

## An Efficient Error Correction Technique for Parallel FFTs

M. THARANI

PG Scholar, Dept of ECE (DECS), KSRMCE, Kadapa, AP, India.

**Abstract:** The complexity of communications and signal processing circuits increases every year. This is made possible by the CMOS technology scaling that enables the integration of more and more transistors 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 susceptible to errors caused by noise and manufacturing 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) techniques 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 Parseval or sum of squares checks 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 Parseval checks are proposed and evaluated.

**Keywords:** Error Correction Codes (ECCS), Fast Fourier Transforms (FFTS), Soft Errors.

### I. 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 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. 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, you 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 you can clearly identify the frequency components which you want to retain & the ones which you want to take out. 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 communications, 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 information. 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 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. 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. 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. 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. 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.

## II. PREVIOUS WORK

Signal processing and communications circuits are well suited for ABFT as they have regular structures and many algorithmic properties. Over the years, many ABFT techniques have been proposed to protect the basic blocks that are commonly used in those circuits. Several works have considered the protection of digital filters. For example, the use of replication using reduced precision copies of the filter has been proposed as an alternative to TMR but with a lower cost. The knowledge of the distribution of the filter output has also been recently exploited to detect and correct errors with lower overheads. The protection of fast Fourier transforms (FFTs) has also been widely studied. As signal-processing circuits become more complex, it is common to find several filters or FFTs operating in parallel. This occurs for example in filter banks or in multiple-input multiple-output (MIMO) communication systems. In particular, MIMO orthogonal frequency division modulation (MIMO-OFDM) systems use parallel iFFTs/FFTs for modulation/demodulation. MIMO-OFDM is implemented on long-term evolution mobile systems and also on WiMax. The presence of parallel filters or FFTs creates an opportunity to implement ABFT techniques for the entire group of parallel modules instead of for each one independently. This has been studied for digital filters initially in where two filters were considered. More recently, a general scheme based on the use of error correction codes (ECCs) has been proposed. In this technique, the idea is that each filter can be the equivalent of a bit in an ECC and parity check bits can be computed using addition. This technique can be used for operations, in which the output of the sum of several inputs is the sum of the individual outputs. This is true for any linear operation as, for example, the discrete Fourier transform (DFT).

## III. 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. 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  $h_1$ ,  $h_2$ ,  $h_3$  are computed as a data bits  $c_1$ ,  $c_2$ ,  $c_3$ ,  $c_4$  as described below:

$$\begin{aligned} h_1 &= c_1 \oplus c_2 \oplus c_3 \\ h_2 &= c_1 \oplus c_2 \oplus c_4 \\ h_3 &= c_1 \oplus c_3 \oplus c_4 \end{aligned}$$

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

### A. Fault tolerant FFT based on Parseval's check

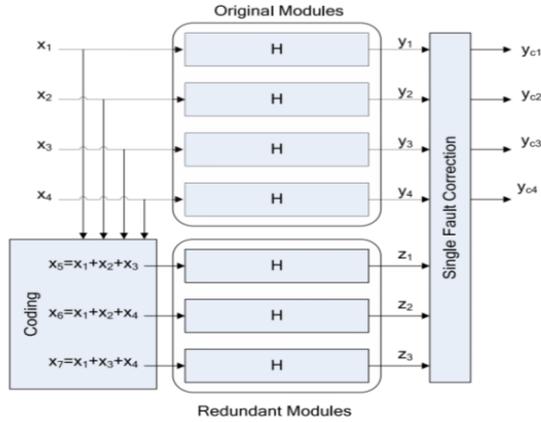
Parseval's method is one of the techniques to detect errors parallel in multiple FFT. This is achieved with Sum of Squares (SOSs) check based on Parseval's theorem. The error free FFT should have its Sum of Squares of the input equaling the Sum of Squares of its frequency domain output. This correlation can be used 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, 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. 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 Parsavel's theorem. The use of parseval check is exponentially reduced to the direct comparisons of FFTs input and outputs used to protect parallel FFTs

## IV. 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 for digital filters. This scheme is shown in Fig1. In this example, a simple single error correction Hamming code is used. The

## An Efficient Error Correction Technique for Parallel FFTs

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 and since the DFT is a linear operation, its output  $z_5$  can be used to check.



