International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 5 30 - 35

FPGA based Design and Simulation of Extended Golay Codec with Hardware Optimization for high speed Applications

Ujjvala Rangare
M.Tech Scholar Department Of Electronics & Communication
NRI Institute Of Information Science & Technology,
Bhopal, India
e-mail: ujjvalarangare16@gmail.com

Rajeev Thakur
Associate Prof. Department Of Electronics & Communication,
NRI Institute Of Information Science & Technology,
Bhopal, India
e-mail: rajeevthakur82@gmail.com

***Abstract*—**In wireless communication systems the ability of the receiver to detect and correct the error from the received information is become one of the most important issue, so as to provide the processor the correct information data. To achieve this there are numbers of such methods are available to implement the hardware and software. But, length of the communication link plays an important role because the distance of the transmitter and the receiver depends on the length as length increases the distance between the transmitter and the receiver, and multiple bits of the transmitted information may change due to the effect of noise on the transmitted signal. This can cause extreme loss in many cases. This paper presents a brief of Field Programmable Gate Array (FPGA) based design and simulation of Golay Code (G23) and Extended Golay Code (G24) Encoding scheme. This paper using the Golay Encoder to work on the optimization of the time delay of the operational circuit to encode a data packet.

Keywords- *FPGA, Golay Code, Extended Golay Code, Operational Delay.*

*****

I. INTRODUCTION *(HEADING 1)*

Communication is become an important part of humans life. Use of phones, satellites, computers and other devices now become a basic need to solve our day to day problems like send messages through a channel to a receiver. Unfortunately, errors in the messages that are being sending are caused by the noise. Particularly when sending messages is a complicated or expensive task, for example in satellite communication, it is important to find ways to restrained the amount 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 simple and free of error we use error-correcting codes. Addition of the redundancy bits to the messages is one of the best idea through which we enable to find out or correct the errors that may have occurred. 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 proposed a specific 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.There are three steps to transfer the information, a channel transmits, and a receiver receives. There is an alternative that at the time of transmission the information is changed to noise so to avoid this condition we use error correction codes. Figure 1 show that a message is encoded into a codeword, it is sent to the receiver through a channel, in this channel the possibility exists that errors occur, and the receiver tries to obtain the original message by decoding the word.

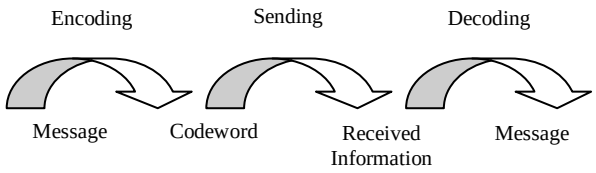

Figure 1. Process of Error Correction Code

One of the best error correcting code that is proposed in this paper is Golay code which is an error correcting code which is used to specifies that what we have received and what is send. A detailed description of the extended Golay is described by some of the most important properties of such codes that is given by :

- 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 0 is received when 1 was sent, or 1 is received when 0 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 belongs 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. LITERATURE REVIEW

In reference [1] the proposed paper presented error correcting phenomena using Golay code encoder. The algorithm of encoding data for error detection and correction was in the beginning proposed by Marcel J. E. Golay in 1949 [2]. A concise introduction and explanation of Golay coding scheme is presented in [3]. An implementation with complete verification of multiplication is simulated and an FPGA based 4-bit Golay Encoder and Decoder design in [4] using Xilinx ISE and ModelSim Tools. A soft algorithm based decoding orientation to hardware functioning of (24, 12, 8) Golay code with functioning of the algorithm on FPGA is presented by Reference [5]. In [6] it has been shown that (24, 12, 8) Golay code can be constructed with the help of the sum of two direct array codes that involve four component codes from which two are simple linear block codes. Assembly of Golay Code matching Sequences is presented in [7] for application of Golay Coding in the fields of surface acoustics, physics, combinatorial (orthogonal designs and Hadamard matrices), and tele-communication. Reference [8] represents a one on one mapping between the syndrome "S1" and correctable error patterns scheme based adapted algorithm for decoding Binary Golay. In this proposed work the error location is determine by using look-up tables without the multiplication operation over a finite field. This algorithm has been established by the scholars on a C-language based software simulation platform. The work presented in [9] forced 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 BPSK modulation with Golay code and MSK modulation with Golay code is presented in [10], which concludes that MSK Golay code is comparatively more robust. In reference [11] a technique based upon reversing the conventional scheme of Golay code (24, 12, 8) that maps 24-bit vectors into 12-bit message words is focused to improve the search operation when multi-attribute objects are partially distorted. The work in [12] presents Golay code transformation for Ensemble Clustering in application to Medical Diagnostics. This clustering methodology is unique to outperform all other conventional techniques because of its linear time complexity. The work in [13] presents generation of Doppler Resilient waveforms using Golay Complementary sequence which have ideal ambiguity along the zero Doppler axis but are sensitive to non-zero Doppler shifts..

