FAULT-TOLERANT PARALLEL FFTS ARE IMPLEMENTED BY USING ERROR CORRECTION CODES AND PARSEVAL CHECKS

Myana Vinay[1] , Dr S Kishore Reddy[2] , Dr V Naga Raju[3]

[1]M.Tech Student, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.

[2]Associate Professor, HOD, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.

[3]Assistant Professor, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.

Abstract: *Modern electronic circuits are prone to reliability problems due to the occurrence of soft mistakes. Consequently, there is a necessity for safeguarding against tiny errors across a wide range of applications. These advancements also influence the systems responsible for signal processing and communications. The algorithmic fault tolerance (ABFT) methodologies offer a compelling choice for some applications. The objective of these methodologies is to detect and rectify faults by leveraging algorithmic traits for this purpose. The ABFT program demonstrates exceptional performance in both signal processing and information transfer. The utilisation of quick Fourier transforms, also referred to as FFTs, is essential in a wide range of diverse systems. Several protection mechanisms for FFTs have been proposed to detect and rectify mistakes. The Parseval check, also referred to as the sum of squares check, is widely recognised as one of the most famous among these few checks. Contemporary communication networks are progressively defined by the existence of numerous components that function simultaneously. This occurrence is increasingly prevalent. Recently, this fact has been utilised to establish a method for including fault tolerance in parallel filters. In this overview, the primary application for this strategy is safeguarding FFTs. Subsequently, two recommendations for improved security protocols are proposed and evaluated. One of these methods utilises parseval checks, while the other employs error correction codes. Based on the statistics, it appears that the proposed methods have the potential to further reduce the cost of protection.*

Keywords: *ABFT, FFT's, Fourier Transforms, Communication Networks, Signal Processing.*

I. INTRODUCTION

The complexity of signal processing and transmission circuits continues to grow with each passing year. Because of the scalability of CMOS technology, it is possible to integrate a rising number of transistors into a single chip. This makes the idea of this happening a feasible one. As the complexity of the circuits continues to increase, the likelihood of errors occurring in them is also rising. As a result of scaling, transistors operate at lower voltages, which makes them more susceptible to mistakes brought on by noise and flaws in manufacture. Alongside the development of new technology comes an increase in the significance of radiation-induced soft mistakes. If you change the logical value of a circuit node, you can be making a soft mistake that results in a momentary error that might potentially have an impact on the functioning of the system. Through the use of a number of different methods, it is possible to guarantee that soft faults will not have an impact on the functioning of a particular circuit. Utilizing silicon-on-insulator and several other specialized manufacturing techniques for integrated circuits is one of the methods that may be used. Creating fundamental circuit blocks or whole design libraries is yet another approach that may be taken to reduce the likelihood of soft mistakes occurring. At the level of the system, redundancy may also be added assist in the detection and resolution of errors.

Utilizing triple modular redundancy (TMR), which entails tripling a block and then utilizing the three outputs to vote on which one is the best for discovering and resolving problems, is a typical example of this kind of redundancy. The most significant problem with these solutions for reducing the impact of soft errors is the large overhead that is needed in the construction of circuits. As an example, TMR has an overhead that is around 200 percent. The reason for this is that the unprotected module is duplicated three times, which results in an increase in overhead that is higher than the original by a factor of 200 percent. In addition, voters are required to correct mistakes, which results in an increase in overhead that is more than 200 percent. There are a great number of applications in which this expense becomes superfluous. The use of the computational capabilities of the circuit as a means of identifying and fixing problems is an additional technique. The acronym ABFT, which stands for algorithm-based fault tolerance, is often what people mean when they speak about this. Using this strategy, it is possible to decrease the amount of overhead that is necessary to secure a circuit. Because of their consistent designs and the many algorithmic components that they include, circuits that are used in signal processing and communications are perfect for ABFT analysis. Throughout the years, several ABFT strategies have been offered with the purpose of safeguarding the fundamental components that are often used in these circuits. The preservation of digital filters has been the subject of a great number of publications and articles. It is possible that replication, which makes use of duplicates of the filter with a lesser precision, might be a more cost-effective alternative to TMR. Over the last few years, there has been an increase in the use of the distribution of the filter output information locate and fix problems with a low amount of overhead. There has also been a significant amount of research conducted on the protection method known as the fast Fourier transform (FFT).