**Fig 1. ECC approach**

$$X_5 = x_1 \oplus x_2 \oplus x_3$$

$$Z_5 = z_1 \oplus z_2 \oplus z_3$$

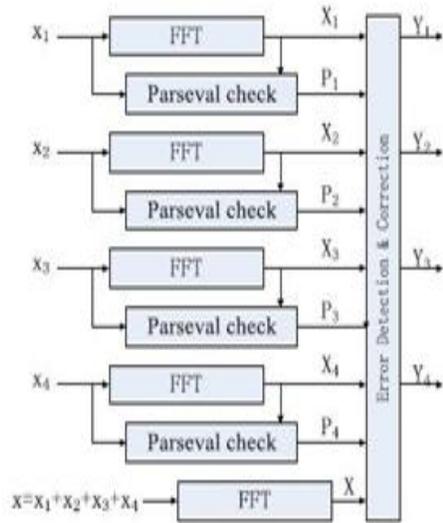
This will be denoted as  $c_1$  check. The same reasoning applies to the other two redundant modules that will provide checks  $c_2$  and  $c_3$ . Based on the differences observed on each of the checks, the module on which the error has occurred can be determined. The different patterns and the corresponding 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  $z_1$ , this can be done as follows:

$$z_{1c}[n] = z_5[n] - z_2[n] - z_3[n]. \tag{1}$$

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 is only four. This shows how the overhead decreases with the number of FFTs.

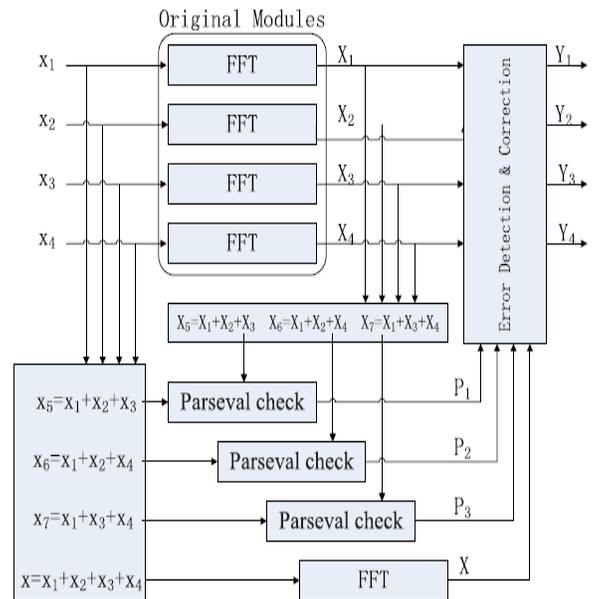
**Table I. Error Location In The Hamming Code**

$c_1 c_2 c_3$	Error Bit Position
000	No error
111	$z_1$
110	$z_2$
101	$z_3$
011	$z_4$
100	$z_5$
010	$z_6$
001	$z_7$



**Fig 2. 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 (SOS) check 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 adds for SOS per sample).



**Fig 3. Parity-SOS-ECC approach**

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. 3, 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

$$X1c = X - X2 - X3 - X4$$

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 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 implementations, 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). On the other hand, the SOS check detects most errors but does not guarantee the detection of all errors. 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.

V. RESULTS

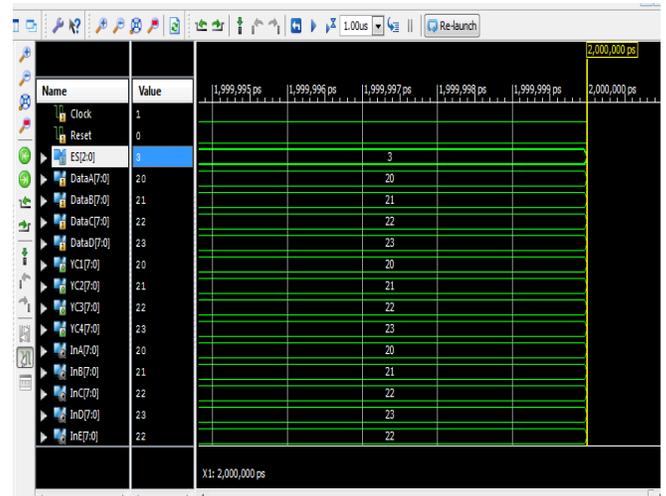


Fig 4. Simulation results for ECCs approach

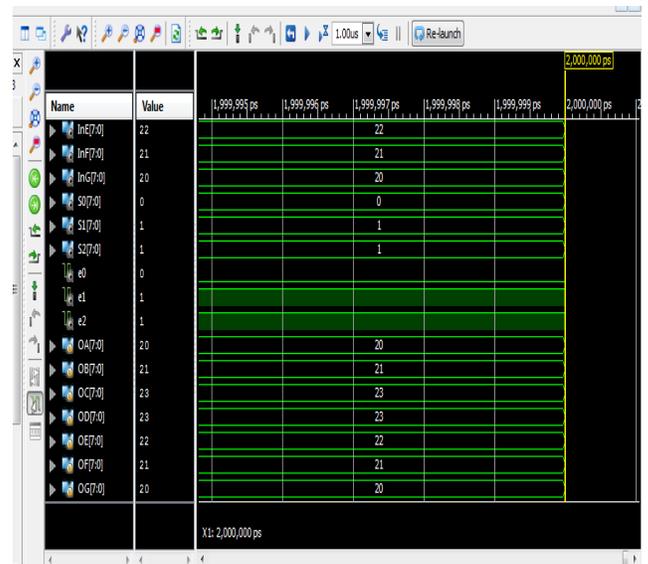


Fig 5. Simulation results for ECCs approach(contd.)

