International Journal of Modern Electronics and Communication Engineering (IJMECE) ISSN: 2321-2152
Volume No.-6, Issue No.-6, November, 2018

Implementation of Double Fault Tolerant Full Adder Using Fault Localization with Pipelining Mechanism

Akhila Gorla, PG Scholar
Department of E.C.E, Vignan's Institute of Management and Technology for Women, Hyderabad, India
E-mail: akhilagorla@gmail.com

Harikrishna Ponnam, Assistant Professor
Department of E.C.E, Vignan's Institute of Management and Technology for Women, Hyderabad, India
E-mail: ponnamhk@gmail.com

Abstract: In the era of advanced microelectronics, rate of chip failure is increased with increased in chip density. A system must be fault tolerant to decrease the failure rate. The presence of multiple faults can destroy the functionality of a full adder and there is a trade-off between number of fault tolerated and area overhead. This paper presents an area efficient fault tolerant full adder design that can repair single and double fault without interrupting the normal operation of a system. RTL synthesis has been done by using Xilinx 14.7 and simulation is done by using Xilinx Isim. In this work we used to detect the fault based on internal functionality by using the self checking full adder and pipeline concept. by the proposed work of fault tolerated we can get effective results in terms of Area, Delay, Power consumption aspects and number of fault tolerated when compared to the existing designs.

Keywords: Single fault, Double fault, Self checking adder, Self repairing, Fault tolerant adder.

I. INTRODUCTION

Fault tolerant designs are mainly used for variety of operations such defense system, satellite and safety measures etc. An error occurred in a system may cost several damages or affect human life also. However, some systems are reconfigurable, which repair it [1]. These system can't be dismantle and repair that error automatically. In the current scenario, VLSI circuits become more complex as the CMOS feature size is scaling in nanometer regime. This downscaling makes the circuit more compact and sensitive to the transient faults [2]. Transient fault occurred in the integrated circuit due to electromagnetic noises, cosmic rays, cross-talk and power supply noise. In addition to this, technology scaling further increases the chances of the presence of permanent fault also [3].The concept of self checking and fault tolerant is introduced to deal with the problem of fault. Self checking indicates the detection of faults and overlooks the overhead associated with fault recovery. Most of the self checking approaches re-execute the instruction for the fault recovery. But, this process decreases the performance of the system because all faulty

$$Sum = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + BC_{in} + C_{in}A$$

nodes are reelected. However, this process does not guarantee fault recovery if the fault is permanent.

Adder performs a variety of applications in digital system and has great importance DSP operations. There are two types of faults occur in full adder design i.e. single fault and double fault. Single fault makes only one output faulty at a time. However, double fault makes both the outputs faulty at a time. It is very difficult to detect and repair the double fault. This makes the design more complex and requires more area overheads. It makes the fault tolerant full adder design a matter of great importance [4]. In the past, many approaches are introduced to design the self checking full adder using hardware redundancy or time redundancy.

These approaches become successful in detecting the fault but fails in indicating the exact location of that fault. Hence, it makes the other module faulty due to the propagation of the carry. In this paper, we propose a new fault tolerant design which indicates the single and double fault with its and automatic fault repairing is also possible in this design. The rest of the paper is organized as follows. In Section 2, some existing approaches used for fault checking and repairing are discussed. In section 3, the proposed self checking full adder is described. Proposed self repairing full adder is described in section 4. The simulation results of the entire referred self checking; self repairing and proposed fault tolerant full adder are presented in section 5. The

proposed fault tolerant design is compared with the existing design in the section 6. Section 7 draws the conclusion.

II. PREVIOUS DESIGN APPROACHES

A. Time Redundancy

Redundancy is required in self checking system. Time redundancy is a self checking approach and is used to detect the transient fault. This approach used the duplicate hardware in addition to the original hardware to perform the same operation at different interval of time [6]. The delayed clock is provided to the duplicate hardware. 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. The main limitation of this approach is that it does subsequent computation to reduce the propagation delay before comparing the outputs. Hence, if the first computation result is faulty and it is used for other computations, also makes the subsequent modules faulty.

