Microelectronics Reliability 51 (2011) 1152–1156

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel

Research Note

Low-complexity Concurrent Error Detection for convolution with Fast Fourier Transforms

C.J. Bleakley [a,1], P. Reviriego [b,*], J.A. Maestro [b,2]

[a] University College Dublin, Belfield, Dublin 4, Ireland
[b] Universidad Antonio de Nebrija, C/ Pirineos, 55 E-28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 6 October 2010
Received in revised form 15 February 2011
Accepted 16 February 2011
Available online 11 March 2011

ABSTRACT

In this paper, a novel low-complexity Concurrent Error Detection (CED) technique for Fast Fourier Transform-based convolution is proposed. The technique is based on checking the equivalence of the results of time and frequency domain calculations of the first sample of the circular convolution of the two convolution input blocks and of two consecutive output blocks. The approach provides low computational complexity since it re-uses the results of the convolution computation for CED checking. Hence, the number of extra calculations needed purely for CED is significantly reduced. When compared with a conventional Sum Of Squares – Dual Modular Redundancy technique, the proposal provides similar error coverage for isolated soft errors at significantly reduced computational complexity. For an input sequence consisting of complex numbers, the proposal reduces the number of real multiplications required for CED in adaptive and fixed filters by 60% and 45%, respectively. For input sequences consisting of real numbers, the reductions are 66% and 54%, respectively.

1. Introduction

Due to shrinking process geometries and reducing operating voltages, soft errors are becoming an increasingly important reliability problem in the implementation of digital systems [1]. The traditional approach to deal with errors or faults has been the use of Modular Redundancy (MR) in which the circuit is replicated such that errors can be detected when two identical modules are used and corrected when three identical modules are used [2]. The first configuration, known as Dual Modular Redundancy (DMR), and the latter, known as Triple Modular Redundancy (TMR), are widely used in fault tolerant systems.

An alternative approach is Algorithm-Based Fault Tolerance (ABFT) [3] in which fault tolerance is incorporated in the algorithm at the system level. ABFT has been applied to the computation of the Fast Fourier Transform (FFT) [3,4]. For example, in [3] the Sum Of Squares (SOS) technique was proposed for detection of errors in the FFT by computing and comparing the sums of the squares of the inputs and the outputs. From Parseval's theorem, if no error has occurred, then the SOSs should be equal [5]. ABFT approaches have also been proposed for fault-tolerant convolution. For example, in [6], cyclic error-correcting codes were used in implementing fault-tolerant convolution. The approach works for direct implementation of convolution but not for transform-based convolution. This is a major drawback as the computational cost of direct implementation is, in most cases, much larger than that of transform-based implementations. In [7–9] approaches based on the use of Residue Number Systems (RNS) for the computation were presented. These approaches typically incur significant area overhead and require specialized arithmetic units. The use of two independent convolutions with different transform lengths was recently proposed in [10]. The approach uses recognition of error patterns in the convolution outputs to determine the module in error and perform correction. The use of two independent convolutions leads to high computational complexity. In [11], an extra zero sample was added to all input data blocks, and error checking was performed at the block output. The cost of implementing the technique is very low, and it provides some error detection. However, error coverage is poor in the inverse transform.

In this work, a novel scheme for detecting errors in FFT-based convolutions is introduced. The technique is based on checking the equivalence of the results of time and frequency domain calculations of the first sample of the circular convolution of two data blocks. This check is applied to the two convolution input blocks and to two consecutive convolution output blocks. The method is of low computational complexity because it re-uses results available as part of the convolution process for CED checking. In addition, the computational complexity of the output block checking is shared between two consecutive convolutions. The computational complexity and single error coverage of the proposed

* Corresponding author. Tel.: +34 914521100; fax: +34 914521110.
E-mail addresses: chris.bleakley@ucd.ie (C.J. Bleakley), previrie@nebrija.es (P. Reviriego), jmaestro@nebrija.es (J.A. Maestro).
[1] Tel.: +353 17165353; fax: +353 12695396.
[2] Tel.: +34 914521100; fax: +34 914521110.

