Designing Reliable Processor Cores in Ultimate CMOS and Beyond: a Double Sampling Solution

Thierry Bonnoit
Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, 38000 Grenoble, France
thierry.bonnoit@univ-grenoble-alpes.fr

Fraidy Bouesse
Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, 38000 Grenoble, France
fraidy.bouesse@univ-grenoble-alpes.fr

Nacer-Eddine Zergainoh
Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, 38000 Grenoble, France
nacer-eddine.zergainoh@univ-grenoble-alpes.fr

Michael Nicolaidis
Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, 38000 Grenoble, France
michael.nicolaidis@univ-grenoble-alpes.fr

***Abstract*— The double sampling paradigm is an efficient method to protect the circuits against soft-errors. But the data that are going out of the area protected by double sampling are still vulnerable. In this paper we proposed an architectural solution that uses three latches to remove those constraints and protect the area outside the double sampling domain without adding a buffer stage.**

Keywords— double sampling; soft-errors; LEON3; reliability

I. Introduction

Aggressive technology scaling has dramatic impact on the reliability of circuits produced in nanometric fabrication. Several solutions were already proposed to detect and/or mask the errors in operating circuits. Those based on hardware redundancy must make a tradeoff between the fault coverage, the hardware overhead, and the speed penalty [1]-[2]. Current sensors can detect single event transients (SETs), but will induce a lot of false positives depending of their sensitivities [3]. Some sensors that controls the characteristics of the transistors and circuits mimicking the critical paths of logic circuits are used to detect the degradation of a circuit to regulate its operating frequency and power consumption [4]-[5]. However, they cannot detect errors due to radiations and electromagnetic interferences.

An alternative approach to detect the errors uses double sampling design paradigm [6], which addresses all the failures at low area and power consumption penalties. The double sampling (DS) approach adds to the main flip-flop a redundant sampling element (latch or flip-flop), which stores the same data at a different time. The error is detected by comparing the two sampling elements (SEs). A major issue concerns the vulnerability of data that go out of the part of the circuit protected by DS (DS area) and reach another part of the circuit called outside DS area (ODS area), like memories, registers file or parts protected by triple modular redundancy (TMR). This may allow an error from the last DS area stage to contaminate the ODS area. To eliminate this vulnerability without having additional constraints on the datapaths, the most common solution adds a buffer stage between the DS area and the ODS area of the circuit [7]-[8]. Thus, when an error is detected in the DS part, this stage will prevent further propagation of corrupted data in the circuit. However, this buffer stage, as well as the connected logic at the inputs and outputs, must also be protected using the TMR approach which considerably increases the hardware overhead. Another solution uses the error signal to avoid any contamination of the ODS area by erroneous data of the DS area. But implementing such a solution without any additional buffer stage requires additional implementation constraints. The architecture presented in [9] reduces those constraints, without removing them.

The main contribution of this paper is an architecture that enables the connection of the DS area to the ODS area without any buffer stage or additional implementation constraint. Section 2 present the basis of the DS paradigm, section 3 explains the issue due to the data that go out of the DS area, presents the related works and the proposed solution, and section 4 shows its implementation on the LEON3 processor.

II. Basic double sampling architectures

Most of the DS architectures could be represented as in Fig. 1. The result of a combinational logic bloc (COMB) is stored in a Main Sampling Element (MSE). The same result is also stored in a Redundant Sampling Element (RSE) at a different time. Thus, those two SEs store the same signal when no error occurs. The outputs of MSE and RSE are then compared (Comp) to generate an alarm signal in case they have not stored the same bit. In Fig. 1 a single pair (MSE, RSE) is displayed for the sake of simplicity. Actually the comparator must check the bits of a set of pairs (MSE, RSE). The resulting error signal is stored in an Error Sampling Element (ESE). Depending of the implementation, each SE can be a flip-flop or a latch [6]. Moreover, RSE and ESE use clock signals that are delayed compared to the main clock signal CLKm used by

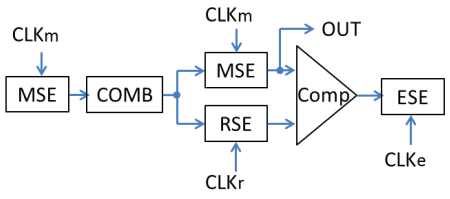

Fig. 1. Basic double sampling architecure

* Institute of Engineering Univ. Grenoble Alpes.

