ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2015.0019

HIGH SPEED ONLINE FAULT DETECTION OF 64-BIT RIPPLE CARRY ADDER USING MODIFIED MODULAR REDUNDANCY

Jisha M. Nair[1] and C. Pradeep[2]
Department of Electronics and Communication Engineering, Saintgits College of Engineering, India
E-mail: [1]jishamnr92@gmail.com, [2]pradeepcee@gmail.com

Abstract

Developments in VLSI technology has increased density of chips so that, processing elements are capable of doing complex computations. The increase in complexity and density of the VLSI chip has made electronic systems more susceptible to defects. Therefore, testing and fault tolerance techniques are required to guarantee reliable operations of systems. Online fault detection techniques are active during normal operations. This paper proposes a novel online fault detection technique for 64-bit Ripple Carry Adder based on type of input data. A technique is developed to detect the single stuck-at faults that occur in the Ripple Carry Adder by using two rail checker. The design was modeled using Verilog HDL and simulated and synthesized in Xilinx ISE 14.5. A comparison is made by implementing the design in different FPGA devices. The results show that the proposed design has better device utilization and less delay in Virtex 5 FPGA. The proposed technique has only 16.2% delay overhead compared to simple Ripple Carry Adder without fault detection.

Keywords:

Defect, Fault, FPGA, Online Fault Detection, Two Rail Checker

1. INTRODUCTION

The advent in VLSI technology has increased the complexity and density of VLSI chips, hence high speed digital systems are more prone to defects. Defects are the unintended change in design from original design [1]. A fault is a representation of a defect reflecting a physical condition that causes a circuit to fail to perform in a required manner. The probability of faults occurring in the VLSI systems has increased as the dimension of transistor is increased [2]. Thus, occurrence of faults has become common in electronic systems and therefore it is desirable to incorporate fault tolerant techniques into designs to ensure reliability.

Since adders are the vital processing unit of a processor, it is essential to guarantee fault free operation of these units. It is necessary to use efficient online fault detection methods with less area overhead and performance penalties. Online fault detection techniques detect the fault in normal operation of system without shutting down system as in offline testing. Hence, it is gaining importance as both permanent and transient fault are detected at normal operation itself. Moreover, online fault detection is the crucial step for designing a self-repairable system. The online fault detection using hardware redundancy technique detects faults immediately when they occur. In addition, these techniques have less impact on timing performance [3].

In this paper, an online fault detection method for 64-bit ripple carry adder for adder units is proposed. The proposed technique uses the concept of conventional Dual Modular Redundancy (DMR) in efficient manner to ensure high speed and less area overhead fault detection. The proposed technique named Modified Modular Redundancy (MMR) detects the single stuck-at faults in adder when same inputs are given to the adder units. The design is modeled in Verilog HDL, simulated and synthesized in Xilinx ISE 14.5. The synthesis result is used to study the performance of proposed method in terms of area and delay. The single stuck-at faults are injected into the design by editing the Xilinx Design Language (xdl) file and faults are detected at post route simulation. The proposed technique can be enhanced to n-bit adders.

The rest of the paper is organized as follows. Section 2 gives a brief description of related works on fault detection. Section 3 presents the design of the proposed system. The discussion of results is provided in section 4 and concluding remark and future work is presented in section 5.

2. PREVIOUS WORKS

Fault detections approaches are the first step for design of fault tolerant systems. The modular redundancy is widely used methods for online fault detection in Field Programmable Gate Array (FPGA). It is used in form DMR and Triple Modular Redundancy (TMR) to detect and correct the errors [4]. The instant fault coverage comes with the expense of area overhead.

In the recent years, various self-checking circuits using different coding schemes such as parity prediction, arithmetic codes, two rails coding etc. are designed to check the functionality of the circuits online. In [5] self- checking adder circuits using arithmetic codes was proposed. Even though the arithmetic codes are efficient for checking arithmetic circuits as they preserve arithmetic operations, checkers suffers from high complexity and area overhead. In the self-checking arithmetic circuit using parity prediction scheme [6], a technique was proposed which is capable of detecting fault at the output of adder. However, any fault present at the carry out is propagated to other outputs and remain undetected. An online fault detection technique for carry select adder using time redundancy concept was proposed [7].

The drawback of this technique is that the computation time is more than double the execution time, thus making the addition operation extremely slow. In [8] an efficient implementation of self-checking adder in which combined double-rail and parity codes are used to implementation all the known ALU and adder schemes. The limitation is that it is not capable of detecting singe stuck-at fault.

