Implementation of Low Bandwidth ECC Code on FPGA

Ramalingegowda.H.L
Department of Electronics& Communication Engineering
PES College of Engineering,
Mandya, Karnataka

Jyothi S
Department of Electronics& Communication Engineering
PES College of Engineering,
Mandya, Karnataka

***Abstract*— In this paper a method to detect two simultaneous bit errors and correct a single bit error during data transmission is devised. The proposed method is developed based on Hamming Code. The key point for the implementation of error-free data transmission is the encoding of the information to be transmitted in such a way that some extent of redundancy is included in the encoded data, and a method for efficient decoding at the receiver is available. These two requirements have been achieved in the new method in an efficient and simple way. In this paper, one bandwidth reduction technique is included along with Hamming code. The proposed method is implemented using XILINX, and has been demonstrated through examples.**

Keywords— **Hamming code, Redundancy Bits, Parity Method, Double Data Rate Transmission, FPGA.**

I. INTRODUCTION

Data transmission is the physical transfer of data over a point-to-point or point-to-multipoint communication channel. In communication system, data transmission may be in the form of Analog or Digital. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. It is physically impossible for any data recording or transmission medium to be 100% perfect. Thus, error detection and correction is critical to accurate data transmission, storage and retrieval. An error is a deviation from a correct value caused by a malfunction in a system or a functional unit. Errors can completely change the meaning of the data sent. Error occurs because of transmission impairments. Single Bit and Burst errors are the most common form of errors affecting data transmission. Single bit error mainly occurs in parallel transmission and Burst error is most likely to happen in a serial transmission [3].

Error detection is the ability to detect the presence of errors caused by noise or other impairments during transmission from the transmitter to the receiver. Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination. Error coding is a method of detecting and correcting the errors to ensure information is transferred intact from the source to destination. Error coding uses mathematical formulas to encode data bits at the source into longer bit words for transmission. The code word can then be decoded at the destination to retrieve the information.

Hamming code error detection and correction methodology is used for error free communication in communication system. Hamming code is a linear error-correcting code named after its inventor, Richard Hamming. Hamming codes can detect up to two simultaneous bit errors, and correct single-bit errors. In hamming code error detection and correction technique is used to get error free data at destination, information data according to parity check method is encrypted before transmission of information at source end. Hamming codes are code words formed by adding redundant check bits, or parity bits, to a data word [7] [1].

In this paper, a novel method is devised for elimination of errors and improving the speed of data transmission. The proposed method is explained using the transmission of 8-bit data information. In this work hamming code is implemented to minimize the errors occurring during transmission, for this 5 redundancy bits are used. The resultant 13 bit information is required to be transmitted to the destination end. The time taken to transmit 13 bit data is more compared to that of original 8 bit data [1]. In the proposed work, DDR method is implemented along with Hamming code to reduce the Bandwidth of the data during transmission, and thereby increasing the speed of data transmission compared to that of the original 8-bit data.

In this paper, VHDL language is used for writing VHDL code for source and destination. At source, VHDL code is used for generating 5 redundancy bits for 8 bit information. At destination, VHDL code is written for detecting presence of error and correction. Further VHDL code is written for implementation of DDR method and finding 8-bit actual information data from the 13 bit received encrypted data. Here Xilinx ISE 12.3 simulator is used to simulate VHDL code and to obtain the simulation results [2] [4] [13]

II. RELATED WORK

Hamming code is well known for its single-bit error detection and correction capability. In one of the earlier works done in the field of error detection and correction, is the improved hamming code for error detection and correction [8]. In this method the redundancy bits were appended at the end of data bits. This method reduced the effort of interspersing the redundancy bits at the sender end and their removal at the receiver end after checking for single-bit error and consequent correction, if any. Due to this,

