Concurrent Error Detecting and Correcting Carry Select Adders

Greeshma Saju
Dept. of Electronics and Communication
College of Engineering
Trivandrum, India
greeshmasaju93@gmail.com

David Solomon George
Dept. of Electronics and Communication
College of Engineering
Trivandrum, India
david@cet.ac.in

***Abstract*—Highly reliable, low power and low area overhead functional units are of high importance in critical digital systems such as those required in defence applications, medical supervisory equipment and other safety and security systems. The occurrence of a fault in any part of such systems can impact the performance of the entire system. In this new era, where transistors employing current technologies are more vulnerable to both permanent and transient errors, design for concurrent error correction is required. For a 32 bit concurrent error detecting adder with easy testability feature, error correction circuit designs are proposed. The adder is testable under single stuck-at fault model with only 10 input patterns. The proposed adder is based on a 32 bit multi-block carry select adder.There are many techniques to accomplish fault tolerance of adders at the circuit level. In this project, hardware redundancy and time redundancy are taken into account for error correction. Power, area utilization and delay of adders are taken into account to compare the performance.**

keywords: Multi-Block Carry Select Adder, Fault Tolerant Carry Select Adder, Hardware Redundancy, Time Redundancy.

I. INTRODUCTION

Advanced microelectronic technologies have made current digital systems to become more vulnerable to errors. Compact circuit design on a chip is advantageous in terms of noise but it creates various problems in terms of reliability. Among the arithmetic circuits, adder circuit act as the basic building block for each design. For critical systems such as embedded systems, concurrent detection of errors is important. And hence fault tolerance of adders are crucial for the correct operation of most systems[14].

The main aim of employing fault tolerance techniques is to build reliable systems that will minimize the probability of system failures with minimum cost overhead. This means that the circuit must have the ability to detect as many faults as possible while its design requires a high speed, minimum area and reduced power overhead[5,6]. The adder to be proposed is based on a 32-bit multi-block carry select adder. The multi-block carry select adder calculates the sum result by selecting a result from two candidates according to the actual value of the carry input. One is by assuming 0 as the carry input and the other is by assuming 1 as the carry input[15].

This paper is organized as follows: Section I gives an introduction to this area. Section II deals with the design of concurrent error detectable adder with easy testability followed by section III which discusses the fault correction techniques. Section IV comes up with the proposed error-correcting adder designs followed by section V showing the obtained implementation results. Section VI concludes this paper.

II. CONCURRENT ERROR DETECTABLE ADDER WITH EASY TESTABILITY

For critical circuits, the first fault must be noticed before a second fault occurs. If a second fault has occurred before the first fault is found, then the concurrent error detectability can be lost. Therefore, it is important for a concurrent error detectable adder that it also has easy testability so that a fault in it can be found easily. Any erroneous output of the adder can be detected by comparing the parity of its sum result with the predicted parity of sum output and comparing its duplicated carry outputs. The adder is also testable with only ten patterns under a single stuck-at fault model[1,13]. Section A shows error detection based on parity prediction followed by Section B, which discusses easy testability of the adder. Fig 1 shows the design of block k of concurrent error detectable adder with easy testability.

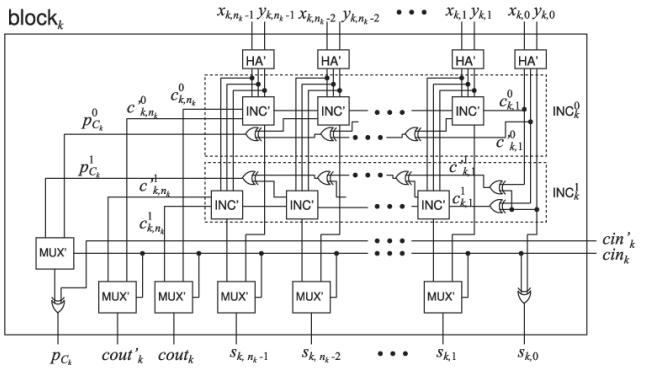

Fig. 1. Design of block k of concurrent error detectable adder with easy testability

