International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

A Soft-Error Tolerant Design for TMR Implementation in Real Time Systems

Revathi. P[1], Sathya. S .S[2]

[1]PG Student, M.E Communication Systems, Department of Electronics and Communication Engineering, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India.
[2]Assistant Professor, Department of Electronics and Communication Engineering, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India.

Abstract - *The Dual modular redundancy (DMR) uses two identical modules and a voting circuit, is an existing architecture for soft-error tolerance. To overcome these designs, a complementary dual-modular redundancy (CDMR) implemented through two-stage TMR(triple modular redundancy) is a proposed technique for tolerating many soft errors. The two-stage TMR includes two stages: primary and supplementary stage. Two copies are executed in the primary stage and their results are matched, it discards the supplementary stage execution saving its energy. As a result, the proposed method reduces the area overhead by tradingoff reliability and vulnerability of the system.*

***Key Words*: Soft-error tolerance, CDMR, Two-stage TMR, Soft errors and Reliability.**

1. INTRODUCTION

Hard real-time systems presently appear in various application areas, from medical operations to space systems. Computer systems optimized for the execution of such applications should offer high criteria of reliability, because a system failure may result in costly disastrous events. Hence, to avoid such failures, the system should tolerate potential faults, either induced by environmental disorders or issued as a result of hardware aging. Redundant execution is a common approach to achieve fault tolerance and provide desirable reliability. Redundant execution refers to multiple executions of the same task for the sake of decoupling the results of flawed executions from the outcome of the system.

Fault tolerance means basically ensuring the correct operation instead of fault occurrence, and provides significance of higher reliability. Occurrence of a fault can be internal or external to a function module, but it should not affect the actual output of that function module. If it is ensured, then the fault is said to be masked or successfully hidden from the observer by the outside world. On the contrary, if the fault is not masked, the desired output would be affected from the expected function module by causing an erroneous output to be produced instead of producing the correct output. Hence, faults do not cause an error are said to be masked (concealed) and on the other hand, faults that result in an error are said to be exposed (revealed).

In short, the manifestation of a fault is interpreted as an error. Thus in the electronic systems, occurrence of faults has become common and it is reasonably desirable to incorporate fault tolerant techniques into some designs to ensure reliability. Since adders are the key processing unit of a processor, it becomes necessary to guarantee fault free operation of these units.

Simple faults mainly labelled as the Transient/temporary faults are also called as soft errors because they are correctable. At the logic level, soft errors might tend to get result as single-event effects. Single-event transients (SETs) occur due to high-energy particle strikes, might cause a bit-flip at a gate output node or in interconnects formed between logic elements. However, a transient fault occur as an error depends upon the electrical, logical, and timing masking of the design. Transient faults can be overcome through radiation hardening of underlying combinational and sequential logic and memory elements by employing the redundancy technique.

Triple Modular Redundancy (TMR) is a redundancy technique used as a long-time and historical approach for masking various kinds of faults. By employing redundancy and analysing the results of three separate executions of the same program, TMR is able to attain excellent levels of reliability. While TMR provides a desirable level of reliability, a severe detriment to its broad adoption suffers from high power consumption of the redundant hardware.

2. RELATED WORKS

The literature survey focus about the soft-error in combinational based, voter circuit and logic based circuit, etc. To study about the soft error on the circuits with their results help to analysis the type of error occurred and the techniques needs to tolerance. Soft-error tolerance is an important problem that is needed to be addressed at the circuit, architectural, and algorithmic levels. In all cases, enhancing the robustness of systems and circuits to soft errors introduces redundancy in some form. Doing so results can produce efficient area and power overhead. Therefore, there is a fundamental tradeoff between energy efficiency

and robustness. The first survey describes about the types of soft errors and the methods to reduce the errors based on the algorithm and logic. The other survey describes about the error based design and solutions for circuits are defined.