In increasingly complicated signal processing systems, several parallel filters or fast Fourier transforms (FFTs) are used. Filter banks and MIMO communication systems are two examples of places where this kind of behavior might be seen. Parallel IFFTs and FFTs are used to carry out the modulation and demodulation procedures in MIMO orthogonal frequency division modulation (MIMO-OFDM) systems when it comes to modulation and demodulation. WiMax and mobile systems that utilize long-term evolution are both examples of technologies that make use of MIMO-OFDM. It is possible to apply ABFT methods to the whole collection of parallel modules if you have parallel filters or FFTs. This is an alternative to applying these techniques to each individual parallel module. The first study that was conducted on this subject for digital filters took into consideration two filters. In a recently implemented comprehensive plan, error correcting codes, often known as ECCs, play a crucial role. Considering that parity check bits are nothing more than addition operations and that each filter is analogous to a bit in an ECC, the reasoning behind this technique is that it ought to be effective. It is possible to take use of this method in processes in which the accumulation of a number of inputs results in a single output. Any linear operation and the DFT are both examples of what are included in this category.

II. LITERATURE SURVEY

The purpose of wavelet analysis is to examine signals through the lens of short constrained energy functions discover new and improved methods of using signals. "WT" is an abbreviation that conveys the meaning of "wavelet transform" and explains this transformation. Through the analysis of the signal f (t), it is possible to represent it as

For sums that are either finite or infinite, the integer index l is chosen as the appropriate index. The symbol al is used to indicate the representation of the set of expansion coefficients that have real values, while the symbol σl (t) is used to represent the set of expansions. In the event that the expansion is one of a kind, the set should be defined as the foundation for a category of functions. The fact that the basis is orthogonal allows us to use the inner product to get the coefficients, provided that the second condition is met.

Mother Wavelets, also known as wavelet expansion functions, are a selection of wavelet expansion functions that may be used for practical signal analysis. However, the optimality of this wavelet is dependent on the application. In the event that the shape of the wavelet at a certain scale and area is very comparable to the shape of the signal, then a significant transform value is produced; in the absence of such a similarity, the contrary is in fact the case. It is likely that the capability to vary the frequency resolution will make it feasible to identify signal qualities that might be of assistance in assessing the condition of the system after an interruption or the reason of the transient. Wavelet analysis of signals with concentrated impulses and oscillations is improved when transient signals contain fundamental and low-order harmonics. This is because wavelets are able to focus on short time periods for high-frequency components, which allows them to better analyze these signals. As a result, Wavelet is an efficient time-frequency approach for signal analysis over a variety of frequency ranges. This is accomplished by extending and translating a single function known as the Mother wavelet. Split a signal into scales that have varying temporal and frequency resolutions, the DWT implementation uses a multiresolution signal decomposition technique. This enables the signal to be divided into scales.

III. FFTS AND ERROR CORRECTION

In the realm of computers, the term "Fourier transform methods" or "spectral methods" encompasses a wide range of challenging issues that need to be addressed. Only a handful of these issues may benefit from the use of the Fourier transform as a computing tool; it is only beneficial for completing certain basic data processing tasks. When used in a variety of settings, the Fourier transform (or its associated "power spectrum") may be an attractive notion in and of itself.

Both of these types of issues are approached in a manner that is comparable. Throughout the course of history, there has been a distinct divide between the body of literature on "classical" numerical analysis and that which discusses Fourier and spectrum approaches. At this point in time, it is not essential to maintain things in such a split manner.

It is hardly likely that we will find Fourier techniques to be unique or peculiar because of the broad use of these approaches in research. Although this is a widespread area of computer science, we are also aware of the fact that many users have less experience in areas such as numerical integration or differential equations. This is something that we need to take into consideration.

An method known as the fast Fourier transform (FFT) may be used calculate the inverse of a sequence or its DFT. transform a signal from its original domain, which is often space or time, into a representation in the frequency domain, Fourier analysis is used. This process may also be performed in the other direction. These changes are easily calculated using the fast Fourier transform (FFT), which reduces the DFT matrix to a product of sparse entries, the majority of which are zero. [1]

Based on this, it is possible that the difficulty of calculating the DFT might be reduced from o (n2), which is the result when depending only on the DFT definition, to o (n log n), where n is the quantity of data.

3.1 **PROPERTIES:**

The symmetry criterion will be satisfied by the outputs of the DFT (Differential Fourier Transform) if the input data is completely real, which is the case in many applications.

$$X_{N-k} = X_k^*$$

illustrate this point, Sorensen (1987) provides examples of effective FFT algorithms that were developed specifically for situations like this one. The adoption of a standard algorithm (such as Cooley-Tukey) and the elimination of all unnecessary steps from the computation is one method that may be used to reduce the amount of time spent on memory and processing by about a factor of two.

