A Study Of Fault Tolerance In High Speed VLSI Circuits

Somashekhar, Dr.Vikas Maheshwari, Dr.R. P. Singh

Abstract: The main motive for introducing fault tolerance in VLSI circuits is yield enhancement, by increasing the percentage of fault free chips. In nm technologies, circuits are more and more sensitive to a variety of perturbations. Transient faults can take place in a processor as a result of electrical noise, and alpha particles. These faults are able to cause a program running on the processor to behave inconsistently, if they propagate and change the architectural state of the processor. These faults can occur in memory arrays, sequential elements or in the combinational logic in the processor. Protection against transient faults in combinational logic has not received much attention traditionally because combinational logic has a natural barrier stopping the propagation of the faults. This paper presents fault tolerance in VLSI circuits.

Keywords: VLSI, Fault Tolerance, Errors, CED, Watchdogs, Processor, Processing Element, Wafer, TMR and DMR.

INTRODUCTION

Faults in a scattered embedded system can be permanent, intermittent or transient. Permanent faults cause long-term malfunctioning of components. These faults emerge for a short time. Causes of intermittent faults are within system boundaries, while causes of transient faults are external to the system. They might damage data or lead to logic miscalculations, which can outcome in a fatal failure. Due to their higher rate, these faults cannot be addressed in a cost-effective way by applying traditional hardware-based fault tolerance techniques suitable for tolerating stable faults. Embedded systems along with fault tolerance have to be carefully designed and optimized, in order to assure strict timing requirements without exceeding a definite limited amount of resources. Moreover, not only performance and cost related requirements have to be considered but also other issues such as debug ability and testability have to be taken into account.

ERRORS IN VLSI CIRCUITS

Failures in VLSI systems might result from varied types of faults that can be classified as either soft (transient) or hardware ones. Transient faults are induced by temporary environmental surroundings, such as cosmic rays, EMI and for example cause information alteration in memory elements. Permanent faults are the result of irreversible device and circuit changes, such as the following: Electromigration, which causes thinning and eventual open circuit of metal tracks. Hot carrier effect, which causes shift in device threshold voltage and it does convey conductance. Time dependant dielectric breakdown, which causes gate oxide to substrate short circuit.

FAULT TOLERANCE OF VLSI DESIGNS

Increasing the yield of ICs proves especially important for new designs and manufacturing processes, which have a high density of process-induced defects and consequently a low yield. Yield improvements of early prototypes of an IC can reduce the product’s introduction time and determine its commercial success. Defect tolerance has proved successful in such cases, and spectacular 30-fold increases in yield have been reported. Yield improvements due to defect tolerance tend to decrease as the manufacturing process matures. But even mature processes with lower defect densities have experienced 1.5 to 3 fold yield increases, proving the effectiveness of defect- tolerance techniques[5]. Logic ICs: The development of efficient defect tolerant designs for random logic ICs like microprocessors is considerably more complex than for memory ICs. However, if some regularity exists in the structure of a given logic circuit, it might be possible to incorporate redundancy. A natural target for defect tolerant designs - programmable logic arrays (PLAs) have a regular structure in VLSI chips. The control sections of many microprocessors use large PLAs. Some de- signs have employed PLAs with as many as 50 inputs and almost 200 product terms. Since these PLAs require large silicon areas, the incorporation of redundancy in their design can considerably improve the overall yield. Researchers have investigated defect tolerant designs of PLAs and proposed adding spare programmable product lines, input lines, and output lines to protect against all types of possible defects. This technique resembles the redundant row/column scheme for memory ICs. However, unlike memory ICs, where all defects can be identified by applying test patterns outwardly, the recognition of defects in a PLA re- quires some built-in testing aids, like adding inputs to the AND plane. Defect-tolerant PLAs with spare programmable lines and extra inputs for fault identification have recently been implemented within a 16-bit microprocessor.’ This microprocessor also includes a defect-tolerant data path. A microprocessor’s data path includes ALU units, registers, and buses and usually occupies a large percentage of the overall area. A bit-sliced data path, with the inclusion of one or more spare slices, can use the