Yufeng, et.al,.[1], inspired by the Markov Random Field (MRF) theory, a two-stage voting system is proposed in the Complementary Dual Modular Redundancy (CDMR). In MRF based design, the basic elements include feedback loops which help them to achieve high soft-error tolerance. To solve soft error issues in the voter and save area overhead, proposed a new complementary DMR (CDMR) scheme. The MRF design is implemented and produce a NAND-NAND based feedback structure. The stage 2 structure can complement the loss of error tolerance for the first stage using its latch property. For multistage voting design, the voter is added at each stage to improve the overall system reliability. The voter has two outputs without voting duplication with enclosed feedback loops. This produces two complementary outputs as references for error correction.

Lizheng, et.al,.[2], The Autonomous Error Tolerant (AET) architecture aims to have a self-repairing capability and tests the performance variation of AET system under different reliability requirements. The AET structure is designed by the nearby error sensing mechanism to detect the errors timely, backup circuit switching strategy is used to bypass the failed nodes and active evolution scheme is studied to handle unrecoverable errors. The board-level prototype is used based on dual-core embedded processors. The analysis show that the error tolerant capability is better than the conventional multi modular redundant system.

Robert, et.al,.[3], a dynamic memory core based on Gain-Cell (GC-eDRAM) is used, which is more susceptible to soft errors than a static memory cell. It is a fully logic-compatible implementation which provides a reduced silicon footprint compared with SRAM, but lacks the internal feedback that ensures strong storage levels. Hence, it is used with a circuit-level SEU tolerance by entirely removing the feedback. The transistor count with reduced amount and the physical properties of the dynamic circuit allows to internally apply complementary DMR (CDMR) to achieve per-bit error detection and per-bit error correction.

Ahmad, et.al,.[4], a selective-transistor scaling method is used that protects individual sensitive transistors of a circuit. A sensitive transistor is a transistor whose soft error detection probability is relatively high used until desired circuit reliability is achieved or a given area overhead constraint is met. Transistor duplication and asymmetric transistor sizing are the methods used to applied to protect the most sensitive transistors of the circuit. In asymmetric sizing, nMOS and pMOS transistors are in independent size. Reliability is performed with different protection thresholds and area overhead constraints. Finally, a novel gate level soft error reliability evaluation technique for combinational circuits is proposed that produces similar results as produced by transistor-level simulations.

Amin, et.al,.[5], presents a fast and efficient technique for resizing the large-scale combinational circuits. In this methodology, the circuit is partitioned into a set of smaller sub circuits based on the structures which are originated from the primary outputs (POs). The extracted sub circuits are topologically levelized and then, by starting from the minimum level, the sub circuits located at the same level are resized individually and independently. This procedure is continued level by level until the sub circuits in all levels are resized. During resizing process, after each gate sizing, the sub circuit error probability (SEP) of the sub circuit in which the resized gate is located is computed. Based on the computed SEP, the effects of gate sizing on the total circuit SER are evaluated. Since the SEP computation is much faster than circuit SER computation, evaluating the effects of gate sizing on circuit SER locally by the SEPs results in significantly accelerating in the gate sizing optimization algorithm. This technique is evaluated for various large scale circuits and compared the obtained results with the similar gate sizing optimization approaches.

In the designs, the basic elements include feedback loops which help them to achieve high soft-error tolerance. However, these implementations require higher area overhead with low reliability. To solve soft-error issues and save area overhead, proposed a new complementary DMR (CDMR) scheme; The CDMR scheme ensures the signifies of soft-error tolerance even for the voting circuit. This can be achieved by separately processing one module (M1) through a structure with a stable logic "1" as output, and processing another identical module (M2) through a structure with a stable logic "0" as output. Another feedback structure is then used to merge the stable logic "1" and stable logic "0" outputs from the first stage. The CDMR scheme performs the existing designs in two aspects: 1) tolerating soft errors when propagated to the voting circuit and 2) saving the area overhead.