A half-length complex DFT is a potential alternative to defining an even-length real-input DFT. In this DFT, the real and imaginary sections reflect the even and odd parts of the starting real data, respectively. do this, O(N) post-processing operations would be necessary. It was formerly thought that the discrete Hartley transform, sometimes known as the DHT, was a more effective approach for calculating real-input discrete Fourier transforms. A more recent argument, , states that, given an input count, it is often possible to identify a tailored real-input DFT technique (FFT) that makes use of fewer operations than its associated DHT method (FHT). Bruun's method (which may be seen above) has not been able to attract considerable support, it was first presented with the intention of capitalizing on genuine inputs. When dealing with actual data that has even or odd symmetry, you could come across additional FFT specializations. It is possible that under some circumstances, one might possibly get an additional memory and time factor of around two by transforming the DFT to the discrete cosine/sine transform (DCT/DST). However, under some circumstances, it is possible to compute DCTs and DSTs by combining FFTs of actual data with O(N) pre/post processing. This is an alternative to just altering an FFT .

IV. RECOMMENDED PROTECTION METHODS FOR COMPANY FFTS

We begin our study by focusing on the ECC-based protection method for digital filters for digital filters. Figure 4.1 may be used to illustrate the plan. correct a single error, a straightforward Hamming algorithm is used here. identify and correct problems, the basic system, which consists of four modules, has been expanded to include three redundant FFT modules. To ensure that linear combinations of outputs are acceptable, the three redundant modules accept linear combinations of inputs as inputs. This allows them to check the outputs. An example of this would be the input to the first redundant module, which is

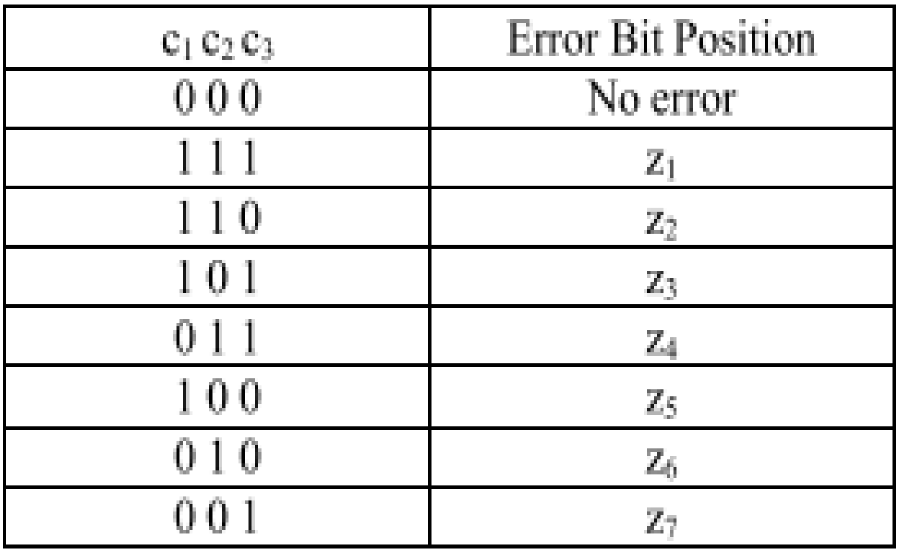
$$x5 = x1 + x2 + x3 \text{ .... } (1)$$

$$z5 = z1 + z2 + z3 \text{ .... } (2)$$

It is anticipated that the designation "C1 check" will arise. The logic behind the two more redundant modules, c2 and c3, which will be responsible for providing the tests is the same. It is possible to identify the malfunctioning module by paying close attention to the discrepancies that are seen throughout each examination with great care. The mistakes that correspond to each pattern are summarized in Table 1, which may be accessed by following the link. Following the identification of the faulty module, it is conceivable that it will be feasible to fix it by replicating its output with the help of the other modules. As an example, this may be carried out in the event that z1 is impacted by an error.

$$z1c[n] = z5\ [n] - z2[n] - z3[n] \ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

fix faults that have occurred in the other modules, similar correction equations may be used. In the event that a large number of modules need error correction, more sophisticated ECCs could be used in that particular application. In light of the fact that the number of duplicate FFTs is precisely proportional to the logarithm of the original FFTs, this technique has a lower overhead than TMR, as was shown before. As an illustration of this concept, it is necessary to have four redundant FFTs protect eleven FFTs, while it is necessary to have three redundant FFTs protect four FFTs. The overhead is shown to decrease as the number of FFTs increases, as can be seen in this example.



