RISC Processor with Single Event Transient Detection and Instruction Roll-Back

Rafael Kioji Vivas Maeda
School of Electrical Engineering
Universidade Federal de Minas Gerais
Email: rafaelkioji@gmail.com

Frank Sill Torres
Department of Electronic Engineering
Universidade Federal de Minas Gerais
Email: franksill@ufmg.br

***Abstract*— As technology scales, transient faults due single event transient have emerged as a important challenge for reliability of processors. This work presents a system level intervention to overcome single event transient faults restoring a processor's state to a safe one when it is detected. A new level of memory hierarchy is defined, empowering the processor to roll back a few number of instructions. By doing so, the modified registers and memory elements may be restored to a safe value. A pipelined reduced instruction set computer architecture was modified to support this feature. The proposed architecture was synthesized using *Cadence* tools and sent to production at *ams*. A 12% total layout area increase was obtained with achievable 65MHz using this technique. Future work will be its characterization using the fabricated chip.**

I. Introduction

Embedded computing systems have become a pervasive part of daily life, used for tasks ranging from providing entertainment to assisting the functioning of key human organs [1]. Added with the large number of semiconductor devices in a single electronic system, it's reasonable to require very low device failure rates.

Aiming to overcome reliability problems of integrated circuits, a significant number of works has been released. Several techniques try to reduce permanent failures, like wearout, defects, etc. While other techniques cover temporary failures. The main causes of temporary failures is based on two kind of sources, classified as: transient and intermittent. Intermittent temporary failures are consequence of process variation, weak parts, operation margins and others. Transient temporary failures may be a result from radiation incidence, like single event transient (SET) that can lead to soft errors [2]. The latter severely impact the field-level product reliability, as identified by the International Technology Roadmap for Semiconductors [3].

The approach presented in this work tries to overcome transient faults by allowing soft errors to occur, but when they do, the processor will roll-back a few number of possibly corrupted instructions. Combined with a set of bulk built in current sensors attached to a group of cell, the processor will be able to recover its state, to a safe one, when a SET is detected.

The proposed architecture was sent to production at *ams* .35um technology. Future work will be based on its characterization with the produced chip.

The section II describes basics concepts regarding reliability and current solutions for enhancement. Section III introduces the theory of the sensors applied in this work. In section IV the roll-back processor's architecture is detailed including some comparison with the generic processor. In section V results data are presented, while section VI concludes this work.

II. Preliminaries

This section describes reliability concerns about integrated circuits and presents solutions already proposed to enhance reliability.

A. Reliability in current and future technologies

As Moore's law states, the number of transistors on integrated circuits doubles approximately every two years. Nowadays, some processors are already made out of billions of transistors with still increasing tendency. This development is possible due to continuously shrinking technology sizes. As consequence, though, reliability concerns are rising with alerting pace. Reliability is even more important in real-time safety-critical high performance products such as industrial automation, medical instrumentation, aerospace, automotive, military and nuclear reactor systems.

B. Solutions to enhance reliability

One proposed solution on the low-level design, is the insertion of redundant transistors, also called shadow-transistors [4]. An alternative low-level design approach is the enhancement via sleep transistors technique, using the well-known standby leakage reduction, increasing the lifetime reliability (classified as permanent failure) [5].

A set of system-level solutions already exist to overcome transient faults. One of them is based on error detection and correction. Specific modules are created to detect whether the memory or registers contents are corrupted and correct it based on a specialized algorithm for an transient fault tolerant microcontroller [6]. In contrast, the approach presented in [7] is based on redundancy of space and/or time for arithmetic operations.

Several low-level design approaches, like BBICS (bulk built in current sensors), requires some System-Level Interventions. Those interventions vary from the usage of redundant combinational logic blocks [8] to sequential rolling-back techniques [9]. This present work describes a system-level solution

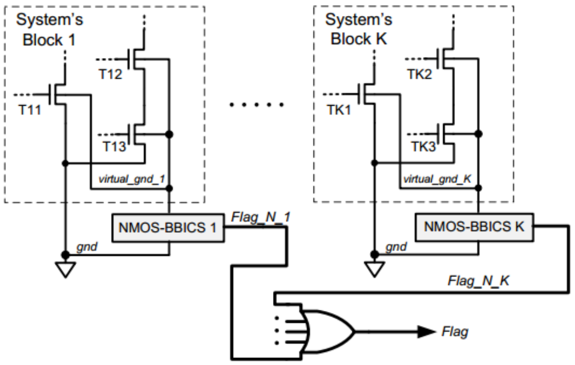

Fig. 1. A wired-*OR* set of nBBICS sensors. All sensors output generate a single flag of SET [11].

