Encoder Decoder Design of Extended Golay Codec and Simulation for High Speed Applications

Dhanraj singh yadav[1], Aman Saraf[2], Pushpraj Tanwar[3]

[1,2,3]*Departmant of Electronics and Communication Engineering, Radharaman Institute of Technology, RGPV, Bhopal, India*

Abstract - In wireless communication systems the most important issue to be considered is the ability of the receiver to detect the errors and correct them from the received information, so as to provide correct information data to the processor. A number of different methods are available to implement the hardware and software with such preference. But, when the length of the communication link becomes very long, i.e., the distance between the wireless transmitter and receiver is very large, the effect of noise on the transmitted signal may cause a change in multiple bits of the transmitted information. This can cause drastic loss in many cases. In this brief a Field Programmable Gate Array (FPGA) based design and simulation of Golay Code (G23) and Extended Golay Code (G24) Encoding scheme are presented. This work is based on the optimization of the time delay of the operational circuit to encode a data packet using the Golay Encoder.

Keywords: Golay code, extended golay code, encoder Algorithm, Decoder algorithm, Xilinx ISE.

I. INTRODUCTION

In day to day life use of cell phone, computers, satellite and other devices that is used to send messages through channel is becoming a major part of humans life so we say that digital communication is a important part for us. Unfortunately, most types of communication are subject to noise, which may reason for the presence of errors in the messages that are being sent. Particularly when sending messages is a complicated or expensive task, for example in satellite communication, it is important to find ways to moderate the occurrence of errors as much as possible. This is the central idea in coding theory: what we have received and what message was being sent? To make this problem uncomplicated we use error-correcting codes. The foremost idea is to add redundancy to the messages which enables us to both recognize and correct the errors that may have occurred. This paper presents a particular type of error-correcting codes, the extended Golay code G24. The extended Golay code was used for sending images of Jupiter and Saturn from the Voyager 1 and 2.The information is transfer in three steps a source sends, a channel transmits, and a receiver receives. There is an option that at the time of transmission the information is altered to noise so to avoid this condition we use error correction codes. Fig. 1 show that a message is encoded into a codeword, it is sent to the receiver through a channel, in this channel the chance exists that errors occur, and the receiver tries to obtain the original message by decoding the word.

Golay code is an error correcting code which is used to specifies that what we have received and what is send .some of the most important properties of such codes which allows us to give a detailed description of the extended Golay is:

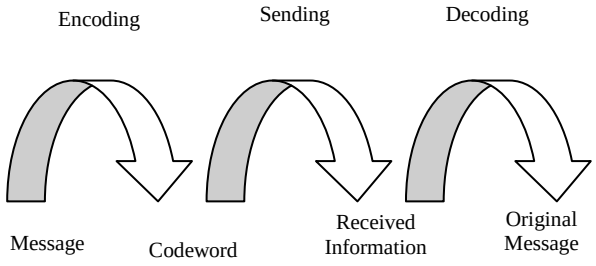

Fig 1. Process of Error Correction Code

Firstly a message m of length k is a sequence of k symbols out of some finite field F, so m = (m1 : : :mk) belongs to F^k. Then an n-code C over a finite field F is a set of vectors in F^n, where $n \leq k$. Since we will be commerce with a binary code only, we will assume codes are binary from now on. Second property says that the error probability p is the probability that 1 is received when 0 was sent, or 0 is received when 1 was sent. Third property says that the hamming weight of a vector belongs to a function F^n is the number of its non zero elements. Fourth property says that the humming distance of two vectors related to a function F^n is the number of place where they differ. The idea is that an n-code C is a strict subset of F^n in which we want the Hamming distance between any two vectors to be as large as possible. Therefore, the minimum Hamming distance is an important characteristic of the code. Fifth property says that the minimum Hamming distance d of a code C is defined as d = min {dist(x, y) I x, y belongs to C} where c is the code. The description of work in this paper is

arranged as follows: Section-II gives an overview on the work performed by other scholars in Golay Code implementation and applications. Introduction on Golay code and its encoding algorithm is described in Section-III. Section-IV presents the simulation and synthesis results of the performed work. The conclusion based on the proposed work and the future work scope is presented in Section-V. In the last the references are mentioned.

