International Journal of Engineering and Techniques - Volume 4 Issue 2, Mar – Apr 2018

RESEARCH ARTICLE OPEN ACCESS

Parallel Error Correction for Parallel FFTs

Dr.R.Naveen[1], L Pavithra[2]

[1]Associate Professor, Info Institute of Engineering, Coimbatore, Tamilnadu.

[2]M.E VLSI Design, Info Institute of Engineering, Coimbatore, Tamilnadu.

Abstract:

In modern electronic circuit soft error is reliability in pose. This protects against soft errors and is requirement for many applications. In this trend communication and signal processing systems are not exception. In some applications the interesting choice is used in algorithmic based fault tolerance (ABFT) technology which tries to detect the errors and to correct it. Communication and signal processing units are well suitable for ABFT. An example for the above mentioned ABFT is (FFT) is a building block in some application. An algorithm to compute discrete fourier transform is known as (**FFT**) Fast Fourier Transform. This FFT converts time domain signals into frequency domain. Hence FFT is used widely in DSP technology and also in rudimentary operations in the field of digital signal and image processing. It is indispensable in most signal processing operations. This paper consists of efficient multiplication technology to reduce the partial product which is banned in conventional multiplication technology. Hence FFT and inverse fast fourier transform (**IFFT**) with efficient multiplication and with increased speed is used for orthogonal frequency division multiplexing (**OFDM**) modulation and demodulation blocks. Normally I many application high speed and efficient multiplication is desired. Hence conventional multicarrier technology is chosen, but results in lower spectrum efficiency. So the principle of OFDM are used.

I. INTRODUCTION

A fast Fourier transform (FFT) is an algorithm that samples a signal by a period of time (or space) and divides it into its frequency components.[1] These components are single sinusoidal oscillations at distinct frequencies each with their own amplitude and phase. Over the time period measured, the signal contains 3 distinct dominant frequencies. Fourier analysis converts a signal from its original domain to a representation in the frequency domain and vice versa. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors.[2] As a result, it manages to reduce the complexity of computing the DFT from which appear if one simply applies the definition of DFT, to where is the data size. Fast Fourier transforms are widely used for many applications in engineering, science, and mathematics.

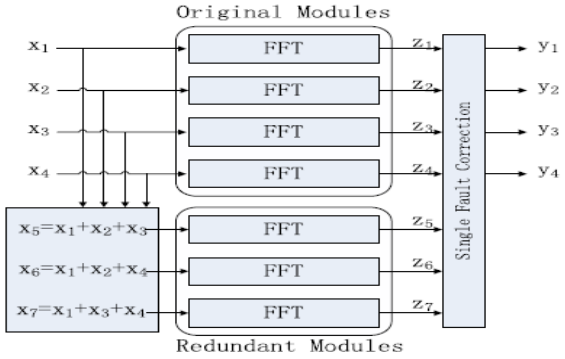

Fig 1 : Error Correction Code

Parallel FFT protection using ECC is shown above. The two suggested approach afford new choice to assure parallel FFTs that can be more efficient than protecting each of the FFTs

independently. The suggested schemes have been evaluated using FPGA implementations to assess the certainity overhead.Fault injection experiments have also been conducted to verify the ability of the implementations to detect and correct errors.The implementation overheads and fault tolerance of the schemes are evaluated[6]. Finally, the conclusions are drawn in . The complexity of communications and signal processing circuits boost 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 boost as technology scales. Soft errors can change the logical value of a circuit node creating a temporary error that can affect the system operation.[2] To ensure that soft errors do not affect the operation of a given circuit, a wide variety of techniques can be used. 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. [3]Finally, it is also possible to add redundancy at the system level to detect and correct errors.

TABLE I
ERROR LOCATION IN THE HAMMING CODE

$c_1 c_2 c_3$	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

The above mentioned tabular column is used to find the position of the error in the above said error correction code operation.