$c_1\ c_2\ c_3$	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

TABLE1: ERROR LOCATION IN THE HAMMING CODE

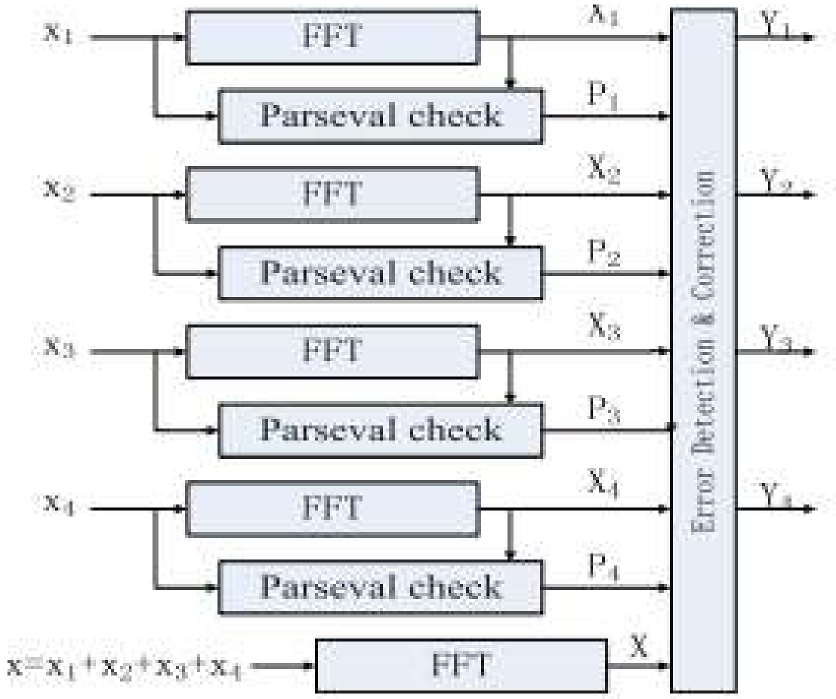


FIG 4.1: PARITY-SOS (FIRST TECHNIQUE) FAULT-TOLERANT PARALLEL FFTS.

As an illustration of this concept, it is necessary to have four redundant FFTs protect eleven FFTs, while it is necessary to have three redundant FFTs protect four FFTs. We can observe that the overhead is reduced as the number of FFTs grows in this particular instance.

Various methods for protecting the FFT have been proposed throughout the years, as seen in Section I. One method for detecting errors is the Sum of Squares (SOS) check. In order for the SOS check to work, the Parseval theorem must be satisfied. The SOSs of the FFT's inputs and outputs are the same, with the exception of a scaling factor, according to this theorem. Because this connection only requires one multiplication for both the input and output samples, it allows for little overhead error detection. This is different from SOS, which requires two adders and multiplications for every sample.

By combining the ECC technique with the SOS check, it is possible to lessen the load of protection that is placed on parallel FFTs. In light of the fact that the SOS check can only detect mistakes, the ECC component need to be able to implement the correction. By leveraging what is effectively a simple parity bit, you are able to do this for each and every FFT. Additionally, the SOS check is used by each and every FFT identify mistakes. When utilizing the output of the parity FFT, it is feasible to find solutions to difficulties.

Through the use of an illustration, this explanation is made more explicit. Within the context of four simultaneous FFTs, the first technique that has been developed is shown in Figure 4.2. The total of the inputs to the initial FFTs is what is used as the input for a redundant FFT known as the parity data. Together with the first FFT, there is also an SOS check provided. When testing for mistakes using P1, P2, P3, and P4, it is feasible to amend them by recalculating the necessary FFT using the parity FFT (X) result in conjunction with the other FFT results. This will allow for the correction of the errors. If the initial FFT is found to have an error, for instance, P1 will be set, and the process will continue.

$X1c = X - X2 - X3 - X4$………………………… (4)

The combination of the parity FFT with the SOS check results in the elimination of one extra FFT, which may therefore contribute to a decrease in the amount of protection overhead. Beginning from this point forward, we will refer to this strategy as parity-SOS, which is an abbreviation that stands for first recommended method. combine the SOS check with the ECC approach, one alternative is to utilize an ECC for the SOS checks rather than an SOS check per FFT. This is an additional option.