II. LETERATURE REVIEW

In reference [1] the proposed paper addresses error correcting phenomena using Golay code encoder. The algorithm of encoding data for error detection and correction was originally proposed by Marcel J. E. Golay in 1949 [2]. A brief introduction and explanation of Golay coding scheme is presented in [3]. An FPGA based 4-bit Golay Encoder and Decoder design and implementation with complete verification of multiplication is simulated in [4] using Xilinx ISE and Models in Tools. Reference [5] presents a soft algorithm based decoding orientation to hardware implementation of (24, 12, 8) Golay code with implementation of the algorithm on FPGA. In [6] it is shown that the (24, 12, 8) Golay code can be constructed as a direct sum of two array codes that involve four component codes from which two are simple linear block codes (repetition code and SPC code). Construction of Golay Code Complementary Sequences is presented in [7] for application of Golay Coding in the fields of physics, combinatory (orthogonal design and Hadmard matrices), surface acoustics and tele-communication. A modified algorithm for decoding Binary Golay Code is presented in [8] which is based on one-to-one mapping between the syndrome “S1” and correctable error patterns. In this proposed work the algorithm determines the error locations directly by using look-up tables without the multiplication operation over a finite field. This algorithm has been verified by the scholars on a C-language based software simulation platform. The work presented in [9] focuses on Golay code decoding using symbol-by-symbol soft-in/soft-out APP (a posteriori probability) algorithm through co-set based technique. A study based on discussion on the error correction capability of MSK modulation with Golay code and BPSK modulation with Golay code is presented in [10], which concludes that MSK Golay code is comparatively more robust. A technique based upon reversing the conventional scheme of Golay code (24, 12, 8) that convert 24-bit vectors into 12-bit message words is presented by reference [11] to improve the look for operation when multi-attribute objects are partially distorted. The work in [12] presents invention of Doppler Resilient waveforms by means of Golay matching sequence which have ideal uncertainty along the zero Doppler axis but are sensitive to non-zero Doppler shifts. The work in [13] presents Golay code transformation for Ensemble Clustering in application to Medical Diagnostics. The work proposed in this paper shows clustering methodology is unique to outperform all other conventional techniques because of its linear time complexity. Reference [14] proposed an error correction Golay code for clustering huge amount of Big data Streams by using error correction Golay codes and this method is used in the field where the requirement to accumulate multidimensional data. In Reference [15] the proposed methodology accomplish the requirement of the reducing peak to average ratio (PTAR) with the help of special Fractional Fourier Transform (FRFT) followed to the low complicity Golay sequence coder in order to offer optimal de-correlation between signal and noise. To achieve the condition of low complexity, low bit error rate and peak to average power ratio. Reference [16] proposed an algorithm for the hardware implementation of (24, 12, 8) Golay code in FPGA (Field programmable gate array) based system. To get rid of the complexity of arithmetic operations this arises in the existing algorithm. The proposed algorithm the chooses absolute value rather than bit error probability to receive better results as compared to the existing algorithms. Reference [17] proposes a new algorithm to accomplish the requirement of faster decoding for the Gosset Lattice, Golay code and Leech Lattice. The projected design presents two approaches to first when charge in of length n and taking soft decoding algorithm at an arbitrary point R^n in to the adjacent code word and second a decoding algorithm for a lattice A in R^n changes an arbitral point of R^n into a closest lattice point. Reference [18] proposed an efficient soft-decision decoder of the (23, 12, 7) binary Golay code up to the four errors and almost all pattern of three errors and all fewer random error can be corrected with the help of proposed algorithm.

III. GOLAY CODE ENCODER ALGORITHM

A binary Golay code is represented by (23, 12, 7), which depicts the minimum distance between two binary Golay codes is 7 and the message is of 12-bits while the length of codeword is 23 bits. It is necessary to construct binary codes in a Galois Field (GF). Binary field is denoted by GF(2), which supports different binary arithmetic operations. Generator polynomial is needed by the generation of coding sequence. The possible generator polynomials [13] over GF (2) for Golay (23, 12, 7) code are $x11 + x10 + x6 + x5 + x4 + x2 + x1$ and $x11+x9+x7+x6+x5+x1+1$. AE3h is considered as the characteristic polynomial in this paper. Using this polynomial a 12-bit binary number can be encoded into a 23-bit Golay code by performing the long division to generate check bits (11-bit). The 12-bit data and the 11-bit check-bit information together make the Golay code of the 12-bit information data. The extended Golay code (24,

12, 8) is not much different it can be generated by appending a parity bit with the binary Golay code or using a generator matrix *G*, which is defined as [*I*, *B*] or [*B*, *I*], where *I* denotes an identity matrix of order 12. The matrix B is shown in fig. 2. B*i* represents i^{th} row of the matrix B.

$$B = \begin{bmatrix} 110111000101 \\ 101110001011 \\ 011100010111 \\ 111000010111 \\ 110001011011 \\ 100010110111 \\ 000101101111 \\ 001011011101 \\ 010110111001 \\ 101101110001 \\ 011011100011 \\ 111111111110 \end{bmatrix}$$

Fig. 2 Matrix-B

The algorithmic steps that are required to accomplish the encoding process are enlisted as follows:

1) For check bits generation a characteristic polynomial is chosen.
2) The data 'M' participate in long division method with the characteristic polynomial. So, 11 zeros are appended to the right of data message M.
3) For *G*23 check bits are generated by division operation it can be assumed the most significant bit (MSB) resulted at the end of the division operation.
4) Appending check bits with the message gives us the encoded Golay (23, 12, 7) codeword.
5) For conversion of binary Golay code into extended binary Golay code (24, 12, 8) a parity bit is added. If the weight of binary Golay code is even, then parity bit 0 is appended, otherwise 1 is appended.

