Engineering Science and Technology, an International Journal 19 (2016) 1465–1472

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology,
an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

Real-time fault tolerant full adder design for critical applications

Pankaj Kumar *, Rajender Kumar Sharma

National Institute of Technology, Kurukshetra, India

ARTICLE INFO

Article history:
Received 6 January 2016
Revised 11 April 2016
Accepted 1 May 2016
Available online 9 May 2016

Keywords:
Adder
Single fault
Double fault
Self checking adder
Fault tolerant
Reliability

ABSTRACT

In the complex computing system, processing units are dealing with devices of smaller size, which are sensitive to the transient faults. A transient fault occurs in a circuit caused by the electromagnetic noises, cosmic rays, crosstalk and power supply noise. It is very difficult to detect these faults during offline testing. Hence an area efficient fault tolerant full adder for testing and repairing of transient and permanent faults occurred in single and multi-net is proposed. Additionally, the proposed architecture can also detect and repair permanent faults. This design incurs much lower hardware overheads relative to the traditional hardware architecture. In addition to this, proposed design also provides higher error detection and correction efficiency when compared to the existing designs.

1. Introduction

Now days, fault tolerant system is very crucial in the critical applications, where the immediate human action is not possible. Space application, defense surveillance, medical supervisory system and other safety related services are the example of such critical applications. The presence of the faults in such applications can destroy the functionality of the overall system. The complexity of integrated circuits is increasing with the advancement of technology. Technology advancement results in reducing the size of integrated circuits. This makes the design more compact and sensitive to the transient faults. The main reason of the presence of transient fault in the integrated circuit is 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. The compact design is good for reducing the noise but it will increase the chance of hardware failure in the advanced processor [1]. It is very difficult to detect these faults during offline testing. Therefore, these problems are the challenges for the researchers working in the field of on-line fault detection and correction techniques.

Digital Signal Processing is an important unit in electronics devices. Addition is a most fundamental arithmetic operation performed in many Very Large Scale Integration (VLSI) systems such as Digital Signal Processors (DSPs) and microprocessors [2–5]. Full adders are used for variety of operations in a complex arithmetic circuit like multiplication, division and address calculation [6–11]. These full adders are the nucleus of any system. In most of the systems full adders are encountered in critical path and can significantly influence the performance of any system. The design criteria for full adder are usually multifold. Transistor count is the primary concern, which determines the system complexity of the arithmetic circuits like multiplier and Arithmetic Logic Unit (ALU) e.t.c. [12,13]. Therefore, to design a fault tolerant full adder with lesser area overhead is a matter of great importance. Many researchers have been worked on different types of fault tolerant full adders [14–17]. Arithmetic residue code is used for designing the first self checking full adder. This adder can detect the single fault at a time [18,19]. However, the limitations of arithmetic residue codes are its complex checker circuits and incompatibility with its self checking memory systems [20]. After this, many of the self checking approaches used re-execution of instruction for repairing the fault but it will increase the propagation delay of the design up to a great extent. However, fault recovery is not possible in this approach if the faults are permanent [15]. In the literature, researchers have introduced time based redundancy and hardware redundancy techniques to detect the faults in the full adder circuits. These approaches are limited to the single fault detection at a time and can't detect the exact location of the fault. Hence, it makes the other module faulty due to the propagation of the carry. Therefore, to remove these problems, a new fault tolerant full adder is proposed. This design can detect both single and double fault, in addition to the permanent fault, at a time with

* Corresponding author.
E-mail addresses: pankajkumar_6120011@nitkkr.ac.in (P. Kumar), rksharma@nitkkr.ac.in (R.K. Sharma).

Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2016.05.001
2215-0986/© 2016 Karabuk University. Publishing services by Elsevier B.V.

the indication of the exact location of fault. Additionally, this design can repair all the detected faults with the smaller area overhead of the repairing unit. Carry select adder (CSA) is considered for the proposed design because it results the fastest addition and also has smaller area overheads.

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 fault tolerant full adder is described. The simulation results of the entire referred self checking, self repairing and proposed fault tolerant full adder are presented in Section 4. The proposed fault tolerant design is compared with the existing design in the Section 5. In Section 6, fault tolerant multiplier is implemented. Section 6 draws the conclusion.

2. Previous self checking and self repairing design approaches

In the literature, many self checking and self repairing approaches are available. All the approaches have its own advantages and disadvantages. Some of the popular approaches are described below.

2.1. Time redundancy

Redundancy is required in the self checking system. Time redundancy approach is proposed to protect the design from the transient fault. In this approach, similar operation is performed by the duplicate hardware in addition to the original hardware, at different interval of time [21]. Delayed clock is used to provide the difference in time interval of the duplicate hardware as shown in Fig. 1. Finally, the fault is detected by comparing the two outputs obtained at different interval of time. 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.

Authors in [22] also proposed a fault detectable adder based on the concept of time redundancy. This design reduces the area overhead and cost of the design by performing the similar operation at different interval of time. The results are compared to indicate the presence of fault. However, the main limitations of this design are 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. Additionally, this design cannot detect the stuck-at fault.

