IJCSC International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012, pp. 1-7

30 BIT HAMMING CODE FOR ERROR DETECTION AND CORRECTION WITH ODD PARITY METHOD BY USING VHDL

Brajesh Kumar Gupta[1], Rajeshwar Lal Dua[2] and B. Surya Narayana Raju[3]

[1]M. Tech Scholar, *E-mail: brajeshgupta1@gmail.com.*

[2]HOD, *E-mail: rndua@yahoo.com.*

[3]Assistant Professor, *E-mail: raju.iitbombay@gmail.com.*

[1,2,3]Department of Electronics and Communication, Jaipur National University, Jaipur.

ABSTRACT

In communication system, a secure data transmission from transmitter to receiver is very major issue for error free transmission there are number of technologies. One of them is hamming code method. Hamming code works on the parity check method. Parities are two types first even parity and second odd parity here we use odd parity method to encrypt data before transmission.

Here we design 30 bit data for transmission by transmitter. In 30 bit data string 25 bits are input data and 5 bits are redundancy bit.

In this paper we describe how can generate 5 redundancy bit for 25 bit of input data and how we can add these redundancy bits in input data string to make 30 bit data for transmission in transmitter section. Suppose, input data is 25'h17EE656 (1011111101110011001010110), for this input data redundancy bits are 5'h18 (11000) thus transmitted data string is 30'h2FDCE5B0 (10111111011100111001010110110000). Now this data string is used to transmit information from transmitting end to receiving end.

All type of communication is possible by the channel between the transmitter and the receiver, this channel may be noisy. Due to this noisy channel transmitted data may be get corrupted and receiver receives this corrupted data. To find that this received data is corrupted or not we again used hamming code error detection and correction technique with odd parity method for finding the error location of corrupted data at receiving end.

In this paper we have used the VHDL language to generate the redundancy bit to add in information input data string before transmission by transmitter. At receiver side we also write the code for finding error data location from received data string. If any error is occurred, we also write the code for correct data bit by correcting the corrupted data bit location.

Here, we have used Xilinx ISE 10.1 Simulator for simulating VHDL Code. Xilinx ISE 10.1 Simulator is a simulator which is used for simulating HDL language and schematic circuit diagram. Here we have used Xilinx simulator to simulate VHDL code for transmitter and receiver.

Keywords: Hamming code, odd parity check method, Redundancy bits, VHDL language, Xilinx ISE 10.1 Simulator.

1. INTRODUCTION

Hamming code error detection and correction with odd parity method by using VHDL is a technique to find the error location and correct that error. In communication system transmission of data from transmitter to receiver safely is very difficult task [1] [4] [5].

In this paper, we use hamming code method for generating redundancy bit and the address of error location. Hamming code methodologies are given below in details.

Here transmitter transmits 30 bit data string. How we can generate this 30 bit data string for transmission, studied in details in transmission section [1] [4] [5].

Here Receiver receives 30 bit encrypted data and checks any error or not? And gives proper 30 bit error free output data. How we can find the error and how can correct it is described in details in receiver section [4][5].

Here VHDL used for generating the code for transmitter and receiver. Xilinx ISE 10.1 Simulator is used to simulate this VHDL code and shows timing diagram waveform [2][3][6][7][8][9][10].

Here we have used some terminologies like input means 25 bit information input data string, data out means 30 bit encrypted output data is transmitted by transmitter in transmission section and at receiver section, input means 30 bit data is received by receiver, erroradd means 5 bit error bit location address, data_out means 30 bit error free data, whose transmitted by transmitter [1] [4] [5].

2. HAMMING CODE METHODOLOGIES

Hamming code was invented by Richard Hamming. Hamming code is an error detection correction code that can we used to detect single and double bit errors and

correct single bit errors that can occur when binary data is transmitted from transmitter into receiver [1] [4] [5].

In 1950 Hamming introduced the (7, 4) code. It encodes 4 data bits into 7 bits by adding three parity bits. Hamming (7, 4) code can detect and correct single bit errors. Hamming code is an improvement on parity check method. It can correct 1 error bit only. Hamming code method works on two methods (even parity, odd parity) for generating redundancy bit [1] [4] [5].