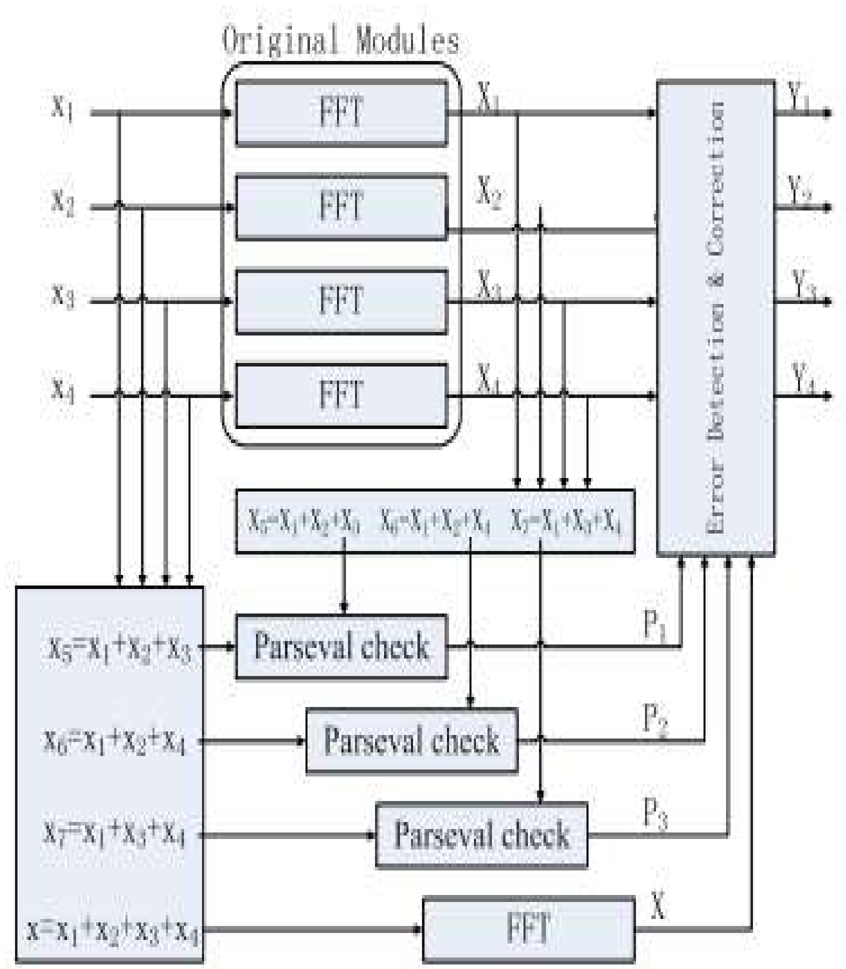


FIG. 4.2: Another Method Is Parity-Sos-Ecc, Which Stands For Parallel Ffts With Failure Tolerance.

.

Following that, an extra parity FFT approach, which is analogous to the parity-SOS technique, is used rectify the faults. As may be seen in Figure 4, this second technique is illustrated. One of the most notable differences is that it needs less SOS tests when compared to the original paritySOS approach. The following are the procedures that may be used to locate mistakes and rectify them, which are analogous to the ECC system shown in Figure 4.1 and the parity-SOS approach. This system will now be known as parity-SOS-ECC, which is an acronym that represents the second solution that was submitted for consideration.The amount of extra FFTs and SOS check blocks that are required is one example that may be considered when beginning the process of evaluating the overheads of the two proposed systems. This information is compiled in Table II for a collection of k original FFT modules, with the assumption that k is a power of two. The two suggested approaches are shown to minimize the number of extra FFTs to a single one, as shown by the results described. Furthermore, the second technique also results in a lower number of SOS checks being produced. When it comes to an FPGA implementation, Section III offers a comprehensive analysis of the relative overheads that are associated with the suggested techniques.Moreover, the objects that are supplied for protection in any of the aforementioned methods are susceptible to being harmed by soft mistakes as well. With the ECC approach in mind, we investigated several strategies to preserve the integrity of these components. Given that a mistake cannot spread to the data outputs or cause a correction to be triggered, it is impossible for it to have an effect on the redundancy or parity FFTs.

An error in the SOS check might result in a repair even if the FFT does not provide evidence of such a remedy. Despite the fact that this may result in an unneeded patch, it will allow for the achievement of the intended aim. It is possible that faults in the blocks shown in Figures 4.2 and 4.3, which are responsible for detecting and repairing errors, might have an effect on the outputs at the end. Such blocks are safeguarded by TMR in our implementations of the specification. The double FFTs shown in Figure 4.1, as well as the SOS checks shown in Figure 4.3 and the adders that calculate their inputs, are all examples of situations in which this is confirmed. The inclusion of three of these blocks has a little effect on the complexity of the circuit since the calculations that they do are far less intricate than those that are performed by FFT.Before we come to a conclusion, it is important to point out that the quantization that was used in the process of generating the FFTs is the same that the ECC system employs detect and eliminate any mistakes. , despite the fact that the SOS check identify the majority of problems, it makes no guarantees that it will identify every single failure. Therefore, assess the three for a particular implementation, it is necessary to perform fault injection tests ascertain the proportion of each approach.

