IJE TRANSACTIONS B: Applications Vol. 34, No. 02, (February 2021) 433-442

International Journal of Engineering

Journal Homepage: www.ije.ir

An Enhanced Self-checking Carry Select Adder Utilizing the Concept of Self-checking Full Adder

M. Valinataj

School of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran

P A P E R I N F O

Paper history:
Received 23 May 2019
Received in revised form 10 November 2020
Accepted 14 December 2020

Keywords:
Carry Select Adder
Self-checking Adder
Fault/Error Detection

A B S T R A C T

In this paper, an enhanced self-checking carry select adder (CSeA) architecture is introduced. However, we first show that the carry select adder design presented by akbar and lee does not have the self-checking property in all of its parts in spite of the stated claim. Then, we present a corrected design with the self-checking property that requires more overheads. In addition, we reveal some mistakes in reporting the transistor count of the proposed design in the literature in different sizes, and correct them which again leads to more transistor count and overhead. At the end, due to the fact that the performance of a CSeA depends on its grouping structure, the area overheads of different CSeAs including the corrected designs and the best of previous self-checking designs will be evaluated with respect to the same-size and different-size grouping structures. These evaluations show the comparison of different CSeAs, more appropriate compared to the previous evaluations.

***doi**: 10.5829/ije. 2021.34.02b.15*

1. INTRODUCTION

Current technologies used for manufacturing digital processing units are highly susceptible to environmental effects which may lead to improper operation of a processing unit because of a fault or error. Therefore, handling the erroneous situations in the form of fault/error detection or correction is highly required.

Among the processing units, the adders are very important due to their usage as one of the main functions. Thus, there exist various reported designs to address error detection in the adders such as those reported in literature [1-5] in addition to the multipliers [6,7] even in the reversible logic domain. The carry select adder (CSeA) is one of the fast adders utilized in the processing systems. Many researches have been conducted in recent years to reduce the cost and enhance the performance and reliability of the CSeAs [1, 5, 8-10]. The self-checking CSeA design is discussed in the researches of Akbar and Lee [1, 8]. However, Valinataj et al. proposed a method to achieve multiple fault/error detection is proposed for the CSeAs [5]. Moreover, some improvements in delay, energy or power and the silicon area were reported in literature [9,10].

The CSeA proposed by Akbar and Lee [1] is the best design with respect to the area overhead among its previous CSeA designs that attempt to achieve the self-checking property. However, this design is not entirely self-checking because a fault on some parts can result in an erroneous output without any detection. The concept of self-checking is related to the fault detection, and includes fault-secure and self-testing characteristics. A design with these characteristics is called totally self-checking [11]. A circuit is said fault-secure if it remains unaffected by a fault or it indicates a fault once as soon as it occurs [11]. However, a circuit is said self-testing if it is guaranteed that for each modelled fault there is at least one input vector, occurring during the normal operation of the circuit that detects it [12]. In summary, a single fault will be detected in a self-checking design if it shows a wrong result. The CSeA operation of Akbar and Lee [1] is correct at the absence of faults. However, if a fault occurs and produces a wrong result, the error detection probability is not 100%. Thus, in this paper, the self-checking property of the CSeA presented by Akbar

*Corresponding Author Email: *m.valinataj@nit.ac.ir* (M. Valinataj)

Please cite this article as: M. Valinataj, An Enhanced Self-Checking Carry Select Adder Utilizing the Concept of Self-Checking Full Adder, International Journal of Engineering, Transactions B: Applications Vol. 34, No. 02, (2021) 433-442

and Lee [1] is attained by adding some gates. In addition, the mistakes regarding transistor count will be removed that were caused by the incorrect assumptions of the required transistors for some gates and modules.

The speed of the CSeA is because of the parallel additions and also its grouping structure. Many of the existing CSeAs in literature [13-15] incorporate the square-root (SQRT) architecture to have a lower delay. This architecture utilizes a different-size grouping. However, the same-size groups can be utilized as well, to consume lower area or transistor count. The basic CSeA with either the SQRT grouping or the same-size grouping includes two ripple carry adders (RCA) in parallel in each group. However, the binary to excess-1 convertor (BEC) was used by Ramkumar and Kittur [13] instead of the second RCA in each group, which leads to lower area and power consumption but more delay. In this paper, the corrected self-checking CSeA based on the design in literature [1] is also investigated with respect to different grouping structures as the existing design lacks this evaluation.

2. INVESTIGATION OF the CSeA proposed by Akbar and Lee

Akbar and Lee [1] designed a single-group or single-stage of CSeA with different sizes with the aim of being low-cost and self-checking. This design can be used in real CSeAs with a specific grouping structure. The foundation of this CSeA is the BEC-based CSeA design proposed in literature [13] in which the RCA with the input carry equal to '1' is replaced by the BEC circuit for lowering the required area and power consumption. Therefore, this design is naturally more cost effective than the predecessor self-checking CSeA in literature [8,16] which are based on the basic CSeA design. In the design of Akbar and Lee [1], the self-checking property of the CSeA is obtained utilizing the self-checking modules. This CSeA is stated below.