0026-2714/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.microrel.2011.02.010

method is compared to that of a conventional Sum Of Squares – Dual Modular Redundancy (SOS-DMR) approach [3]. The SOS-DMR technique was selected for comparison, since it is a similar low-complexity, detection-only method and does not require the use of a specialized number system. Results show that the proposed technique outperforms the SOS-DMR approach both in terms of error coverage and complexity. This makes it an interesting option for applications in which error detection is required.

The rest of the paper is structured as follows. Section 2 covers the background to the problem. Section 3 details the proposed CED technique, and Section 4 compares the performance of the technique to that of a conventional approach. The paper is concluded in Section 5.

2. Background

The linear convolution $y(n)$ of two sequences $h(n)$ and $x(n)$ can be defined in the time domain as:

$$y(n) = \sum_{i=0}^{L-1} h(i)x(n-i) \tag{1}$$

where L is the length of sequence $h(n)$ [5].

In common filtering applications, the input sequence $x(n)$ is long. For example in many cases the input signal comes from an ADC that produces millions of samples per second. Hence for practical implementation of the convolution, the sequence is segmented into blocks. Convolution is then performed on successive blocks, and allowance is made for the necessary overlapping of data between blocks using, for example, the overlap-save method [5].

The block length depends on the filter impulse response, for example in 10 Gb/s Ethernet transceivers, filters with hundreds of coefficients are used. For large block lengths, the computational complexity of convolution can be reduced by means of the FFT via a frequency domain calculation. The N-point circular convolution, $r(l)$, of two N-point data sequences, $x_1(n)$ and $x_2(n)$, is equal to the Inverse FFT (IFFT) of the multiplication of the FFTs, $X_1(k)$ and $X_2(k)$, of the original sequences [5]:

$$r_{xx}(l) = \sum_{n=0}^{N-1} x_1(n)x_2((l-n)\mathrm{mod}N) = IFFT[X_1(k)\cdot X_2(k)] \tag{2}$$

The implementation of the technique is illustrated in Fig. 1.

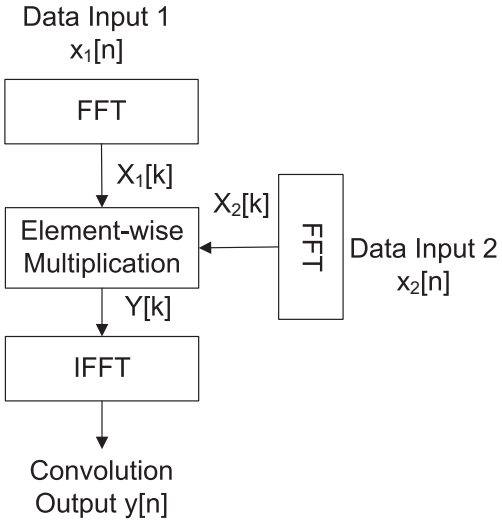

Fig. 1. Implementation of FFT-based convolution.

3. Proposed technique

Firstly, we consider the general case where the convolution input sequences consist of complex numbers, $x_1(n)$ and $x_2(n)$.

Single errors in the calculation of FFTs, $X_1(k)$ and $X_2(k)$, of two N-point sequences, $x_1(n)$ and $x_2(n)$, can be detected by comparing the first sample of the circular convolution $r_{xx}(0)$ as calculated in the time domain $r'_{xx}(0)$ with the same value obtained using a frequency domain calculation, $r''_{xx}(0)$.

In the time domain, $r_{xx}(0)$ can be calculated as:

$$r'_{xx}(0) = x_1(0)\cdot x_2(0) + \sum_{m=1}^{N-1}[x_1(m)\cdot x_2(N-m)] \tag{3}$$

Using the frequency domain, $r_{xx}(0)$ can be calculated as:

$$r''_{xx}(0) = \frac{1}{N}\cdot\sum_{k=0}^{N-1}[X_1(k)\cdot X_2(k)] \tag{4}$$

Most single errors in calculation of $X_1(k)$ and $X_2(k)$, i.e. the FFT computations, can be detected by comparing the time domain results $r'_{xx}(0)$ with the frequency domain result $r''_{xx}(0)$. If the values are the same, then it is highly likely that there are no errors in the FFT outputs $X_1(k)$ and $X_2(k)$. If they differ, then an error has occurred – either in the time domain or the frequency domain computation.