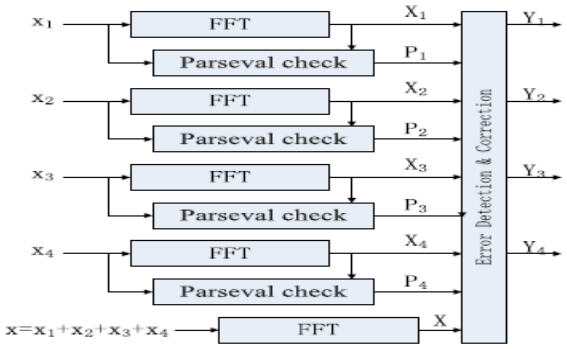

Fig 2 : Parity SOS

Parity SOS is shown above. The first proposed approach 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 of an error is detected (using *P*1, *P*2, *P*3, *P*4*)*, 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, *P*1 will be set and the error can be corrected by doing $X1c = X - X2 - X3 - X4$. (4) This bringing together of a parity FFT and the SOS check decreases the number of additional FFTs to just one and may, therefore, reduce the protection overhead. In the following, this approach will be referred to as parity-SOS (or first proposed technique). (OFDM) Orthogonal frequency division multiplexing has become a modulation technology for broadcasting wireless and wire line applications. It is also adopted for broadcasting audios (DAB) and digital terrestrial television broadcasting (DVB).OFDM is a special case of Multicarrier transmission, where a single data stream is transmitted over number of lower rate Subcarrier. The problem of inter symbol interference (ISI) introduced by multipath channel is significantly decreased in OFDM by using the cyclic prefix (CP) as a guard interval between OFDM blocks[1].

II. EXISTING METHODOLOGY

The main advantage over the first parity-SOS approach is to decrease the number of SOS checks needed. The error location process is the

same as for the ECC scheme and correction is as in the parity-SOS scheme.

2.1 PARITY-SOS-ECC

In the following, this scheme will be referred to as parity-SOS-ECC (or second proposed approach). The overheads of the two proposed schemes can be initially estimated using the number of additional FFTs and SOS check blocks needed. This information is summarized in Table II for a set of *k* original FFT modules assuming *k* is a power of two.

It can be observed that the two proposed schemes reduce the number of additional FFTs to just one. In addition, the second technique also reduces the number of SOS checks. In Section III, a detailed evaluation for an FPGA implementation is discussed to illustrate the relative overheads of the proposed techniques. 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. In the case of the redundant or parity FFTs,[5] an error will have no effect as it will not propagate to the data outputs and will not cause a correction.

In the case of SOS checks, an error will cause 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 can propagate errors to the outputs. In our implementations, those blocks are guarded with TMR. The same applies for the adders used to compute the inputs to the redundant FFTs or to the SOS checks in. 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 [4]. Therefore, to compare the three techniques for a given implementation, fault injection experiments should be done to determine the percentage. Architecture of the FFT implementation of the SOS check the errors that are actually corrected. This means that an evaluation has to be done both in terms of overhead and error coverage.

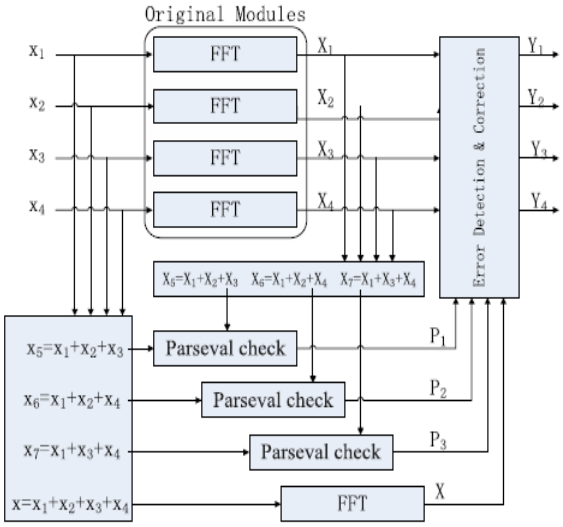

Fig 3 : Parity SOS ECC

2.2 EVALUTION

Proposed approach and the ECC approach presented in 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 implementation of the four-point FFT core is shown in[14]. 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). Since both the inputs and outputs to the FFT are sequential, the SOS check is also done sequentially using accumulators that are compared at the end of the block. To minimize the impact of roundoffs on the fault coverage, the outputs of the accumulator are 39-bit wide. For the