2.1. Redundancy

The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits. The concept of including extra information in the transmission for error detection is a good one. But instead of repeating the entire data stream, a shorter group of bits may be appended to the end of each unit. This technique is called redundancy because the extra bits are redundant to the information [1] [4] [5].

2.2. Even Check Parity Method

In even check parity method count the number of one's at transmitter and receiver side, if number of one's are odd, add one else add zero [1] [4] [5].

2.3. Odd Parity Check Method

In odd parity check method count the number of one's, if number of ones are odd add zero and if number of one's are even add one [1] [4] [5].

In this paper we have used odd check parity method. In Hamming code methodologies at transmission section before transmission of data add some redundancy bit according to odd check parity method. At the receiving section find the address of error bit. If any error occurred in transmitted data we can detect and correct the error with the help of odd parity check method [1] [4] [5].

3. TRANSMISSION SECTION

In this paper we want to communicate with 25 bit information from transmitter. To transmit 25 bit information data we need to add 5 redundancy bit [1] [4] [5].

Calculation of these redundancy bit are given below.

$$2^{r} >= d + r + 1$$

Here d = number of information data.

r = number of redundancy bit.

To transmit 25 bit information data need to add minimum 5 redundancy bits. These redundancy bits are $r(1), r(2), r(4), r(8), r(16)$.

$r(1)$ = 1, 2, 4, 5, 7, 9, 11, 12, 14, 16, 18, 20, 22, 24

$r(2)$ = 1, 3, 4, 6, 7, 10, 11, 13, 14, 17, 18, 21, 22, 25

$r(4)$ = 2, 3, 4, 8, 9, 10, 11, 15, 16, 17, 18, 23, 24, 25

$r(8)$ = 5, 6, 7, 8, 9, 10, 11, 19, 20, 21, 22, 23, 24, 25

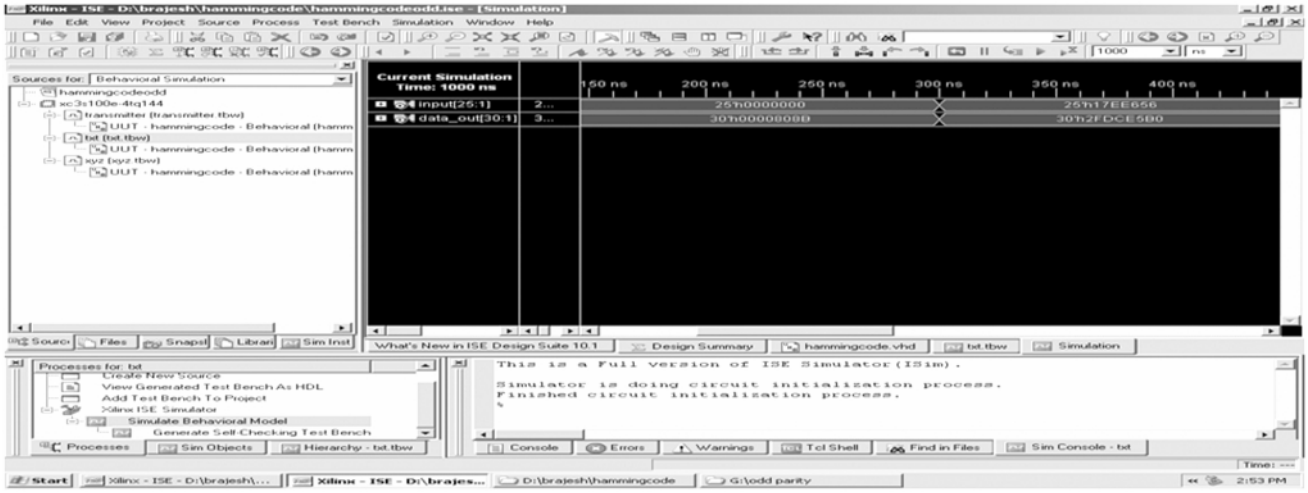
$r(16)$ = 2, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25

In VHDL, to calculate the value of redundancy bits $r(1), r(2), r(4), r(8), r(16)$ we use XOR and NOT gate. After find these redundancy bits value add these bits in information input data and transmit this 30 bit data string by transmitter [2] [3] [8] [9] [10].