A. Error Detection based on Parity Prediction

The process of detecting error can be performed either off-line when the system is inactive, or online when the

system is operational. The latter mode is referred to as concurrent error detection. Continuous and concurrent error detection is required for transient fault detection. The Off-line checks usually include testing of the system by supplying specific input test patterns to the components and analysing the outputs. These types of checks have a high coverage but are expensive in terms of time and resources, and so they are rarely implemented as diagnostic checks. The operational period for high availability requirement systems, such as Web and database servers, is almost 100% of the time. For ensuring the dependability of such systems concurrent error detection becomes very important.

The parity prediction scheme has the advantage that it requires the minimum hardware overhead. The parity predicting logic will predict the correct parity for the output sum of the adder using the parities of inputs X, Y and cin as in equation 1. The predicted parity of sum could then be compared against the generated parity (which is the parity of generated adder sum output) for checking the correctness of the adder outputs [1,4,7]. Fig 2 shows a datapath circuit of the proposed adder using parity-based error detection.

$$Predicted\ parity\ of\ sum, Ps = Px \oplus Py \oplus Pcin \quad (1)$$

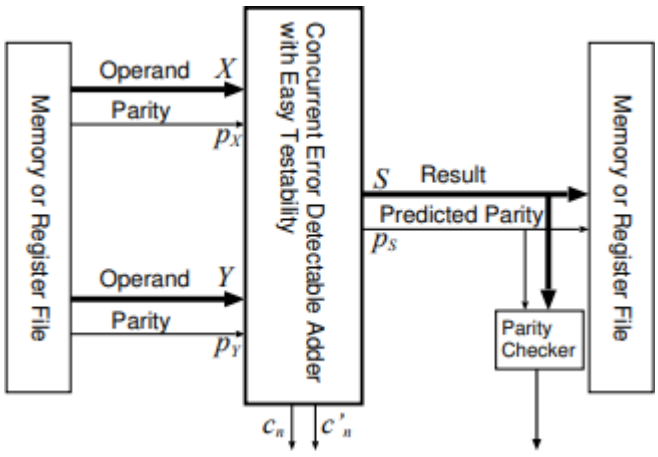

Fig. 2. Example of a datapath circuit of the proposed adder using parity-based error detection.

In the adder, each adder block has two carry inputs cink and cin'k. sk,0 will be erroneous if there is an error on cink. Furthermore, the error on cink can also induce errors on several other sum outputs. The output of an XOR or a MUX' affects only one sum or carry bit or the predicted parity. Therefore, the effect of a fault in them is detected by comparing the predicted parity of sum Ps with the generated parity of the the sum output and comparing two carry outputs cn and c'n of the adder.

B. Easy Testability

C-testability can be defined as a property of a circuit to be testable with a constant size test set regardless of its bit-width. The adder described here is C testable, the property which is also useful for testing during operation[3]. The adder is testable with 10 input patterns under the single stuck-at fault model, through designing test patterns. Each test pattern is obtained by packing input patterns for testing the addition blocks. A test set for a 32-bit adder consisting of four 8-bit blocks is discussed in [1].

III. Fault Correction Techniques

Designing fault-tolerant adders are of great importance in the process of reliable system design. At the circuit level, there are many techniques to accomplish fault tolerance of adders[2,8,9,12]. And some type of redundancies are always associated with it: (a) hardware redundancy, (b) time redundancy, (c) information redundancy, and (d) software redundancy.

Hardware redundancy can be achieved by adding either multiple complete copies or even partial parts of hardware components. This extra redundancy can be done at the transistor level, gate level, functional unit level, architecture level, or even at the system level. Time redundancy relies on executing the same operation multiple times using the same hardware and comparing the obtained results to discover the presence of any faults[10,11]. In information redundancy, extra redundancy is included or appended into the original data to detect or correct faults. There are several schemes for information redundancy such as parity bit addition, coding techniques and maintaining reasonable Hamming distance. Software redundancy achieves fault tolerance at the application level where fault tolerance techniques can appear as multiple extra code lines, based on some sort of replication over a single or multiple computers.

