A Fault Tolerant Carry Select Adder with Modular Self Checking Scheme

Gunbae Kim
Computer Systems & Reliable SOC Lab.
School of Electrical and Electronic Engineering
Yonsei University, Seoul, korea
kgb9572@soc.yonsei.ac.kr

IlWoong Kim
Computer Systems & Reliable SOC Lab.
School of Electrical and Electronic Engineering
Yonsei University, Seoul, korea
woong@soc.yonsei.ac.kr

Ilgweon Kang
Computer Systems & Reliable SOC Lab.
School of Electrical and Electronic Engineering
Yonsei University, Seoul, korea
neokings@soc.yonsei.ac.kr

Sungho Kang
Computer Systems & Reliable SOC Lab.
School of Electrical and Electronic Engineering
Yonsei University, Seoul, korea.
shkang@yonsei.ac.kr

Abstract - *This paper presents Fault Tolerant Carry Select Adder (FT-CSA), most widely used type of adder, based on the self checking scheme with modular architecture. The error recovery capability is derived using generic input/output combination of carry select adder with predetermined fault and error set. The experimental results show that proposed FT-CSA has nearly 50% overhead compared with the typical CSA without fault-tolerant scheme but achieves the significant improvement compared with typical redundancy scheme.*

Keywords: fault tolerant design, self-checking adder, on-line test, carry select adder

1 Introduction

With the rapidly shrinking dimensions and the diminishing voltage levels of VLSI circuits, high performance systems with lower area and faster operation speed are being developed and implemented in the various fields. But, due to this enhanced technology, systems are also becoming increasingly sensitive to temporary or transient faults caused for instance by crosstalk, power supply noise, alpha particles and other reasons. These transient faults can be only detected by on-line detection or concurrent checking but not by testing. Since adders are essential building blocks in all data processing systems, the design of arithmetic structures with on-line error detection and correction capabilities represents an important research topic. In the literature, a number of self-checking adder implementations have been proposed, such as based on residue codes, parity codes or Berger codes. Other solutions based on redundancy, such as N-modular redundancy (NMR) and time-redundancy, have been proposed in the literature such as in [1], [2], [3]. The main concept of these schemes is to add additional time, information and hardware redundancy in the original system. Meanwhile, other strategies to develop the modified adder architecture for self-checking and fault tolerant scheme also have been proposed in [4], [5], [6], [7], [8], [9]. Among them, [9] proposed a modular self-checking adder design method by using simple, but very clever observation that the internal partial sums of two full adder block with the same carry in value of 1 in the 2-bit CSA are always complementary if there is no error. With this observation, the modular self-checking CSA block is implemented using just one EX-NOR gate per block and 2-pair 2-rail checker. But this method just has self-checking capability and there are no additional considerations for error correction after error detection. To enhance the capability of error correction, we present the fault tolerant CSA using self-checking method previously presented in [9].

This paper proposes the Fault Tolerant Carry Select Adder (FT-CSA) based on the modular architecture with self-checking capability. The heuristic method for FT-CSA architecture using the intensive analysis of CSA will be proposed in 2. The hardware implementation results and conclusion are discussed in section 3 and 4.

2 Fault Tolerant Carry Select Adder (FT-CSA)

2.1 Intensive observation of CSA for error correction

Before the representation of the proposed architecture, several assumptions are to be made in advance.

Assumption 1.
The error type targetted in the proposed method is a single event upset which is widely used in the on-line test area.

Assumption 2.
There are no errors in the primary data bits and actual carry in signal. We only consider the internal error of CSA.

To utilize the method proposed in [9] and supplement the error correction capability, additional intensive observations of the CSA are needed. Talble 1

shows the reudced set of the all possible combinations of CSA and all possible sum bit error patterns. As shown in this table, it can be observed that there are only 4-possible normal sum bit patterns in the combinations of PI patterns in CSA and two final carry-out bits of 2-bit CSA in each group have some specific characteristics. Also, error patterns of 4 partial sum bits are partially correlated with G1 and G3 or G2 and G4 respectively. The shaded rows in the table represent the correlated error patterns and the other ones are the identical error patterns. Therefore, if the aliasing problem can be solved, corrected 4-bit partial sum patterns also can be made.