III. GOLAY CODE ENCODER ALGORITHM

A binary Golay code is represented by (23, 12, 7), which shows the message is of 12-bits while the length of codeword is 23 bits and the minimum distance between two binary Golay codes is 7 and It is necessary to build binary codes in a Galois Field (GF). Binary field is denoted by GF(2), which supports different binary arithmetic operations. An Extended Golay codeword of a 12-bit data bits can be generated with the help of generator matrix by performing following matrix operation:

$$W = [\, D * G \,]$$

Since, D = 12-bit data is a row matrix of 12 elements, G = generator matrix of size 12x12, so the size of word matrix will be 1x24, i.e., "generated code word will be a 24-bit row matrix". The golay code can be generated 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 as follows. B*i* represents i^{th} row of the matrix B.

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

The generator matrix is shown as follows:

$$G = [I , B] = \begin{bmatrix} 100000000000 & 110111000101 \\ 010000000000 & 101110001011 \\ 001000000000 & 011100010111 \\ 000100000000 & 111000010111 \\ 000010000000 & 110001011011 \\ 000001000000 & 100010110111 \\ 000000100000 & 000101101111 \\ 000000010000 & 001011011101 \\ 000000001000 & 010110111001 \\ 000000000100 & 101101110001 \\ 000000000010 & 011011100011 \\ 000000000001 & 111111111110 \end{bmatrix}$$

Te Golay codeword wich is generated using a generator matrix has following properties:

1. It has length n=24, dimension k=12 and 2^{12}=4096 codewords.
2. A Parity Check Matrix for codework is 24x12 matrix
3. Another Parity Check Matrix is 24x12 matrix
4. Another Generator Matrix is 12x24 matrix [B , I]
5. Codeword is self –dual, i.e., $W = W^{\perp}$
6. The distance of codeword is 8.
7. Extended Golay Code is a 3-bit error correction perfect code

A simple flow of generation of extended golay codeword is represented using block diagram in figure 2.

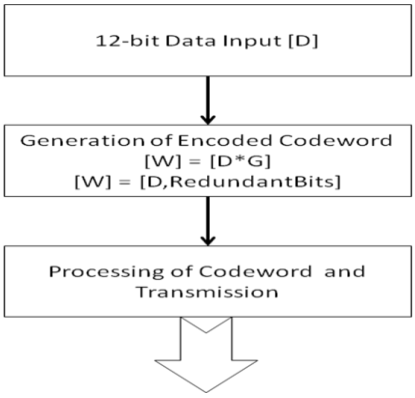

Figure 2 Generation of Extended Golay Codeword

The generated extended Golay codeword is verified by measuring the weight of the G(24) codeword. In a verified G(24) codeword the weight is multiple of 4 and greater than equal to 8. A simple block diagram that shows the flow of the decoder for golay encoded codeword is shown in figure 3.