based on sequential (pipelined) logic blocks. Specifically, it will be applied to a pipelined RISC general-purpose processor. Therefore, this solution will act directly on the program counter (PC) and define a new level of memory hierarchy to roll-back a few number of instructions in case of detection of a SET by the BBICS. A similar work has been already proposed using BBICS for a 8051 processor whose architecture, though, is not based on a pipeline [10].

III. Bulk built in current sensors

The Bulk Built-In Current Sensors (BBICS) are a low-level design approach to detect soft errors. Hence, the BBICS do not avoid the SET to occur, instead, the sensor just detects it and outputs a digital signal (active low or high). Therefore, the BBICS requires some System-Level Interventions to recover to a safe state.

This section defines SET and states the theory behind the operation of the BBICS.

A. Single event transient (SET)

A single event transient (SET) is a temporary disruption of the output of a device or circuit, caused by an ionizing particle passing through the device [12]. Particles are present at space environment generated by solar activity, which may create secondary ions such as alpha particles when interacting with atoms in the target device [3].

A SET may occur both, in analog and digital circuits. In CMOS digital circuits, when a particle penetrate the struck transistor in a circuit element, it may inject or extract charge from the node that causes a temporary voltage swing around the struck mode. As the technology sizes shrinks, the amount of charge needed to store information of a bit (*minimum charge* [13]) at one node is continuously decreasing. Hence, less charge from an energetic particle is needed to change the logical state of a node, making it more susceptible to reliability problems [3], [14].

In digital circuits, when a SET occurs at a node, it may propagate as a transient voltage pulse. When this propagating pulse reaches a flip-flop, and combined with clock transition, the SET may corrupt the value of the sequential element, generating a soft error.

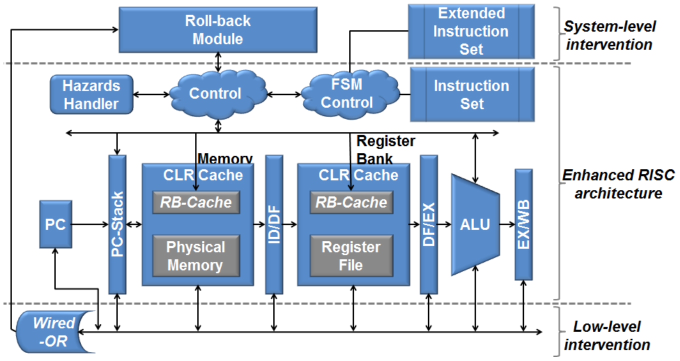

Fig. 2. Roll-back processor's architecture.

B. BBICS theory of operation

The sensing device, Bulk Built-In Current Sensors (BBICS), detects the transient current generated by the impact of an energetic particle at a sensitive circuit node [15]. Each BBICS is connected directly to the bulk of a set of transistors. The sensor continuously checks if there is a current discrepancy that may occur during a particle strike in that silicon substrate region [11].

Since each BBICS detects a possible fault inside one cell, every cell must be re-designed with this structure attached to it. Therefore, there will be an area and power penalties. Some works around BBICS technology [11] already proposed a modular BBICS that applies functional block sharing to mitigate these problems. The proposed area increase is close to 25% and a very low increase in power dissipation.

Whenever there is a SET on a specific node, the sensor attached to its closest cell will detected this event generating a digital signal (active high or low). Figure 1 shows a collection of wired-*OR* of BBICS [11].

IV. Roll-back processor architecture

To recover from a potential transient failure, the basic processor architecture must be modified. It is required to flush its pipeline vector, recover its memory contents (registers or memory cache) and take back the last reliable program counter value. This section describes to modified processor architecture.

