Low Power SEC-DED Hamming Code Using Reversible Logic

M.Sai Sarath, T.Subbarao and Sarada Musala

Abstract—The errorless data transmission in optimal communication depends on flaw identification and renewal techniques. In one bit error identification and error modification codes the hamming code is mostly used, because of its efficiency. To increase performance, system capability we need the low power design. To reduce the rate of information loss and power dissipation the reversible logic is well suited. When a reversible logic gates used in the design of hamming code, in place of irreversible logic gates gives low power dissipation, which identify and modify the error if any. In this document we construct the hamming code using the FinFET technique, because of its capability of operating at lower supply voltages and it is suppress the short channel effects in sub-micron region, through this documentation we represent the propagation delay values. The transient responses are taken by using cadence virtuoso. The power consumption of overall design is 0.48□W.

***Index Terms*—Hamming code, FinFET, Reversible logic gates, Communication, parity checking, parity bits.**

I. Introduction

THE data transfer is the base of all modern-day applications. Due to noise and environmental interference in transmission between the transmitter and receiver gives errors in the original data. The error is defined as the bit alternation between the original data and receiving data at the receiving side. The flaw identification and renewal circuits are used in communication systems to eliminate the error in all digital circuits. To resistant external disturbances in data transmission, we add the redundant bits in original data for error correction.

In communication system different flaw identification and renewal codes are presented like Check sum, longitudinal redundancy check and hamming code. Hamming code is the widely using flaw identification and renewal code because it is easy to design. In this code we attach some extra bits to the message and this process is kept the message is as minimum as possible. This code is most effective and efficient in the category of one error renewal code. The usage of hamming code was where the flow rate is minimized [1].

When hamming code system is designed using traditional irreversible gates it consumes more power. In any circuit designing the primary consideration is power consumption. So in this paper to reduce the overall power consumption, we used reversible gates in place of irreversible gates.

M.Sai Sarath, T.Subbarao and Sarada Musala are with the Vignan's Foundation for Science, Technology & Science, A.P, India. (email: saisarath.munnangi@gmail.com, tsubbarao049@gmail.com, sarada.marasu@gmail.com)

Landauer defines that when data transmitting some weightage of energy is consumed for each bit of data lost. Later Charles Bennett [2,3] reported that to eliminate or minimize the heat dematerialization by using the reversible logic in the construction of all the blocks. In reversible logic circuits there is corresponding between each input to output and the outputs can be reproduced from inputs and there is associated between the each input and its output. The circuit can operate in both backward and forward. It gives to go back at any point in the computation history. It also increases the total response of the circuit by reducing power dissipation by approving high densities and high speeds.

In this paper, the construction structure of Hamming code encrypting and decrypting circuit's clarification is divided into three sections. Reversible logic gates are given under Section II. The hamming code construction is given in Section III. The imitation of process and power utilization of every block is mention under section IV. The mathematical analysis of Hamming code encrypting and decrypting of each block is explained under section V and the cessation of the paper in VI.

II. Reversible Logic Gates

The reversible logic is said to be when the input vector is retrieved from output vector, one to one corresponding between input and output assignments then it is called reversible logic. In reversible logic there are two types of computations are available. One is a forward computation and second is backward computation. In Hamming code encoding and decoding we used three reversible logic gates.

1. Fredkin gate (F)

   Fredkin gate shown in Fig. 1 is one of the reversible nature of logic gate and it contains three outputs and three inputs. In forward computation the outputs are followed

M = I1
If I1=0, then N = I2 and O = I3
Else N = I3 and O = I2

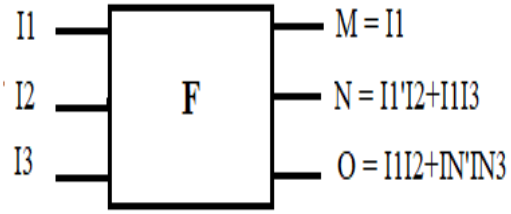

Fig. 1. Fredkin gate

978-1-7281-4988-2/20/$31.00 ©2020 IEEE

In backward computation the outputs are followed

I1 = M

If I1 = 0 then I2 = N and I2=O

Else I2 = O and I3 = N

2. Feynman gate (FG)

FG shown in Fig. 2 is another type of reversible logic gate and it's contain two input and two outputs. The forward computation the outputs are follows

M = I1

If I1 = 0 then N = I2 else N = I2'