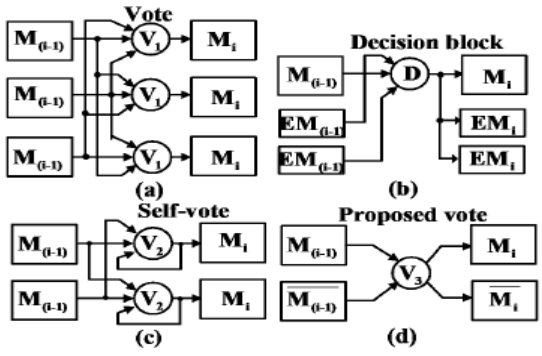

Fig -1: Conventional Voting structure in multistage design (a) TMR (b) FGSET (c) DMR (d) existing voting module

The existing design has achieved soft-error tolerance but TMR can only tolerate soft errors when the probability of three or two modules failing at the same time is much lower than that of a single module.

For multistage logic, the voter is concatenated at each stage to improve the overall system reliability, as shown in Fig. (a)–(d). The original TMR, FGSET, and DMR voters for multistage are simply duplicated [refer to Fig. (a)–(c)]. However, the proposed voter has two outputs without voting duplication between two stages and computed with the enclosed feedback loops, as shown in Fig. (d). Note that two complementary outputs as references for error correction are obtained.

3. PROPOSED SYSTEM

In this, CDMR is performed to increase the ability of the design for saving the area overhead propagating with multiple module system. The MRF design is implemented in CDMR by substituting an inverting module for one of the identical modules. The MRF based design is based on the energy function specifically the clique energy. The CDMR ensures the soft error tolerance which is implemented through two stage TMR.

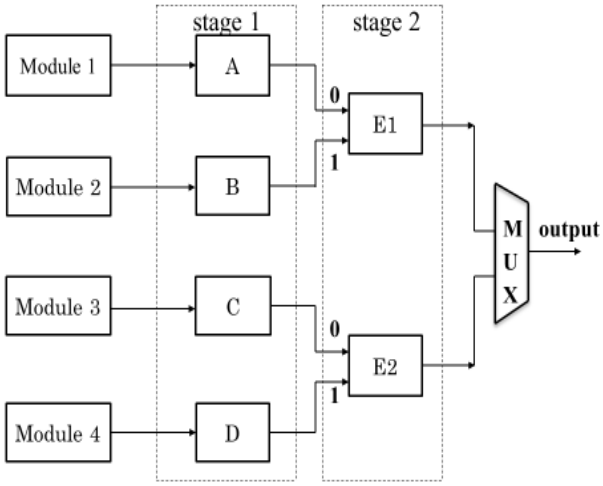

Fig -2: CDMR design with Multiplexer

3.1 IMPLEMENTING TWO-STAGE TMR

As it is obvious, conventional TMR can produce a system without redundancy and results in tremendous energy overhead. Two-stage TMR tries to overcome this obstacle by considering the dominance of fault-free execution. As transient faults rarely occur, two-stage TMR partitions the operations of conventional TMR into two stages: primary stage and supplementary stage. Two copies of each task are scheduled for execution in the primary stage, and the third copy is mapped to the supplementary stage. In the primary stage, two copies are executed, and their results are compared. If the results are identical, the system does not execute the third copy since its result cannot affect the final result. Therefore, the third copy is safely dropped for the sake of saving precious energy. Because of the infrequency of transient faults, most of the time the results of two executions in the primary stage match, and consequently, the two-stage TMR discards the supplementary stage execution, saving its energy. An example of executing a single task under two-stage TMR is depicted.