IJERT

the effort needed in identifying the values of the redundancy bits was lower compared to normal hamming code method.
Another promising work done in this field is 30 BIT Hamming Code for Error Detection and Correction with Even Parity and Odd Parity Check Method by using VHDL [6]. In this method, error detection and correction of 25 bit data, even and odd parity check methods in the hamming code at the source and destination ends was devised. VHDL code was written for finding error location and to correct error bit in the above mentioned method. This method eliminated the requirement of two different circuits, one for correcting error bit and another for finding the information data from encrypted data.
In another novel work, titled Indication of Efficient Technique for Detection of Check Bits in Hamming Code [9], two different schemes for detecting the parity check bits is introduced and compared. The paper explains the use of Ex- OR logic and AND-OR logic and sketches a general idea on their performance and efficiency compared to each other. The paper concludes that due to the number of increasing gates in a circuit the propagation delay increases, out of the above mentioned techniques the use of Ex-OR logic is found to be better than AND-OR logic.

In one of the works titled Decoder Design for a New Single Error Correcting/Double Error Detecting Code [10] presents the decoder design for the single error correcting and double error detecting code. The paper mentions that the speed of error detection and correction of a code is largely dependent upon the associated encoder and decoder circuits and the complexity and speed of such circuits are determined by the number of 1's in the parity check matrix. The paper proposes a simplified encoding and decoding circuitry for error detection and correction.

Numerous research works has been carried out in the field of error detection and correction. However, only few research papers have been mentioned here. In all the above mentioned methods, parity bits are added by increasing the length of data string. Transmission of the modified data takes longer time compared to the original data. In the proposed work, Hamming code is used for the double bit error detection and single bit error correction. In addition to this, Double data rate technique is implemented to reduce the bandwidth during transmission thereby improving the speed of data transmission [1][6][7].

III. PROPOSED METHOD

In this work, with the help of 8-bit information data transmission, the use of Hamming code along with Low band width data transmission is proposed. The proposed error correction scheme is found to improve the speed of data transmission.

A. Source Section

1) *Parity Method:* During transmission of information, it is required to know whether error has occurred during transmission. Parity method is one such procedure to find out if any error has occurred during data transmission. In Parity Method, extra bits known as redundant bits are added. For 8-bit data transmission, 5 redundant bits are needed. Suppose the redundant bits are parity(0), parity(1), parity(2), parity(3), parity(4). The value of these redundancy bits is found by XORING the different location of information data [6] [7]. The proposed scheme is explained with the help of information data string "01100001". For single bit error correction four redundant bits are used.
Calculation for redundancy bit parity(0) is done by XORING input bit address given below

parity(0) = 2,4,6,8,10

The value of parity(0) = 0.
Similarly the redundancy bit parity(1), parity(2), parity(3) is given by:

parity(1) = 2,5,7,9,10 parity(2) = 4,5,7,11
parity(3) = 8,9,10,11

The value of parity(1) is 1 (as the number of one`s are 3) value of parity(2)is 0 (as the number of one`s are 0) the value of parity(3) is 0 (as the number of one`s are 2). Now we know the value of redundancy bits is "0010". So the encrypted 12-bit data string is "011000000110". For Double bit error detection, an extra redundant bit denoted by parity(4) is used. Calculation of the additional parity bit, parity(4) is obtained by XORING other 12 output bits, which is represented by using the bit address positions as shown below:

parity(4) = 1,2,3,4,5,6,7,8,9,10,11,12.

Thus, the value of parity(4) is 0. Due to addition of an extra bit, 13-bit data string "0011000000110" is obtained [5] [6] [14] [15].

2) *Low Bandwidth Data Transmission:* The task of reducing the bandwidth of the transmitted data is done by using Double Data Rate (DDR) Transmission method. DDR uses rising and falling edges of a clock to transmit data, which facilitates data transmission at twice the rate of a Single Data Rate (SDR) architecture, using the same clock speed. A main clock is considered in the present work for data transmission and is denoted by 'clockM'. The clockM is multiplied by 2 to obtain a clock with twice the frequency of clockM and denoted as 'clock1'. The clock multiplication is done by using clock multipliers. The data transfer takes place at the rising edge of the clock1. Many clock multiplier ICs are available, however the IC used in the present work is 'CDCF5801A'. The CDCF5801A is PLL based multiplier which provides clock multiplication from a reference clock signal. Using this clock multiplier, the clock multiplication of x1, x2, x3 and x4 can be achieved by using the required selection pins available in the IC. The main advantages of this IC are low power consumption and no external circuit requirement for its functioning [12].