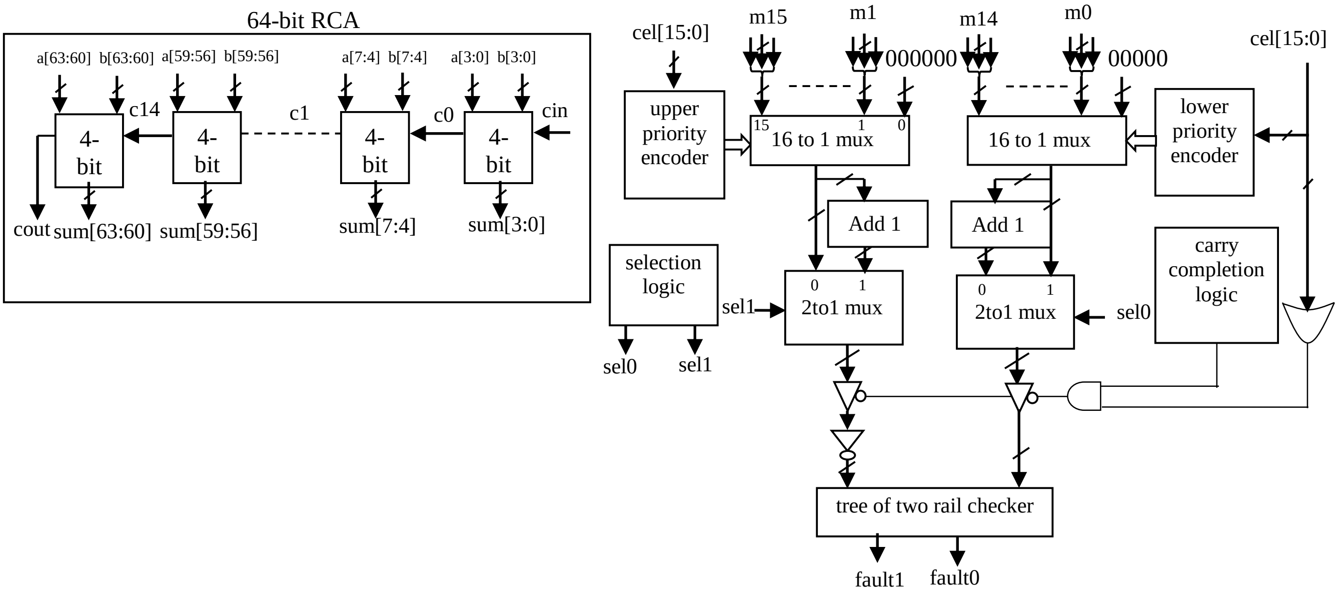

Fig.1. Architecture of 64-bit Ripple Carry Adder with proposed online fault detection technique

In [9] presented the requirement of online testing and described different online testing techniques that could provide adequate solution. The design of self-checking circuits and checker for different data paths and error detection codes are detailed in this literature.

In [10] a technique for online fault detection in carry select adder was proposed using two rail checker. The design suffered from limitation of high area overhead due to complex encoding logic. A literature [2] presented a self-checking carry select adder with reduced area over ahead. The problem with this approach is that the design provides wrong indication of fault even if the fault is absent when carry-in to system is equal. In [11] presented a corrected design of [1]. The drawback of this approach is that area overhead almost doubles in comparison to [1]. In [12], an online fault detection of Wallace tree multiplier based on data dependant mechanism is proposed.

3. PROPOSED FAULT DETECTION METHOD

The proposed online fault detection method uses the technique of hardware redundancy. An N-bit adder implemented using 4-bit adders as basic functional unit (cell) has in-built modular redundancy. This modular redundancy is used to ensure online fault detection. The detection of faults takes place according to the type of data input to these adder cells. When the inputs to two cells are equal their outputs should also be equal, any mismatch in the outputs of these adder cell indicate the presence of fault in the system.

N-bit adder architecture has N/4 adder cells and each cell is a 4-bit adder. The number of combinations where the inputs to any two adder cells are equal can be obtained by using theory of probability and is given by ${}_nC_r$.

$$ {}_nC_r = \frac{n!}{r!(n-r)!} \qquad (1) $$

The number of combinations by which two cells (r = 2) is selected from 'n' number of cells is given by,

$$ {}_nC_2 = \frac{n!}{2!(n-2)!} \qquad (2) $$

Thus, the total number of checking conditions required for both the inputs of these cells is given by,

$$ T={}_nC_2+{}_nC_2 \qquad (3) $$

By using this equation, the total number of checking conditions for different input length adder is calculated and is given in Table.1.