Herein we note that, in the case of FFT-based convolution, the first sample of the circular convolution calculated using the frequency domain method $r''_{xx}(0)$ is obtained as a result of the convolution calculation. Hence, it is available to the CED checker with no computational overhead. As can be noted from Eq. (4), most errors in the FFT and multiplication operations will cause the value of $r''_{xx}(0)$ to differ from the correct value. Therefore, the $r'_{xx}(0) = r''_{xx}(0)$ check provides good single error coverage for both the FFT and the multiplication stages of the convolution process.

Errors in the IFFT stage can be detected by considering the outputs of two successive convolution blocks $y_j(n)$ and $y_{j+1}(n)$. A time domain calculation of the first sample of the circular convolution of the two block outputs is given by:

$$r'_{yy}(0) = y_j(0)\cdot y_{j+1}(0) + \sum_{m=1}^{N-1}[y_j(m)\cdot y_{j+1}(N-m)] \tag{5}$$

Again, this result can be checked against a frequency domain calculation of the same quantity where $Y_j(k)$ and $Y_{j+1}(k)$ are the FFTs of $y_j(n)$ and $y_{j+1}(n)$ respectively.

$$r''_{yy}(0) = \frac{1}{N}\cdot\sum_{k=0}^{N-1}[Y_j(k)\cdot Y_{j+1}(k)] \tag{6}$$

From (2), it can be seen that $Y_j(k)$ and $Y_{j+1}(k)$ are available as part of the convolution process. They are the outputs of the multiplication stages of the successive block computations, i.e. the inputs to the IFFT stages. Hence they are available to the CED checker with no computational overhead.

Most errors in the IFFT stages will cause $r'_{yy}(0)$ to differ from $r''_{yy}(0)$. Thus the $r'_{yy}(0) = r''_{yy}(0)$ check provides good single error coverage for the IFFT stages of two consecutive convolutions.

Thus most single errors in the convolution calculation will be detected by testing the conditions $r'_{xx}(0) = r''_{xx}(0)$ and $r'_{yy}(0) = r''_{yy}(0)$. If they are not met, then an error has occurred, either in the convolution calculation or in the CED checkers. The proposed technique is illustrated in Fig. 2.

It is known that complex multiplication can be implemented as three real multiplications and five real additions [12]. Based on this, the additional computational complexity of calculating the first point of the convolution using the time domain ($r'_{xx}(0)$, $r'_{yy}(0)$) or frequency domain approach ($r''_{yy}(0)$) is $3N$ real multiplies