evaluation, several values of the number of parallel FFTs are considered[15]. This is done to compare the different approaches as a function of the number of parallel FFTs in the original system. The error detection and correction blocks (Figs. 1–3) 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[7]. Then, the design has been mapped to a Virtex-4 xc4vlx80 FPGA setting the maximum effort on minimizing the use of resources. The results show that the FFT is more complex.

III.PROPOSED METHODOLOGY

In this proposed scheme,the multiple bit has been detected and corrected in parallel FFTs by parallel correction method which is shown in the block diagram given below.

3.1 PARALLEL ERROR CORRECTION

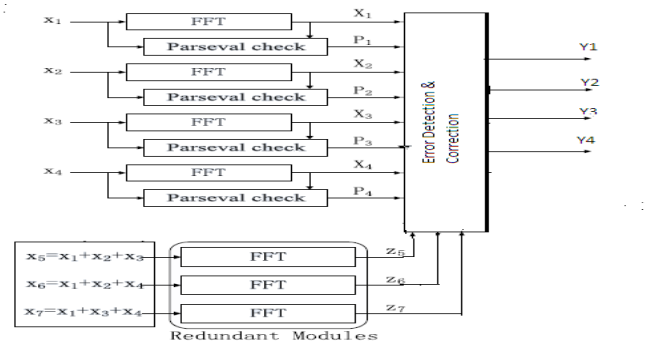

Fig 4 : Parallel Correction

In this proposed methodology, four number of fast fourier transform (FFT) Acts as original module. An Individual Perseval check are used for each FFT to detect the errors, hence this part acts as a parity SOS. X1,X2,X3&X4 are the input to the FFT of original module.

Three number of redundant module is used in this methodology to detect the additional errors occurred in original module. The input of redundant module is the linear combination of the original module‘s input.

They are

X5=X1+X2+X3 ---------- (1)

X6=X1+X2+X4 ---------- (2)

X7=X1+X3+X4 ---------- (3)

The output of redundant module is Z5, Z6&Z7 these outputs are given to the error detection and correction block. P1,P2,P3&P4 are the output of parseval check , this is used to detect the errors in the FFT hence all the FFT in this methodology consists of individual parseval checks.if the output of the parseval check is 1 the FFT contains of error. If the output of the parseval check is 0 the FFT error free.

For example If P1=1 & P2=1 error may be detected in the input X1 and X2

The above said statement is used to correct the detected errors.

X1=X5-X3-X2 ----------(4)

X1=X6-X2-X4 ----------(5)

X1=X7-X3-X4 ----------(6)

Three equations (4),(5),(6) are used to correct the detected error in these equation (6) “X1=X7-X3-X4” it is used for correcting the error because this equation does not consists of X2, hence it is selected for the error correction. Now equation (6) is substituted in (4)

“X1=X5-X3-X2” by rearranging this equation “X2=X5-X3-X1”. Through this the value of X1 & X2 the error has been corrected in the location of Z5, Z6 & Z7.After the block of error detection and correction ,the corrected output has seen in Y1,Y2,Y3& Y4.This multiple bit error detection and correction part is very useful in many applications .

The main objective of this design is an OFDM transmitter and receiver using FPGA. In transmitting end, the OFDM signal is generated by implementing the IFFT. In receiving end, the FFT is implemented. Hence this FFT acts as an important role in communication processing system.

