

Fault tolerant design of Signed Digit based FIR filters

G.C. Cardarilli, S. Pontarelli, M. Re, A. Salsano

{pontarelli,salsano}@ing.uniroma2.it

{marco.re, g.cardarilli}@ieee.org

Department of Electronic Engineering

University of Rome "Tor Vergata"

Via Del Politecnico 1 00133 Rome, ITALY

Abstract— This paper proposes a methodology for the development of fault tolerant arithmetic circuits using an r -radix Signed Digit (SD) representation. A residue checking based technique is applied to detect errors caused by faults belonging to the considered stuck-at fault set. We developed a technique to detect the presence of a fault in the adder by using a completely independent circuit that uses check symbols that are residues of the numbers modulo a suitable base. We show that for any radix $r > 3$ the correctness of the adder operation can be checked simply by using two check symbols. This property is used to extend the fault detection also to SD constant multipliers and to the rounding operation. The extension to SD constant multiplier can be easily obtained by implementing the multipliers using a shift and add architecture. For the truncation operation we notice that many error detection techniques based on some arithmetic properties of the circuits fails when the truncation operation is performed. Instead, we show how our method can be applied to this operation with low area overhead and therefore it is useful to implement self-checking Finite Impulse Response (FIR) filters.

I. INTRODUCTION

The silicon integrated circuits trend is characterized by a steady reduction in the feature size combined with a steady rise in density and speed [1]. A lot of new problems related to this incredible increase of complexity must be faced both from the technological and architectural point of view. In this scenario, both permanent and transient fault probability increases limiting the silicon foundry yield and the reliability and availability of the implemented circuits [1], [2]. The success of the emerging technologies depends on an early identification and analysis of these problems together with the development of suitable design methodologies and techniques both in terms of cost and complexity. This can be obtained by approaching the problem at different abstraction levels and from different point of views: technological, architectural and design methodology. While the silicon foundries will face the technological problems in order to optimize costs and productivity, from the point of view of the design of electronic systems, the final targets in terms of availability and reliability can be obtained by avoiding faults (fault avoidance), or by using fault tolerance techniques [3], [4]. The first step in fault tolerant circuits is the detection of the occurrence of a fault. After the detection of a fault different strategies like fault masking or reconfiguration and repair [3], [4] can be used

to obtain the requested reliability levels. These considerations are general, but in particular are valid for DSP applications, where often the requirements in term of speed and density are very high, and the mandatory use of technologies with the best available feature size implies the increase of the probability of fault occurrence. The basic building blocks of all DSP applications are adders and multipliers. Many works have been published on the detection of faults in these structures. In particular self-checking adders based on residue codes [5], [6], parity codes [7], or Berger codes [8] have been proposed. From the performance standpoint the implementation of an adder in Signed Digit representation such as in [9], allows obtaining fast arithmetic circuits due to the its carry-free property. In the previous literature [10], [11] other solutions based on radix-2 signed digit self checking adders have been proposed. In particular, in [9] an inherent parity coding scheme is proposed to code the digits while in [10] a 1 out of 3 scheme is investigated. For complex structures such as FIR filter techniques based on the Residue Number System (RNS) have been proposed [12],[13]. We notice that those methods presents serious drawbacks related to the scaling operations because the self-checking property is not guaranteed. The solution proposed in this paper is based on r -radix signed digit representation and the correctness of the adder operation is obtained by using two check symbols that form a residue of the input operands. The fault detection is extended to SD constant multipliers and to the truncation operation. These extensions allows to apply this method to the design of self-checking FIR filters. The paper is organized as follows: in section II a background of SD arithmetic is given, while in Section III the error detection method is presented for the adder operation. Section IV extends this method to the SD constant multiplier. In Section V, the method is used for the truncation operation and in Section VI the implementation of the whole self-checking FIR filter is shown. The Conclusions are drawn in Section VII.