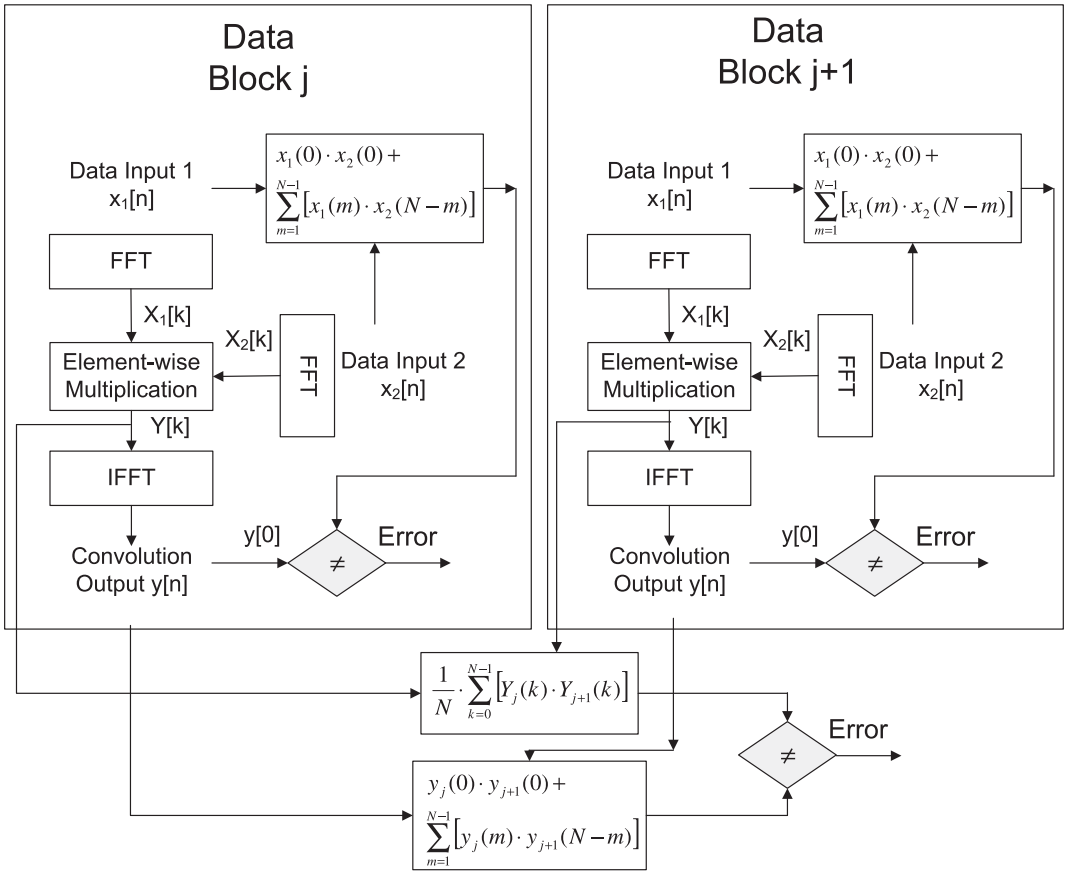

Fig. 2. Implementation of FFT-based convolution with the proposed CED technique.

plus 7*N*-2 real adds. By taking into account that the computational cost of $r'_{yy}(0)$ and $r''_{yy}(0)$ is shared among two blocks, the total computational complexity of the CED checks is 6*N* real multiplications and 14*N*-4 real additions per output block.

In the case of input sequences of real numbers, the computational complexity of the overall convolution can be reduced by noting that the Discrete Fourier Transform (DFT) of a sequence of real numbers is symmetric [13]. If this optimization is employed, then the proposed CED method must be augmented by adding DMR to the calculation of the imaginary part of the element-wise multiplication, i.e. *imag*(*Y*[*k*]).

By taking advantage of symmetry and real data, and allowing for DMR, the computational complexity in the case of input sequences of real numbers is $2.5N + 2$ real multiplications and $4N + 3$ real additions per output block.

For practical implementations, the proposed technique must be robust to round-off errors in FFT and IFFT implementation. This can be done by using a tolerance level, τ, in the check, such that small differences do not trigger an error, for example:

$$\begin{aligned}&(|r'_{xx}(0) - r''_{xx}(0)| < \tau) \text{and} (|r'_{yy}(0) - r''_{xx}(0)| < \tau) \quad \text{no error}\\ &(|r'_{xx}(0)| - r''_{xx}(0)| \geqslant \tau) \text{or} (|r'_{yy}(0) - r''_{yy}(0)| \geqslant \tau) \quad \text{error}\end{aligned} \tag{7}$$

This approach was considered in detail in [3] for checking stand-alone FFTs using Parseval's theorem. In a practical design, the threshold level is determined by simulation. The threshold level is typically set to the value of the maximum observed difference over all error-free simulations plus a safety margin.

When an error is detected, the FFTs associated with the CED checker must be re-computed. The proposed method is not able to determine which FFT suffered the error and therefore, in the worst case, both FFTs/IFFTs must be re-computed. Individual CED checkers applied to each individual FFT/IFFT stage would ensure that in the worst case only one FFT/IFFT would need to be re-computed. However, since error events are rare, the average number of additional operations due to the extra re-computation is negligible when compared to the total number of operations for convolution.

If sufficient memory is available, then the computational overhead of re-computation can be reduced by comparison of the outputs of the first re-calculated FFT/IFFT with the original results. If the outputs differ and errors are rare events, then it can be assumed that the error occurred in the original calculation of the first FFT/IFFT and not in the original calculation of the second FFT/IFFT. Based on this, the system can proceed without re-calculating the second FFT/IFFT. This approach could be used for the FFTs calculating $X_1(k)$ and $X_2(k)$ in the case of the first check and for the IFFTs calculating $y_j(n)$ and $y_{j+1}(n)$ in the case of the second check.

