



Title	A Novel Concurrent Error Detection Technique for the Fast Fourier Transform
Authors(s)	Reviriego, P., Bleakley, Chris J., Maestro, J.A.
Publication date	2012-06
Publication information	Reviriego, P., Chris J. Bleakley, and J.A. Maestro. "A Novel Concurrent Error Detection Technique for the Fast Fourier Transform." The Institution of Engineering and Technology, 2012.
Conference details	23rd IET Irish Signals and Systems Conference, 28th - 29th June 2012, Maynooth
Publisher	The Institution of Engineering and Technology
Item record/more information	http://hdl.handle.net/10197/3860

Downloaded 2024-06-16 12:12:31

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

A Novel Concurrent Error Detection Technique for the Fast Fourier Transform

Pedro Reviriego*, Chris J. Bleakley, Juan Antonio Maestro***

*Universidad Antonio de Nebrija, C/ Pirineos, 55
E-28040 Madrid, Spain
email: {previrie, jmaestro}@nebrija.es

**UCD School of Computer Science and
Informatics, University College Dublin
email: chris.bleakley@ucd.ie

Abstract— A novel Concurrent Error Detection technique for the Fast Fourier Transform (FFT) is proposed in this paper. The technique is similar to the conventional Sum of Squares (SOS) approach but is of lower computational complexity. Complexity reduction is achieved by checking the FFTs of two data blocks in a single calculation. 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 blocks. In the case of error, the FFTs of both blocks must be recomputed. Assuming that errors are rare, this additional cost has negligible impact on the average number of operations per block.

Keywords – Soft Errors, Concurrent Error Detection.

I INTRODUCTION

Due to decreases in process geometry and operating voltage, soft errors are becoming an increasingly important problem jeopardizing the reliability of digital systems [1]. Digital Signal Processing (DSP) systems are among those affected by soft errors. A number of techniques have been proposed to protect DSP circuits. Algorithm level fault tolerant techniques rely on checking the numerical properties of the system inputs and outputs [2],[3],[4]. Due to its importance in DSP systems, several schemes have been proposed for detection of errors in the computation of the Fast Fourier Transform (FFT) [5],[6],[7].

One of the most commonly Concurrent Error Detection (CED) schemes for the FFT is the Sum of Squares (SOS) check based on Parseval's theorem [2]. The theorem states that the energy, or SOS, of an N point sequence, $x(n)$, and its Discrete Fourier Transform (DFT), $X(k)$, are equal:

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \cdot \sum_{k=0}^{N-1} |X(k)|^2 \quad (1)$$

This theorem is used to detect errors in an FFT computation by computing and comparing the

SOS of the inputs and outputs of the FFT. If they differ then it is assumed that an error occurred in the calculation of the FFT and the FFT is re-computed and re-checked. In [2], the coverage of the SOS method was analyzed in detail showing that it can detect most errors in the FFT. For a real sequence, N real multiplications are needed to perform the SOS of the input points and $N+2$ for the output points as the DFT of a real signal needs only $N/2+1$ values due to its symmetry [8]. Therefore $2*N + 2$ real multiplications are needed to implement the SOS check.

In many DSP applications, the FFT is computed for multiple consecutive data blocks corresponding to different time windows in a long sequence [8]. Hence two SOS calculations and one comparison must be performed for every block. This is illustrated in Figure 1.

In this work, a novel scheme for detecting errors in FFT implementations is introduced. The proposed method uses a single check for pairs of data blocks, thus reducing computational complexity.

Section II describes the proposal in detail. The effectiveness of the technique in the two-block case is considered in Section III. Section IV considers the extension of the proposal to the single-block case. Section V concludes the paper.