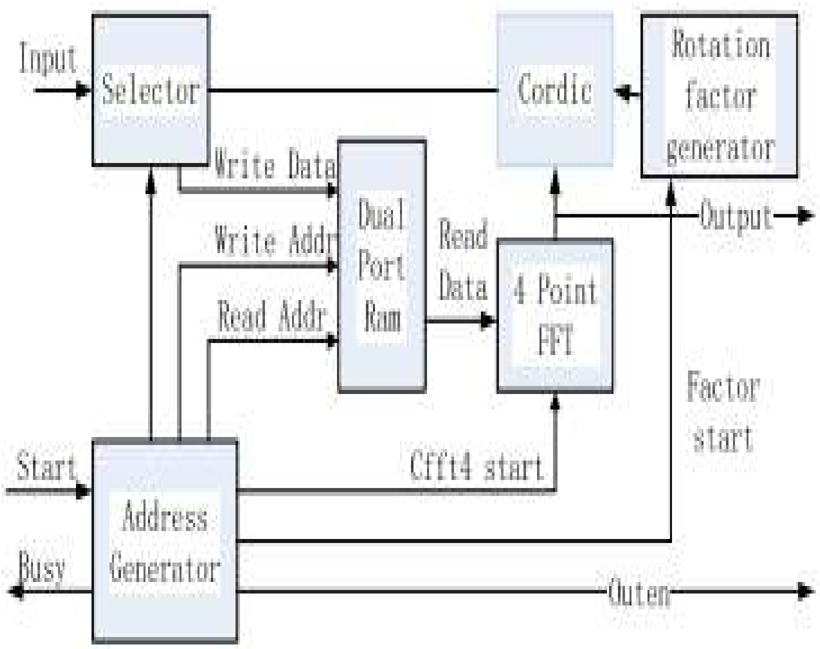


FIG4.3: Framework for the FFT Deployment

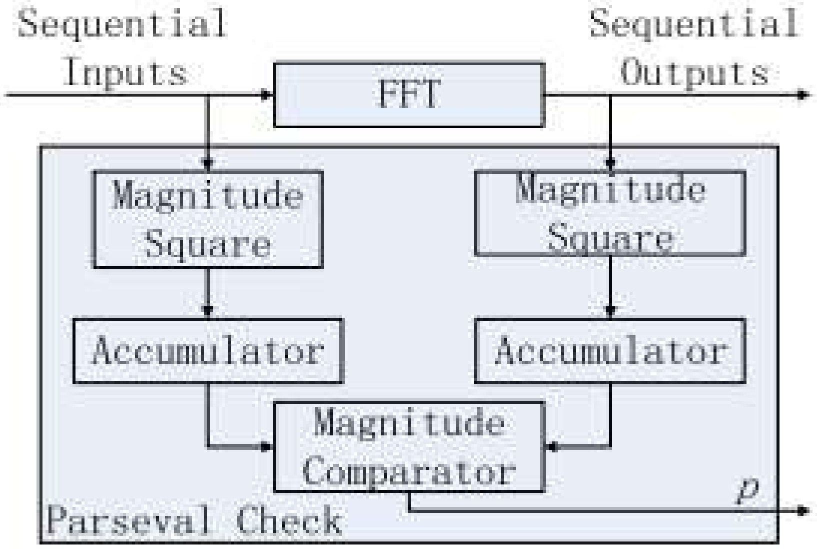


FIG4.4: STARTING THE SOS STUDY

corrections made to errors that have been made. Due to this, it is necessary to do an analysis of error coverage and overhead.