2.2. Hardware redundancy

Hardware redundancy required more than one hardware to produce the different outputs. The outputs of the original and redundant hardware are compared to indicate the faulty and fault free conditions. Triple modular redundancy and double modular redundancy are the most commonly used redundancy schemes [23,24].

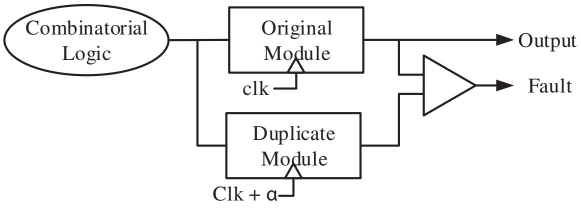

Fig. 1. Time redundancy fault detection technique.

2.2.1. Triple-modular redundancy (TMR)

This is the conventional method to detect the single fault. This approach requires three identical modules in parallel to detect the fault as shown in Fig. 2. A fault is detected if the outputs of the modules are different [25]. Therefore, the reliability of this algorithm is based on majority function and it will not indicate the exact location of the fault. Therefore, this approach creates the problem if two single full adder cells become faulty and produce the faulty output. This is due to the absence of self repairing circuits. This problem can be removed by increasing the hardware but the resulting design requires more than 500% hardware. However, the absence of the self repairing process can increase the probability of more than one fault.

In addition to this, the major drawbacks of this approach are that it requires more than 300% area overhead due to the triplication of the primary module. Secondly, the detection of double fault at a time is not possible in this approach [26]. The another problem is that it works on majority function if two modules generate the faulty output then the correct output of the third module is treated as faulty [27]. The detection of the double fault is not possible in this approach

2.2.2. Double-modular redundancy (DMR)

In this approach, the original module is duplicated to perform the similar operation in parallel as shown in Fig. 3. This approach can detect the single fault at a time by comparing the outputs of operation performed in parallel. Hence, this approach becomes successful to increases the reliability of the design at the minimum cost [28]. But, fault correction is not possible in this approach because the voting circuit can not detect the location of the faulty module [25].

The major drawback of this approach is that it requires more than 200% area overhead and cannot detect double fault at a time [22]. 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.

2.3. Operand width aware hardware reuse (OWHR)

This approach is based on Narrow width values (NWV) and used for designing the fault tolerant Arithmetic and logical unit (ALU). In this approach, most of the data coming into the ALU consists of NWVs. Hence, the arithmetic and logical operations are divided into two parts. Normal operations are performed over the first part, whereas the second part is used for the redundant operation. Consider the case of 64-bit full adder where 32-bit is used for addition and the remaining 32 bit is unused due to NWV. Thus, this 32 bit is used for the parallel operation like in duplex. In this way, a duplex system with minimum hardware cost has been obtained.

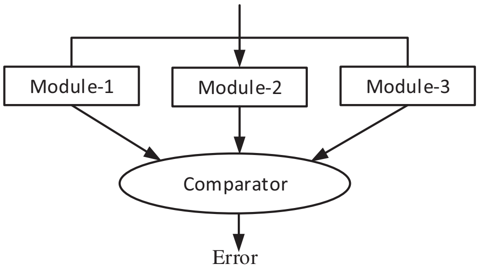

Fig. 2. Triple modular redundancy fault detection technique.

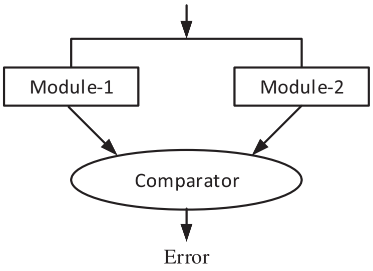

Fig. 3. Double modular redundancy fault detection technique.

The main limitations associated with this approach are that it increases the computational complexity and associated power overhead of the design. Secondly, this approach has the problems of common mode failure and fault propagation due to the carry propagation. In addition to this, single fault can only be repaired if both the operands are NWVs. The reliability of the system is also dependent on NVWs.

2.4. Self -checking CSA

Self checking carry select adder is proposed by Vasudevan et al. [29]. This design can detect any single fault and single-at fault in the design online. It requires two adders, 2-pair-2-rail-checker, self-checking multiplexers and XOR gates to detect the faults online. The complete self checking design is shown in Fig. 4. The checker has two outputs 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 and is not capable of identifying the location of faulty adder cells. Therefore, fault recovery is not possible in this approach and complete unit needed to be replaced to resolve the problem. In addition to this, it can't detect the double fault at a time and also have a problem of fault propagation through carry. This design also required very high hardware cost.

2.5. Self repairing adder

Self repairing adder is proposed by Mohammad ali akbar et at. [16]. This design can detect faults and also identify the location of single net and multi-net faults online. It removes the some limitations of the self checking adder proposed in [29]. It requires single full adder cell, equivalent tester and two XNOR gates for self testing the fault while it uses two full adder cells during self repairing of the faults. The XNOR gate (X1) is used for comparing the sum and carry outputs. It works on the principle that the sum and carry outputs will be equal when all the input applied are equal and the sum and carry outputs will be complemented to each other when any of the three inputs applied is different from remaining inputs. The XNOR gate (X2) is used to detect the fault in the full adder cell. The fault is represented in the form of E_f as shown in Fig. 5. The faults, which are detected during the self checking process, can be removed by replacing the faulty full adder cell with another redundant full adder as shown in Fig. 6.