2. 1. Self-checking Full Adder There exist various full adder (FA) designs in the literature such as [17, 18]. However, the concept of self-checking FA was initially introduced by Akbar and Lee [1], and later was used in other researches [3, 5]. This concept is based on the fact that when all three inputs of a FA, i.e. two input operands A and B and input carry Cin, are equal, then, the output sum (Sum) and output carry (Cout) will be equal, as well. On the other hand, these outputs will not be equal if all of the three inputs are not equal. This property can be used for designing a self-checking FA if a tester circuit called the equivalence tester (Eqt) and the error detection logic are appended to the FA, as shown in Figure 1. In the following, Equations (1) and (2) describe the operation of FA according to Figure 2a, and Equations (3) and (4) show the Boolean operation of equivalence tester and error detection logic shown in Figure 1, respectively.

$$Sum = A \oplus B \oplus C_{in} \quad (1)$$

$$C_{out} = A.B + C_{in}.(A + B) \quad (2)$$

$$Eqt = \overline{A.B.C_{in} + \bar{A}.\bar{B}.\overline{C_{in}}} \quad (3)$$

$$Ef = Sum \odot C_{out} \odot Eqt \quad (4)$$

In the equations above, the symbols $\oplus$ and $\odot$ depict XOR and XNOR operations, respectively. Based on Equation (3), the equivalence tester checks the inputs, and Eqt will become '0' if the equivalence of the inputs is verified; otherwise, it will become '1'. Then, the error function (Ef) is computed in Equation (4) using two XNOR operations. This equation sets Ef to '1' if a single fault has been detected based on the values of two outputs and Eqt. Otherwise, the FA is fault-free or more than a

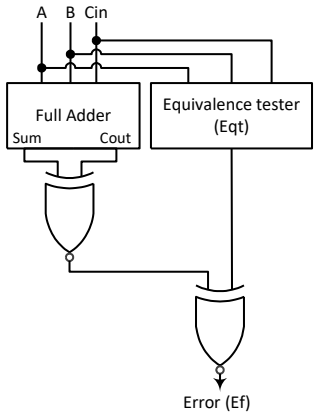

Figure 1. Proposed self-checking full adder in [1]

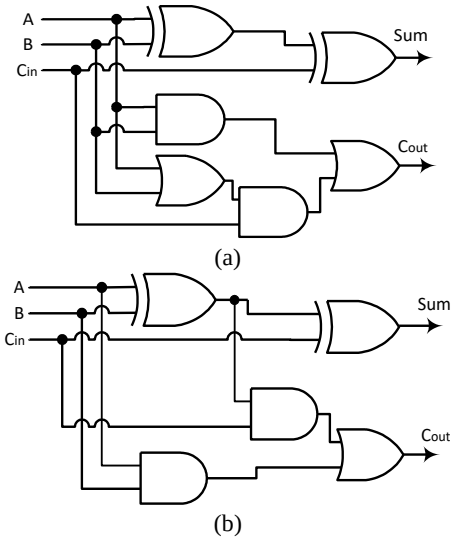

Figure 2. The full adder (a) without logic sharing used in [1], and (b) with logic sharing

single fault exist (which is not important because a self-checking design only guarantees single fault detection). It is worth mentioning that Equations (1) and (2) describe a FA without logic sharing between the outputs Sum and Cout as shown in Figure 2a. The usage of a FA without logic sharing is required to attain the self-checking property. In fact, a common logic between the outputs prevents to detect a single fault occurring in the common logic since it affects both outputs Sum and Cout, and destroys the self-checking property. The FA shown in Figure 2b uses a shared logic between the outputs and can be used for the low-cost adders.

2. 2. Entirely Self-checking Problem The two-bit single-group CSeA proposed by Akbar and Lee [1] is depicted in Figure 3. This design can be extended to larger groups by replicating the middle part (the logic for Sum bit 1) as many as required. In this CSeA, a NOT gate, some AND and XOR gates are used based on the BEC circuit which is utilized instead of the RCA with the input carry equal to '1'. The existing FAs constitute the RCA with the input carry equal to '0', and 2-to-1 multiplexers (MUX) operate in parallel to produce sum bits and the final C_{out} of the group based on the value of the input carry (C_{in}).

The CSeA shown in Figure 3 uses the self-checking FAs and thus any single fault in each self-checking FA will be detected by an *Ef* signal if it produces a wrong result. In [1] for making the entire CSeA a self-checking design, the pass-transistor-based XOR-XNOR gate from the literature [19] with the self-checking property is utilized instead of the XOR gates shown in Figure 3. This self-checking XOR-XNOR gate is shown in Figure 4. As shown in this figure, two inputs *a* and *b* with their complements enter the gate and the results of XOR and XNOR operations are produced. If a fault occurs inside this gate, the outputs will be the same instead of being the complement of each other and thus the fault can be detected. In addition, in [1] instead of the ordinary 2-to-1 MUX, the self-checking 2-to-1 MUX proposed in literature [16] is utilized. As shown by Vasudevan at al. [16], this 2-to-1 MUX produces two one-bit outputs that are complement of each other. Similar to the self-checking XOR-XNOR gate, if a fault occurs inside this MUX, the outputs will not be the complement of each other and the fault can be detected. It should be noted that the second outputs of self-checking XOR gates and multiplexers are not shown in Figure 3.

The main problem of the design shown in Figure 3 is that the NOT and AND gates are not self-checking and thus destroy the overall self-checking property. The output of these gates enters a self-checking MUX or XOR gate. However, due to the fact that these self-checking modules can help to detect only the internal faults, their outputs cannot assist to show an erroneous input. In fact, there is not any control on the inputs of these modules. For example, if a fault on the AND gate shown in Figure 3 causes its output to toggle, the output of the XOR gate in the MOFC part will definitely change, and this error can propagate to the output carry of the group. A similar case exists for the NOT gate in the first bit position of the adder that can make Sum bit 0 erroneously.

