Design of Hamming Code Encoding and Decoding Circuit Using Transmission Gate Logic

Debalina Roy Choudhury1, Krishanu Podder2

[1]*Department of Electronics and Communication Engineering, Heritage Institute of Technology, Kolkata, 700107, India*

[2]*Department of Electronics and Communication Engineering, Tripura Institute of Technology, Tripura, 799009, India*

Abstract - ***In this paper, the Hamming code encoder and decoder circuit is implemented using transmission gate logic. The architecture is simulated with different technologies (16nm, 22nm, 32nm, and 45nm) with the help of TANNER EDA Tool for the study of total power dissipation of the circuit. The analysis shows that with the decrease of channel length, there is an decrease of 12.65 % and 2.37 % power dissipation in 16nm compared to 22nm in encoder and decoder circuit. The values of model parameters are used from Predictive Technology Model (PTM). For circuit verification, the hamming code algorithm is implemented in Spartan 3A family XC3S700A FPGA device using VHDL coding technique.***

Key Words: VHDL, FPGA-Xilinx, Hamming, TANNER

1. INTRODUCTION

In digital communication, errors are introduced during the transmission of data from the transmitter to receiver due to noise or environmental interference [1]. These errors can become a serious problem for achieving accuracy and performance of the system [1]. Therefore, the reliability of data transmission is required to be improved. To improve the reliability, it is essential to detect and correct the error. Hence, we have to use some kind of error control coding for error detection and error correction. In this coding, one or more than one extra bit is added to the data bits at the time of transmitting the data. These extra bits are called parity bit that helps to detect the errors. The data bits along with the parity bit form a code word [1].

There has different type of error control coding such as parity checking, check sum error detection, cyclic redundancy check, VRC, LRC & Hamming code. Compare with other error controlling code, hamming code has high efficiency for error detection and as well as error correction and this code is also easy to implement. Because of the simplicity of hamming code, they are widely used in computing memory, data compression & other application of telecommunication [2].

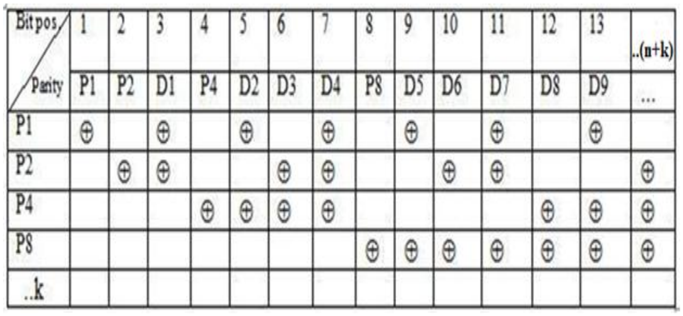
In this paper, we have designed a circuit to implement the hamming code. Here we analyze the hamming code algorithm for 8 bit data word. Thus we need 4 parity bits and make 12 bit code word. This architecture is designed in XILINX ISE 13.44 design suite and implemented in Spartan 3A family XC3S700A FPGA device. For the analysis of VLSI design attributes we have implemented the circuit in TANNER EDA tool and computed the total power for different biasing voltages.

The rest of the paper is organized as follows. In section II, we describe about the hamming algorithm. Section III presents the proposed circuit of hamming encoding and decoding. Section IV deal with the description of the implementation of hamming code in FPGA. In Section V, hamming code implementation in TANNER EDA tools describes and depicts the synthesis result. Section VI is the conclusion.

2. HAMMING CODE ALGORITHM

General algorithm for hamming code- is as follows:

1. k parity bits are added to an n-bit data word, forming a code word of n+k bits [4].
2. The bit positions are numbered in sequence from 1 to n+k.
3. Those positions are numbered with powers of two, reserved for the parity bits and the remaining bits are the data bits.
4. Parity bits are calculated by XOR operation of some combination of data bits. Combination of data bits are shown below following the rule [4].



Bit pos	1	2	3	4	5	6	7	8	9	10	11	12	13	..(n+k)
Parity	P1	P2	D1	P4	D2	D3	D4	P8	D5	D6	D7	D8	D9	...
P1	⊕		⊕		⊕		⊕		⊕		⊕		⊕	
P2		⊕	⊕			⊕	⊕			⊕	⊕			⊕
P4				⊕	⊕	⊕	⊕					⊕	⊕	⊕
P8								⊕	⊕	⊕	⊕	⊕	⊕	⊕
..k														

Fig. 1. Calculation method of parity bit

Thus P1 = XOR of bit positions (1, 3, 5, 7, 9, 11, 13...)
P2 = XOR of bit positions (2, 3, 6, 7, 10, 11...)
P4 = XOR of bit positions (4, 5, 6, 7, 13....)
P8 = XOR of bit position (8, 9, 10, 11, 12, 13...)