A. Pipelined RISC processor

The proposed architecture is divided in three major levels of abstraction. The *System-Level* layer is defined by all extended hardware needed at the datapath to adapt the enhanced pipelined RISC architecture. The *enhanced pipeline RISC* layer is described by the general MIPS architecture [16] with the roll-back caches (RB-cache) inserted on its storage components (physical memory and register bank). The *Low-Level Intervention* is defined by all hardware that is added during the place & route phase, i.e. the BBICS and related circuitry. Figure 2 shows the proposed RISC architecture.

The applied processor consists of a four stage pipeline. The presented architecture is able to roll-back up to four instructions. This number can easily during design phase by adjusting the size of RB-cache and the number of stored program counter values. Therefore, the response time of its BBICS may have some flexibility with the trade-off of cache size.

Some pipeline hazards are handled in this architecture, such as data and branch hazards. This pipeline hazard controller stall the processor and/or forward data when it is detected. Also a simple static branch prediction technique is used for conditional jump instructions.

B. *Instruction set*

A basic instruction set was implemented for this processor including: arithmetic and logic operations, jumps and memory access.

Further, instructions with the purpose of simplify the debugging were added. These instructions can enable/disable the rolling-back module, insert simulated errors at specific nodes and control I/O signals (memory mapped I/O). To simulate a specific error in a selected path, user may set the special purpose register *Enable Error Register* using instruction *ERRSIM* (Error simulation). Setting a specific bit, the XOR gate will act as a inverter on the selected node simulating a corrupted node.

C. *Roll-Back cache (RB-cache)*

To make recovery possible, a new level for memory hierarchy is proposed. Figure 3 shows the Roll-Back cache (RB-cache) block diagram. The physical memory and the register bank have been redesigned to work with its own RB-cache. Figure 4 shows the interconnection between physical memory and its RB-cache.

Any write operation to either physical memory or register bank will generate a write inside its RB-cache. This cache stores the data, the address and sets a flag (*W*, storing that it was a write operation) for each written information. All saved information moves through the stack. Reaching the bottom of the stack, it automatically generates a write memory request.

Whenever there is a SET detected by the BBICS, a signal will be generated to clear all RB-caches, clearing all unsafe modifications within 4 clock cycles (instructions). There is a minimal critic area necessary for safe operation, defined as *roll-back protected* (RB-*protected*) [9]. If a SET occur at this critical zone the roll-back module cannot guarantee correctly recovery. Area increasing low-level design techniques like Selective node engineering [17] or Circuit hardening [18] should be applied to reduced drastically the probability of SET at this critical zone. The dashed zone in figure 3 defines the RB-*protected* area.

D. *Rolling-back instructions*

Similarly to the RB-cache, a stacked set of registers must be used to store the last safe program counter value. To prevent false values of program counter due to wrong branch predictions, the current program counter propagates through the pipeline. Only at *ID-DF* pipeline's stage, see figure 2, it is guaranteed that the current program counter value was actually executed and hadn't being jumped. Therefore, the program counter stack must be stored there.

Whenever a BBICS detects a SET event, its output will trigger the roll-back module. The roll-back module will, then, clear all RB-caches, flush the pipeline vector and recover the last reliable PC at the bottom of the stacked values. Using a wired-*OR* logic with all detection devices, the processor may recover from the SET event wherever it occurred.

Since the pipeline vector is flushed and the last reliable program counter is taken from the bottom of the PC-stack, stored at the *ID-DF*, third pipeline's stage, then the worst case penalty time is six clock cycles.

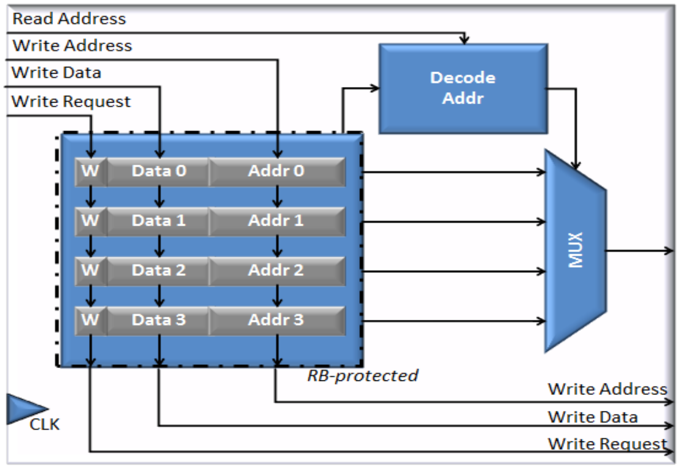

Fig. 3. RB-cache. New level of memory hierarchy. Physical memory and register bank will have a RB-cache to store temporary data. Any data written will be delayed for 4 clock cycles to guarantee a safe 4 instruction roll-back.