- *Somashekhar, Research Scholar, Dept. of ECE, SSSUTMS, Sehore (M.P), India, Email: somashekhar49@gmail.com.*
- *Dr.Vikas Maheshwari, Associate Professor, Dept. of ECE, BIET, Hyderabad. India, Email: maheshwarivikas1982@gmail.com.*
- *Dr. R. P. Singh, Vice-Chancellor & Professor, Dept. of ECE, SSSUTMS, Sehore (M.P), India, Email: provc@sssutms.co.in.*

regularity in the circuit. However, not all parts of the data path are regular. For example, the logic circuits associated with the status bits are highly irregular. Since we cannot replace such parts with common spare circuits, we must exclude them from the bit-slice organization. In logic ICs with irregular structures, duplication or even triplication of certain circuits might prove beneficial. If we use duplication, we must employ fault identification and then restructuring after manufacturing. In the case of triplication, we can avoid these additional steps by using a majority voter at the output, if only one defective circuit out of the identical ones is allowed to fail. In its attempt to build a mainframe based on wafer scale emitter coupled logic technology, Trilogy employed replication for defect tolerance in random logic. However, the extremely large overhead associated with these techniques has substantially limited their use in general. Several defect-tolerance approaches have been proposed for processor arrays. In one of the earliest schemes, the row and column exclusion approach, redundant rows and columns of processors are implemented in the array. Each processor converts into a connecting element (CE) if required. If so, it no longer performs computational operations but merely passes signals from input to output. If a processor in the array fails, all other processors in the row and column containing the failed processor become CEs, as shown in Fig.1. A reconfigured fault- free array with one less row and column results. The failure of a horizontal (vertical) link between two adjacent processors requires turning only the processors along the corresponding row (column) into CEs. While attractive in its simplicity and low hardware overhead, the above scheme only proves effective when we expect few failures.

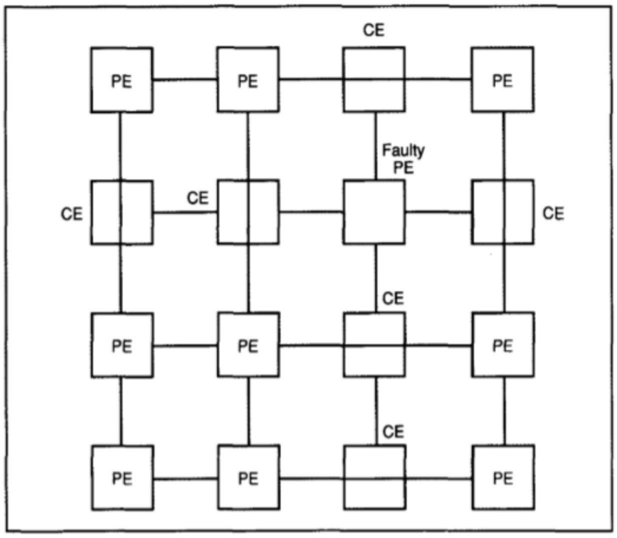

Fig.1. Row and column exclusion scheme

PE-processing element, CE-connecting element. Since an entire row and column must usually be disabled for each fault, multiple faults quickly degrade the array. To get around this problem, several other restructuring schemes have been proposed. Designers considering a defect tolerance technique for a VLSI circuit must estimate the projected yield. This allows them to determine the optimal amount of redundancy and the suitability of a proposed re- configuration scheme. The difficulty in modeling the yield of fault-tolerant IC chips arises mainly from the clustering of manufacturing defects during chip fabrication. Yield modeling proves relatively simple when we use Poisson statistics to describe the distribution of faults per chip. According to this distribution, the probability of having exactly x faults in a chip is given by

$$\text{Prob}\{X=x\}=e^{-\lambda}\lambda^{x}/x!$$

where X is a random variable denoting the number of faults and h denotes the average number of faults expected per chip. For chips with no redundancy the yield is

$$Y = \text{Prob}\ \{X = 0\} = e^{-\lambda}$$

DESIGN OPTIMIZATION

protection critical applications contain strict time and cost constraints, which means that not only faults have be to tolerated and it also imposed constraints have to be fulfilled. Usually, hardware replication was used as a fault tolerance technique against transient faults.

