Resilient Microprocessor Design for Improving Performance and Energy Efficiency

Keith A. Bowman and James W. Tschanz
Intel Corporation, Hillsboro, OR
keith.a.bowman@intel.com

***Abstract*–In this tutorial, a 45nm resilient microprocessor core with error-detection and recovery circuits demonstrates the opportunity for improving performance and energy efficiency by mitigating the impact of dynamic parameter variations. The design methodology describes the additional steps beyond a standard design flow for integrating error-detection and recovery circuits into a microprocessor core. Silicon measurements indicate that the resilient design enables a 41% throughput benefit at iso-energy or a 22% energy reduction at iso-throughput, as compared to a conventional design.**

***Keywords*–Dynamic variations; timing errors; error detection; error recovery; resilient design; resilient circuits**

I. Introduction

Conventional microprocessors require clock frequency (F_{CLK}) guardbands to ensure correct functionality during worst-case dynamic operating variations in supply voltage (V_{CC}), temperature, and transistor aging. Consequently, these inflexible designs cannot exploit opportunities for higher performance by increasing F_{CLK} or lower energy by reducing V_{CC} during favorable operating conditions and lack of aging degradation. Since most systems usually operate at nominal conditions where worst-case scenarios rarely occur, these infrequent dynamic parameter variations severely limit the performance and energy efficiency of conventional microprocessor designs.

This tutorial reviews a 45nm resilient microprocessor core [1] to reduce the F_{CLK} guardbands for dynamic parameter variations to enhance performance and energy efficiency. The core supports two separate designs for error detection and two separate techniques for error recovery, allowing a direct comparison of the relative trade-offs. The design methodology for integrating resilient circuits into a microprocessor core describes the additional steps beyond a standard design flow. Silicon measurements from the resilient microprocessor core demonstrate the benefits in performance and energy efficiency as compared to a conventional microprocessor.

II. Microprocessor Overview

Fig. 1 provides a block diagram of the resilient microprocessor [1]. The core is an open-source, synthesizable design [2] of a 32-bit, RISC [3], seven-stage in-order pipeline that is modified to incorporate resiliency features. The first five pipeline stages are protected with error-detection circuits to identify late timing transitions. The exception (X) and write-back (WB) stages are designed with additional timing guardband to ensure dynamic-variation timing failures do not occur in these two stages. If a dynamic parameter variation induces a timing failure in any of the first five pipeline stages, the error-detection circuits identify the error and generate a single pipeline-error signal (e.g., $error_{DE}$ for the DE stage). This error signal is pipelined to the WB stage to invalidate the errant instruction and to the error-control unit (ECU) to enable error recovery. The ECU executes an error-recovery technique based on replaying the errant instruction. If the errant instruction executes correctly during the replay, the instruction commits data to the architectural state, and then subsequent instructions continue normal operation. The microprocessor also contains an adaptive clock control to monitor recovery cycles for long durations and adapt to the operating environment by changing F_{CLK} via the phase-locked loop (PLL) for maximum efficiency.

III. Error Detection

The resilient microprocessor core integrates two distinct designs for timing-error detection: (i) Embedded error-detection sequential (EDS) and (ii) Tunable replica circuit (TRC). As described in Fig. 2, the embedded EDS circuit is a double-sampling with a time-borrowing latch (DSTB) design [4] with an additional scan-enabled datapath latch to disable the EDS for testing. When the scan-enabled mode signal is logically-low, the circuit operates as an EDS circuit. In this scenario, input data is double-sampled with a shadow flip-flop (FF) on the rising clock edge and with a datapath latch on the falling clock edge. The latch and FF outputs are compared with an XOR logic gate to generate an error signal (ERROR). If input data transitions late, latch and FF