IV. Concurrent Error Correcting Carry Select Adders

Error-correcting carry select adder designs are proposed for a concurrent error detecting adder with easy testability. The proposed adder is based on a 32 bit multi-block carry select adder. It receives parities of two operands in addition to the operands and produces predicted parity of the sum result and two carry outputs in addition to the sum result. Any erroneous output of the adder by a fault modelled as a single stuck-at fault is detected by parity checking and comparison of the two carry outputs. Error correction is carried out based on hardware redundancy and time redundancy schemes.

A. Hardware Redundancy

Hardware redundancy is provided by adding extra hardware into the design to either detect or cover the effects of a failed component. It can be used to overcome both permanent and transient faults and has the lowest effect on the performance of the system. However, it has significant penalties in terms of area and power consumption.

Here dual redundancy based on Hardware redundancy is considered for error correction as shown in fig 3. The select lines for sum MUX and cout MUX are the AND gates output from the two redundant concurrent error detecting adder

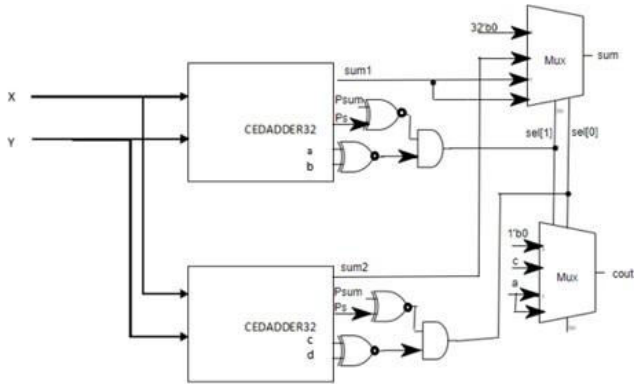

Fig. 3. Error correction design based on Hardware Redundancy

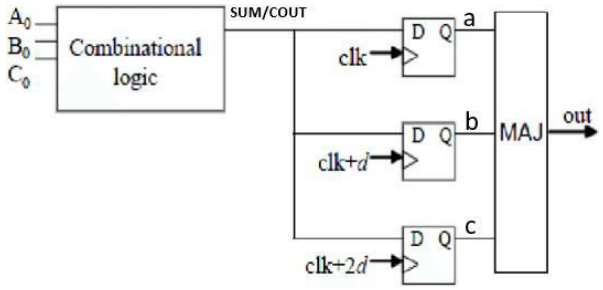

Fig. 4. Error correction design based on Time Redundancy

circuits. If the first adder is found erroneous the second adder output is selected from the MUXs and if the second one is found to be wrong then the first output is selected. And if both the adder outputs are erroneous then the result is pulled down to zero.

The 32 bit concurrent Error-Correcting adder circuit consists of 8 blocks each being a 4 bit adder. In the adder circuit, manipulation has been induced using the manipulation bits as mba, mbb, mbpx1, mbpx2, mbpy1 and mbpy2. mba - manipulation induced for the 8 blocks in first cedadder, mbb - manipulation induced for the 8 blocks in second cedadder, mbpx1 - manipulation induced for the parity of input value X in first cedadder, mbpx2 - manipulation induced for the parity of input value X in second cedadder, mbpy1 - manipulation induced for the parity of input value Y in first cedadder, mbpy2 - manipulation induced for the parity of input value Y in second cedadder

A 32-bit concurrent error-correcting CSeA is designed based on dual redundancy, which is a hardware redundancy method. Even though manipulation bits are set to induce error into the circuit, the sum and cout outputs are corrected ones. This can be achieved by dual redundancy. If first adder is having an error, then the second adder outputs are taken. Similarly if second adder is having error, then the first adder output is taken as the corrected one.

