

Received July 29, 2020, accepted August 4, 2020, date of publication August 13, 2020, date of current version August 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3016427

# Self-Repairing Hybrid Adder With Hot-Standby Topology Using Fault-Localization

MUHAMMAD ALI AKBAR<sup>ID</sup>, BO WANG<sup>ID</sup>, (Member, IEEE),  
AND AMINE BERMAK<sup>ID</sup>, (Fellow, IEEE)

Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha 34110, Qatar

Corresponding author: Muhammad Ali Akbar (maakbar@mail.hbku.edu.qa)

Open Access funding provided by the Qatar National Library. This work was supported by NPRP under Grant NPRP11S-0110-180246 from the Qatar National Research Fund (a member of Qatar Foundation). The findings herein reflect the work and are solely the responsibility of the authors.

**ABSTRACT** Effective self-repairing can be achieved if the fault along with its exact location can be determined. In this paper, a self-repairing hybrid adder is proposed with fault localization. It uses the advantages of ripple carry adder and carry-select adder to reduce the delay and area overhead. The proposed adder reduces the transistor count by 115% to 76.76% as compared to the existing self-checking carry-select adders. Moreover, the proposed design can detect and localize multiple faults. The fault-recovery is achieved by using the hot-standby approach in which the faulty module is replaced by a functioning module at run-time. In case of 3 consecutive faults, the probability of fault recovery has been found to be 96.1% for a 64-bit adder with 8 blocks, where each block has 9 full adders.

**INDEX TERMS** Self-repairing adder, fault localization, hybrid adder, real-time self-repairing.

## I. INTRODUCTION

The possibility of single-event-upset (SEU) in digital systems has risen as a result of the increase of on-chip system complexity as well as reduced clock cycles [1], [2]. The presence of radiation and other environmental conditions further enhance the probability of SEU [3], [4]. To handle SEU, the concept of “totally self-checking” was introduced. A system is characterized as totally self-checking if it remains unaffected by a fault, or produces a non-coded output for every generated fault [5]. In addition to fault detection, fault recovery should also be considered to ensure hardware reliability [6]. This is why the concept of built-in self-repair is becoming increasingly pertinent to current digital systems [7]. Fault recovery however becomes challenging in an inter-connect hardware design because of fault propagation. Therefore, fault localization becomes necessary for such type of hardware design.

Adder is an essential element present in almost all digital systems thus the introduction of built-in self-repair in adders can play a vital role in digital designs [8], [9]. Moreover, the presence of carry propagation chain makes adder an ideal case to understand the phenomenon of handling faults between inter-connected modules. To achieve this, both fault detection and localization should be performed.

The associate editor coordinating the review of this manuscript and approving it for publication was Heng Wang<sup>ID</sup>.

Ripple carry adder (RCA) and carry select adder (CSeA) are among the most commonly adopted adder topologies, hence many of the reported reliable adders are based on these topologies. In [9], a self-checking CSeA using 2-pair-2-rail checker encoding approach was proposed. It uses the advantage of the parallel rail of RCA present in CSeA, where each RCA rail produces an output for one of the initial carry inputs, i.e.  $C_{in} = 0$  or 1. Outputs of the two parallel blocks were compared to detect the presence of fault. This design is only valid for 2-bits and later an improved n-bit CSeA design was proposed in [10]. The relationship between the two parallel rails of RCA in [9] was further utilized in [11] to design a single RCA-Based self-checking CSeA with fault localization. The reported RCA performs addition for  $C_{in} = 0$  and the resulting sum-bits are used to generate the sum-bits for  $C_{in} = 1$ . The design in [11] requires 12% less transistor count than the self-checking CSeA in [9]. In [12], a self-checking CSeA is proposed using parity prediction approach in which the operands are provided to the adder along with their respective parities. It however cannot perform fault localization and has limited fault coverage, because it can only indicate fault if it occurs in odd number of bits. Although these approaches were shown to be effective for SEUs, they can not perform fault recovery with minimum area overhead.

To boost the reliability of adders, the most conventional method is known as triple modular redundancy (TMR), which involves two redundant modules employed to produce

additional outputs, and the final output is selected via a voter circuit [13], [14]. However, the fault propagation phenomenon may cause a common mode failure which cannot be handled by the TMR. To address this issue, a shifted operand approach is used in [15] for TMR based self-checking ALU design. A similar concept of shifted and rotated operands is also used in [16] to minimize the required diversity in ALU architecture for self-checking TMR design. Another problem of TMR is its large area which is at least twice of a normal design. To overcome this limitation, a partial adoption of TMR is utilized in [17]. In this approach, only the most significant bits (MSB) block will be triplicated, which however increases the possibility of system failure to more than 50%. The concept of TMR despite its simplicity, cannot be applied in systems where area is the major concern.

Some non-conventional techniques have also been adopted for reliable adder designs. One of such techniques is the self-repairing signed digit adder proposed in [18], in which fault localization is achieved because of the limited propagation of carry chains to the neighboring adder block. It uses both hardware and time redundancy for self-checking and self-repair. However, this design can only provide fault detection when odd number of bits are faulty, and it is sensitive to the parity predictor and error indicator. In [19] a self-repairing conditional sum adder (CoSA) with single spare hot-standby approach is presented. However, it only provides self-checking and repairing for the conditional selection cell module which is the building block of CoSA. This limited fault coverage approach makes the design less robust. These techniques provide less area overhead than TMR but the fault coverage is limited and also fault recovery is not always possible.