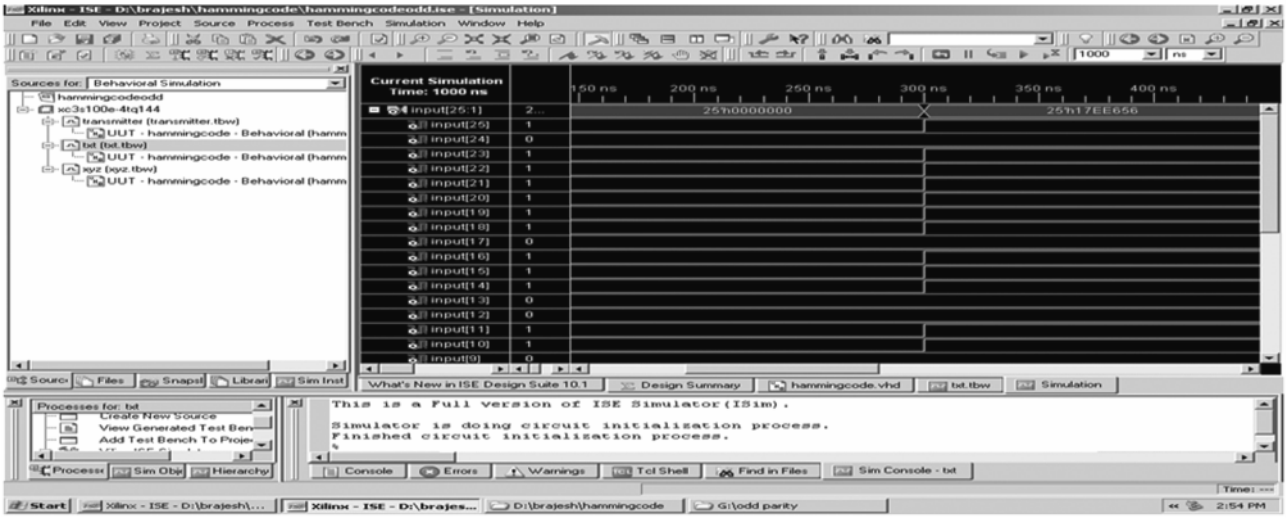
Suppose, we want to transmit information data 25`h17EE656 (10111111011100110010101 10), now write VHDL code for redundancy bits given below. After writing VHDL code simulate it by Xilinx ISE 10.1 simulator and get the value of $r(1), r(2), r(4), r(8), r(16)$. We get $r(1)$ is 0 (zero), $r(2)$ is 0(zero), $r(4)$ is 0(zero), $r(8)$ is 1(one) and $r(16)$ is 1 (one). Now add these redundancy bits to the information data and find encrypted 30 bit data is 30`h2FDCE5B0 (10111111011110010110110000) for transmission. The VHDL code and simulate results are given below in Xilinx ISE 10.1 simulation window [2][3][6][7][8][9][10].

Here input means 25 bit information data and data_out means 30 bit encrypted data for transmission.