B. Time Redundancy

Time redundancy based fault-tolerance is achieved without introducing too much hardware overhead and can be used in applications where time is not critical. Time redundancy relies on executing the same operation multiple times using the same hardware and comparing the obtained results to discover the presence of any faults. For fault detection, two-time redundancy is adequate, while in fault correction it requires at least three-time redundancy. Clearly, this method almost has no area overhead, but it introduces a large delay and can only overcome temporary or transient faults. Time redundancy is useful in applications where time is of less importance than hardware complexity.

$$\textit{Majority voteroutput, out} = a \cdot b + b \cdot c + c \cdot a \quad (2)$$

Time redundancy techniques rely on the advantage of the transient fault by comparing the output signals at different moments. As shown in fig 4, sum output from the concurrent error detecting adder is given to three D flip flops at three different periods each with d time gap. The outputs from these will be given to a majority voter circuit to find the correct output. Correct sum output and cout output are found using this time redundant circuit. Majority voter output equation is given in equation 2. Here the manipulation bit mba is set to induce error. As three D flipflops are clocked at three different periods, the error induced is not propagated. And hence majority voter circuit output will be having the correct value.

The area overhead in time redundancy comes from the added sample flipflops. The basic concept of time redundancy is that the same hardware will perform a single operation at different intervals of time, and error can be detected by comparing the outputs obtained at different time instances.

The performance penalty is given by clk+(N-1)d+tp, where d denotes the expected duration of the transient pulse, N is the number of latched times and tp is the majority voter delay. Time redundancy is a desirable approach to fault-tolerance in applications where speed is not critical. And as the technology for high-speed circuits improved, time redundancy has become practical for many systems where it was not possible before.

V. IMPLEMENTATION RESULTS

Concurrent error correcting 32 bit adders are designed based on hardware redundancy and time redundancy. The adder considered is a multi-block carry select adder. The designing of circuit is done using verilog language in ModelSim and synthesized using Xilinx ISE. Comparison is achieved in terms of power,area and delay. Power is almost same for both adders. Fig 5 represents area utilization of hardware redundancy and time redundancy based concurrent error correcting circuits. And fig 6 represents time delay of hardware redundancy and time redundancy based concurrent error correcting circuits.
Results obtained are tabulated as below:

Parameter	Value
Number of 4 input LUTs	407 out of 9312
Number of IOs	119 out of 232
Number of occupied slices	243 out of 4656
Delay	15.519ns

TABLE I
VARIANTS OF HARDWARE REDUNDANCY BASED ERROR CORRECTING CIRCUIT

Parameter	Value
Number of 4 input LUTs	204 out of 9312
Number of IOs	112 out of 232
Number of occupied slices	151 out of 4656
Delay	24.283ns

TABLE II
VARIANTS OF TIME REDUNDANCY BASED ERROR CORRECTING CIRCUIT

VI. CONCLUSION

Error correcting designs for a 32 bit concurrent error detecting carry select adder with easy testability is proposed. Any erroneous output of the adder by a fault modeled as a single stuck-at fault is detected by parity checking and comparison of the two carry outputs. Error correction will be carried out based on hardware redundancy and time redundancy schemes. Hardware redundancy has the lowest effect on the performance of the system and can be used to overcome both permanent and transient faults. However, it has significant penalties in terms of area and power consumption. Time redundancy techniques rely on the advantage of the transient fault by comparing the output signals at different moments. It is a desirable approach to fault-tolerance in applications where speed is not critical.

