Volume: 4 Issue: VIII Month of publication: August 2016

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Fault Detection and Soft Errors Correcting Codes Using Parallel FFT

Vennam Ratheeshma[1], M. Ramakrishna[2]
M. Tech, (VLSI), Assoc. Prof.

Abstract: The increasing demand of low complexity and error tolerant design in signal processing systems is a reliability issue at ground level. Complex circuit is consistently affected by soft errors in modern electronic circuits. Fast Fourier transforms (FFTs) plays a key role in many communication and signal processing systems. Different algorithms have been used in earlier techniques for achieving fault tolerant coverage design. In real time application systems, numbers of blocks operating in parallel are frequently used. The proposed work exploits a technique to implement fault tolerance parallel FFT with reduced low complexity of circuit area and power. In Partial summation along with error detection and correction hamming code is used for designing soft error tolerant parallel FFT shelter. This new method achieves lower complexity proportional to that of FFT design size. Based on these two schemes, two modified preserve techniques that combine the use of error correction codes and Partial summation are proposed and evaluated. First method, the Parity-Partial Summation and ECC uses one FFT with minimum Partial Sum blocks for reducing hardware area. Secondly, Parallel Partial Summation ECC used for correcting errors in multiple FFTs protective methods. The result for 4-paralle1 and 6-paralle1 FFTs shows that the proposed technique effectively reduces area and power of fault tolerant design along with improved fault coverage.

Keywords: Fast Fourier transforms (FFT), EIC, Partial Summation, and ABFT.

I. INTRODUCTION

The CMOS technology scaling has made today's designs more susceptible to radiation induced soft errors. Soft errors can alter the logical output of a circuit node creating an error that affects the system functionality [2]. The problem becomes more complexity of the soft error rate exponentially increases with that of circuits scaling. Single Event Upsets (SEU) also affects the reliability of the circuit due to variation in set-up and hold time. Various methods have been adopted earlier to mitigate soft errors. Specifically designing libraries used for complex circuit and modified manufacturing process such as the silicon on insulator design also used for minimizing the error probability. Adding redundancy to keep the design free from temporary errors is also adopted in existing designs.

Five modular redundancy mitigation techniques recovers the Error Module used to overcome the unintended behavior of the system by adding redundancy [7], [8], [9]. It adds two identical designs and a voter along with the original design to produce correct results. It increased the area overhead which not suitable for complex designs. Some techniques have been introduced to eliminate this problem by making changes in the algorithm (ABFT) [10]. Based on algorithms, the use of the Parseval theorem or sum of squares check is one the frequently used method. The SOS check states that the SOSs of the inputs to the FFT are equal to the SOSs of the outputs of the FFT. This correlation can be used to detect errors while using multiple FFTs.

As the growth of SoC and NoC becoming increasingly popular, it is frequent to find FFTs working in parallel. This happens especially for applications in orthogonal frequency demodulation multiplexing (OFDM) systems, such as IEEE 802.11a/g, Long Term Evolution (LTE), and Digital Video Broadcasting-Terrestrial (DVB-T) [11], [12]. A MIMO-OFDM design also sees several FFTs operating parallel for obtaining diversity gain and to combat signal fading. So, keeping the FFTs secure was one of the important tasks in these kinds of applications.

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.

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. The importance of radiation-induced soft errors also increases as technology scales [2]. Soft errors can change the logical value of a circuit node creating a temporary error that can affect the system operation. To ensure that soft errors do not affect the operation of a given circuit, a wide variety of techniques can be used [3]. These include the use of special manufacturing processes for the integrated circuits like, for example, the silicon on insulator. Another option is to design basic circuit blocks or complete design libraries to minimize the probability of soft errors. Finally, it is also possible to add redundancy at the system level to detect and correct errors.

One classical example is the use of triple modular redundancy (TMR) that triples a block and votes among the three outputs to detect and correct errors. The main issue with those soft errors mitigation techniques is that they require a large overhead in terms of circuit implementation. For example, for TMR, the overhead is >200%.This is

because the unprotected module is replicated three times (which requires a 200% overhead versus the unprotected module),and additionally, voters are needed to correct the errors making the overhead >200%. This overhead is excessive for many applications. Another approach is to try to use the algorithmic properties of the circuit to detect/correct errors. This is commonly referred to as algorithm-based fault tolerance (ABFT) [4]. This strategy can reduce the overhead required to protect a circuit.Signal processing and communications circuits are well suited for ABFT as they have regular structures and many algorithmic properties [4]. 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 [5], [6]. 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 [7]. The knowledge of the distribution of the filter output has also been recently exploited to detect and correct errors with lower overheads [8]. 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.