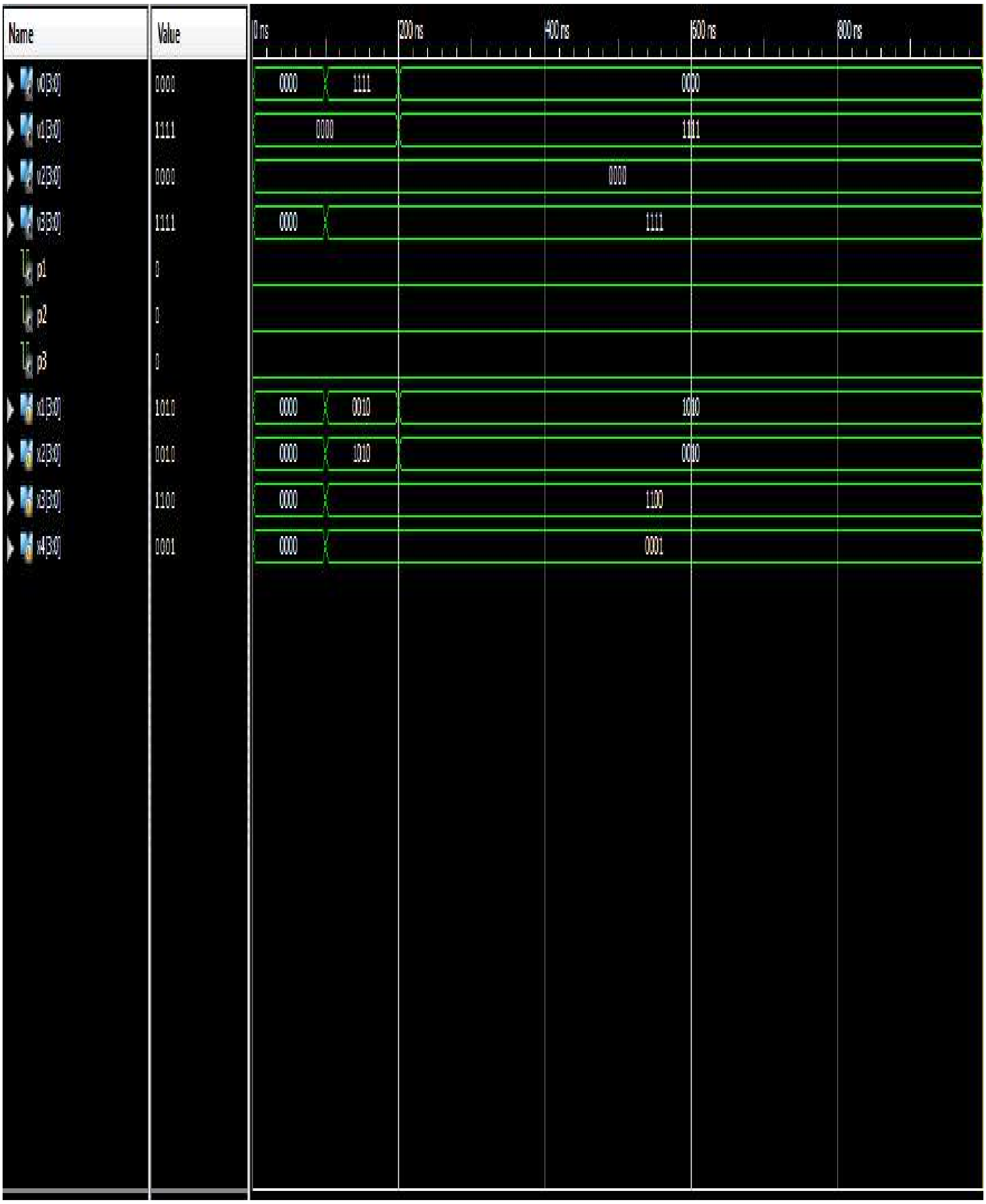
4.1 EVALUATION:

A four-point decimation-in-frequency FFT core is used to complete the FFT calculations several times. The idea that MIMO-OFDM may have applications in wireless networks prompted us to build this core. Figure 4.4 shows the implementation of the four-point FFT core.You may include the number of fast Fourier transform points and the online computation and storage of the rotation coefficients for each step into the code. complete the assessment by computing the FFT for 1024 input samples, which involves five stages of a 10-point FFT (log41024= 5). To compute them, this is essential. The total number of cycles needed is 5,120, to rephrase. The inputs can take up to 12 bits of width, while the outputs can take up to 14 bits.

the inputs to the redundant FFT are a mixture of a large number of signals, the bit widths have been increased to 14 bits and 16 bits, respectively, support a wider dynamic range. The SOS check makes use of accumulators that are compared at the conclusion of the block in a sequential manner. This is quite similar to the way that the inputs and outputs of the FFT are sequential at the same time. As may be seen in Figure 4.5 is this. reduce the effect that roundoffs have on the fault coverage, the outputs of the accumulator, which are 39 bits wide, are used. When doing the assessment, a variety of alternative values for the number of parallel FFTs are taken into consideration. This is done ensure that the various methods may be assessed in a manner that is proportional to the number of parallel FFTs that were present in the initial system.

Multiplexers (Figures 4.1-4.3) are used by the error detection and correction blocks choose the appropriate output in accordance with the error pattern that has been identified. As was noted earlier, these blocks are subjected to a triple check guarantee that any faults that may have an effect on them do not taint the final outputs. Verilog has made it possible to build a number of different security methods, one of which is the FFT system. Following that, the design was transferred to a field-programmable gate array (Virtex-4 xc4vlx80) increase performance while simultaneously decreasing resource usage. Following is a summary of the findings that may be found in the tables. In the first table, the resources that are required to carry out one SOS check and one FFT are explained in detail. According to the findings, the FFT proved to be more difficult than the SOS check, which was to be anticipated. Through the use of a completely parallel FFT solution, the discrepancy will become substantially more pronounced.

