International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 8, Number 2 (2015), pp. 99-106
© International Research Publication House
http://www.irphouse.com

Detection Of Fault In Self Checking Carry Select Adder

S. Muthulakshmi @Subha

Asst. Prof, Department Of ECE,
Chendhuran College Of Engineering And Technology,
Pudukkottai, Tamilnadu. India.

Abstract

Carry select adders are one of the faster types of adders. Self checking carry select adder encodes the sum bits using two rail codes; the encoded sum bits are then checked by self checking checkers. The common problem in self checking adders is the fault propagation due to carry. such a fault can misguide the system to detect the particular fault module. This paper proposes a self checking carry select adder with fault localization for more than 2 input bits. By using this scheme, instead of replacing the whole system we can now replace the particular faulty modules.

Keywords- self checking adder, carry select adder, fault localization

I. INTRODUCTION

In digital system designs the adder has a wide variety of applications. carry select adder is one of the faster types of adders, and has smaller area overhead than all other types of adders except for the carry skip adder. Advanced microelectronic technologies have made the current digital system to become more vulnerable for faults. A system will be fault secure, if it remains unaffected by the fault or indicate the fault as soon as it occurs [1]. A system will be self-testing, if it produces a non-coded output in response to every generated fault. A system will be totally-self checking (TSC) if it is both fault secure and self-testing.

This paper is organized as follows Section II describes the fault propagation problem due to carry. Section III discusses the proposed design approach for self checking full adder. In Section IV discusses about efficient self-checking Carry Select Adder with inputs more than 2 bits. section V discusses self checking carry select adder with fault localization. Section VI concludes the paper.

II. FAULT PROPAGATION

The common design problem in various approaches for self-checking adders is the fault propagation due to carry. Such a fault can misguide the system from detecting the particular faulty region in a module because propagated error can represent the correct region of module to be a faulty one as well.

A duplex system[2] has been shown in Fig. 1, in which each module is having three full adders. Let us assume that an error occurs in the carry generation part of the first adder. The propagated error will indicate the second and third full adders to be the faulty modules, while the first adder will be indicated as a fault free module. Therefore, in order to achieve recovery we will be replacing the second and third modules and hence, the faulty module has not been recovered. Similarly, if we detect that a module having 5 full adders generate faulty output then by most of the current approaches we cannot detect that, which of the 5 full adders is faulty. Hence, if a fault is indicated in any region of a large module then we need to replace the whole module in order to accomplish the recovery process. This kind of replacement approach will waste resources because the proper functioning block has also been replaced.

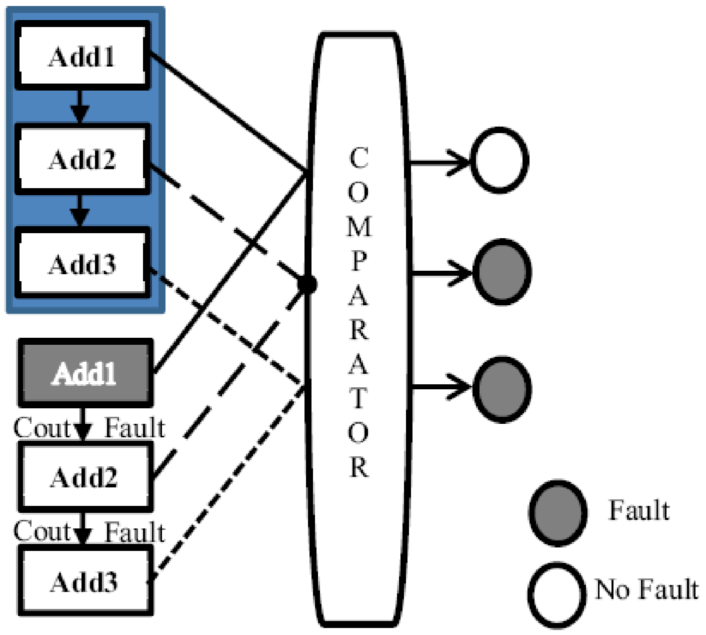


Figure1 Fault propagation through carry

III. PROPOSED SELF CHECKING FULL ADDER

From the truth table of full adder

- The Sum and Carry bit will be equal to each other when all the three inputs are equal.
- The Sum and Carry bit will be complemented when any of the three inputs is different.