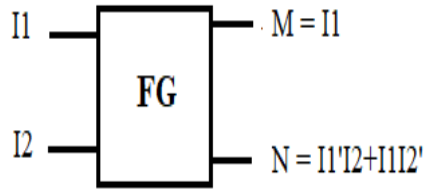

Fig. 2. Feynman gate

In backward computation the outputs are followed

I1 = M

If M = 0 then I2 = N else I2 = N'

3. Double Feynman gate (F2G)

F2G shown in Fig. 3 is a similar type of reversible logic gate when it is compared to FG. The only difference is it contains one more input and output. The forward computation responses are followed

M= I1, N = I2 and O = I1 XOR I2 XOR I3

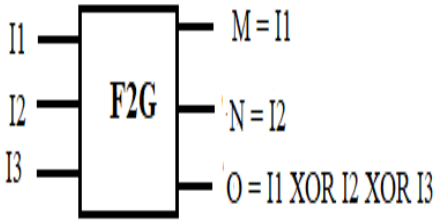

Fig. 3. Double Feynman gate

In backward computation the outputs are followed

I1 = M, I2 = N and I3= MXOR N XOR O

III. Constriction Of Hamming Code

This delivers reversible logic power minimized of single bit flaw identification and renewal hamming code encrypting and decrypting circuits. The hamming code circuit design divided into three different logic blocks, one is to encrypt the original message by including the parity bits. The next circuit produces check bits and the generated check bits helps in error identification. The last circuit uses to detect any error introduced in the transmission process, decrypts the check bits, detects non efficient bit and corrects the bit by alters the value. By above process, we have the restriction free transmission. Every one of the before mention the blocks explanation in upcoming sections. The Section III. A gives details of the circuit of Hamming Code Encrypting Reversible logic cell, section III.B gives details of the circuit of Hamming CG Reversible Logic Cell and section III.C elucidates the Error identification and changing the required circuit Encoder circuit (EC) is constructed using two FG and three F2G gates.

A. Encoder Logic Circuit

The parity bits are mention in input message the positions that are places in powers of 2. The parity bits are determined using the algorithm mention in [4].

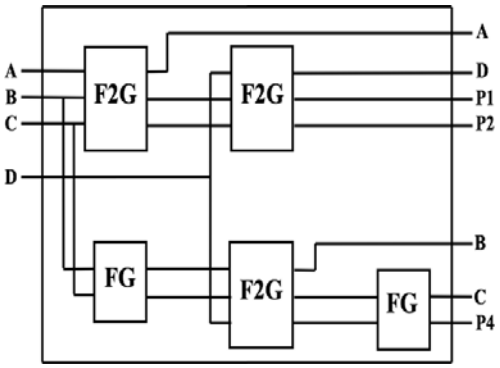

Fig. 4. Block diagram of Encoder circuit

The outputs of FG and F2g are XOR operation of the input vectors. The output of the encoder contains seven bit message length with one nibble message bits and 3 parity bits and the outcome of this circuit as taken to CG cell input. The block diagram of EC is shown in Fig. 4.

B. Check Bit Generator Circuit

To verify if the data have arrived without any loss of the original message to the receiver, check bits are determined on the destination side. Check bit generator (CG) calculates check bits according to the algorithm mention in [4]. If any error presents in the received data the flaw bit location is identified by using the check bits. The check bits$C_4C_2C_1$gives location of error in binary data

To design CG we need five FG gates and two F2G gates. For input of CG we had taken output of the EC as input to CG [5,6]. The CG contains 7 input data word and output check bits. Fig. 5 shows the block diagram of CG Reversible Logic cell.

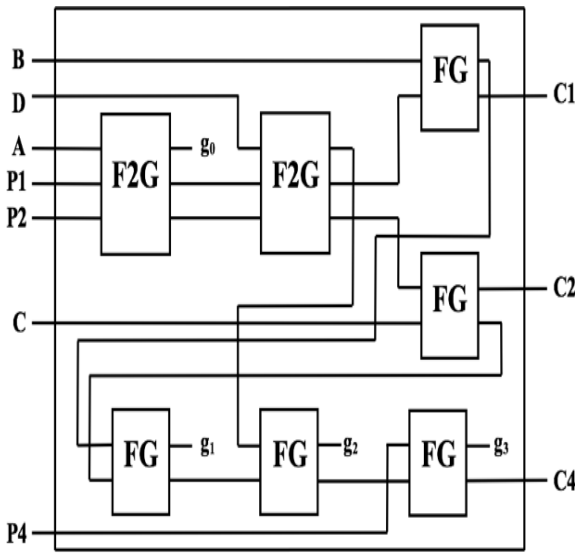

Fig. 5. Block diagram of check bit generator

C. *Error Detection And Correction Circuit*

From above block the difference between the original data and received data has been observed. To modify the error, we must alters value in the position of the error bit with help of $C_4C_2C_1$. The error correction is done by logic cell EDC. The EDC cell contains a decoder which is detected an error by taking $C_4C_2C_1$as inputs and identify the error bit position [7-10]. The decoder has been designed using Fredkin gates.