Table.1. Total no. of combinations for different input length

Input Bit length	No. of cells	Total no. of checks
16-bit	4	12
32-bit	8	56
64-bit	16	240
128-bit	32	992

Thus, it is evident from the Table.1 that, the number of equality detector required for checking equality of input data increases as the input bit length increases. Therefore finding the equality of data inputs to adder cells can be assumed to be done by a software comparator whose design is out of area of research.

The Fig.1 shows the block diagram of 64-bit Ripple Carry Adder (RCA) with proposed online fault detection technique. The software comparator set the 16-bit register 'cel' to indicate which two adder cells have same inputs. The lower and upper priority encoder encoded these values to select the outputs of corresponding adder cells using two 16 to 1 multiplexers. The input of multiplexer contains carry-out, sum bits and carry-in bit of selected cell and is represented as m0, m1,..m15, where m0 = {c0, sum[3:0], cin},.. and m15 = {cout, sum[63:60], c14} . Even though the data inputs A and B to adder cells are equal their outputs differ according to the third input i.e. carry-in of each cell.

So whenever the carry input to a cell is logic ‘1’ then a logic ‘1’ is added to the output of other cell by using an add one circuit [13].

The carry-in to the selected cells is compared using an E-XOR gate and if they are equal the outputs of adder cell is passed to tri-state buffer through a 5-bit 2 to 1 multiplexer otherwise, the output of add one circuit is passed to tri-state buffer corresponding to carry-in using selection logic. The tri-state buffer is only enabled when the carry completion logic provide a logic ‘1’. The output of tri-state buffer is passed to a two-rail checker. For a two-rail checker the inputs to these checkers should be complementary [14]. The identical outputs at checker indicate the presence of fault. If none of the inputs to these adder cells are equal, then checker are inactive as the tri-state buffer get disabled.

The proposed technique can be scaled to n-bit adder for n>8. For n-bit adder there are n/4 adder cells then corresponding register ‘cel’, upper and lower priority encoder and multiplexers are to be designed.

4. RESULTS AND DISCUSSIONS

In order to find the validity of the proposed online fault detection, the design was modeled using Verilog HDL, simulated, and synthesized using Xilinx ISE 14.5. We implemented 64-bit RCA with proposed online fault detection, conventional DMR and without fault detection and the results were used to compare the performance proposed technique.

First, 64-bit RCA with and without fault detection mechanism and conventional DMR was modeled and the functional simulation was done by using ISim with exhaustive test bench. After the functional simulation, fault was injected into design by editing Xilinx Design Language (xdl) file and was post route simulated to confirm the online detection of fault. The design was done in Xilinx FPGA Virtex-5 xcvlx30t. These designs are also implemented in other FPGA devices to compare the design with various Xilinx FPGA’s to obtain device utilization for FPGA and the maximum combinational path delay.

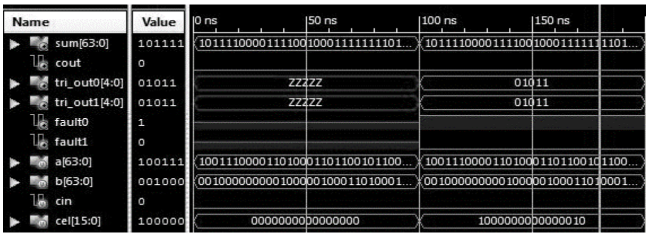

Fig.2. Simulation waveform for the proposed online fault detection before fault injection

The Fig.2 shows the functional simulations waveform for the 64-bit RCA with proposed online fault detection method. In the first case, none of the adder cell have same inputs, so, the input ‘cel’ = 0000000000000000. Therefore, the checker is inactive and the tri-state buffer produce high impedance ‘zzzzz’ at output so, invalid outputs comes at ‘fault0’ and ‘fault1’. The addition output is obtained at sum and cout lines. In the second case, the inputs to adder cell 1 and 15 are equal and corresponding bits of ‘cel’ is set i.e. ‘cel’ = 1000000000000010. Since the adder is fault free, ‘tri_out0’ and ‘tri_out1’ have same value, and two rail checker provide complementary value at output signals ‘fault0’ and ‘fault1’ to indicate fault free operation. The {fault1, fault0} = {0, 1} is obtained at output.