In this paper, a self-checking and repairing hybrid adder (HA) design with reduced area and time overhead is proposed. The proposed adder utilizes the advantage of the low complexity of RCA and the high speed of CSeA. Fault-detection and localization are realized by using a self-checking full-adder (SFA), in which fault detection is independent of the propagated carry. To minimize the area, a single RCA based CSeA approach is adopted together with a hardware friendly implementation using pass transistor logic. Moreover, square-root topology is used to reduce the delay in the proposed design. The proposed self-checking HA with fault localization and multiple fault detection feature, requires on average 50% more transistor count as compared to traditional CSeA design. A distributed fault-recovery mechanism using hot-standby approach is further proposed to reduce the probability of system failure.

The remainder of this paper is organized as follows. Section II describes the proposed self-repairing HA design. Comparative analysis with previous approaches is presented in Section III. Finally, concluding remarks are presented in Section IV.

## II. PROPOSED SELF-REPAIRING HYBRID ADDER DESIGN

The self-repairing HA design is proposed by considering the area overhead, delay and the fault coverage.

### A. HYBRID-ADDER DESIGN

The time required for CSeA to compute the lowest bits is more than the required time for RCA. This additional delay is caused by the MUX. Therefore, if a simple RCA for initial bits is employed, the design will be more efficient in terms of hardware and time-delay. The complexity will also be reduced with the use of RCA as the beginning block. This is why in the proposed HA design, the least significant bits are computed using RCA, while a single RCA-based CSeA is used for computing the higher bits, as shown in Fig. 1(a) and (b). In addition to this, the proposed HA design follows the square-root topology because a linear CSeA design has similar time delay as that of simple RCA. Therefore, sub-linear delay approach has been considered to balance the delay path by dividing the adder in to blocks where the size of the block increases linearly from  $m, m + 1, \dots, m + l$ .

It should be noted that RCA-Block (RBL) is the fundamental building block of RCA, shown in Fig. 1(a), whereas, the CSeA constitutes of two fundamental blocks that is the initial block (INL) and the Adder Block (ABL), as shown in Fig. 1(b). The reason for having two fundamental block for constructing CSeA is because of the basic principle of single RCA based CSeA design which states that:

*Except for the least significant bit which are always complement to each other, the Sum bit computed for complement value of initial  $C_{in}$  will also be complement to each other if all the lower Sum bits are equal to logic 1.*

The initial block (INL) is therefore responsible for generating the least significant Sum bit by taking the complement of the Sum bits generated at initial  $C_{in} = 0$ . All the other Sum bits will be generated by using the Adder Block (ABL) in which the AND gate is used to determine the status of the previous SUM bits computed for  $C_{in} = 0$  while the XOR gate generates the corresponding SUM bit for  $C_{in} = 1$  by considering the status of the previous SUM bits. The number of ABL used for designing CSeA block is equal to the  $(size\_of\_the\_CSeA\_block - 1)$ .

In Fig. 1(b), the partial Sum and  $C_{out}$  bit is represented by  $S_i^j$  and  $C_i^j$ , respectively, where  $j$  indicates the initial  $C_{in}$  and  $i$  indicates the bit number. The fault is indicated by the error signal  $E_f$ . The final  $C_{out}$  will be generated by using the Module of Final  $C_{out}$  (MOFC). The  $C_{out}$  generated by the MOFC after each CSeA block will be treated as an actual  $C_{in}$  for the next CSeA Block. Whereas, the  $C_{out}$  of RCA block is used as an actual  $C_{in}$  for the first CSeA block.

### B. FAULT DETECTION AND LOCALIZATION

Fault localization is achieved by using the approach of self-checking, independent of the propagated carry. In [11], a self-checking full adder was presented which can detect a fault