In the EDC logic cell we use the Feynman gates, which are acting as the XOR gates. When the one input in XOR gate is logic '1' then it is acting as an inverter. So after finding an error bit position the Feynman gate invert faulty bit and gives the correct data [11-13].

The EDC cell consists a decoder and seven FG gates. The output of the CG is given, input to the EDC cell. Fig. 6 gives the details of circuit EDC [14, 15]. The Transient responses are scrutinized in section 4 to examine the hamming code design [16,17].

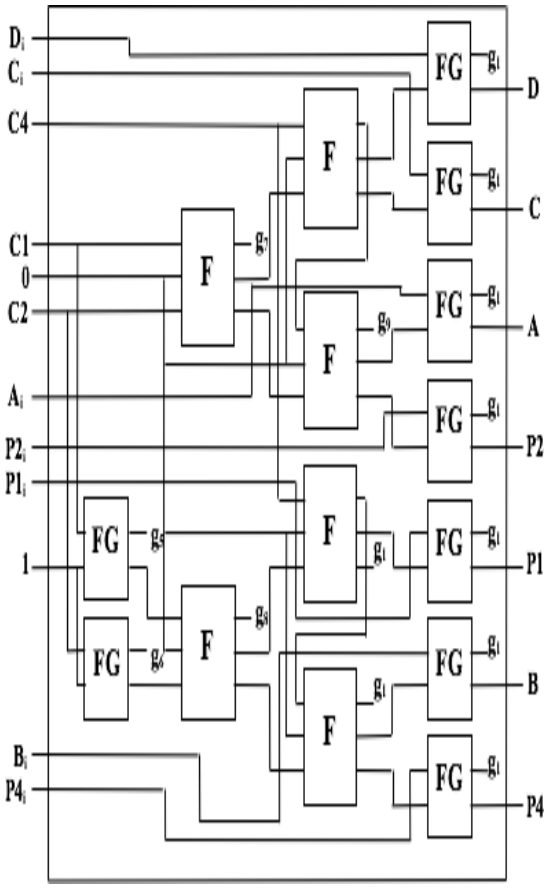

Fig. 6. Block diagram of EDC

IV. Simulation Results

In this section we illustrate the transient responses of every reversible logic circuit used in this hamming code design are represented in this section as Fig. 7, Fig. 8 and Fig. 9. The transient responses are taken using cadence virtuoso.