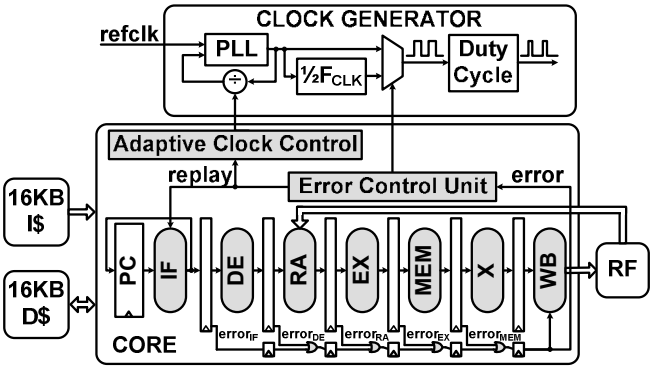


Fig. 1. Resilient microprocessor block diagram. Error-detection circuits interface with the write-back (WB) stage to invalidate errant instructions and with the error-control unit for recovery. Adaptive clock control monitors the recovery rate to dynamically change F_{CLK} during a persistent variation.

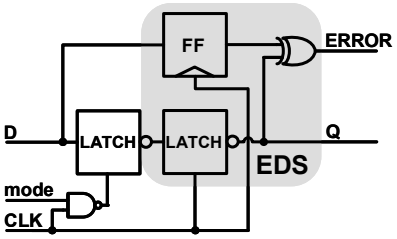


Fig. 2. Error-detection sequential (EDS). Scan-configured mode signal enables either the EDS (mode=0), where the initial datapath latch remains transparent, or a traditional master-slave flip-flop (mode=1), where ERROR is ignored.

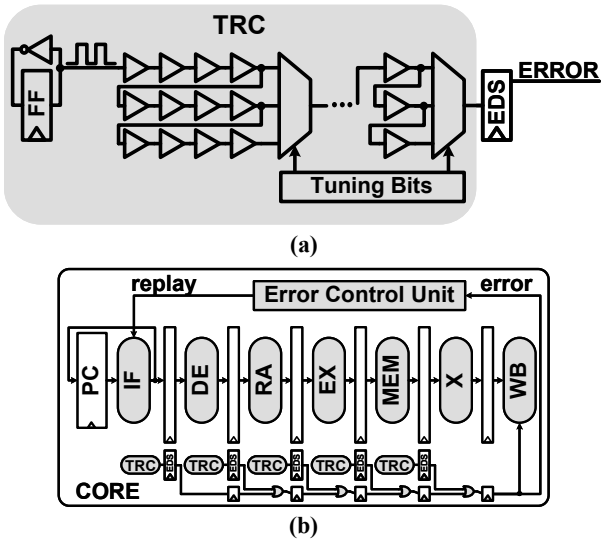


Fig. 3. (a) Tunable replica circuit (TRC) with an EDS. (b) TRC design interfaces with error recovery to detect and correct fast-changing variations such as high-frequency V_{CC} droops.

outputs differ, resulting in a logically-high error signal. The error signals from all of the EDS circuits in a pipeline stage are combined through an OR tree to produce a single pipeline-error signal. As described in section II, the pipeline-error signal propagates to the WB stage to invalidate the errant instruction and to the ECU for error recovery.

Since the high clock phase defines the error-detection window for DSTB, the minimum path delay (min-delay) with EDS circuits must not arrive before the falling clock edge. For a target error-detection window, min-delay requirements are satisfied in pre-silicon design by buffer insertion and sizing. To ensure protection from min-delay violations, the high clock phase is tuned at post-silicon with a duty-cycle control circuit as described in Fig. 1.

In previous work, a variety of EDS circuits have been proposed [4]-[10]. The salient advantages of the DSTB EDS circuit include: (i) Elimination of datapath metastability, (ii) Simple static-CMOS design, and (iii) Low clocking energy [4]. For these reasons, the DSTB circuit is chosen as the embedded EDS for the resilient microprocessor core.

In comparison to the embedded EDS circuit, the TRC design in Fig. 3 is a less-intrusive error-detection approach [11] that does not affect critical-path timing. The TRC consists of a toggle FF and a scan-configurable buffer delay chain. The toggle FF switches the input to the buffer delay chain every cycle. The TRC output drives an EDS circuit to detect timing failures due to dynamic variations.

