International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

Design an High speed Digital Fault Tolerant Architecture

Goga Subramanyam & Sk. Tajuddin

[1]M-Tech Embedded VLSI, Assistant-Professor, Rise Krishna Sai Gandhi Group of Institution A.P
[2] M-Tech, Assistant-Professor, CMOS Research Labs Vijayawada, AP

ABSTRACT:

In the era of deep sub-micron technology, probability of chip failure has been increased with increase in chip density. A system must be fault tolerant to decrease the failure rate and increase the reliability of it. Multiple faults can affect a system simultaneously and there is a trade-off between area overhead and number of faults tolerated. This paper presents high speed fault tolerant architecture design for digital applications. The fault containment and parallel processing capabilities of computers network are being exploited to provide a high performance, high availability network capable of tolerating a broad scope of hardware, software, and operating system faults.

Keywords—Fault tolerant**;** self-reconfigurable**;** ripple carry adder**;** conditional sum adder**;**
Double fault**.**

I. INTRODUCTION

Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system in which even a small failure can cause total breakdown. Fault tolerance is particularly sought after in high-availability or systems. A fault-tolerant design enables a system to continue its intended operation, possibly at a reduced level, rather than failing completely, when some part of the system fails. The term is most commonly used to describe computer systems designed to continue more or less fully operational with, perhaps, a reduction in throughput or an increase in response in the event of some partial failure. That is, the system as a whole is not stopped due to problems either in the hardware or the software. An example in another field is a motor vehicle designed so it will continue to be drivable if one of the tires is punctured. A structure is able to retain its integrity in the presence of damage due to causes such as fatigue, corrosion, manufacturing flaws, or impact. Within the scope of an individual system, fault tolerance can be achieved by anticipating exceptional conditions and building the system to cope with them, and, in general, aiming for self-stabilization so that the system converges towards an error-free state. However, if the consequences of a system failure are catastrophic, or the cost of making it sufficiently reliable is very high, a better solution may be to use some form of duplication. In any case, if the consequence of a system failure is so catastrophic, the system must be able to use reversion to fall back to a safe mode. This is similar to roll-back recovery but can be a human action if humans are present in the loop.

The most common arithmetic unit in a digital processor is an Adder. Different types of adders are available in practice. The speed of digital arithmetic processor depends on the choice of high-performance sub-modules, among which adder is the most essential one, not only for addition, but also for subtraction, multiplication, and division. Different fast adding topologies using carry speed up techniques are also available in literature.

Fault tolerance is defined as the ability to continue operating after the failure of a given system component. To be fault tolerant, a system must have one or more redundant components that can take over the function when the primary component fails. In addition, the system must have both a means of Detecting failures in the components and a means of transferring to working components after a failure has been detected. Fault-tolerant system configurations are used extensively in processes where the system must remain on-line in the event of component failure. Although applications are widespread, industrial processes, aerospace vehicles, and ground transportation are especially noteworthy. The need for optimization in the design of these systems is apparent when one thinks of the complexity of modern spacecraft. The high costs associated with their fabrication and launch dictate that any design proposal be assured a very high probability of success at the lowest possible system cost.

II. FAULT MODELING

Reliability and availability have become increasingly important in today‟s computer dependent world. In many applications where computers are used, outages or malfunction can be expensive, or even disastrous. Just imagine the computer system in a nuclear plant malfunctioning or the computer systems in a space shuttle booting just when the shuttle is about to land... These are the more exotic examples. More close to everyday life are the telecommunications switching systems and the bank transaction systems.

High area overhead is a major concern in many applications like satellite or defence, where reliability is also another major issue. Hence we have adopted dynamic recovery technique as our fault tolerant methodology. Primary requirement for applying dynamic recovery technique is that the system must have structural regularity. To achieve the needed reliability and availability, we need fault-tolerant computers. They have the ability to tolerate faults by detecting failures, and isolate defect modules so that the rest of the system can operate correctly. Reliability techniques have also become of increasing interest to general-purpose computer systems.

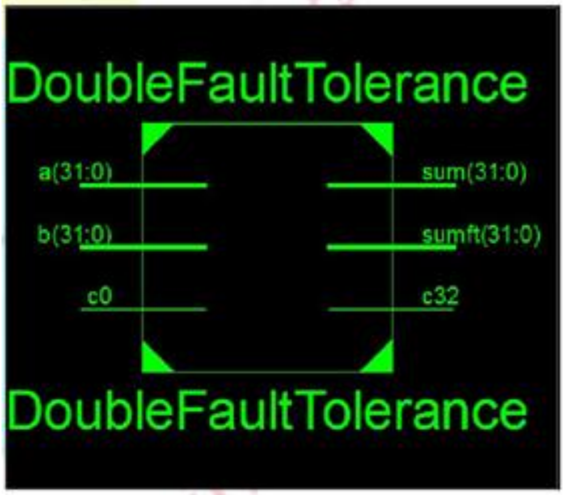


Fig. 1 Double Fault Tolerance

The first is that computers now have to operate in harsher environments. Earlier, computers operated in clean computer rooms, with stable climate and clean air. Now the computers have moved out to industrial environments, with temperatures over a wide range, dust, humidity and unstable power supply. All these factors alone could make a computer fail. Second, the users have changed. Earlier, computer operators were trained personnel. Now, with an increasing number of users, the typical user knows less about proper operation of the system.