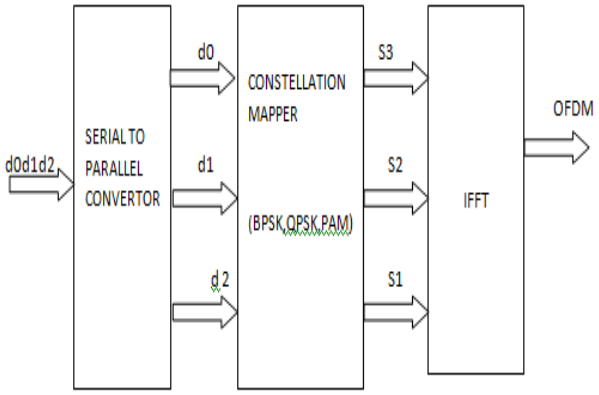

Fig 5:Implementation Of Digital OFDM

Implementation of Digital OFDM objective is to use High-Speed Integrated-Circuit to produce VHDL codes that carry out FFT and IFFT function.

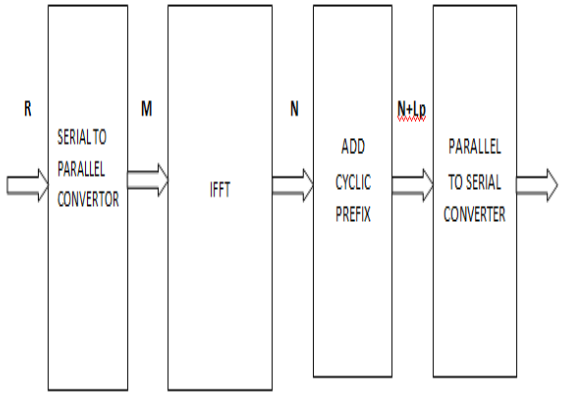

Fig 6:OFDM transmitter

The OFDM transmitter communication system is shown above. The main parts are the FFT and IFFT part of the OFDM system. The input are fed to the transmitter in sequence at R symbols second. These symbols pass through a serial to parallel converter and output data on M lines in parallel.An IFFT converts a number of complex data points, of length that is power of 2, into the same number of points in time domain. The output is N time-domain samples. In order to preserve the sub-carrier orthogonality a cyclic guard interval is introduced[8]. The M symbols are sent to an IFFT block that performs N point IFFT operation.

The IFFT transform a spectrum (amplitude and phase of each component) into a time. Domain signal In this case, assumed a cyclic prefix of length Lp samples is pre-pended to the N samples. For example, assume N=4 and Lp=2; if the outputs of a 4 point inverse Fourier transform is [1 2 3 4], the cyclic prefix will be [3 4]. The cyclically elongated symbol would be [3 4 1 2 3 4]. Therefore, the length of the transmitted OFDM symbol is N+Lp. Prependin the cyclic prefix aids in removing the effects of the channel at the receiver. ISI can take place when multi path channel cause delayed version of previous OFDM symbol to corrupt the current received symbol. ISI will affect the cyclic prefix[9]. When the value of Lp is greater than or equal to the size of the transmission channel.

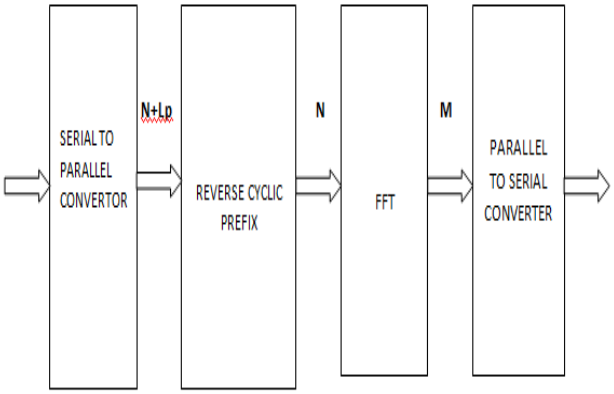

Fig 7:OFDM receiver

The received symbol is in the time domain and is distorted to cause the channel. The received signal force through a serial to parallel converter and cyclic prefix removal. After the cyclic prefix removal, the signals are passed through an N205 point of fast Fourier transform to convert the domain signal to frequency domain[9]. The output of the FFT is formed from the first M samples of the output. The work of the project is focused on the design and implementation of FFT for a FPGA kit. The direct mathematical derivation method is used for this design. In this paper project the coding is done in VHDL & the FPGA synthesis and logic

simulation is done using Xilinx ISE Design Suite. An efficient pipelined FFT Processor for OFDM communication systems