2. 3. Transistor Count Problem There exist some mistakes in the work of Akbar and Lee [1] regarding transistor count of some gates and modules based on the CMOS implementation as follows:

The first mistake is that the authors used the transistor count of FA presented by Vasudevan et al. [16] as the number of required transistors for their FA. However, the transistor implementation of the FA by Vasudevan et al. shows that it is a type of FA with logic sharing, and thus cannot be used inside a self-checking FA. Therefore, according to Figure 2a that shows the FA without logic sharing requires an extra OR gate compared to Figure 2b, the real transistor count for the FA part of the self-checking FA presented Akbar and Lee [1] will be more.

To obtain the transistor count of a FA shown in Figure 2a, different approaches can be used. The gate level implementation of carry generation logic requires 18 transistors, after adding 20 transistors for the two CMOS XNOR gates based on [16], the summation of 38 transistors is achieved. It should be noted that this is

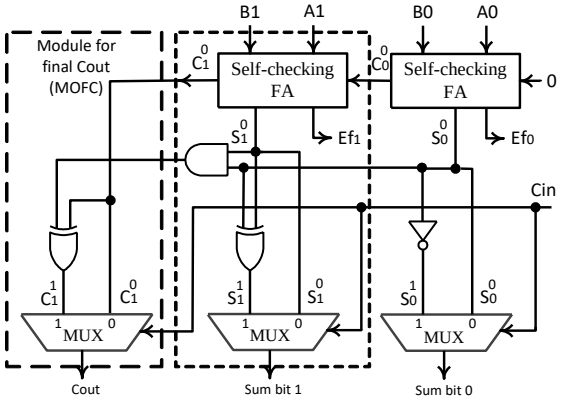

Figure 3. The 2-bit CSeA proposed in [1] as the self-checking design

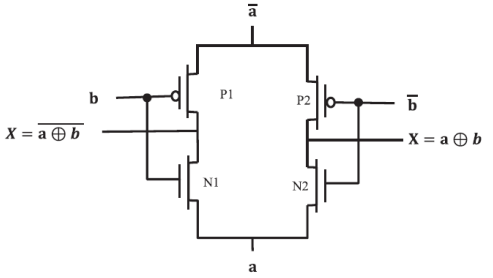

Figure 4. Self-checking XOR-XNOR gate proposed in [19]

based on the fact that the two XOR gates of Figure 2a can be replaced by two XNOR gates without altering the Sum. In another calculation, the FA without logic sharing has an extra OR gate compared to the FA with logic sharing. Thus, it requires 28 transistors based on [16] plus 6 transistors for the OR gate which leads to 34 transistors. Besides, if we use the 6-transistor CMOS XOR gate (such as the one discussed by Fathi et al. [20]) that does not lead to any voltage loss in the output, the minimum number of transistors require for the FA without logic sharing will be 18+2×6=30 in which 18 is for the carry generation logic and the other term is for two XOR gates. Therefore, the FA without logic sharing will have at least two more transistors than those stated Akbar and Lee [1].

The second mistake is that the mentioned number of transistors for the equivalence tester module is 12. However, according to Equation (3), this module cannot be implemented with less than 18 transistors because it includes six PMOS transistors, six NMOS transistors and three NOT gates according to the basic CMOS implementation depicted in Figure 5.

The third mistake is that the authors assumed four transistors for each self-checking XOR which is based on the self-checking XOR-XNOR gate proposed in literature [19]. However, Figure 4 shows that this gate requires eight transistors due to the fact that the inverted inputs should also be produced before entering the gate and these inverted inputs do not exist beforehand in the circuit.

The fourth mistake, as a minor mistake, is that the authors did not account the transistors of the NOT gate in the first bit position. It is obtained based on the transistor count calculation presented in Table 4 from the literature [1].

Table 1 shows the reported number of transistors in [1] and the corrected transistor counts for the related gates and modules. In this table, transistor count of the self-checking FA is computed based on its components

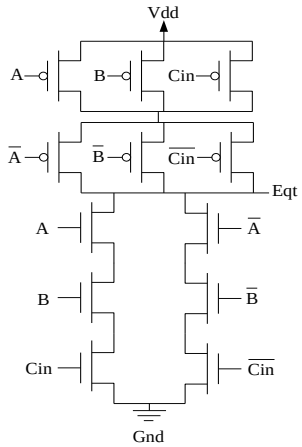

Figure 5. Transistor-level CMOS implementation of Equation (3) as *Eqt* signal

TABLE 1. Transistor count correction of different modules in CSeA

Module	No. of transistors reported in [1]	Corrected No. of transistors
FA	28	30
Equivalence tester (Eqt)	12	18
Self-checking FA	48	56
Self-checking XOR from [19]	4	8
Module for final C_{out} (MOFC)	16	20

including a FA, the equivalence tester and two XNOR gates for the error detection logic in which a four-transistor implementation is assumed for the XNOR gate. In addition, MOFC includes an XOR and a 2-to-1 MUX based on Figure 3, and will have 20 transistors since the self-checking MUX presented by Vasudevan et al. [16] includes 12 transistors and the self-checking XOR requires eight transistors.