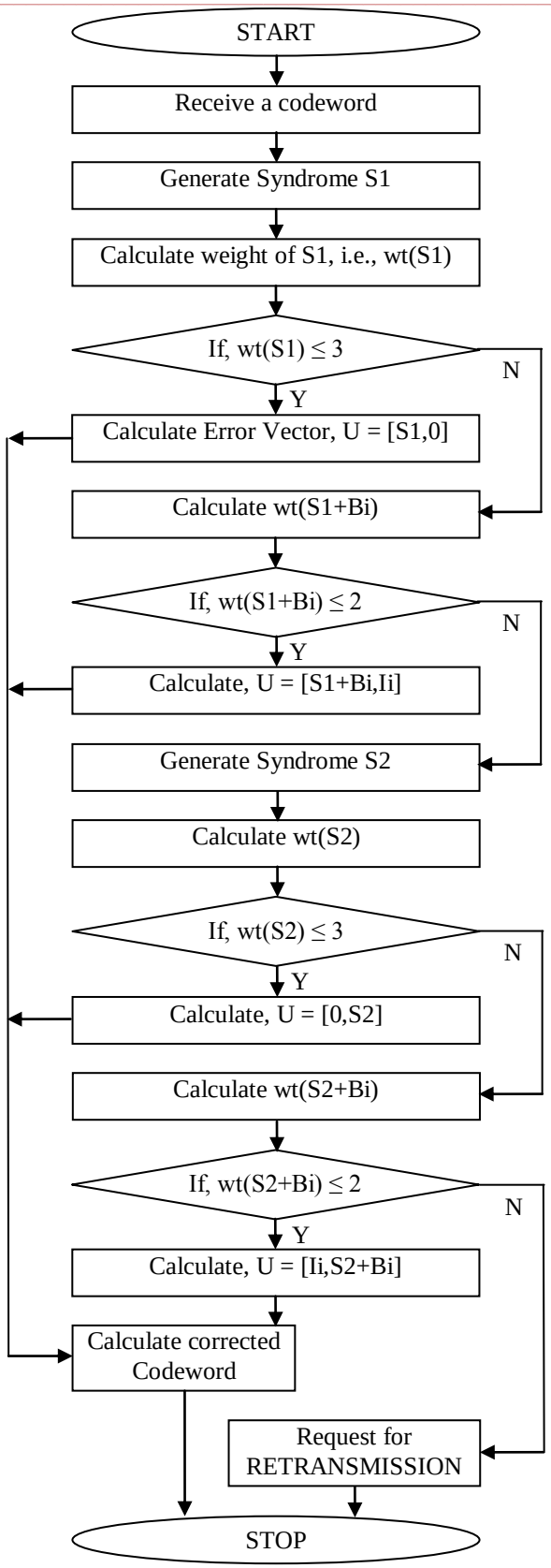

Figure 3 Decoding Steps of Extended Golay Codeword

Now to identify the error pattern we have to calculate the syndrome using the received codeword at the decoder end. Figure 4 shows the equation involved in the generation of syndrome bits.

S[11] = w[23] xor w[11] xor w[10] xor w[8] xor w[7] xor w[6] xor w[2] xor w[0]

S[10] = w[22] xor w[11] xor w[9] xor w[8] xor w[7] xor w[3] xor w[1] xor w[0]

S[9] = w[21] xor w[10] xor w[9] xor w[8] xor w[4] xor w[2] xor w[1] xor w[0]

S[8] = w[20] xor w[11] xor w[10] xor w[9] xor w[5] xor w[3] xor w[2] xor w[0]

S[7] = w[19] xor w[11] xor w[10] xor w[6] xor w[4] xor w[3] xor w[1] xor w[0]

S[6] = w[18] xor w[11] xor w[7] xor w[5] xor w[4] xor w[2] xor w[1] xor w[0]

S[5] = w[17] xor w[8] xor w[6] xor w[5] xor w[3] xor w[2] xor w[1] xor w[0]

S[4] = w[16] xor w[9] xor w[7] xor w[6] xor w[4] xor w[3] xor w[2] xor w[0]

S[3] = w[15] xor w[10] xor w[8] xor w[7] xor w[5] xor w[4] xor w[3] xor w[0]

S[2] = w[14] xor w[11] xor w[9] xor w[8] xor w[6] xor w[5] xor w[4] xor w[0]

S[1] = w[13] xor w[10] xor w[9] xor w[7] xor w[6] xor w[5] xor w[1] xor w[0]

S[0] = w[12] xor w[11] xor w[10] xor w[9] xor w[8] xor w[7] xor w[6] xor w[5] xor w[4] xor w[3] xor w[2] xor w[1]

Figure 4 Generation of bits of Syndrome 'S'

To calculate (S + B*i*), for 1≤*i*≤12, the implemented hardware use bit inversion of 'S' as per the logic shown in figure 5.

