[Article] A novel adaptive fault tolerant flip-flop architecture based on TMR

A novel Adaptive Fault Tolerant Flip-Flop Architecture based on TMR

Luca Cassano[1,2], Alberto Bosio[3], Giorgio Di Natale[3]
[1]Dep. of Information Engineering, University of Pisa, Italy
[2]Dep. of Electronics, Informatics and Bioengineering, Politecnico di Milano, Italy
[3]Dep. de Microelectronique, Lab. d'Informatique, de Robotique et de Microelectronique de Montpellier (LIRMM), France
luca.cassano@{ing.unipi, polimi}.it, {giorgio.dinatale, alberto.bosio}@lirmm.fr

I. Introduction and Motivation

The use of *Triple Modular Redundancy (TMR)* was historically introduced long time ago for improving reliability of computer systems [1]. Recently, the advances in miniaturizing of CMOS devices made digital circuits more and more unreliable. The current trend goes towards the Internet of Things and the cloud computing, where small devices have high requirements in terms of reduced power consumption and increased reliability [2]. Classical TMR solutions allow for high reliability but they cannot satisfy low-power requirements, since they consume about three times more than the equivalent single device. However, the type of applications that are implemented in the new cloud scenario do not require high reliability all the time, but it can be assumed that some computations are more important, and thus require to be executed by a reliable hardware, while other computations are less important, and thus they can tolerate failures [3].

In this paper we propose a Novel TMR-based architecture that allow restoring the correct execution in each redundant element and where the redundant elements are dynamically activated or not based on the needed reliability level. This on-demand reliability mechanism allows reducing the overall power consumption of the system, while guaranteeing high reliability when needed. Moreover, all elements of the redundant architecture will be equally used during the functioning of the system in order to guarantee the same aging effects on the whole circuit. Further, a graceful degradation mechanism based on disabling faulty components of the architecture has been introduced. We also introduced Design for Testability feature that allows reducing by one third the test time of the whole circuit[1].

II. The proposed Adaptive Fault Tolerant Flip-Flop

The goal of the proposed work is to implement an adaptive fault tolerant mechanism that allows trading off power consumption and reliability. In a classical TMR scheme (Fig. 1a) the whole systems (combinational logic and memory elements) is triplicated and primary outputs are compared in a voter in order to correct possible errors. In the proposed architecture, the triplication is performed separately for combinational logic and Flip-Flops (FFs). Figure 1b shows the proposed design. The idea is to share a main controller in charge of on-line configuring the TMR FFs based on the needs of the application. In particular, when no reliability is required, only one combinational replica is fed by the inputs. When a higher level of dependability is required, 2 combinational replicas will be fed by the actual values and compared between them to detect possible errors. Finally, when fault tolerance is required, the 3 replicas are use at the same time. In case of error, the TMR FFs are able to correct it.

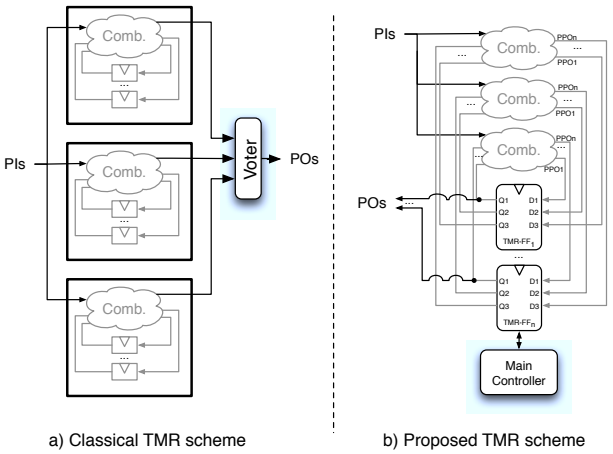

Fig. 1. The structure of a generic system using the proposed TMR architecture.

A. The Adaptive TMR Flip Flop

The idea underlying the proposed mechanism is to allow either no redundancy or error detection and/or correction mechanisms, based on the required reliability and power consumption. Figure 2 shows the proposed architecture.

An input multiplexer allows selecting the correct input data for each FF. The block *Smart Voter* (which is detailed in Figure 3) is in charge of checking the values of the three FFs and, when needed, to force the correction of the outputs via the signals C1/C2/C3 that drive the XOR gates. All the control signals come from a main controller that manages the

[1]This work has been supported by the joint FP7 Collaboration Project CLERECO (Grant No. 611404)

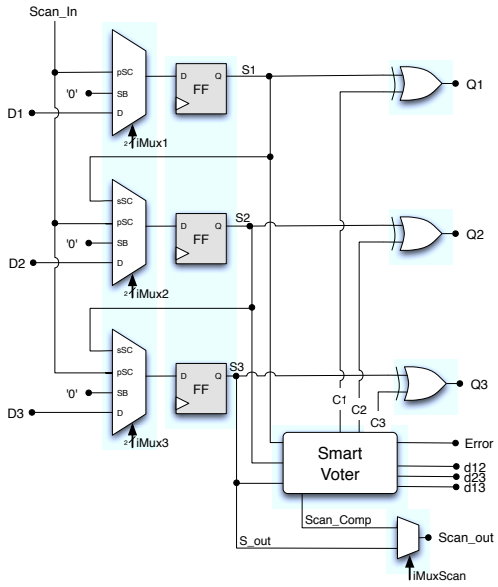

Fig. 2. The proposed TMR architecture.