2. 4. The Need for Two-pair Two-rail Checkers

As stated by Akbar and Lee [1], the proposed CSeA could remove a tree of two-pair two-rail checkers needed to compare the outputs of each two FAs of the *i*th location in the two RCAs of the self-checking CSeA proposed by Vasudevan et al. [16]. However, two complement outputs of each self-checking MUX in [1] were left intact similar to that of the design in [16]. In addition, the complement outputs of the self-checking XOR-XNOR gates used by Akbar and Lee [1] were left intact, as well. Despite the fact that each of these self-checking components produces two identical outputs when a fault internally occurs, there should be another circuit that produces a general error signal by investigating all existing output pairs. The best way for this integration is the use of two-pair two-rail checkers similar to the ones used Vasudevan et al. [16]. Therefore, for two output pairs of two multiplexers a two-pair two-rail checker is required that is implemented by eight transistors according to the literature [21]. Similarly, two output pairs of two self-checking XOR-XNOR gates require a two-pair two-rail checker. Two-pair two-rail checkers can be arranged in the tree structure, a tree for self-checking multiplexers and a tree for self-checking XOR-XNOR gates. An *m*-bit self-checking CSeA requires (*m*–1) two-pair two-rail checkers for each tree. Thus, the CSeA proposed by Akbar and Lee [1] requires around 15% to 20% more transistors for handling all of the complement outputs.

3. THE PROPOSED SELF-CHECKING CSeA

Based on the entirely self-checking problem described earlier, the corrected self-checking CSeA with the

general size of *n* bits is proposed according to Figure 6. Based on this figure, a small redundant logic is utilized for each non-self-checking gate (NOT or AND) in each bit position. It should be noted that the proposed corrected self-checking CSeA is an *n*-bit single-stage or single-group CSeA which can be used to construct larger CSeAs with the same-size or different-size groups.

In a self-checking design, a single fault is detected if it can change the output. Thus, to make entirely self-checking the first bit position of the CSeA depicted in Figure 6, a fault on the NOT gate must be detected as well as a fault inside the self-checking FA or MUX. This can simply be performed using an ordinary XNOR gate comparing S_0^0 and S_0^1 since these signals should be the complement of each other. To make entirely self-checking the bit positions from the second to the last, the same logic is used in these bit positions to cover the probable faultiness of the AND gates. In fact, in each bit position the complemented output of the AND gate is generated by a NAND gate, and then an ordinary XNOR gate compares these outputs. If these two outputs are the same, the output of the XNOR gate will be set to '1' indicating a fault inside the AND or NAND gate behind it. It is worth mentioning that the XNOR gates used for the comparisons to have the self-checking property is not required; because a fault on an XNOR gate will set its output to '1' remembering that a single fault is guaranteed in a self-checking design. The new error signals E_0 to $E_{(n-1)}$ shown in Figure 6 can enter an OR gate together with the *Ef* signals to produce the overall error detection signal.

The following theorem can be used to prove the self-checking property of the corrected CSeA shown in Figure 6:

Theorem 1. A logic fault occurred inside the CSeA shown in Figure 6 will be detected if it makes an internal or external error. Thus, this CSeA has the self-checking property.

Proof. The internal components of the CSeA shown in Figure 6 can be divided in three parts, the self-checking FAs, multiplexers, and the simple gates. As shown in Section 2.1, a single fault inside a self-checking FA will be detected if it affects one of its components including the internal FA, the equivalence tester logic or one of the XNOR gates producing the error signal *Ef*. In addition, all the multiplexers and XOR gates shown in Figure 6 are the self-checking designs according to Section 2.2, and each of them generates two complement outputs. Therefore, these complement outputs can be checked to detect probable internal errors. The remaining gates are a NOT and an XNOR in the first bit position, and three gates including AND, NAND and XNOR in the other bit positions. In the first bit position, an internal error in the NOT or XNOR will set E_0 to '1'. In the other bit positions (*i* from 1 to *n*–1), an internal error in the AND, NAND or XNOR will definitely set E_i to '1' since the outputs of AND and NAND gates must be the complement of each other. It is worth mentioning that most of internal errors lead to a wrong result (erroneous Sum or C_{out}). However, some internal errors such as the output of a faulty NAND gate do not lead to a wrong result although they activate an error signal.

Based on Theorem 1, the detection of a single logic fault is guaranteed. However, many multiple-fault situations can be detected in this CSeA because of the existence of many internal error indicators.

It should be noted that the delay of the proposed single-stage self-checking CSeA shown in Figure 6 is the same as that of the CSeA based on Figure 3 with the same size. In fact, the new gates and also all the error indicating signals are outside of the critical path.

4. EXPERIMENTAL RESULTS AND EVALUATIONS

In this section, at first, the effect of corrected transistor count on the CSeA design of Akbar and Lee [1] is evaluated along with the area overhead of the proposed corrected self-checking CSeA. Then, the effect of different grouping structures is evaluated when multi-group CSeAs with different sizes are constructed based on the corrected self-checking CSeA, the CSeA design of

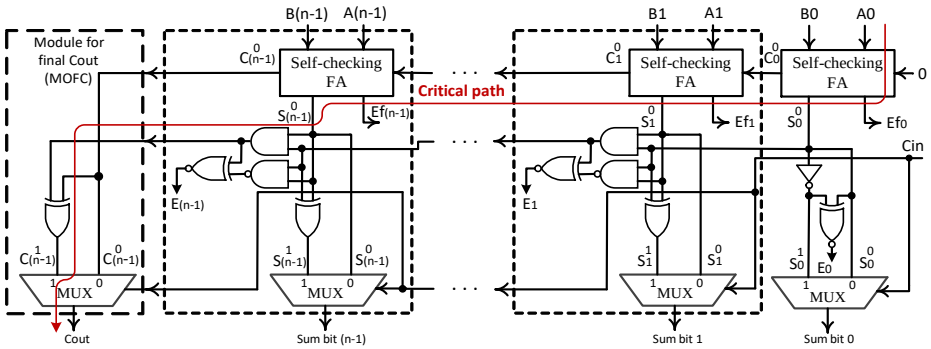

Figure 6. The proposed *n*-bit self-checking CSeA

Akbar and Lee [1] after transistor count correction, and the best of previous self-checking designs. As mentioned before, this investigation has not been performed by Akbar and Lee [1] and only single-group CSeAs have been evaluated.

