

## **COMMUNICATION BY 31 BIT HAMMING CODE TRANSCEIVER WITH EVEN PARITY AND ODD PARITY CHECK METHOD BY USING VHDL**

**<sup>1</sup>Mr.Brajesh Kumar Gupta <sup>2</sup> Prof. Rajeshwar Lal Dua**

M.Tech scholar

HOD

Department of Electronics & Communication engineering JNU, Jaipur

### **Abstract**

In communication system communication is possible in three modes. They are simplex, half duplex and full duplex mode. Here, we are working on full duplex mode by using the property of transceiver. Transceiver can transmits and receives data simultaneously.

Here we generate 31 bit code to transmit 25 bit information data. And also find 25 bit actual information data from 31 bit received code.

To generate 31 bit data string form 25 bit actual information data for transmission at transmitting end we use Hamming code method. Here we also use Hamming code methodology for finding 25 bit actual information data from received 31 bit data string at receiving end.

To transmit 25 bit actual information data by using Hamming code even parity and odd parity check method we have to add 5 redundancy bits and 1 bit for deciding the type of parity used (even parity and odd parity) in actual data string. After adding these 6 bits in 25 bit information data we get 31 bit data string for transmission at transmitting end.

At receiver section, we find 25 bit actual information data string from 31 bit received data string. To find 25 bit information data from 31 bit received data string we need 5 bit for finding error bit location (if any single bit or double bit error is occurred) and 1 bit is needed for selecting the same parity check method, which we have used at transmitting end.

Here we have written VHDL code for generating 31 bit data string code form 25 bit information data by Hamming code even parity and odd parity check methodology for transmission at transmitting end. Here we also written VHDL code at receiving end for finding 25 bit actual information data from received 31 bit data string code by Hamming code even parity and odd parity check method.

Here we have used Xilinx ISE 10.1 simulator to simulate this VHDL code. Xilinx simulator is a tool which is used for simulation of VHDL, Verilog HDL and schematic circuits.

In this paper we have described, what is communication and their respective mode in detail, in communication section.

In this paper we have also described, what is Hamming code and how it can generate 31 bit data for transmission by 25 bit information data at transmitting end, and how it can find 25 bit actual information from 31 bit received data string at receiving end.

In this paper, we have also described what transceiver is and how it works for communication at transmitting and receiving end.

Till now, from transmitting end we can only transmit data, not receive. And at receiving end we can only receive data but can't transmit.

Now, we can transmit as well as receive data at both ends (transmitting and receiving end) by using transceiver at both ends.

**Key words-** Hamming code, even parity check method, odd parity check method, redundancy bit, transceiver, transmitter, receiver, VHDL, Xilinx ISE 10.1 simulator

### **I. Introduction**

Communication through 31 bit Hamming code transceiver with even and odd parity method is possible in full duplex mode. Here we design transceiver with even and odd parity check method for source and destination using VHDL code. [1][2][15][20].

In this paper we want to communication in full duplex mode with even and odd parity check method with 25 bit information data string. [1][2][15][20].

To transmit 25 bit information data string need minimum 5 redundancy bit and one extra bit for decide parity used for generate redundancy bit.

Here we generate 31 bit data string for transmit 25 bit information data at both section with even and odd parity check method. Suppose at source section we want to transmit '25'h1B7777D' 25 bit information data, to transmit this 25 bit information data we generate 31 bit data string for secure communication with even and odd parity check method. To transmit '25'h1B7777D' 25 bit information data we get '31'h6DDCEFC9 or 31'h6DDDEEDE' 31 bit data string by even parity and odd parity check method respectively. To know how we get this 31 bit data string from 25 bit information data string we must be go through working of transceiver at source section.

Suppose, destination section transmit '31'h7DDDF6CF or 31'h7DDCF7D8' 31 bit data string for '25'h1F777BD' 25 bit information data string with even and odd parity check method . During transmission due to noise, source section's transceiver receives '31'h79DDDF6CF or 31'h7D9CF7D8'. Now source section transceiver finds actual information 25 bit data from corrupted data string. How source section transceiver find actual information data from corrupted data string describe in working of transceiver at source section.

