ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design and Analysis of Fault Tolerant Parallel FFTs Based on Error Correction Codes and Perseval Checks

Kummara Theja
PG Scholar,
Electronics and Communication Engineering,
Intell Engineering College, AP, India

P.Deepthi Jordhana
Associate Professor
Electronics and Communication Engineering,
Intell Engineering College, AP, India

Abstract:
The complexity of communications and signal proc esssing circuits increases every year. This is made possible by the CMOS technology scaling that enables the integration of more and more transi stors on a single device. This increased complexity makes the circuits more vulnerable to errors. At the same time, the scaling means that transistors operate with lower voltages and are more susc eptible to errors caused by noise and manufact uring variations. Soft errors pose a reliability threat to modern electronic circuits. This makes protection against soft errors a requirement for many applications. Communications and signal processing systems are no exceptions to this trend. For some applications, an interesting option is to use algorithmic-based fault tolerance (ABFT) techn iques that try to exploit the algorithmic properties to detect and correct errors. Signal processing and communication applications are well suited for ABFT. One example is fast Fourier transforms (FFTs) that are a key building block in many systems. Several protection schemes have been proposed to detect and correct errors in FFTs. Among those, probably the use of the Perseval or sum of squares check is the most widely known. In modern communication systems, it is increasingly common to find several blocks operating in parallel. Recently, a technique that exploits this fact to implement fault tolerance on parallel filters has been proposed. In this brief, this technique is first applied to protect FFTs. Then, two improved protection schemes that combine the use of error correction codes and Perceval checks are proposed and evaluated.

1 Introduction:

Error correction code (ECC) techniques have been widely used to correct transient errors and improve the reliability of memories. ECC words in memories consist of data bits and additional check bits because the ECCs used in memories are typically from a class of linear block codes. During the write operations of memories, data bits are written in data bit arrays, and check bits are concurrently produced using the data bits and stored in check bit arrays. The check bit arrays, just like the data bit arrays, should be tested prudently for the same fault models if reliable error correction is to be insured[1].

Fast Fourier transform is used to convert a signal from time domain to frequency & this is needed so that you can view the frequency components present in a signals. If you know the frequency components present in a signals you can play with the signals :) Let's say, u want to design a low pass filter and want to decide on the cut off frequency of the filter. If you have the frequency domain details for a signals u can clearly identify the frequency components which u want to retain & the ones which u want to take out[2].

Environmental interference and physical defects in the communication medium can cause random bit errors during data transmission. Error coding is a method of detecting and correcting these errors to ensure information is transferred intact from its source to its destination. Error coding is used for fault tolerant computing in computer memory, magnetic and optical data storage media, satellite and deep space commu nications, network communications, cellular telephone networks, and almost any other form of digital data communication. Error coding uses mathematical formulas to encode data bits at the source into longer bit words for transmission. The "code word" can then be decoded at the destination to retrieve the infor mation. The extra bits in the code word provide redundancy that, according to the coding scheme used, will allow the destination to use the decoding process to determine if the communication medium introduced

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

errors and in some cases correct them so that the data need not be retransmitted. Different error coding schemes are chosen depending on the types of errors expected, the communication medium's expected error rate, and whether or not data retransmission is possible. Faster processors and better communications technology make more complex coding schemes, with better error detecting and correcting capabilities, possible for smaller embedded systems, allowing for more robust communications. However, tradeoffs between bandwidth and coding overhead, coding complexity and allowable coding delay between transmissions, must be considered for each application. Transient errors can often upset more than one bit producing multi-bit errors with a very high probability of error occurrence in neighboring memory cells . Bit interleaving is one technique to remedy multi-bit errors in neighboring memory cells as physically adjacent bits in memory array are assigned to different logical words [5],[6]. The single-error-correction, double-error-detection, and double-adjacent-error-correction (SEC-DED-DAEC) codes have previously been presented to correct adjacent double bit errors [4]-[7]. The required number of check bits for the SEC-DED-DAEC codes is the same as that for the SEC-DED codes.