This design requires redundant adder for repairing the single fault. But it fails when double faults occur at a time. In this case, the fault indication output (E_f) of this design shows that there is no fault. Therefore, the fault is not repaired by the self repairing adder. This shows that this design is less reliable. These are the major limitations of this design.

3. Proposed design

3.1. Main Idea

The key idea of the proposed fault tolerant design is as follows.

1. The sum and carry bits are tested individually to improve the efficiency of fault diagnosis.
2. The proposed design does not require two rail checker used in [29] for detecting the error and redundant full adder used in [16] for repairing the full adder. This reduces the overall hardware cost of the proposed design.
3. The proposed fault tolerable CSA architecture can detect and repair transient and permanent faults simultaneously.

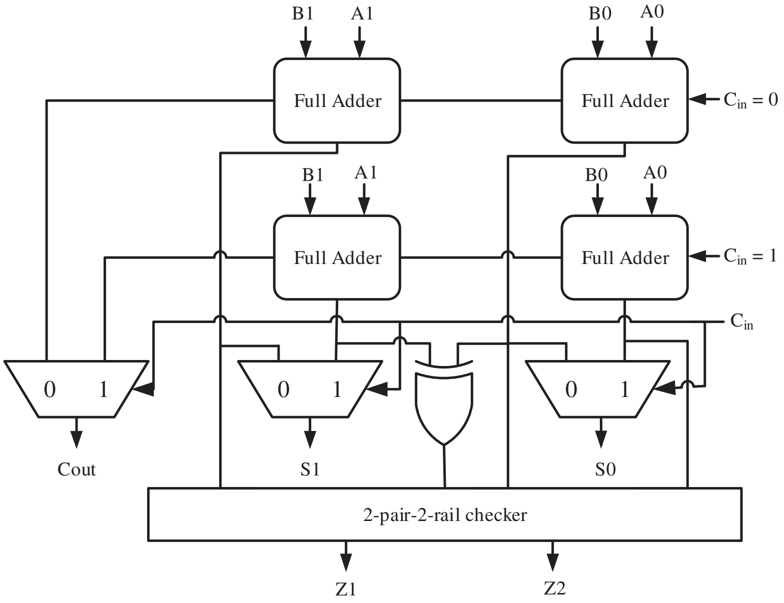

Fig. 4. 2-Bit self checking carry select adder.

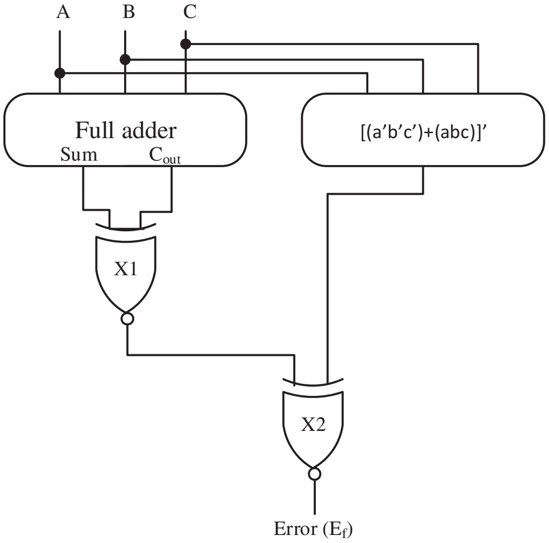

Fig. 5. Self checking full adder.

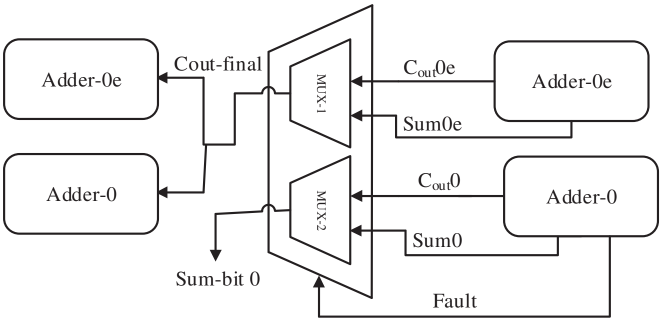

Fig. 6. Self repairing adder.

3.2. Proposed self checking adder with fault localization

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

$$Sum = A \oplus B \oplus C_{in} \tag{1}$$

$$Count = AB + BC_{in} + C_{in}A \tag{2}$$

The proposed self checking full adder can detect the faults with the indication of its exact location. In this design, carry select adder (CSA) is used, whereas the Ripple Carry Adder (RCA) is simple in architecture and working. The drawback of RCA is that it requires more computation time. However, CSA is the fastest adder and also requires minimum hardware cost compared to other adders except carry skip adder. In the proposed design, both the sum and carry outputs of CSA full adder are checked online individually to detect the fault in both the outputs. The detailed diagram of the proposed self checking full adder is shown in the Fig. 7.