II. SIGNED DIGIT REPRESENTATION BACKGROUND

The general theory of the SD representation is reported in [9], [14]. In this section a brief illustration of its basic theory is shown. In a radix r SD representation a number x can be represented as

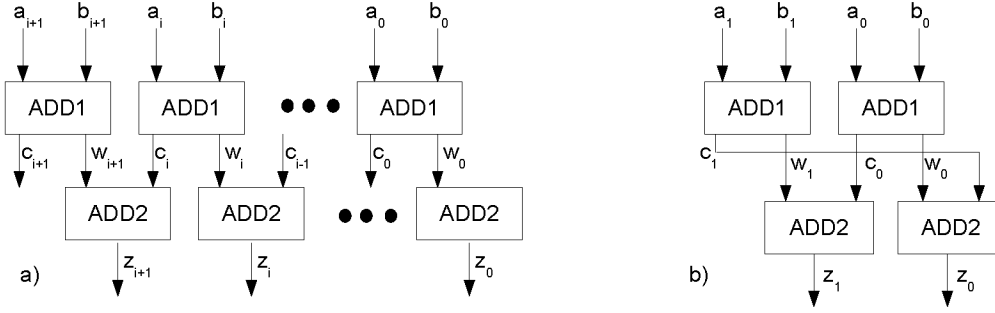


Fig. 1. a) Signed Digit Adder b) Signed Digit Adder modulo $r^2 - 1$

$$x = \sum_{i=0}^{n-1} x_i r^i \quad (1)$$

Where the digit set is $x_i \in \{-a, \dots, -1, 0, 1, \dots, a\}$, with $\lceil \frac{r-1}{2} \rceil \leq a \leq r-1$.

The original motivation for introducing SD representation was to eliminate the carry propagation chains in addition and subtraction [9]. In fact, given two operands x and y the addition operation can be split in the two operations

$$w_i = x_i + y_i - r c_i \quad (2)$$

$$z_i = w_i + c_{i-1} \quad (3)$$

where

$$c_i = \begin{cases} 1 & \text{if } (x_i + y_i) \geq a \\ -1 & \text{if } (x_i + y_i) \leq -a \\ 0 & \text{if } |x_i + y_i| < a \end{cases} \quad (4)$$

being $w_i \in \{-a+1, \dots, -1, 0, 1, \dots, a-1\}$ an auxiliary variable. This representation allows to implement a carry-free adder using a block ADD1 for the equations (2) and (4) and a block ADD2 for the equation (3), connected like described in Fig. 1a).

III. ERROR DETECTION IN SD ADDERS

It is widely known that an adder can be checked using check symbols that are residues of the number modulo some base. The check symbols are able to detect any kind of fault inside the adder if the erroneous value has the check symbols different from the ones of the faulty free operation. For an adder with the check symbol based on residue codes this property can be written as:

$$\langle a + b \rangle_m \neq \langle a + b + e \rangle_m \Rightarrow \langle e \rangle_m \neq 0$$

Where $\langle \cdot \rangle_m$ is the modulo m operation, a and b are the operand of the adder and $e \neq 0$ is the error due to a fault inside the adder. For a radix r signed digit adder the faults can occur in the ADD1 and ADD2 blocks of fig.1a).

A fault in the ADD1 block (or in an input digit a_i or b_i) can change the values of the intermediate results w_i and c_i . If we define $\bar{w}_i \in \{-a+1, \dots, -1, 0, 1, \dots, a-1\}$ and $\bar{c}_i \in \{-1, 0, 1\}$ the faulty intermediate results, the error value can be computed as:

$$e = e_i r^i = e_w r^i + e_c r^{i+1} = (\bar{w}_i - w_i) r^i + (\bar{c}_i - c_i) r^{i+1} \quad (5)$$

with $|e_w| = |\bar{w}_i - w_i| \leq 2(a-1)$ and $|e_c| = |\bar{c}_i - c_i| \leq 2$. Therefore for e_i we obtain: $|e_i| \leq 2r + 2a - 2$. A fault in the ADD2 block can change the values of the result z_i to \bar{z}_i , the error value is $e = e_i r^i$ with $|e_i| \leq 2a$. Now we can choose an integer m for which $\langle e \rangle_m = 0$ iff $e = 0$. This condition is satisfied for all m for which

$$m > 2r + 2a - 2 \geq 2r + 2(r-1) - 2 = 4r - 4 \quad (6)$$

Using $m = r^2 - 1$ equation (6) becomes $r^2 - 4r + 3 > 0$ that is satisfied for all radix $r > 3$. The implementation of the adder modulo $r^2 - 1$ can be implemented by using the end-around carry structure shown in fig. 1b).