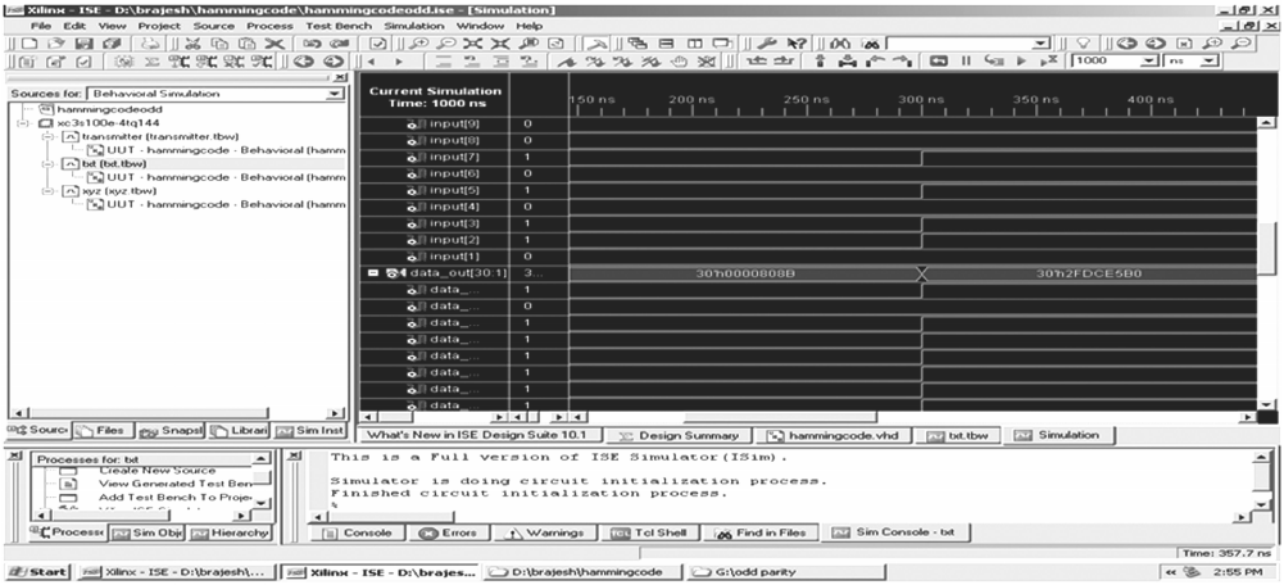
Window for VHDL Code for Transmitter



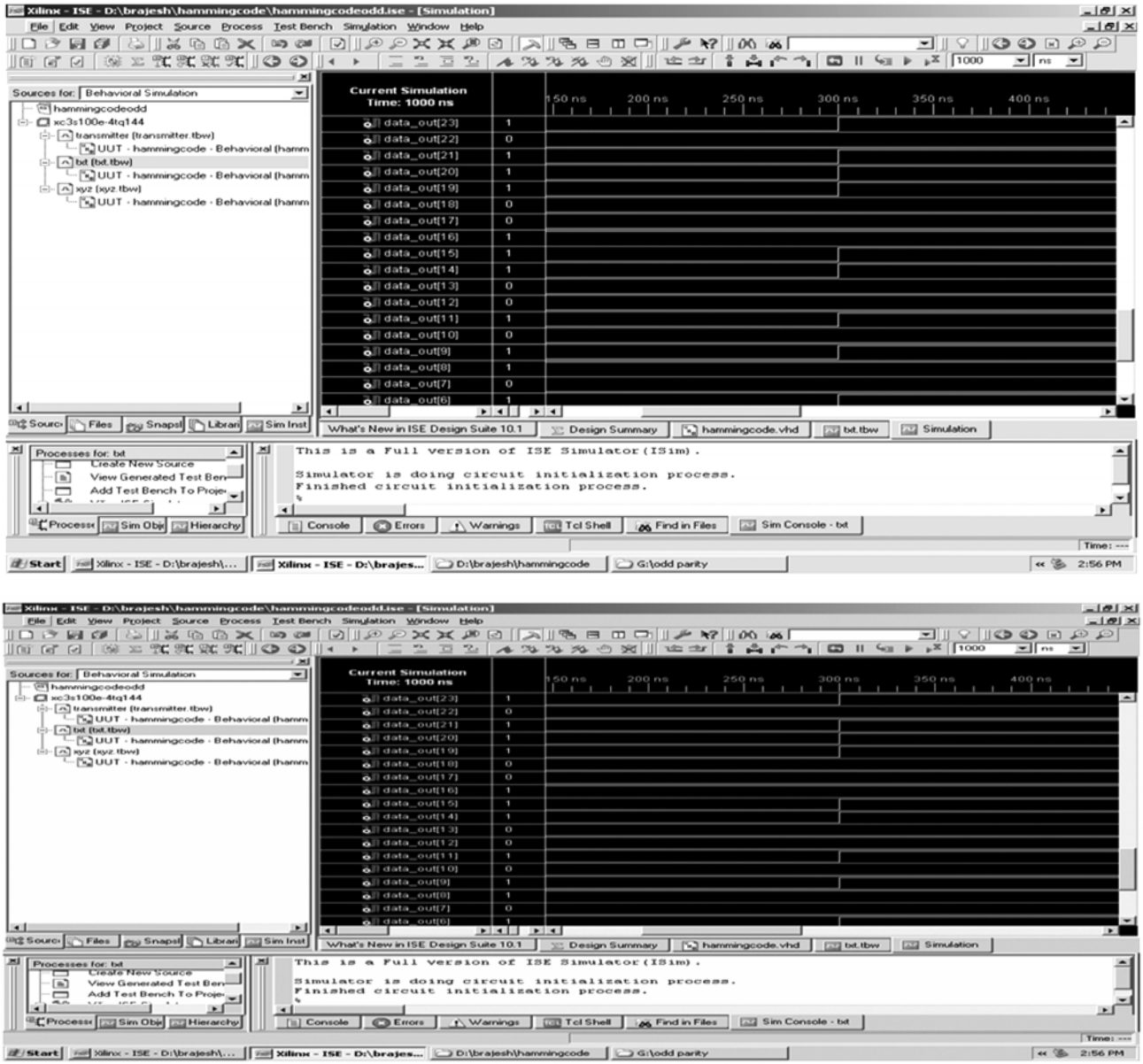
Input Output Wave form in Hexadecimal Format.