The Golay code encoding is explained using example in fig. 3.

Data (12-bit)	Appended zeros	Encoder Operation
101000100111	00000000000	
101011100011		… xor
0000110001000000		… 4-time shift
101011100011		… xor
0110101000110		… 1-time shift
101011100011		… xor
0111101001010		… 1-time shift
101011100011		… xor
0101101010010		… 1-time shift
101011100011		… xor
000110110001000		… 3-time shift
101011100011		… xor
0111011010110		… 1-time shift
101011100011		… xor
010000110101		
<Check-bits>		

Fig 3 Long Division of Data for Check bits generation

In the example that is shown in fig. 3, the 12-bit data is represented by 101000100111 and the characteristic polynomial is represented by 101011100011. The check-bit sequence (11-bits) that is generated by long division operation is represented by is 10000110101. The 23-bit encoded Golay codeword (G23) is represented by 101000100111-10000110101. This encoded word be converted by appending a parity bit in it. In the G(23) word the weight is 11, i.e., the encoded word has 11 1's, so a 1 will be appended in it. This will produce extended codeword G(24) as (101000100111-10000110101-1). The parity bit generation is done by implemented by XOR-ing the bits of G(23) codeword. This is depicted in fig. 4.

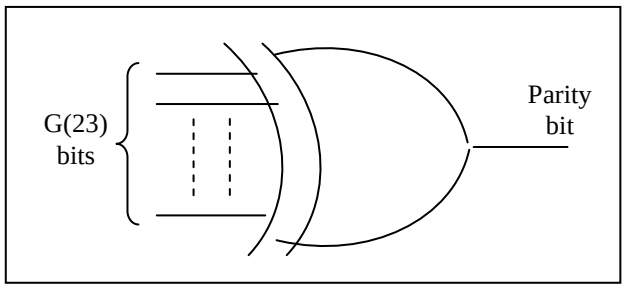

Fig 4 Parity Bit Generation using XOR-ing operation

The generated extended Golay codeword is validated by measuring the weight of the G(24) codeword. In a valid G(24) codeword the weight is multiple of 4 and greater than equal to 8. In the example, the weight of the G(24) codeword is 12, so it is a valid codeword.

IV. GOLAY CODE DECODER ALGORITHM

$$S[11] = W[23] \text{ xor } W[11] \text{ xor } W[10] \text{ xor } W[8] \text{ xor } W[7] \text{ xor } W[6] \text{ xor } W[2] \text{ xor } W[0]$$

$$S[10] = W[22] \text{ xor } W[11] \text{ xor } W[9] \text{ xor } W[8] \text{ xor } W[7] \text{ xor } W[3] \text{ xor } W[1] \text{ xor } W[0]$$

$$S[9] = W[21] \text{ xor } W[10] \text{ xor } W[9] \text{ xor } W[8] \text{ xor } W[4] \text{ xor } W[2] \text{ xor } W[1] \text{ xor } W[0]$$

$$S[8] = W[20] \text{ xor } W[11] \text{ xor } W[10] \text{ xor } W[9] \text{ xor } W[5] \text{ xor } W[3] \text{ xor } W[2] \text{ xor } W[0]$$

$$S[7] = W[19] \text{ xor } W[11] \text{ xor } W[10] \text{ xor } W[6] \text{ xor } W[4] \text{ xor } W[3] \text{ xor } W[1] \text{ xor } W[0]$$

$$S[6] = W[18] \text{ xor } W[11] \text{ xor } W[7] \text{ xor } W[5] \text{ xor } W[4] \text{ xor } W[2] \text{ xor } W[1] \text{ xor } W[0]$$

$$S[5] = W[17] \text{ xor } W[8] \text{ xor } W[6] \text{ xor } W[5] \text{ xor } W[3] \text{ xor } W[2] \text{ xor } W[1] \text{ xor } W[0]$$

$$S[4] = W[16] \text{ xor } W[9] \text{ xor } W[7] \text{ xor } W[6] \text{ xor } W[4] \text{ xor } W[3] \text{ xor } W[2] \text{ xor } W[0]$$

$$S[3] = W[15] \text{ xor } W[10] \text{ xor } W[8] \text{ xor } W[7] \text{ xor } W[5] \text{ xor } W[4] \text{ xor } W[3] \text{ xor } W[0]$$

$$S[2] = W[14] \text{ xor } W[11] \text{ xor } W[9] \text{ xor } W[8] \text{ xor } W[6] \text{ xor } W[5] \text{ xor } W[4] \text{ xor } W[0]$$

$$S[1] = W[13] \text{ xor } W[10] \text{ xor } W[9] \text{ xor } W[7] \text{ xor } W[6] \text{ xor } W[5] \text{ xor } W[1] \text{ xor } W[0]$$

$$S[0] = W[12] \text{ xor } W[11] \text{ xor } W[10] \text{ xor } W[9] \text{ xor } W[8] \text{ xor } W[7] \text{ xor } W[6] \text{ xor } W[5] \text{ xor } W[4] \text{ xor } W[3] \text{ xor } W[2] \text{ xor } W[1]$$