5. To check for the error, check all parity bits by the checker bit.
C1= XOR of bit position (1, 3, 5, 7, 9, 11, 13...)
C2 = XOR of bit position (2, 3, 6, 7, 10, 11...)
C4 = XOR of bit position (4, 5, 6, 7, 13....)
C8= XOR of bit position (8, 9, 10, 11, 12, 13...)

2.1. Calculation of Hamming Code for 8 Bit Data

If we take 8 bit data, then 4 parity bit is needed because 20, 21, 22, 23 , these 4 position are reserved for parity bit.
Let Data Word = D1 D2 D3 D4 D5 D6 D7 D8 and
Parity bit = P1, P2, P4, P8.
P1 = D1 ⊕ D2 ⊕ D4 ⊕ D5 ⊕ D7
P2 = D1 ⊕ D3 ⊕ D4 ⊕ D6 ⊕ D7
P4 = D2 ⊕ D3 ⊕ D4 ⊕ D8
P8 = D5 ⊕ D6 ⊕ D7 ⊕ D8
CODE WORD: P1 P2 D1 P4 D2 D3 D4 P8 D5 D6 D7 D8
Then check the parity bit for detecting the error by check bit.
C1 = P1 ⊕ D1 ⊕ D2 ⊕ D4 ⊕ D5 ⊕ D7
C2 = P2 ⊕ D1 ⊕ D3 ⊕ D4 ⊕ D6 ⊕ D7
C3 = P3 ⊕ D2 ⊕ D3 ⊕ D4 ⊕ D8
C4 = P4 ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D8
If the result, C = C8 C4 C2 C1 =0000, indicates that no error has occurred. However, if C ≠ 0 , then 4 bit binary number is formed and gives the location of the error bit.

3. PROPOSED CIRCUIT OF HAMMING ENCODER AND DECODER

Proposed circuits of hamming code are designed for 4 bit data word. Thus we require 3 parity bits to form 7 bit code word. There has an encoder and a decoder in the proposed circuit to encode and decode the data for error detection and correction.

3.1. Encoder Circuit

Data word is applied as an input in the encoder circuit which performs XOR operations on the given data word and thus the required parity bits are generated from the parity generator. Parity bits and data bits together form the code word. An encoder circuit of hamming code for 4 bit data word is shown below. Following this circuit pattern we can design an encoder circuit of hamming code for 8bit data word and realized it by means of tanner EDA tools.

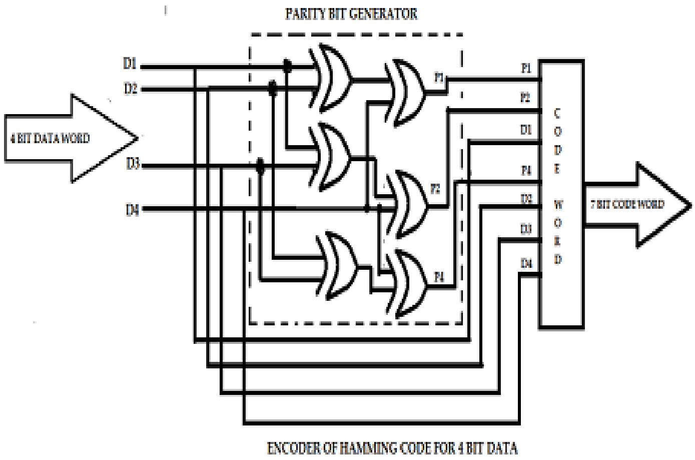


Fig. 2. Encoder Circuit of Hamming Code for 4 Bit Data

3.2. Decoder Circuit

In the decoder circuit, code word is applied as input . Then check bits are generated by the checker bit generator to check the parity bits. These check bits locates the error in the code word by means of decoder circuit. The Output of decoder enables a demultiplexer which are connected to the input code words. If no error occurs then the select line of demultiplexer flows the input form line I0 and the I1 is set to logic '1'. So from the logic OR gate we can obtain the data. Now if an error occur then the select line of the demultiplexer flows the code word from line I1 and I0 is set to logic '0'. Thus inverting the bits, the error bit is corrected and thus we can obtain the error free data. A decoder circuit of hamming code for 4 bit data word is also shown below