III. B. HARDWARE REDUNDANCY:

The commonly used hardware redundancy approaches are triple modular redundancy (TMR) and Double modular redundancy (DMR) [7]. Triple modular redundancy approach is used to detect the single fault. As indicates by its name, it requires three identical modules in parallel to detect the fault. A fault is detected if anyone output of the modules are different. But this approach is not capable in indicating the exact location of the fault. Therefore, this approach provides the faulty outputs if two single full adder cells become faulty at a time. This problem can be removed by increasing the hardware but the resulting design requires more than 500% hardware.

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. This approach requires 200 % hardware cost whereas the hardware requirement of TMR is 300%. Hence, this approach becomes successful to increases the reliability of the design at the minimum cost. But, fault correction is not possible in this approach because the voting circuit cannot detect the location of the faulty module. The major drawbacks of the hardware redundancy approach are that it requires more than 200% area overhead and can-not detect double fault at a time.

The second problem is that fault recovery is not possible because it is not able to detect the faulty module. In addition to this, stuck-at faults are not detectable in this approach and a problem is creates when both the modules experience the fault.

A. C. Self -checking CSA

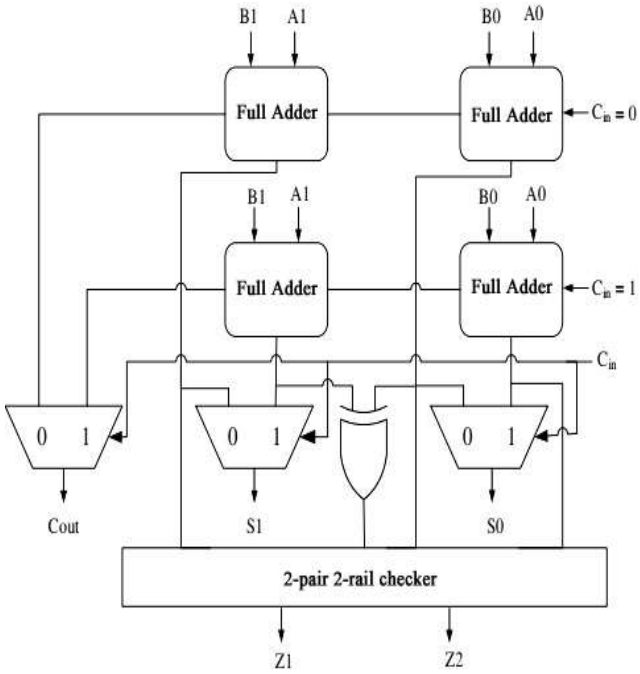

Figure.1. 2-bit self checking carry select adder

In self checking CSA, any single fault and single stuck-at fault can be detected successfully by 2 pair-2-rail checker during the online testing. Additionally, self-checking multiplexers and XOR gates are also used to detect these faults. This self checking adder is proposed by Vasudevan et al and shown in figure. 1. The full adder used in figure.1 is designed with 28 transistors. Faults are detected at the primary outputs by the self checking multiplexers. The checker has two outputs combinations 01 and 10 and indicated by Z1 and Z2. The combination of the two outputs indicates the presence of faults. 00 and 11 indicates the fault in the full adder cells. However, 01 and 10 indicates that the full adder cells are fault free the limitations of this design are that it can detect single net fault without indicating its exact location and also the problem of fault propagation through carry. Therefore, fault recovery is not possible in this approach.

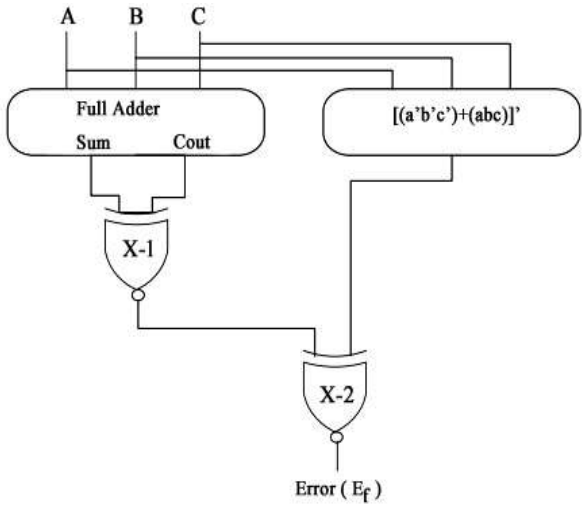

Figure 2. Self checking full adder.