Fig. 5 Measurement of bits of Syndrome 'S'

The algorithm that is used to implement the decoder design and simulation is based on Imperfect Maximum Likelihood Decoder (IMLD) design. The algorithm assumes that 'W' is the received codeword with 'E' be the error pattern. Two syndromes are calculated using the received codeword; S and SB. Syndrome S is evaluated by performing multiplication of received codeword W with parity check Matrix 'H' which is given by as [*I*, *B*] or [*B*, *I*]. Therefore, the bits of syndrome 'S' are calculated using the syndrome measurement equations as shown in fig. 5.

In the error decoding process calculation of weight of (S + B*i*) and (SB + B*i*) is required to identify the error pattern. To calculate (S + B*i*), for $1 \le i \le 12$, the implemented hardware use bit inversion of 'S' as per the logic shown in fig. 6.

$$S + B_1 = \{ \sim S[11], \sim S[10], S[9], \sim S[8], \sim S[7], \sim S[6], S[5], S[4], S[3], \sim S[2], S[1], \sim S[0] \}$$

$$S + B_2 = \{ \sim S[11], S[10], \sim S[9], \sim S[8], \sim S[7], S[6], S[5], S[4], \sim S[3], S[2], \sim S[1], \sim S[0] \}$$

$$S + B_3 = \{ S[11], \sim S[10], \sim S[9], \sim S[8], S[7], S[6], S[5], \sim S[4], S[3], \sim S[2], \sim S[1], \sim S[0] \}$$

$$S + B_4 = \{ \sim S[11], \sim S[10], \sim S[9], S[8], S[7], S[6], \sim S[5], S[4], \sim S[3], \sim S[2], S[1], \sim S[0] \}$$

$$S + B_5 = \{ \sim S[11], \sim S[10], S[9], S[8], S[7], \sim S[6], S[5], \sim S[4], \sim S[3], S[2], \sim S[1], \sim S[0] \}$$

$$S + B_6 = \{ \sim S[11], S[10], S[9], S[8], \sim S[7], S[6], \sim S[5], \sim S[4], S[3], \sim S[2], \sim S[1], \sim S[0] \}$$

$$S + B_7 = \{ S[11], S[10], S[9], \sim S[8], S[7], \sim S[6], \sim S[5], S[4], \sim S[3], \sim S[2], \sim S[1], \sim S[0] \}$$

$$S + B_8 = \{ S[11], S[10], \sim S[9], S[8], \sim S[7], \sim S[6], S[5], \sim S[4], \sim S[3], \sim S[2], S[1], \sim S[0] \}$$

$$S + B_9 = \{ S[11], \sim S[10], S[9], \sim S[8], \sim S[7], S[6], \sim S[5], \sim S[4], \sim S[3], S[2], S[1], \sim S[0] \}$$

$$S + B_{10} = \{ \sim S[11], S[10], \sim S[9], \sim S[8], S[7], \sim S[6], \sim S[5], \sim S[4], S[3], S[2], S[1], \sim S[0] \}$$

$$S + B_{11} = \{ S[11], \sim S[10], \sim S[9], S[8], \sim S[7], \sim S[6], \sim S[5], S[4], S[3], S[2], \sim S[1], \sim S[0] \}$$

$$S + B_{12} = \{ \sim S[11], \sim S[10], \sim S[9], \sim S[8], \sim S[7], \sim S[6], \sim S[5], \sim S[4], \sim S[3], \sim S[2], \sim S[1], S[0] \}$$

Fig. 6 Calculation of (S + B*i*)

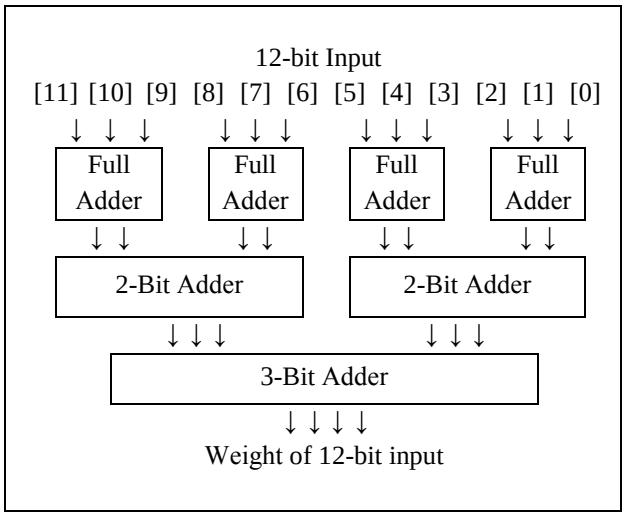

Fig. 7 Architecture of Weight Calculation Unit.