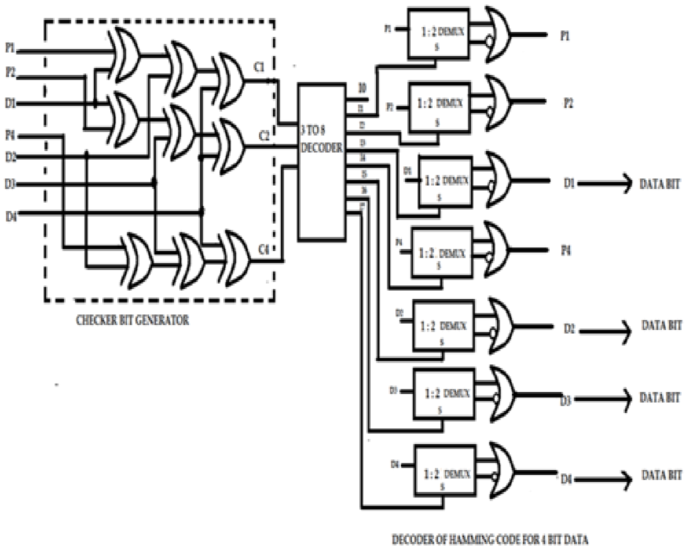


Fig.3. Decoder Circuit of Hamming Code for 4 Bit Data

4. IMPLEMENTATION IN FPGA

For the hamming code implementation, in FPGA we have designed the encoder and decoder with the Hardware Description Language (VHDL) in Xilinx ISE 13.4 design suite Software. We have also implemented the architecture on Spartan 3A family XC3S700A FPGA device. Table I describes computation of parity bits from input data word. Figure 4 describes the simulated waveform of the encoder circuit.

Table 1. Computation Of Parity Bit From Data Word

Input Data Word									Output Parity bits				
Bit No.	1	2	3	4	5	6	7	8	Bit No.	1	2	3	4
Data Word 1	1	1	0	0	0	1	0	0	Parity Bit	0	0	1	1
Data Word 2	0	0	1	1	1	0	1	1	Parity Bit	1	1	1	1
Data Word 3	0	1	0	1	0	0	0	1	Parity Bit	0	1	1	1
Data Word 4	1	0	1	0	1	0	1	0	Parity Bit	1	1	1	0

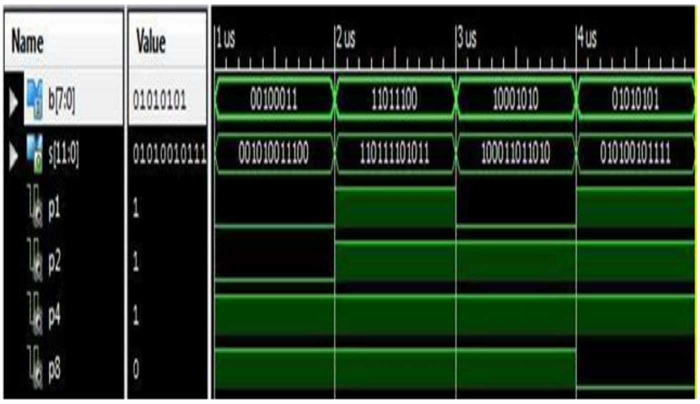


Fig. 4 Simulated Waveform of Encoder

In the decoder circuit we have the code word as the input and checker bits are computed using XOR operation. During the transmission if error occurred in the code word as shown in the code word 2, code word 3 and code word 4 the checker bit shows the position of the error as shown in the Table II .The simulated waveform is shown in the figure 5. This implementation helps for the logical verification of the circuit. The Table III shows the device utilization summary, timing analysis and power utilization of the hamming code encoder and decoder circuit.

Table 2. Computation Of Checker Bit From Code Word

	Input Code Word													Checker Bit			
Bit No.	1	2	3	4	5	6	7	8	9	10	11	12		C1	C2	C3	C4
	0	0	1	1	1	0	0	1	0	1	0	0		0	0	0	0
	1	1	0	1	1	1	1	1	1	0	1	1		0	1	0	1
	0	1	0	1	1	1	1	1	0	0	0	1		0	1	1	0
	1	1	1	1	0	0	0	0	1	0	1	0		0	1	1	0