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 [16] where two filters were considered. More recently, a general scheme based on the use of error correction codes (ECCs) has

been proposed [17]. 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

II. EXISTING SYSTEM

A. *Parallel FFT Protection Using ECCs*

His technique provides new alternatives to protect parallel FFTs that can be more efficient than protecting each of the FFTs independently.

This work starts with the protection scheme based on the use of parity-SOS for digital filters. In the first technique, original module consist of 4 FFTs. Input is given to each FFTs separately as x1,x, x3, x4. Here 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.

Each FFT consist of a SOS check in parallel to detect the error in the FFT. The output of the Parseval check is represented as P1, P2, P3, and P4. If there is any error in the FFT then the Pi will set to 1.

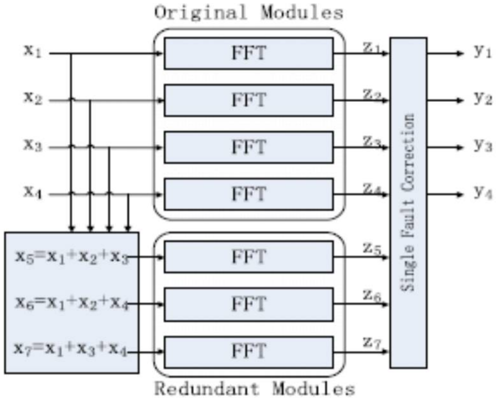


Fig 1 Parallel FFT protection using ECCs

The output of the FFT and the Pi outputs are given to error detection and correction block. If there is any error in the FFT output then the additional FFT which is given parallel to the FFT module will correct the error. This Parseval check can only detect the errors in the FFT. It can't correct it by its own. For that an additional FFT is used. We can correct the error using its output.

The SOS check can be combined with the ECC approach to reduce the protection overhead in first technique for parallel FFTs. Since the SOS check can only detect errors, the ECC part should be able to implement the correction. This is done using the equivalent of a simple parity bit for all the FFTs. The SOS check is used on each FFT to detect errors. When an error is detected then the output of the parity FFT can be used to correct the error. Instead of using an SOS check per FFT we can use an ECC for the SOS checks Next technique used is parity-SOS-ECC. In this technique instead of using parallel SOS for each original module 3 SOS block is used separately which is driven by hamming code output. If there is any error in the output then the output of the Parseval check block will be set.

B. Parity-SOS Fault-Tolerant Parallel FFTS

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 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 recomposing 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.

All these are single error detection and correction method. Even though these are algorithm based error correction technique it require more area for implementation as it consist of many squaring and summing blocks. So in order to reduce the area and power consumption new technique is introduced. It is an algorithmic based method.

