New Approach for Channel Encoding and Decoding Using (8,4) Extended Hamming Code

M.M.M. Mushfick*, A.Puspatheepan *,
Dilanka De Silva†
* † School of Engineering, Sri Lanka Technological Campus, Sri Lanka
Email:*[mushfickm,pushpadeepana]@sltc.edu.lk,†[dilankad@sltc.ac.lk]

***Abstract*—In this paper, the Channel Encoding and Decoding process is implemented using the (8,4) Extended Hamming code, in Quartus II 13.0 design software through VHDL programming language and ModelSim-Altera software for simulating the circuit designs. With a minimum Hamming distance of four, Single-Error Correction and Double-Error Detection (SEC-DED) can be achieved. Dissipation of power through heat and high processing time are the major shortcomings of using non-reversible logic gates. The combinational logic gate designs are built using reversible logic to reduce dissipation of heat and processing delay. FPGAs are preferred over micro-controllers for the implementation of the designs, due to low signal processing latency and high parallel processing capability. The detection and correction of errors is achieved by following a new effective concept/algorithm to achieve SEC-DED at the decoder. The hardware implementation of the prototype was done using the Altera DE0-Nano FPGA board**

***Index Terms*—Check-Bits Generator (CBG), Double Bit Error (DBE), Error Detection and Correction(EDC), Garbage Output, Reversible gates, Single Bit Error (SBE), VHSIC Hardware Description Language (VHDL).**

I. Introduction

Digital communication gradually started to replace the pre-existing analog communication owed to various reasons. The effect of distortion, noise and interference is comparably less in effect for digital signals. But above all, the ability to deploy error detecting and correcting algorithms is one of the most significant benefits. [1].

Channel Encoder performs the addition of redundant bits at the transmitter while the Channel Decoder performs the EDC functionality at the receiver, in a digital communication system. Hamming coding is a block-code based channel coding technique, where additional parity bits are added to the original message bits to aid the EDC at the receiver. Minimum Hamming Distance (d_{min}) of the code-words is used to determine the EDC capacity of the code. The EDC capacity of the Hamming code is determined using the following relationships.

$$E_d = \frac{d_{min}}{2}, \tag{1}$$

$$E_c = \frac{d_{min} - 1}{2}. \tag{2}$$

where E_d and E_c denotes the error correction and error detection capacity respectively,lower truncated for decimal values. Thus, with a d_{min} of 4, all double-bit errors can be detected and all single bit errors can be corrected. The conventional (7,4) Hamming code, which has a (d_{min}) of 3 can be further modified as the Extended Hamming code (8,4) to achieve the d_{min} of 4. This is achieved by an additional redundant bit (P_{TX}), which is the overall parity check bit of the preceding 7 bits.

A. Motivation

The Encoder, CBG and the Decoder are generally designed using irreversible logic, which leads to very high consumption/dissipation of power as heat. This is not ideal in designing low power consuming circuits in the real world. This problem can be controlled by designing the circuits through reversible logic. In particular, reversible gates such as Feynman Gate and Feynman-Double Gate, which have the least quantum cost of all reversible gates, are most suitable for this application.

The decoding process at the receiver can be done by following a simple effective algorithm by comparing the Check-bits value, the overall Transmitter parity ($P_{TX'}$) bit at the receiver and the overall Receiver parity bit (P_{RX}).

B. Related Work in Literature

Based on the paper [2], the use of simple parity allows detection of single-bit errors in a received message. Correction of these errors requires more information, since the position of the corrupted bit must be identified, if it is to be corrected. Further, if more bits are included in a message, and if those bits can be arranged such that different corrupted bits produce different error results, then corrupted bits could be identified. The circuits that are designed in [1] involve the use of only XOR gates (non-reversible gates), 3 TO 8 decoder, eight 1:2 demultiplexers. The power consumption and dissipation for several MOS technology (45nm, 32nm, 22nm and 16nm) has been identified by simulating the design, which resulted in a considerably high consumption of power. But [3] points out that working with reversible logic over non-reversible logic minimizes the power that dissipates as heat since reversible computation doesn't require the erasing of information bits which consequently leads to the elimination of energy loss. Reaffirming this theory, the [4] does a mathematical analysis of quantum cost calculation, garbage outputs, delay and power dissipation in using Reversible logic gates for a (7,4) hamming code scheme. The simulations