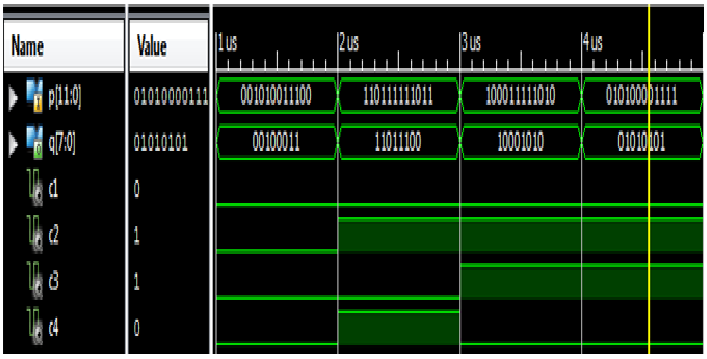


Fig. 5. Simulated Waveform of Decoder

TABLE III. SIMULATION SUMMARY OF ENCODER AND DECODER

	Device Utilization			Timing Summary		Power Utilization
	No of Slice	*No of 4input LUTs*	*No. of bonded IOBs*		*No of Slice*	*No of 4input LUTs*
Encoder	3	5	20	*Encoder*	3	5
Decoder	11	21	20	*Decoder*	11	21

5. IMPLEMENTATION IN TANNER EDA TOOLS

After the verification of logic design, we implemented this encoder and decoder in Tanner EDA tools for circuit design. Tanner EDA provides with a complete line of software solution for the design, layout and verification [5]. We have implemented the circuit of hamming code in tanner using MOS transistors. Simulation of the circuit is done by 45nm, 32nm, 22nm, 16nm MOS technology. In the encoder circuit, Parity generator is designed by 14 XOR gates and each XOR gates are designed by transmission gate. It is the optimized way to design the parity generator using minimum number of MOS transistor for hamming code.

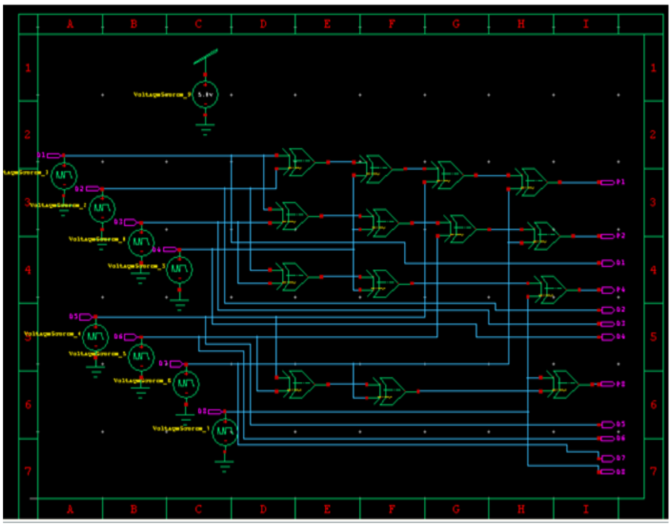


Fig.6. Encoder Circuit of Hamming Code for 8 Bit Data Word.

Similarly, In decoder circuit, checker generator is designed by minimum number of MOS transistor. After the generation of checker bits to trace the location of error, we have implemented 4 to 16 decoder where 13 output lines are only used. The output of the decoder enables the DEMUX. In this circuit, 12 DEMUX is implemented for 12bit code word.

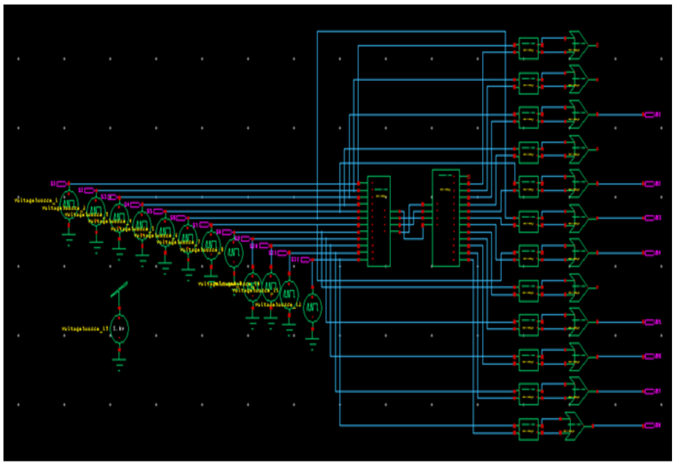


Fig.8. Decoder Circuit of Hamming Code for 8bit Data Word

5.1. Average Power of the Circuit:

In T-Spice of TANNER Tools, We have analysis the power dissipation of encoder and decoder circuit with 1v, 2v of supply voltage for different technology and we assume as shown in the Table IV and V.