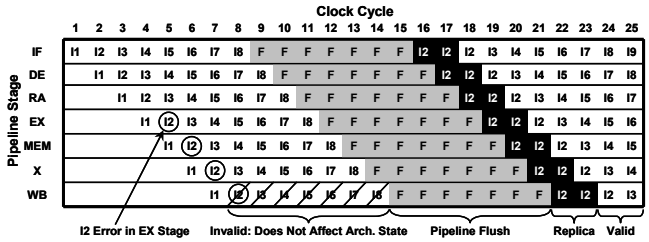


Fig. 4. Multiple-issue (MI) instruction replay with N=3: After flushing the pipeline, issue errant instruction N times; N-1 issues are replica instructions to setup pipeline registers; Nth issue is a valid instruction.

In the microprocessor core, a TRC with an EDS is placed adjacent to each pipeline stage with error detection. At test time, the TRC delays are calibrated to track critical-path delays per pipeline stage. The TRC and the pipeline stage use the same local V_{CC} and clock, enabling the TRC to detect V_{CC} droops at fine granularity and to capture clock-to-data correlations per pipeline stage. If a dynamic variation induces a late timing transition in the TRC, the EDS generates an error signal, which represents the single pipeline-error signal as discussed in section II. Although an actual timing error may not have occurred in the pipeline if the critical paths are not activated, this design inherently assumes a critical-path error did occur and initiates recovery. As described for the embedded EDS circuits, the single pipeline-error signal propagates to the WB stage to prevent the potentially errant instruction from committing data to the architectural state and to the ECU to enable recovery.

IV. Error Recovery

The resilient microprocessor core employs two separate techniques for error recovery: (i) Instruction replay at $½F_{CLK}$ and (ii) Multiple-issue instruction replay at F_{CLK}. Reducing F_{CLK} in half ensures the replayed instruction executes correctly even if dynamic variations persist [4], [10]. As described in Fig. 1, the PLL drives a clock-divider circuit to generate the $½F_{CLK}$ signal. When initiating an error recovery, the ECU signals the clock generator to reduce F_{CLK} in half while the duty-cycle control circuit maintains a constant high-phase delay for the clock to provide min-delay protection for the embedded EDS circuits. This design allows fast clock control without requiring PLL relock. After the replayed instruction finishes, the ECU signals the clock generator to resume at the target F_{CLK}.

The multiple-issue instruction replay guarantees correct execution of the replayed instruction without changing F_{CLK}. As illustrated with an example in Fig. 4, the instruction replay starts after the detected error reaches the WB stage. After flushing the pipeline, this algorithm issues the errant instruction multiple (N) times without changing F_{CLK}. The first N-1 issues are replica instructions, and the N^{th} issue is a valid instruction. The replica instructions flow through the pipeline to setup the register nodes for the valid instruction, which is allowed to commit data to the architectural state. Any error that occurs in the execution of the replica instructions is ignored and if the number of replica instructions is sufficient, each pipeline stage statically settles to the correct value, allowing the N^{th} instruction to execute correctly. If an insufficient number of replica instructions are issued such that an error occurs during the execution of the N^{th} issued instruction, then the errant instruction is replayed a 2^{nd} time with an N of eight to guarantee correct operation. Since this error-recovery design relies on setting up path nodes, this technique is directly applicable to static-CMOS logic circuits, and not dynamic logic circuits.

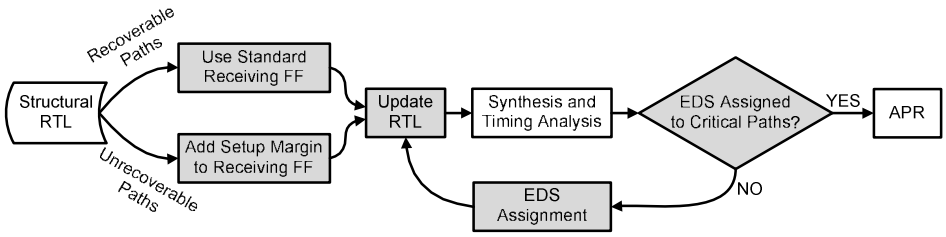


Fig. 5. Design methodology for integrating resilient circuits into a standard microprocessor synthesis flow.