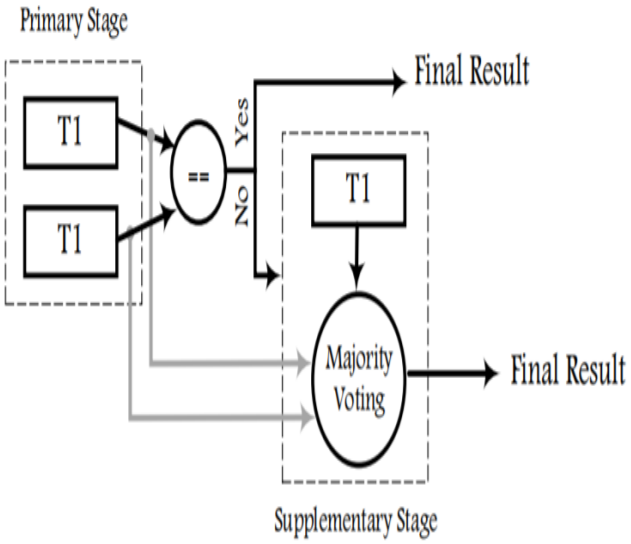

Fig -3: Two Stages TMR

Note that two-stage TMR does not double the Worst-Case Execution Time (WCET) as compared to the conventional TMR. The reason is that the supplementary stage is statically determined to operate at maximum voltage and frequency (i.e., minimum latency); hence its latency is much less than that of the primary stage. Finally, as the supplementary stage is rarely executed (because of the less incidence of transient faults), the significant energy overhead associated with this stage (which comes from operating at a high voltage), does not considerably affect the total energy consumption of the whole execution at long-term.

3.2 FAULT COVERAGE OF THE PROPOSED ALGORITHM

The transparent SOA-MATS++ algorithm is intended for test of stuck-at fault, transition fault, and read disturb fault used for tests developed during field operation of FIFO memories. The fault coverage of the algorithm is shown. In the both figures, the word size of FIFO memory is assumed to be of 4 bits. The text against the arrows indicates the operation performed and also indicates the text corresponds to the variables used.

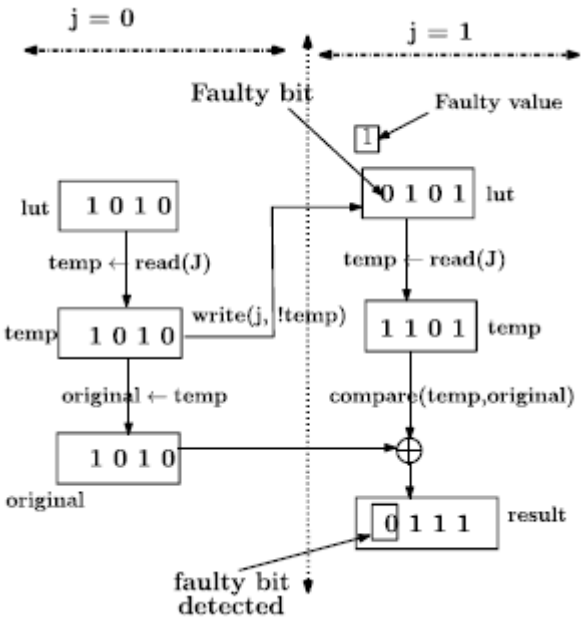

Fig -4: Proposed algorithm

As shown in Fig, assume the data word present in *lut* be 1010. The test cycle begins with the invert phase (memory address pointer *j* with 0 value) during which the address of the content location is read into *temp* and then backed up in the *original*. The data which is written back to *lut* is the complement of content of *temp*. Thus, at the end cycle, the data present in *temp* and *original* is 1010, while *lut* contains 0101. Assume a stuck-at-1 fault at the most significant bit (MSB) position of the lut in which the word stored Thus, instead of storing 0101, it actually stores 1101 and as a result, the stuck-at-fault at the MSB will get excited.

During the second iteration of *j*, when *lut* is readdressed, the data read into *temp* is 1101 and at this point, the data present in *temp* and *original* are compared (bitwise XORed). An all 1's pattern is expected as result. Any 0 within the pattern is considered a stuck-at fault at that bit position. This situation is shown in Fig, where the XOR of 1010 and 1101 yields a 0 at the MSB position of the *result* indicating a stuck-at-fault at the MSB position. However, the initial data for a bit position is different from the faulty bit value and the stuck-at-fault cannot be detected for the bit position after the restore phase of the test. Therefore, it requires one more test cycle to excite such faults.