TABLE IV. COMPUTATION OF AVERAGE POWER OF ENCODER

Supply Voltage	Average Power Of Encoder			
	45nm	32nm	22nm	16nm
2v	4.13910w	3.90823w	3.83136w	3.29828w
1v	3.90495w	3.89752w	3.841300w	3.355061w

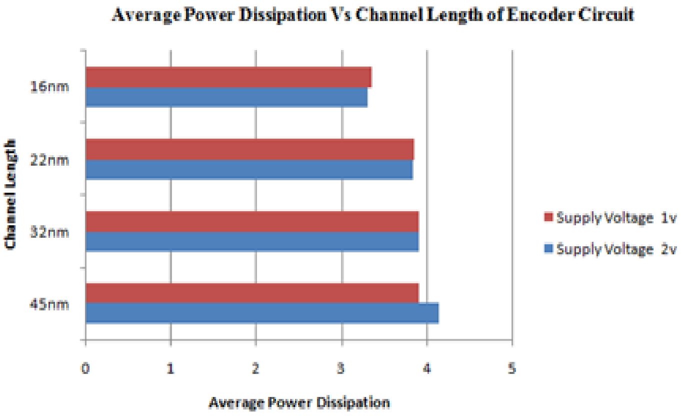


Fig. 9. Total Power Vs Channel Length of Encoder

Fig 9 shows that total power dissipation of encoder is changing with the variation of supply voltage for 16nm, 22nm, 32nm, 45nm.

TABLE V. COMPUTATION OF AVERAGE POWER OF DECODER

Supply Voltage	Average Power Of Decoder			
	45nm	32nm	22nm	16nm
2v	4.56090w	4.33613w	4.32417w	4.33314w
1v	3.668174w	3.624410w	3.264414w	3.187025w

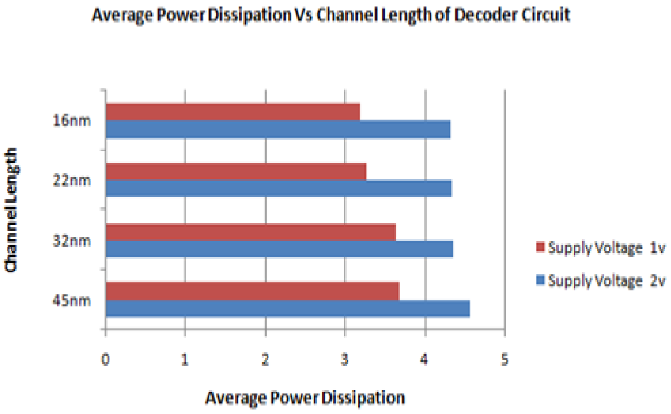


Fig. 10. Total Power Vs Channel Length of Decoder

Fig 10 shows that total power dissipation of decoder is changing with the variation of supply voltage for 16nm, 22nm, 32nm, 45nm.

6. CONCLUSIONS

Thus we have implement the hamming code in VHDL language as well as in TANNER Tools. From table 4 and 5 , we can find the minimum power dissipation varying the supply voltage for different technology. The analysis shows that with the decrease of channel length, decrease of power dissipation as a result there is a decrease of 12.65% and 2.35% power in 16nm compared to 22nm in encoder and decoder circuit.

REFERENCES

[1] Digital communication, Sanjay Sharma, 2nd edition SK Kataria & Sons Pulication
[2] Implementation of Hamming code using VLSI Nutan Shep1, Mrs. P.H. Bhagat2 Department of Electronics & Telecommunication Dr .B.A.M.U,Aurangabad Government College of Engineering Aurangabad (M.S.)
[3] Design of (15, 11) Hamming Code Encoding and Decoding System Based on FPGA, Tongsheng Zhang, Qun Ding Heilongjiang University Key Laboratory of Electronic Engineering, College of Heilongjiang Province Harbin, China, 2011
[4] Data communication and networking , Behrouz A Forouzan,4th edition
Tata McGrawHill Publication
[5] http://www.tannereda.com/

BIOGRAPHIES (Optional)

Debalina Roy Choudhury is a M.Tech Student in the Electronics and Communication Engineering-VLSI Department, Heritage Institute of Technology, WBUT. She received Bachelor of Engineering (B.E) degree in 2013 from Tripura Institute of Technology, Narsingarh, Tripura, India. Her research interests are VLSI Design.

Krishanu Podder holds a Post Graduate Diploma in Advanced computing from CDAC SunBeam Pune. He received Bachelor of Engineering (B.E) degree in 2013 from Tripura Institute of Technology, Tripura. His research interest is VLSI Design and Computer languages.