Table 1. Reduced set of all possible combinations for partial-sum bit pattern and errorneous pattern

	Good Pattern (GP)	Primary Input (PI)		Erroneous Pattern (EP)
	S01 S11 S00 S10	a1 b1 a0 b0	C02 C12	S01 S11 S00 S10
G1	0 0 0 1	0000	00	1 0 0 1
		0111	11	0 1 0 1
		1011	11	0 0 1 1
		1100	11	0 0 0 0
G2	0 1 1 0	0001	00	1 1 1 0
		0010	00	0 0 1 0
		1101	11	0 1 0 0
		1110	11	0 1 1 1
G3	1 1 0 1	0011	00	0 1 0 1
		0100	00	1 0 0 1
		1000	00	1 1 1 1
		1111	11	1 1 0 0
G4	1 0 1 0	0101	10	0 0 1 0
		0110	10	1 1 1 0
		1001	10	1 0 0 0
		1010	10	1 0 1 1

Table 2 shows the result of $a_i \oplus b_i$ for each group to find the characteristics for group selection. As shown in the table, G2 and G4 have unique results but G1 and G3 has the same results and it is another aliasing problem such as that of EP. Therefore, additional decision process is also needed in the corresponding Error Correction Logic (ECL).

Table 2. Group selection using XORed data bit

$L_i = a_i \oplus b_i$			
G1	G2	G3	G4
L1 L0	L1 L0	L1 L0	L1 L0
0 0	0 1	0 0	1 1
1 0	0 1	1 0	1 1
1 0	0 1	1 0	1 1
0 0	0 1	0 0	1 1

Figure 1 shows the entire procedure of the proposed method for group selection and error correction. If error detected by the 2-rail checker output, first $a_i \oplus b_i$ is calculated to decide the corresponding group for input data combination. Then, ECL1 and 2 make the corrected partial sum bit patterns for group 2 and 4 or 1 and 3 respectively. At the same time, as final carry out bits may be corrupted, carry-out bits are also checked and corrected by each ECL. The final outputs generated by each ECL are selected by the signals $a_i \oplus b_i$.

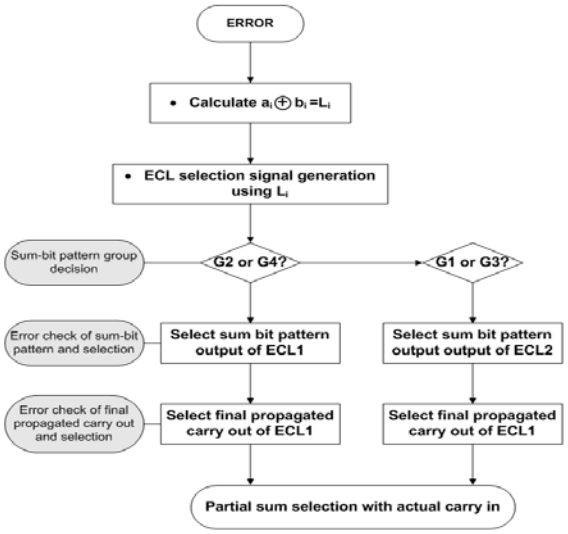

Figure 1. The entire procedure of proposed method

2.2 Proposed architecture for FT-CSA

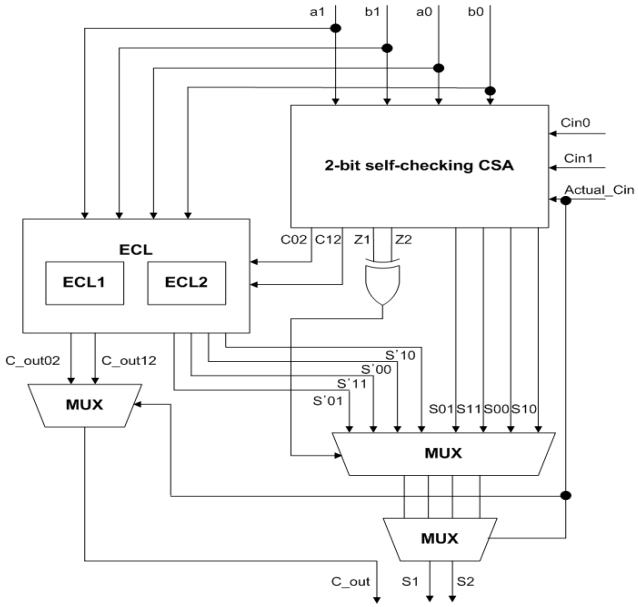

Figure 2. The entire architecture of FT-CSA

Figure 2 is the entire architecture of the proposed FT-CSA. It is composed of 2-bit CSA with self-checking capability proposed in [9], ECL which has two parallel error correction logics, ECL1 and ECL2, XOR gate to generate error detection signal using 2-rail checker output and MUX to select the normal signal and corrected signal for error.

Figure 3 shows the detailed internal architecture of ECL. ECL is composed of two sub-ECL block, ECL 1 and ECL2, two MUXs, one for the selection between the two corrected sum bit patterns, the other for the selection between the two pairs of corrected final carry out signals using the same signal and group selection logic which is implemented just using the characteristics of the result of Table 2. As FT-CSA adopts the parallel architecture to generate corrected signal and normal output, performance degradation can be minimized.