Fault Tolerance Techniques: In order to get fault tolerance, a primary requirement is that transient faults have to be detected. Researchers have proposed several error-detection techniques against transient faults: watchdogs, assertions, signatures, duplication, and memory protection codes.

Signatures: Signatures are one of the most powerful error detection techniques. In this technique, a set of logic operations can be assigned with pre-computed -check symbols‖ that indicate whether a fault has happened during those logic operations. Signatures can be implemented either in hardware, as a parallel test unit, or in software.

Concurrent error detection technique (CED): CED in processor control logic technique takes benefit of the fact that transient faults in gates along some paths are much more likely to propagate to an architectural state under normal running of the processor than others and protects against errors in these paths. The technique automatically extracts the control conditions under which these paths are sensitized and converts these conditions into assertions [1].The CED technique protects the combinational portion of the circuit against transient errors. Fig.2 shows a circuit which has CED capability. The technique introduces an Assertion Checker which takes as inputs and the outputs of the combinational circuit and which gives out a signal whether they conform to a series of assertions.

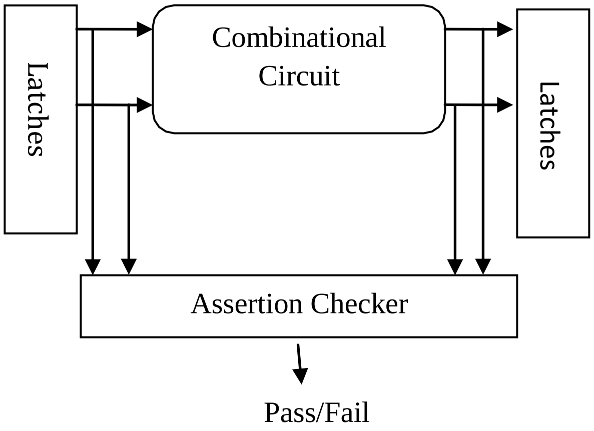

Fig.2. Block diagram of CED

Time Redundancy:

Redundancy is necessary in self checking system. Time redundancy is a self checking approach and is used to detect the transient fault. The block diagram of time redundancy is shown in fig.3. This method used the duplicate hardware in addition to the original hardware to perform the same operation at different interval of time. The fault is detected by comparing the outputs of both original and duplicate hardware. If the outputs of both the hardware are found to be same, it represents fault-free condition. However, if the outputs of both the hardware are different, it represents the faulty condition. If the first calculation result is faulty and it is used for other computations, also makes the subsequent modules faulty.

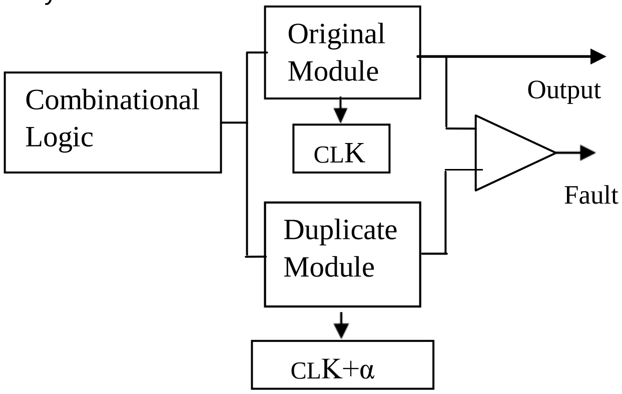

Fig.3. Time Redundancy technique

Hardware Redundancy:

The commonly used hardware redundancy approaches are Triple modular redundancy (TMR) and Double modular redundancy (DMR). TMR method is used to detect the single fault. The name itself indicates three so it requires three similar modules in parallel to detect the fault. A fault is detected if anyone output of the modules are different. But this method is not able in demonstrating the exact location of the fault. The block diagram of DMR is shown in the fig.4.