A equivalent tester(E_{qt}) is introduced in Fig 2

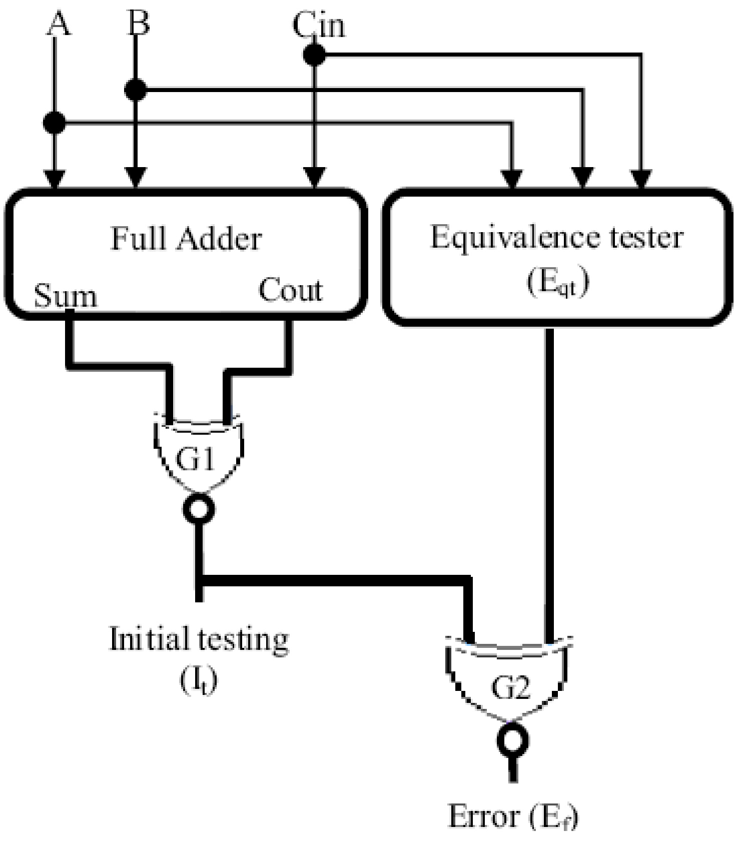


Figure2 Proposed self-checking full adder

The purpose of equivalence tester is to check the equivalence of all inputs. And the expression is given by

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

Cout=A. B+C_{in}. (A+B)

Equivalence tester (E_{qt})= $\overline{(\bar{A}\bar{B}\bar{C} + ABC)}$

Error(E_f)=sum $\odot$ cout $\odot$ E_{qt}

The final fault is computed by using two XNOR gates. The purpose of first XNOR gate (G1) is to check whether the Sum and Cout bit are equal or complemented.
We need a second XNOR gate (G2) because of the previously mentioned observation that Sum and Cout will always be complement to each other except when all inputs are equal. Thus, the output of G1 will indicate the equality or
difference of Sum and Cout and G2 will verify the output of G1 by comparing it with an equivalence tester and thus generate the final error indication. When the Eqt is zero the output of G1 and G2 should be logic 1 and 0, respectively. While, if the Eqt

indicates logic 1 then both the XNOR gates should generate logic-0 (i. e. It =0 and Ef =0) and in any other case the fault will be indicated.
Figure 3 shows the RTL implementation of proposed self checking full adder