4. 1. Area Estimation Transistor count has been used for the area estimation of the similar designs in literature [1,8,16]. Thus, in this paper this parameter is used for the comparisons. In addition to the corrected number of transistors presented in Table 1, we should know the transistor count of the remaining gates or components according to Table 2. In this table, similar to the work of Akbar and Lee [1], we assume four transistors for XNOR based on one of its CMOS implementations.

Based on Figure 3 and Tables 1 and 2, the first bit position of the CSeA in the literature [1] with corrected transistor count requires 70 transistors as it includes a self-checking FA, a self-checking MUX, and a NOT gate. But the other bit positions require 82 transistors. Moreover, based on Figure 6, the proposed corrected self-checking CSeA requires four and eight more transistors compared to that of the CSeA in [1], for the first and the remaining bit positions, respectively, because of the extra gates. Thus, the following equations can be used to compute the transistor count of each *n*-bit group in the CSeAs:

$$\begin{aligned}&\text{T.C. of n-bit group after transistor count correction}\\&= \text{T.C. of 1st bit position} + (n-1) \times \text{T.C. of other bit positions} + \text{T.C. of MOFC}\\&= 70+82(n-1)+20 = 82n+8\end{aligned} \quad (5)$$

$$\begin{aligned}&\text{T.C. of n-bit group in the corrected self-checking CSeA}\\&= 74+90(n-1)+20 = 90n+4\end{aligned} \quad (6)$$

where T.C. stands for transistor count.

Table 3 depicts the number of transistors required for the implementation of CSeAs in addition to their overheads. The transistor overheads are obtained compared to the basic non-self-checking CSeA. It should be noted that all CSeAs in this table are single-stage or single-group which can be used to construct larger adders. In addition, for simplicity, extra two-pair two-rail checkers have not been accounted in the corrected results. As stated in Section 2.4, these checkers lead to more transistor count. For more illustration, Figure 7 shows the area overheads in percent based on transistor count for the initial CSeA in [1] and its corrected versions compared to the basic non-self-checking CSeA. As perceived from this figure, the real overheads are much more than the ones reported by Akbar and Lee [1].

4. 2. Effect of Grouping Structures If a CSeA is utilized as the adder part of a processing core, it is used in a multi-group structure. Thus, as none of the grouping structures has been investigated by Akbar and Lee [1], in this section, the area overheads are evaluated with respect to different grouping structures.

4. 2. 1. SQRT Grouping This structure utilizes a different-size grouping in such a way that it leads to the minimum delay. In fact, in a SQRT grouping, the group sizes are determined in such a way that the delay required for the preparation of two sum bits in a group to be almost the same as the delay for the input carry arrived from the previous group. A basic 16-bit CSeA with the SQRT grouping is shown in Figure 8. Based on this figure, the

TABLE 2. Transistor count of the other utilized gates or modules in the self-checking CSeA

Module	No. of transistors
NOT	2
2- input NAND	4
2- input AND	6
XNOR	4
Self-checking MUX from [16]	12

TABLE 3. Comparison of single-group CSeAs based on transistor count

Adder size (bit)	Non-self-checking CSeA	CSeA in [1] with mistakes		CSeA in [1] after transistor count correction		Proposed corrected self-checking CSeA (Figure 6)	
	Transistor count [16]	Transistor count [1]	Transistor overhead [1]	Transistor count	Transistor overhead	Transistor count	Transistor overhead
4	284	286	2	336	52	364	80
6	420	426	6	500	80	544	124
8	556	566	10	664	108	724	168
16	1100	1126	26	1320	220	1444	344
32	2188	2246	58	2632	444	2884	696
64	4364	4486	122	5256	892	5764	1400

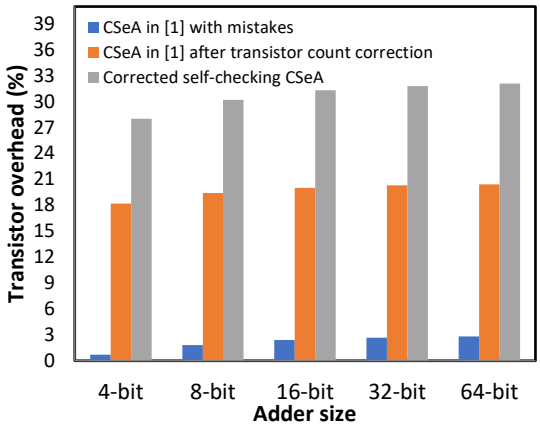

Figure 7. Transistor count overheads of the initial and corrected single-group CSeAs

16-bit SQRT CSeA consists of a single 2-bit RCA in the least significant bits and the groups with the sizes of 2, 3, 4 and 5 bits, respectively. The best SQRT grouping for the 8-bit CSeA includes a single 2-bit RCA in the least significant bits, and two groups with the sizes of 2 and 4 bits, respectively. For the 32-bit SQRT CSeA after the first 2-bit RCA, the groups with the sizes of 2, 3, 4, 6, 7 and 8 bits lead to the minimum delay, and for the 64-bit SQRT CSeA after the first 2-bit RCA, the groups with the sizes of 2, 3, 4, 6, 7, 8, 9, 11 and 12 bits are the best sizes [15,22].

Here, to achieve more real results compared to Section 4.1, a comparison is performed for the SQRT CSeAs between the basic or non-self-checking CSeA, the CSeA in [1] after transistor count correction, the corrected self-checking CSeA in this paper based on [1], and the self-checking CSeA in [8] as the best of predecessor self-checking designs. It should be noted that the design described in [8] is the corrected version of the CSeA design proposed Vasudevan et al. [16]. Thus, the design in [8] has been used for comparisons.