The consequence is that computers have to be able to tolerate more. Third, the service costs increases relative to hardware costs. Earlier the average machine was a very expensive, big monster. At that time, it was common with one or several dedicated operators to keep the system up and running. Today, a computer is cheap, and the user has the job of being the "operator". The user can not afford frequent calls for field service. The fourth and last trend is larger systems. As systems become larger, there are more components that can fail. This means, to keep the reliability at an acceptable level, designs have to tolerate faults resulting from component failures.

Adder is absolutely essential block in any digital architecture. Among different types of adders ripple carry adder (RCA) is most popular in different type of computing machines because it is simple in structure and easy to implement. It has also high throughput for bit level pipelining. We have taken a 4-bit RCA to make itself reconfigurable. We will also discuss how the same approach can be applied to any system, which can be divided into some identical modules, to make the system fault tolerant. The fault tolerant designs are also cascadable to increase the number of input bits Making a system module wise self-reconfigurable is more cost effective and hardware efficient rather than trying to make the whole system fault tolerant at a time.

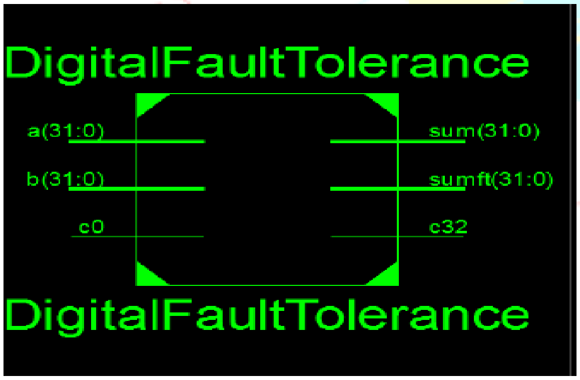


Fig. 2 Digital Fault Tolerance

The above Fig 2 shows the digital fault tolerance with four faults. We will also elaborate how the approach to design a self reconfigurable 4-bit RCA can be applied to design any fault tolerant module. Carry-look ahead adder is the fastest adder among the all adders. It generates carry bit in advance before it generates the sum, which reduce computation time to obtain final results. Here the Carry look ahead adder is implemented using the New Fault Tolerant gate (NFT) gate and Double Feynman gate.

Fault tolerant carry look ahead adder is used to add the partial product of the blocks. I assumed that all possible cell inputs must be applied to each cell in order to test it completely and that a fault in a cell may affect the cell outputs in any arbitrary manner. These are known as general fault assumptions.

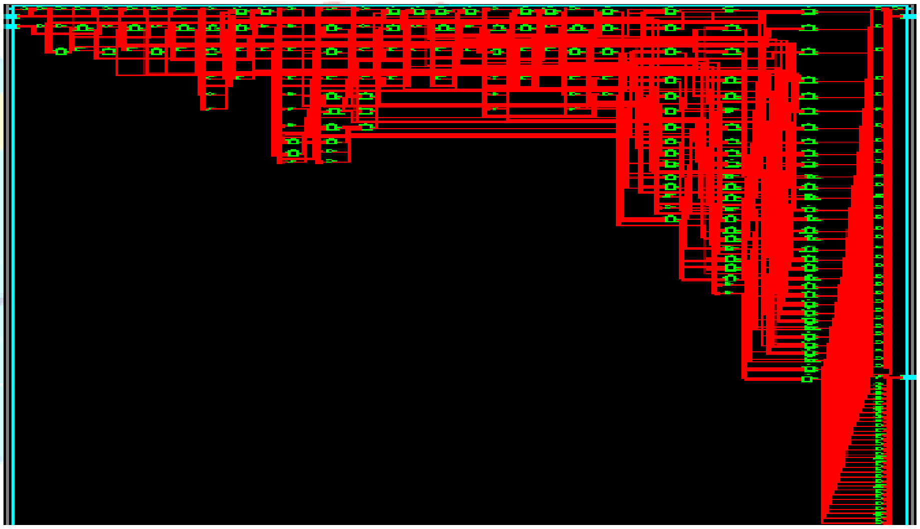


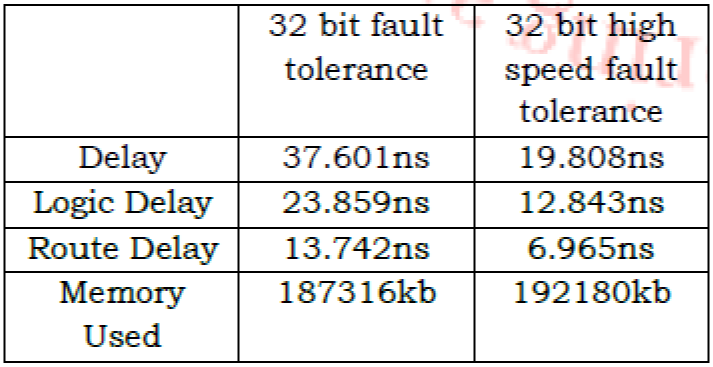
Fig. 3 Technology Schematic Digital Fault Tolerance

