



Title	Low-Complexity Concurrent Error Detection for Convolution with Fast Fourier Transforms
Authors(s)	Bleakley, Chris J., Reviriego, P., Maestro, J.A.
Publication date	2011-06
Publication information	Bleakley, Chris J., P. Reviriego, and J.A. Maestro. "Low-Complexity Concurrent Error Detection for Convolution with Fast Fourier Transforms" 51, no. 6 (June, 2011).
Publisher	Elsevier
Item record/more information	http://hdl.handle.net/10197/7130
Publisher's statement	This is the author's version of a work that was accepted for publication in Low-Complexity Concurrent Error Detection for Convolution with Fast Fourier Transforms. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Low-Complexity Concurrent Error Detection for Convolution with Fast Fourier Transforms (VOL 51, ISSUE 6, (2011)) DOI: 10.1016/j.microrel.2011.02.010.
Publisher's version (DOI)	10.1016/j.microrel.2011.02.010

Downloaded 2024-05-29 02:22:01

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

Low Complexity Concurrent Error Detection for Fast Fourier Transform based Convolution

C.J. Bleakley, P. Reviriego, J.A. Maestro and A.B. O'Donnell

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 purely for CED is significantly reduced. When compared with a conventional Sum Of Squares – Dual Modular Redundancy technique, the proposal provides similar single error coverage at significantly reduced computational complexity. For complex input sequences, the proposal provides reductions in the number of real multiplications of 60% and 33% for adaptive and fixed filters, respectively. For real input sequences, the reductions are 70% and 59%, respectively.

Index Terms— Convolution, FFT, Soft Errors.

I. INTRODUCTION

DU E 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 modules are used and corrected when three identical modules are used [2]. The first configuration is known as Dual Modular Redundancy (DMR) and the last configuration is known as Triple Modular Redundancy (TMR) and is widely used for fault tolerance.

An alternative approach is the use of Algorithm Based Fault Tolerance (ABFT) [3] in which fault tolerance is incorporated into the algorithm at the system level. ABFT has been applied to the computation of the Fast Fourier Transform [3],[4]. For example in [3] the Sum Of Squares (SOS) technique was

proposed to detect errors in the FFT by computing the sums of the squares on the inputs and outputs. Based on Parseval's theorem, if no error has occurred then the SOSs should be equal [5]. Different ABFT approaches have been proposed for fault tolerant convolution. For example, in [6], cyclic error-correcting codes are used to implement fault tolerant convolution. The proposed approach works for direct implementation of convolution but not for transform based convolution. This is a major drawback as the cost of the direct implementation is, in most cases, much larger than that of transform based implementations. In [7],[8],[9] approaches based on the use of Residue Number Systems (RNS) for the computation are presented. While effective, these approaches typically incur significant area overhead and require specialized arithmetic units. The use of two independent convolutions with different transform lengths has been recently proposed in [10]. The approach uses recognition of error patterns in the convolution output to determine the module in error and perform correction.

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 method is compared to that of a conventional Sum Of Squares – Dual Modular Redundancy (SOS-DMR) approach [3].

The rest of the paper is structured as follows. Section II covers the background to the problem. Section III details the proposed CED technique and Section IV compares the performance of the technique to that of a conventional approach. The paper is concluded in Section V.

II. BACKGROUND

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

Manuscript received February 17th, 2010.

C. J. Bleakley and A. B. O'Donnell are with University College Dublin, Belfield, Dublin 4, Ireland (phone: +353-17165353; fax: +353-12695396; email: chris.bleakley@ucd.ie, anneodonnell@eircom.net).

P. Reviriego and J.A. Maestro are with Universidad Antonio de Nebrija, C/ Pirineos, 55 E-28040 Madrid, Spain (phone: +34-914521100; fax: +34-914521110; email: {previrie, jmaestro}@nebrija.es).

$$y(l) = \frac{1}{L} \sum_{n=0}^{L-1} h(n-l)x(n) \quad (1)$$

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

In common filtering applications, the input sequence $x(n)$ is long. Hence for practical implementation of the convolution, the sequence is segmented into blocks. Convolution is then performed on successive blocks and allowance is made of the overlap of data between blocks using, for example, the overlap-save method [5].