At the third iteration of *j,* it performs only a read operation of *lut*. Then the contents of *lut* is read into *temp* and compared with the contents of *original*. At this stage of the test, the result of all 0's pattern signifies fault free location, while deviation at any bit position from all 0's pattern means fault at that particular bit position. The last read operation ensures the detection of faults, which remains undetected during the earlier two test runs. At the end of the three test runs (iterations of *j*), the loop index *i* is incremented by one for the start of test to mark the next location.

4. EXPERIMENTAL RESULTS

To demonstrate the inherent ability of the error tolerance in the voting circuit, the output shows consistent results at the final stage and the desired outcome is obtained. As shown in the Fig. there is a propagation of a signal injected by a soft error is kept to be hold and then recomputing a error bit, it produces the correct output. By this, it is used to redundant the errors while it is propagating to the internal or external of the system.

Experimental conditions include the simulation tool of Xilinx and the result is viewed in Modelsim, Any Intel x64 Processor, CPU 2.2 GHz min, RAM Memory 2048 MB and 32-bit Microsoft Windows Operating System. In this project, the proposed scheme is implemented with 8 x 8 router handling the error tolerant design. The input bits are applied to the router with the proposed redundancy technique. It improves the performance of the system by enabling the inputs to a desired output. In addition, these techniques have less impact on timing performance.

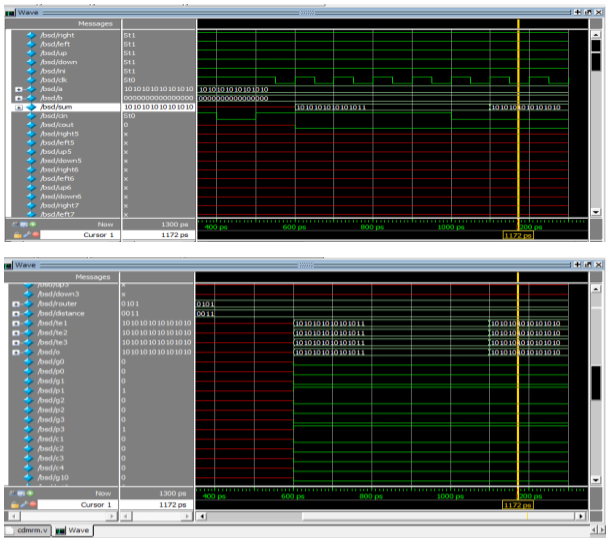

Fig -5: Error tolerance of the simulation results

Explanation: The input level of 1s and 0s bits are used. When the input is increased to be changed, the output will also be changed. If any input to be missed in the circuit, then the error occurs.When the input level is increased then the bits are added and it produces the correct output which reduces the errors in the redundancy circuit.

5.CONCLUSION

The Complementary Dual modular redundancy (CDMR) with multiplexer modules is proposed for soft-error tolerance. In the proposed system CDMR is implemented through two-stage TMR to improve the reliability of the design and area overhead performance. The design was implemented by Xilinx and Modelsim to achieve better soft-

error tolerance. In the future work of the project, compared to MRF – inspired design some other techniques could be used to increase the testability of the design and improve the error tolerance with the multiple system.

REFERENCES

[1] Yufeng Li, Jianhoa Hu, Xuang Zeng, Bruce Cockburn (2018), 'Feedback-Based Low-Power Soft-Error-Tolerant Design for Dual-Modular Redundancy', IEEE Transaction on Very Large Scale Integration Systems, Vol.26, No.8, pp.1585-1589.

[2] Lizheng Liu, Ning Ma, Yuxiang Huan (2017), 'A Design of Autonomous Error-Tolerant Architectures for Massively Parallel Computing', IEEE Transaction on Very Large Scale Integration Systems, Vol.26, No.10, pp.2143-2154.