$S + b_1$ = { ~S[11], ~S[10], S[9], ~S[8], ~S[7], ~S[6], S[5], S[4], S[3], ~S[2], S[1], ~S[0] }

$S + b_2$ = { ~S[11], S[10], ~S[9], ~S[8], ~S[7], S[6], S[5], S[4], ~S[3], S[2], ~S[1], ~S[0] }

$S + b_3$ = { S[11], ~S[10], ~S[9], ~S[8], S[7], S[6], S[5], ~S[4], S[3], ~S[2], ~S[1], ~S[0] }

$S + b_4$ = { ~S[11], ~S[10], ~S[9], S[8], S[7], S[6], ~S[5], S[4], ~S[3], ~S[2], S[1], ~S[0] }

$S + b_5$ = { ~S[11], ~S[10], S[9], S[8], S[7], ~S[6], S[5], ~S[4], ~S[3], S[2], ~S[1], ~S[0] }

$S + b_6$ = { ~S[11], S[10], S[9], S[8], ~S[7], S[6], ~S[5], ~S[4], S[3], ~S[2], ~S[1], ~S[0] }

$S + b_7$ = { S[11], S[10], S[9], ~S[8], S[7], ~S[6], ~S[5], S[4], ~S[3], ~S[2], ~S[1], ~S[0] }

$S + b_8$ = { S[11], S[10], ~S[9], S[8], ~S[7], ~S[6], S[5], ~S[4], ~S[3], ~S[2], S[1], ~S[0] }

$S + b_9$ = { S[11], ~S[10], S[9], ~S[8], ~S[7], S[6], ~S[5], ~S[4], ~S[3], S[2], S[1], ~S[0] }

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

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

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

Figure 5 Calculation of (S + B*i*)

To calculate the weight of a 12-bit (S + B*i*) and (SB + B*i*), for 1≤*i*≤12 a simplified adder based architecture is

implemented in this work. The architecture of adder based weight calculation unit is shown in figure 6.

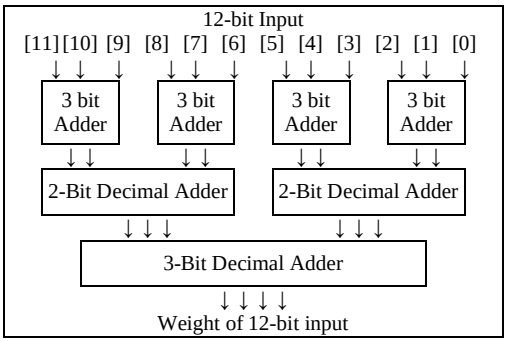

Figure 6 Architecture of Weight Calculation Unit

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

1) For the received codeword 'W' and matrix 'H', where H = [I / B] Compute the Syndrome 'S'.
2) Error vector, E = [S, 0], If weight of 'S' is less than or equal to 3, i.e., wt(S) ≤ 3.
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) The second syndrome SB can be computed
5) If wt(SB) ≤ 3, then E = [0, SB]
6) If wt(SB+B*i*) ≤ 2, then E = [I*i*, S+B*i*]
7) If E is still not determined then received data is required to be retransmitted.

IV. Simulation and Synthesis Results Styling

Table –I and Table –II respectively represents the FPGA based hardware utilization summary of the proposed Encoder and Decoder designs. Table-III represents a comparative analysis of the hardware resource utilization based results of the proposed work with some existing works.

TABLE I. Hardware Utilization Summary of Encoder

Vertex-IV XC4VLX160-12FF1148	**Total**	**Encoder**	
		Used	**%**
Slices	67584	14	0
LUTs 4-Inputs	135168	25	0
Bonded IOBs	768	38	6

TABLE II. Hardware Utilization Summary of Decoder

Vertex-IV XC4VLX160-12FF1148	**Total**	**Decoder**	
		Used	**%**
Slices	67584	291	0
LUTs 4-Inputs	135168	544	0
Bonded IOBs	768	52	6

Xilinx is the software which is used in the present work and simulation using The RTL Schematic diagrams of Encoder and Decoder designs are shown in figure 7 and figure 8 respectively.