XNOR gates and equivalent functional unit are used for detecting the fault at the carry output. XNOR gates are design using Double Pass transistor Logic (DPL) based logic. The output of the functional unit (A′B′C + ABC′) and (XNOR-1) are compared using the (XNOR-5) to detect the fault. The output expressions of (XNOR-1), functional unit and (XNOR-5) are represented in the form of (G1), (F1) and Fc and shown in the Eqs. (3)–(5), respectively.

$$G1 = \left(\overline{C_{out} \oplus C_{in}}\right) \tag{3}$$

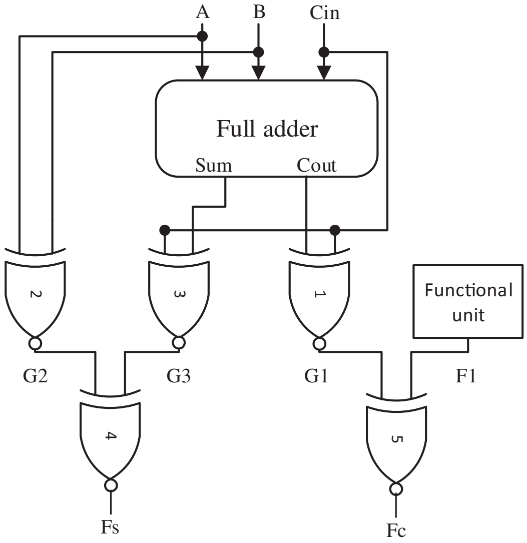

Fig. 7. Proposed self checking full adder design.

$$F1 = \left(\overline{A'B'C + ABC'}\right) \tag{4}$$

$$Fc = \left(\overline{G1 \oplus F1}\right) \tag{5}$$

If the output (Fc) indicates 0, it represents the fault free condition. On the other side, if the output of (Fc) indicates 1, it shows that the carry output of the full adder is faulty. Similarly, for detecting the fault in the sum output three XNOR gates are used. The output of the XNOR-2 and XNOR-3 are compared using the XNOR-4 to detect the fault. The output expressions of (XNOR-2), (XNOR-3) and (XNOR-4) are represented in the form of G2, G3 and Fs and given in Eqs. (6)–(8), respectively.

$$G2 = \overline{(A \oplus B)} \tag{6}$$

$$G3 = \left(\overline{Sum \oplus C_{in}}\right) \tag{7}$$

$$Fs = \left(\overline{G2 \oplus G3}\right) \tag{8}$$

If the output (F_S) is 0, it represents the fault free condition. On the other hand, if the output (Fs) is 1, it shows the fault in sum output. In this way, the proposed design can detect the single and double faults occurred in at a time. Finally, the faults, detected at the sum and carry output are represents in the form of Fs and Fc, respectively. If any one or both of these signals indicate high, it represents the fault in the corresponding outputs. In case of no fault, both the signals indicate low.

3.3. Proposed self repairing adder

The proposed self repairing adder is used for repairing the faults detected during the self checking approach of the carry select adder. This design guarantees to repair all the faults (transient and permanent) and makes the adder completely fault free. The proposed self repairing design does not need any standby adder cell to replace the faulty adder as used in the previous self repairing full adder [16]. In this approach, faults are repaired by using an inverter in place of the standby full adder cell. Therefore, the proposed self repairing approach requires a negligible area compared to existing self repairing approach [16].This approach is based on the principle given below.

1. The sum and carry outputs will be either 1 or 0 depending on the input combination applied to the full adder cell.
2. If the signal Fs indicates the fault in sum output and the inverted sum output is selected by the multiplexer under the control of Fs.
3. If the signal Fc indicates the fault in carry output, inverted carry output is selected by the multiplexer under the control of Fc.

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 as shown in Fig. 8. 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 as shown in Fig. 8. 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.

3.4. Simulation results and comparison

The proposed fault tolerant full adder and some popularly known self checking and self repairing full adder architectures have been implemented using UMC 55-nm standard cell library in cadence virtuoso tool. The hardware overhead of the proposed design is found better in comparison to the existing self repairing design [16] and shown in Table 1.

The hardware overhead is computed on the basis of the transistor count. The proposed fault tolerant full adder requires only one self checking full adder cell and an inverter in place of redundant full adder, which is used in the previous design. The hardware cost of the proposed and self repairing adder can be calculated and compared using the Eq. (9) [30].

$$Area_{overhead} = \frac{Area_{withFT} - Area_{withoutFT}}{Area_{withoutFT}} \times 100\% \tag{9}$$

The graph shown in Fig. 9, shows the comparison of the proposed design and self repairing adder on the basis of transistor count and area overhead.