Simplified adder based architecture is implemented in this work to calculate the weight of a 12-bit (S + B*i*) and (SB + B*i*), for $1 \le i \le 12$. The architecture of adder based weight calculation unit is shown in fig. 7.

The algorithmic steps that are required to accomplish the decoding process are enlisted as follows:

1) Compute the Syndrome 'S' from the received codeword 'W' and matrix 'H', where H = [I / B]
2) If weight of 'S' is less than or equal to 3, i.e., wt(S) ≤ 3, then error vector, E = [S, 0]
3) If wt(S+B*i*) ≤ 2, then E = [S+B*i*, I*i*]. Where I*i* represents i^{th} row of the identity matrix I.
4) Compute the second syndrome SB
5) If wt(SB) ≤ 3, then E = [0, SB]
6) If wt(SB+B*i*) ≤ 2, then E = [I*i*, S+B*i*]

If E is still not determined then received data is required to be retransmitted.

V. SIMULATION AND SYNTHESIS RESULTS

The present work is simulated using Xilinx. The RTL Schematic diagrams of Encoder and Decoder designs are shown in Fig-5 and Fig-6 respectively.

The Encoder and Decoder simulation waveforms are shown in Fig-7 and Fig-8 respectively. A 12-bit data is used to encode using the proposed encoder. The input data bits are followed by logic-'0' inputs. The FPGA based hardware utilization summary of the proposed Encoder and Decoder designs is presented in Table-I and Table-II respectively.

TABLE I. HARDWARE UTILIZATION SUMMARY OF ENCODER

Vertex-IV XC4VLX160-12FF1148	**Total**	**12-bit Golay Encoder**	
		Used	**%**
Slices	67584	47	0
Flipflops	135168	44	0
LUTs 4-Inputs	135168	89	0
Bonded IOBs	768	50	6

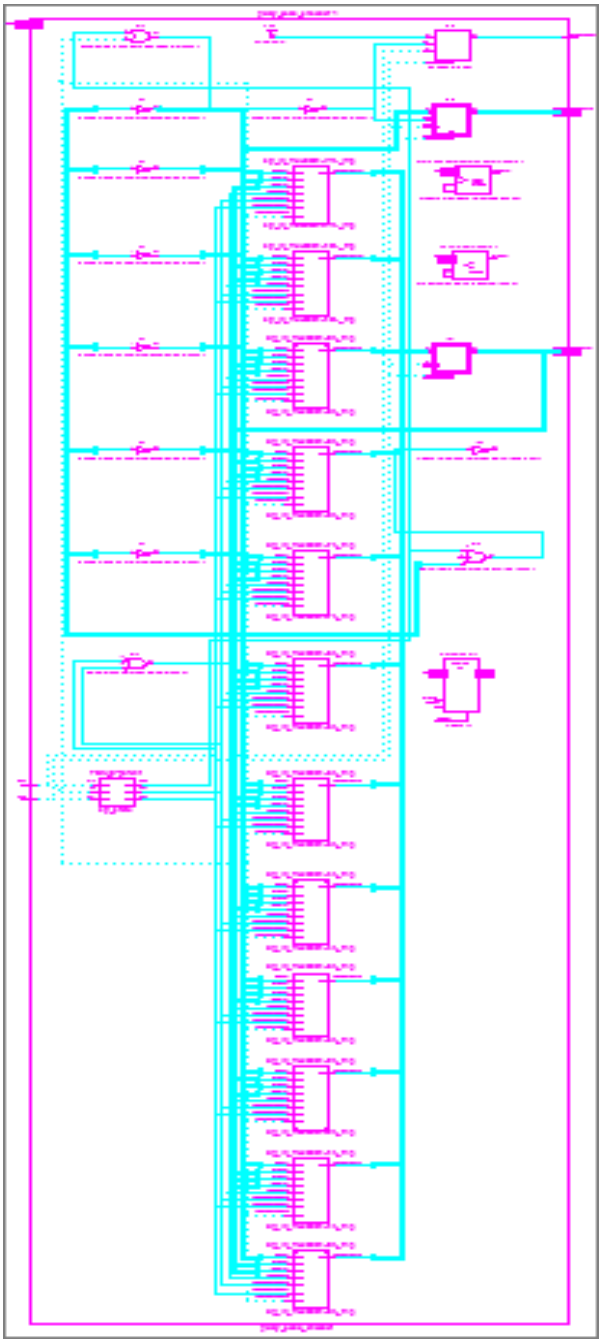

Fig. 5 RTL Schematic Diagram of Proposed Golay Code (24, 12, 8) Encoder

TABLE II. HARDWARE UTILIZATION SUMMARY OF DECODER

Vertex-IV XC4VLX160-12FF1148	Total	12-bit Golay Decoder	
		Used	%
Slices	67584	360	0
Flipflops	135168	305	0
LUTs 4-Inputs	135168	695	0
Bonded IOBs	768	55	6