IV. ERROR DETECTION IN SIGNED DIGIT CONSTANT MULTIPLIERS

The shift and add implementation of constant multiplication performing the operation $c = k \cdot a$ can be easily extended to the signed digit representation using a shift and add/subtract architecture [14]. The multiplication for the radix r is performed simply shifting the digits of a , while the multiplication for $-r$ is done shifting and inverting all the digits. To minimize the number of adders required to perform constant multiplication the number of non-zero digits must be minimized e.g. factorizing the constant of the multiplier (see [15] for details). As an example we show the implementation of a radix-4 constant multiplier using as a constant the value 35. The direct implementation can be realized as $35 \cdot a = (16 + 16 + 4 - 1) \cdot a = 16a + 16a + 4a - a$, using four adders. The value can be factorized as $35 = 7 \cdot 5 = (4 + 4 - 1) \cdot (4 + 1)$, and using the auxiliary variable b the multiplication can be performed using 3 adders as shown in the following formulas:

$$\begin{aligned} b &= 5 \cdot a = 4a + a \\ c &= 7 \cdot b = 4b + 4b - b \end{aligned}$$

In Fig. 2 the direct and factorized implementations of this multiplier are shown.

We notice that each adder used in the constant multiplier satisfies the condition described in the previous section and therefore can be checked by using the residue modulo $r^2 - 1$ of the operand. Using the single fault occurrence hypothesis

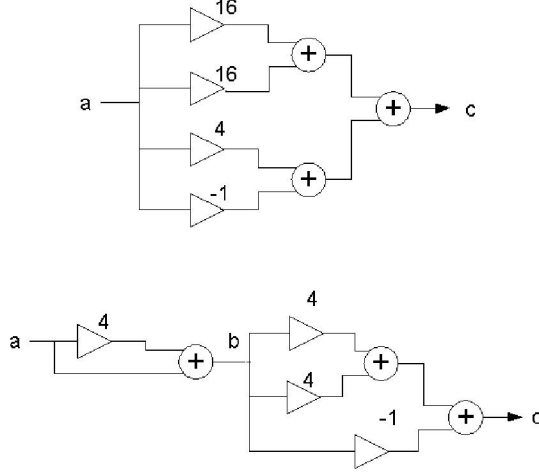


Fig. 2. example of a radix 4 Signed Digit constant multiplier (k=35)

[3], [4] we can use a block that take as input the check symbol $\langle a \rangle_m$ and compute directly the operation $\langle k \cdot a \rangle_m = \langle \langle k \rangle_m \cdot \langle a \rangle_m \rangle_m$. This check symbol can be used to detect the occurrence of a fault inside the constant multiplier checking if $\langle c \rangle_m = \langle \langle k \rangle_m \cdot \langle a \rangle_m \rangle_m$ only after that all the addition and shift operations are performed. This approach can be used also if the constant multiplier is a block of a more complex circuit, allowing to perform the error detection only at the last stage of the overall structure.

V. ERROR DETECTION IN SIGNED DIGIT TRUNCATION OPERATION

The truncation operation is commonly used in complex arithmetic systems to avoid the increase of the number of digit during the computation. To avoid that the circuit become unfeasible due to the number of digit to be processed truncation operations are performed in order to reduce the dynamic range of the intermediate results. For positional number representations the truncation operation consists in discarding the less significant digits, while for non positional number system like RNS [14] a suitable scaling block must be used. If we discard the less significant digits a problem arise in the error detection strategy explained in the previous sections. We outline that this problem arises for almost all the error detection methods based on the arithmetic properties of the operands.