In addition, the area and timing overheads for encoder and decoder of the SEC-DED-DAEC codes are similar to those of the SEC-DED codes. Consequently, adjacent double bit errors can be remedied with very little additional cost using the SECDED-DAEC codes. The SEC-DED-DAEC codes may be an attractive alternative to bit interleaving in providing greater flexibility for optimizing the memory layout. Furthermore, the SEC-DED-DAEC code can be used in conjunction with bit interleaving and this method can efficiently deal with adjacent multi-bit errors [1]

The FFTs in parallel increases the scope of applying error correction codes together. Generating parity together for parallel FFTs also helps in minimizing the complexity in some ECC [15]. By assuming that there can only be a single error on the system in the case of radiation-induced soft errors and may be two in worst case. The proposed new technique is based on the combination of Partial Summation combined with parity FFT for multiple error correction.

Fast Fourier Transform

Fast Fourier Transform (FFT) algorithm converts a signal from time domain into a sequence in the frequency domain [13]. Fast Fourier transforms are widely used for many applications which include engineering, science, and mathematics. It computes transformations through DFT matrix. The FFT operation starts with decomposing N-point time domain signal and calculating N frequency spectra and finally forming a single spectrum.

The Discrete Fourier Transform (DFT)

Discrete Fourier Transform (DFT) is an important unit in many communication applications like OFDM, etc. DFT is also measured as one of the tools to act upon frequency analysis of discrete time signals. The Discrete Fourier Transform is a continuous Fourier transform for the use of discrete functions. Given a real sequence as the input, the DFT outputs them as a sequence of complex numbers. The mathematical representation of the transform is

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}, n = 0,1, \ldots\ldots N-1 \qquad (1)$$

If an N – point DFT is implemented directly, the necessity of arithmetic units is of the order of O(N2) that is N2 multiplications and N (N-1) additions. Thus FFT is used for designing the DFT. Depending on inputs being real or complex, the design of adders and multipliers are formed.

Divide and Conquer Approach to Computation of the DFT

The reduction of computational complexity algorithm for DFT is made possible by using a divide and conquers approach. This approach decomposes a larger DFT into smaller one forming a collective FFT algorithm. Let us consider N-point DFT. It is one of the well-organized ways to implement Discrete Fourier Transform (DFT) due to its compact use of arithmetic

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

blocks. The FFT and inverse FFT of an N point signals are given below.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad (2)$$

$$x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(k) W_N^{-kn} \qquad (3)$$

Where

$$W_N^{kn} = e^{-j\frac{2\pi}{N}kn}$$

From the above equation it is evident that both FFT and its inverse are almost same with small variation. Inverse FFT design is formed by using divide by N-point and taking the conjugate of the twiddle factors. Due to the frequent use of FFT in modern wireless designs, higher radix FFTs such as radix-4, radix-8, radix-2k, split radix, etc. are designed for improving the performance timing and reducing the complexity. The difference between these designs is based on their butterfly units.

Basic concept of 4-points DIF FFT circuit which applies Radix-2 architecture is in Figure 1 shows calculation signal flow graph about discrete Fourier coefficient of N=4. Here W40, W41 are the twiddle factors of the four point Fast Fourier Transform. Note, under the arrow of is subtraction; twiddle factor at the top of line is multiplication. In butterfly processing element that is shown red line make be corresponding to next block in the slide diagram. Another butterfly processing element colors are same. This is the 4 points FFT circuit for parallel input.

Generally, FFT analyzes an input signal sequence by using decimation-in-frequency (DIF) or decimation-in-time (DIT) decomposition to design an efficient signal-flow graph (SFG). Here, the paper work focuses DIF decomposition because it matches with various pipelined designs. x(0), x(1), x(2) and x(3) are the input time domain signals with 1, -1 and –j as the twiddle factors producing X(0), X(1), X(2) and X(3) as the frequency domain outputs.