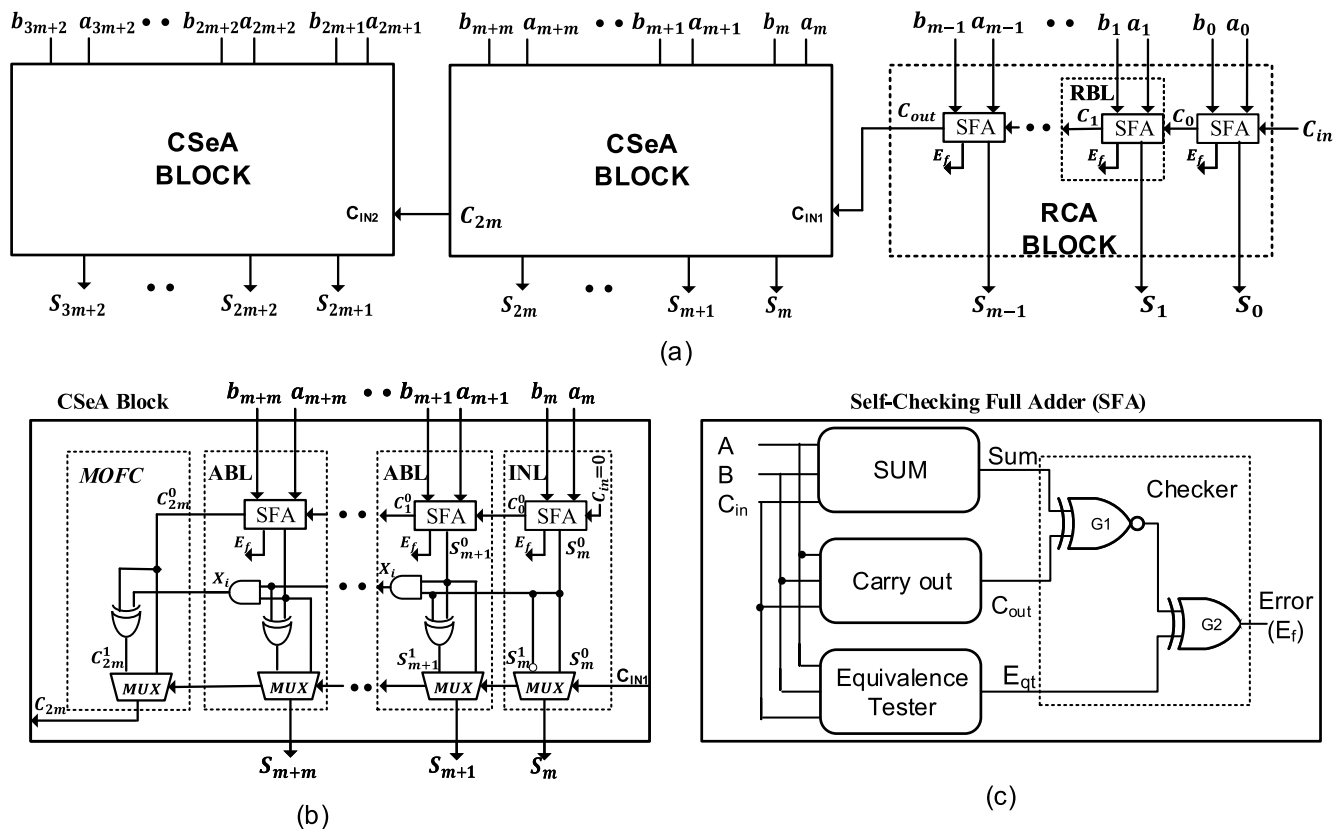


FIGURE 1. (a) HA design with RCA block for least significant bits (b) Self-checking CSeA block and (c) self-checking full adder (SFA) design.

based on its internal functionality and is independent of the propagated carry. The relationship between input and output bits of full adder was utilized for self-checking. Consider a full adder with inputs A, B, \$C\_{in}\$, and the outputs Sum, \$C\_{out}\$, as shown in Fig. 1(c). The fault will not be indicated until Property 1 remains valid for that full adder:

Property 1:

$$\begin{aligned}
 & \text{if } (A == B == C_{in}) \\
 & \text{Then } (Sum \oplus C_{out} == 0) \\
 & \text{Else } (Sum \oplus C_{out} == 1)
 \end{aligned}$$

It can be observed from Fig. 1(c), that the self-checking full adder can be designed with the expense of an extra Equivalence Tester (\$E\_{qt}\$) bit, which is required to indicate the relationship of the input bits. The \$E\_{qt}\$ will be equal to 1 if all input bits are equal and vice versa. Hence, the following three equations from Eq. (1) to (3) need to be implemented for designing a self-checking and fault localized adder.

Since the goal of this design is to reduce the area overhead without compromising the reliability, Equations Eq. (1) to (3) which are used for designing a self-checking and fault localized full-adder need to be implemented with minimum transistor count. A high speed and area efficient full adder design is found in [20]. However, this approach cannot be adopted completely because of the logic sharing between Sum and

\$C\_{out}\$, due to which the probability of common mode failure increases. Therefore, the equation and transistor level implementation of \$C\_{out}\$ has only been adopted from their design.

The final implementation of Eq. (1) to (3) using pass transistor-based approach is shown in Fig. 2.

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

$$C_{out} = \overline{(A \odot C_{in}) \cdot \overline{A} + (A \oplus C_{in}) \cdot \overline{B}} \tag{2}$$

$$Equ.Tester(E_{qt}) = \overline{(A \oplus B) + (A \oplus C_{in})} \tag{3}$$

C. SELF-REPAIRING APPROACH

A hot-standby approach has been adopted for fault recovery. In this approach, if the fault is detected in any of the full adders, the generated error signal will shift the input bits such that the faulty adder will not be used for computation. The main challenge in doing this shift operation is the carry chain which is linked between each consecutive full adder, and the \$X\_i\$ bit which is indicating the status of all previous Sum bits in each CSeA block, as shown in Fig. 1(b). The problem of carry chain has been resolved by making \$C\_{out}\$ to be dependent on error signal \$E\_f\$ of the SFA. In case of fault, the \$C\_{out}\$ (i.e. \$C\_i\$) will be equal to \$C\_{in}\$ (i.e. \$C\_{i-1}\$).

$$C_{out} = \begin{cases} C_{out}, & \text{if } E_f == 0 \\ C_{in}, & \text{otherwise} \end{cases}$$