If a is represented in the radix r SD number system the truncation can be defined as:

$$b = \text{trunc}(a) = (a - \langle a \rangle_{r^n}) / r^n$$

where the $\langle \cdot \rangle_{r^n}$ operation is performed taking the less significant n digit of a and it is easily to see that $\langle b \rangle_m \neq \langle a \rangle_m$ with $m = r^2 - 1$. If we suppose that the truncation operation is performed on an even number of digits n we obtain $\langle b \rangle_m = \langle a - \langle a \rangle_{r^n} \rangle_m$ and we can compute $\langle b \rangle_m$ starting from $\langle a \rangle_m$ and from the less significant n digits of a is this way:

$$\langle b \rangle_m = \langle a \rangle_m - \langle \langle a \rangle_{r^n} \rangle_m$$

Now, let us analyze the behavior of this structure in presence of a fault. For the most significant digits a fault can only affect one of the lines transporting the digits throughout this block and therefore change the value of a single digit. Instead, if a fault occurs in one of lines transporting the less n digit, in a line transporting the check symbol $\langle a \rangle_m$, or in the modulo m adder that computes $\langle b \rangle_m$ only the check symbol is affected by this fault and therefore can be detected as in the cases described in the previous sections.

VI. ERRORS IN FIR FILTERS

In this section the considerations discussed above are used to implement a complex arithmetic structure like a Finite Impulse Response (FIR) filter. The equation of a FIR filter is:

$$y(n) = \sum_{k=0}^P a_k \cdot x(n-k) \quad (7)$$

where $x(n)$ and $y(n)$ are the input and output sequences, and a_k are the filter coefficients. If no truncations are performed in the hardware implementation of the FIR filter the Concurrent Error Detection (CED) can be done computing the residue modulo m of $y(n)$ directly starting from the residue modulo m of the input sequence $x(n)$. In Fig. 3 an implementation of the CED scheme is presented,

where the operations performed by the block computing $\langle y(n) \rangle_m$ are performed modulo m . If no truncations are performed in the filter structure the computing of $\langle y(n) \rangle_m$ can be done in parallel with the computing of $y(n)$, otherwise the discarded digits are treated as additional inputs for the block computing the check symbol, following the equation presented in section V. The area overhead of the CED structure with respect to the basic implementation is quite independent by the number of digits used for the FIR filter. In fact, only the area overhead of the two blocks computing the $\langle \cdot \rangle_m$ operation, and eventually of the blocks used to face the effect of the truncation operation depend by the number of digit used in the filter. The other operations performed in the additional hardware requires only the two digits needed to represent the numbers modulo m . Finally, the block named equality checker can be implemented as a two-rail equality checker [3], [4] in order to avoid undetectable fault inside this block.

VII. CONCLUSIONS

In this paper a methodology for the concurrent error detection of fault in r -radix Signed Digit (SD) arithmetic operations is presented. A residue checking based technique is applied to detect errors caused by faults inside the hardware used to implement operations in the SD number system. A fault inside the adder can be checked by using a completely independent circuit that use check symbols that are residues of the numbers modulo a suitable base. This property is used to extend the fault detection also to SD constant multiplier and to the truncation operation. These properties allows to easily implement complex arithmetic circuits that had the concurrent error detection property and additionally can exploits the carry-free