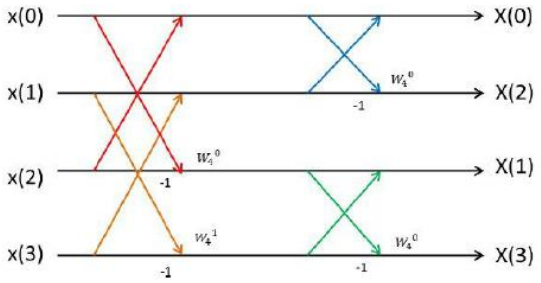

Figure 1. Signal Flow Graph of 4-point FFT

2. Error Tolerant Techniques for Parallel FFTs

Error Correction based on Hamming Codes

The aim of error tolerant design is to protect parallel FFTs from errors. Various schemes have been proposed for error detection and correction in FFTs. One of the basic and simple methods is error correction using hamming codes. Unlike parity code which can detect only odd bit error, the hamming code can detect two bit errors and correct one error. Similar to other error correction codes, hamming codes also utilizes the parity bit which is generated for the corresponding input sequence for detecting errors [14]. It achieves higher code rate with minimum distance of three. The number of parity bits depends on the total number of data bits. For example, hamming code with 4 information bits produces 7 encoded data bits with its difference being the parity. In this case, the three parity bits h1, h2, h3 are computed as a data bits c1, c2, c3, c4 as described below:

$$h1 = c1 \oplus c2 \oplus c3 \qquad (4)$$

$$h2 = c1 \oplus c2 \oplus c4 \qquad (5)$$

$$h3 = c1 \oplus c3 \oplus c4 \qquad (6)$$

The limitations of is that, during the multiple error scenario hamming code will not be able to exactly identify the individual FFTs with error.

Fault tolerant FFT based on Parseval's check

Parseval's method is one of the techniques to detect errors parallel in multiple FFT. This is achieved with Sum of Squares (SOSs) check [5] based on Parseval's theorem. The error free FFT should have its Sum of Squares of the input equaling the Sum of Squares of its frequency domain output. This correlation can be used

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

to identify errors with minimum overhead. For parallel FFTs, the Parseval's check can be combined with the error correction codes to minimize the area overhead.
Multiple error detection and correction is achieved through this combination. One of the easy ways is to generate the redundant input for single FFT with all the four FFT inputs. To correct error the parity FFT output is XORed with fault free outputs of the FFTs. Compared to the previous schemes presented in the Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks [1], this technique reduced the total number of Sum of Squares used. Another existing work done is by combining SOS checks with hamming codes instead of using Parseval's check individually as shown in Figure 2.

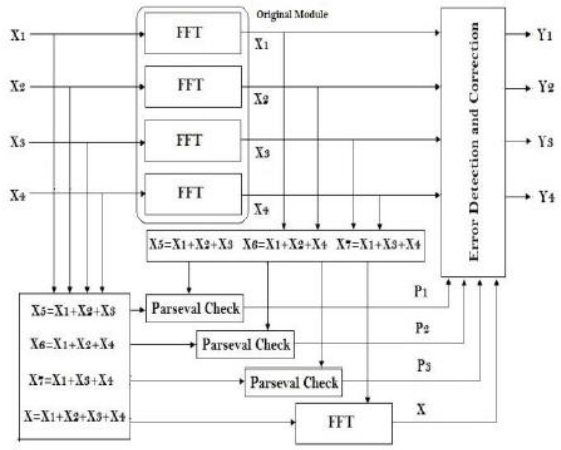

Figure 2. Parity-SOS-ECC fault-tolerant parallel FFTs
This method combines the feature of parity calculation of hamming codes and error detection process of Sum of Squares. Concurrent Error Detection (CED) schemes for the FFT are the Sum of Squares (SOS) check based on Pa theorem. The use of parseval check is exponentially reduced to the direct comparisons of FFTs inputs and outputs used to protect parallel FFTs

3. PROPOSED PROTECTION SCHEMES FOR PARALLEL FFTS

The starting point for our work is the protection scheme based on the use of ECCs that was presented in [17] for digital filters. This scheme is shown in Fig. 1. In this example, a simple single error correction Hamming code [18] is used. The original system consists of four FFT modules and three redundant modules is added to detect and correct errors. The inputs to the three redundant modules are linear combinations of the inputs and they are used to check linear combinations of the outputs. For example, the input to the first redundant module is

$$x_5 = x_1 + x_2 + x_3 \qquad (1)$$

and since the DFT is a linear operation, its output $z5$ can be used to check that