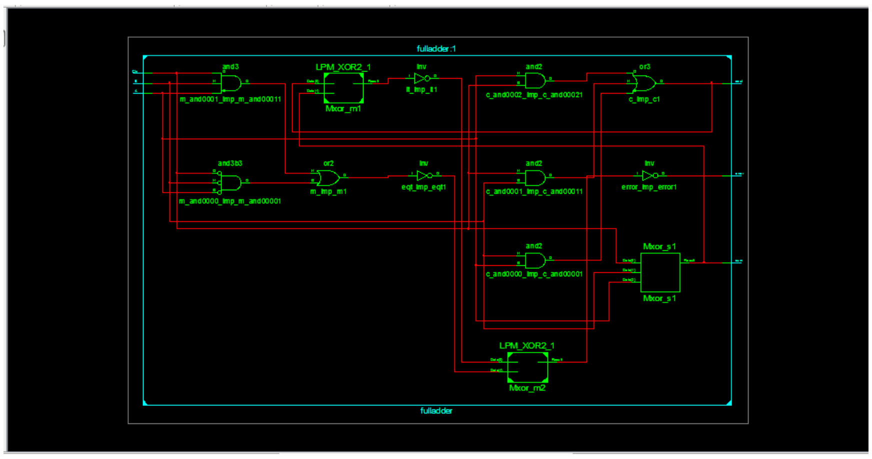
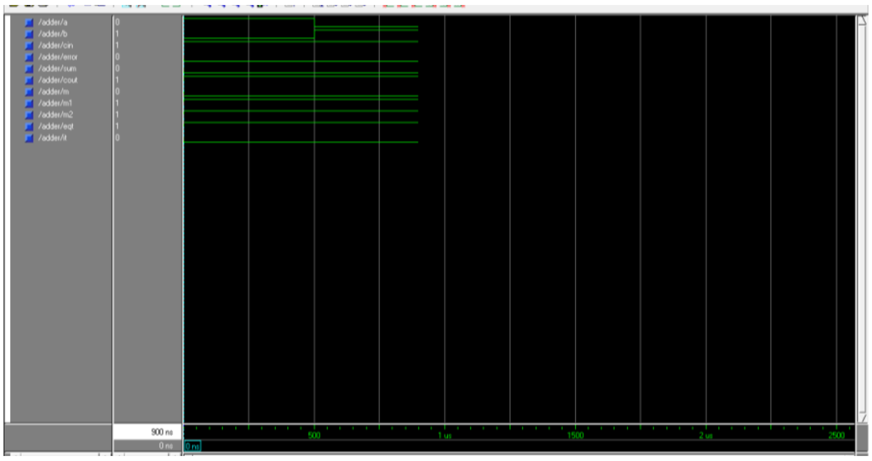


Figure3 RTL schematic of proposed full adder

From the stimulation result shown in figure 4, we can able to see that the inputs a and b are given. the outputs sum and cout are shown. error indicates the whether there is a fault in full adder or not. if it is 0, then there is no fault. if the error is 1, then there is a fault.



```
/adder/a      0
/adder/b      1
/adder/cin    1
/adder/error  0
/adder/sum    0
/adder/cout   1
/adder/m      0
/adder/m1     1
/adder/m2     1
/adder/eqt    1
/adder/it     0
```

Figure 4 stimulation result of full adder

IV EFFICIENT SELF CHECKING CARRY SELECT ADDER

A carry-select adder[3] pre computes sum bits using two parallel ripple-carry adders (RCAs), with complemented values of the initial C_i, and the actual value of the cin will be used to determine the final sum bit. In existing design utilized both RCAs to obtain the complementary behavior of the corresponding sum bits. However, it is possible to perform a logical operation such that one of the RCA blocks should always provide inverted sum bits with respect to the opponent block for checking purposes only. This will provide a more simplified and systematic design, which can be extended easily. Here there is only one possible way in which the sum bits calculated at initial Ci=0 are altered, such that they become complementary to the sum bits calculated at an initial Cin=1for comparison. Except for the least significant bit, the sum bit computed when initial carry in equals 0 will be complementary to the corresponding sum bit with an initial carry- in equal to 1 only when all the lower sum bits are equal to logic-1

For example
Adding two 4-bit number will give 4+12→ (input) 0100+1100→output(01)
The code 01→valid code (indicate there is no fault in the circuit) with the help of this we can find the actual fault in the circuit

- 4-bit CSA (carry select adder) is design by cascading two 2-bit CSA.
- The carry input of **first** CSA is only **complement** the carry input of the **second** CSA is **different**

So, some logical operation is performed before it is given to the input of 2-pair 2-rail checker

- The sum output of the first CSA is multiplied with the sum output of the initial full adder of the second CSA.