The latency required for error detection is mainly related to the fact that the technique relies on computing the first sample of the convolution of two consecutive output blocks. That means that one block delay is added for the purpose of error detection. For error correction, the time required for re-computation of the blocks in error is added to the latency, such that in the worst case a three-block delay is added to perform error detection and correction.

4. Evaluation

In this section, the proposed technique is evaluated in terms of complexity and fault coverage. For complexity, the required number of operations will be used as the metric for the evaluation. For

Table 1
Computational complexity of the proposed CED technique and of the conventional SOS-DMR technique.

Convolution type	Op	SOS-DMR	Proposed technique	Saving (%)
Complex inputs, adaptive filter	Mult	15N	6N	60.0
	Add	17N-6	14N-4	17.6
Complex inputs, fixed filter	Mult	11N	6N	45.4
	Add	13N-4	14N-4	−7.7
Real inputs, adaptive filter	Mult	7.5N + 9	2.5N + 2	66.6
	Add	8.5N + 5	4N + 3	52.9
Real inputs fixed filter	Mult	5.5N + 7	2.5N + 3	54.5
	Add	6.5N + 5	4N + 3	38.5

fault coverage, the analysis focuses on soft errors [1,2]. These are transient changes in the logic value of a register or logic gate due to a radiation particle hit. After a soft error, the circuit continues its normal operation, and all elements remain fully operational. However, the incorrect logic value can propagate to the module output, leading an output error. For a purely data-path circuit, such as a convolution module, most soft errors propagate through the circuit, leading to output errors. Therefore the effectiveness of the proposed technique is evaluated in terms of the percentage of errors that are detected when soft errors are inserted in the circuit.

4.1. Error model

It is assumed that soft errors are isolated events that occur rarely. For terrestrial applications, soft error frequency is in the order of days or months, while in space applications they are much more frequent [1,2]. In most applications, sampling rates are in the order of kHz–MHz, and block sizes are in the range of 64–5120 samples depending on filter length and memory size. Thus, only a very small percentage of the blocks will suffer errors. Therefore, it is reasonable to assume that soft errors are single events, that is, only one particle hit occurs at a time, and are isolated, that is, errors do not occur in consecutive blocks. However, a single soft error event, caused by a particle hit, can produce a change in the logic value of a single node or of multiple nodes. Changes in the values of multiple nodes arising from a single event are commonly known as Multiple Cell Upsets (MCUs) [14], as opposed to Single Cell Upsets (SCUs). Both SCUs and MCUs were considered herein.

4.2. Computational complexity

The computational complexity of the proposal was compared to that of applying the conventional Sum Of Square (SOS) check [3] to each FFT and IFFT individually and applying Dual Modular Redundancy (DMR) to the multiplication stage. The SOS check relies on Parseval's theorem, which states that the energy, or SOS, of an N point sequence, $x(n)$, and its DFT, $X(k)$, are equal [3].

The computational complexity of the proposed technique and of the SOS-DMR approach is given in Table 1. In the case of fixed filtering, SOS checks are not needed for $X_2[k]$, since it does not need to be re-calculated. In the case of input sequences of real numbers, the number of operations needed for CED is reduced by taking advantage of the symmetry of the DFT for real data.

4.3. Fault coverage

Matlab simulations were run to determine the fault coverage of the proposed method in comparison to the SOS-DMR scheme. The schemes were applied to detect errors in convolutions with inputs sequences consisting of complex random numbers. The real and imaginary parts of the input data were uniformly distributed in the range −0.5 to 0.5. The block length of the data was varied between 64 and 1024 points. Based on the results of error-free simulations, a threshold level of 10^{-5} was used in the comparisons.

Table 2
Single fault coverage for various convolution lengths (N).

N	SOS-DMR (%)	Proposed technique (%)
64	98.70	99.85
128	97.89	99.82
256	96.70	99.79
512	96.42	99.77
1024	96.61	99.73

Table 3
Double fault coverage for various convolution lengths (N).