Input Wave form in Binary Format.



Remaining Input and Some Output Wave form in Binary Format.



Remaining Output Wave Form.

4. RECEIVING SECTION

In receiver section we receiver's data which is transmitted by transmitter. In between transmitter and receiver communication is possible by some medium called channel. Encrypted data travelled by this channel from transmitter to receiver, this channel may be noisy. Due to this noisy channel received data may be corrupted; to find the location of corrupted bit we use hamming code odd parity method [1] [4] [5].

In this paper we transmit 30 bit encrypted data string from transmitter and receiver receives this data string. Transmitted data string travelled by channel from transmitter to receiver. This channel may be noisy; due to this noisy channel received data string may be corrupted. To find address of corrupted bit we need 5 bits [1] [4] [5].

Suppose the name of error bits are erroradd.

erroradd (1) = 1, 3, 5, 7, 11, 13, 15, 17, 19, 21, 23, 27, 29

erroradd (2) = 2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 23, 26, 27, 30

erroradd (3) = 4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30

erroradd (4) = 8, 9, 10, 11, 12, 13, 14, 15, 24, 25, 26, 27, 28, 29, 30

erroradd (5) = 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30

To find the location of error bit, write code in VHDL. In hamming code error detection and correction with odd parity method by using HDL we use XOR and NOT gate for finding the error bit location.

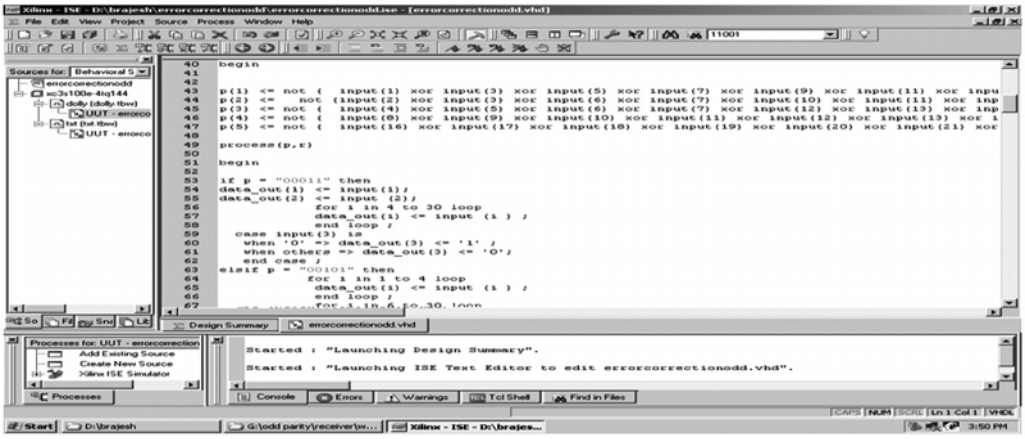
Suppose transmitter transmit data is 30`h2FDC653B (10111111011100011001010011011) and due to noisy channel receiver receives corrupted 30 bit data string is 30`h2EDCE5B0 (10111011011100111001011011000) [2][3][8][9][10].

Now receiver fined location of error bit and corrects them. For finding the error bit location and correcting that error bit we have written code in VHDL [2] [3] [8] [9] [10].