To maintain **complemented input** to the two pair two rail checkers to detect the **actual** fault
Figure 5 shows the self checking carry select adder which can find the faults when the inputs are more than 2 bits.

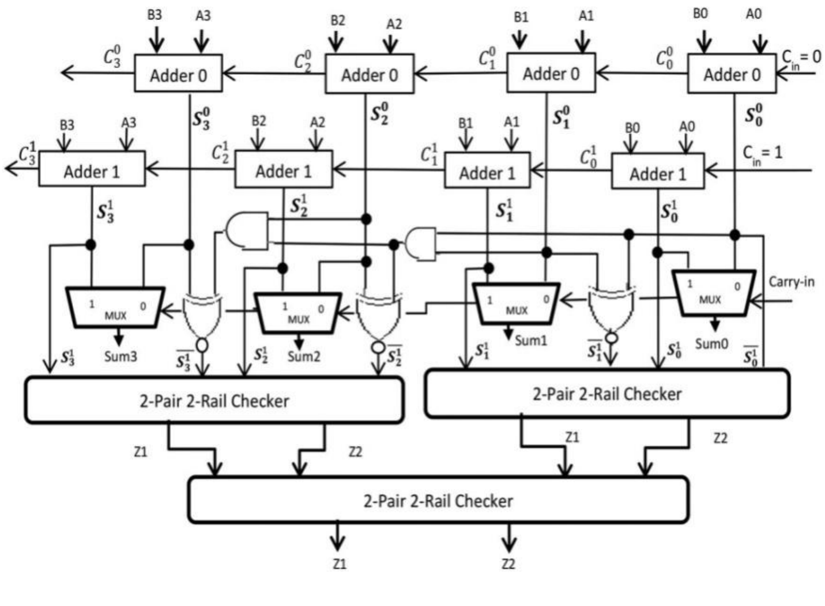


Figure 5 self checking carry select adder

V. SELF CHECKING CARRY SELECT ADDER WITH FAULT LOCALIZATION

Figure 6 shows the proposed full adder is implemented in self checking carry select adder.

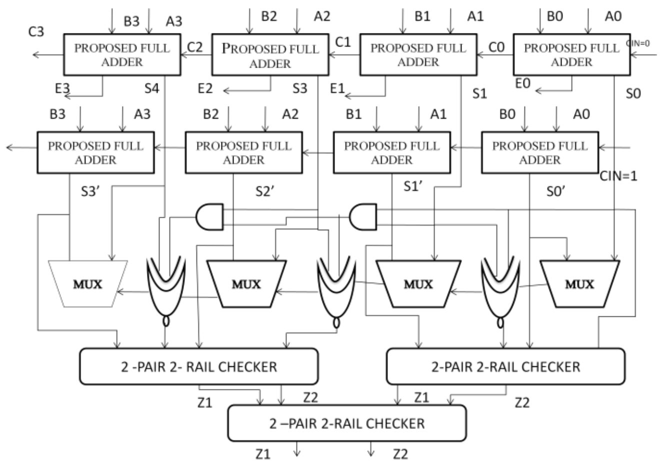


Figure 6 self checking carry select adder with fault localization

The logic-high of final error (Ef) will indicate fault. It can easily be observed that the designed approach guaranties self-checking only when any one of the sum, Cout or E_{qt} line becomes faulty. If two of them have fault at the same time then that fault cannot be detected. The assumption of having a single fault at a time becomes valid because the fault secure property assumes that between two consecutive faults we have enough time to detect the error at its first occurrence. table 1 shows the conditions for fault coverage

Table 1 conditions for fault coverage

CONDITIONS	STATUS
If (Eqt== 0) AND (sum== cout)	No Fault
If (Eqt== 1) AND (sum== cout)	No Fault
If (Eqt== 0) AND (sum≠ cout)	Fault
If (Eqt== 1) AND (sum≠ cout)	Fault