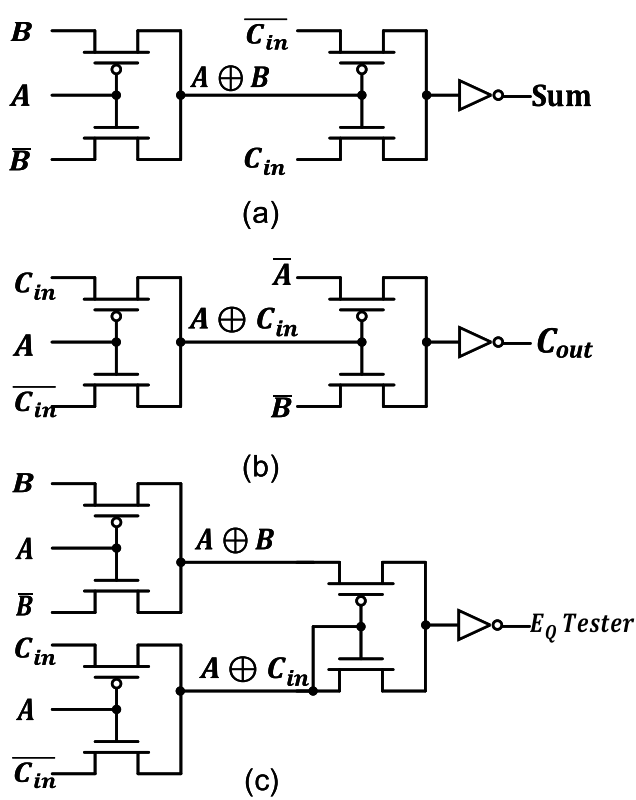


FIGURE 2. Pass-transistor based sub-modules of (a) Sum, (b) Cout and (c) Equivalence tester.

Since,  $X_i$  indicates the status of all previous Sum bits computed for initial  $C_{in} = 0$ , if any previous Sum bit is equal to 0 then  $X_i$  will be 0, else it will be 1. The Sum bit of each ABL is dependent on the previous value of  $X_i$ , therefore in case of fault detection the value of  $X_i$  should not be updated for the next ABL block. In order to achieve this, the Error signal ( $E_f$ ) has been used to replace the SUM bit in case of fault detected, because  $X_i$  is produce through an AND gate and if the current Sum bit value is set to logic 1 then the previous value of  $X_i$  (i.e.  $X_{i-1}$ ) will be propagated.

$$X_i = \begin{cases} X_i, & \text{if } E_f == 0 \\ X_{i-1}, & \text{otherwise} \end{cases}$$

Note that  $X_i$  is only propagated to the ABLs present in each CSeA block along with the next MOFC block, and it will not be propagated to the next CSeA block because each CSeA block is independent of the previous block. In order to accommodate all these changes, the fundamental blocks for extending the CSeA to an n-bit CSeA that is ABL and INL in Fig. 1(b), has been modified, as shown in Fig. 3(a) and (b), respectively.

Since the carry-chain exists in RCA block as well, the fundamental block of RCA (i.e. RBL) has also been modified, as shown in Fig. 3(c). However, the OR gate present in the modified RBL is not applicable for the first full adder of RCA because of the absence of any previous error signal. The final SUM bit generated by the adder also needs to be shifted

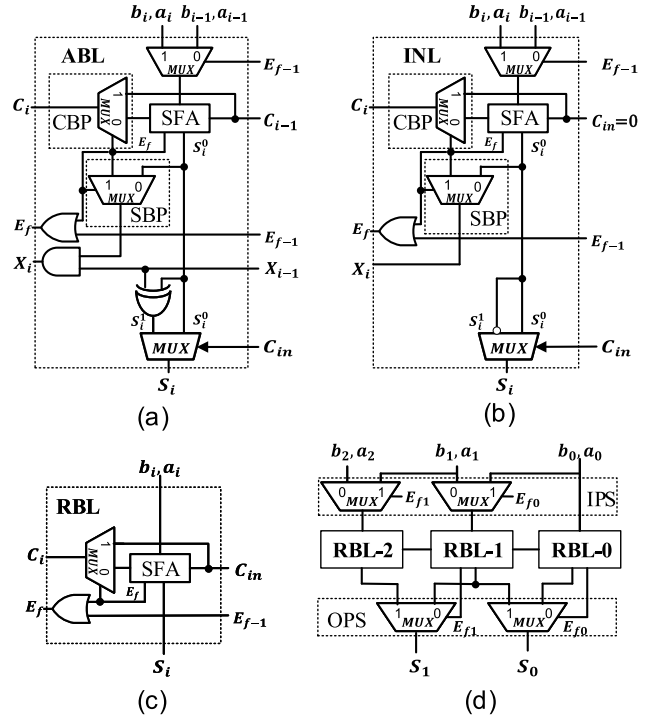


FIGURE 3. Fundamental blocks for constructing the self-repairing HA using (a) ABL (b) INL (c) RBL for CSeA and RCA block along with the shifting approach (d) for input and Sum-out in case of fault detection.

in order to accommodate the shifted operands. Therefore, additional multiplexers have been used to perform the shift operation for SUM bits, as shown in Fig. 3(d).