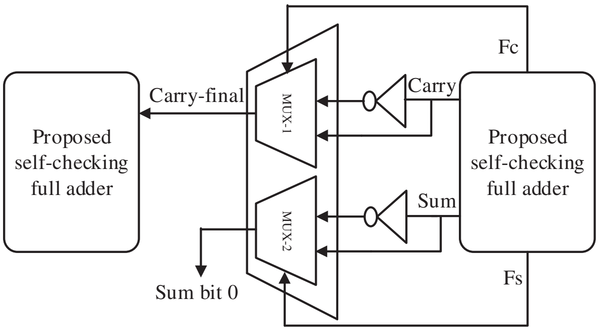
Fc
Carry-final
Carry
MUX-1
Proposed self-checking full adder
Proposed self-checking full adder
Sum
MUX-2
Sum bit 0
Fs

Fig. 8. Proposed self repairing full adder design.

Table 1
Comparison of area overhead.

Designs	Individual transistor count	Total number of transistors	Area overhead (μm^2)
Proposed fault tolerant design	1-Adder 28 5-XNOR 30 1-Fun. Unit 12 2-Mux 08 2-Inverter 04	82	192.85%
Self repairing adder	2-Adder 56 4-XNOR 24 2-Eqt 24 2-Mux 08	112	300%

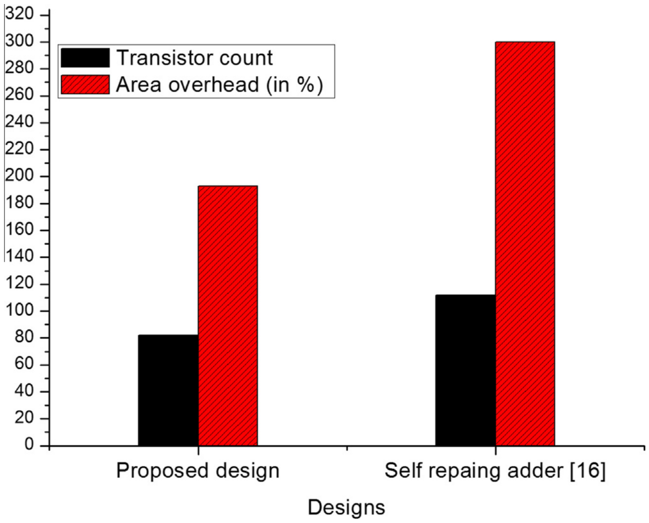

Fig. 9. Comparison of transistor count and area overhead of proposed and self repairing adder [16].

3.5. 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. 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 faulty and fault free conditions are described below.

3.6. Faulty conditions

Faults are shown in the form of Fs and Fc. If there is any fault at the carry in or other input signals applied to the full adder cell, it will change the value of the sum and carry output. After comparison, we found both the signals Fs and Fc. If any one of them will be logic high, this condition is known as faulty output. All the faulty conditions of the proposed and existing designs, which are responsible for the fault generation are given in Table 2 with the status of fault repairing.

3.7. Fault-free conditions

If the fault tolerance full adder does not detect any permanent and transient fault at the input and output terminals, the sum and carry output will not modify and the output will be fault free. In this case, both Fs and Fc will be at logic low and the actual sum and carry outputs of the full adder cell will be selected by the

Table 2
Comparison of fault coverage and repairing.

Designs	Fault coverage	Conditions	Fault repairing
Proposed	Single-net	$\left.\begin{array}{l} Fs = 0 \\ Fc = 0 \end{array}\right\}$ Fault-free	Self repairing is possible in all the faulty cases
	Multi-net	$\left.\begin{array}{l} Fs = 1 \\ Fc = 0 \end{array}\right\}$ Fault in sum	
	Single fault	$\left.\begin{array}{l} Fs = 0 \\ Fc = 1 \end{array}\right\}$ Fault in carry	
	Double fault	$\left.\begin{array}{l} Fs = 1 \\ Fc = 1 \end{array}\right\}$ Fault in both	
	Stuck-at fault		
Self repairing [16]	Single-net	$\left.\begin{array}{c} Eqt = 0 \\ Sum = cout \end{array}\right\}$ Fault -free	Self repairing is not possible when double faults occur at a time
	Multi-net	$\left.\begin{array}{c} Eqt = 1 \\ Sum = cout' \end{array}\right\}$ Fault-free	
		$\left.\begin{array}{c} Eqt = 0 \\ Sum{\neq}Cout \end{array}\right\}$ Faulty	
	Single fault	$\left.\begin{array}{c} Eqt = 0 \\ Sum{\neq}Cout' \end{array}\right\}$ Faulty	
DMR	Single-net	$Ecom = 0\}$ Fault-free $Ecom = 1\}$ Faulty	Self repairing is not possible
TMR	Single-net	$Ecom = 0\}$ Fault-free $Ecom = 1\}$ Faulty	Self repairing is possible with 500% hardware cost

Table 3
Comparison of fault recovery and reliability.