To Implement the OFDM block by block and finally Incorrect all of them together to form complete OFDM Circuit. Here Proposed an optimized implementation of the 8-point FFT processor with radix-2 algorithm in R2MDCarchitecture.The butterfly Processing Element (PE) is used in the 8-FFT processor decreases the multiplicative complexity by using a real constant multiplication in one method and eliminates the multiplicative complexity by using add and shift operations in the proposed method. So they proposed conventional FFT algorithm by using butterfly technique and then proposed algorithm has been implemented by using R2MDC architecture[10]. Then the analysis and design of FFT is done using VHDL.

A paper on Reconfigurable VLSI Architecture for FFT Processor This paper presents a CORDIC (Coordinate Rotation Digital Computer)-based split-radix fast Fourier transform (FFT) core for OFDM systems These FFT processors outperform the conventional ones in terms of both power consumption and core area[11]. implemented the core processing blocks of an OFDM system, namely FFT and IFFT. Instead of the DFT and IDFT, FFT and IFFT have been chosen to implement the layout because they offer high speed with less computational time. The FPGA implementation of the project is performed using VHDL. The performance of the coding is analyzed from the result of timing simulation using Xilinx[12]. Developed FFT &IFFT algorithms to be used in OFDM systems. The speed enhancement is the key contribution of the main processing blocks in OFDM system so they have successfully implemented the 8- point IFFT & FFT algorithm using VHDL to be used in the architecture of OFDM transmitter & receiver Pipeline architecture for WiMAX technology using Radix-4Decimation in frequency FFT algorithm [13]. They proposed a memory based recursive FFT design which has much less gate counts, lower power consumption and higher speed[15].