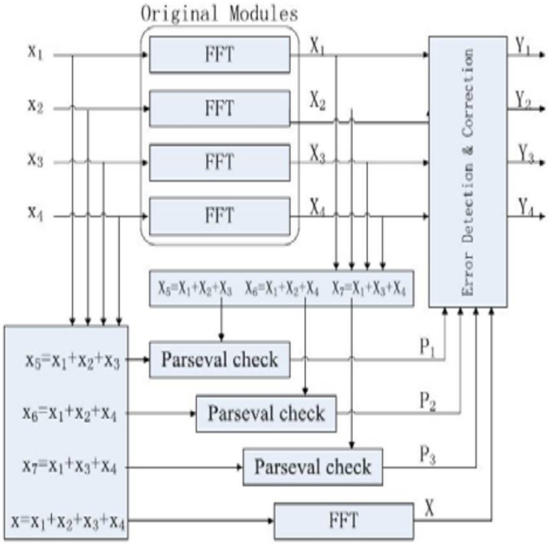


Fig 2 Parity-SOS-ECC fault-tolerant parallel FFTs

III. PROPOSED SYSTEM

This project is detecting multiple errors and corrects it with less over heads. It's better than the traditional method known as triple modular redundancy. A less expensive hardware mitigation strategy for arithmetic circuits called reduced-precision redundancy (RPR) is the proposed project. RPR is designed to protect against large magnitude errors. It is used mainly in arithmetic circuits with the help of redundant, lower precision arithmetic circuits and comparing their results. Using of Reduced Precision Redundancy may introduce low precision errors but still its area reduction make it an attractive alternative for protecting FFTs, transient and soft data errors. There are some conditions to be satisfied while designing a RPR module. These choices include the reduced precision and threshold. RPR is implemented by creating two identical reduced-precision (RP) module of the original full precision FFTs. The outputs of the two RP modules are used to determine if there is any error in the FP module than a preset threshold, *Th*, the FP module is assumed to be in error. When the FP module is found to be in error, then it will discard the FP output and use RP output with 000's appended at the LSB part. If the FP output differs from the RP outputs by less than Th, then it is considered as the error free output and the final output will be FP output. The arithmetic circuits protected by RPR may be of any size. The circuit used may be of basic arithmetic operation such as an adder or a more complex combination of operators and logics such as an infinite impulse response filters, finite impulse response (FIR) filter etc. This paper refers to the combination of FP module and RP module. There are two parameters to be considered before implementing RPR on a module. They are the bit width of the reduced precision module (Br) and the decision threshold (Th). The two values are linked and together greatly affect the cost and performance of RPR.

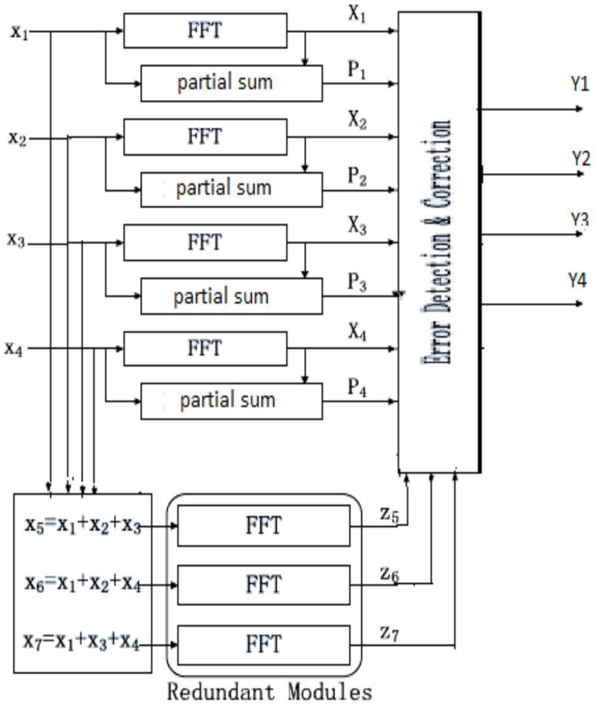


Fig 3 Parallel-PS-ECC fault-tolerant parallel FFTs

In this method, we are using parseval check method to detect the errors in the original module. By using this method we can able to detect more than one error in the original modlue . After that redundant modules are is used to correct those errors.

In partial summation method, we just add the inputs of the FFTs and it will be compare with the first output of the FFTs. In case, these two values are equal no errors in that particular FFT. If it is not equal threes is an error in that FFT. In this method, original module is enough to detect the errors in the FFTs. So, that no need for additional detection circuits. For that reason area and power will be effectively reduced in this method.

In all the techniques discussed, soft errors can also affect the elements added for protection. 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 tothe SOS checks in Fig. 3. 1.The triplication of these blocks has a small impact on circuit complexity as they are much simpler than the FFT computations.

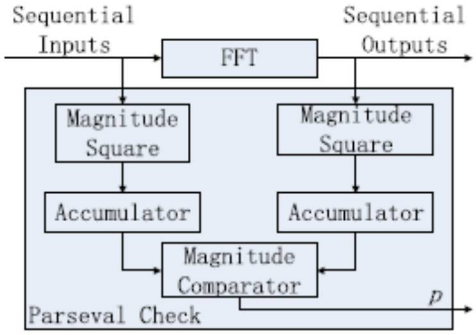


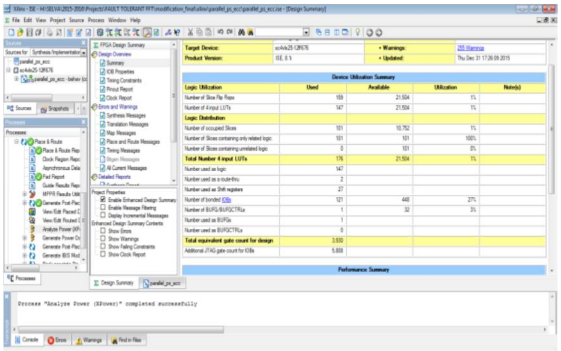
Fig 4 Implementation of Partial Summation