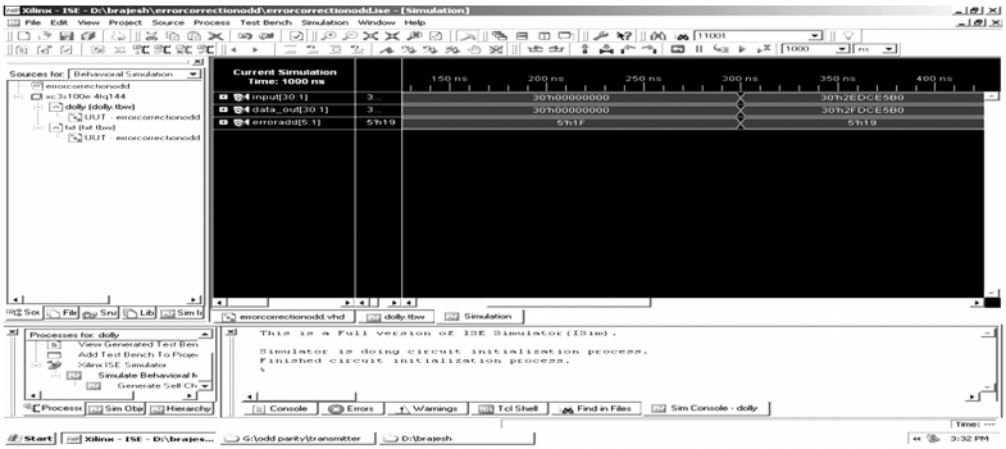
After writing code in VHDL, VHDL code is simulated by Xilinx ISE 10.1 simulator. And get the value of erroradd. The value of erroradd for received data is 5`h19 (11001) here erroradd (1) is 1 (one), erroradd (2) is 0 (zero), erroradd (3) is 0(zero), erroradd (4) is 1 (one) and erroradd (5) is 1 (one). After finding error bit location we replace this error bit zero by one and one by zero[6] [7].

VHDL code and its simulated results are shown in Xilinx ISE 10.1 simulation window[6] [7].

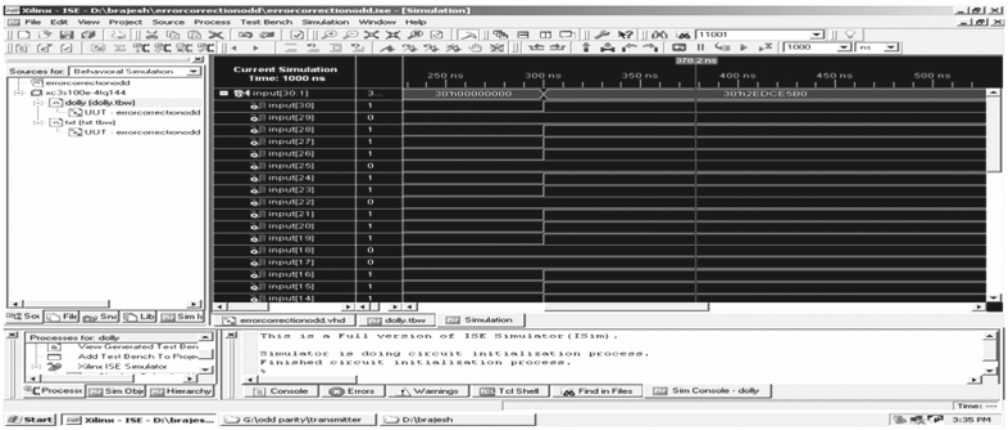
In Xilinx ISE 10.1 simulator, input [30:1] means 30 bit received corrupted data string, data out [30:1] means 30 bit encrypted correct data output string and erroradd [5: 1] means location of error bit [6] [7].