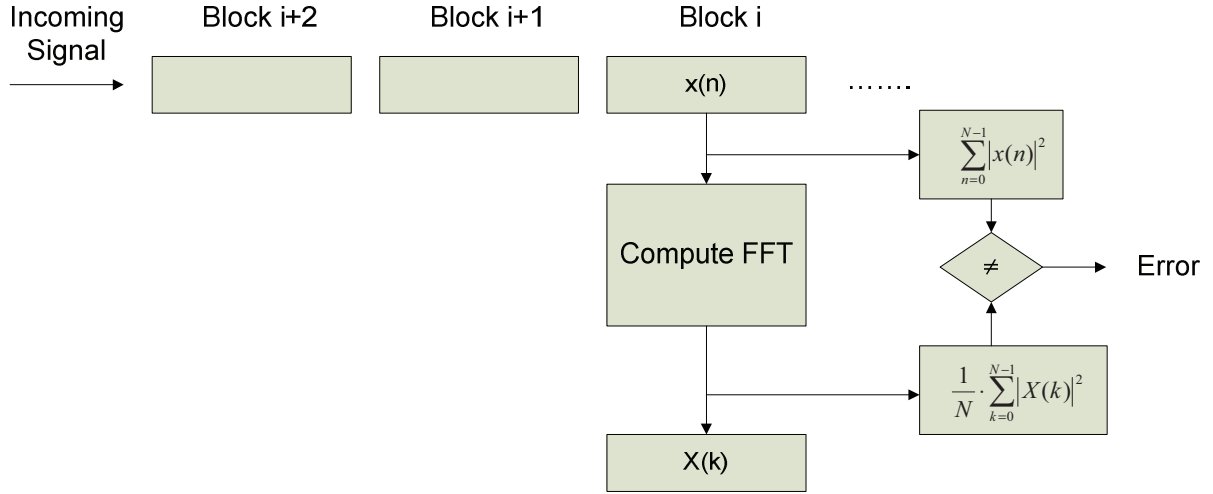


Figure 1: Implementation of the Sum Of Squares check on a long signal.

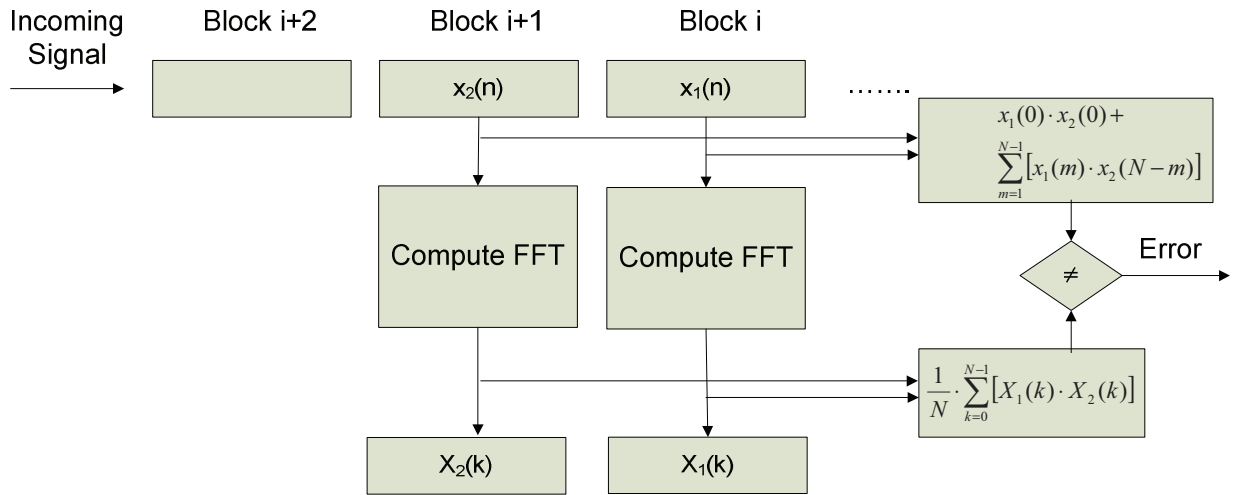


Figure 2: Implementation of the proposed technique on consecutive blocks of a long signal.

II PROPOSED TECHNIQUE

It is well known that 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 DFT of the multiplication of the DFTs, $X_1(k)$ and $X_2(k)$, of the original sequences [8], that is:

$$\begin{aligned} r(l) &= x_1(n) \otimes x_2(n) \\ &= IDFT[X_1(k) \cdot X_2(k)] \end{aligned} \quad (2)$$

where \otimes denotes N -point circular convolution and $IDFT[\cdot]$ denotes the Inverse DFT.

Herein, we propose that single errors in the calculation of $X_1(k)$ and $X_2(k)$ can be detected by comparing the first sample of the circular convolution, $r(0)$, as calculated in the time domain with that obtained using a frequency domain calculation.

In the time domain, $r(0)$ can be calculated as:

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

Using the frequency domain, $r(0)$ can be calculated as:

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

since the first element of the Inverse DFT output is simply the mean of the input sequence.

A single error in the calculation of $X_1(k)$ or $X_2(k)$ will cause the condition $r'(0) = r''(0)$ to be false.

The implementation of the proposed technique on an incoming signal is illustrated in Figure 2.

Since multiplication is the most complex arithmetic operation used in computation of the FFT, the number of multiplications will be used to compare the implementation cost of the proposed technique with that of the SOS check. For real input sequences, the proposed technique requires N real multiplications and $N/2+1$ complex multiplications. As complex multiplications can be implemented with only three real multiplications [9], the total number of multiplications required for the proposed check is $2.5 \cdot N + 3$. However, these operations check two input blocks in a single calculation. So the number of real multiplications per input data block is $1.25 \cdot N + 1.5$. This compares with the $2 \cdot N + 2$ real multiplications needed to perform the SOS check.