At destination section suppose we want o transmit '25'h1F777BD' 25 bit information data string. To transmit this 25 bit information data string we transmit 31 bit data string to make communication secure. Here we transmit '31'h7DDDF6CF or 31'h7DDCF7D8' 31 bit data string for '25'h1F777BD' 25 bit information with even parity and odd parity check method. To know how we get this 31 bit data string for transmission of 25 bit information data describe in working of transceiver at destination section. [1][2][3][4][14][17][18].

Suppose , source section transceiver transmit '31'h6DDCEFC9 or 31'h6DDDEEDE' 31 bit data string by even parity check and odd parity check method respectively for communication between source and destination. During the transmission this data may be corrupt due to noisy and destination transceiver receives noisy '31'h6FDCEFC9 or 31'h6DDAEDE' 31 bit corrupted data string. Now destination transceiver finds actual 25 bit information data from received corrupted 31 bit data string by even parity or odd parity check method. To know how destination transceiver find 25 bit actual information data string from 31 bit corrupted data string describe in working of transceiver at destination section. . [1][2][3][4][14][17][18].

Here we use evenparity and oddparity out put pin to indicate received data is in even parity or in odd parity .

In this paper we write VHDL code for both source section transceiver and destination transceiver for find actual 25 bit information data string from received 31 bit corrupted data string.

In this paper we also write VHDL code for both section (source section and destination section) to generate 31 bit data string for transmission of 25 bit information data. [1][2][3][9][10][11][12][13]

Here we used Xilinx ISE 10.1 simulator to simulate this VHDL code for both section (source section and destination section). Xilinx ISE 10.1 simulator simulates VHDL code and gives their output in time cycle waveform.[7][8]

## II. Communication

For communication we require minimum two sections, one of them is for transmitting data called source section and another for receiving data called destination section. According to the property of source and destination section, communication is possible in three different modes . They are Simplex mode communication , Half Duplex mode communication and Full Duplex mode communication [1][2][4][15][20].

### ii.i simplex mode communication

In this mode communication is possible only in one way. In this mode source section is capable only for transmitting data string (cannot receive data string) and receiver section is capable only for receiving data string (cannot transmit data string) [1][2][15][20].

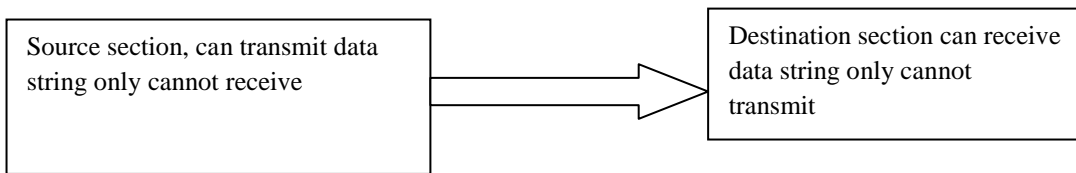


Fig.1 - Communication in simplex mode

### ii.ii half duplex mode communication

In this mode of communication is possible in two ways. In this mode both section can receives data string as well as transmit data string but cannot receives data string when transmitting and vice-versa[1][2][4][15][20].

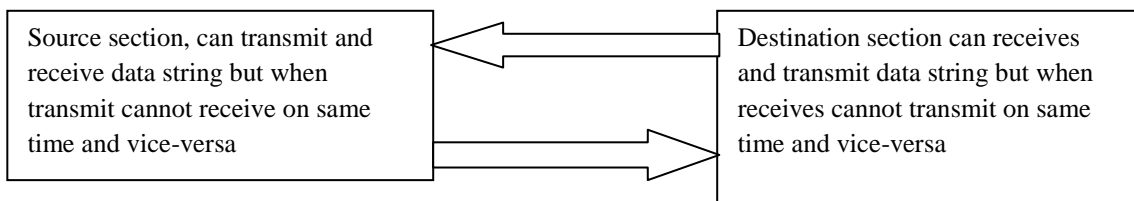


Fig. 2- Communication in Half duplex mode

**ii.iii full duplex mode communication**

In this mode of communication is possible in two ways. In this mode both source and destination section can receive data string as well as transmit data string simultaneously [1][2][4][15][20].