D. Self repairing adder

Self repairing adder removes the problem of fault propagation through carry occurred in the previous self checking design by indicating the exact location of fault. This adder is proposed by Mohammad ali akbar et at and the design of self checking full adder is shown in figure. 2. The hardware requirements of self checking adder are full adder cell, equivalent tester and two XOR gates for self testing the fault.

$$Sum = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + C_{in}(A + B)$$
$$Eqt = NOT((A.B.C_{in}) + (ABC_{in}))$$

The XOR gate (X-1) is used for comparing the sum and carry outputs generated by the full adder cell.It works on the principle that sum and carry outputs will be equal when all the input applied are equal and the sum and carry outputs will be complement to each other when any of the three inputs applied is different from remaining inputs. The XOR gate (X-2) is used to compare the outputs of XOR gate (X-1) and functional unit [(A'B'C')+(ABC)]'.The fault is represented in the form of Ef. If Ef is 0, it shows that there is a fault and if Ef is 1 it shows the fault free condition. The main problem of this design is that it fails, if double fault occur in both sum and carry outputs at a time. In this case, Ef shows the fault free condition and faulty output propagates to the next unit through carry and makes them faulty.

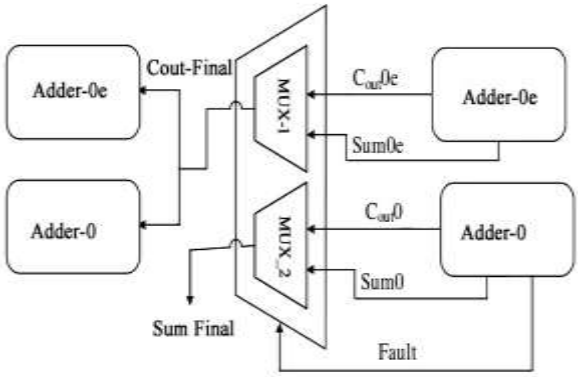

Figure.3. Self repairing adder.

The faults detected in self checking process are repaired by replacing the faulty full adder cell with another redundant full adder as shown in Fig.3. The working behavior is that one adder is work as a normal adder while the another adder is work as redundant adder. The hardware requirements of this design are two self checking full adders and two multiplexers. The limitation of this design is that it fails when double faults occur at a time. In this case, the fault indication output (Ef) of this design shows that there is no fault and self repairing design will not work in this in principle. Therefore, fault is not repaired by the self repairing adder and full adder shows the faulty output.

IV. PROPOSED SELF CHECKING ADDER

The output expressions for the sum and carry outputs of the full adder is shown in equations 1 and 2.

$$Sum = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + BC_{in} + C_{in}A$$

Table.1. Truth Table of Self Checking full adder Design

A	B	C	Sum	Carry	G 1	G 2	G 3	Eq t	F c	F s
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**
0	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**1**
0	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**1**
0	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**
1	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**
1	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**
1	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

The proposed self checking full adder is based on the principle given below.

1. The sum output is opposite to the carry output when vectors of inputs A, B and C are not equal i.e. (010,100). It shows that except (000) and (111) input combinations of A, B and C sum output is opposite to the carry outputs as shown in Table 1.
2. The sum output is equal to the carry output when input vectors A, B and C is equal i.e. (000, 111). It shows that for first (000) and last (111) input combinations of A, B and C, sum output is equal to the carry outputs as shown in Table 1.