	Fazeli et. al.[31]	TMR	Self-repairing [16]	Proposed
Fault recovery	Fault recovery is possible if 1. Both the operands should be NWV 2. Only one module is faulty	Fault Tolerance is possible if only one module goes faulty at a time	Fault Recovery is possible if double fault does not occur simultaneously	Fault recovery is possible even if double fault occur simultaneously
Output reliability on single fault	56%	100%	100%	100%
4-Bit adder output reliability on second fault	27%	27%	85.82%	100%

multiplexer to generate the final sum and carry. This condition is known as faulty-free output and all the fault free condition of the proposed and existing designs are given in Table 2.

3.8. Reliability comparison

In case of fault tolerant approach proposed by Fazeli et al. [31], fault recovery is possible with the adoption of the TMR approach and when all the input bits are NVW. The resulting calculation based on ARM processor showed that this approach is 56% reliable when single fault occur in a full adder module. However, the reliability decreases with the increase of number of faults at a time, which are shown in Table 3.

The output reliability of TMR is 100% if only one module goes faulty at a time. The problem is arising when two modules become faulty at a time. This reduces the reliability and the TMR fails to provide the reliable output as shown in Table 3. This happens due to the absence of the self repairing module. However, if a module is added for self repairing it will require a hardware cost of more than 500%. The probability of the fault recovery can be computed by using the expression given below.

$$Probability_{(\mathrm{TMR})} = \frac{\binom{n}{r} \times 3}{\binom{N}{r}} \qquad (10)$$

where N indicates the total number of full adder and r introducing the random fault. n is the total number of full adder present in all the modules.

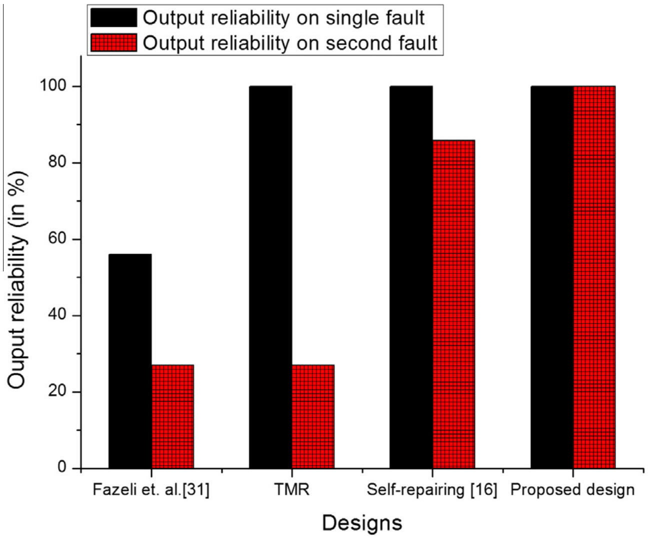

Fig. 10. Comparison of fault recovery and reliability of the proposed and existing designs.

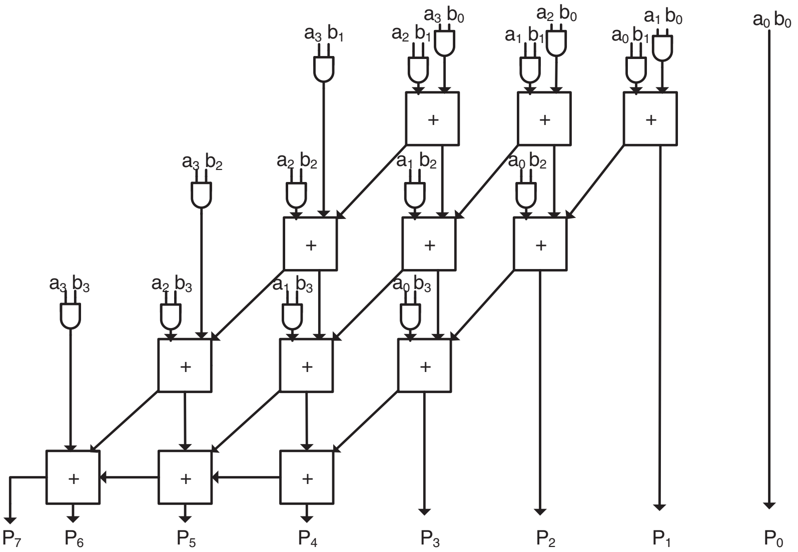

Fig. 11. Fault tolerant multiplier design using 1-bit proposed self repairing full adder.

The output reliability of the self repairing full adder is 100% if a single fault occurs at a time. The output reliability of this design can be calculated with the expression given below.

$$Probability_{(self\text{-}repairing)} = \left(1 - \frac{\binom{N-2}{r-2} \times n}{\binom{N}{r}}\right) \quad (11)$$

where n indicate the number of full adders in a single module. N represents the total number of full adders in all modules and r shows the total number of faults. In case of double fault occur at a time, reliability decreases to 85.82% as shown in Table 3. This happens due to the in-capability of self repairing adder [16] to detect and repair the double fault occurred at a time. The proposed fault tolerant full adder is capable in repairing both single and double faults occur at a time. Hence, the proposed design has 100% reliability, if the single and double faults occur. The graph of Fig. 10, shows the comparison of fault recovery and reliability of the proposed and existing designs. It shows that the proposed design is highly reliable than the existing designs given in the literature.