V. OUTPUT WAVEFORMS



Name	Value
v0[3:0]	0000
v1[3:0]	1111
v2[3:0]	0000
v3[3:0]	1111
p1	0
p2	0
p3	0
x1[3:0]	1010
x2[3:0]	0010
x3[3:0]	1100
x4[3:0]	0001

FIG: 5.1: Waveforms of Output

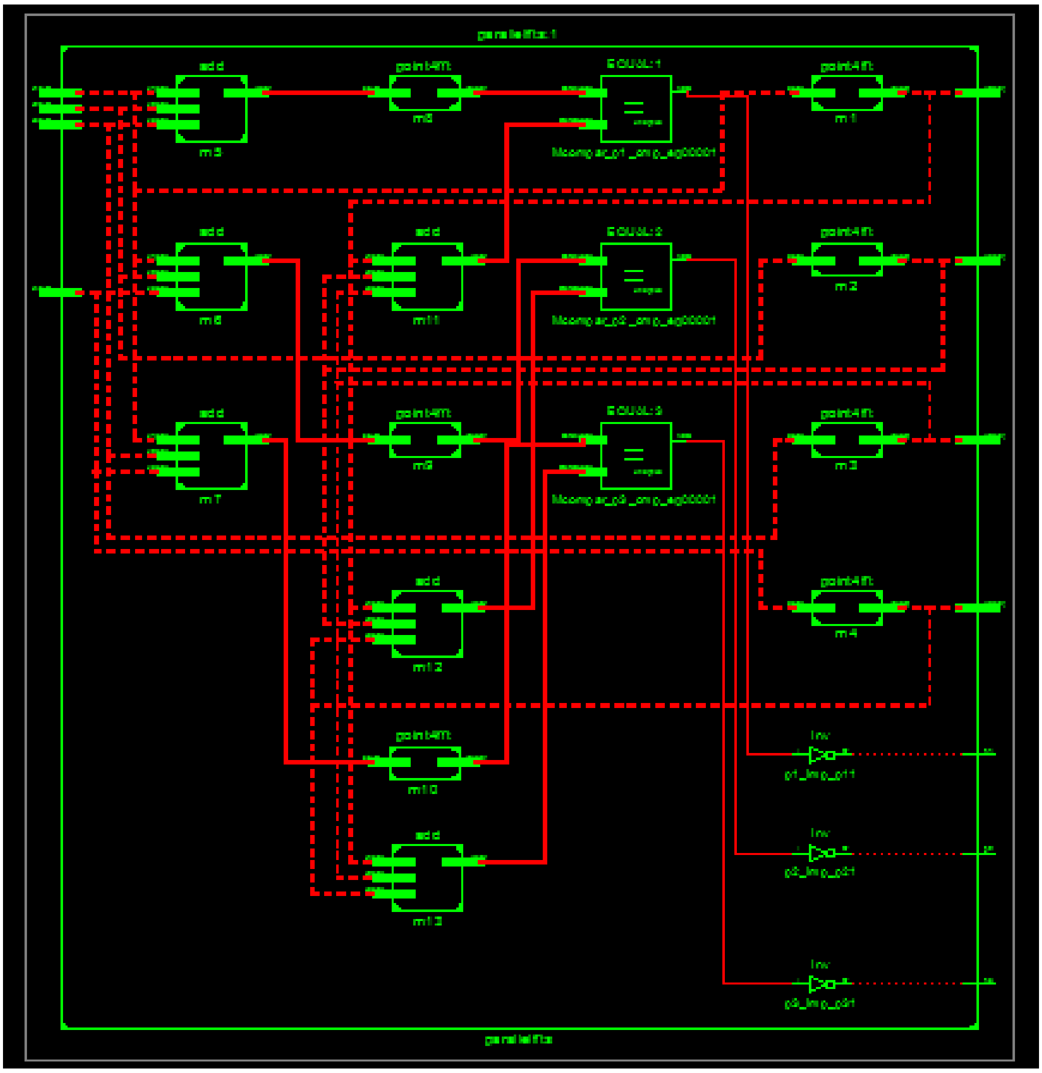


FIG: 5.2: Diagram of a Schematic Block

VI. CONCLUSION

The purpose of this short is to examine the protections that have been included to avoid soft mistakes during the implementation of parallel FFT. A pair of strategies have been analyzed by us. The suggested techniques combine the conventional SOS check with an existing ECC methodology. This technique is based on an existing ECC technique. It is necessary to make simple parity FFT rectify the faults that have been found by the SOS tests. An SOS check per FFT or an ECC, which is a collection of SOS tests, may be used locate and identify the issues that are occurring there.

The offered methods have been reviewed by us based on how simple it is to put them into action and how effectively they identify errors. Taking into account the level of complexity involved in putting the plan into action, the findings indicate that the second choice has the most favorable results. Combining the parity FFT with a number of SOS tests results in the construction of an ECC. After doing research on fault injection, it has been shown that the ECC approach is capable of recovering from any error that falls within the error protection tolerance range. The defect coverage for the parity-SOS scheme and the parity-SOS-ECC system is approximately 99.9% when the tolerance level for SOS check is set to 1.

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/Jun. 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 Integer. (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. Vaidyanathanm, 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