The addition of redundancy bits to the original data in an error detection method results in increase of bandwidth of data transmission. This problem is overcome by use of DDR transmission technique. DDR method uses rising and falling edges of a clock to transmit data, which facilitates data transmission at twice the rate of Single Data Rate (SDR) architecture. The most significant component in the DDR technique is a PLL based multiplier IC. In this work, the PLL

IJERT

based multiplier IC, is used to multiply the main clock by 2 which facilitates the transmission of data at every rising and falling edge of the main clock pulse. The use of multiplier reduces the bandwidth of the data and in turn improving the speed of data transmission compared to that achieved through Hamming code method

3) *Serial Data Transmission:* After the addition of the Parity bit to the original data, the obtained information is in the parallel form. However, for long distance transmission, serial data transmission is preferred over Parallel data transmission. In this paper, Multiplexer is used for converting parallel data into serial form. Multiplexer (or MUX) is a device that selects one of the several Analog or Digital input signals and forwards the selected input into a single line. An electronic multiplexer can be considered as a multiple-input, single-output switch. In the above mentioned case, after the implementation of Hamming code method the output obtained is a 13-bit code word. To convert 13-bit code into serial form, it is required to design a multiplexer with 16 input signal lines that are present at the input portion of the multiplexer. To control this multiplexer 4 selected lines are needed. These 4 selected lines are designed to select one of the 13-bit inputs to the outputs. The 16x1 multiplexer is used in this project. In the 16x1 multiplexer, four selected lines are required, i.e., sel1(0), sel1(1), sel1(2) and sel1(3). Out of which sel1(0) is the LSB and sel1(3) is the MSB. Depending upon these selected lines, the corresponding input is selected to the output.

4) *Counter Design:* The counter acts as an auto selection of MUX and DEMUX. The Binary counter is connected to the data selector lines of multiplexer, .i.e., the selected lines 'sel1(0)' to 'sel1(3)' is selected by the counter. A counter is a device that generates some patterned binary value depending on a clock or some other pulsed input. Counter is able to count from a specific number to another specific number. In this project 4-bit synchronous up-counter is used. In a Synchronous counter, all state bits change under control of a single clock. The single clock used is clock1. It is designed to count from '0' to '12'.

The Source section is a combination of Parity scheme and Low bandwidth data transmission section. The block diagram for the source section is shown in Fig.1.

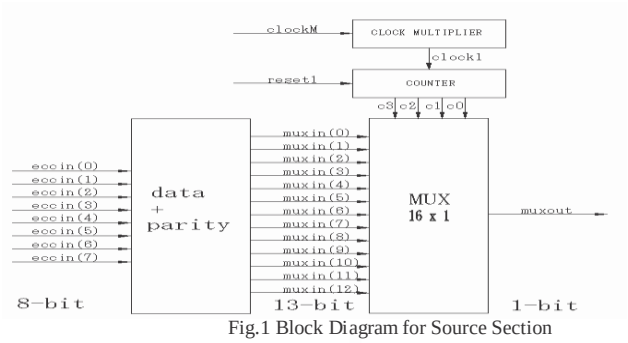

Fig.1 Block Diagram for Source Section

The calculation of redundancy bit and the implementation of DDR method is done by VHDL code using Xilinx ISE 12.3 and the input and output simulated result is shown in Fig.2.