The price paid for the reduced number of operations is that when an error is detected, the proposed method is not able to determine which FFT suffered the error. It may have occurred in the computation of either FFT, i.e. $X_1(k)$ or $X_2(k)$. Therefore both FFTs must be recomputed. This increases the number of operations required relative to the SOS check since, in the SOS case, only one block has to be re-computed. However, if we assume that errors are a rare event then the average number of additional operations per block is negligible. For example, if errors occur in 0.01% of FFT calculations then 0.02% of checks will detect an error, requiring an additional 0.02% of FFTs to be re-calculated.

If sufficient memory is available then the computational overhead can be reduced by comparison of the outputs of the first re-calculated FFT with the original results. If the outputs differ, and errors are rare events, then it can be assumed that a single error occurred in the original calculation of the first FFT and that there was no error in the original calculation of the second FFT. Based on this, the system can proceed without re-calculating the second FFT. In the example, this would reduce the additional re-calculations to 0.015% of FFTs.

As with the SOS check, the proposed technique has to deal with round-off errors in FFT implementation. This can be done by using a tolerance level τ in the check, such that small differences do not trigger an error:

$$\begin{cases} |r'(0) - r''(0)| < \tau & \text{no error} \\ |r'(0) - r''(0)| \geq \tau & \text{error} \end{cases} \quad (5)$$

This approach was considered in detail for the SOS check in [2].

III TWO BLOCK EVALUATION

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

a) Fault Coverage

Matlab simulations were run to test the impact of round off error and tolerance levels on the proposed scheme. The scheme was used to detect errors in FFTs with random data inputs. Soft errors were simulated by inserting a single random error in each FFT pair at a random location. The input data and errors were uniformly distributed in the range -0.5 to 0.5. A tolerance level of 10^{-5} was used.

The results for 100,000 simulations are provided in Table I for different FFT block lengths. It can be observed that good fault coverage is achieved in all cases. The same simulations were performed using the SOS check and the results are shown in Table II. It can be observed that also good fault coverage is achieved although slightly lower than for the proposed technique.

Table 1: Fault coverage of the proposed technique for different FFT lengths.

N	Fault Coverage (%)
64	99.93
128	99.91
256	99.90
512	99.86
1024	99.85

Table 2: Fault coverage of the SOS technique for different FFT lengths.

N	Fault Coverage (%)
64	99.72
128	99.70
256	99.68
512	99.67
1024	99.59

b) Complexity

As discussed before, the proposed technique requires less real multiplications to detect errors than the traditional SOS approach, but it requires more operations to correct errors since both FFTs have to be recomputed. To take this into account, we can assume that the probability of an error occurring in an FFT calculation is f . The average number of multiplications per block can be computed for both techniques as:

$$N_{mult}|_{SOS} \cong 2 \cdot N + f \cdot 3 \cdot \frac{N}{2} \cdot \log_2(N) \quad (6)$$

$$N_{mult}|_{Proposed} \cong 1.25 \cdot N + f \cdot 3 \cdot N \cdot \log_2(N) \quad (7)$$

In Figure 3, the equations are evaluated for $N=1024$ and various values of f . It can be observed that when the probability of error falls below 4% the proposed technique is more efficient than SOS. In typical applications, the probability of faults in a single block calculation is very low and the proposed technique would typically only require $1.25/2 \approx 0.63$ times the number of multiplications needed in SOS check.

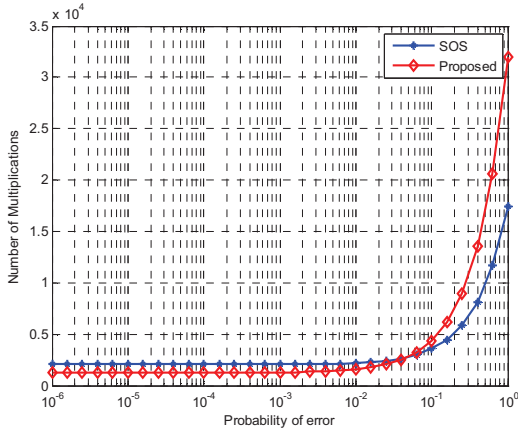


Figure 3: Comparison of the number of multiplications per block.

IV EXTENSION TO THE SINGLE BLOCK CASE

The proposed technique relies on the computation of the FFT over multiple data blocks. However it can be extended to cover computation of a single block.

The FFT is commonly implemented using the Decimation in Time (DIT) or the Decimation in Frequency (DIF) algorithms [8]. In both cases, computation of an N point FFT is decomposed into computation of two $N/2$ point FFTs that are then combined. Implementations of these schemes are illustrated in Figures 4 and 5 for the 8-point case.