resulted in a significant reduction in power consumption. The paper [5] reviews the reversible logic, its applications and also presents the comparative study of various combinational and sequential circuits which use different reversible gates. There were various methods adopted for the EDC process. In [6], the author proposes that comparing the received codeword with all correct code words will aid in finding the minimum distance and then determine the type of error (Single error or Double error). Meanwhile, [7] proposes the syndrome decoding method involving the generator matrix and check matrix for EDC.

C. Paper Structure

This paper has been categorized into 6 sections. The designing process of the Hamming encoder will be discussed on II. The section III, contains a comprehensive explanation on the designing of the Check-bits Generator. A thorough explanation of the Hamming decoder design and the algorithm/concept followed for EDC is discussed in section IV. The section V will demonstrate the simulation process, results obtained and its analysis. Finally, the research is concluded in the section VI

II. Hamming Encoder

The arrangement of the 4 message bits and the parity bits (3 general Parity bits + The overall Transmitter parity bit (P_{TX}) in the encoded data byte is illustrated in Figure 1.

Fig. 1: (8,4) Hamming code block arrangement

The general parity bits P1, P2 and P4 are as described in the following logical expressions [1]. Although P_{TX} is the overall Transmitter parity, it can be further simplified using Boolean algebraic theories and written in a much simpler form.

$$P1 = D1 \bigoplus D2 \bigoplus D4, \tag{3}$$

$$P2 = D1 \bigoplus D3 \bigoplus D4, \tag{4}$$

$$P4 = D2 \bigoplus D3 \bigoplus D4, \tag{5}$$

$$P_{TX} = P1 \bigoplus P2 \bigoplus D1. \tag{6}$$

Utilizing the above equations for the parity bits, the combinational logic gate design for the Hamming encoder was formed using Reversible gates (4 Feynman Gates and 4 Feynman Double gates each) only, as shown in Figure 2. This encoder has 4 input bits which is the original message and 8 outputs which is the resulting Hamming encoded message. The encoder design resulted in zero garbage outputs.

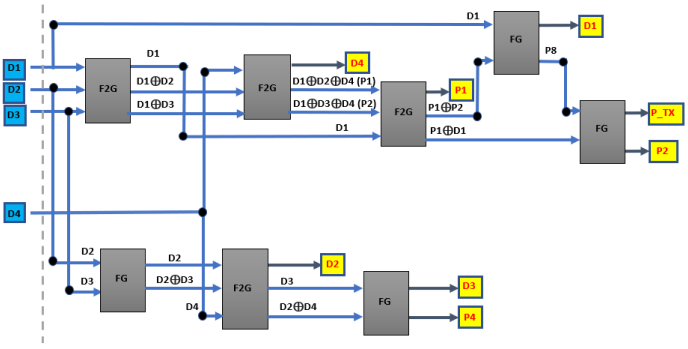

Fig. 2: Combinational logic gate design for the Encoder

The 8-bit output from the encoder goes through a non-ideal channel, which is simulated through VHDL programming in Quartus Platform. This channel randomly introduces error(single-error, double error or zero error) at random positions of the 8 bits. This is done to check the SEC-DED capability of the receiver for all possible events.

III. Check-Bits Generator

The Check-bits Generator calculates the check bits to determine how safely the transmitted byte of data has reached the receiver [8]. The check bits C1, C2 and C4 is determined using the following relationships [1].

$$C1 = P1' \bigoplus D1' \bigoplus D2' \bigoplus D4', \tag{7}$$

$$C2 = P2' \bigoplus D1' \bigoplus D3' \bigoplus D4', \tag{8}$$

$$C4 = P4' \bigoplus D2' \bigoplus D3' \bigoplus D4'. \tag{9}$$

In the above realtionships, the complement symbol denotes that it is a received bit. The check bits when arranged in the order [C4 C2 C1], with C4 being the most significant bit, exactly points out the bit position that is in error and that particular bit needs to be inverted to decode the original message. The all-zero check-bit value corresponds to the zero-error case. This is applicable for single error correction only. The double error detection functionality will be discussed in the Hamming Decoder section. The relationships for the Check-bits can be utilized to formulate the combinational logic design of the check-bits generator, shown in Figure 3. This design was prepared using 2 Feynman Double Gates and 4 Feynman gates. Although there are 4 garbage outputs generated from the design, among the four, two of the garbage outputs can be further utilized to determine the overall Receiver parity bit (P_{RX}), which brings down the total number of garbage outputs to only 2.