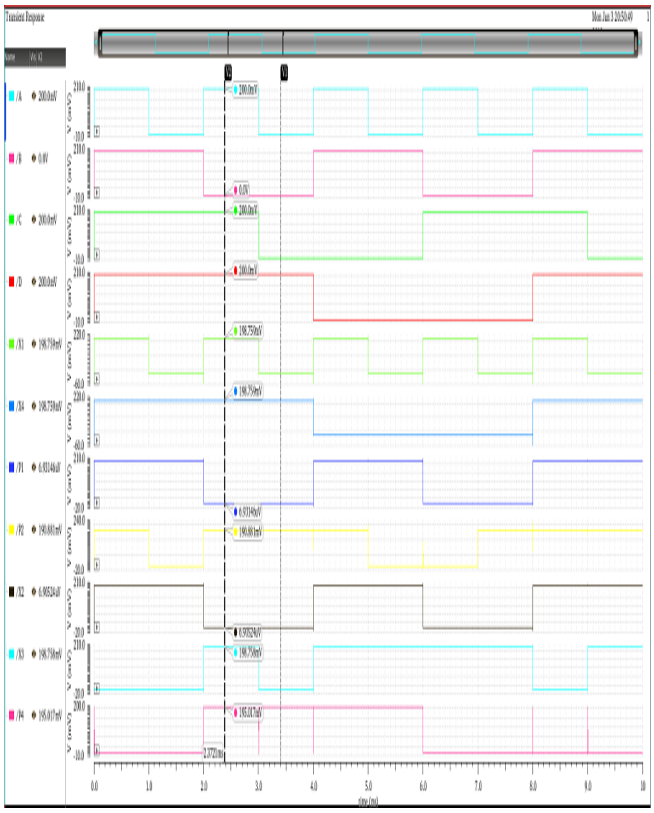

Fig. 7. Simulation result of EC cell

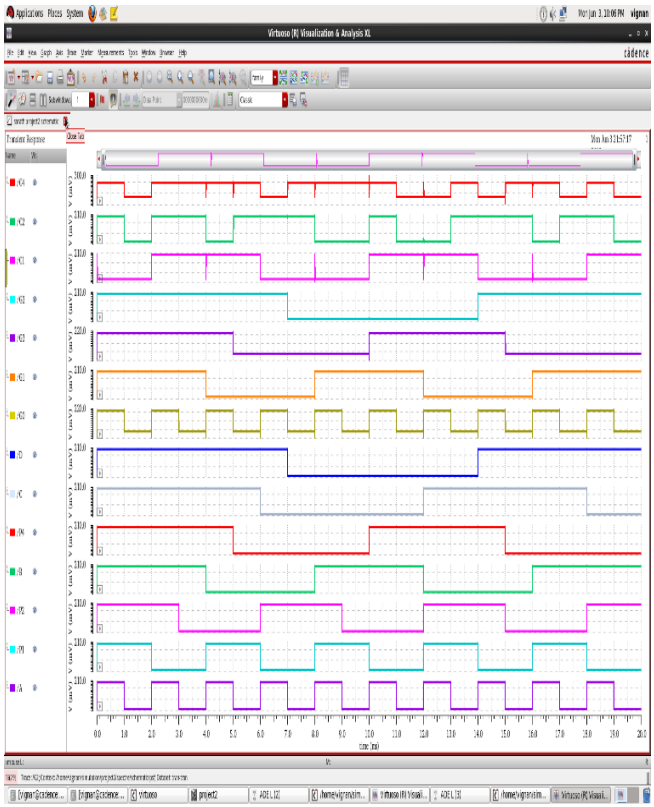

Fig. 8. Simulation result of CG cell

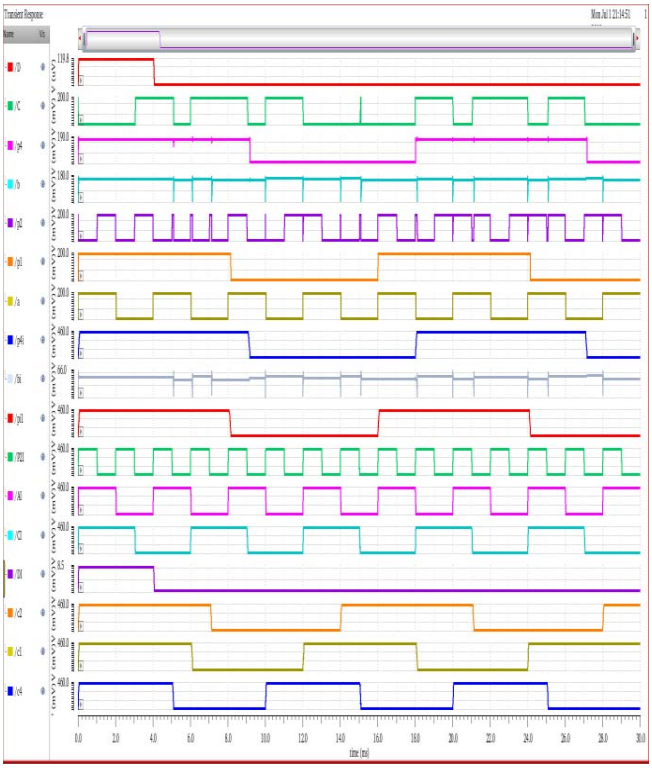

Fig. 9. Simulation result of EDC cell

V. ANALYSIS

A. *Calculation for No.of Gates (NOG)*

To know the total NO.OF gates used in hamming code by adding the NO.OF gates used in every individual block. In hamming code encrypting cell we used 2 Feynman gates and 3 double Feynman gates, for the construction of Check bit generator we used 2 double Feynman gates and 5 Feynman gates. The 6 Fredkin gates and 9 Feynman gates are used in designing of the EDC logic cell. With the help of below equation, we can find the total number of gates used in the overall design. Table I representes the power and delay values.

$$\text{No.OF GATES}_{HMC} = \text{No.OF GATES}_{EC} + \text{No.OF GATES}_{CG} + \text{No.OF GATES}_{EDC}$$

= [3 (double Feynman gates) + 2 (Feynman gates)] + [(double Feynman gates) + 5 (Feynman gates)] + [6 (Fredkin gates) + 9 (Feynman gates)]