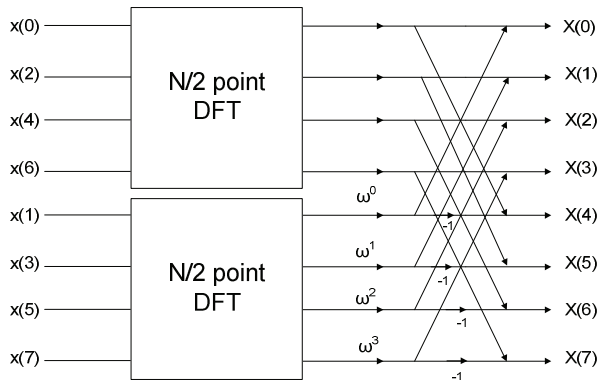


Figure 4: Decimation in Time (DIT) computation of an N point FFT.

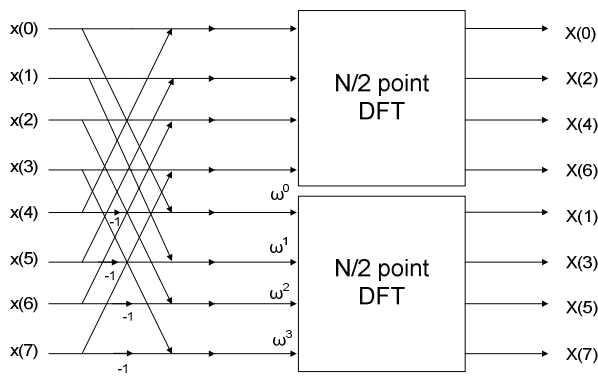


Figure 5: Decimation in Frequency (DIF) computation of an N point FFT.

These structures can be used in conjunction with the proposed checker. In effect, each $N/2$ -point FFT is treated as a block. This ensures that most errors occurring in these sub-blocks are detected. However, errors in the $N/2$ multiplications (ω^0 , ω^1 , ω^2 , ...), outside these blocks, are not detected. To detect these errors, the multiplications must be duplicated and the outputs compared. This incurs an additional cost that makes the technique less attractive. In the case of the DIT computation $N/2$ additional complex multiplications are needed resulting in a cost that exceeds that of the SOS technique. The same occurs for the DIF case where in addition to the $N/2$ multiplications, the $N/2$ point DFT at the bottom has now complex inputs which also increases the cost. The proposed technique can be attractive for a single block when partial coverage for error detection is acceptable and the priority is to minimize the additional cost. In that case, the multiplications can be left unprotected and the total cost would be 0.63 that of the SOS check but with a reduced coverage for error detection.

V CONCLUSIONS

A novel Concurrent Error Detection technique for FFT implementations has been proposed. The technique is similar to the Sum of Squares (SOS) approach but can be implemented more efficiently. More precisely, less than 63% of the multiplications needed to implement the SOS check are sufficient to implement the proposed technique when it is applied to multiple blocks. When applied to FFT computation of a single block the proposed technique reduces the cost at the expense of lower fault coverage. These cost savings make the proposed scheme an interesting alternative to the SOS check to detect errors in the computation of the FFT.

REFERENCES

- [1] Baumann, R., "Soft errors in advanced computer systems", IEEE Design & Test of Computers, Volume 22, Issue 3, May - June 2005, pp. 258 – 266.
- [2] Reddy A. and Banarjee P., "Algorithm-based fault detection for signal processing applications", IEEE Transactions on Computers, Volume 39, Issue 10, Oct. 1990, pp. 1304-1308.
- [3] Huang Y.-H., "High-Efficiency Soft-Error-Tolerant Digital Signal Processing Using Fine-Grain Subword-Detection Processing", IEEE Trans. Very Large Scale Integration Systems, vol. 18, no 2, pp. 291-304, Feb. 2010.
- [4] 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, pp. 4244-4254, Sept. 2008.

[5] Tao D.L. and Hartmann, C.R.P., "A novel concurrent error detection scheme for FFT networks", IEEE Transactions on Parallel and Distributed Systems, vol. 4, no. 2, Feb 1993, pp. 198-221.

[6] Lonmardi, F. and Muzio, J.C., "Concurrent error detection and fault location in an FFT architecture", IEEE Journal of Solid-State Circuits, vol. 27, no. 5, May 1992, pp. 728-736.

[7] Oh, C.G. and Youn, H.Y., "On concurrent error detection, location, and correction of FFT networks", The Twenty-Third International Symposium on Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers, June 1993, pp.596-605.

[8] Oppenheim A. V. and Schafer R. Discrete Time Signal Processing, Prentice Hall 1999.

[9] R.E. Blahut, Fast Algorithms for Digital Signal Processing, Addison-Wesley Publishing Company, 1985..