It is well known that for reasonable block lengths, the computational complexity of convolution can be reduced by means of the FFT using a frequency domain calculation. In fact, 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]:

$$\begin{aligned} r_{xx}(l) &= \frac{1}{N} \sum_{n=0}^{N-1} x_1(n)x_2((n-l) \bmod N) \\ &= \text{IDFT}[X_1(k) \cdot X_2(k)] \end{aligned} \quad (2)$$

Implementation of the technique is illustrated in Figure 1.

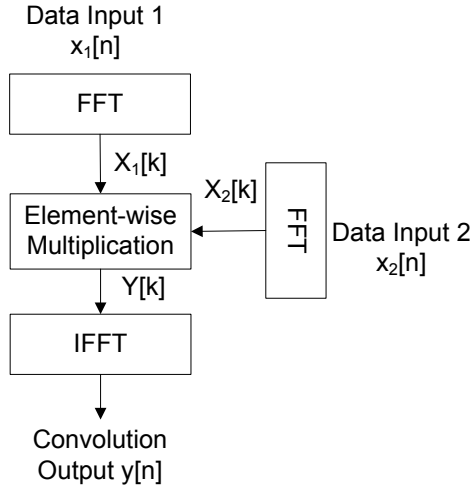


Figure 1: Implementation of FFT based convolution.

III. PROPOSED TECHNIQUE

Firstly, we consider the general case of complex input sequences, $x_1(n)$ 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:

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

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

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

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 frequency domain result $r_{xx}''(0)$. If they 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. Due to its greater computational complexity, it is more likely that that error will have occurred in 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. The first sample of the circular convolution calculated using the time domain method $r_{xx}'(0)$ requires N additional real multiplications and $N-1$ additional real additions.

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 IDFT 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:

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

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.

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

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 both 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 meet then an error has occurred, either in the convolution calculation or in the CED checkers. The proposed technique is illustrated in Figure 2.

It is known that complex multiplication can be implemented as 3 real multiplications and 5 real additions [11]. Based on this, the additional computational complexity of calculating the first point of the cross-correlation using the time-domain ($r_{xx}'(0), r_{yy}'(0)$) or frequency-domain approach ($r_{xx}''(0), r_{yy}''(0)$) is $3N$ real multiplies plus $7N-2$ real adds. Thus, the total computational complexity of the CED checks is $6N$ real multiplications and $14N-4$ real additions per output block.