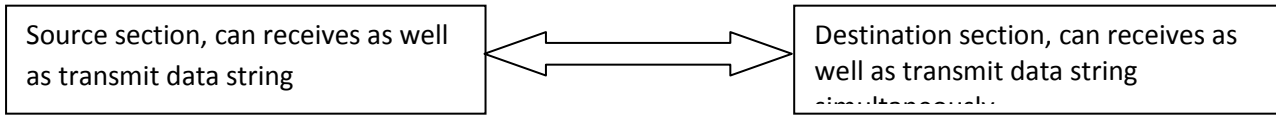


Fig. 3 - Communication in Full duplex mode

**III. Transceiver**

Transceiver is a combination of transmitter section and receiver section. It can transmit data string as well as receive data string simultaneously. By using the transceiver we can communicate in full duplex mode.

Receiver section of transceiver works for finding 25 bit actual data string from received 31 bit data string and then check LSB bit whether it is zero or one. If LSB bit is zero then use odd parity check method else use even parity check method[4][15][19][20].

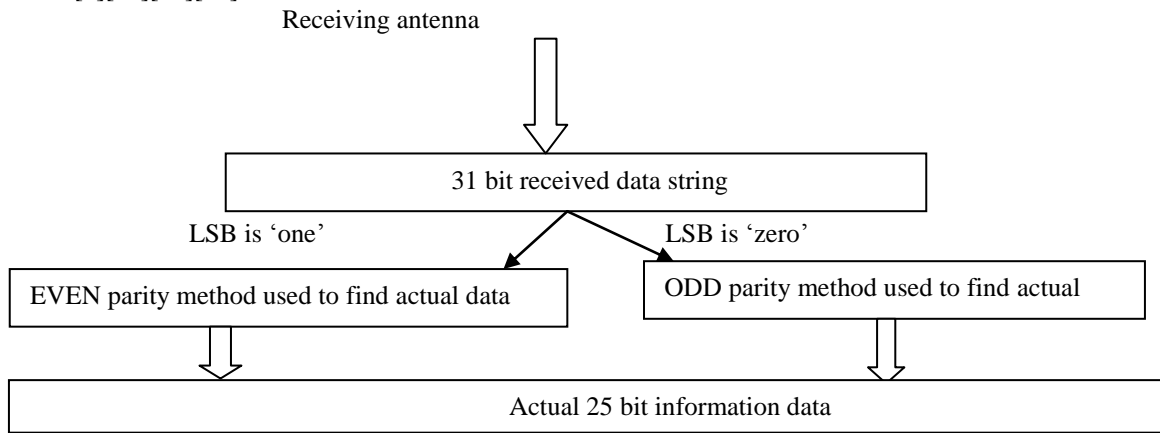


Fig .4 - Working Of Receiver Section of Transceiver

Transmitter section of transceiver generate 31 bit information data string for transmit 25 bit information data string with 5 redundancy bits and 'one' bit for parity decide.