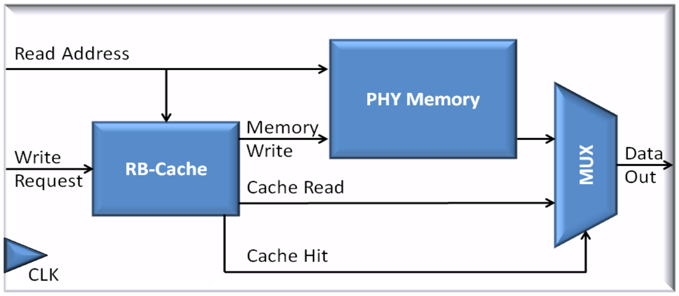

Fig. 4. Chaced Memory - Example of interconnection of physical memory (including all memory hierarchy level) and its RB-cache.

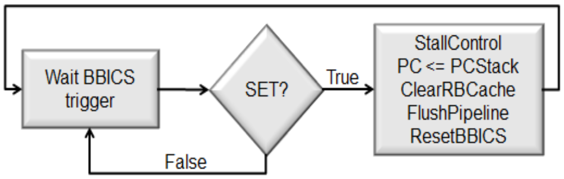

Fig. 5. Rolling back instructions. Simplified state machine for Roll-back module's control.

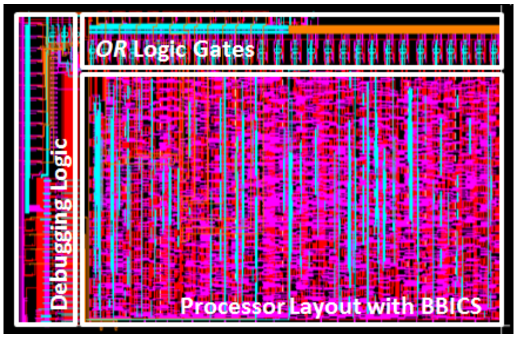

Fig. 6. Synthesized roll-back processor. Semi-custom design using *ams* 0.35um technology. Upper block shows the OR logic gates unifying all BBICS signals, left block is the synthesized debugging logic and bottom block is the processor layout using BBICS.

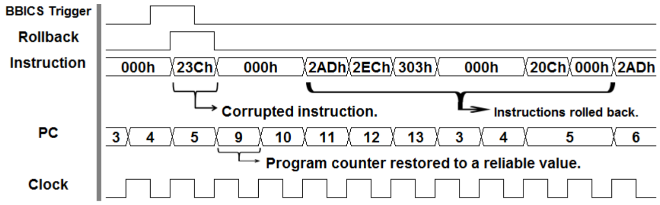

Fig. 7. Simulating a SET with roll-back module enabled. The simulated trigger (*BBICS Trigger* signal) generates the roll-back pulse response (*Rollback*). Although the instruction opcode is still being corrupted, the instructions are executed again correctly, restoring the processor's state to a safe one.

V. Results

The processor was realized in the *ams* 0.35um semi-custom technology and sent for fabrication. Figure 6 shows the final layout of the circuit.

A. Simulation results

For test purposes a *Fibonacci* algorithm was implemented. Next, using the Path-Error Simulation module, the simulated SET occurrence times were varied so that each instruction was tested during at least one SET. In could be observed that for all simulated cases, the processor could return to a save and correct state and continue without any interference. The RTL simulation was described using a Hardware Description Language while *Synopsys VCS* was applied for simulations.

Figure 7 shows the waveform obtained when the roll-back feature was enabled. Here the simulated SET triggers the roll-back module that restore the program counter value and flush all *RB-caches*, rolling back the processor to a well known state. All rolled back instructions are executed again. On the opposite scenario, with the roll-back feature disabled, the corrupted instruction is executed causing a wrong processor behaviour.

B. Synthesis analysis

Due to the addition of the current sensors the area of the processor changed from $1.21\mathrm{mm}^2$ to 1.36, which corresponds to an area increase of 12%. The maximum frequency achieved with this architecture, using the standard cell model, is about 65MHz. Since the BBICS latency increase was not considered, it is expected that the maximum frequency should be lower than that.