In Fig 3, the post route simulated waveform of adder after injection of fault is shown. After the injection of a stuck-at 0 fault in the sum[12] i.e in cell 3, so whenever the inputs to cell 3 and any other adder cells become equal the fault is detected and indicated by identical value at signal ‘fault0’ and ‘fault1’. In the second case, the cells 0 and 3 have equal inputs so ‘cel’ = 0000000000001001 and output signal ‘fault0’ and ‘fault1’ gives 00 indicating presence of fault.

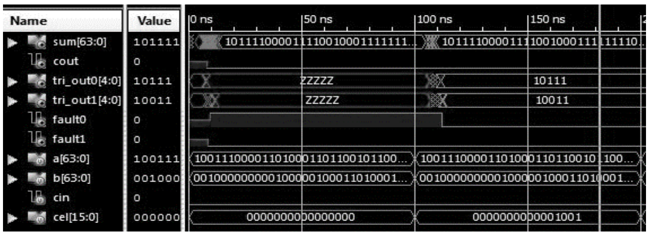

Fig.3. Simulation waveform for the proposed online fault detection after fault injection

Table.2. Device utilization of proposed method with other architectures

Device	Architectures	LUTs used	Slice occupied	No. of bonded IOBs
Spartan3 (xc3s400)	MMR	285	157	212
	DMR	461	256	196
	RCA	128	96	194
Virtex4 (xc4vfx12)	MMR	285	157	212
	DMR	461	256	196
	RCA	128	96	194
Virtex5 (xc5vlx110t)	MMR	195	92	212
	DMR	318	161	196
	RCA	96	79	194

Table.3. Timing details of proposed method with other architectures

Device	Architectures	Timing Delay (ns)
Spartan3 (xc3s400)	DMR	106.317
	MMR	103.744
	RCA	95.675
Virtex4 (xc4vfx12)	DMR	46.435
	MMR	46.127
	RCA	42.48
Virtex5 (xc5vlx110t)	DMR	28.832
	MMR	25.644
	RCA	22.073

The Table.2 shows the comparison of area of the 64-bit RCA with proposed method MMR with simple Ripple Carry Adder

(RCA) and DMR in terms of device utilization in different FPGA devices. The three architectures are compared in Spartan3, Virtex4 and Virtex5. Spartan3 is used for high volume applications and Virtex4 and Virtex5 provides the feature of partial reconfiguration. Our proposed online fault detection method utilizes less number of LUTs, slices, and IOBs when it is implemented on Virtex 5. It takes 195 LUTs, 92 slices and 212 IOBs whereas for Spartan3 and Virtex4 LUTs and slice occupied are 285 and 157 respectively.

The Table.3 shows the maximum combinational path delay of 64-bit RCA with proposed fault detection technique with other design in different FPGAs. The 64-bit RCA with proposed fault detection has least delay of 25.644 ns in Virtex5 comparing to 103.744ns and 46.127 for Spartan3 and Virtex4 respectively.

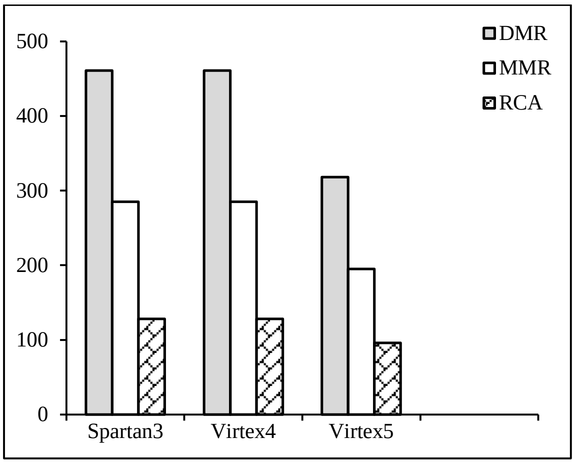

Fig.4. Comparison of area in terms of LUTs used with other design in different FPGA devices

The Fig.4 and Fig.5 shows the comparison of area and delay of 64-bit simple RCA with MMR and DMR. The chart proves that our proposed method has less area and delay compared to DMR and the design has less device utilization and delay in Virtex5. Thus implementing the 64-bit RCA with proposed fault detection in Virtex5 provide better performance of system and partial reconfiguration feature of Virtex5 enable the design of self-repairable systems. Our proposed technique has less delay overhead of 16.2% compared to 30.6 % of DMR.