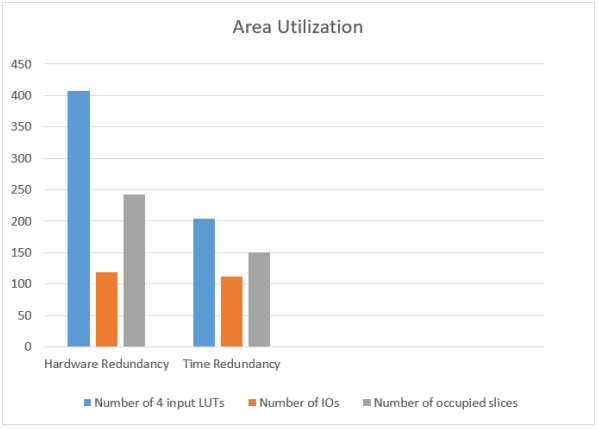

Fig. 5. Area utilization of hardware redundancy and time redundancy based concurrent error correcting circuits

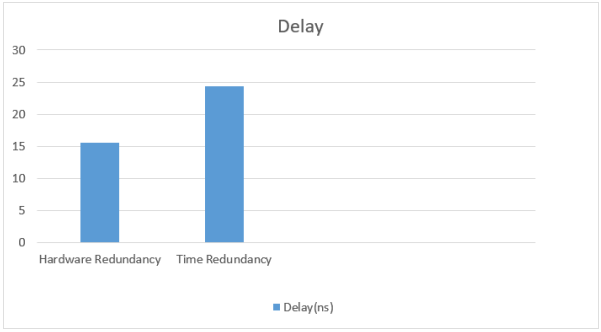

Fig. 6. Time delay of hardware redundancy and time redundancy based concurrent error correcting circuits

REFERENCES

[1] Kito, Nobutaka, and Naofumi Takagi. "Concurrent Error Detectable Carry Select Adder with Easy Testability." IEEE Transactions on Computers 68.7 (2019): 1105-1110.
[2] Talib, Ghashmi H. Bin, Aiman H. El-Maleh, and Sadiq M. Sait. "Design of fault tolerant adders: a review." Arabian Journal for Science and Engineering 43.12 (2018): 6667-6692.
[3] Kito, Nobutaka, Shinichi Fujii, and Naofumi Takagi. "A C-testable multiple-block carry select adder." IEICE transactions on information and systems 95.4 (2012): 1084-1092.
[4] Kumar, B. Kiran, and Parag K. Lala. "On-line detection of faults in carry-select adders." International Test Conference, 2003. Proceedings. ITC 2003.. IEEE Computer Society, 2003.
[5] Xing, Shanzhen, and William WH Yu. "FPGA adders: Performance evaluation and optimal design." IEEE Design & Test of Computers 15.1 (1998): 24-29.
[6] Harish, Basavoju, K. Sivani, and M. S. S. Rukmini. "Design and Performance Comparison among Various types of Adder Topologies." 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2019.
[7] Nicolaidis, Michael. "Carry checking/parity prediction adders and ALUs." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 11.1 (2003): 121-128.
[8] Akbar, Muhammad Ali, and Jeong-A. Lee. "Self-repairing adder using fault localization." Microelectronics Reliability 54.6-7 (2014): 1443-1451.
[9] I. M. Hafez Radwa M. Tawfeek, Mohamed G Egila, Yousra Alkabani."A survey on faults and mitigation techniques in FPGAs." IEEE 4th International Conference on New Paradigms In Electronics & Information Technology(PEIT'017). 2017/11
[10] Hsu, Yuang-Ming, Vincenzo Piuri, and E. E. Swartzlander. "Efficient time redundancy for error correcting inner-product units and convolvers." Proceedings of International Workshop on Defect and Fault Tolerance in VLSI. IEEE, 1995.
[11] Khedhiri, Chiraz, et al. "Concurrent error detection adder based on two paths output computation." 2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops. IEEE, 2011.
[12] Forsati, Rana, et al. "A fault tolerant method for residue arithmetic circuits." 2009 International Conference on Information Management and Engineering. IEEE, 2009.
[13] Vasudevan, Dilip P., and Parag K. Lala. "A technique for modular design of self-checking carry-select adder." 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05). IEEE, 2005.
[14] Rivers, Jude A., et al. "Error tolerance in server class processors." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 30.7 (2011): 945-959.
[15] Bedrij, Orest J. "Carry-select adder." IRE Transactions on Electronic Computers 3 (1962): 340-346.