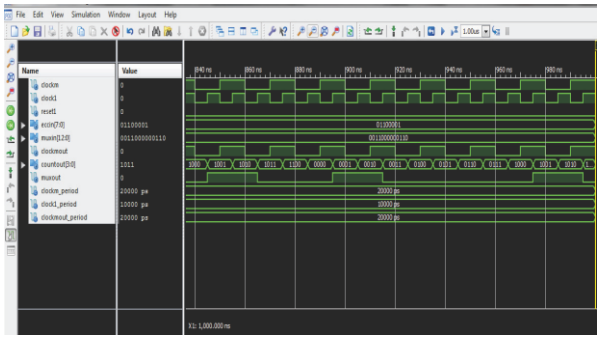

Fig.2 Xilinx ISE 12.3 Output Window for Source Section

B. Destination Section

1) Serial to Parallel Data Conversion: At the receiver section, data is again converted back to parallel form. This is achieved with the help of a demultiplexer. Demultiplexer is a digital function that performs inverse of the multiplexing operation. At the destination section, the received 13-bit data is converted back to parallel form with the help of 1x16 demultiplexer. An electronic demultiplexer can be considered as a single-input multiple-output switch. They are digital switches which connect data from one input source to one of n outputs. To control the 1x16 demultiplexer, 4 selected lines are needed. Like in multiplexer at the transmitter section, these selected lines are also selected by 4-bit synchronous up counter. The 4 selected lines are denoted as sel2(0), sel2(1), sel2(2) and sel2(3). Out of which sel2(0) is the LSB and sel2(3) is the MSB. Depending upon these selected lines, the corresponding input is selected to the output.

2) Destination Section with parity Method: After adding redundancy bit in 8-bit information data, 13-bit encrypted information data is transmitted by transmitter at source end. Destination section receives 13-bit encrypted data and check for errors. If any error has occurred, receiver finds the error location and corrects this error bit. The Number of address of error bit is equal to the number of redundancy bit added by transmitter before transmitting data. In the receiver side, the first step is to compute the parity check bit from the received bits at the receiver section. In this paper the parity bit is calculated for checking presence of any single bit error in the received data and are represented as calparity(0),calparity(1), calparity(2),calparity(3) [5][6][7]. Calculation for redundancy bit calparity(0) found by XORING input bit addresses is as shown below:

Calparity(0)=0,2,4,6,8,10

Similarly ,the remaing redundancy bits obtained are as follows

Shown below:

Calparity(1)=1,2,5,6,9,10
Calparity(1)=3,4,5,6,11
Calparity(1)=7,8,9,10,11

The data string received from transmitter after adding redundancy bit is "0011000000110".

For detection of double bit errors, the 13th bit out of the received 13-bit data string is evaluated and denoted by'calextra'. The value is calculated by XORING the entire 13 bits received at the receiver section and is given

by:

calextra = 0,1,2,3,4,5,6,7,8,9,10,11,12.

In the hamming code, double bit error detection and single bit error correction at the destination section is based on Table 1

TABLE I

If calparity = 0 and calextra = 0	No error occurred
If calparity ≠ 0 and calextra = 1	A single error occurred that can be corrected
If calparity ≠ 0 and calextra = 0	A Double error occurred that is detected but cannot be corrected
If calparity= 0 and calextra = 1	An error occurred in the 13th bit

In order to explain the procedure for detection of error it is assumed that the received data is noisy. Receiver locates the error bit, after finding the error location the error is corrected by the receiver [14][15][16]. Let the received data be "0111000000110". As mentioned above, first the parity bit is calculated. If the parity bit calculated is "0000", no error occurred during transmission. For the above case, the parity bits obtained is 1100. After the calculation of parity bits, the extra parity bit is denoted by 'calextra' is calculated and the resultant bit value is 1. That is, as per the Table 1 the calparity

0 and calextra = 1, that means, single error has occurred. The position of the single error is located, using calparity. The calculated parity bit is 1100, i.e., bit in the 12^{th} position is errored. As per the data string the value at the 12th bit is 1. So flipping the value of 12^{th} bit from '1' to '0' the error can be corrected. That means the corrected data is "0011000000110" obtained. Parity bits are removed and the original data bit is recovered from the address location 2,4,5,6,8,9,10,11. The recovered data is "01100001" which is same as the original data as mentioned in the Source Section. VHDL code is used for finding the error bit location, correcting error bit and decrypt the encrypted data[2][4][13].