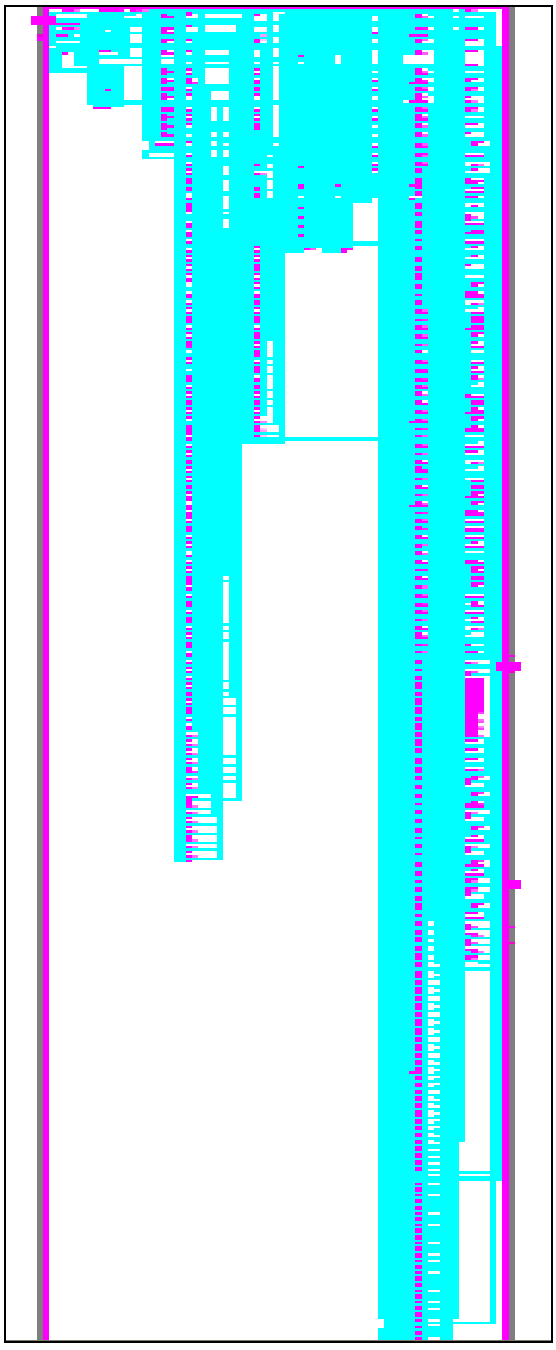

Fig. 6 RTL Schematic of Proposed Golay Code (24, 12, 8) Decoder

TABLE II. DYNAMIC POWER CONSUMPTION OF PROPOSED DESIGN

Work	Operational Frequency (MHz)	
	Encoder	Decoder
Proposed	383.245	311.491
[1]	238.575	195.082
[5]	-	100

Table-III represents a comparative analysis of the delay based results of the proposed work with some existing works. (Please refer Fig. 7 and Fig. 8 at the end of research paper)

VI. CONCLUSION

Delay optimized hardware architecture for extended binary Golay encoder and decoder are designed and simulated in the proposed work. The results obtained from the design synthesis for encoder and decoder supersedes the reference schemes in term of the operational frequency. This makes the proposed design a good option to be used in the high speed application based configurable circuits. In future there is a great scope to further optimize the performance of the proposed algorithm. In future the scholars may undertake the challenge to reduce the ratio of overhead bits versus data bits in the encoded codeword. Or the researchers might increase the length of the data word that can be encoded using the same algorithm with the same or better error detection and correction ability.

REFERENCES

[1] Satyabrata Sarangi and Swapna Banerjee, “Efficient Hardware Implementation of Encoder and Decoder for Golay Code”, IEEE Transaction on very large scale Integration (VLSI) system, Vol.23 Issue No.9, pg.1965-1968, September 2015.

[2] Marcel J.E.Golay, “Notes on Digital Coding”, Reprinted from proc. IRE, Vol.37, pg-657 June 1949.

[3] Jon Hamkins, “The Golay Code Outperforms the Extended Golay Code”, IEEE Transactions on Information Theory, February 19, 2016.