=27 (1)

B. *The complete power dissipation* of reversible hamming code design is calculated using the below expression

$$\text{Powerhouse} = \text{Power}_{EC} + \text{Power}_{CG} + \text{Power}_{EDC} \quad (2)$$

C. *Calculating propagation delay*

Thorough below equation, we can determines the overall propagation delay of hamming code is

$$D_{HMC} = D_{EC} + D_{CG} + D_{EDC} \quad (3)$$

TABLE I
POWER AND DELAY VALUES

	EC	CG	EDC	HMC
Power	0.0492µW	0.1575µW	0.278µW	0.484µW
Delay	3.375µs	3.375µs	2.84µs	8.375 µs

VI. CONCLUSION

This article gives a new method to optimal power usage in reversible logic gates used in hamming code circuit instead of irreversible logic gates with FinFET technology. The reverse logic Hamming Code encrypting and decrypting circuit have 27 numbers of gates and overall power dissipation is 0.484µW. The total propagation delay is 8.375µS. Simulation of the each and every circuit are conducted in Cadence Virtuoso 18nm FinFET technology.

REFERENCES

[1] V. S PNayak, G Prasad, K. DChowdary and K. M Chari, "Design of Compact and Low Power Reversible Comparator", (ICCICCT-2015).

[2] R. Landauer, "Irreversibility and heat generation in the computational process", IBM J of Research and Development, 5(2) 1961, Pp:183–191.

[3] C.H. Bennett, "Logical reversibility of computation", IBM Journal of Research and Development, volume: 17, Issue: 6, 1973, Pages: 525–532.

[4] D R Choudhury, K Podder, "Design of Hamming Code Encoding and Decoding Circuit Using Transmission Gate Logic", IRJET, 2(7) 2015.

[5] V. S Prasad, C Madhulika, U pravali "Design of low power hamming code encoding, decoding and correcting circuits using reversible logic ", 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017.

[6] Landauer R "Irreversibility and heat generation in the computational process", IBMJ. Res. Dev., 1961, 5, pp. 183- 191.

[7] Perkowski, M., et al. "A general decomposition for reversible logic". Proc. RM'2001, Starkville, 2001, pp. 119- 138.

[8] S. Vijayalakshmi and V. Nagarajan (FEB 2019), "Design and Implementation of Low Power High-Efficient Transceiver for Body Channel Communications", Springer- Journal of Medical Systems, DOI:https://link.springer.com/article/10.1007%2Fs10916-019-1204-x

[9] Bennet, C.H.: "Logical reversibility of computation", IBM 1. Res. Dev., 1973, 17, pp. 525- 532.

[10] Quantum cost Akanksha Dixit, Vinod Kapse "Arithmetic & Logic Unit (ALU) Design using Reversible Control Unit", (IlELT) Volume I, Issue 6, June 2012 55, ISSN: 2277-3754.

[11] AI-Rabadi, A.N., "Closed - system quantum logic network implementation of the viterbi algorithm", Facta Universitatis- Elec. Energ., 2009, 22, (1), pp. 1-33.

[12] Rangaraju, H.G., Hegde, V., Raja, K.B., Muralidhara, K.N.: "Design of efficient reversible binary comparator" . Int. Conf on Communication Technology and System Design, 2012, vol. 30, pp. 897- 904.

[13] Rangaraju, H.G., Hegde, V., Raja, K.B., Muralidhara, K.N. "Design of low power reversible binary comparator'. Proc. Engineering (ScienceDirect), 2011.

[14] Thapliyal, H., Ranganathan, N ., Ferreira.R. "Design of a comparator tree based on reversible logic", Proc. Tenth IEEE Int. Conf. on Nanotechnology Joint Symp with Nano, August 2010, pp. 1 I 13- 1 I 16.

[15] Vudadha, C., Phaneendra, P.S., Sreehari, V., Ahmed, S.E.,Muthukrishnan, N.M., Srinivas, M.B, "Design of prefix-based optimal reversible comparator". IEEE Computer Society Annual Symp. on VLSI, 2012, pp. 201- 206.

[16] Nagamani,A.N., Jayashree, H.V., Bhagyalakshmi, H.R. "Novel low power comparator design using reversible logic gates", Indian J. Comput. Sci. Eng., 201 I, 2, (4), pp. 566- 574.

[17] Morrison, M., Lewandowski, M., Ranganathan, "Design of a treebased comparator and memory unit based on a novel reversible logic structure". IEEE Computer Society Annual Symp. on VLSI, 2012, pp.331- 336.