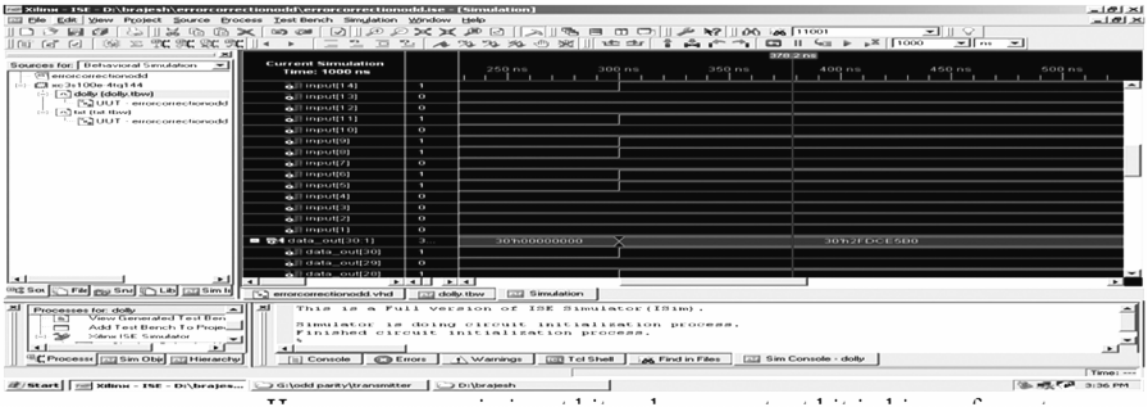
VHDL Code for Receiving Section to Find Error Bit Location and Correct Error Bit.



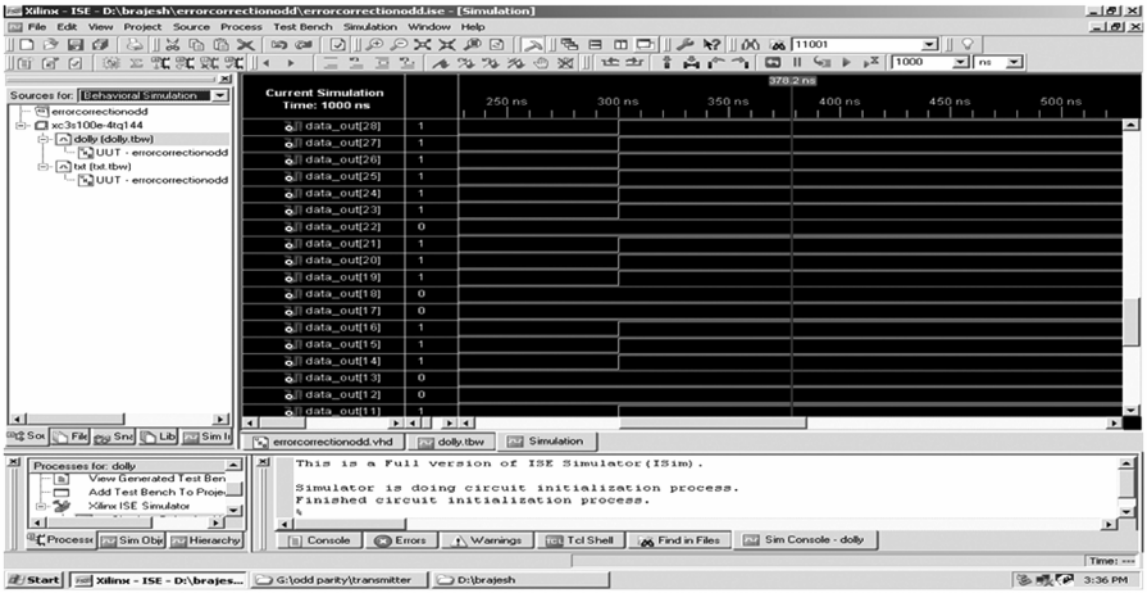
Simulation results in Hexadecimal format, in this window we show 30 bit input received data by receiver, 30 bit output correct data and error bit location.