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

IV. EVALUATION AND RESULT

The proposed system is simulated in ModelSim SE 6.3f. A model of the communication channel is formed. In communication channels the bandwidth will be high. So the FFTs will have large inputs. This project is used for error detection and correction. So designing of FFT is not relevant, in order to evaluate the error correction capability of this technique we are taking a simple FFT. The FFT used here is 4 point radix FFT. Input is given as binary number. Six bit input is given. Initially some vales are given as input for simulation. After running the simulation if we want to check with other input we can force the required input. Before running the program clock is preset to 1. ModelSim is opened and new project is created and the program is typed in the editor. After completing the program compile the program and then start simulation. The program for running is selected and then run the program. Add waves to understand the output. First program without error is simulated. After that program with an error in FFT is simulated and its output and its waveform are observed. From the two cases it's clear that RPR method is best suited for error correction.



Target Device:	xc4vlx25-12ff676	• Warnings:	255 Warnings
Product Version:	ISE, 8.1i	• Updated:	Thu Dec 31 17:26:09 2015

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	159	21,504	1%	
Number of 4 input LUTs	147	21,504	1%	
Logic Distribution				
Number of occupied Slices	101	10,752	1%	
Number of Slices containing only related logic	101	101	100%	
Number of Slices containing unrelated logic	0	101	0%	
Total Number 4 input LUTs	176	21,504	1%	
Number used as logic	147			
Number used as a route-thru	2			
Number used as Shift registers	27			
Number of bonded IOBs	121	448	27%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			
Number used as BUFGCTRLs	0			
Total equivalent gate count for design	3,930			
Additional JTAG gate count for IOBs	5,808			

Performance Summary

```
Process "Analyze Power (XPower)" completed successfully
```

Fig 5. Design summary of 4-parallel FFTs

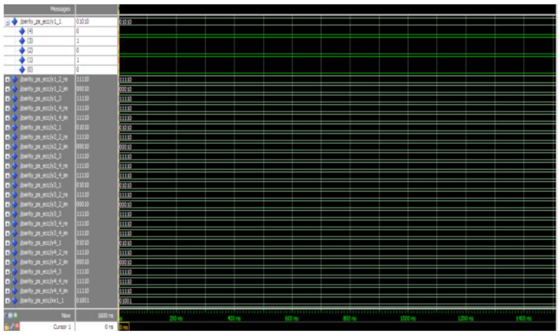


Fig 6. Simulation Results

V. CONCLUSION AND FUTURE WORK

The fault secure design of parallel FFTs against multiple errors is analyzed and two new techniques based on Partial Summation is implemented. The evaluation result shows that both the approaches results in low complexity and low power consumption. The parallel Partial summation ECC can reduce the hardware resource utilized by the Sum of Squares and also correcting errors in multiple FFTs. The second technique without hamming codes further reduces the area which is used less error prone applications. Compared to the existing counter methods, the result of proposed four-parallel and six-parallel shows the area optimization by 59% and 65% along with the 99.97 fault coverage criteria. In the future work, a self adaptive system to overcome hard errors can also be taken into consideration.

REFERENCES

[1] M. Nicolaidis, ―Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks,‖ IEEE transactions on very large scale integration (VLSI) systems, February 26, 2015.

[2] R. Baumann, ―Soft errors in advanced computer systems,‖ IEEE Des.Test Computer., 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] 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.

[5] Riaz Naseer, Rashed Zafar Bhatti, Jeff Draper, ―Analysis of Soft Error Mitigation Techniques for Register Files in IBM Cu-08 90nm Technology,‖ Information Sciences Institute University of Southern California Marina Del Rey, CA 90292 USA 2006 IEEE.

[6] 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.

[7] Haryono, ―Five Modular Redundancy with Mitigation Technique to Recover the Error Module,‖ International Journal of advanced studies in Computer Science and Engineering IJASCSE, Volume 3, Issue 2, 2014.

[8] Pedro Reviriego, Chris J. Bleakley, and Juan Antonio Maestro., ―Structural DMR: A Technique for Implementation of Soft-Error-Tolerant FIR Filters,‖ ieee transactions on circuits and systems—ii: express briefs, vol. 58, no. 8, august 2011.

[9] R. E. Lyons W. Vanderkul k., ―The Use of Triple-Modular Redundancy to Improve Computer Reliability,‖ IBM JOURNAL APRIL 1962.