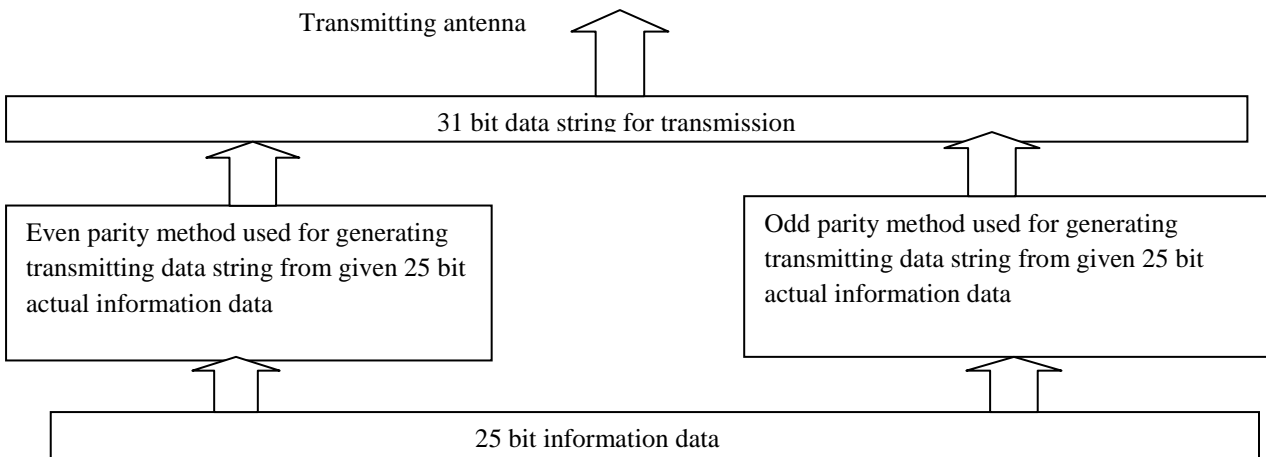


Fig.5 - Transmission section for Transceiver

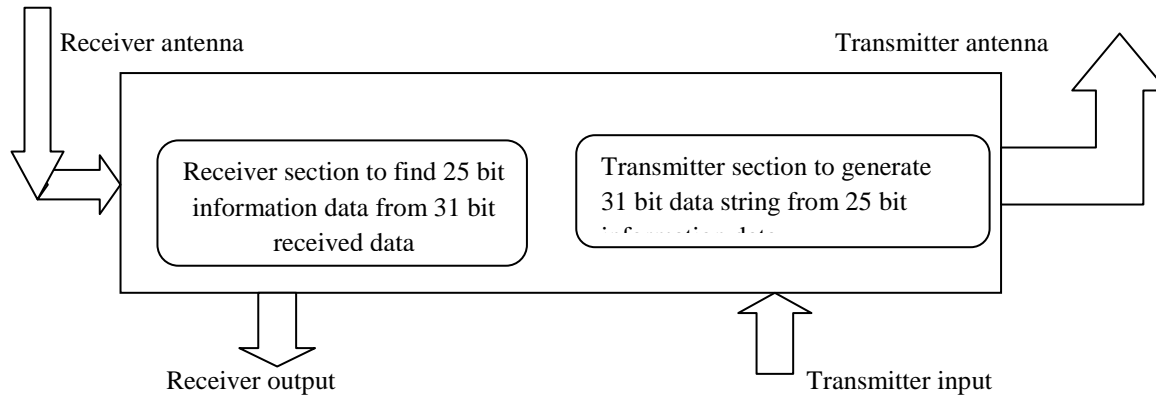


Fig.6 – Transceiver

#### IV. Hamming code

Hamming code is a linear error-correcting code named after its inventor, Richard Hamming. Hamming code can detect maximum two bit error, and correct only single-bit error. Thus, reliable communication is possible when the Hamming distance between the transmitted and received bit pattern is less than or equal to one. While the simple parity code cannot correct errors, it can only detect an odd number of errors.

In 1950 Hamming introduced the (7, 4) code. It encodes 4 data bits into 7 bits by adding three parity bits. Hamming (7, 4) can detect and correct single – bit errors. With the addition of overall parity bit, it can also detect (but not correct) double bit errors. Hamming code is an improvement on parity check method. It can correct 1 bit error [1][2][3][4][14][17][18].

Hamming code method works on only two methods (even parity, odd parity) for generating redundancy bit. In Hamming code method for generating the number of redundancy bit use formula .The number of redundancy depends on the number of information data bits [1][2][3][4][14][17][18].

Formula for generating redundancy bit ----

$$2^r \geq D + r + 1 \text{ ----- (1)}$$

Here r = number of redundancy bit

D = number of information data bit Calculate the number of number of redundancy bit for 25 bit of input data string by above formula We get 5 redundancy bit required.