MSE ($CLKr = CLKm + \delta$ and $CLKe = CLKm + \delta + \delta'$). In the following, *Tm,s*, *Tr,s* and *Te,s* are respectively the setup time of MSE, RSE and ESE. Also, *Tm,h*, *Tr,h*, and *Te,h* stand for the hold time of MSE, RSE and ESE.

A. Implementation to detect short duration SETs

The basic implementation of the DS is suitable to detect short duration SETs [6]. In this scheme MSE and ESE are two flip-flops, and RSE can be a latch or a flip-flop. The results are stored in MSE first, and next in RSE. To work properly the implementation must respect the constraints (1) and (2).

$$Dmin > \delta + Tr,h \quad (1)$$

$$\delta' > Dcomp + Te,s \quad (2)$$

Dmin stands for the delay of the shortest path protected by DS in the design, and *Dcomp* is the largest path of the Comp circuit. To not generate delayed clock signals, MSE and ESE can sample data on the rising edge of CLKm, and RSE can use the falling edge of CLKm. Thus, we consider in the following that for this implementation $\delta = Tu$ and $\delta' = Td$, where *Tu* and *Td* are respectively the duration of the high level and the low level of the clock signal. A drawback of this architecture is the constraint on the *Dmin* parameter that may lead to add a lot of delay elements that increase the hardware and the power consumption of the circuit. However, reducing the parameter *Tu* (δ) will also reduce the duration of the SETs that can be detected. A SET is not detectable if the transient pulse contaminates both MSE and RSE during the same clock cycle. Thus, the minimal duration *Dset* of an undetectable SET is:

$$Dset > Tu - Tr,s - Tm,h \quad (3)$$

Also, a SBU can affect a MSE without being detected. It happens when the SBU occurs soon enough to be propagated to the next MSE, but late enough so that the error signal cannot reach the ESE in time. However, the DS area is protected against all SBUs [6] if:

$$Dmin > Dcomp + Tm,h + Te,s \quad (4)$$

Another implementation is presented in [8] that handle datapath metastability. In this solution, MSE is a latch, RSE is a flip-flop, and ESE is a pair (MSE, RSE) as it is also protected by DS. RSE samples the signal on the rising edge of CLKm. MSE is locked during the low level of CLKm. However, the implementation constraints are barely changed as *Dmin* is also constrained by the parameter *Tu* like in (1).

B. Configuration to detect long duration SETs

The same architecture but with another configuration could be used to detect SETs with large current pulse that can be encountered in space application [6]. For doing so, the result of COMB is stored in the RSE first, and next in the MSE. The error signal is generated during the next clock cycle. In this case, RSE and ESE use the falling edge of CLKm ($\delta = Tu$ and $\delta' = Tu + Td$). Unlike the previous configuration, this scheme does not add any constraint on the *Dmin* parameter to work properly. On the other hand, the operating frequency might be reduced, as the *Tu* parameter should be greater than the delay of the critical path of the DS area. In this scheme, the minimal duration of an undetectable SET is given by:

$$Dset > Td - Tr,h - Tm,s \quad (5)$$

As *Dmin* is not constrained any more, the duration of the detectable SET can be regulated using the *Td* parameter. Moreover, the circuit is protected against all SBUs if:

$$Dmin > Dcomp + Tr,h + Te,s \quad (6)$$

III. Going out of the DS domain

DS cannot protect the whole design. The solution is not suitable for some modules (RAM, register file) and, some parts of the design may store critical data (for instruction replay…) for which it is worth to use a more costly solution (like TMR). Fig. 2 displays a datapath from the DS area to the ODS area. ODS-SE can be either the first pipeline stage of the ODS area, a RAM or a register file. The write enable (wren) of the first SE belonging to the ODS area is controlled by the ESE to avoid any contamination of the ODS area by corrupted data of the DS area. In this path COMB is not protected anymore by DS, and should be protected by another solution as well as the ODS-SE. But, using the output of ESE as the input of wren has consequences on the protection of ODS-SE against the SEUs coming from the DS part. Actually there is an opportunity for a SBU in the DS area to contaminate the ODS area. This is due to the gap between the clock signal CLKo of ODS-SE and CLKe. To detect each SBU, the timing constraints for both configurations (short and long SET detections) (4) and (6) should be increased by $\alpha = CLKo - CLKe$. This new constraint is mandatory for all paths going outside the DS area.