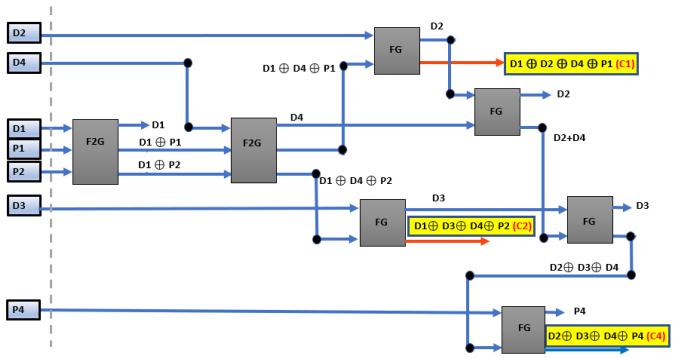

Fig. 3: Combinational logic gate design for the Check-Bits Generator

IV. Hamming Decoder

The Hamming decoder has three important functions to carry out. The first of them would be to determine whether error has occurred in the received set of bits and also the type of error that has occurred. After determining the type of error, if it's either single error or zero error case, the decoder should be able to correct the error and decode the originally transmitted message, and send a positive acknowledgement in return. But if double bit error has occurred, it should detect the event and send a negative acknowledgement in return.

To differentiate among the three possible events that can occur, the decoding algorithm uses the Check-bit value, received $P_{TX'}$ and the overall Receiver parity bit (P_{RX}). The logical expression for P_{RX} is simplified using Boolean algebraic theories for reducing complexity and shown in the following expression

$$P_{RX} = D3' \bigoplus P2' \bigoplus P4' \bigoplus C2. \tag{10}$$

The design for P_{RX}, illustrated in Figure 4, is determined from the garbage and non-garbage output bits of the Check-bits generator design. This design required 3 XOR gates. In this instance, the reversible logic gates couldn't be used as it results undesired number of garbage outputs.

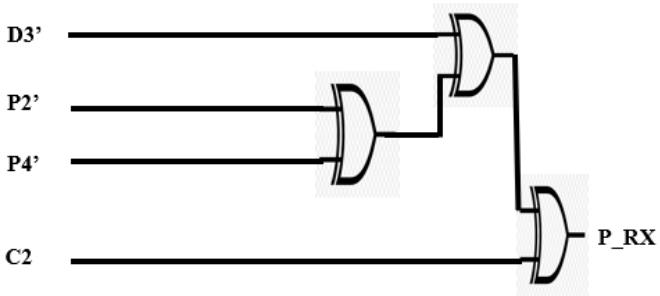

Fig. 4: Combinational logic gate design for the P_{RX} bit

A. Decoding of Zero-Error

In this case, there is no error in the 8 bits that is received at the receiver. The check-bits generator will generate the check-bits, which will denote the value 'zero'. Hence, there is no requirement for any error detection or correction. The bits at bit positions 3, 5, 6 and 7 will be decoded as the originally transmitted message. This is illustrated in Table I.

B. Decoding of Single-Error

Considering even parity check method, if there are even number of ones in the first 7 bits, the P_{TX} bit will be '0' and if there are odd number of ones in the first 7 bits, the P_{TX} bit will be 1. In the event of single bit error, either one of the first 7 bits or else the P_{TX}' bit would be erroneous. When one of the first 7 bits is in error, the Check-bits value will be a non-zero value and P_{TX}' will not be equal to the P_{RX}. Here, The Check-bits value points the erroneous bit position which can be corrected subsequently. In the case of P_{TX}' bit is in error, the Check-bits value to be zero and yet again P_{TX}' will not be equal to P_{RX}. Here, there is no requirement for any correction since there is no error in the first 7 bits. The bits at bit positions 3, 5, 6 and 7 will be decoded as the originally transmitted message. This is illustrated in Table I.