4. Comparison with existing fault tolerant full adder designs

1. The proposed design is capable of detecting and repairing the transient and stuck-at faults occur in single and multi-net. However, the design proposed in [29], TMR and DMR approaches can't detect the stuck-at fault.
2. The fault detection capability of the DMR-based adder is 100% for the single fault at a time. However, error correction capability of TMR-based adder is less than 100% because error occurs in voter circuit is not corrected. But it is limited to the single error. The proposed circuit is more reliable than the DMR and TMR-based adder against single and double fault detection and correction.
3. 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 [29] and self repairing full adders [16] have the problem of fault propagation through carry.
4. The proposed fault tolerant full adder does not require redundant full adder for repairing the faults. However, the self repairing design [16] requires the redundant adder for repairing the fault. Hence the area acquired by the proposed design is less.

Table 4
Power, delay and PDP behavior of the 4-bit and 8-bit proposed fault tolerant multiplier designs.

Designs	Power (μW)	Delay (pS)	PDP (aJ)
4-Bit proposed fault tolerant multiplier design	94.38	598.3	56467.554
8-Bit proposed fault tolerant multiplier design	712	1326	944,112

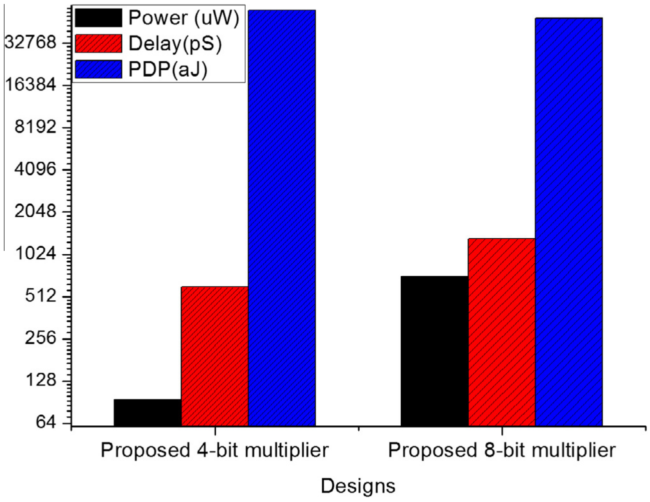

Fig. 12. Power, delay and PDP of 4-bit and 8-bit proposed fault tolerant multiplier.

5. The proposed design requires an inverter to repair the faults occur in sum and carry outputs. However, design in [16] requires the redundant adder to repair the faults occur in sum and carry outputs. Therefore, the proposed design has minimum chances of common mode failure than the existing designs.

5. Performance of proposed fault tolerant multiplier design

A fault tolerant multiplier shown in Fig. 11, is implemented as an application of the proposed fault tolerant full adder. The full adder shown in Fig. 11, is the proposed self repairing full adder shown in Fig. 8. Simulation results show that the fault tolerant multiplier produces the correct output even if the any type of fault (single or double) occur in the full adder cells. The performance of the proposed 4-bit and 8-bit fault tolerant multiplier is computed at 100 MHz and 1.0 V using UMC 55-nm technology. The Cadence spectre simulator tool is used to estimate the power consumption and propagation delay. Table 4 shows the power consumption, propagation delay and PDP behavior of the proposed multiplier design. The graphical representation of the fault tolerant multipliers are shown in Fig. 12.

6. Conclusion

In this paper, a new technique for self checking and self repairable carry save adder with minimal area overhead has been proposed. The proposed design can detect and repair both single and double faults at a time online. Hence, this design is free from the problem of fault propagation through carry. The proposed fault tolerant full adder is compared in terms of single and multi-net error detection and correction possibility with the DMR-based, TMR-based, self testing and self repairing full adder designs. The comparison results of the proposed designs are found better that ensure its superior performance capability. The proposed design is extendable up to a desirable level. A 8-bit fault tolerant multiplier is also implemented using the proposed design. It works efficiently when cascaded and can handle single and multi-net faults successfully.

References

[1] A. Pellegrini, R. Smolinski, X. Chen, X. Fu, S.K.S. Hari, J. Jiang, S.V. Adve, T. Austin, V. Bertacco, CrashTest'ing SWAT: accurate, gate-level evaluation of symptom-based resiliency solutions, in: International Conference on Design, Automation and Test, Europe, 2012.

[2] K. Navi, V. Foroutan, M. RahimiAzghad, M. Maeen, Ebrahimpour, M. Kaveh, O. Kavehei, A novel low-power full-adder cell with new technique in designing logical gates based on static CMOS inverter, Microelectron. J. 40 (10) (2009) 1441–1448.

[3] I. Brzozowski, A. Kos, Designing of low-power data oriented adders, Microelectron. J. 45 (9) (2014) 1177–1186.

[4] M.H. Moaiyeri, M. Nasiri, N. Khastoo, An efficient ternary serial adder based on carbon nanotube FETs, Eng. Sci. Technol. Int. J. 19 (2015) 271–278.