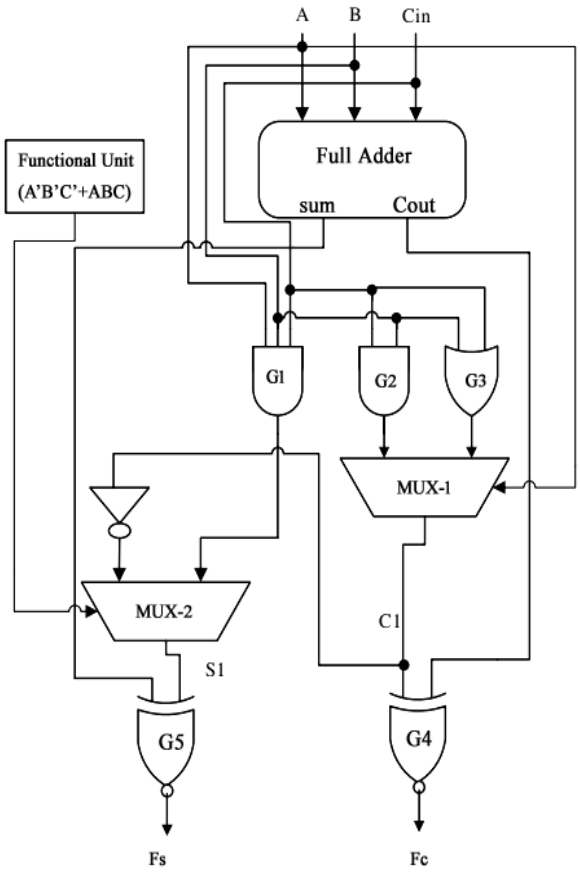

Figure.4. Proposed self checking full adder design.

The complete design of the self checking full adder is shown in figure. 4. The gates G1 and G2 are used to generate the AND and OR operation of the input bits B and C as given in 3 and 4. AND & OR gates are designed using CMOS logic. Then the outputs gates G1 and G2 are selected by the multiplexer under the control of input A to generate the output C1. The output C1 generates the vectors equivalent to the carry outputs. Then C1 and Carryout are compared using XNOR gate G4 as given in equation 5. XNOR gate is designed using DPL logic. If both are same, it shows the fault free condition and Fc will be 1. If both are different, it shows that carry output is faulty and Fc will be 0. Multiplexers are designed through transmission gates.

$$G1 = A + B$$
$$G2 = A.B$$
$$G4 = C1 \oplus C_{out}$$

To detect the fault in the sum output C1 is inverted using an inverter. This output along with the output (ABC) is fed to multiplexer under the control of (A'B'C' +ABC) to generate the S1. This output S1 generates the vectors equivalent to the sum outputs. Then S1 and Sum output are compared using XNOR gate G5 as given in equation 6. If both are same, it shows the fault free condition and Fs will be 1. If both are different, it shows that carry output is faulty and Fs will be 0.

V. PROPOSED SELF REPAIRING FULL ADDER DESIGN

The proposed self repairing full adder requires negligible area overhead than the existing designs. The operation of the proposed design is performed using multiplexers under the control of Fs and Fc. The output Fs and Fc are generated by the proposed self checking full adder. The proposed self repairing design does not need any stand by adder cell that are used to replace the faulty adder as used in the previous self repairing full adder [4]. In this approach, faults are repaired by using inverter in place of the standby full adder cell as shown in figure. 5.

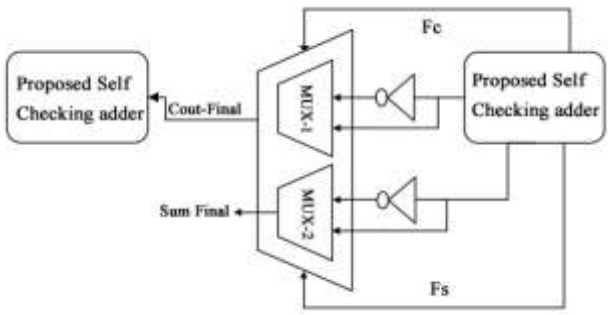

Figure.5. Proposed self repairing full adder design

The operation of the proposed design is based on the control signals (Fs and Fc) provided by the self checking full adder. If the control signal Fs is 0, it shows that there is no fault in the sum output and the sum output coming from the full adder cell will be selected by the multiplexer to generate the final sum. Multiplexers are designed using transmission gates. On the other hand, If the control signal is Fs 1, it shows that there is a fault in the sum output. The faulty sum output coming from the full adder cell is repaired by using the inverter. The inverted sum output is further

selected by the multiplexer to generate the final sum. Similarly, if the control signal Fc is 0, it shows that there is no fault in the carry output, and the carry output coming from the full adder cell will be selected by the multiplexer to generate the final carry.