A. Related work

This issue could be resolved by allowing the first stage of the ODS area to be contaminated by an error that comes from the DS area [7]-[8]. In this case, the data that go outside the DS area are stored first in an intermediate sampling element (ISE). When an error coming from the DS area is stored in ISE, the error signal will be stored in ESE during the same clock cycle. Thus, the ESE will be able to deactivate the write enable of the ODS-SE stage, preventing the error to be propagated during

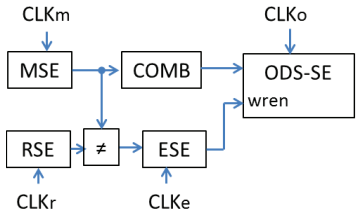

Fig. 2. DS area to ODS area, straightforward implementation

the next cycle. Moreover, if ESE and ISE sample their signal at the same time, there is no additional constraint on the datapath ($\alpha = 0$). To be worth, this solution should consider that the ISE and the combinatory functions connected at its input and output must be protected against the SETs and SBUs. Thus, implementing this stage requires either: to add a pipeline stage protected by TMR; to replace the last stage that could have been protected by DS by a stage protected by TMR. This increases the cost of the DS implementation in both cases. Moreover, the buffer stage protected by TMR cannot be used to preserve some critical data, as it can be contaminated by an error from the DS area. To reduce the hardware overhead, [7]-[8] consider only the timing faults. Therefore, the protection of the path C1 and C2 is ensured by adding a margin on the delay constraints. Some other solutions in the context of the DS that prevent erroneous data to go further use a Muller c-element. It propagates a bit stored in a pair (MSE, RSE) only if its two inputs are identical [10]-[11]. A weak-keeper preserves the previous state of its output in case the bit is not transmitted. The code word state preserving (CWSP) gates [12] are based on the same idea. For each bit the last gate of a logical function protected by DMR propagates the result only if both inputs are equal. But all those solutions ([10]-[12]) also require the presence of an ESE. Actually, a set of pairs (c-element, weak-keeper) controls bit per bit the propagation of data. Thus, it may lead to write data in the ODS area that contain some bits that actually belong to the previous data. As a consequence, the result will not be correct. There are also solutions that mask the errors using three flip-flops that sample the same data at three different times [13]. But to guarantee that it will have the same SEUs detection capacity as the DS the timing constraints must be doubled. Another solution in [14] uses an additional flip-flop that preserves the last correct state of a flip-flop protected by DS. It allows the circuit or a part of the circuit to restart from its last correct state after an error was detected. However, this solution at least requires all the flip-flops of the last stage of the DS area to be tripled.

To avoid any contamination, the first stage of the ODS area must use an ISE that is not a simple flip-flop. The ideal ISE is a master-slave flip-flop (Fig. 3). The first latch ISE1 is locked when CLKi reaches its high level. The write in the latch ISE2 (when the CLKi signal is up) is controlled by the ESE. However, this solution does not completely remove the additional timing constraint α. By definition, to guarantee $\alpha = 0$, ISE1 and ESE shall sample their signals at the same time. But then, the ESE output may not be fast enough to avoid the write operation in ISE2. An example of such a situation can be found in [9]. In this case, CLKe = CLMm ($\alpha = 0$). To be able to lock ISE2 before its contamination, the flip-flop ESE is replaced by a latch that locks the signal at the same time as ISE1. If the error signal reach the ESE soon enough, the error

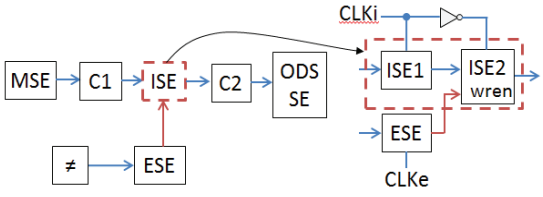

Fig. 3. Ideal ISE paradigm

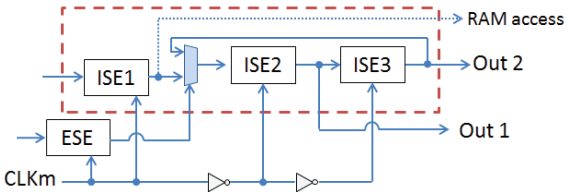

Fig. 4. Proposed ISE with 3 latches

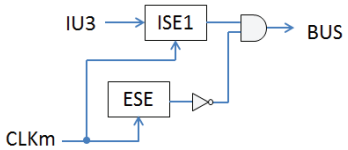

Fig. 5. Simplified ISE for bus access

signal will go through the latch ESE, and the latch ISE2 will be locked. However, this requires that the error signal reaches the latch ESE with a margin before the next rising edge of the clock so that it could lock ISE2 in time. If it does not, the error will be detected, but ISE2 may be contaminated by an error latched in ISE1.