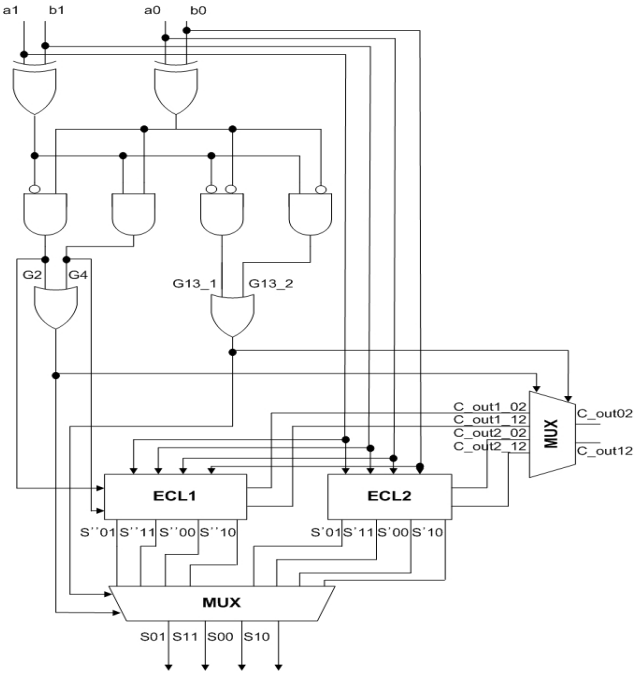

Figure 3. The architecture of ECL

As already mentioned above, ECL has to consider the two types of aliasing problem in EP patterns and group selection. Actually, the group selection process can take care of the aliasing and correlation problem of group G2 and G4 in Table 1. For example, using the specific results of the $a_i \oplus b_i$, G2 and G4 can be distinguished from the other groups. If the group is determined, then the good patterns are just remapped to the corrected output using the input data bits regardless of the error because of the parallel processing architecture. Therefore, in G2 and G4, the error correction logic can be simply implemented.

Figure 4 shows the internal architecture of ECL1 for group G2 and G4. In G2 and G4, the PI patterns of a1, b1, a0, b0 in Table 1 have special characteristics.

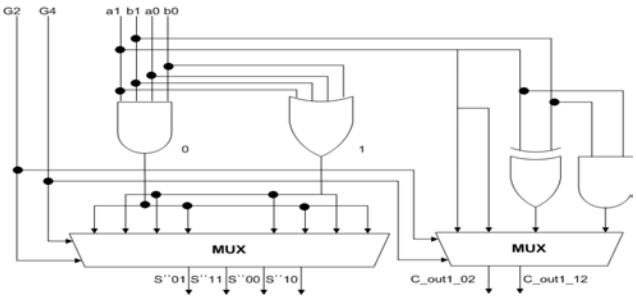

Figure 4. The internal architecture of ECL1

For G2, at least one 0 is in all 4 PI patterns of a1, b1, a0, b0 and 4-bit AND has always the value of 0. Also, for G4, at least one 1 is in all 4 PI patterns of a1, b1, a0, b0 and 4-bit OR has always the value of 1. Since the result of ECL1 is only correct for G2 or G4 case, when G1 or G3 is selected, the result of ECL1 is not meaningless and winnowed by the resultant signal of the group selection logic.

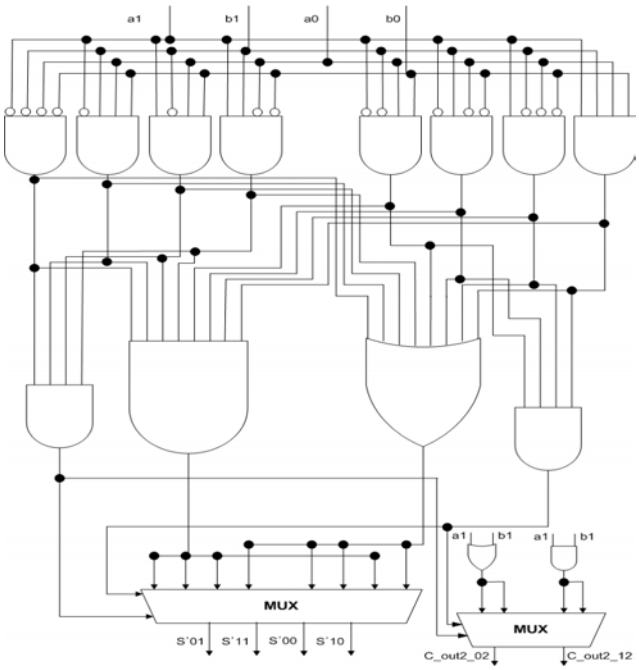

Figure 5. The internal architecture of ECL2

Figure 5 shows the internal architecture of ECL2 for group G1 and G3. ECL2 is also implemented using all 8 PI patterns of a1, b1, a0, b0 in Table 1. Similar to the ECL1, all 4 PI patterns of a1, b1, a0, b0 for G1 have at least one 0 and all 4 PI patterns of a1, b1, a0, b0 for G3 have at least one 1. Using this characteristics, two 4 input XOR gate generate the selection signal to choose the corrected output for G1 or G3. 8-bit AND gate and OR gate is just for generation of the value of 1 and 0.