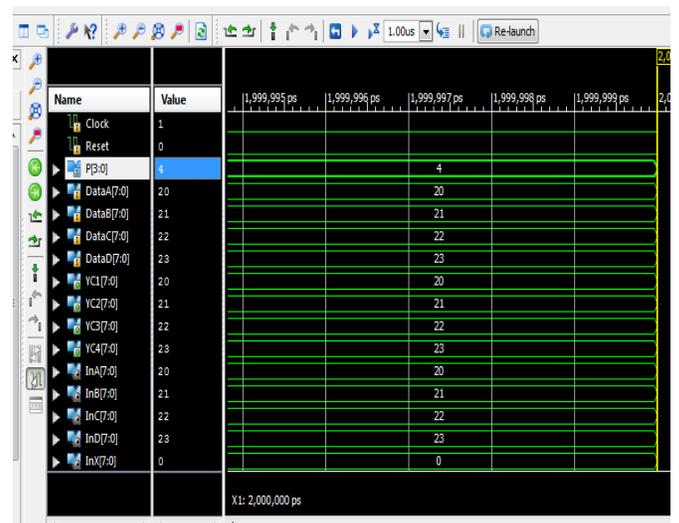
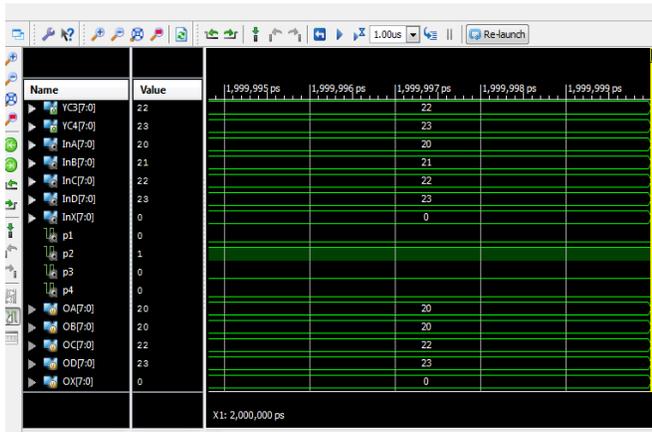


Fig6. Simulation results for PARITY-PARSE VAL CHECKS-ECC approach.

## An Efficient Error Correction Technique for Parallel FFTs



**Fig 7. simulation results for PARITY-PARSE VAL CHECKS-ECC approach(contd.)**

### Advantages:

- Improve error detection capabilities
- reduce complexity

### VI. CONCLUSION

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.

### VII. 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.

### VII. REFERENCES

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

[2] R. Baumann, "Soft errors in advanced computer systems," *IEEE Des. Test Comput.*, vol. 22, no. 3, pp. 258–266, May/June 2005.

[3] M. Nicolaidis, "Design for soft error mitigation," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, Sep. 2005.

[4] A. L. N. Reddy and P. Banerjee, "Algorithm-based fault detection for signal processing applications," *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1304–1308, Oct. 1990.

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

[6] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, "Totally fault tolerant RNS based FIR filters," in *Proc. 14th IEEE Int. On-Line Test Symp. (IOLTS)*, Jul. 2008, pp. 192–194.

[7] B. Shim and N. R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 4, pp. 336–348, Apr. 2006.

[8] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 323–336, Mar. 2012.

[9] J. Y. Jou and J. A. Abraham, "Fault-tolerant FFT networks," *IEEE Trans. Comput.*, vol. 37, no. 5, pp. 548–561, May 1988.

[10] S.-J. Wang and N. K. Jha, "Algorithm-based fault tolerance for FFT networks," *IEEE Trans. Comput.*, vol. 43, no. 7, pp. 849–854, Jul. 1994.

[11] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[12] A. Sibille, C. Oestges, and A. Zanella, *MIMO: From Theory to Implementation*. San Francisco, CA, USA: Academic, 2010.

[13] G. L. Stüber, J. R. Barry, S. W. McLaughlin, Y. Li, M. A. Ingram, and T. G. Pratt, "Broadband MIMO-OFDM wireless communications," *Proc. IEEE*, vol. 92, no. 2, pp. 271–294, Feb. 2004.

[14] S. Sesia, I. Toufik, and M. Baker, *LTE—The UMTS Long Term Evolution: From Theory to Practice*, 2nd ed. New York, NY, USA: Wiley, Jul. 2011.

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

[16] P. Reviriego, S. Pontarelli, C. J. Bleakley, and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," *IET Electron. Lett.*, vol. 48, no. 20, pp. 1258–1260, Sep. 2012.

[17] Z. Gao et al., "Fault tolerant parallel filters based on error correction codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 384–387, Feb. 2015.

[18] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.

[19] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "A novel concurrent error detection technique for the fast Fourier transform," in *Proc. ISSC*, Maynooth, Ireland, Jun. 2012, pp. 1–5.