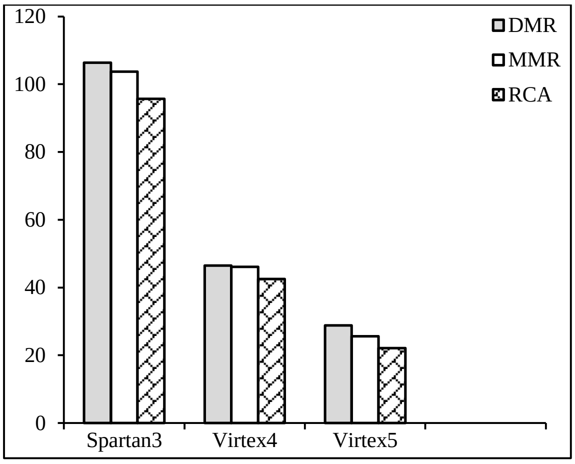

Fig.5. Comparison of delay in ns with other design in different FPGA devices

5. CONCLUSION

In this paper, we propose a Modified Modular Redundancy (MMR) technique for online fault detection in 64-bit ripple carry adder based on type of data inputs. Our proposed method has less delay overhead and smaller area overhead compared to conventional DMR. The proposed design has less device utilization and timing delay in Virtex5 FPGA device. Thus implementing the proposed technique in Virtex5 FPGA ensures better performance; moreover partial reconfiguration feature of Virtex5 permit the design of a self-repairable systems. Our future work is to decrease the area overhead of proposed method for n-bit adders where (n > 8) and to design a self- repairable arithmetic unit.

ACKNOWLEDGMENT

The authors thank the management of Saintgits College of Engineering for providing required facilities.

REFERENCES

[1] M. Bushnell and Vishwani Agrawal, “*Essential of Electronic Testing for Digital, Memory and Mixed Signal VLSI Circuits*”, Springer, pp. 60-80, 225,234, 2002.

[2] D.P. Vasudevan, P.K. Lala and J.P. Parkerson, “Self-Checking Carry-Select Adder Design based on Two-Rail Encoding”, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 54, No. 12, pp. 2696-2705, 2007.

[3] E. Stott, P. Sedcole and P.Y.K. Cheung, “Fault Tolerant Methods for Reliability in FPGAs”, *Proceedings of International Conference on Field Programmable Logic and Applications*, pp. 415-420, 2008.

[4] R.E. Lyons and W. Vanderkulk, “The Use of Triple-Modular Redundancy to Improve Computer Reliability”, *IBM Journal of Research and Development*, Vol. 6, No. 2, pp. 200-209, 1962.

[5] W.W. Peterson, “On Checking an Adder”, *IBM Journal of Research and Development*, Vol. 2, No. 2, pp. 166-168, 1958.

[6] E. Fujiwara and K. Haruta, “Fault-Tolerant Arithmetic Logic Unit using Parity-Based Codes”, *IEICE Transactions*, Vol. E64, No. 10, pp. 653-660, 1981.

[7] F.W. Shih, “High Performance Self-Checking Adder Having Small Circuit Area”, U.S. Patent 5018093 A, May 21, 1991.

[8] M. Nicolaidis, “Efficient Implementations of Self-Checking Adders and ALUs”, *Proceedings of 23rd International Symposium on Fault-Tolerant Computing*, pp. 586-595, 1993.

[9] M. Nicolaidis, “On-Line Testing for VLSI: State of The Art and Trends”, *Integration, the VLSI Journal*, Vol. 26, No. 1-2, pp. 197-209, 1998.

[10] B. Kiran Kumar and P.K. Lala, “On-Line Detection of Faults in Carry-Select Adders”, *Proceedings of IEEE International Test Conference*, pp. 912, 2003.

[11] M.A. Akbar and J.-A. Lee, “Comments on “Self-Checking Carry-Select Adder Design Based on Two-Rail Encoding”,

IEEE Transactions on Circuits and Systems-I: Regular Papers, Vol. 61, No. 7, pp. 2212-2214, 2014.

[12] C. Pradeep, R. Radhakrishnan, R. Saranya and S. Philip, “Synthesis of Data Path Architecture with Online Fault Detection Mechanism for Reconfigurable Systems”, *Australian Journal of Basic & Applied Sciences*, Vol. 8, No. 10, pp. 239-245, 2014.

[13] K. Yotmgjoon and L.-S. Kim, “A Low Power Carry Select Adder with Reduced Area”, *Proceedings of IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 218-221, 2001.

[14] D.P. Vasudevan, P.K. Lala, Jia Di and J.P. Parkerson, “Reversible-logic design with online testability”, *IEEE Transactions on Instrumentation and Measurement*, Vol. 55, No. 2, pp. 406-414, 2006.