VI. Conclusion

This work presents a system level intervention to overcome single event transient faults. The main idea behind the intervention is to restore the processor's state to a safe one when it is detected a single event transient.

A specific sensor devices for transient fault detection were added leading to an area increase of 12%. This work was limited to functional simulation, its correct functionality was validated. Its characterization will be performed using the fabricated chip achieving a more robust test bench using a emulated single event transient.

Acknowledgment

This work was supported by CNPq, INCT/DISSE, FAPEMIG and PRPq/UFMG.

References

[1] Vijaykrishnan Narayanan, and Yuan Xie, *Reliability Concerns in Embedded System Designs*, The Pennsylvania State University.

[2] Subhasish Mitra, Vijay Narayanan, Lisa Spainhower, and Yuan Xie, *Robust System Design from Unreliable Components*, 2005.

[3] Egas Henes Neto, Fernanda Lima Kastensmidt, and Gilson Wirth, *Tbulk-BICS: A Built-In Current Sensor Robust to Process and Temperature Variations for Soft Error Detection*, IEEE, 2008.

[4] Cornelius, C. ; SILL , F. ; Saemrow, H. ; SALZMANN, J. ; Timmermann, D. ; SILVA, D. *Encountering gate oxide breakdown with shadow transistors to increase reliability*. In: Symposium on Integrated Circuits and Systems Design (SBCCI), 2008, Gramado, Brazil. Proceedings of 21st Symposium on Integrated Circuits and Systems Design. New York: ACM, 2008. v. 1. p. 111-116.

[5] Cornelius, Claas ; SILL , F. ; Timmermann, Dirk. *Power-Efficient Application of Sleep Transistors to Enhance the Reliability of Integrated Circuits*. Journal of Low Power Electronics (Print), v. 7, p. 552-561, 2011.

[6] Érika Cota, Fernanda Lima, Sana Rezgui, Luigi Carro, Marcelo Lubaszewski, and Ricardo Reis, *Synthesis of an 8051-Like Micro-Controller Tolerant to Transient Faults*, 2001.

[7] Lorena Anghel, Dan Alexandrescu, and Michael Nicolaidis, *Evaluation of a Soft Error Tolerance Technique Based on Time and/or Space Redundancy*, TIMA Laboratory. Grenoble, France.

[8] Franco Ripoll Leite, *Estudo e implementao de um microcontrolador tolerante radiao*, Porto Alegre, Rio Grande do Sul - Brazil, 2009.

[9] Meeta S. Gupta, Krishna K. Rangan, Michael D. Smith, and David Brooks, *DeCoR: A Delayed Commit and Rollback Mechanism for Handling Inductive Noise in Processors*, Harvard University.

[10] Franco Ripoll Leite, *Estudo e implementação de um microcontrolador tolerante radiação*, Universidade Federal do Rio Grande do Sul, 2009.

[11] Sill, and Bastos, *Robust Modular Bulk Built-In Current Sensors for Dection of Transient Faults*, SBCCI, 2012.

[12] Stephen Buchner, and Dale MacMorrow, *Single Event Transients in Linear Integrated Circuits*, IEEE, 2005.

[13] Tino Heijmen, *Radiation-induced soft errors in digital circuits: A literature survey*, Philips Electronics Nederland BV, 2002.

[14] Chong Zhao, *Analysis and Design of Reliable Nanometer Circuits*, University of California - San Diego, 2007.

[15] Gilson Wirth, *Bulk built in current sensors for single event transient detection in deep-submicron technologies*, Rio Grande do Sul, Brazil, 2008.

[16] John L. Hennessy, and David A. Patterson, *Computer Architecture: A Quantitative Approach*, 4th Edition, ISBN-10: 0123704901, Appendix A.

[17] T. Calin, M. Nicolaidis, and R. Velaco, *Upset Hardened Memory Design for Submicron CMOS Technology*, IEEE Trans. Nucl. Sci., Vol. 43, pp. 2874-2878, Dec. 1996.

[18] T. Karnik, et al., *Selective Node Engineering for Chip-level Soft Error Rate Improvement*, VLSI Circuits Symp., pp. 204-205, 2002