Figure 8 also shows the critical path for delay. The SQRT grouping reduces the delay compared to the same-size grouping by trying to make identical the delay of the multiplexers chain on the shown critical path and the delay of the RCAs in the last group. However, after synthesising, because of the increasing fan-out on the output carries of the groups, the multiplexers chain will have a higher delay. As a result, the corrected self-checking CSeA in this paper has the same delay compared to the self-checking CSeA in [1]. However, both have lower delay compared to the CSeA in [8] since it uses a tree of two-pair two-rail checkers in each group that finally increases the total delay. The exact delay improvement over [8] can be estimated using the method introduced by Fathi et al. [20].

As stated before, Equations (5) and (6) show the transistor count of the *n*-bit groups for the CSeA in [1] after transistor count correction and the corrected self-checking CSeA, respectively. However, these equations can be used to compute the transistor count of multi-group CSeAs with either different-size or same-size grouping because the total transistor count can be obtained by adding the transistor count of all the groups. To compute the transistor count of an *n*-bit group in the non-self-checking CSeA and the self-checking design in [8], some equations similar to Equations (5) and (6) can be derived which lead to $70n+14$ and $78n-4$, respectively. However, it should be noted that the self-checking CSeA in [8] requires some extra transistors equal to $8\times(m-1)$ that should be added to the total transistor count since an *m*-bit multi-group CSeA based on [8] requires $(m-1)$ number of two-pair two-rail checkers, as well. However, for a fair comparison, transistors of extra two-pair two-rail checkers have not been accounted.

Based on the literature [16], each two-pair two-rail checker can be implemented by eight transistors. In addition, each FA of the non-self-checking CSeA and the self-checking design in [8] is implemented by 28 transistors according to Vasudevan et al. [16].

Using the equations obtained for *n*-bit groups of four different CSeAs, the number of required transistors for each adder size is computed. The obtained results are shown in Table 4. For example, in the 8-bit corrected self-checking CSeA, a 2-bit RCA, a 2-bit group and a 4-bit group require 112, 184, and 364 transistors, respectively, which lead to the total transistor count of

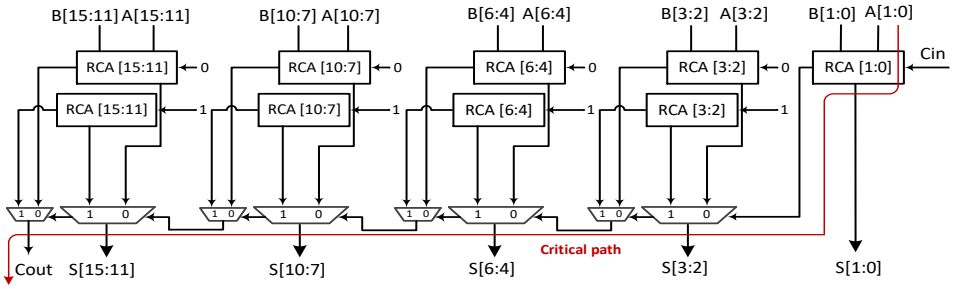

Figure 8. Basic non-self-checking 16-bit CSeA with the SQRT grouping

TABLE 4. Transistor count of different CSeAs utilizing the SQRT grouping

Adder size (bit)	Non-self-checking CSeA	Self-checking design [8]	CSeA in [1] after transistor count correction	Corrected self-checking CSeA
8	504	668	620	660
16	1092	1348	1292	1388
32	2240	2716	2620	2836
64	4522	5456	5268	5728

660. It is worth mentioning, in contrary to the other adders, the first 2-bit RCA in the self-checking design of [8] should be a 2-bit group in all adder sizes shown in Table 4 because of its own architecture to achieve the self-checking property.

For more illustration, Figure 9 depicts transistor overheads of three different SQRT CSeAs compared to the non-self-checking design based on Table 4. According to Figure 9, different from Figure 7, in all adder types the overhead decreases when the adder size increases. In addition, the corrected self-checking CSeA with the SQRT grouping requires lower overheads compared to its single-group counterpart based on Figure 7.

4. 2. 2. Same-size Grouping Another grouping structure utilized in the CSeAs is the same-size grouping in which all the groups and the single RCA in the least significant bits of the adder have the same size. Generally, this structure leads to lower transistor count compared to the SQRT grouping. However, it requires more delay, as well. The best size for the same-size groups is $\sqrt{m}$ for each m-bit CSeA.

Here to achieve the results for the same-size grouping, only the 4-bit and 8-bit sizes are investigated for simplicity, which are the best sizes for 16-bit and 64-bit CSeAs, respectively. Using the equations stated before for different CSeAs, the transistor counts of the CSeAs with the same-size grouping are obtained and shown in Tables 5 and 6 for 4-bit and 8-bit group sizes, respectively. In addition, Figures 10 and 11 depict transistor overheads of three different CSeAs compared to the non-self-checking design based on Tables 5 and 6, respectively. Similar to Figure 9, Figures 10 and 11 show that in all adder types the transistor overhead decreases when the adder size increases. Moreover, an important result is that the corrected self-checking CSeA which is based on the design in [1] is not always better than that

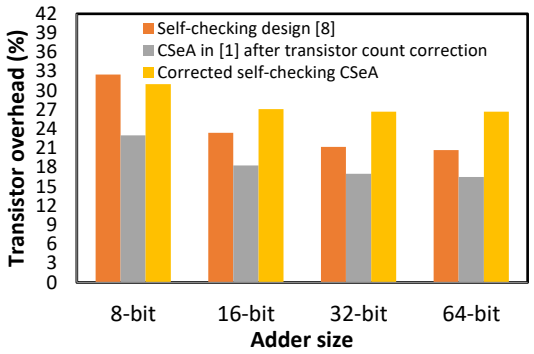

Figure 9. Transistor count overheads of the SQRT CSeAs

TABLE 5. Transistor count of different CSeAs utilizing 4-bit groups.