IV. SIMULATION AND RESULTS:-

4.1 Simulation result of parallel correction:-

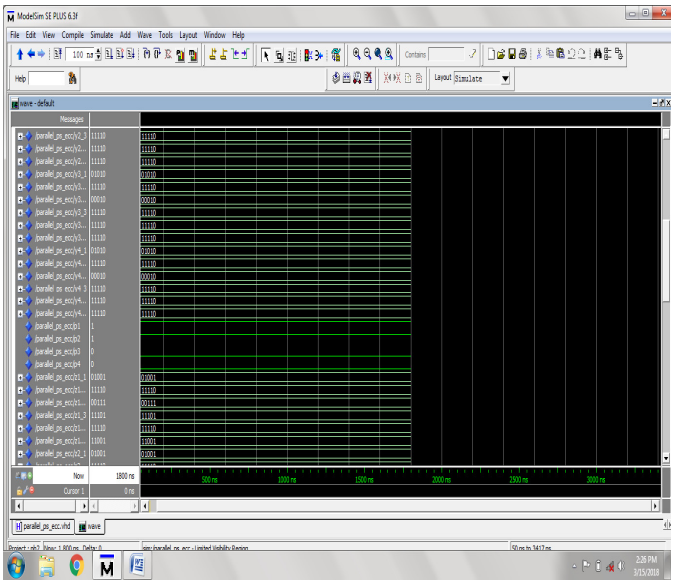

Fig 8:parallel error location

Area analysis:-

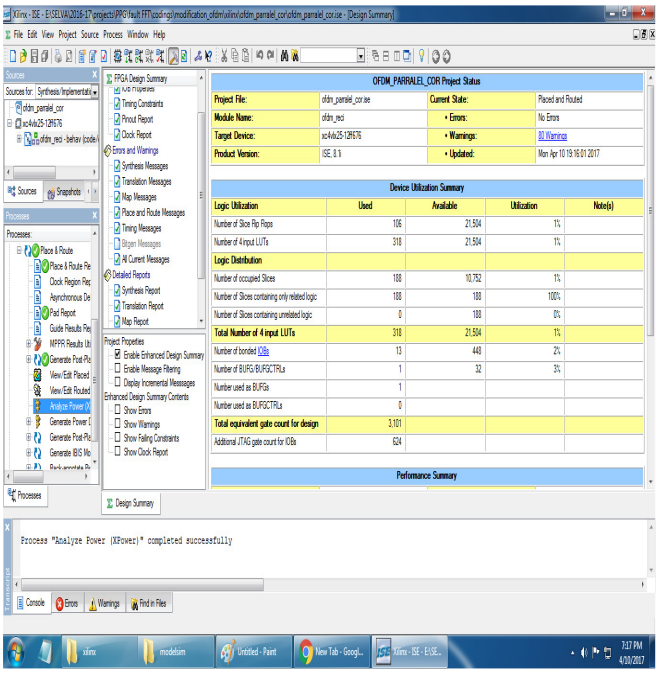

Fig 9: Area Analysis

Delayanalysis

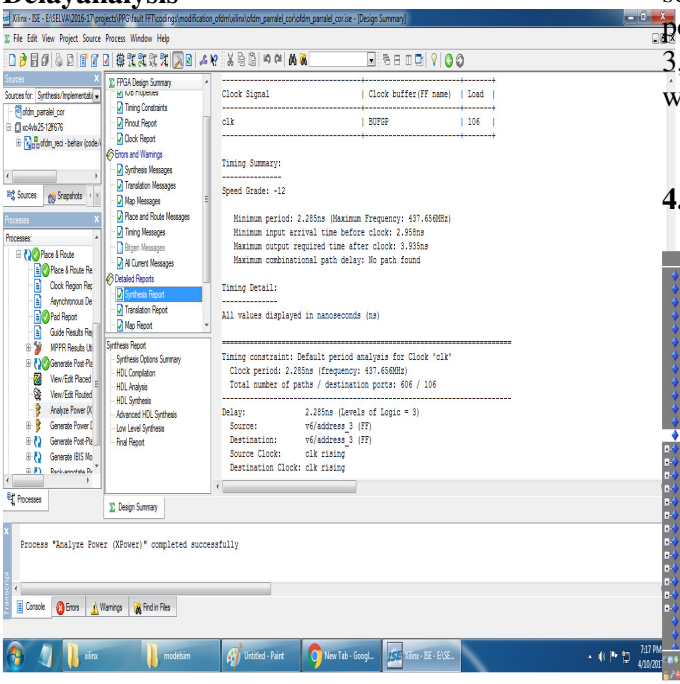

Fig 9: Delay Analysis

Power analysis:-

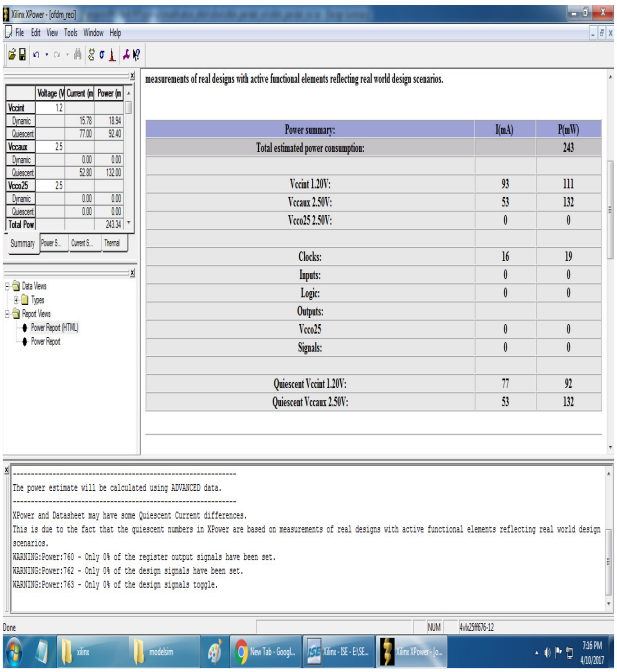

Fig 10:Power Analysis

This section deals with simulation result and discussion of the existing Parity-SOS-ECC. The software tool used is Xilinx. The simulation and power analysis as 243 mW and area analysis as 3,101 and delay analysis as 2.285 ns are obtained with the help of Xilinx tool.