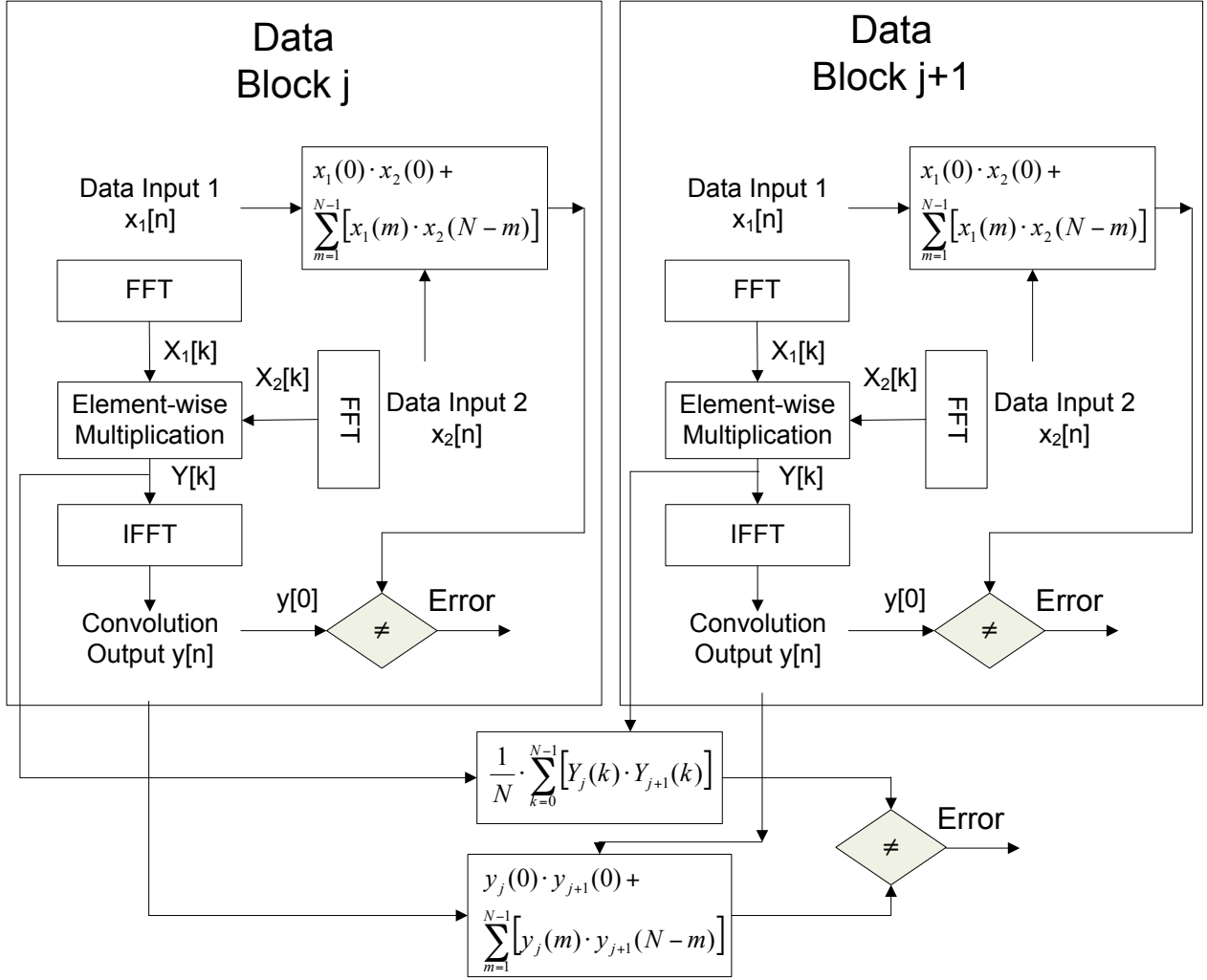


Figure 2: Implementation of FFT based convolution with the proposed CED technique.

In the case of real input sequences, the computational complexity of the overall convolution can be reduced by noting that the Discrete Fourier Transform (DFT) of a real sequence is symmetric [12]. If this optimization is employed, the proposed CED method must be augmented by adding DMR to calculation of the imaginary part of the element-wise multiplication, i.e. $\text{imag}(Y[k])$.

By taking advantage of symmetry and real data, and allowing for DMR, the computational complexity in the real case is $2.25N+1.5$ real multiplications and $3N+1$ 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:

$$\begin{aligned} & \left(|r_{xx}'(0) - r_{xx}''(0)| < \tau \right) \text{ and } \left(|r_{yy}'(0) - r_{yy}''(0)| < \tau \right) \text{ no error} \quad (7) \\ & \left(|r_{xx}'(0) - r_{xx}''(0)| \geq \tau \right) \text{ or } \left(|r_{yy}'(0) - r_{yy}''(0)| \geq \tau \right) \text{ error} \end{aligned}$$

This approach was considered in detail in [3] for a check for stand-alone FFTs using Parseval's theorem.

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 recomputed. Clearly 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 recomputed. 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 used to calculate $X_1(k)$ and $X_2(k)$ in the case of the first check and for the IFFTs used to calculate $y_j(n)$ and $y_{j+1}(n)$ in the case of the second check.