The self-repairing part is only limited to the number of spare SFA. However, the self-checking property of adder block remains active even if the fault recovery is not possible, which illustrates that, after replacement if any SFA gets faulty then the fault will be indicated but cannot be handled. In order to improve the rate of recovery for larger adder size of more than 8-bit, the number of spares needs to be increased such that each block has one spare module, which means that each block can handle single fault recovery at time. The reason of keeping a single spare in each block is because the probability of having multiple faults in smaller blocks is less than the larger blocks. To illustrate this idea, let  $n - bit$  adder is divided in to  $N$  blocks with each block have  $t$  full adders. Let  $r$  random faults be introduced to the system, then the probability of having  $x$  faults in a same block without replacement can be computed by Eq. (4). where; the range of  $x$  will be equal to  $0 < x \leq t$ . However, the system will not be able to recover the fault in a block if  $x > 1$ . The recovery will still remain possible in all other blocks where  $x < 1$ . Therefore, the probability of the system failure when every single block gets more than 2 faults is given by Eq. (5). The probability of fault recovery in a block can be computed by Eq. (6).

$$Probability_{block-failure}(P_B) = \frac{{}^t C_x * {}^{t(N-1)} C_{r-x}}{t^N C_r} \quad (4)$$

$$Probability_{system-failure}(P_s) = P_B * N \quad (5)$$

**TABLE 1. Probability of fault recovery for X = 2 and R = 3.**

Size of Adder	16		32		64	
No. of Blocks	2	4	2	4	4	8
No. of FA/Block	9	5	17	9	17	9
$P_s$ (%)	79.4	52.6	77.2	54.4	55.3	30.4
$P_B$ (%)	39.7	13.1	38.6	13.6	13.8	3.8
$P_R$ (%)	60.2	86.8	61.3	86.3	86.1	96.1

$$Probability_{Block-Recovery}(P_R) = 1 - \frac{{}^t C_2 * {}^{t(N-1)} C_{r-2}}{{}^{tN} C_r} \quad (6)$$

The probability of fault recovery if 2 out of 3 faults occurred in a single block of the adder is shown in Table. 1. To analyze the impact of block size on fault recovery, three different size of adders are considered such that each adder is constructed using two different block sizes. It can be observed from Table. 1 that the number of full adders in each block decreased with the increase in number of blocks. Whereas, the number of spare modules increased with number of blocks because each block has single spare module for recovery. The overall size of adder can be determined by Eq. 7. It can also be observed from Table.1 that increasing the number of full adders in a block will increase the probability of failure of that block. For example, a 32-bit adder can be built by using 2 blocks and 4 blocks, each of which have 17 and 9 full-adders, respectively. It can easily be observed that as the number of blocks increases from 2 to 4 the probability of block failure decreases from 38.6% to 13.6%. However, the area-overhead of adder will also increase with the increase in number of blocks.

$$Size_{Adder} = Total_{Block}(1 - Total_{FAperblock}) \quad (7)$$

### III. RESULTS AND BENCHMARK

The proposed design with self-checking property is compared in terms of area overhead and fault coverage with the reported self-checking CSeA in [10] and [12]. Also, the proposed self-repairing HA is compared with self-repairing CoSA approach [19] and reduced precision TMR [17].

The transistor counts of each module for self-checking and self-repairing HA is presented in Table. 2. It should be noted that the Sum and Carry bypass modules are required for self-repairing design, therefore, they are not considered while comparing the area-overhead of self-checking design. The required number of logic gates and other modules along with the total transistor overhead is shown in Table. 3, where  $n$  is the size of adder,  $m$  is the size of RCA block and  $k$  is equal to the total number of CSeA blocks used in the design. The value of  $k$  varies with adder size and in this work, the value of  $k$  has been selected to be 1, 2, 3, 5 and 8 for 4-, 8-, 16-, 32- and 64-bit adder, respectively. In standard CSeA design without self-checking, the transistor count for full adder and MUX has been reduced to 12 and 4 respectively, because the *Eqt* and checker module are not required.

**TABLE 2. Transistor count for individual modules used in implementing self-checking and repairing hybrid adder.**

SFA Module	Transistor Count	Total
<i>SUM</i>	6	28
<i>Cout</i>	6	
<i>Equivalence Tester (Eq)</i>	8	
<i>Checker</i>	8	
Other Modules		Transistor Count
	<i>With Self-Checking</i>	<i>Without Self-Checking</i>
<i>OR/AND/XOR Gate</i>	6	4
<i>MUX</i>	6	4
<i>MOFC</i>	12	8
<i>Carry Bypass (CBP)</i>	6	4
<i>Sum Bypass (SBP)</i>	6	4

**TABLE 3. Required number of modules along with the total transistor count for hybrid adder without self-checking, with self-checking and -repairing.**

Modules	Standard CSeA	With Self-Checking	With Self-Repairing
<i>AND</i>	N/A	$n - m - k$	$n - m - k + 1$
<i>OR</i>	N/A	N/A	$n - 1$
<i>XOR</i>	N/A	$n - m - k$	$n - m - k + 1$
<i>MUX</i>	$n$	$n - m$	$n - m + 1$
<i>MOFC</i>	N/A	$k$	$k$
<i>Full adder / SFA</i>	$2n$	$n$	$n + 1$
<i>Carry-bypass</i>	N/A	N/A	$n + 1$
<i>Sum bypass</i>	N/A	N/A	$n - m + 1$
<i>IPS MUXs</i>	N/A	N/A	$2(n - 1)$
<i>OPS MUXs</i>	N/A	N/A	$n - 1$
<b>Total No. of Transistors</b>	$28n$	$46n - 18m$	$82n - 24m + 34$

#### A. COMPARISON WITH SELF-CHECKING CSeA

The area overhead of the proposed self-checking HA without recovery is compared with the previously reported self-checking CSeA design. It should be noted that a uniform complementary pass transistor logic design approach has been adopted while comparing the transistor counts, such that an inverter has been used after every stage of pass transistor. The implementation of sub modules with and without self-checking is shown in Fig. 4.