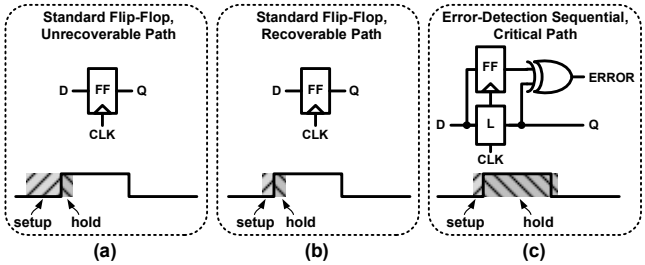


Fig. 6. Illustration of timing constraints for various sequential designs. (a) Unrecoverable paths apply an additional setup-time margin on the standard flip-flop. (b) Recoverable non-critical paths retain standard flip-flop timing. (c) Recoverable critical paths insert EDS circuits as the receiving sequential with the setup-time margin based on the shadow FF and the hold-time margin based on the error-detection window.

V. Design Methodology

The integration of resilient error-detection and correction circuits into a microprocessor core requires two additional steps beyond the typical design flow. First, the design is separated into two categories: (i) Recoverable circuits and (ii) Unrecoverable circuits. Error recovery for some paths in the design is too expensive to implement. For these unrecoverable circuits, extra timing margin is added during design and timing analysis to prevent these circuits from being susceptible to dynamic-variation timing errors. For the error-detection designs in section III, examples of unrecoverable circuits for the core pipeline in Fig. 1 include any operations in the X or WB pipeline stages. When an error occurs in the core pipeline, the resilient design must prevent the erroneous data from corrupting the microprocessor architectural state. As described in section III, the timing-error detection for a path in a given clock cycle occurs during the next cycle. Thus, the error-detection latency prevents these circuits from protecting the X or WB stages since an error in either of these stages would be identified after erroneous data had already started writing to the register file.

Second, the recoverable circuits are further subdivided into critical and non-critical paths for the embedded EDS circuits only. After timing analysis, paths with the least timing margin are classified as critical, and consequently, could limit the core performance under worst-case dynamic variations. An EDS replaces the receiving FF for these critical paths to detect potential timing errors from dynamic variations. Non-critical paths have sufficient timing margin and should not limit performance even with worst-case dynamic variations. An EDS circuit does not replace the receiving FF for non-critical paths. This second step is unnecessary for the TRC error-detection design.

As described in Fig. 5, these two additional steps are inserted into a standard register-transfer-level (RTL) to layout synthesis flow. The flow consists of RTL synthesis, timing analysis, and automatic place and route (APR) with extraction and timing convergence. The design methodology starts with the structural RTL of the microprocessor core. Next, the FFs in the core are separated into two lists: (i) Receiving FFs for recoverable paths and (ii) Receiving FFs for unrecoverable paths. The RTL is then updated with these two lists of FFs. Additional timing margin is applied on receiving FFs for unrecoverable paths to ensure correct timing even in the presence of dynamic variations. These FFs map to a unique timing model, which contains extra setup-time margin as illustrated in Fig. 6(a). At this point in the design flow, the receiving FFs for recoverable paths use standard library FFs as provided in Fig. 6(b). The updated RTL is run through the synthesis and timing analysis flow, including the physical compiler for floor-plan generation. The FFs are appropriately sized during synthesis for timing analysis while maintaining the distinction between recoverable and unrecoverable paths. Static-timing analysis generates a timing report specifying all critical paths.