On the other hand, if the control signal is Fc 1, it shows that carry output is faulty. The faulty carry output is repaired by using the inverter. The inverted carry output is further selected by the multiplexer to generate the final carry. In this way, the faulty adder cell is repaired and converted into the fault free adder. Therefore, this approach can repair single and double fault occurs at the sum and carry outputs at the cost of minimum hardware.

VI. RESULTS AND COMPARISION

The proposed fault tolerant full adder and some popularly known self check and self repairing full adder architectures have been implemented using UMC 55-nm standard cell library in cadence virtuoso tool. Specter simulator is used to perform the simulation. The hardware overhead and fault detection capability of the proposed design is found better in The hardware overhead is computed on the basis of the transistor count. The proposed fault tolerant full adder requires only one full adder cell with an inverter and no redundant full adder is required which are used in the previous designs. The hardware cost of the proposed and self repairing adder can be calculated and compared using the equation 7. 100% *Area.*

A. Fault Coverage and Repairing:

In the proposed fault tolerant design, faults in sum and carry outputs are indicated in the form of Fs and Fc respectively. The logic high of Fs will indicate a fault in the sum output while logic high of Fc will indicate a fault in the carry output as shown in the output waveform of Figure.6. and Figure.7. The proposed design has guaranteed to detect and repair the faults (transient and permanent) online even if the double fault occurs at the same time. The fault free output of the proposed fault tolerant full adder is shown in figure. 6. In this output Fc and Fs is at logic 1. The faulty output waveform of the proposed fault tolerant full adder is shown in figure. 7. In this output Fc and Fs is at logic 0 at particular instance, it shows that the output is faulty. This faulty output is repaired by the proposed design and final sum and carry outputs are shown in the form of Sum F and Carry F

VII. COMPARISION WITH EXISTING DESIGNS

1. The proposed fault tolerant full adder can detect single and double fault at a time. Hence, the proposed design is free from the problem of fault propagation through carry. However the TMR, DMR, self checking [10] and self repairing full adders have the problem of fault propagation through carry.
2. The proposed design is capable of detecting and repairing the transient and stuck-at faults occur in single and multinet. However, design proposed in, TMR and DMR approaches can't detect the stuck-at fault.
3. The proposed fault tolerant full adder does not require redundant full adder for repairing the faults. However, the self repairing design [4] requires the redundant adder for repairing the fault. Hence the area acquired by the proposed design is less.
4. The proposed design requires an inverter to repair the faults in place of the redundant adder used in the existing design. Therefore, the proposed design has minimum chances of common mode failure than the existing designs.

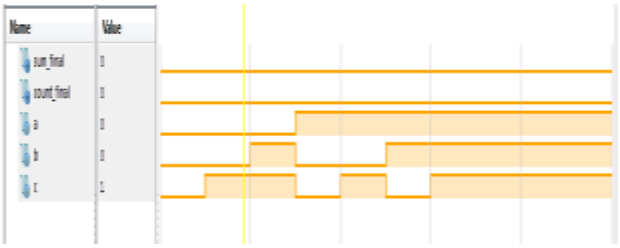

Figure .6. Simulation results of Proposed Self Repairing adder with Pipelining.