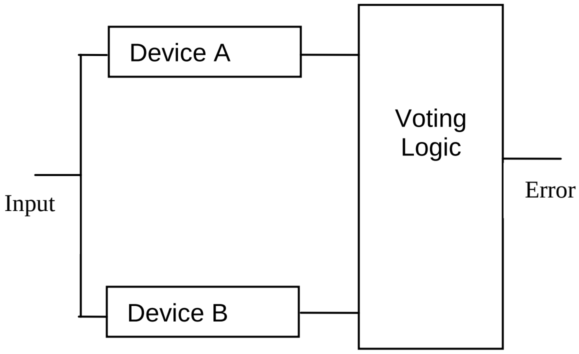

Fig.4. Block diagram of DMR

Double modular redundancy approach is used to detect the single fault at a time by comparing the outputs of operation performed by the original hardware and duplicate hardware in parallel. The block diagram of TMR is shown in the fig.5. It consists of 3 devices namely device A, device B and device C and voting logic. The voting logic is the majority voter which takes the majority of inputs to be the output value. Device B and Device C are replication of Device A and they all accept the same input value, the output of A, B and C should be consistent in theory. Because of fault in the system, one of these devices might have an error within and generate a dissimilar output. This inconsistency will be caught and corrected by voting logic [7].

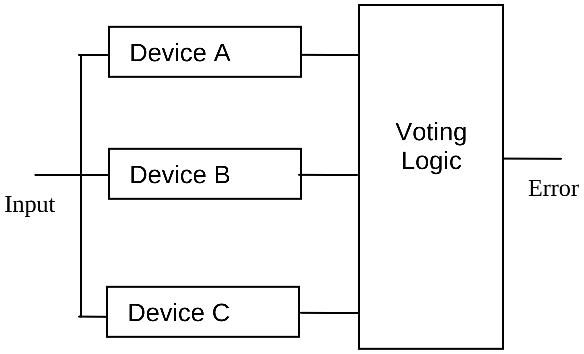

Fig.5. Block diagram of TMR

CONCLUSION

In this paper, the various fault tolerant techniques are discussed. When the complexity of IC is increasing with the advancement of technology, results in reducing the size of an integrated circuit. This makes the design more compact and sensitive to the transient faults. This paper describes about the fault tolerances in VLSI circuits. The presence of the faults may destroy the functionality of the system.

REFERENCES

[1] M. Maniatakos and P. Kudva, “Low-cost concurrent error detection for floatingpoint unit (fpu) controllers,” IEEE Transactions on, vol. 62, no. 7, pp. 1376– 1388, 2013.

[2] V. Chandra and R. Aitken, “Impact of technology and voltage scaling on the soft error susceptibility in nanoscale cmos,” in Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS ’08. IEEE International Symposium on, Oct. 2008, pp. 114–122.

[3] B. Sierawski, B. Bhuva, and L. Massengill, “Reducing soft error rate in logic circuits through approximate logic functions,” Nuclear Science, IEEE Transactions on, vol. 53, no. 6, pp. 3417–3421, Dec. 2006.

[4] Pandeeswari.R, Mohamadi Begum “Rsfts: Rule-Based Semantic Fault Tolerant Scheduling For Cloud Environment”, Council for Innovative Research International Journal of Computers & Technology. Volume 4 No. 2, March-April, 2013.

[5] I. Koren, “A Reconfigurable and Fault-Toler- ant VLSI Multiprocessor Array,” Proc. 8th Ann. Symp. Computer Architecture, May 1981, pp. 425-441.

[6] Somashekhar, Vikas Maheshwari, R. P. Singh, “Analysis of Micro Inversion to Improve Fault Tolerance in High Speed VLSI Circuits”, International Research Journal of Engineering and

Technology (IRJET) Volume 6, Issue 3, March 2019.

[7] Samudrala et al.,“Selective triple modular redundancy based single-event upset tolerant synthesis for FPGAs”, Nuclear Science, IEEE Transactions on, Vol.51,2004.