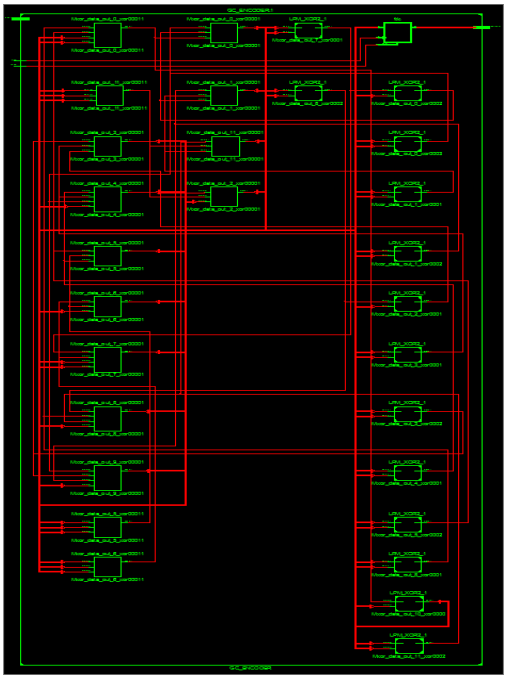

Figure 7 RTL Schematic Diagram of Proposed Golay Code (24, 12, 8) Encoder

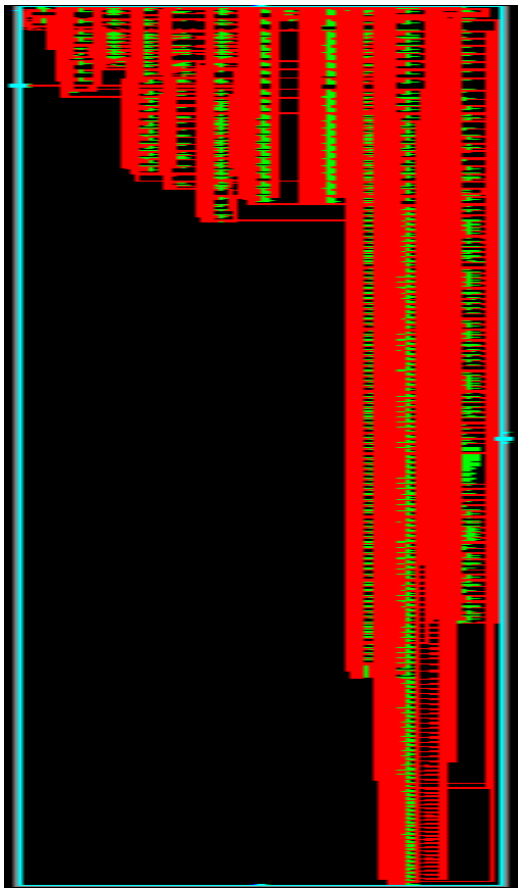

Figure 8 RTL Schematic of Proposed Golay Code (24, 12, 8) Decoder

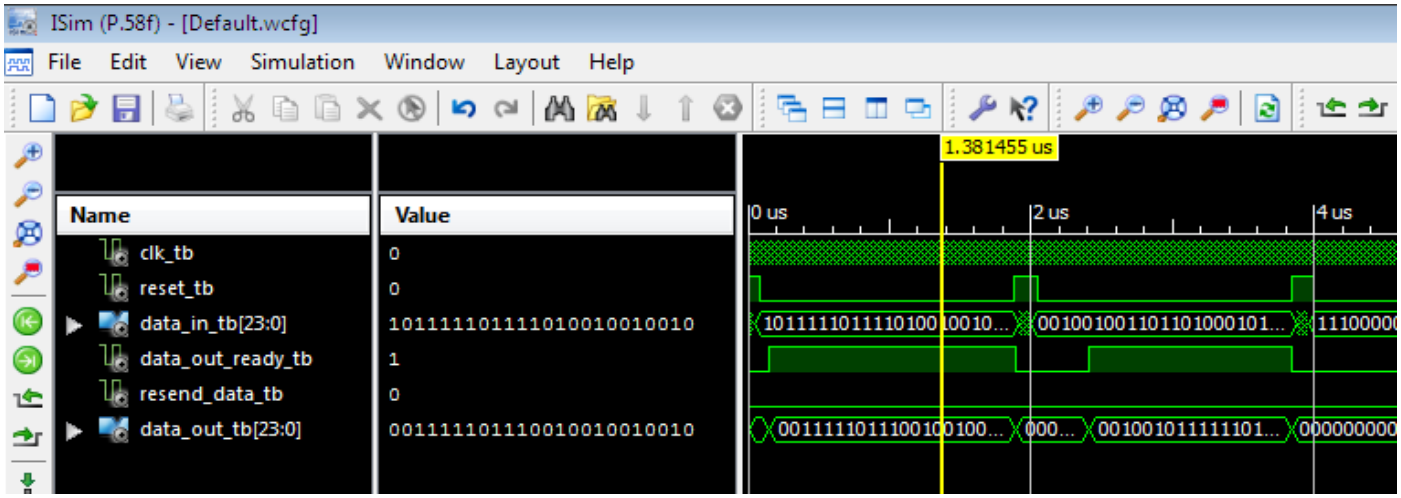

Figure 9 Encoder Simulation Waveform for Proposed Golay Code (24, 12, 8) Decoder

The Decoder simulation waveform is shown in figure 9. A 12-bit data is used to encode using the proposed encoder. The input data bits are followed by logic-'0' inputs. A comparative chart that represents the reduction in the hardware resource utilization of the proposed encoder and decoder designs with respect to the existing design is shown in figure 10 and figure 11 respectively. The comparative result shows that the proposed implementation requires less number of hardware resources from the FPGA device.

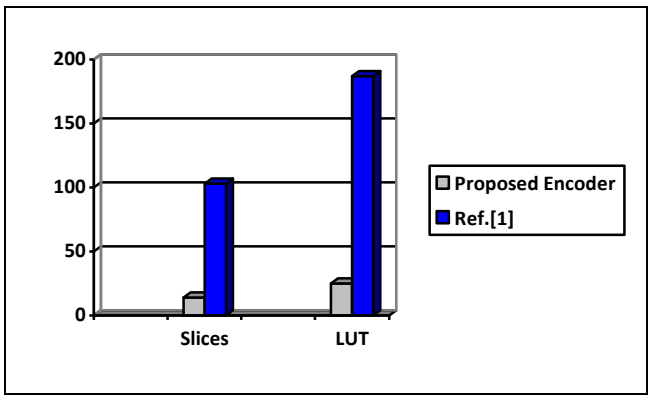

Figure 10 Hardware utilization compression chart of Encoder

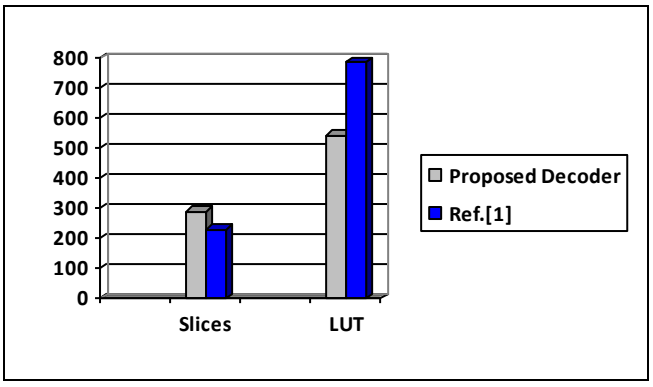

Figure 11 Hardware utilization compression chart of Decoder

V. CONCLUSION

In the proposed work hardware optimized architecture of extended binary Golay encoder and decoder are designed and simulated. The consequences obtained from the design combination for encoder and decoder supersedes the reference schemes in term of the operational frequency. This makes the proposed design a good quality option to be used in the high rate application based configurable circuits. In future there is a enormous scope to further optimize the performance of the proposed algorithm. In future the scholars may assume 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 capacity.

ACKNOWLEDGMENT

The authors thank Mr. Piyush Jain (Director, Innovative Technology Design and Training Center, Bhopal India) for sharing his ideas in-line with the presented work.

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] Mario de Boer and Ruud Pellikaan, "The Golay codes" Springer, pg.338-347, September 1995.

[4] Dr. Ravi Shankar Mishra, Prof Puran Gour 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.

[5] Dongfu Xie, "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.

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

[7] Matthew G. Parker, Kenneth G. Paterson and Chintha Tellambura, "Golay Complementary Sequences", January 2004.

[8] 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.

[9] 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.

[10] 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

[11] Eyas El-Qawasmeh, Maytham Safar and Talal Kanan, "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.

[12] Faisal Alsaby, Kholood Alnoowaiser 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.

[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 , pg.4254-4266, September 2008.