$$z_5 = z_1 + z_2 + z_3. \qquad (2)$$

This will be denoted as $c1$ check. The same reasoning applies to the other two redundant modules that will provide checks $c2$ and $c3$. Based on the differences observed on each of the checks, the module on which the error has occurred can be determined. The different patterns and the corresponding errors are summarized in Table I.
Once the module in error is known, the error can be corrected by reconstructing its output using the remaining modules. For example, for an error affecting $z1$, this can be done as follows:

$$z_{1c}[n] = z_5[n] - z_2[n] - z_3[n]. \qquad (3)$$

Similar correction equations can be used to correct errors on the other modules. More advanced ECCs can be used to correct errors on multiple modules if that is needed in a given application. The overhead of this technique, as discussed in [17], is lower than TMR as the number of redundant FFTs is related to the logarithm of the number of original FFTs. For example, to protect four FFTs, three redundant FFTs are needed, but to protect eleven, the number of redundant FFTs in only four. This shows how the overhead decreases with the number of FFTs.

$c_1\,c_2\,c_3$	Error Bit Position
0 0 0	No error
1 1 1	z_1
1 1 0	z_2
1 0 1	z_3
0 1 1	z_4
1 0 0	z_5
0 1 0	z_6
0 0 1	z_7

TABLE I ERROR LOCATION IN THE HAMMING CODE

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

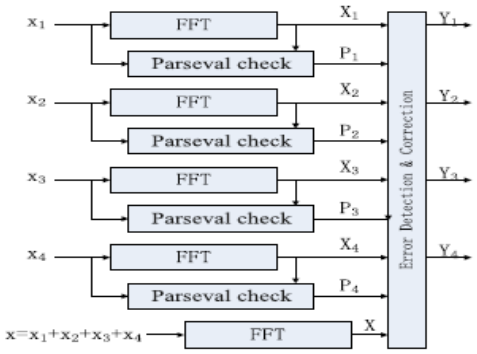

Fig. 3. Parity-SOS (first technique) fault-tolerant parallel FFTs.

In Section I, it has been mentioned that over the years, many techniques have been proposed to protect the FFT. One of them is the Sum of Squares (SOSs) check [4] that can be used to detect errors. The SOS check is based on the Parseval theorem that states that the SOSs of the inputs to the FFT are equal to the SOSs of the outputs of the FFT except for a scaling factor. This relationship can be used to detect errors with low overhead as one multiplication is needed for each input or output sample (two multiplications and adders for SOS per sample). For parallel FFTs, the SOS check can be combined with the ECC approach to reduce the protection overhead. Since the SOS check can only detect errors, the ECC part should be able to implement the correction. This can be done using the equivalent of a simple parity bit for all the FFTs. In addition, the SOS check is used on each FFT to detect errors. When an error is detected, the output of the parity FFT can be used to correct the error. This is better explained with an example. In Fig. 2, the first proposed scheme is illustrated for the case of four parallel FFTs. A redundant (the parity) FFT is added that has the sum of the inputs to the original FFTs as input. An SOS check is also added to each original FFT. In case an error is detected (using $P1$, $P2$, $P3$, $P4$), the correction can be done by recomputing the FFT in error using the output of the parity FFT (X) and the rest of the FFT outputs. For example, if an error occurs in the first FFT, $P1$ will be set and the error can be corrected by doing

$$X_{1c} = X - X_2 - X_3 - X_4. \quad (4)$$

This combination of a parity FFT and the SOS check reduces the number of additional FFTs to just one and may, therefore, reduce the protection overhead. In the following, this scheme will be referred to as parity-SOS (or first proposed technique). Another possibility to combine the SOS check and the ECC approach is instead of using an SOS check per FFT, use an ECC for the SOS checks. Then as in the parity-SOS scheme, an additional parity FFT is used to correct the errors. This second technique is shown in Fig. 3. The main benefit over the first parity- SOS scheme is to reduce the number of SOS checks needed. The error location process is the same as for the ECC scheme in Fig. 1 and correction is as in the parity-SOS scheme. In the following, this scheme will be referred to as parity-SOS-ECC (or second proposed technique).The overheads of the two proposed schemes can be initially estimated using the number of additional FFTs and SOS check blocks needed.