[5] M. Shoba, R. Nakkeeran, GDI based full adders for energy efficient arithmetic applications, Eng. Sci. Technol. Int. J. 19 (01) (2016) 485–496.

[6] C.-K. Tung, Y.C. Hung, S.H. Shieh, G.S. Huang, A low-power high-speed hybrid CMOS full adder for embedded system, in: International Conference on Design and Diagnostics of Electronic Circuits and Systems, 2007.

[7] H.T. Bui, Y. Wang, Y. Jiang, Design and analysis of low-power 10-transistor full adders using novel XOR–XNOR gates, IEEE Trans. Circuits Syst. II: Analog Digital Signal Process. 49 (1) (2002) 25–30.

[8] M. Alioto, G. Palumbo, Analysis and comparison on full adder block in submicron technology, IEEE Trans. Very Large Scale Integr. VLSI Syst. 10 (6) (2002) 806–823.

[9] K. Navi, Keivan Navi, O. Kavehei, M. Rouholamini, A. Sahafi, S. Mehrabi, N. Dadkhahi, Low-power and high-performance 1-bit CMOS full-adder cell, J. Comput. 3 (2) (2008) 48–54.

[10] M.R. Azghadi, O. Kavehie, K. Navi, A novel design for quantum-dot cellular automata cells and full adders. arXiv preprint arXiv:1204.2048, 2012.

[11] M.H. Ghadiry, A.K. A'ain, M. Nadi, Design and analysis of a novel low PDP full adder cell, J. Circuits Syst. Comput. 20 (03) (2011) 439–445.

[12] P. Kumar, R.K. Sharma, Low voltage high performance hybrid full adder, Eng. Sci. Technol. Int. J. 19 (01) (2016) 559–565.

[13] M. Jhamb, H. Lohani, Design, implementation and performance comparison of multiplier topologies in power-delay space, Eng. Sci. Technol. Int. J. 19 (01) (2016) 355–363.

[14] S. Ghosh, K. Roy, Novel low overhead post-silicon self-correction technique for parallel prefix adders using selective redundancy and adaptive clocking, IEEE Trans. Very Large Scale Integr. VLSI Syst. 19 (8) (2011) 1504–1507.

[15] M.H. Hajkazemi, A. Baniasadi, H. Asadi, FARHAD: a fault-tolerant power-aware hybrid adder for add intensive applications, International Conference on Application-Specific Systems, Architectures and Processors (ASAP), 2013.

[16] M.A. Akbar, J.-A. Lee, Self-repairing adder using fault localization, Microelectron. Reliab. 54 (6) (2014) 1443–1451.

[17] C.D. Martinez, L.D. Bollepalli, D.H. Hoe, A fault tolerant parallel-prefix adder for VLSI and FPGA design, IEEE Southeastern Symposium on System Theory (SSST) , 2012.

[18] W. Peterson, On checking an adder, IBM J. Res. Dev. 2 (2) (1958) 166–168.

[19] W.W. Peterson, E.J. Weldon, Error-correcting Codes, MIT press, 1972.

[20] M. Nicolaidis, Carry checking/parity prediction adders and ALUs, IEEE Trans. Very Large Scale Integr. VLSI Syst. 11 (1) (2003) 121–128.

[21] M. Nicolaidis, Time redundancy based soft-error tolerance to rescue nanometer technologies, in: IEEE VLSI Test Symposium, 1999.

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

[23] J. Von Neumann, Probabilistic logics and the synthesis of reliable organisms from unreliable components, Automata Stud. 34 (1956) 43–98.

[24] D. Siewiorek, R. Swarz, Reliable Computer Systems: Design and Evaluation, Digital Press, 2014.

[25] P. Reviriego, C.J. Bleakley, J.A. Maestro, Diverse double modular redundancy: a new direction for soft-error detection and correction, IEEE Des. Test. 30 (2) (2013) 87–95.

[26] M. Valinataj, A novel self-checking carry lookahead adder with multiple error detection/correction, Microprocess. Microsyst. 38 (8) (2014) 1072–1081.

[27] A. Majumdar, S. Nayyar, J.S Sengar, Fault tolerant ALU system, IEEE International Conference on Computing Sciences (ICCS), 2012.

[28] A. Ziv, J. Bruck, Performance optimization of checkpointing schemes with task duplication, IEEE Trans. Comput. 46 (12) (1997) 1381–1386.

[29] D.P. Vasudevan, P.K. Lala, J.P. Parkerson, Self-checking carry-select adder design based on two-rail encoding, IEEE Trans. Circuits Syst. I Regul. Pap. 54 (12) (2007) 2696–2705.

[30] A. Mukherjee, A.S. Dhar, Real-time fault-tolerance with hot-standby topology for conditional sum adder, Microelectron. Reliab. 55 (3) (2015) 704–712.

[31] M. Fazeli, A. Namazi, S.G. Miremadi, A. Haghdoost, Operand width aware hardwarereuse: a low cost fault-tolerant approach to ALU design in embedded processors, Microelectron. Reliab. 51 (12) (2011) 2374–2387.