It can be observed from Table. 4 that the proposed design requires on average 50% more transistor count as compared to the standard CSeA design without self-checking. Whereas, the required number of transistors are reduced by 76.76% and 115% as compared to [12] and [10], respectively. It should be noted that the proposed approach also requires 68.68% less transistor count as compared to our previous proposed self-checking CSeA [11]. The increase in transistor count for different adder sizes has been shown in Fig. 5. It can be observed that our proposed approach shows least overhead as compared to the previous approaches.

In addition to the reduced area overhead, the proposed design possesses fault localization property and can detect multiple faults, with the condition that a single module should not have multiple faults, while [10] can only detect single fault at a time and [12] can only detect faults in odd number of bits without fault localization. In addition to the problem associated with odd number of erroneous bits, the approach

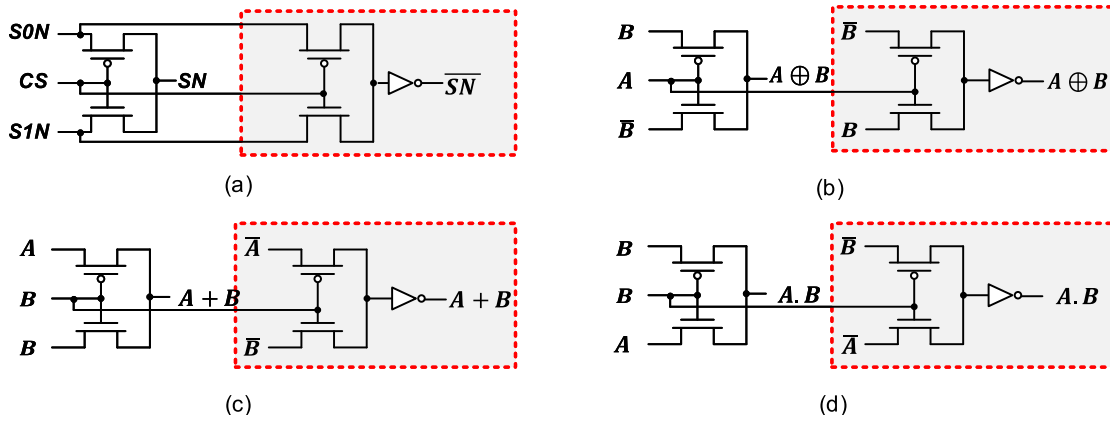


FIGURE 4. Pass-transistor based design modules of self-checking (a) MUX, (b) XOR gate, (c) OR gate and (d) AND gate.

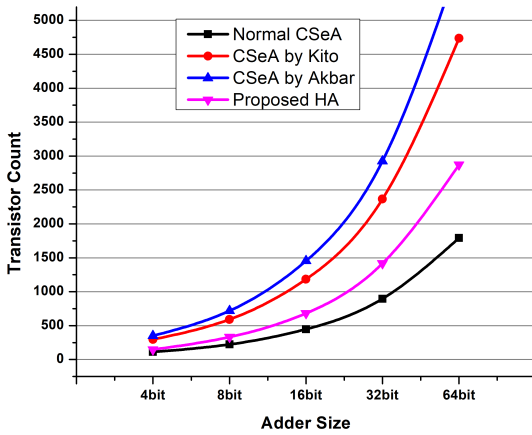


FIGURE 5. Transistor overhead comparison of self-checking CSeA proposed by Kito [12], Akbar [10] and our proposed HA with respect to standard CSeA.

in [12] is not totally self-checking because of the presence of logic sharing between the SUM and the propagated carry block. Any fault in the shared logic will easily get masked and cannot be detected with the parity prediction approach.

The power estimation is done using Cadence tool and it has been found that the traditional 32-bit CSeA design requires 4.51 mw power which increased to 6.90 mw for our proposed self-checking HA. Hence, 52.9% power consumption has been increased by using our proposed design. The delay for computing the final  $C_{out}$  using traditional CSeA design and the proposed self-checking CSeA has been shown in Eq. (8) and (9), respectively, where  $h$  is the number of full adders in

the final CSeA block. In can be observed that delay has been increased by a factor of only two logic gates.

$$t_{CSeA} = ht_{fa} + t_{MUX} \tag{8}$$

$$t_{SCH-HA} = ht_{fa} + t_{MUX} + t_{XOR} + t_{AND} \tag{9}$$

**B. COMPARISON WITH SELF-REPAIRING APPROACHES**

The proposed self-repairing adder with single spare module required an average of 186% area overhead as compared to traditional CSeA without self-checking as shown in Table. 4. The power consumption for 32-bit HA design has also been increased by 184.2% as compared to traditional CSeA. In terms of time overhead, the latency can be observed by Eq. (10). The overhead is mainly caused by the MUXs controlling the carry propagation chain.

$$t_{SR-HA} = ht_{fa} + (h + 2)t_{MUX} + t_{XOR} + t_{AND} \tag{10}$$

The proposed self-repairing HA design is compared with the previously reported self-repairing CoSA [19] and reduced precision redundancy adders (RPRa) [17]. It should be noted that both CoSA and RPRa approaches consider graceful degradation in which some portions of the circuitry have been considered for fault detection and recovery. In case of CoSA, the design cannot detect fault during the actual addition operation. Moreover, only single conditional selection cell (CSC) module can be tested at a time with a given test pattern. The self-checking property has not been considered for modules other than CSC like chain of MUXs, shift registers etc. Furthermore, the self-repairing process is also expensive

TABLE 4. Comparison of our proposed self-checking and -repairing HA with other approaches using CSeA.