Adder size (bit)	Non-self-checking CSeA	Self-checking design [8]	CSeA in [1] after transistor count correction	Corrected self-checking CSeA
8	406	672	560	588
16	994	1352	1232	1316
32	2170	2712	2576	2772
64	4522	5432	5264	5684

TABLE 6. Transistor count of different CSeAs utilizing 8-bit groups

Adder size (bit)	Non-self-checking CSeA	Self-checking design [8]	CSeA in [1] after transistor count correction	Corrected self-checking CSeA
16	798	1360	1112	1172
32	1946	2728	2440	2620
64	4242	5464	5096	5516

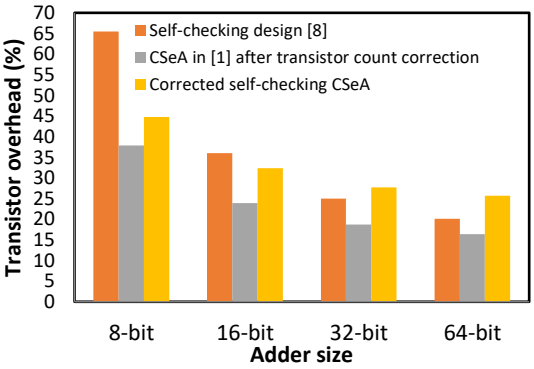

Figure 10. Transistor count overheads of the CSeAs with 4-bit groups

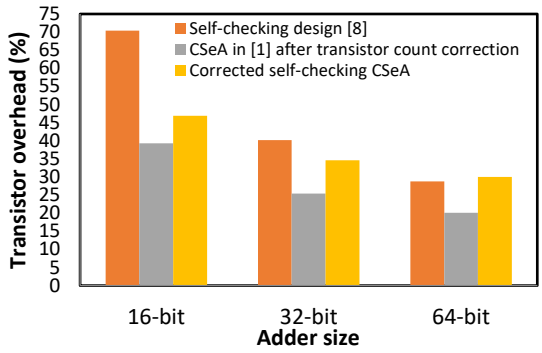

Figure 11. Transistor count overheads of the CSeAs with 8-bit groups

of in [8] with respect to transistor count. In fact, according to Figures 9 to 11 the corrected design is better in smaller sizes, and in larger CSeAs the self-checking design in [8] will require fewer transistors.

7. CONCLUSIONS

In this paper, we showed that the CSeA proposed by Akbar and Lee [1] is not a complete self-checking design and also has some mistakes in counting the number of required transistors. Thus, the transistor counts were re-evaluated, and a new design was proposed to achieve a self-checking CSeA. Both transistor count re-evaluation and design modification show that this CSeA incurs more overheads. Moreover, to obtain more realistic results and overheads, different grouping structures including the same-size and different-size groups were applied on the corrected CSeAs and the best of previous self-checking designs proposed in [8]. The evaluations revealed that the CSeA proposed in [1] after applying all needed corrections requires lower transistor count only in some adder sizes compared to the best of previous self-checking CSeAs.