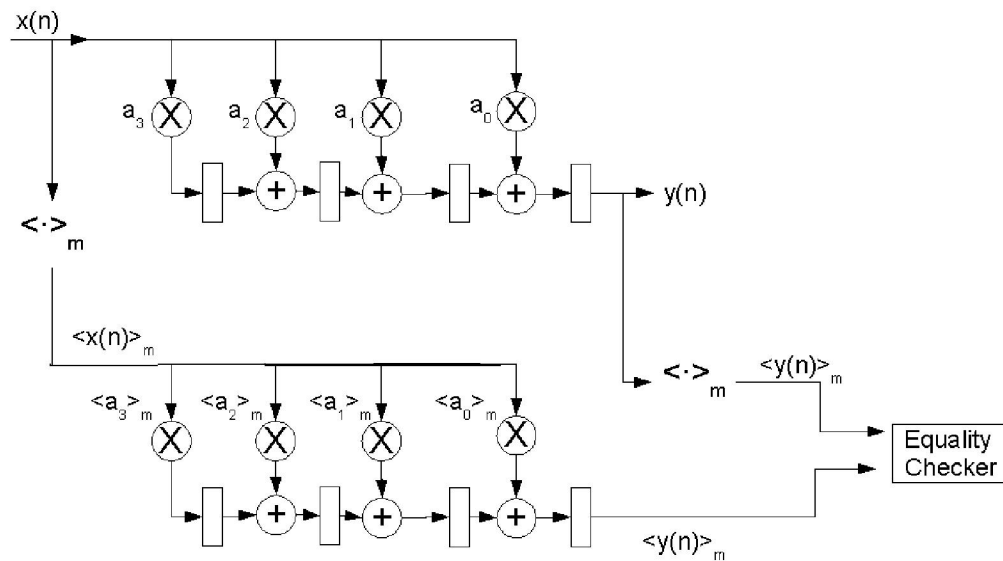


Fig. 3. CED scheme for a FIR filter

property of the SD representation to realize very fast arithmetic computations. In the paper the method is applied to a FIR filter architecture in order to demonstrate its effectiveness. The example shown that the CED scheme requires a low overhead with respect to the basic implementation (the non-CED scheme) and this overhead is quite independent by the number of digits used to perform the filter operation.

REFERENCES

- [1] "2001 International Technology Roadmap for Semiconductors", <http://public.itrs.net/>.
- [2] R. C. Aitken, "Nanometer Technology Effects on Fault Models for ICs Testing" Computer, pp. 46- 51, November 1999.
- [3] P. K. Lala, "Self-Checking and Fault-Tolerant Digital Design", San Mateo, CA: Morgan Kaufman, 2001.
- [4] D. K. Pradhan, "Fault Tolerant Computing, Theory and Techniques", Edited by Prentice-Hall, 1986.
- [5] W. W. Peterson "On checking an adder" I.B.M. J. Res. Develop., vol 2, pp. 166-168, Apr 1958.
- [6] D. Nikolos, A.M. Paschalis, G. Philokyprou "Efficient design of totally self-checking checkers for all low-cost arithmetic codes", IEEE Trans. on Computers, vol. 37, issue 7, pp. 807 - 814, July 1988
- [7] Nicolaidis, M. "Carry checking/parity prediction adders and ALUs" IEEE Transactions on very Large Scale Integration (VLSI) Systems, Volume: 11 Issue: 1 , Feb 2003 pp. 121 -128
- [8] J.-C. Lo, S. Thanawastien, T. R. N. Rao, and M. Nicolaidis, An SFS berger check prediction ALU and its application to self-checking processors designs, IEEE Trans Computer-Aided Design, pp. 525-540, Mar. 1992.
- [9] A. Avizienis, Signed-Digit Number Representations for Fast Parallel Arithmetic IRE Trans. Electronic Computers, vol. 10, pp. 389-400, 1961.
- [10] M. A. Thornton, "Signed binary addition circuitry with inherent even parity outputs" IEEE Trans. on Computers, vol. 46, pp.811-816, July 1997.
- [11] W. J. Townsend, M. A. Thornton, and P. K. Lala, "On-Line Error Detection in a Carry-Free Adder", 11th IEEE/ACM International Workshop on Logic & Synthesis pp. 251-254, New Orleans, LA, June 4-7, 2002
- [12] L. Imbert, G.A. Jullien, "Efficient fault-tolerant arithmetic using a symmetrical modulus replication RNS", Signal Processing Systems, 2001 IEEE Workshop on 26-28 Sept. 2001 Page(s):93 - 100
- [13] M.G. Parker, M. Benaissa, "Fault-tolerant linear convolution using residue number systems", Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on Volume 2, 30 May-2 June 1994 Page(s):441 - 444 vol.2
- [14] M. Ercegovic, T. Lang, "Digital Arithmetic", Morgan Kaufman, 2004.
- [15] A.G. Dempster, M.D. Macleod, "Constant integer multiplication using minimum adders" Circuits, Devices and Systems, IEE Proceedings, Volume 141, Issue 5, Oct. 1994 Page(s):407 - 413