In all the techniques discussed, soft errors can also affect the elements added for protection. For the ECC technique, the protection of these elements was discussed in [17]. In the case of the redundant or parity FFTs, an error will have no effect as it will not propagate to the data outputs and will not trigger a correction. In the case of SOS checks, an error will trigger a correction when actually there is no error on the FFT. This will cause an unnecessary correction but will also produce the correct result. Finally, errors on the detection and correction blocks in Figs. 2 and 3 can propagate errors to the outputs. In our implementa tions, those blocks are protected with TMR. The same applies for the adders used to compute the inputs to the redundant FFTs in Fig. 1 or to the SOS checks in Fig. 3. The triplication of these blocks has a small impact on circuit complexity as they are much simpler than the FFT computations. A final observation is that the ECC scheme can detect all errors that exceed a given threshold (given by the quantization used to implement the FFTs) [17]. On the other hand, the SOS check detects most errors but does not guarantee the detection of all errors [4]. Therefore, to compare the three techniques for a given implementation, fault injection experiments should be done to determine the

percentage of errors that are actually corrected. This means that an evaluation has to be done both in terms of overhead and error coverage.

5. CONCLUSIONS

Detecting and correcting errors such as critical reliability are difficult in signal processing which increases the use of fault tolerant implementation. In modern signal processing circuits, it is common to find several filters operating in parallel. Proposed is an area efficient technique to detect and correct single errors. This brief has presented a new scheme to protect parallel FFT using cordic that is commonly found in modern signal processing circuits.

The approach is based on applying SOS-ECC check to the parallel FFT outputs to detect and correct errors. The SOS checks are used to detect and locate the errors and a simple parity FFT is used for correction. The 8 point FFT with the input bit length 32 is protected using the proposed technique. . The detection and location of the errors can be done using an SOS check per FFT or alternatively using a set of SOS checks that form an ECC. This technique can detect and correct only single bit error and it reduces area results in high speed compared to existing techniques.

6. FUTURE WORK

In Future, utilization of DCT instead of FFT will be carried out. Since SOS-ECC technique can detect and correct only single bit fault, this will be extended to multi bit faults by using the trellis code and hence area will be further reduced.

REFERENCES

[1] R. Baumann. 2005. Soft errors in advanced computer systems. IEEE Des. Test Comput. 22(3): 258-266.

[2] M. Ergen. 2009. Mobile Broadband-Including WiMAX and LTE. New York, NY, USA: Springer-Verlag.

[3] Z. Gao *et al*. 2015. Fault tolerant parallel filters based on error correction codes. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 23(2): 384-387.

[4] R. W. Hamming. 1950. Error detecting and error correcting codes. Bell Syst. Tech. J. 29(2): 147-160.

[5] T. Hitana and A. K. Deb. 2004. Bridging concurrent and non-concurrent error detection in FIR filters. in Proc. Norchip Conf. pp. 75-78.

[6] J. Y. Jou and J. A. Abraham. 1988. Fault-tolerant FFT networks. IEEE Trans. Comput. 37(5): 548-561.

[7] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu. 2010. Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances. New York, NY, USA: Springer-Verlag.

[8] E. P. Kim and N. R. Shanbhag. 2012. Soft N-modular redundancy. IEEE Trans. Comput. 61(3): 323-336.

[9] M. Nicolaidis. 2005. Design for soft error mitigation. IEEE Trans. Device Mater. Rel. 5(3): 405-418.

[10] S. Pontarelli, G. C. Cardarilli, M. Re and A. Salsano. 2008. Totally fault tolerant RNS based FIR filters. In Proc. 14th IEEE Int. On-Line TestSymp. (IOLTS). pp. 192-194.

[11] A. L. N. Reddy and P. Banerjee. 1990. Algorithmbased fault detection for signal processing applications. IEEE Trans. Comput. 39(10): 1304-1308.

[12] P. Reviriego, C. J. Bleakley and J. A. Maestro. 2012. A novel concurrent error detection technique for the fast Fourier transform. In Proc. ISSC, Maynooth, Ireland. pp. 1-5.

[13] P. Reviriego, S. Pontarelli, C. J. Bleakley and J. A. Maestro. 2012. Area efficient concurrent error detection and correction for parallel filters. IET Electron. Lett. 48(20): 1258-1260.