4.2 Simulation result of OFDM :-

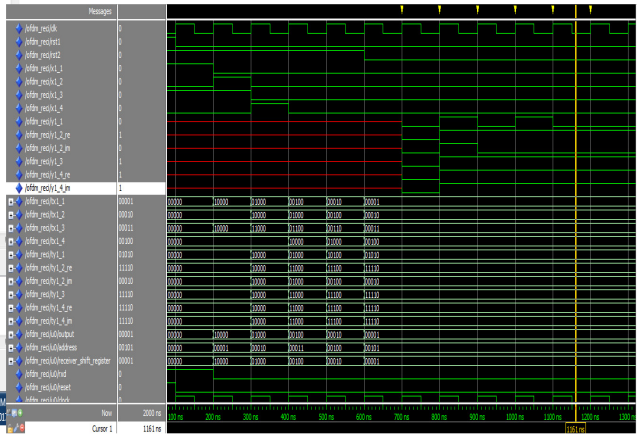

Fig 11 :OFDM error detection and correction

V. CONCLUSION

In OFDM receiver multiple blocks of error correction and detections are enrolled in FFT block of the OFDM receiver to obtain parallel and serial outputs. We could propose a 32-point FFT & IFFT design for communication application like OFDM. The main purpose of the proposed architecture is to design efficient multiplication of FFT & IFFT using Vedic multiplication. It has numerous advantages such as: increase the speed, efficient timing, and better resource utilization parameter. In summary, speeds performance of our design easily satisfies most application requirements based on OFDM modulated wireless communication system.

REFERENCES

1)Md. Mahmudul Hasan "Performance Analysis of Wavelet Based Multiuser MIMO OFDM" *World Academy of Science, Engineering and Technology International Journal of Computer, Information, Systems and Control Engineering Vol: 7 No: 3, 2013.*

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

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

4)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.

5) 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.

6)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.

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

8)Zheng Wang, Mingke Dong, Yuping Zhao, "Design and implementation of efficient FFT processor for multicarrier system," *CanadianConference on Electrical and Computer Engineering,* pp. 1384-1387, May 2005.

9)Nilesh Chide, Shreyas Deshmukh, Prof. P. B. Borole, "Implementation of OFDM System using IFFT and FFT," *International Journal of Engineering Research and Applications (IJERA),* Vol. 3, pp. 2009-2014, Jan. –Feb. 2013.

10) Pradeepa M., Gowtham P., "Optimized Implementation of FFT Processor for OFDM system," *International Journal of Advances in Engineering & Technology (IJAET),*vol. 3, pp. 429-441,May 2012.

11) Manjunath Lakkannavar, Ashwini Desai, "Design and Implementation of OFDM (Orthogonal Frequency Division Multiplexing) using VHDL and FPGA," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 1, pp. 211-213, Aug. 2012.

12) Verma, P., Kaur, H., Singh, M., Singh, B., "VHDL Implementation of FFT/IFFT Blocks for OFDM," *International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 186-188, 27-28 Oct. 2009.

13)N. Mahdavi, R. Teymourzadeh, IEEE Student Member, Masuri Bin Othman, "VLSI Implementation of High Speed and High Resolution FFT Algorithm Based on Radix 2 for DSP 208 Application," *The 5th Student Conference on Research and Development*, pp. 1-4,11-12 Dec. 2007.

14) Haveliya A., "Design and Simulation of 32-Point FFT Using Radix-2 Algorithm for FPGA Implementation," *Second International Conference on Advanced Computing & Communication Technologies (ACCT),* pp. 167-171, 7-8 Jan. 2012.

15) Correa I. S., Freitas L.C., Klautau, A., Weyl Albuquerque Costa J.C., "VHDL Implementation of a Flexible and Synthesizable FFT Processor," *IEEE Latin America Transactions*, vol. 10, pp. 1180-1183, Jan. 2012.