[3] Robert Giterman, Lior Atias and Adam Teman (2017), 'Area and Energy- Efficient Complementary Dual-Modular Redundancy Dynamic Memory for Space Applications', IEEE Transaction on Very Large Scale Integration Systems, Vol.25, No.2, pp.502-509.

[4] Ahmad T. Sheikh, Aiman H. El-Maleh and Sadiq M. Sait (2017), 'A Fault Tolerance Technique for Combinational Circuits Based on Selective-Transistor Redundancy', IEEE Transaction on Very Large Scale Integration Systems, Vol. 25, No.1, pp.224-237.

[5] Amin Sabet, Behnam Ghavami, and Mohsen Raji (2017), 'A Scalable Solution to Soft Error Tolerant Circuit Design Using Partitioning-Based Gate Sizing', IEEE Transaction on Reliability, Vol. 66, No.1, pp.245-256.

[6] Aiman H. El-Maleh, and Khaled A. K. Daud (2015), 'Simulation-Based Method for Synthesizing Soft Error Tolerant Combinational Circuits', IEEE Transaction on Reliability, Vol. 64, No.3, pp.935-948.

[7] Byonghyo Shim, and Naresh R. Shanbhag, Fellow (2006), 'Energy-Efficient Soft Error-Tolerant Digital Signal Processing', IEEE Transaction on Very Large Scale Integration Systems, Vol. 14, No.4, pp.336-348.

[8] Gehad I. Alkady, Ali AbdelKader, Ramez M. Daoud, Hassanein H. Amer (2016), 'Integration of Multiple Fault-Tolerant Techniques for FPGA-Based NCS Nodes', International Conference on Computer Engineering and Systems (ICCES).

[9] Hsuan-Ming Chou, Ming-Yi Hsiao, Yi-Chiao Chen (2015), 'Soft-Error-Tolerant Design Methodology for Balancing Performance, Power, and Reliability', IEEE Transaction on Very Large Scale Integration Systems, Vol.23, No.9, pp.1628-1639.

[10] I-Chyn Wey, Chien-Chang Peng, and Feng-Yu Liao (2015), 'Reliable Low-Power Multiplier Design Using Fixed-Width Replica Redundancy Block', IEEE Transaction on Very Large Scale Integration Systems, Vol.23, No.1, pp.78-87.

[11] Mohammed Hadifur Rahman, Shahida Rafique, Mohammad Shafiul Alam (2015), 'A Fault Tolerant Voter Circuit for Triple Modular Redundant System', Journal of Electrical and Electronic Engineering, Vol.5, No.15, pp.156-166.

[12] Mohammad Salehi, Alireza Ejlali and Bashir M. Al-Hashimi (2016), 'Two-Phase Low-Energy N-Modular Redundancy for Hard Real-Time Multi-Core Systems', IEEE Transaction on Parallel and Distributed Systems, Vol.27, No.5, pp.1497-1510.

[13] Mojtaba Ebrahimi, Hossein Asadi and Mahdi Fazeli (2013), 'Low-Cost Scan-Chain-Based Technique to Recover Multiple Errors in TMR Systems', IEEE Transaction on Very Large Scale Integration Systems, Vol.21, No.8, pp.1454-1468.

[14] Rahul Parhi, Chris H. Kim, Keshab K. Parhi (2015), 'Fault-Tolerant Ripple-Carry Binary Adder using Partial Triple Modular Redundancy (PTMR)', IEEE International Symposium Circuits Systems (ISCAS), Lishon, Portugal, pp.41-44.

[15] Yuan-Hao Huang (2010), 'High-Efficiency Soft-Error-Tolerant Digital Signal Processing Using Fine-Grain Subword-Detection Processing', IEEE Transaction on Very Large Scale Integration Systems, Vol.18, No.2, pp.291-304.