Adder Size	Standard CSeA without Self-Checking			CSeA Kito et al. [12]		CSeA Akbar et al. [10]		Proposed Self-Checking HA		Proposed Self-Repairing HA	
	Transistor Required	Transistor Required	%Overhead	Transistor Required	%Overhead	Transistor Required	%Overhead	Transistor Required	%Overhead	Transistor Required	%Overhead
4-bit	112	296	164.28	350	212.5	148	32.14	314	180.35		
8-bit	224	592	164.28	718	220.53	332	48.21	642	186.60		
16-bit	448	1184	164.28	1454	224.55	682	52.23	1274	184.37		
32-bit	896	2368	164.28	2926	226.56	1418	58.25	2586	188.61		
64-bit	1792	4736	164.28	5870	227.56	2872	60.26	5186	189.39		

because the whole CSC module which is responsible for 2-bit addition, has to be replaced with the spare one. In addition to this problem the designed CoSA will not have fault diagnosis ability, if there is no further spare module available.

The RPRA [17] approach on the other hand can only correct the error in the MSB, while the Least Significant Bits (LSB) is fed directly to the output. Hence, fault in both LSB and the voter for MSB, is undetectable. Also, the fault propagated through LSB to MSB via  $C_{out}$  can not be detected.

In contrast to the previous approaches, the proposed design can perform run-time fault detection during actual addition process and also can detect multiple-faults at a time with the condition that each module should not have more than one fault. The fault recovery is dependent on the number of spare modules but the self-checking property of the design remain valid whether the recovery is possible or not.

#### IV. CONCLUSION

A self-checking and repairing HA design has been presented with reduced area overhead and increased fault coverage as compared to the previously presented design approaches. The HA design follows the architecture of single RCA based CSeA with the only difference of initial bits, which has been computed using RCA. A run-time self-repairing approach has been adopted by using hot-standby topology. The proposed design can be extended easily to any size by using fundamental block design presented in the paper.

The proposed design with self-checking has been compared with the previously reported self-checking CeSA in terms of area and fault coverage. It has been observed that the proposed self-checking HA approach with the delay overhead of only two logic gates, requires 50% more transistors as compared to the traditional CSeA without self-checking. Whereas, the required overhead is 76.76% and 115% less than the previously proposed self-checking CSeA approaches. Moreover, due to the distributed self-checking mechanism, the proposed approach can detect and localized multiple faults with the condition that a single module should have single fault at a time.

A hot-standby approach has been adopted for fault recovery. The area overhead has been increase to 186% as compared to standard CSeA approach. However, the probability of recovering multiple faults has been increase as compared to the previous self-repairing CSeA approaches. A 64-bit adder with 8 equally sized blocks can handle 3 consecutive faults with 96.1% probability with the condition that each block have single spare module. It should be noted that the self-checking property remained valid irrespective of the possible recovery which was not possible in previous approaches.

#### REFERENCES

[1] N. Mehdizadeh, M. Shokrolah-Shirazi, and S. G. Miremadi, "Analyzing fault effects in the 32-bit OpenRISC 1200 microprocessor," in *Proc. 3rd Int. Conf. Availability, Rel. Secur.*, 2008, pp. 648–652.

[2] A. Meixner, M. E. Bauer, and D. Sorin, "Argus: Low-cost, comprehensive error detection in simple cores," in *Proc. 40th Annu. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2007, pp. 210–222.

[3] H. Baig and J.-A. Lee, "An island style routing compatible fault-tolerant FPGA architecture with self-repairing capabilities," in *Proc. Field-Programm. Technol. (FPT)*, 2012, pp. 301–304.

[4] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 3, pp. 583–602, Jun. 2003.

[5] J. E. Smith and P. Lam, "A theory of totally self-checking system design," *IEEE Trans. Comput.*, vol. C-32, no. 9, pp. 831–844, Sep. 1983.

[6] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era," *IBM Syst. J.*, vol. 42, no. 1, pp. 5–18, 2003.

[7] T. Koal, D. Schiet, and H. T. Vierhaus, "A concept for logic self repair," in *Proc. 12th Euromicro Conf. Digit. Syst. Design, Architectures, Methods Tools*, Aug. 2009, pp. 621–624.

[8] V. Ocheretnij, D. Marienfeld, E. S. Sogomonyan, and M. Gossel, "Self-checking code-disjoint carry-select adder with low area overhead by use of add1-circuits," in *Proc. 10th IEEE Int. On-Line Test. Symp.*, Jul. 2004, pp. 31–36.

[9] D. P. Vasudevan, P. K. Lala, and J. P. Parkerson, "Self-checking carry-select adder design based on two-rail encoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 12, pp. 2696–2705, Dec. 2007.

[10] M. A. Akbar and J.-A. Lee, "Comments on 'self-checking carry-select adder design based on two-rail encoding,'" *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 7, pp. 2212–2214, Jul. 2014.

[11] M. A. Akbar and J.-A. Lee, "Self-checking carry select adder with fault localization," in *Proc. Euromicro Conf. Digit. Syst. Design (DSD)*, Sep. 2013, pp. 863–869.