##### Iv.I 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][2][3][4][14][17][18].

##### Iv.Ii Even Check Parity Method

In even check parity method, count the number of one`s at transmitter and receiver section. if number of one`s are odd, add `one` else add `zero.` [1][2][3][4][14][17][18].

##### Iv.Iii 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][2][3][4][14][17][18].

#### V. Working Of Transceiver At Source Section

At source section of transceiver, transceiver wants to transmit 31 bit encrypted data string to transmit 25 bit information data with 5 redundancy bit and 1 bit for parity check. And it receives 31 bit encrypted data string for finding 25 bit information data which was transmitted by destination section of the transceiver.

Suppose, Source section wants to transmit `25`h1B7777D` information data. To transmit `25`h1B7777D` information data transceiver needs to add 5 redundancy bit and one bit for deciding parity with 25 bit information data to make 31 bit encrypted data string for transmission. [1][2][3][4][14][17][18].

Suppose we want to transmit '25'h1B777D' information data with even parity and odd parity check method. For even parity and odd parity check method transceiver generates 5 redundancy bits are '01000 = 5'h08 or 10111 = 5'h17' respectively. [1][2][3][4][14][17][18].

Now transceiver transmits '31'h6DDCEFC9 = 110110111011100111011111001001 or 31'h6DDDEEDE = 110110111011101110111011011110' 31 bit data string for 25'h1B777D = 110110111011101110111101, 25 bit information data for even parity and odd parity check method respectively.

To know how we can get this 31 bit data string and 5 redundancy bit for 25 bit information data we must go through the section of working of transmitter of transceiver at source section. [1][2][3][4][14][17][18].

Suppose, Destination section transmits '31'h7DDDF6CF = 111110111011101111011011001111 and 31'h7DDCF7D8 = 111110111011100111101111011000' 31 data string for '25'h1F777BD = 111110111011101110111101' bit information data with even parity check and odd parity check method respectively. Now transmitter of destination section transmits 31 bit data string travel through channel from destination to source section. [1][2][3][4][14][17][18].

Suppose, this channel is noisy due to this noisy channel source section transceiver receives corrupted 31 bit data string. After receiving this corrupted 31 bit data string, source section transceiver find the corrupted data bit location and correct that error bit and find correct 31 bit data string whose transmit by destination section. After finding correct 31 bit data string transceiver pass actual 25 bit information data. To know how transceiver find actual 25 bit information data string from 31 bit corrupted data string go through working of receiver of transceiver at source section transceiver.

### V.I Working Of Transmitter Of Transceiver At Source Section

Transmitter of transceiver at source section generates 31 bit data string for 25 bit information data.

In this paper we want to transmit 25 bit information data with even and odd parity check method. To transmit 25 bit information data we need minimum 5 redundancy bit according to equation 1.

Suppose these redundancy bits are  $r(1), r(2), r(4), r(8), r(16)$ . To calculate the redundancy bit, we count the number of ones in appropriate address of information data bit according to given below...

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

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

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

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

$r(16) = 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25$

Suppose, we want to transmit '25'h1B777D = 110110111011101110111101' 25 bit information data with even and odd parity check method. To transmit this 25 bit information data we must need to add 5 redundancy bit. Calculation of redundancy bit depends on the parity check method (which one we use). For even parity method the value of these 5 redundancy bits are '5'h08 = 01000' and for odd parity method the value of redundancy bits are '5'h17 = 10111'. Now we add these redundancy bit with 25 bit information data and get 30 bit data string according to parity check method. [1][2][3][4][14][17][18].

Before transmission of this data string we must add decided parity bit "stransmitterevenodd" bit to this 30 bit data string to make 31 bit data string for transmission. If stransmitterevenodd bit is one it indicates that we have used even parity check method else we have used odd parity check method to generate 31 bit data string for transmission. Now we have 31 bit data string for transmit of 25 bit information data with even and odd parity check method. [1][2][3][4][14][17][18].