3 Experimental results

The Proposed FT-CSA is implemented using SYNOPSYS design compiler with TSMC 0.25 micron

library. Table 3 shows the hardware synthesis results of the proposed architecture. As shown in this table, hardware overhead is nearly 50% for n-bit addition and slightly reduced as the number of bits for addition increases.

The previous method proposed in [9] guarantees only the detection of error. Because only detection is meaningless, to use this information for error correction, re-computation of the original data is the only way for the error correction. But this strategy may cause serious performance degradation unnecessarily such as pipeline stall.

Table 3. Hardware syntehsis results

Number of bits	Cell area	Overhead (%)
4 bit addition	2012	54.24
8 bit addition	3987	53.15
16 bit addition	8024	52.96
32 bit addition	15989	52.07
64 bit addition	31043	51.99

Although the hardware overhead of the proposed architecture is high, it is much less than that of the fault tolerant architecture using redundancy scheme such as NMR. NMR duplicates the original module with N-1 redundant modules and requires reliable voter. Therefore, its overhead depends on the number of modules used in the system and is greater than N*100%.

Table 4. Overhead analysis compared with NMR system

Fault tolerant scheme	Required Overhead
NMR	≥ (N-1) x 100% (at least N≥3)
Proposed	≈53%

4 Conclusion

The fault tolerant CSA architecture based on the modular self-checking capability is proposed with predetermined condition. The proposed architecture is derived from the inherent combinations of CSA and additionally analyzed information. To minimize the performance degradation, all components in the proposed FT-CSA are implemented with only combinational logic. To enhance the fault tolerance capability of proposed architecture, the methodology to cover the errors in the primary data bits and actual carry in signal is to be developed and is our future remaining work.

Acknowledgment

The authors would like to appreciate CAD tool support from IC Design Education Center (IDEC).

References

[1] Alderighi M., D'Angelo S., Metra C., and Sechi G.R., "Achieving Fault-Tolerance by Shifted and Rotated Operands in TMR Non-Diverse ALUs," *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 155-163, Oct. 2000.

[2] Townsend W.J., Thornton M.A., and Lala P.K., "On-Line Error Detection in a Carry-Free Adder," *Proceedings of 11th IEEE/ACM International Workshop Logic & Synthesis*, pp. 251-254, June 2002.

[3] Shih, F.W, "High performance self-checking adder for VLSI processor," *Proceedings of the IEEE 1991 Custom Integrated Circuits Conference*, pp. 15.7/1-15.7/3 May 1991.

[4] Cardarilli, G.C., Ottavi, M., Pontarelli, S., Re, M., Salsano, A., "Fault localization, error correction, and graceful degradation in radix 2 signed digit-based adders," *IEEE Transactions on Computers* Volume 55, Issue 5, pp. 534 – 540, May 2006.

[5] Song Peng, Manohar, R., "Fault Tolerant Asynchronous Adder through Dynamic Self-reconfiguration," *Proceedings of 2005 International Conference on Computer Design*, pp. 171-178, Oct. 2005.

[6] Ocheretnij, V., Gossel, M., Sogomonyan, E.S., Marienfeld, D., "A modulo p checked self-checking carry select adder," *Proceedings of 9th IEEE International On-Line Testing Symposium 2003*, pp. 25–29, July 2003.

[7] Ocheretnij, V., Marienfeld, D., Sogomonyan, E.S., Gossel, M., "Self-checking code-disjoint carry-select adder with low area overhead by use of add1-circuits," *Proceedings of 10th IEEE International On-Line Testing Symposium, 2004*, pp. 31-36, July 2004.

[8] Sogomonyan, E.S., Marienfeld, D., Ocheretnij, V., Gossel, M., "A new self-checking sum-bit duplicated carry-select adder," *Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2004*, Volume 2, pp. 1360 – 1361, Feb. 2004.

[9] Vasudevan, D.P., Lala, P.K., "A technique for modular design of self-checking carry-select adder," *Proceedings of 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems 2005,* pp. 325–333, Oct. 2005.