[4] Mario de Boer and Ruud Pellikaan, “The Golay codes” Springer, pg.338-347, September 1995.

[5] Dr. Ravi Shankar Mishra, Prof PuranGour and Mohd. Abdullah, “Design and Implementation of 4 bits Galois Encoder and Decoder in FPGA”, International Journal of Engineering Science and Technology (IJEST), Vol.3 No.7, pg.5724-5732, July 2011.

[6] DongfuXie, "Simplified algorithm and hardware implementation for the (24,12,8) Extended Golay soft Decoder up to 4 Errors", The International Arab Journal of Information Technology, Vol.11 No.2, pg.111-115, March 2014.

[7] Xiao-Hong Peng and Paddy G. Farrell, "On Construction of the (24, 12, 8) Golay Codes", December 2005.

[8] Matthew G. Parker, Kenneth G. Paterson and ChinthaTellambura, "Golay Complementary Sequences", January 2004.

[9] Yan-Haw Chen, Chih-Hua Chine, Chine-Hsiang Huang, Trieu-Kien Truong and Ming-Haw Jing, "Efficient Decoding of schematic (24,12,7) and (41,21,9) Quadric Residue codes", Journal of Information science And Engineering Vol.26, pg.1831-1843, December 2010.

[10] Li Ping and Kwan L. Yeung, "Symbol-by-Symbol APP Decoding of the Golay Code and Iterative Decoding of Concatenated Golay Codes", IEEE Transaction on Information theory, Vol.45, No.7, pg.2558-2562, November 1999.

[11] Yihua Chen, Juehsuan Hsiao, PangFu Liu and Kunfeng Lin, "Simulation and Implementation of BPSK BPTC of MSK Golay code in DSP chip", Communications in Information Science and Management Engineering, Vol.1 No.4, pp.46-54, Nov.2011

[12] Eyas El-Qawasmeh, Maytham Safar and TalalKanan, "Investigation of Golay code (24, 12, 8) Structure in improving search techniques",The International Arab Journal of Information Technology, Vol.8, No.3, pg.265-271, July 2011.

[13] Ali Pezeshki, A. Robert Calderbank, William Moran and Stephen D. Howard, "Doppler Resilient Golay Complementary Waveforms", IEEE Transaction on Information Theory, Vol. 54, No. 9, September 2008.

[14] Faisal Alsaby, Kholood Alnoo waiser and Simon Berkovich, "Golay code Transformation for ensemble clustering in application of medical Diagnostics", International Journal of Advanced Computer Science and Applications (IJACSA), Vol.6 No.1, pg.49-53, 2015.

[15] V.Bhushan Kumar and K.Yoga Prasad, "Reduction of PAPR and BER by Using Golay Sequences for OFDM System", International Journal of Emerging Engineering Research and Technology, Vol. 2, Issue 7, PP 191-198, October 2014.

[16] A.Iniya Mary and , N.A. Pappathi, "Simplified Algorithm and Hardware Implementation for A VLSI Implementation of Data transmission Error Detection Based Encoder And Decoder" International Conference on Current Research in Engineering Science and Technology, Vol. 11, No. 2, PP-11-13 March 2014.

[17] John H. Conway and N. J. A. Slpane, "Soft Decoding Techniques for Codes and Lattices, Including the Golay Code and the Leech Lattice, Including the Golay Codeand the Leech Lattice", IEEE Transaction on Information Theory, PP-41-51Vol.32, NO. 1, January 1986.

[18] Wen-Ku Su, Pei-Yu Shih, Tsung-Ching Lin and Trieu-Kien Truong, "Soft-decoding of the (23, 12, 7) Binary Golay" International Multi Conference of Engineers and Computer Scientists Vol. 2, PP- 19-21 March, 2008.