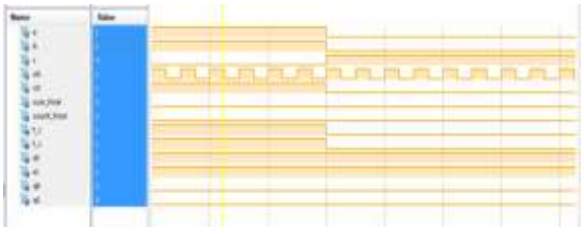

Figure .7. Simulation results of Proposed Self Repairing adder with Pipelining.

VII. CONCLUSION

In our design, A Carry select adder architecture is presented aiming the multiple fault detection and correction capability. The proposed design consume lesser area because instead of replacing the faulty adder with redundant

adder, faulty sum and carry output is inverted using an inverter and multiplexer. Hence, proposed fault tolerant design consumes 19.5 % lesser area than the self repairing full adder. This design is capable in tolerating the double fault in addition to the single fault whereas self repairing full adder can't detect the double fault. The comparison results of the proposed pipe line concept technique are found better that ensure their superior performance capability when compared to the existing designs. We obtained more effective results in proposed design of double fault tolerant full adder in terms of Area, Delay, and power and speed factors.

REFERENCES

[1] A. Mukherjee and A.S. Dhar, " Design of a Self-Reconfigurable Adder for Fault-Tolerant VLSI Architecture," IEEE International Symposium on Electronic System Design (ISED), 2012.

[2] Adve, T. Austin and V. Bertacco."CrashTest'ing SWAT: accurate, gatelevel evaluation of symptom-based resiliency solutions"' in Proceedings of the conference on design, automation and test in Europe. 2012.

[3] Kaveh, O. Kavehei, O, "A novel low-power full-adder cell with new technique in designing logical gates based on static CMOS inverter," Microelectronics Journal, 40(10), 2009, p. 1441 1448. for digital adder," IEEE Students' Technology Symposium (TechSym), 2014.

[4] A. Mukherjee and A.S. Dhar, "Real-time fault-tolerance with hotstandby topology for conditional sum adder," Microelectronics Reliability, 55(3), 2015, p. 704-712.

[5] M. Nicolai is, "Time redundancy based soft-error tolerance to rescue nanometer technologies," 17th IEEE VLSI Test Symposium, 1999.

[6] C. Khedhiri, M. Karmani, B. Hamdi, Ka Lok Man. "Concurrent error detection adder based on two paths output computation," Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops (ISPAW),. 2011.

[7] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," Automata studies, 34, 1956, p. 43-98.

[8] D. Siewiorek and R. Swarz, "Reliable Computer Systems: Design and Evaluatuion," Digital Press, 2014.

[9] P. Reviriego, C.J. Bleakley and J.A. Maestro, "Diverse double modular redundancy: A new direction for soft-error detection and correction," IEEE Design & Test 30(2), 2013, p. 87-95.

[10] A. Ziv and J. Bruck, "Performance optimization of checkpointing schemes with task duplication," IEEE Transactions on Computers, 46(12), 1997, p. 1381-1386.

[11] D.P. Vasudevan, P.K. Lala and J.P. Parkerson, "Self-checking carryselect adder design based on two-rail encoding," IEEE Transactions on Circuits and Systems I: Regular Papers, 54(12), 2007, p. 2696-2705.

Akhila Gorla was born in Telangana, India on September 1995.She received her B.Tech, degree in Electronics and Communication Engineering in the year of 2016, from JNTU Hyderabad, Telangana, India. She is currently pursuing M.Tech degree in VLSI Design from Vignan's Institute of Management and Technology for Women, Hyderabad, Telangana, India. Her research interest in Resource Constrained High Speed VLSI Architectures, Internet of Things and Low-complexity Digital Arithmetic for Signal Processing Applications.

.

Harikrishna Ponnam was born in Telangana, India on July 1984. He received his B.Tech, degree in Electronics and Communication Engineering in the year of 2006, M.Tech degree in VLSI System Design in the year of 2011 from the JNTU Hyderabad, Telangana, India. He is currently pursuing Ph.D degree in the department of ECE, Vignan's Foundation for science, Technology & Research, Guntur, India. His research interest in low power signal processing techniques, Internet of Things, and Real-time healthcare monitoring Applications.