B. *Proposed solution*

The Fig. 4 shows the proposed implementation for the first stage of the ODS area (the ISE). It uses three latches ISE1, ISE2 and ISE3. This solution enables the first stage of the ODS area to be protected against the errors from the DS part without adding a buffer stage. It preserves the last correct state of the ISE and removes the additional timing constraint α. Moreover this solution can be used with both short and long SETs detection configurations. When ISE1 and ISE3 are transparent, ISE2 is locked and vice versa. Because we should enforce $\alpha = 0$, ISE1 is locked during the high level of the clock for the short SETs detection configuration, and during the low level of the clock for the long SETs detection configuration. ISE3 preserves the last correct state, and reinjects it in ISE2 when the error signal from ESE is active. As the output of ISE can be reinjected in the DS area, the output of ISE shall be available on the same rising edge than MSE. As a consequence, the outputs Out1 and Out2 are respectively used with the short SETs detection configuration and the large SETs detection configuration. Also, the output of ISE1 can be directly connected to a memory input (RAM access) if it is allowed by the timing constraints. The write enable of the memory should also be controlled by the ESE. Thus, it avoids implementing ISE2 and ISE3. When ISE is connected to a bus, it is not always necessary to transmit the correct data if the instruction replay ensures that the same data will be written during the re-execution process. In this case, it is sufficient to write neutral data (reset) into the bus (Fig. 5).

IV. Case Study

The double sampling configured for the large SETs detection has been implemented on the integer unit IU3 of the processor LEON3 [15] along with instruction replay. The modified IU3 has nine pipeline stages that could be divided in two parts (Fig. 6). The original seven pipeline stages with some

minor modifications perform nearly the same operations as the seven original one. It is protected by DS to detect an error occurrence (DS area). Two additional buffers stages preserve the data required to start the instruction replay in case an error is detected. They are protected by TMR (ODS area). When an error is detected, the alarm signal will trigger the flush of the pipeline stages of the DS area the cycle after the alarm occurrence. Also, it will lock immediately the two stages of the ODS area as soon as the alarm appears. Thus, it avoids an error to be propagated in those two stages. The first stage of the instruction replay save area is implemented with a three latches ISE (ISE (3-L)) to protect it against the errors coming from the DS area. The second stage of the save area only contains classic flip-flops (ODS-SE). The first stage prevents the register file to be modified in case a multicycles instruction is annulated and re-executed. The data cache and the bus access, as well as the read ports of the register file, are protected by a single latch ISE (ISE (1-L)) as in Fig. 5.

The RTL design has been generated with the 28nm FDSOI technology of STmicroelectronics. The DS implementation with the basic timing constraints (6) leads to a hardware overhead of 100%. 70% of the overhead is due to the added buffer required to respect the timing constraints. The implementation of the solution proposed in [9] requires more additional buffers to remove the opportunity window. Considering that the circuit is configured to detect transient pulses of 0.5 ns, those solutions increase the hardware overhead of the DS implementation by 10% and the power consumption by 7%. On the other hand, our solution leads to 3% of hardware overhead and increase the power consumption by 8%. The global implementation (DS+ ISE+ instruction replay) increases the hardware of the original iu3 by 170% and the power consumption by 48%. If a contaminable buffer stage protected by TMR were implemented instead of our ISE, the area and power consumption overhead would be equivalent to the TMR. Compare to those solutions, we manage to reduce the hardware and the power consumption by respectively 10% and 22%.