The block diagram for the destination section is shown in Fig.3. The simulated result for destination end is shown in Fig.4 using Xilinx ISE 12.3 Simulation window.

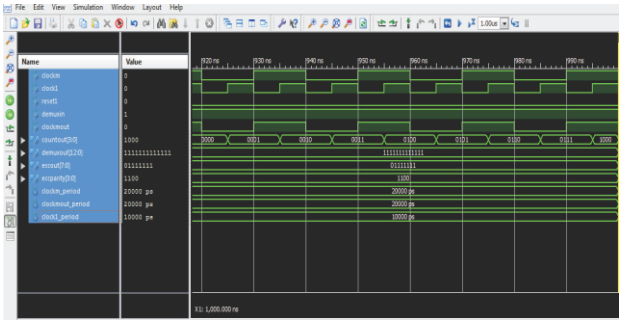

Fig.4 Xilinx ISE 12.3 Output Window for Destination Section

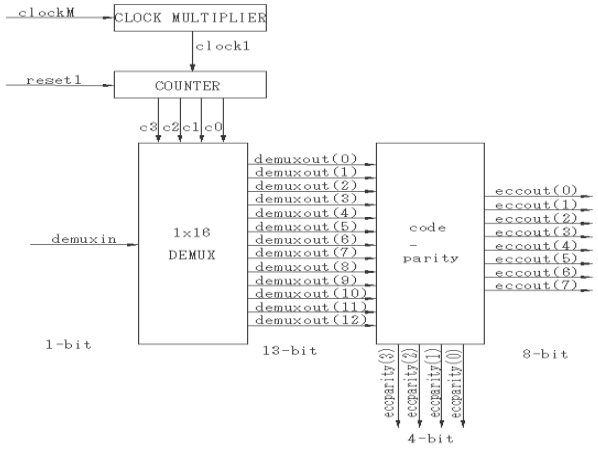

Fig.3 Block Diagram for Destination Section

C. Combined Error Detection and Correction System

The addition of redundancy bits to the original information for minimizing errors is explained in section A. Further explanation for improving speed of data transmission is provided. Section B describes the method to determine the occurrence of error in the received data string and the elimination and the recovery of the original transmitted data

In the present section, the source and receiver section is combined using VHDL code. The block diagram for Error Detection and Correction Section is shown in Fig.5. The Simulated result for Error Detection and Correction Section is shown in Fig. 6 using Xilinx ISE 12.3 Simulation window.