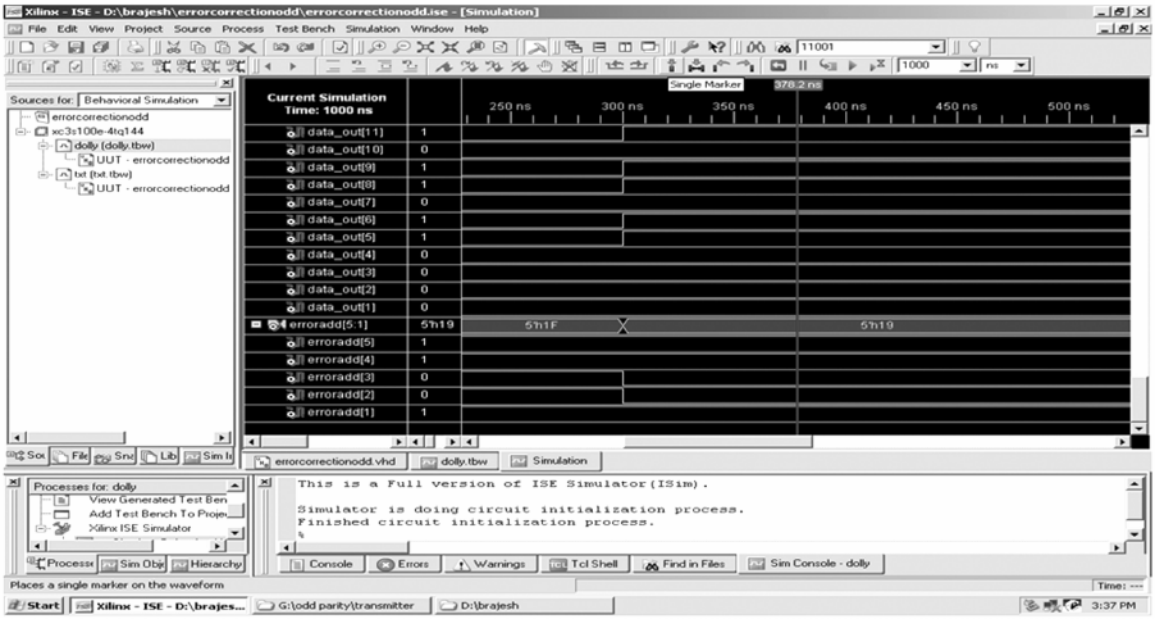
Here We See Input Data in Binary Format from Input Bit 30 to 14.



Here we see remain input bit and some output bit in binary format.



Here we see remaining output bit in binary format.



Here we see error bit location and remaining output data bit in binary format.

5. APPLICATION

An application of hamming code error detection and correction with odd parity check method is that no need to transmit data again by transmitter if only single bit error is occurred by noisy channel because receiver can regenerate the original data string which is transmitted by transmitter. Error detection and correction codes are used in many common systems including: storage devices (CD, DVD, DRAM), mobile communication (cellular telephones, wireless, microwave links), digital television, and high-speed modems (ADSL, xDSL).

6. ADVANTAGES

Speed of communication depends on the number of frame (combination of number of bit is called frame) can be transmitted in a second. To increase the speed of communication system increase the number of frame per second or increase the number of bits in a frame. Here we have increased the frame size to increase the number of bits in a single frame. Up to today we can transmit only 11 bit (7 bit data and 4 redundancy bit) in a frame but now we can transmit 30 bit (25 bit information data with 5 redundancy bit) in a single frame.

7. CONCLUSION

The overall conclusion of this paper is that, speed of communication system can be increased by using these methodologies, and more combination of data (more information in a single frame).

Up to today we can transmit 7 bit information data string means only 2^7 (128) combination of data, means only 128 type of information can be transmit at a time.

Now by using 30 bit hamming code error detection and correction with odd parity method we can transmit 25 bit information data string means 2^25 (33554432) combination of input data. Now we can transmit 33554432 type of information at a time.

REFERENCES

[1] "Data Communication and Networking", Behrouz A. Forouzan, 2nd edition Tata McGrawHill publication.

[2] A VHDL Primer, J. Bhasker, 3rd Edition PHI publication.

[3] Digital Logic Design with VHDL, Stephen Brown & Zvonko Vranesic, 2 nd edition TMH publication.

[4] http://www.pragsoft.com/books/CommNetwork.pdf.

[5] http://www.eng.uwaterloo.ca/~tnaqvi/downloads/DOC/sd192/ISE8_1i_manuals.pdf

[6] http://www.xilinx.com/training/xilinx-training-courses.pdf.

[7] http://www.xilinx.com/itp/xilinx10/books/docs/qst/qst.pdf

[8] http://en.wikipedia.org/wiki/VHDL

[9] http://www.vhdl-online.de/tutorial/

[10] http://www.doulos.com/knowhow/vhdl_designers_guide/