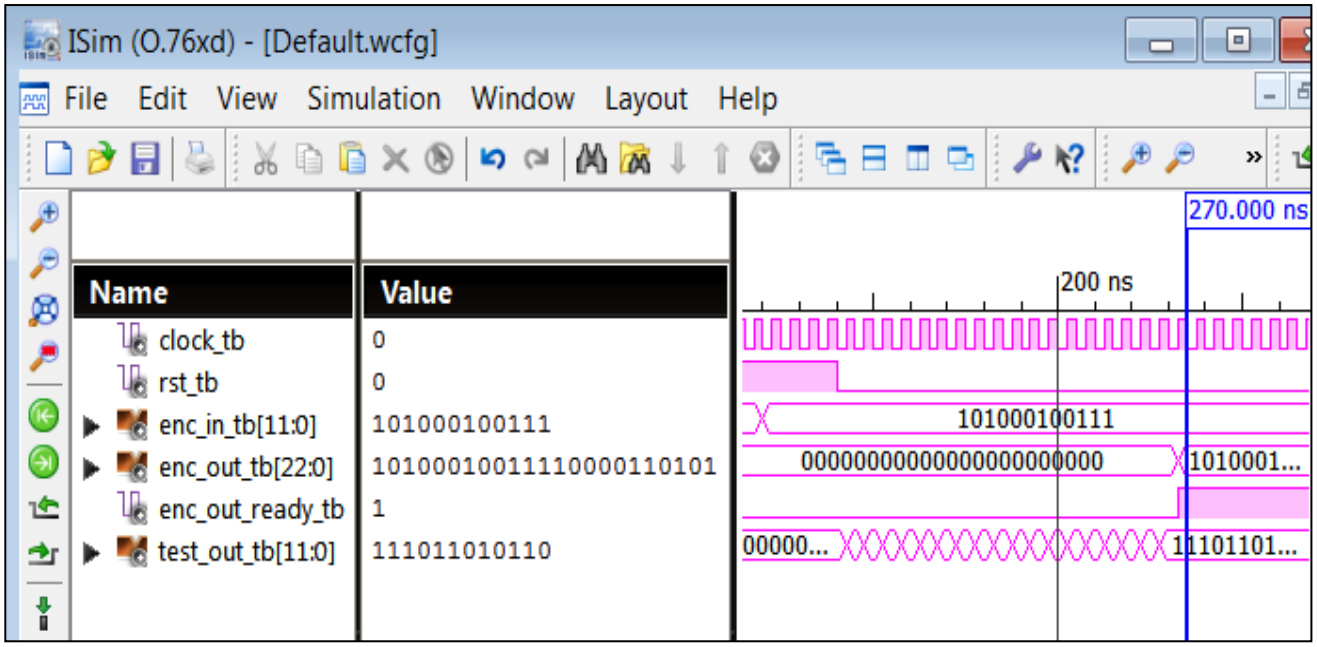

Fig. 7 Encoder Simulation Waveform for Proposed Golay Code (24, 12, 8) Encoder

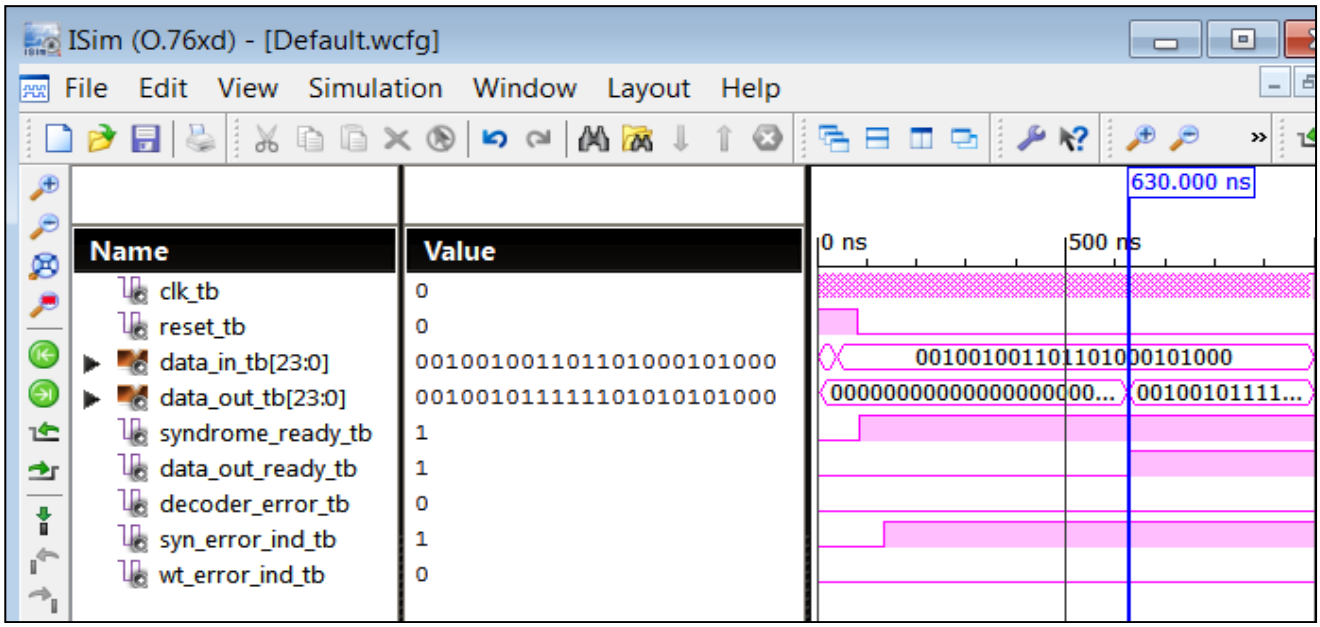

Fig. 8 Encoder Simulation Waveform for Proposed Golay Code (24, 12, 8) Decoder