C. Decoding of Double-Error

In the event of double bit error, either two of the first 7 bits or one of the first 7 bits and the P_{TX}' bit will be erroneous. In the first case, where two of the first 7 bits is in error, Check-bits value will be non-zero value and P_{TX}' bit will be exactly equal to the P_{RX} bit. Here, The Check-bits value will no longer point the erroneous bit position since two of the first 7 bits are in error. Similarly, in the 2nd case, where one of the first 7 bits and the P_{TX}'bit are in error, that will make check-bits value to be a non-zero and still P_{TX}' bit will be exactly equal to the P_{RX} bit. Therefore, in the event where check-bits value is non zero and P_{TX}' bit equals P_{RX} bit, it can be concluded that double error has occurred in the received data byte as summarized in Table I.

TABLE I: Summary of Decoding Algorithm

Check-Bits Value (C)	P_{TX}' Bit	P_{RX} Bit	Decision
C = 0	0	0	No Error
C = 0	1	1	No Error
C = 0	1	0	P_{TX}' Error
C = 0	1	0	P_{TX}' Error
C > 0	0	1	**Single-Bit Error**
C > 0	1	0	**Single-Bit Error**
C > 0	0	0	**Double-Bit Error**
C > 0	1	1	**Double-Bit Error**

V. Simulation Results

A. RTL Schematic Designs

The Combinational logic gate designs were created in Quartus II 13.0 design software through VHDL programming language. The RTL schematic designs resulted for Encoder, Encoder+Noise Channel, Check-Bits generator and the Decoder are illustrated in Figures 5, 6, 7 and 8 respectively.

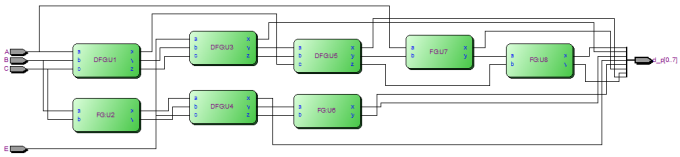

Fig. 5: RTL Schematic of the Hamming Encoder

The Hamming Encoder and the noise channel programmed together to ensure that the encoded data byte gets randomly affected with a maximum bit error of 2, before it reaches the Receiver.

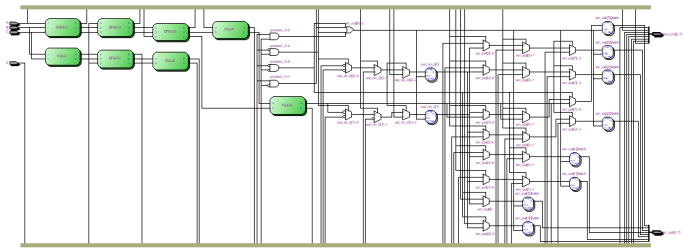

Fig. 6: RTL Schematic of the Hamming Encoder+ Noise Channel

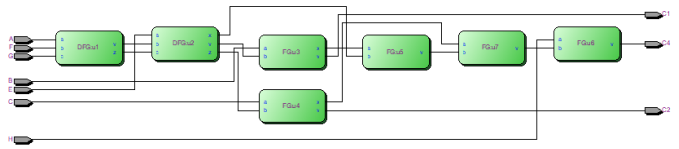

Fig. 7: RTL Schematic of the Check-Bits Generator

The received Data Byte initially goes through the Check-Bits Generator and then moves to the Decoder for EDC.