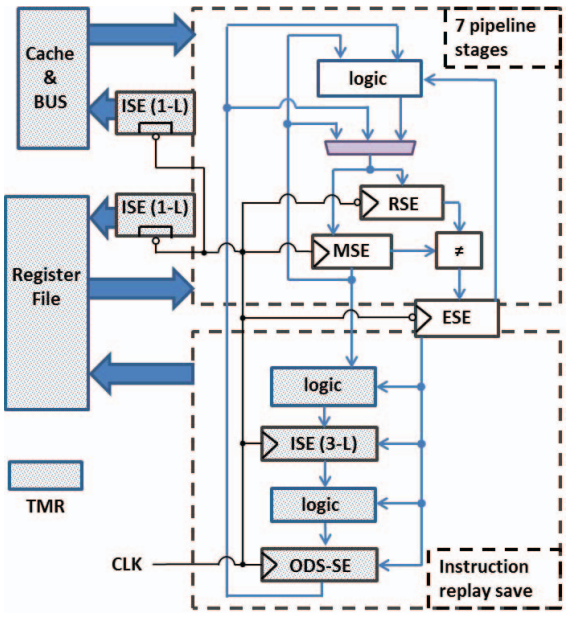

Fig. 6. Modified IU3

V. Conclusion

Double sampling is an efficient way to protect a circuit. But the signals that go out of the area protected by double sampling are vulnerable, and the solutions that use a buffer stage are costly. To reduce the cost and prevent any erroneous data to be written in the first stage which is not protected by double sampling, a solution using three latches has been proposed. We implemented the proposed solution on the integer unit of the LEON3 processor.

References

[1] M.R. Choudhury and K. Mohanram, "Low Cost Concurrent Error Masking Using Approximate Logic Circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 8, pp. 1163-1176, Aug. 2013.

[2] V. Sapozhnikov, V. Sapozhnikov, D. Efanov and A. Blyudov, "On the synthesis of unidirectional combinational circuits detecting all single faults," in *Proc. IEEE EWDTS*, 2014, pp.1-8.

[3] R. Possamai Bastos, F. Sill Torres, G. Di Natale, M. Flottes and B. Rouzeyre, "Novel Transient-Fault Detection Circuit Featuring Enhanced Bulk Built-in Current Sensor with Low-Power Sleep Mode," *J. Microelectronics Rel., Elsevier*, 52 (9-10), pp. 1781-1786. 2012.

[4] M. Nakai *et al.*, "Dynamic voltage and frequency management for a lowpower embedded microprocessor," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 28–35, Jan. 2005.

[5] K. Nowka *et al.*, "A 32-bit power PC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1441–1447, Nov. 2002.

[6] M. Nicolaidis, "Double-Sampling Design Paradigm—A Compendium of Architectures," *IEEE Trans. Device Mater. Rel.*, vol. 15, no. 1, pp. 10-23, March 2015.

[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, April 2006.

[8] K. A. Bowman *et al.*, "A 45 nm Resilient Microprocessor Core for Dynamic Variation Tolerance," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 194-208, Jan. 2011.

[9] M. Krstić, S. Weidling, V. Petrović and M. Goessel, "Improved circuitry for soft error correction in combinational logic in pipelined designs," in *Proc. IEEE IOLTS*, Platja d'Aro, Girona, 2014, pp. 93-98.

[10] Y. Miura and Y. Ohkawa, "A noise-tolerant master-slave flip-flop," in *Proc. IEEE IOLTS*, Platja d'Aro, Girona, 2014, pp. 55-61.

[11] M. Fazeli, A. Patooghy, S. G. Miremadi and A. Ejlali, "Feedback Redundancy: A Power Efficient SEU-Tolerant Latch Design for Deep Sub-Micron Technologies," in *Proc. DSN 37th Annu. IEEE/IFIP Int. Conf. on*, Edinburgh, 2007, pp. 276-285.

[12] L. Anghel, D. Alexandrescu, and M. Nicolaidis, "Evaluation of a soft error tolerance technique based on time and/or space redundancy," in *Proc. 13th Symp. on Integrated Circuits and Syst. Design*, 2000, pp. 237–242.

[13] N. D. P. Avirneni and A. Somani, "Low Overhead Soft Error Mitigation Techniques for High-Performance and Aggressive Designs," *IEEE Trans. Comput.*, vol. 61, no. 4, pp. 488-501, April 2012.

[14] A. Bouajila, J. Zeppenfeld, W. Stechele and A. Herkersdorf, "An architecture and an FPGA prototype of a reliable processor pipeline towards multiple soft- and timing errors," in *Proc. 14th IEEE Int. Symp. on Design and Diagnostics of Electron. Circuits and Syst.*, Cottbus, 2011, pp. 225-230.

[15] LEON-3 [Online]. Available: http://gaisler.com/index.php/products/processors/leon3