[12] N. Kito and N. Takagi, "Concurrent error detectable carry select adder with easy testability," *IEEE Trans. Comput.*, vol. 68, no. 7, pp. 1105–1110, Jul. 2019.

[13] A. Majumdar, S. Nayyar, and J. S. Sengar, "Fault tolerant ALU system," in *Proc. Int. Conf. Comput. Sci. (ICCS)*, Sep. 2012, pp. 255–260.

[14] T. Koal, M. Ulbricht, and H. T. Vierhaus, "Virtual TMR schemes combining fault tolerance and self repair," in *Proc. Euromicro Conf. Digit. Syst. Design (DSD)*, Sep. 2013, pp. 235–242.

[15] J. H. Patel and L. Y. Fung, "Concurrent error detection in ALU's by recomputing with shifted operands," *IEEE Trans. Comput.*, vol. C-31, no. 7, pp. 589–595, Jul. 1982.

[16] M. Alderighi, S. D'Angelo, C. Metra, and G. R. Sechi, "Achieving fault-tolerance by shifted and rotated operands in TMR non-diverse ALUs," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2000, pp. 155–163.

[17] A. Ullah, P. Reviriego, S. Pontarelli, and J. A. Maestro, "Majority voting-based reduced precision redundancy adders," *IEEE Trans. Device Mater. Rel.*, vol. 18, no. 1, pp. 122–124, Mar. 2018.

[18] G. C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Fault localization, error correction, and graceful degradation in radix 2 signed digit-based adders," *IEEE Trans. Comput.*, vol. 55, no. 5, pp. 534–540, May 2006.

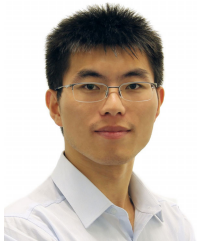
[19] A. Mukherjee and A. S. Dhar, "Real-time fault-tolerance with hot-standby topology for conditional sum adder," *Microelectron. Rel.*, vol. 55, nos. 3–4, pp. 704–712, Feb. 2015.

[20] J.-F. Lin, Y.-T. Hwang, M.-H. Sheu, and C.-C. Ho, "A novel high-speed and energy efficient 10-transistor full adder design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 5, pp. 1050–1059, May 2007.



**MUHAMMAD ALI AKBAR** received the B.Eng. degree (Hons.) in electronic engineering from NED University, Karachi, Pakistan, in 2011, and the M.Sc. degree in computer engineering from Chosun University, South Korea, in 2014. He is currently pursuing Ph.D. in computer engineering from Hamad Bin Khalifa University, Qatar.

He joined Qatar University, as a Research Assistant, in 2014, to work on different research project related to hardware design and machine learning applications. His main research interests include fault localization, self-reliable systems, sensor design, and artificial intelligence.



**BO WANG** (Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from Zhejiang University, Hangzhou, China, in 2010, and the M.Phil. and Ph.D. degrees in electronic and computer engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2012 and 2015, respectively.

He joined HKUST, as a Postdoctoral Researcher, in 2015, and led the HKUST-MIT consortium project on wireless sensing node design for smart green building applications. He was with the Massachusetts Institute of Technology, in 2016, on low power data converter for this project. He joined Hamad Bin Khalifa University, Qatar Foundation, as a Founding Faculty, in 2017, where he is currently an Assistant Professor with the Division of Information and Computing Technology, College of Science and Engineering. His research interests include energy-efficient analog/mixed-signal circuits, CMOS sensor/sensor interface, and heterogeneous integrated sensing systems for *in vitro/vivo* health monitoring. He was a recipient of the IEEE ASP-DAC Best Design Award in 2016.



**AMINE BERMAK** (Fellow, IEEE) received the master's and Ph.D. degrees in microelectronics and microsystems from Paul Sabatier University, Toulouse, France, in 1994 and 1998, respectively.

He joined the ECE Department, Hong Kong University of Science and Technology (HKUST), where he held all academic ranks and subsequently promoted to a full Professor. He was also the ECE Associate Head for Research and Postgraduate studies. He is currently a Full Professor and an

Associate Dean of Hamad Bin Khalifa University, Qatar Foundation, Qatar. He taught 25 different courses at the undergraduate and postgraduate levels. He is recognized as the world-leading author in the sensors area as well as the inventor of time-domain sensing. He has published over 350 articles in journals, book chapters, and conference proceedings and designed over 30 chips. He has graduated 25 Ph.D. students and 20 master's students.

Dr. Bermak was nominated for the 2013 Hong Kong UGC Best Teacher Award (for all HK Universities). He was a recipient of the 2011 University Michael G. Gale Medal for distinguished teaching (Highest University-wide Teaching Award) and the Engineering School Teaching Excellence Award from HKUST, in 2004 and 2009. He has received Six distinguished awards, including the Best University Design Contest Award at ASP Design Automation Conference (DAC) in Macau 2016, the Best Student Paper Award at IEEE International Symposium on Circuits and systems ISCAS 2010, the 2004 IEEE Chester Sall Award from IEEE Consumer Electronics Society, the IEEE Service Award from IEEE Computer Society, and the Best Paper Award at the 2005 International Workshop on System-On-Chip for Real-Time Applications. He has served on the Editorial Board of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS and *Sensors* journal. He is also serving on the editorial board of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS and the IEEE TRANSACTIONS ON ELECTRON DEVICES. He is also an Editor of *Nature Scientific Report*. He is an IEEE Distinguished Lecturer.

• • •