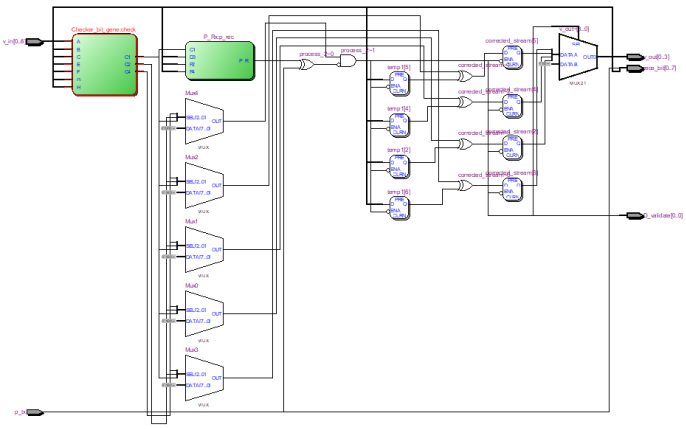

Fig. 8: RTL Schematic of the Hamming Decoder

B. *ModelSim-Altera Simulation*

The VHDL programs are simulated using ModelSim-Altera 10.1d Software. The simulation output for Encoder + Noise channel program is shown in Figure 9, in which the *enco_bit* signal is exactly similar to the *err_out signal*. This event represents an error free transmission. The Figure 10 illustrates the scenario where the *enco_bit* signal differs the *err_out* signal at a single position, which is the single error transmission. Meanwhile, the Figure 11 represents the event where the *enco_bit* signal differs the *err_out* signal at two separate positions, demonstrating the double error transmission.

Name	Value	Kind	Mode
A	0	Signal	In
B	1	Signal	In
C	0	Signal	In
E	0	Signal	In
enco_bit	10011001	Signal	Out
err_out	10011001	Signal	Out

Fig. 9: Simulation of error-free Transmission

Name	Value	Kind	Mode
A	0	Signal	In
B	0	Signal	In
C	1	Signal	In
E	0	Signal	In
enco_bit	01010101	Signal	Out
err_out	01011101	Signal	Out

Fig. 10: Simulation of Single-error Transmission

Name	Value	Kind	Mode
A	1	Signal	In
B	0	Signal	In
C	0	Signal	In
E	0	Signal	In
enco_bit	11100001	Signal	Out
err_out	11110101	Signal	Out

Fig. 11: Simulation of Double-error Transmission

The ModelSim-Altera simulation results for the Check-Bits Generator + Decoder are shown in Figure 12, where the error-free transmission is decoded. The *chebit* signal denotes '0 0 0' in this case. The *v_out* signal represents the decoded message.

The Figure 13 illustrates the scenario where single error transmission is decoded. The *D_validate* signal represents the detection of double error. Here, *D_validate* signal is LOW and the non-zero *chebit* signal represents the position where SBE has occurred. The message is decoded and represented by the *v_out* signal. Meanwhile, the Figure 14 represents the event where double-error transmission is decoded. . Since the *D_validate* signal is HIGH, it has successfully detected the Double error.