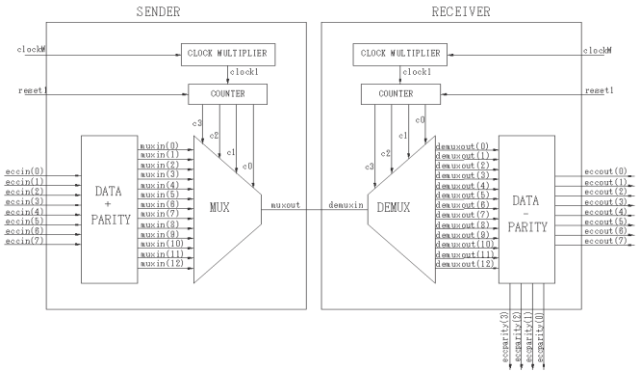

Fig.5 Block Diagram for Error Detection and Correction Section

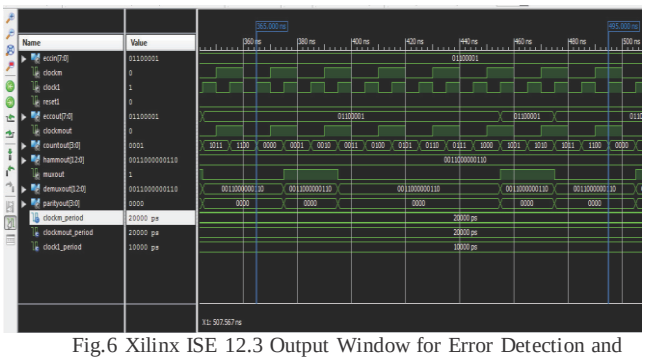

Fig.6 Xilinx ISE 12.3 Output Window for Error Detection and Correction Section

IJERT

IV. CONCLUSION

Error-correction code has been mainly an iterative process. Each code examines the successes and failing of the previous code and builds upon it. These codes have been getting more complex and effective over time. However, surprisingly, the best code we currently have is: Hamming Code. In this work, a method is devised to detect two simultaneous bit errors and correct a single bit in the data message based on Hamming Code. With the implementation of Double data rate transmission scheme in this work it was possible to achieve data transmission in almost half the time compared to normal transmission. The overall conclusion of this paper is that, speed of communication system can be increased by using these methodologies; we can transmit more combination of data to make transmission of data faster.

REFERENCES

[1]A.F.Behrouz,Data communication and networking,2nd ed ,Tata Mc Gra Hill publication2003

[2] J Bhaskar ,A vhdl primer 3rd Ed ,pearson education ,pearson prebtice Hall1998.

[3] S.A.jeevan kumara,Elements of data communication and networking ,1st rd,University science prss.

[4] S.Brown and S Vranesic,Fundamentals of digital logic design with VHDL 2nd EdTMH publication.2007

[5]P .J. Ashenden, *Digital Design - An Embedded Systems Approach Using VHDL*, 1st ed., Morgan Kaufmann Publishers, Aug. 2007.

[6] B. K. Gupta and R. L. Dua, "30 BIT Hamming Code for Error Detection and Correction with Even Parity and Odd Parity Check Method by using VHDL," *International Journal of Computer Applications*, vol. 35,no.13, pp.31-38, Dec. 2011.

[7] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal*, vol.29, pp. 147-160, Apr. 1950.

[8] U. K. Kumar, and B. S. Umashankar, "Improved hamming code for error detection and correction," *2nd International Symposium on Wireless Pervasive Computing*, pp. 498-500, 2007.

[9] R. Ullah, J. Khan, S. Latif, and I. Ullah, "Indication of Efficient Technique for Detection of Check Bits in Hammi ng Code," *International Journal of Computer Science Issues*, vol. 8, iss. 5, no.1, pp. 241-246, Sep. 2011

[10] M.T. Anwar, P. K. Lala, and P. Thenappan, "Decoder Design for a New Single Error Correcting/Double Error Detecting Code," *Proceedings of World Academy Of Science, Engineering and Technology*, vol. 22, pp. 247-251, Jul. 2007.

[11] P. Shankar, "Error Correcting Codes: The Hamming Codes," *Resonance*, pp. 34-43, Jan. 1997.

[12] "Clock Multiplier with Delay Control and Phase Alignment," *Texas Instruments Incorporated*, Austin Texas, 2006. [13] "VHDL," *http://en.wikipedia.org/wiki/VHDL.html*

[14] "Logic and Computer Design Fundamentals," *http://logos.cs.uic.edu/366/notes/ErrorCorrectionAndDetectionSupple ment.pdf*, 3rd ed., 2004.

[15] S.Tam, "Single Error Correction and Double Error Detection," *http://www.xilinx.com/support/documentation/application_notes/xapp6 45.pdf*, Aug. 2006.

[16] "Error Detection and Correction, Module 3: Data Link control," *http://nptel.iitm.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Computer%20networks/pdf/M3L2.pdf*, Version 2 CSE IIT, Kharagpur

IJERT