates at the point of first failure (PoFF). Two types of comparisons are shown in Fig. 6. On the one hand, the frequencies are compared at the same supply voltage. On the other hand, the energies are compared at the same frequency. The benchmark is Dhrystone-2.1.

When the supply voltage is 0.6 V, the MIERFF processor operates at 27.5 MHz (PoFF) under typical conditions (25°C, typical process). However, the baseline processor can only operate at 1.78 MHz to ensure always correct executions across the worst conditions (10% supply voltage drop: 0.54 V, -20°C , and three-sigma process variation). The MIERFF processor is 14.41 times faster than the baseline processor. In order to achieve the same frequency (27.5 MHz), the baseline processor needs to operate at 180 mV higher voltage (0.78 V). Therefore, the MIERFF processor achieves 47% energy reduction compared with the baseline processor.

It can be seen that performance and energy benefits of the MIERFF increase as the supply voltage decreases. This is because PVT variations have a larger effect on the circuit delay at lower supply voltages. The frequency of the baseline processor decreases 4.39 times when the supply voltage decreases from 0.7 V to 0.6 V. The margins reserved for the worst conditions lead to large frequency and energy losses at the near-threshold regime.

Fig. 6. Frequency and energy comparisons between the MIERFF processor and the baseline processor.

5.4 Different benchmarks

In order to fully stress the static critical paths and illustrate that benefits are applicable to different benchmarks, we add the EEMBC-1.1 benchmarks to compare their results as shown in Fig. 7. At 0.6 V, the MIERFF processor (at the PoFF) can improve frequency by 13.92–14.41 times over the baseline processor. At the same frequency determined by the PoFF, the MIERFF processor can achieve 45%–47% energy reduction over the baseline processor. Therefore, the frequency and energy benefits have no significant difference under different benchmarks.

Fig. 7. Frequency and energy benefits of the MIERFF processor under different benchmarks.

5.5 Comparison with previous techniques

Besides the immunity to the metastability mentioned in Section 5.1, the MIERFF also has low correction penalty and design overhead. To sufficiently illustrate these advantages, the MIERFF processor is compared with the processor using Razor-Lite [8] and the processor using TET-FF [12]. Razor-Lite adopts instruction replay to correct errors. TET-FF adopts error-resilient flip-flops and clock gating. MUX-FF [13] is not used for comparison since it does not compensate for the time borrowed from next stages. Three processors are implemented in the same physical flow and compared under same conditions (0.6 V, 25°C, typical process). The benchmark is Dhrystone-2.1.

Fig. 8. Throughput and energy comparisons between MIERFF and previous techniques.

In order to evaluate the correction penalty, these processors operate beyond the PoFF by increasing the frequency. Fig. 8 (left) illustrates the throughput and the error rate as a function of frequency. The throughput at each frequency point is normalized against that of the baseline processor having worst case margins. As can be seen, the throughput increases linearly with frequency until PoFF. With the further increase of frequency, the error rate increases significantly. The throughput of Razor-Lite processor declines sharply since it consumes up to 11 cycles per correction. In contrast, the throughput of MIERFF processor can increase further due to the one-cycle penalty per correction. At 30.6 MHz, the MIERFF improves throughput by 1.39 times over the Razor-Lite. TET-FF Processor also adopts time borrowing and clock gating. It is omitted from the Fig. 8 (left) since the throughput of it is similar to that of MIERFF processor.

Energies of three processors are compared under different error rates, as shown in Fig. 8 (right). The energy of Razor-Lite processor rapidly increases since the increase of error rate contributes to the longer execution time. In contrast, the energy of MIERFF or TET-FF processor changes very small due to the low correction penalty. When the error rate is 14%, the MIERFF processor can achieve 38% energy reduction over the Razor-Lite processor. As mentioned in Section 5.1, the TET-FF has 62%/55% energy overhead (normal/correction) compared with the MIERFF. The MIERFF processor achieves 6%–8% energy reduction over the TET-FF processor under different error rates, because the energy reduction of resilient flip-flops is averaged by the total system. As shown in Eq. (6), the system can achieve more energy benefits when the replacement rate of resilient flip-flops is larger. EIR_{SYS} stands for the energy improvement rate of the system. E_{COMs_CFFs} stands for the total energy of combinational logics and conventional flip-flops. $E_{MIERFFs}$ or E_{TETFFs} stands for the total energy of error-resilient flip-flops.

$$EIR_{SYS} = 1 - \frac{E_{COMs_CFFs} + E_{MIERFFs}}{E_{COMs_CFFs} + E_{TETFFs}} \quad (6)$$

6 Conclusion

In this paper, a metastability-immune error-resilient flip-flop (MIERFF) is proposed. It detects timing errors by generating a pulse in response to the data input

transition and capturing the pulse during the high clock phase. The pulse is wide enough to avoid metastability. Timing errors are immediately corrected by dynamically making the master latch transparent to resample the late-arriving data. The MIERFF is implemented in a 32-bit embedded processor under a 40 nm CMOS technology. This proposed processor under 0.6 V consumes 47% less energy than the traditional worst case design and achieves 6%–38% energy benefits over previous error detection and correction designs.

Acknowledgments

This work was supported by the Ministry of Science and Technology of the People's Republic of China under Grant 2015AA016601.