8. REFERENCES

1. Akbar, M. A. and Lee, J., "Self-repairing adder using fault localization", ***Microelectronics Reliability***, Vol. 54, No. 6-7, (2014), 1443-1451. DOI: 10.1016/j.microrel.2014.02.033.
2. Mukherjee, A. and Dhar, A. S., "Real-time fault-tolerance with hot-standby topology for conditional sum adder", ***Microelectronics Reliability***, Vol. 55, No. 3-4, (2015), 704-712. DOI: 10.1016/j.microrel.2014.12.011.
3. Moradian, H., Lee, J.-A. and Hashmi, A., "Self-repairing radix-2 signed-digit adder with multiple error detection, correction, and fault localization", ***Microelectronics Reliability***, Vol. 63, (2016), 256–266. DOI: 10.1016/j.microrel.2016.06.010.
4. Moradian, H., Lee, J.-A. and Yu, J., "Efficient low-cost fault-localization and self-repairing radix-2 signed-digit adders applying the self-dual concept", ***Journal of Signal Processing Systems***, Vol. 88, No. 3, (2017), 297-309. DOI: 10.1007/s11265-016-1162-1.
5. Valinataj, M., Mohammadnezhad, A. and Nurmi, J., "A low-cost high-speed self-checking carry select adder with multiple-fault detection", ***Microelectronics Journal***, Vol. 81, (2018), 16-27. DOI: 10.1016/j.mejo.2018.08.014.
6. Valinataj, M., "Novel parity-preserving reversible logic array multipliers", ***The Journal of Supercomputing***, Vol. 73, No. 11, (2017), 4843-4867. DOI: 10.1007/s11227-017-2057-z.
7. Eslami-Chalandar, F., Valinataj, M. and Jazayeri, H., "Reversible logic multipliers: novel low-cost parity-preserving designs", ***International Journal of Engineering, Transactions C: Aspects***, Vol. 32, No. 3, (2019), 381-392. DOI: 10.5829/ije.2019.32.03c.05.
8. Akbar, M. A. and Lee, J., "Comments on "Self-Checking Carry-Select Adder Design Based on Two-Rail Encoding"", ***IEEE Transactions on Circuits and Systems-I: Regular Papers***, Vol. 61, No. 7, (2014), 2212-2214. DOI: 10.1109/TCSI.2013.2295930.
9. Nam, M., Choi, Y. and Cho, K., "High-speed and energy efficient carry select adder (CSLA) dominated by carry generation logic", ***Microelectronics Journal***, Vol. 79, (2018), 70-78. DOI: 10.1016/j.mejo.2018.07.001.
10. Mohammadnezhad, A. and Valinataj, M., "Enhancing speed, area and power consumption of carry select adders using a new grouping structure", ***Iranian Journal of Electrical and Computer Engineering***, Vol. 16, No. 4-B, (2019), 310-318.
11. Smith, J. E. and Lam, P. "A theory of totally self-checking system design", ***IEEE Trans. on Computers***, Vol. 32, No. 9, (1983), 831–844. DOI: 10.1109/TC.1983.1676332.
12. Carter, W. C. and Schneider, P. R., "Design of dynamically checked computers", 4th Congress IFIP, Vol. 2, (1968), 878-883.
13. Ramkumar, B. and Kittur, H. M., "Low-power and area-efficient carry-select adder", ***IEEE Transactions on Very Large Scale Integration Systems***, Vol. 20, No. 2, (2012), 371-375. DOI: 10.1109/TVLSI.2010.2101621.
14. Manju, S. and Sornagopal, V., "An efficient SQRT architecture of carry select adder design by common Boolean logic", Int. Conf. Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), (2013), 1-5. DOI: 10.1109/ICEVENT.2013.6496590.
15. Mohanty, B. K. and Patel, S. K., "Area–delay–power efficient carry-select adder", ***IEEE Trans. on Circuits and Systems-II: Express Briefs***, Vol. 61, No. 6, (2014), 418-422. DOI: 10.1109/TCSII.2014.2319695.
16. Vasudevan, D. P., Lala, P. K. and Parkerson, J. P., "Self-checking carry-select adder design based on two-rail encoding", ***IEEE Trans. on Circuits and Systems-I: Regular Papers***, Vol. 54, No. 12, (2007), 2696-2705. DOI: 10.1109/TCSI.2007.910537.
17. Mehrabani, Y. S. and Shafiabadi, M. H., "Symmetrical, low-power, and high-speed 1-bit full adder cells using 32nm carbon nanotube field-effect transistors technology", ***International Journal of Engineering, Transactions A: Basics***, Vol. 28, No. 10, (2015), 1447-1454. DOI: 10.5829/idosi.ije.2015.28.10a.07.
18. Ghorbani, S., Ghorbani, S. and Kashyzadeh, K. R., "Taguchi approach and response surface analysis for design of a high-performance single-walled carbon nanotube bundle interconnects in a full adder", ***International Journal of Engineering, Transactions B: Applications***, Vol. 33, No. 8, (2020), 1598-1607. DOI: 10.5829/ije.2020.33.08b.18.
19. Belgacem, H., Chiraz, K., and Rached, T., "Pass transistor based self-checking full adder", ***International Journal of Computer Theory and Engineering***, Vol. 3, No. 5, (2011), 608-616. DOI: 10.7763/IJCTE.2011.V3.379.
20. Fathi, A, Mashoufi, B., and Azizian, S., "Very fast, high-performance 5-2 and 7-2 compressors in CMOS process for rapid parallel accumulations", ***IEEE Transactions on Very Large***

Scale Integration Systems, Vol. 28, No. 6, (2020), 1403-1412. DOI: 10.1109/TVLSI.2020.2983458.

21. Lo, J. C., "Novel area-time efficient static cmos totally self-checking comparator", ***IEEE Journal of Solid-State Circuits***, Vol. 28, No. 2, (1993), 165-168. DOI: 10.1109/4.192049.

22. Kim, Y. and Kim, L.-S., "64-bit carry-select adder with reduced area", ***Electronics Letters***, Vol. 37, No. 10, (2001), 614-615. DOI: 10.1049/el:20010430.

Persian Abstract

چکیده

در این مقاله، یک جمع‌کننده انتخاب رقم نقلی بهبودیافته با ویژگی خودتست معرفی می‌گردد. با این حال، در ابتدا نشان می‌دهیم که جمع‌کننده انتخاب رقم نقلی ارائه شده در [۱] برخلاف ادعای مطرح شده در آن دارای ویژگی خودتست در همه بخش‌های طرح نیست. سپس، طرح اصلاح شده با ویژگی خودتست که نیازمند سربارهای بیشتری است ارائه می‌گردد. علاوه براین، چندین اشتباه در محاسبه تعداد ترانزیستور طرح پیشنهادی [۱] برای اندازه‌های مختلف جمع‌کننده را بیان می‌نماییم که اصلاح آن‌ها منجر به تعداد ترانزیستور و سربار بیشتر می‌گردد. در پایان، با توجه به این که کارایی یک جمع‌کننده انتخاب رقم نقلی وابسته به ساختار گروه‌بندی استفاده شده در آن است، سربارهای مساحت جمع‌کننده‌های مختلف شامل طرح‌های اصلاح شده و بهترین طرح خودتست قبلی با توجه به گروه‌بندی‌های با اندازه یکسان و اندازه متفاوت مورد ارزیابی قرار خواهد گرفت. این ارزیابی‌ها در مقایسه با ارزیابی‌های گذشته، مقایسه میان جمع‌کننده‌های انتخاب رقم نقلی متفاوت را مناسب‌تر نشان می‌دهند.