IV. EVALUATION

In this section the proposed technique is evaluated in terms of complexity and fault coverage.

A. Complexity

The computational complexity of the proposal can be compared with 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 I. 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 real input case, the number of operations needed for CED is reduced by taking advantage of the symmetry of the DFT and of real data.

B. Fault Coverage

Matlab simulations were run to test the impact of round off error and tolerance levels on the ability of the proposed and SOS-DMR schemes to detect errors. The schemes were applied to detect errors in convolutions with random data complex inputs, inserting a single random error in either the FFT, multiplication or IFFT stage at a random location. The input data and errors were uniformly distributed in the range -0.5 to 0.5 either real or imaginary. A tolerance level of 10^{-5} was used. The results for 100,000 simulations are provided in Table II. It can be observed that good fault coverage is achieved in both cases with the proposed technique detecting slightly more errors.

Table I: 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	$17N-6$	$14N-4$	19.7
Complex inputs, fixed filter	Mult	$9N$	$6N$	33.3
	Add	$13N-4$	$14N-4$	-7.7
Real inputs, adaptive filter	Mult	$7.5N+9$	$2.25N+1.5$	70.0
	Add	$8.5N+5$	$3N+1$	64.7
Real inputs, fixed filter	Mult	$5.5N+7$	$2.25N+1.5$	59.1
	Add	$6.5N+5$	$3N+1$	53.9

Table II: Fault coverage of the proposed technique for different 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

V. CONCLUSIONS

A novel Concurrent Error Detection technique for FFT-based implementations of convolution has been 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] Bauman, R., "Soft errors in advanced computer systems", IEEE Design and Test of Computers, vol. 22, no. 3, May - June 2005, pp. 258 - 266.
- [2] Nicolaidis, M., "Design for soft error mitigation", IEEE Trans. Device and Materials Reliability vol. 5, no. 3, Sept. 2005, pp. 405 - 418.
- [3] Reddy, A., and Banarjee, P., "Algorithm-based fault detection for signal processing applications", IEEE Trans. Computers, vol. 39, no. 10, Oct. 1990, pp. 1304 - 1308.
- [4] Jou, J., and Abraham, J.A., "Fault-tolerant FFT networks", IEEE Trans. Computers, vol. 37, no. 5, May 1988.
- [5] Oppenheim A. V. and Schaffer R. *Discrete Time Signal Processing*, Prentice Hall, 1999.
- [6] Redinbo, G.R., "System level reliability in convolution computations" IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, no. 8, Aug. 1989, pp.1241 - 1252.
- [7] Beckmann, P.E., and Musicus, B. R., "Fast fault-tolerant digital convolution using a polynomial residue number system" IEEE Trans. Signal Processing vol. 41, no. 7, July 1993, pp. 2300 - 2313.
- [8] Sundaram, S., and Hadjicostis, C.N., "Fault-Tolerant Convolution Via Chinese Remainder Codes Constructed From Non-Coprime Moduli", IEEE Trans. Signal Processing, vol. 56, no. 9, Sept. 2008, pp. 4244 - 4254.
- [9] O'Donnell, A. B., and Bleakley, C.J., "Area efficient fault tolerant convolution using RRNS with NTTs and WSCA", Electronics Letters, vol. 44, no. 10, May 2008, pp. 648 - 649.
- [10] P. Reviriego, J.A. Maestro, A. O'Donnell, C. Bleakley, "Soft error detection and correction for FFT based convolution using different block lengths", Proc. of IEEE International On-Line Testing Symposium (IOLTS'09), June 2009, pp. 138-143.
- [11] Blahut, R.E.: 'Fast Algorithms for Digital Signal Processing. Addison-Wesley Publishing Company, 1985.
- [12] Bergland, G.D.: 'Numerical Analysis: A fast Fourier transform algorithm for real-valued series', Communications of the ACM, vol. 11, no. 10, pp. 703-710,1968.