N	SOS-DMR (%)	Proposed technique (%)
64	99.99	99.997
128	99.99	100
256	99.98	99.998
512	99.98	99.998
1024	99.97	99.995

To assess SCU performance, a single soft error was inserted during the computation of a pair of convolution blocks. The soft error was simulated by adding a random value to either the real or imaginary part of one of the following: (1) the FFT butterfly, (2) the complex multiplication, or (3) the IFFT butterfly output. The operation output in error was selected at random. Since a single soft error can affect a number of module output bits, the simulated error was uniformly distributed in the range −0.5 to 0.5. For example, a SCU immediately prior to a multiplication will cause an error equal to the value of the multiplicand. The simulation was iterated 100,000 times, and the results are presented in Table 2. It can be observed that good fault coverage (>95%) is achieved in both cases with the proposed technique detecting slightly more errors than the SOS-DMR approach. To investigate MCU performance, simulations were performed as in the SCU case except that two simulated errors with random locations and values were introduced in every iteration. The results are given in Table 3.

Both for SOS-DMR and for the proposed technique, a small percentage of errors are left undetected. In most cases, this occurs when the errors at the output are small and, typically, their impact is negligible. The proposed technique is therefore suitable for applications that can tolerate a small number of undetected errors. An example of such application is in communication receivers in which an undetected error may cause internal bit errors that can be corrected by subsequent processing of an error correction code or by retransmission of the data [15].

5. Conclusions

A novel Concurrent Error Detection technique for FFT-based implementations of convolution was proposed. The method is

based on comparison of time and frequency domain calculations of the first sample of the circular convolution of two sequences. The comparison check is applied to each convolution output block and between successive convolution output blocks to ensure adequate error coverage. The method is shown to provide similar single error coverage to, and significant computational complexity savings over, the conventional module SOS-DMR CED technique.

References

[1] Baumman R. Soft errors in advanced computer systems. IEEE Des Test Comp 2005;22(3):258–66.
[2] Nicolaidis M. Design for soft error mitigation. IEEE Trans Dev Mater Reliab 2005;5(3):405–18.
[3] Reddy A, Banarjee P. Algorithm-based fault detection for signal processing applications. IEEE Trans Comp 1990;39(10):1304–8.
[4] Jou J, Abraham JA. Fault-tolerant FFT networks. IEEE Trans Comp 1988;37(5).
[5] Oppenheim AV, Schafer R. Discrete time signal processing. Prentice Hall; 1999.
[6] Redinbo GR. System level reliability in convolution computations. IEEE Trans Acoust, Speech Signal Process 1989;37(8):1241–52.
[7] Beckmann PE, Musicus BR. Fast fault-tolerant digital convolution using a polynomial residue number system. IEEE Trans Signal Process 1993;41(7): 2300–13.
[8] Sundaram S, Hadjicostis CN. Fault-Tolerant Convolution Via Chinese Remainder Codes Constructed From Non-Coprime Moduli. IEEE Trans Signal Process 2008;56(9):4244–54.
[9] O'Donnell AB, Bleakley CJ. Area efficient fault tolerant convolution using RRNS with NTTS and WSCA. Electron Lett 2008;44(10):648–9.
[10] Reviriego P, Maestro JA, O'Donnell A, Bleakley C. Soft error detection and correction for FFT based convolution using different block lengths. In: Proc. IEEE international on-line test symposium (IOLTS'09); June 2009. p. 138–43.
[11] Radhakrishnam C, Jenkins WK. Fault tolerant adaptive filters based on number theoretic transforms. In: 42nd asilomar conference on signals, systems and computers; 2008. p. 261–5.
[12] Blahut RE. Fast algorithms for digital signal processing. Addison-Wesley Publishing Company; 1985.
[13] Bergland GD. Numerical analysis: a Fast Fourier Transform algorithm for real-valued series. Commun ACM 1968;11(10):703–10.
[14] Reviriego P, Maestro JA, Cervantes C. Reliability analysis of memories suffering multiple bit upsets. IEEE Trans Dev Mater Rel 2007;7(4):592–601.
[15] Sklar B. Digital communications: fundamentals and applications. Prentice Hall; 1988.