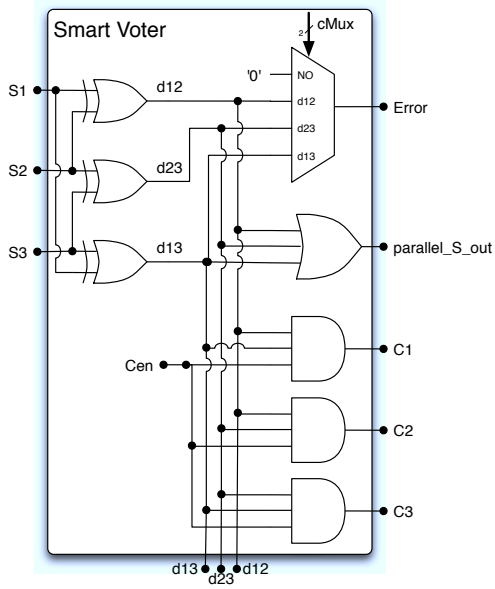

Fig. 3. The structure of the smart voter.

adaptive reconfiguration (not explicitly drawn in the figures).

When no redundancy is required, e.g., because the application does not need reliability, the circuit uses only one FF to reduce power consumption (*Single Channel* mode). The other two FFs are forced to a constant value (SB signal in the input multiplexer), i.e., they are fed by a constant value ('0' in Figure 2) so that the downstream combinational logic will not switch, thus not contributing to the dynamic power consumption. In this case, the *Smart Voter* does not enable any of the C1/C2/C3 signals, nor checks for errors. In order to be less sensitive to aging, the main controller can configure the TMR FFs to balance the use of every single FF.

When error detection is required, two out of three FFs are used (*2oo2* mode). Each FF is fed by the corresponding input data, while the third FF is placed in stand-by to reduce power consumption. In this case, the *Smart Voter* compares the states (S1, S2, S3 in Figure 3) of only the two running FFs by enabling the related cMux input. The detection of an error may lead the main controller to change configuration and to switch to the correction mode. Again, in order to balance the use of the whole circuit and therefore to balance the effect of aging, the detection configuration could be implemented by rounding the pair of used FFs.

When error correction is required, each FF is fed by the corresponding input data. The *Smart Voter* uses the differences among states (d12, d23, d13) to identify which FF stores the wrong value (*2oo3* mode). The voting is implemented by the following relation: FF_i stores a wrong value (and therefore the correction signal C_i will be asserted to invert the value of the FF) if FF_i stores a different value than FF_j and FF_k.

Signals d12, d23, d13 can be possibly used by the main controller to allow graceful degradation of the system. In particular, when an error is detected, the main controller can store the information regarding the FF that caused the problem in order to disable error correction (2oo3 mode) and to enable error detection (2oo2 mode) by using only the remaining two unfaulty FFs.

B. Design for Testability

The presence of redundant FFs and voting mechanism can be exploited to reduce the test time when scan chains are used. Indeed, instead of scanning-in test vectors in a long chain including all flip flops, it is enough to feed the 3 FFs in parallel via the pSC (parallel Scan Chain) input of the iMuxes. The output of one single FF is connected to the chain (in Figure 2, FF3 is connected to the chain via S_out). After applying the capture cycle, FFs' contents can be internally compared through the 3-input OR gate shown in Figure 3. The comparison among the 3 FFs (Scan_Comp) is then shifted out during the first cycle of shift-out, followed by regular shift operations to flush the whole response to the external tester. When no errors are revealed, the whole test response contains only '0' to state that there are no differences among the responses of the 3 FFs.

This parallel procedure reduces by one third the test application time. However, it does not allow diagnosing possible faults, nor detecting faults in the voting mechanism. To allow full diagnosticability, we have also kept a scan chain including all FFs. This chain uses the sSC ("serial Scan Chain") input of the iMuxes and flows via the S_out signal.

REFERENCES

[1] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM J. Res. Dev.*, vol. 6, no. 2, pp. 200–209, Apr. 1962. [Online]. Available: http://dx.doi.org/10.1147/rd.62.0200

[2] L. Massengill, B. Bhuva, W. Holman, M. Alles, and T. Loveless, "Technology scaling and soft error reliability," in *2012 IEEE International Reliability Physics Symposium*, 2012, pp. 3C.1.1–3C.1.7.

[3] O. Gonzlez, H. Shrikumar, J. A. Stankovic, and K. Ramamritham, "Adaptive fault tolerance and graceful degradation under dynamic hard real-time scheduling," in *Proceedings of the 18th IEEE Real-Time Systems Symposium (1997)*, 1997, pp. 79–89.