The above fig 3 shows the Technology Schematic Digital Fault Tolerance. In simultaneous addition is performed in parallel generating provisional carry and sum outputs without waiting for the original carry to be arrived and true sums and carries are selected by multilevel two-input MUX on arrival of the original carry. It mainly consists of seven conditional cells (CC) and several two input MUXes. The CC block is a combinational unit which pre-computes the carry and sum outputs assuming previous level carry value is equal to 0 and 1 both.

Fig. 4 Output Waveform

Efforts to attain that can tolerate faults have made use of static and dynamic redundancy approaches similar to those used for hardware faults. One such approach, uses static redundancy in the form of independently that perform the same functions and their outputs are voted at special checkpoints.

Here, of course the data being voted may not be exactly the same and a criterion must be used to identify and reject faulty versions and to determine a consistent value that all good versions can use. An alternative dynamic approach is based on the concept of recovery blocks. Programs are partitioned into blocks and acceptance tests are executed after each block. If an acceptance test fails, a redundant code block is executed. The design will be having two inputs A & B and a control line ctrl which will controls mode of operation i.e. when ctrl is at logic 0, the circuit will acts as half adder and when ctrl is at logic 1, the circuit will acts as half subtraction.



	32 bit fault tolerance	32 bit high speed fault tolerance
Delay	37.601ns	19.808ns
Logic Delay	23.859ns	12.843ns
Route Delay	13.742ns	6.965ns
Memory Used	187316kb	192180kb

Table 1 Comparison of 32 bit fault tolerance and 32 bit high speed fault tolerance

The above table shows the Comparison of 32 bit fault tolerance and 32 bit high speed fault tolerance. The high speed fault tolerance gives less delay that leads very high speed.

III. CONCLUSION

A system must be fault tolerant to decrease the failure rate and increase the reliability of it and if the system is having less delay it will execute quickly. Multiple faults can affect a system simultaneously and there is a trade-off between area overhead and number of faults tolerated. This paper presents high speed fault tolerant architecture design for digital applications. The fault containment and parallel processing capabilities of computers network are being exploited to provide a high performance, high availability network capable of tolerating a broad scope of hardware, software, and operating system faults.

REFERENCES

[1] Atin Mukherjee & Anindya Sundar Dhar "Double-Fault Tolerant Architecture Design for Digital Adder" Proceeding of the 2014 IEEE Students' Technology Symposium

[2] Y Sato, S. Kajihara, Y. Miura, T. Yoneda, S. Ohtake, I Inoue, H.Fujiwara, "circuit failure prediction mechanism (DART) for high field reliability," IEEE 8th International Conference on ASICON '09, pp.581-584, Oct. 2009.

[3] B. A. Prasad, and F. G. Gray, "Multiple fault detection in arrays of combinational cells," IEEE Transactions on Computers, vol. 100, no. 8, pp. 794-802, 1975.

[4] Francisco J. O. Dias, "Truth-table verification of an iterative logic array." IEEE Transactions on Computers, vol. 100, no. 6, pp. 605-613, 1976.

[5] S.-K. Lu, J.-C. Wang and C.-W. Wu, "C-testable design techniques for iterative logic arrays," IEEE Transaction on VLSI Systems, vol. 3, no. 1, pp. 146–152, Mar. 1995,.

[6] W.-T. Cheng and J. H. Patel, "A minimum test set for multiple fault detection in ripple carry adders," IEEE Transaction on Computers, vol. 36, no.7, pp. 891-895, July 1987,.

[7] A. Mukherjee; A. S. Dhar, "Design of a Self-Reconfigurable Adder for Fault-Tolerant VLSI Architecture," 2012 International Symposium on Electronic System Design (ISED), pp. 92-96, Dec. 2012,.

[8] J. Sklansky, "Conditional-sum addition logic," IRE Transactions on Electronic Computers, vol. 2, pp. 226-231, 1960.

[9] A. Mukherjee, A. S. Dhar, "Design of a Fault-Tolerant Conditional Sum Adder", H. Rahaman et al. (Eds.):

VDAT 2012, Springer LNCS 7373, pp. 217–222, 2012.

[10] Vengadapathiraj.M, Rajendhiran.V, Gururaj.M, Vinoth Kannan.A and Mohamed Nizar.S,"Design and FPGA Implementation of High Speed 128x 128 bits Vedic Multiplier Using Carry Look-Ahead Adder", international journal of advanced research in electronics and communication engineering ,Volume 4, Issue 2, february 2015.