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

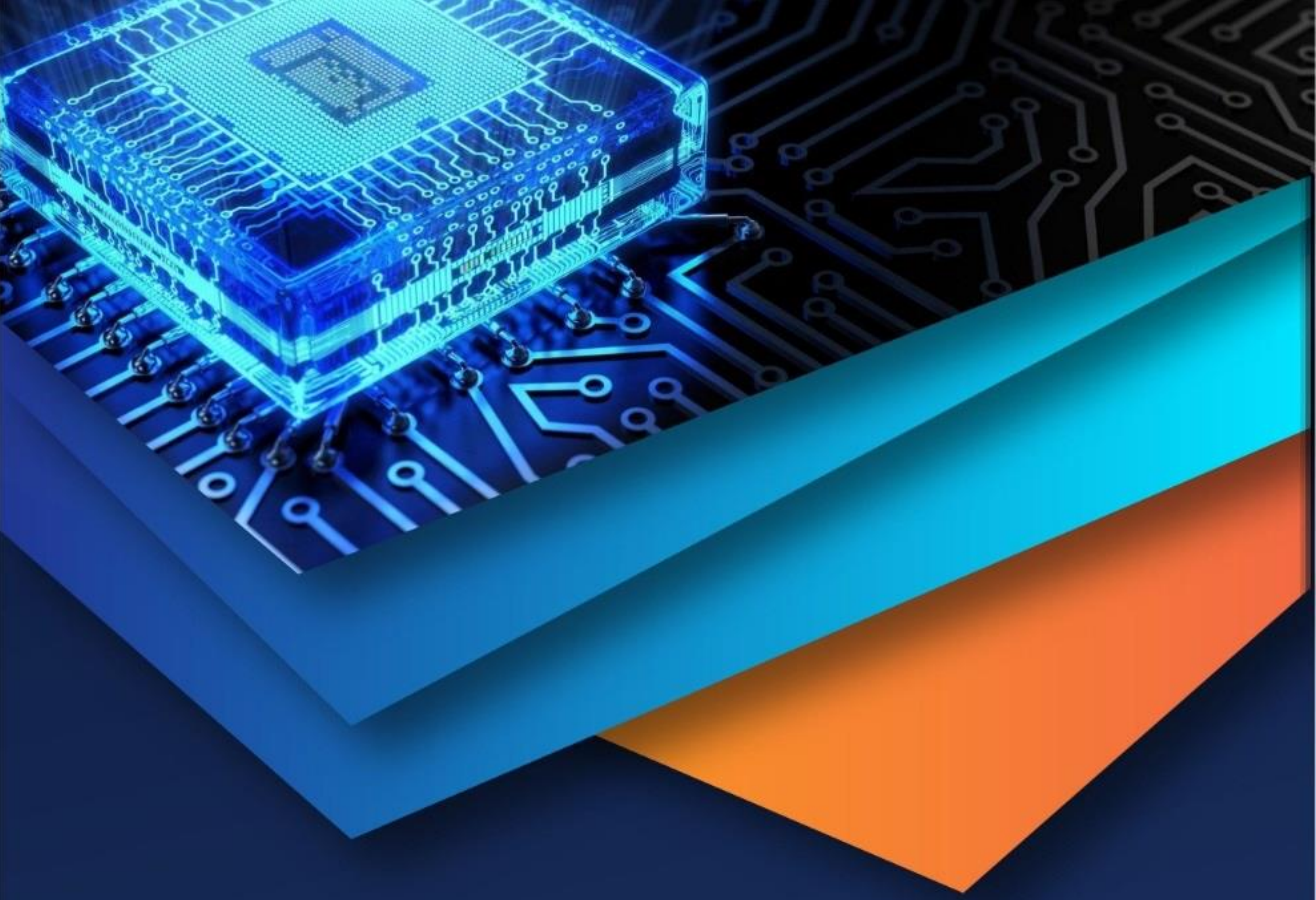
[11] G. L. Stüber, J. R. Barry, S. W. McLaughlin, Y. Li, M. A. Ingram, and T. G. Pratt, ―Broadband MIMO-OFDM wireless communications,‖ Proc.IEEE, vol. 92, no. 2, pp. 271–294, Feb. 2004.

[12] Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television,‖ Final draft ETSI EN 300 744 V1.5.1 (2004-06) European Standard (Telecommunications series).

[13] Jen-Chih Kuo, ―VLSI Design of a Variable-Length FFT/IFFT Processor for OFDM-Based Communication Systems,‖ EURASIP Journal on Applied Signal Processing.

[14] Hamming Codes,‖ James Fiedler Fall, 2004.

[15] Error detection and correction,‖ Logic and computer design fundamentals.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY