Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 12, Number 1 (2016), pp. 691-700

http://www.ripublication.com

Implementation of Soft Error Tolerant Filters for Error Detection and Correction Using ECC

S. Sri Jamiya[1], Dr. P. G. Kuppusamy[2], P. Balavenkateshwarlu[3], and J. T. Arun Ragesh

PG Scholar[1], Professor[2], Assistant Professor[3, 4]
[1, 2, 3, 4]Department of Electronics and Communication Engineering,
[1, 2, 3, 4]Veltech Multitech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai, Tamil Nadu, India.

Abstract

The focus of this paper is a lossy Fourier-transform-based compression algorithm for implementation on field programmable gate arrays in the space environment. The algorithmic based fault tolerance technique computes the fast Fourier transform (FFT) of a wide band signal, divides it into user-defined time frequency ranges of interest, and transmits only those frequency domain portions of the signal that exceeds the predefined thresholds. Error detection against single event upsets for the FFT is implemented and compared the sum of the squares of the input to the scaled SOS of the FFT output, which should be equal according to the Parseval's Theorem. Error correction is implemented by duplicating the FFT calculation, error detection and choosing output of the FFT that is not in error.

Keywords: Error correction codes (ECCs), Fast Fourier Transforms (FFTs), Soft errors.

Introduction

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 [1]. 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 [9], [10].

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 [11] or in multiple-input multiple-output (MIMO) communication systems [12]. In particular, MIMO orthogonal frequency division modulation (MIMO-OFDM) systems use parallel iFFTs/FFTs for modulation/demodulation [13]. MIMO-OFDM is implemented on long-term evolution mobile systems [14] and also on WiMax [15]. 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 the individual outputs. This is true for any linear operation as, for example, the discrete Fourier transforms (DFT).

In this paper, the protection of parallel FFTs is studied. In particular, it is assumed that there can only be a single error on the system at any given point in time. This is a

common assumption when considering the protection against radiation-induced soft errors [3]. There are three main contributions in this brief. 1) The evaluation of the ECC technique [17] for the protection of parallel FFTs showing its effectiveness in terms of overhead and protection effectiveness. 2) The proposal of a new technique based on the use of Parseval or sum of squares (SOSs) checks [4] combined with a parity FFT. 3) The proposal of a new technique on which the ECC is used on the SOS checks instead of on the FFTs.

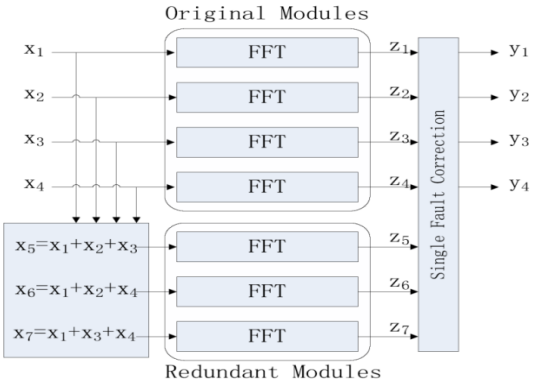

Figure 1: Parallel 4 - Point FFT protection using ECCs

The one proposed techniques provide new alternatives to protect parallel FFTs that can be more efficient than protecting each of the FFTs independently. The proposed schemes have been evaluated using FPGA implementations to assess the protection overhead. The results show that by combining the use of ECCs and Parseval checks; the protection overhead can be reduced compared with the use of only ECCs as proposed in [17]. Fault injection experiments have also been conducted to verify the ability of the implementations to detect and correct errors.

The rest of this brief is organized as follows. Section II presents the one proposed schemes. In Section III, the implementation overheads and fault tolerance of the schemes are evaluated. Finally, the conclusions are drawn in Section 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 [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

$$\mathbf{x_5 = x_1 + x_2 + x_3} \quad \textbf{(1)}$$

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

$$\mathbf{z_5 = z_1 + z_2 + z_3} \quad \textbf{(2)}$$

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:

$$\mathbf{z_1c[n] = z_5[n] - z_2[n] - z_3[n]} \quad \textbf{(3)}$$

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.

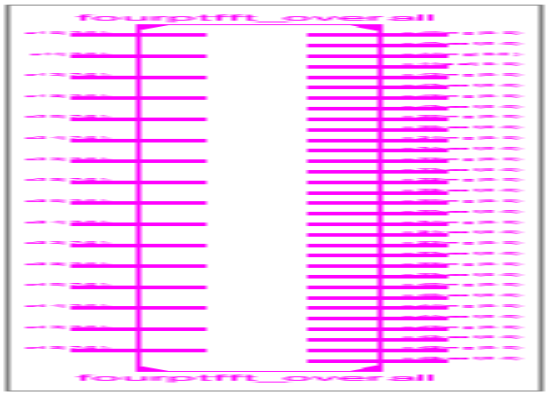

Figure 2: 4 Input 4-Point FFT RTL diagram using Xilinx.

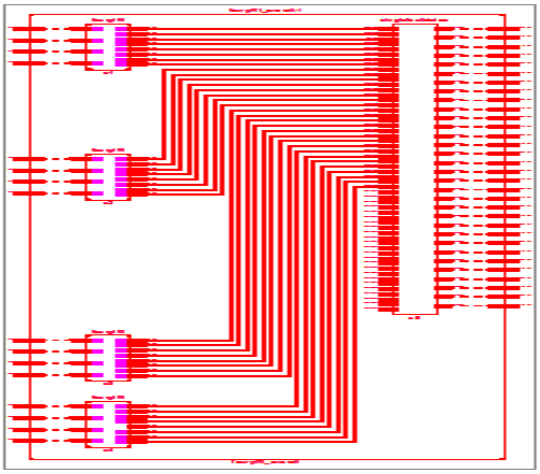

Figure 3: 4 Input 4-Point FFT RTL diagram using Xilinx.

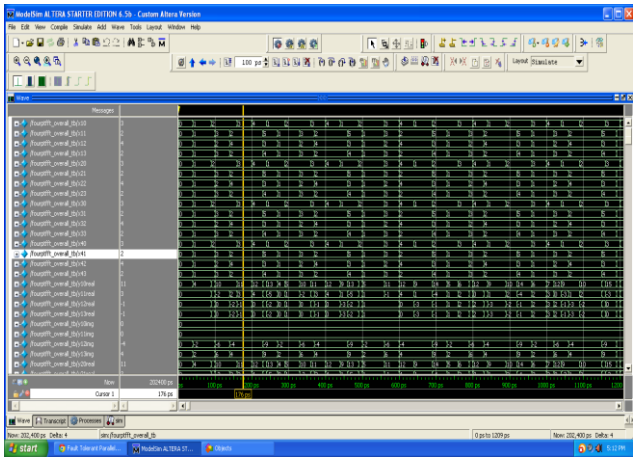

Figure 4: Simulation results for 4-Point FFT

Table 1: Error Location In The Hamming Code

c1c2c3	Error bit position
000	No error
111	z1
110	z2
101	z3
011	z4
100	z5
010	z6
001	z7

Evaluation

The one proposed schemes and the ECC scheme presented in [17] have been implemented on an FPGA and evaluated both in terms of overhead and error coverage. A four-point decimation-in-frequency FFT core is used to compute the FFT iteratively. This core has been developed to implement MIMO-OFDM for wireless systems. The number of FFT points is programmable and the rotation coefficients are calculated on-line for each stage and stored in registers. For the evaluation, a 1024 points FFT is configured with five stages calculation (log41024 = 5), so in total 5∗1024 = 5120 cycles are needed to calculate the FFT for 1024 input samples. The inputs are 12-bit wide and the outputs are 14-bit wide. For the redundant FFT, the bit widths are extended to 14 and 16 bit, respectively, to cover the larger dynamic range (as the inputs are the sum of several signals). To minimize the impact of round offs on the fault coverage, the outputs of the accumulator are 39-bit wide. To evaluate, the number of parallel filters considered should be large. This is done to compare the different techniques as a function of the number of parallel FFTs in the original system.

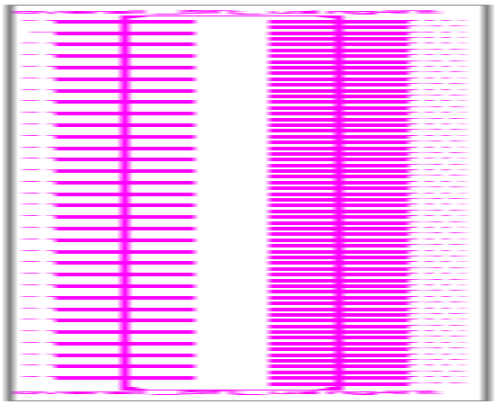

Figure 5: 8 Input 4-point FFT RTL diagram using Two 4 Input FFTs

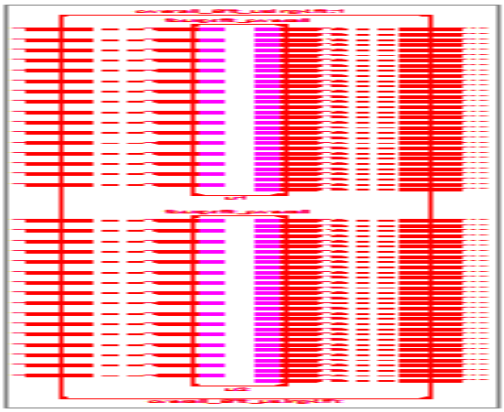

Figure 6: 8 Input 4-point FFT Inner RTL diagram using Two 4 Input FFTs

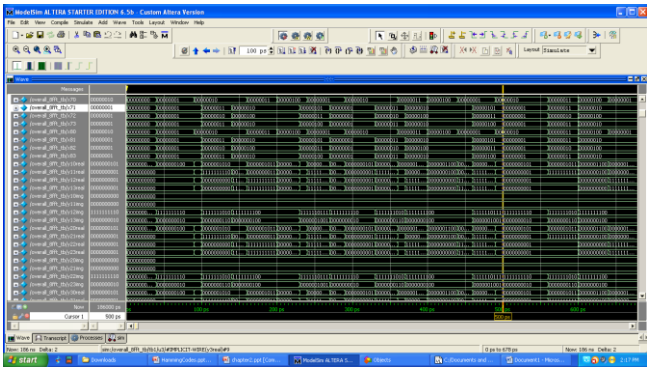

Figure 7: Simulation results for 8 Input using Two 4 Input 4-Points FFT

The error detection and correction blocks (Figs. 1) are implemented as multiplexers that select the correct output depending on the error pattern detected. As mentioned before, these blocks are tripled to ensure that errors that affect them do not corrupt the final outputs.

The FFT and the different protection techniques have been implemented using Verilog. Then, the design has been mapped to a Virtex-4 xc4vlx80 FPGA setting the maximum effort on minimizing the use of resources. The results obtained are summarized in Tables II.

Tables - 2

	4 input FFTs	**8 input Using 4 FFTs**	**8 input FFTs**
Slices	147	294	300
LUTs	269	538	549
BELs	763	1524	1535
Memory in KBs	16456	188092	168216

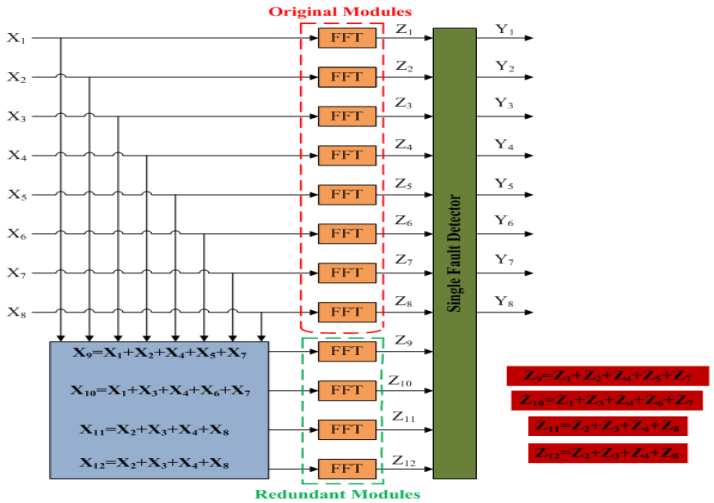

Figure 8: Parallel 8 - Point FFT protection using ECCs.

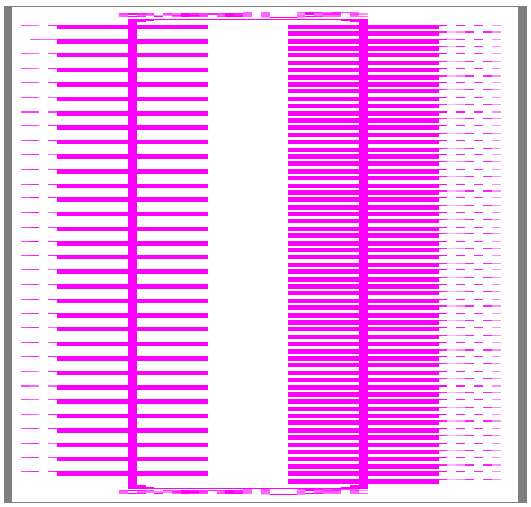

Figure 9: 8 Input 4-point FFT RTL diagram

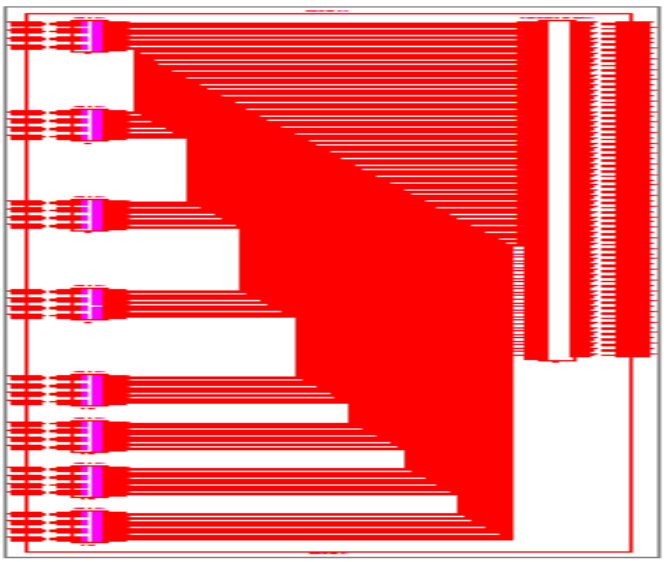

Fig. 10. 8 Input 4-point FFT Inner RTL diagram

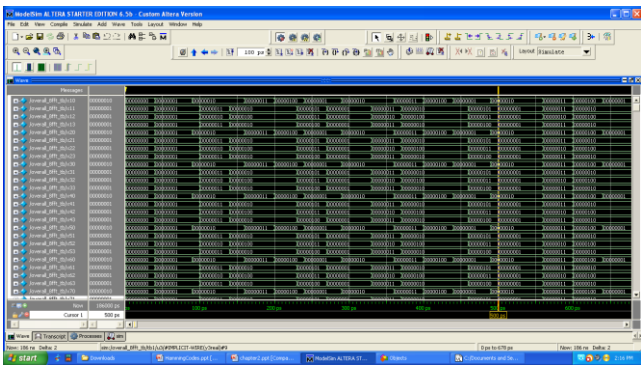

Figure 11: Simulation results for 8 Input using 4-Points FFT

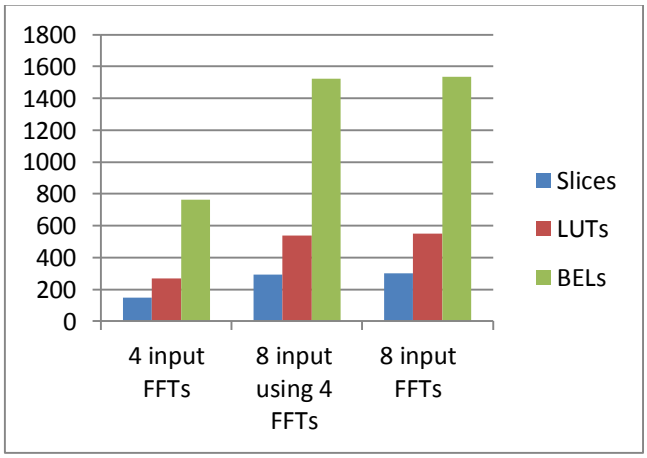

Figure 12: Resources Usage For Four And Eight Parallel FFTs

Conclusion

In this brief, the protection of parallel FFTs implementation against soft errors has been studied. Two techniques have been proposed and evaluated. The proposed techniques are based on combining an existing ECC approach. There are used to detect the errors and a simple parity FFT is used for correction. The detection of the errors can be done using ECC FFT. It provides the best results in terms of implementation complexity. In terms of error protection, fault injection experiments show that the ECC scheme can recover all the errors that are out of the tolerance range. The fault coverage for ECC scheme is ∼99. 9% when the tolerance level for ECC check is 1.

Acknowledgement

Authors deliver their graduate to Govt. of India and Financial Assistance from DST-FIST department. F. NO:SR/FST/Vel tech multi tech College-189/2013

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] B. Senthilkumar and **V. Rajamani**, A novel round keys generation technique for cryptography application, RP-CP4, National research scholar's seminar on green technologies in communication and computing, 3-4, Oct, 2013, St. Peters University, Chennai.

[5] B. Senthilkumar and **V. Rajamani** "A Novel Round keys Generation Technique for Cryptography applications", **International Journal of Technology and Engineering** Science [IJTES] Volume 1(7), pp1076-1080, October 2013.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] E. P. Kim and N. R. Shanbhag, “Soft N-modular redundancy, ” IEEE Trans. Comput., vol. 61, no. 3, pp. 323–336, Mar. 2012.
[11] J. Y. Jou and J. A. Abraham, “Fault-tolerant FFT networks, ” IEEE Trans. Comput., vol. 37, no. 5, pp. 548–561, May 1988.
[12] 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.
[13] P. P. Vaidyanathanm, Multirate Systems and Filter Banks. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
[14] A. Sibille, C. Oestges, and A. Zanella, MIMO: From Theory to Implementation. San Francisco, CA, USA: Academic, 2010.
[15] 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.
[16] 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.
[17] M. Ergen, Mobile Broadband—Including WiMAX and LTE. New York, NY, USA: Springer-Verlag, 2009.
[18] P. Reviriego, S. Pontarelli, C. J. Bleakley, and J. A. Maestro, “Area efficient oncurrent error detection and correction for parallel filters, ” IET Electron. Lett., vol. 48, no. 20, pp. 1258–1260, Sep. 2012.
[19] 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.
[20] R. W. Hamming, “Error detecting and error correcting codes, ” Bell Syst. Tech. J., vol. 29, no. 2, pp. 147–160, Apr. 1950.
[21] 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.
[22] Dr. Kuppusamy P. G “A New Power Gating Scheme Using Multi-Mode Power Switches For Static Power Reduction” in International Journal of Applied Engineering Research -2015.
[23] Dr. Kuppusamy P. G “Design Of Level Shifter Using Dual Cascode Voltage Switch For Low Power Application” in International Journal of Applied Engineering Research -2015.