Figure 7 shows the RTL schematic of proposed self checking carry select adder

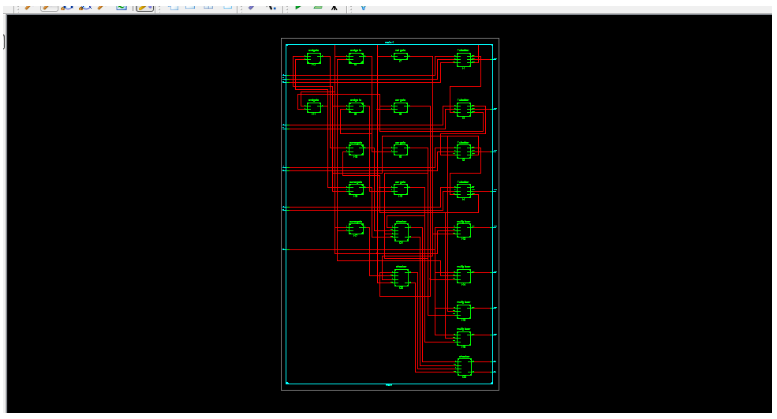


Figure 7 RTL schematic

With the proposed approach for self-checking adder a fault generated in any full adder can be detected individually.
Even if the generated carry has a fault it will not create a fault indication in the next full adders because all the full adders are self-checking with respect to their individual functionality and are independent on their carry input. the final output is shown in figure 8

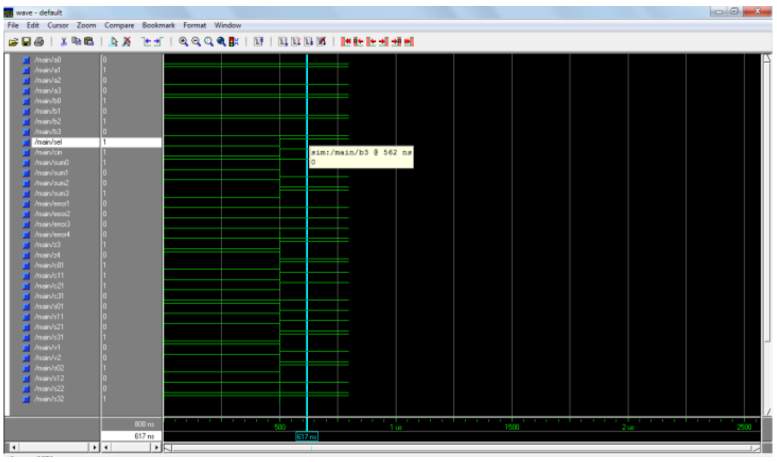


Figure 8 Final output

The total delay for self checking carry select adder with fault localization is 9. 41 ns.

VI CONCLUSION

I have proposed a new design for self checking carry select adder with fault localization based on 2 -pair 2-rail checker that encodes the sum bits and detect all single struck at fault on the line even if the input bit is greater than 2. Moreover the proposed self checking carry select adder can be easily extended to multi digit addition. However the detection of double fault is not guaranteed. With the proposed self checking full adder design a fault generated in any full adder can be detected individually. Even the error propagated through Carry will not create fault indication in the next full adders because all the full adders are self checking with respect to their individual functionality. Due to the fault localization property it will provide minimum area-overhead for self-recovery as well, because instead of replacing the whole system we can now replace the faulty modules only.

REFERENCES

[1] Smith, James E., and Paklin Lam. "A theory of totally selfchecking system design. " *Computers, IEEE Transactions on* 100, no. 9, pp. 831-844, Sept. 1983.

[2] Muhammad Ali Akbar, and Jeong-A. Lee, Self-Checking Carry Select Adder with Fault Localization. *DSD, page 863-869. IEEE Computer Society,* (*2013*)

[3] Muhammad Ali Akbar and Jeong-A Lee Comments on “Self-Checking Carry-Select Adder Design Based on Two-Rail Encoding” IEEE transactions on Circuits and systems-I, Reg. Papers, vol 61, no. 7, july 2014

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