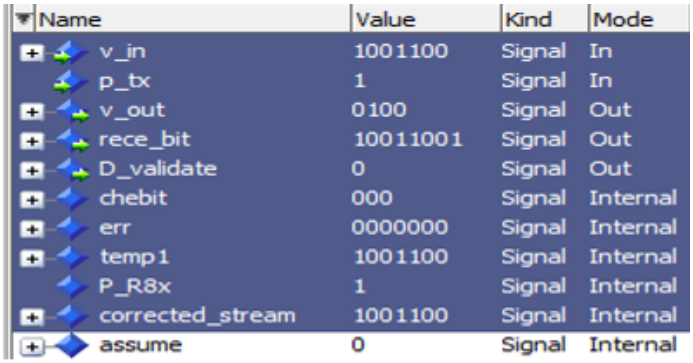

Fig. 12: Simulation of Error-free Decoding

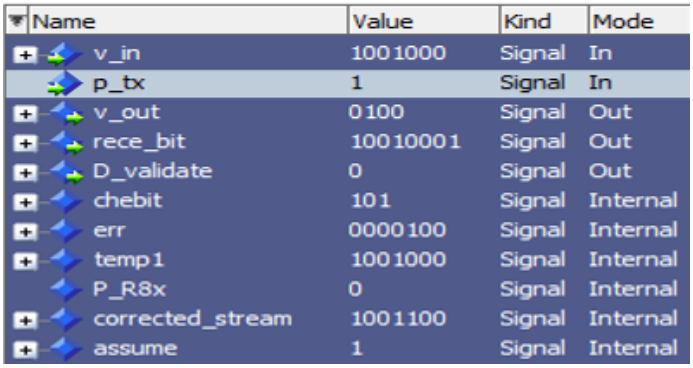

Fig. 13: Simulation of Single-Error Decoding

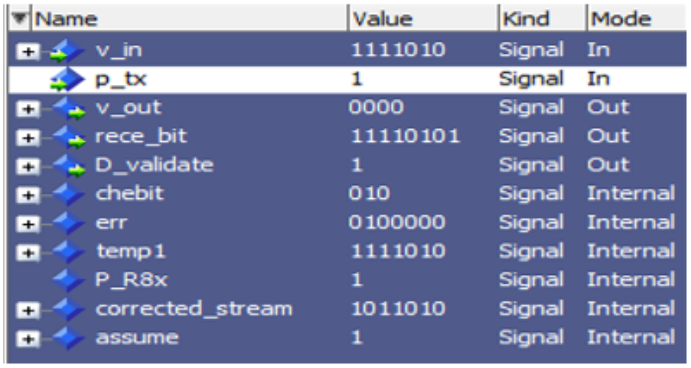

Fig. 14: Simulation of Double-Error Decoding

In the event where the message is successfully decoded, a positive acknowledgement is sent back to the transmitter and a negative acknowledgement when double-error is detected.

C. Power Dissipation of the design circuits

The authors in [1] have proposed the designing of (7,4) Hamming code encoding and decoding circuits using irreversible logic gates. Its simulation resulted an average power dissipation of 3.7W and 4.4W for the Encoder and CBG+Decoder circuits respectively.

Taking this into account, the power consumed by each individual component was estimated using Altera-PowerPlay Early Power Estimator Software and tabulated in Table II. The static power consumption was constant for all the individual components but the thermal power(dissipation as heat) was observed to be varied.

TABLE II: Estimated Dissipation of Power

	Encoder	**Check Bits Generator**	**Decoder**
Static Power (W)	0.097	0.097	0.097
Thermal Power (W)	0.004	0.001	0.003
Total Power (W)	0.101	0.098	0.100

In comparison to the values observed in [1], due to the usage of reversible logic, the power dissipation of the encoder circuit has been reduced from 3.7W to 0.101W and that of the CBG+Decoder has been reduced from 4.4W to 0.198W.

VI. Conclusion

This Paper proposes a new approach to the Error Detection and Correction process using (8,4) Extended Hamming code. The decoder corrects all possible single errors and detects all possible double errors that occur in the transmission. The programs were build using VHDL in Altera-Quartus Platform and their functionalities were verified by simulating them in ModelSim-Altera software. The power consumption of the designs was determined using the Altera-PowerPlay Estimator software. The dissipation of power has been effectively reduced by employing reversible logic for the designs. After the simulations yielded successful results, the programs were implemented on the Altera DE0-Nano FPGA board, which led to successful results.

References

[1] K. P. Debalina Roy Choudhury, "Design of hamming code encoding and decoding circuit using transmission gate logic," *International Research Journal of Engineering and Technology*, vol. 02, no. 07, pp. 1165–1169, 2015.

[2] V. Jindal, "Developing hamming code using verilog hdl," pp. 94–96, 2006.

[3] R. Landauer, "Irreversibility and heat generation in the computing process," *BM Journal of Research and Development*, vol. 05, no. 03, pp. 183–191, 1961.

[4] S. P. Nayak, C. Madhulika, and 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)*, pp. 778–781, 2017.

[5] R. Khanam and P. Abdul Rahman, "Review on reversible logic circuits and its application," *International Conference on Computing, Communication and Automation (ICCCA2017)*, pp. 1537–1542, 2017.

[6] K. Jamieson, *Detecting and correcting bit errors*, cs . princeton.edu/courses/archive/spring18/cos463/lectures/L08-error-control.pdf.

[7] Q. Ding and T. Zhang, "Design of (15, 11) hamming code encoding and decoding system based on fpga," *International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 704–707, 2011.

[8] Monika and Kavitha, "Design and analysis of hamming code encoding, decoding and correcting circuits using reversible logic," *International Journal of Research*, vol. 07, no. 12, pp. 36–41, 2018.