After timing analysis, the timing report specifies the minimum timing margin for each receiving FF to separate the recoverable paths into critical and non-critical. The non-critical receiving FFs should not limit performance even under worst-case variations, so these sequentials remain as standard library FFs. Next, EDS circuits replace the critical receiving FFs. The RTL is now updated again with the new EDS assignments. Although EDS circuits contain a datapath latch, the setup-time margin is based on the shadow FF. Since the transparency window of the latch defines the error-detection window, traditional time-borrowing is not allowed and FF-based timing is maintained as illustrated in Fig. 6(c). The EDS circuit requires a longer hold-time margin based on the target error-detection window. The target error-detection window is designed as a specific fraction of the target cycle time, which determines the maximum potential benefits for the EDS design. The updated RTL is re-synthesized to resize logic gates in both critical and non-critical paths to minimize power for specific cycle-time and error-detection-window targets. After running timing analysis again, the timing report is verified to ensure every unrecoverable path contains sufficient max-delay margin, every recoverable critical path is assigned an EDS, and min-delay margins are satisfied. If there is a discrepancy in the timing report, this portion of the design flow is repeated. Once the design is validated with the timing report, the standard APR flow is performed.

For the embedded EDS design, 12% of the core sequentials are converted to EDS circuits, resulting in a 2.2% area penalty. The area overhead for satisfying min-delay paths with EDS circuits is 0.2%. The area penalty for the TRC design is 0.8%. The total area overheads for EDS and TRC designs are 3.8% and 2.2%, respectively, including a 1.4% area increase for the ECU and adaptive clock control. At a V_{CC} of 1V, the total power overheads are 0.9% for the EDS design and 0.6% for the TRC design, as compared to a conventional design with an equal F_{CLK} and V_{CC}.

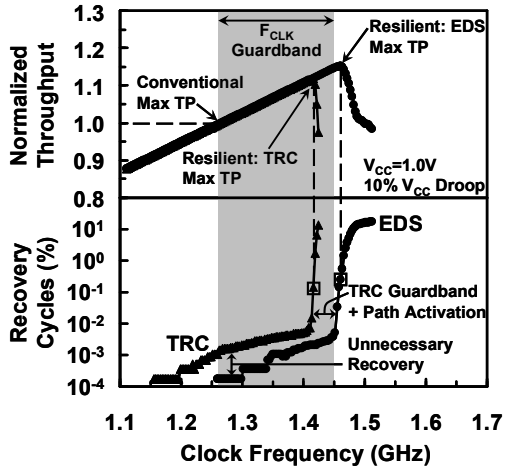
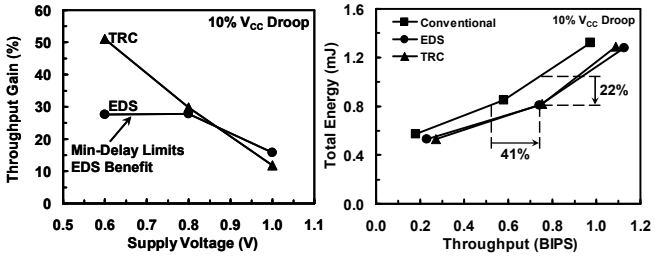


Fig. 7. Measured throughput (TP), as normalized to the conventional maximum TP, and recovery cycles, as a percentage of total cycles, versus F_{CLK}.



(a) (b)

Fig. 8. (a) Measured throughput gain versus V_{CC}. (b) Measured total energy consumption versus throughput.

VI. Measurement Results

In Fig. 7, the microprocessor core without error detection and correction (i.e., conventional design) executes a benchmark program at a maximum clock frequency (F_{MAX}) of 1.45GHz at 1.0V. When a dynamic variation in the form of a 10% V_{CC} droop occurs during program execution, the F_{MAX} reduces to 1.26GHz, corresponding to a normalized throughput (TP) of one. The shaded region represents the F_{CLK} guardband for a 10% V_{CC} droop in the conventional design. Enabling EDS or TRC designs allow detection and correction of infrequent errors from the V_{CC} droop, resulting in a higher F_{CLK} and TP. The optimal F_{CLK} for the resilient designs (1.46GHz for EDS, 1.42GHz for TRC) occurs at the point of maximum TP. Pushing F_{CLK} beyond this point reduces TP because the increasing number of recovery cycles outweighs the benefit of a larger F_{CLK}. In comparison to the conventional design, EDS and TRC designs improve TP by 16% and 12%, respectively, at 1.0V. Since the EDS design captures actual critical path errors, the TP benefit results from detecting and correcting V_{CC} droop errors as well as operating faster than infrequently-activated slow paths. In comparison, the TRC design has a lower TP than the EDS design at 1.0V for two reasons: (i) The TRC design requires a delay guardband to ensure the TRC always fails if any critical path in the pipeline stage fails, thus the performance is limited by the slowest path, and (ii) TRCs induce unnecessary recovery cycles since an actual critical path may not have been activated in the cycle with the TRC failure.

Although the EDS design provides a larger benefit at 1.0V, the error-detection window, and corresponding potential TP gain, is limited by min-delay paths. In contrast, the TRC error-detection window is not limited by min-delay paths, allowing the TRC design to capture a wider range of dynamic delay variation as measured in Fig. 8(a). At low V_{CC}, the impact of variations increases and the TRC design provides more TP gain than the EDS design (51% vs. 28% at 0.6V). In comparing the EDS and TRC designs to a conventional design in Fig. 8(b), silicon measurements indicate that resilient circuits enable either a 41% TP gain at equal energy or a 22% energy reduction at equal TP.

VII. Conclusion

A 45nm microprocessor core integrates resilient error-detection and recovery circuits to mitigate the clock frequency (F_{CLK}) guardbands for dynamic parameter variations to improve throughput and energy efficiency. The core supports two distinct error-detection designs. The first design embeds error-detection sequentials (EDS) into actual critical paths to detect late timing transitions. The second design places a tunable replica circuit (TRC) with an EDS per pipeline stage to monitor critical-path delays. Although the TRC requires a delay guardband to ensure the TRC delay is always slower than critical-path delays, the TRC design captures most of the benefits from the embedded EDS design with less implementation overhead. Furthermore, while core min-delay paths limit the potential benefits of the embedded EDS design, a salient advantage of the TRC design is the ability to detect a wider range of dynamic delay variation, as demonstrated through low supply voltage (V_{CC}) measurements. The core also supports two separate error-recovery techniques. The first technique requires clock control to replay errant instructions at ½F_{CLK}. The second technique is a multiple-issue instruction replay to correct errant instructions without requiring clock control. The design methodology describes the additional steps beyond the typical design flow for integrating resilient circuits into a microprocessor core. Silicon measurements demonstrate that resilient circuits enable a 41% throughput gain at equal energy or a 22% energy reduction at equal throughput, as compared to a conventional design.

References

[1] J. Tschanz, et al., "A 45nm Resilient and Adaptive Microprocessor Core for Dynamic Variation Tolerance," in *Proc. IEEE ISSCC Dig. Tech. Papers*, Feb. 2010, pp. 282-283.

[2] http://www.gaisler.com/cms/index.php?option=com_content&task=section&id=4&Itemid=33

[3] D. A. Patterson et al., *IEEE Computer*, Vol. 15, Issue 9, pp. 8-21, Sept. 1982.

[4] K. A. Bowman, et al., "Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance," *IEEE JSSC*, pp. 49-63, Jan. 2009.

[5] P. Franco and E. J. McCluskey, "Delay Testing of Digital Circuits by Output Waveform Analysis," in *Proc. IEEE Intl. Test Conf.*, Oct. 1991, pp. 798-807.

[6] P. Franco and E. J. McCluskey, "On-Line Testing of Digital Circuits," in *Proc. IEEE VLSI Test Symp*, Apr. 1994, pp. 167-173.

[7] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies," in *Proc. IEEE VLSI Test Symp.*, Apr. 1999, pp. 86-94.

[8] D. Ernst, et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," in *Proc. IEEE/ACM Intl. Symp. Microarchitecture (MICRO-36)*, Dec. 2003, pp. 7-18.

[9] S. Das, et al., "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction," *IEEE JSSC*, pp. 792-804, Apr. 2006.

[10] S. Das, et al., "Razor II: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE JSSC*, pp. 32-48, Jan. 2009.

[11] J. Tschanz, et al., "Tunable Replica Circuits and Adaptive Voltage-Frequency Techniques for Dynamic Voltage, Temperature, and Aging Variation Tolerance," in *IEEE Symp. VLSI Circuits Dig. Tech. Papers*, June 2009, pp. 112-113.