Here "stransmitterinput" indicates 25 bit information data string, which we want to transmit by transceiver at Source section.

The block diagram of transmitter of transceiver shown below..

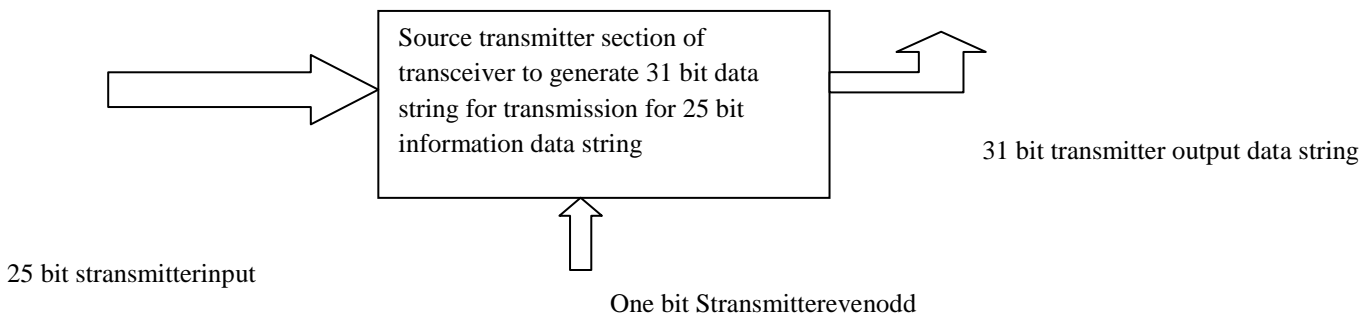


Fig. 7 Block diagram of transmitter of transceiver at source section

## V.II Working Of Receiver Of Transceiver At Source Section

At receiver section of transceiver, transceiver receives 31 bit data string which was transmitted by source section. After receiving this 31 bit data string receiver checks that is there any error in data string? If any error is occurred receiver finds the error bit location and corrects that error bit. [1][2][3][4][14][17][18].

Suppose, at source section receiver receives '31'h79DDF6CF = 11110011101110111110110110011111 or 31'h7D9CF7D8 = 1111101100111001111011110111011000' 31 bit data string in place of '31'h7DDDF6CF = 1111101110111011110110110011111 or 31'h7DDCF7D8 = 1111101110111001111011110111011000' 31 bit data string due to noisy channel. Now to find the error bit location we need minimum 5 error address bit. [1][2][3][4][14][17][18].

Suppose the name of error address bit is "serroraddress". To find the value of "serroraddress" bit we use formula given below. [1][2][3][4][14][17][18].

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

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

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

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

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

Before calculation of 'serroraddress' bits, we must know about the received data is generated in even parity method or odd parity check method at destination section. To know about 31 bit received data is in even parity method or odd parity check method, first check the LSB bit of received data. If LSB bit is 'one' means it generated in even parity check method else is generated in odd parity method. [1][2][3][4][14][17][18].

Suppose, the name of 31 bit input pins are "sreceiverinput". At receiver for '31'h79DDF6CF or 31'h7D9CF7D8' 31 bit data string the value of sreceiverinput(0) is 'one' and 'zero' respectively. Sreceiverinput(0) bit indicate that '31'h79DDF6CF' 31 bit data string is designed by even parity check method and '31'h7D9CF7D8' 31 bit data string is designed by odd parity check method at destination section.

After finding the parity, we find the value of "serroraddress" bits are '5'h1A = 11010 or 5'h16 = 10110' for '31'h79DDF6CF or 31'h7D9CF7D8' received 31 bit data string by even and odd parity check method. Now receiver find correct 31 bit data string is '31'h7DDDF6CF or 31'h7DDCF7D8' whose transmitted by destination section. After correcting the corrupted received 31 bit data string, receiver of source section regenerates the 25'h1F777BD' actual 25 bit information data string which was transmitted by the destination section. [1][2][3][4][14][17][18].

The block diagram of receiver section shown below...

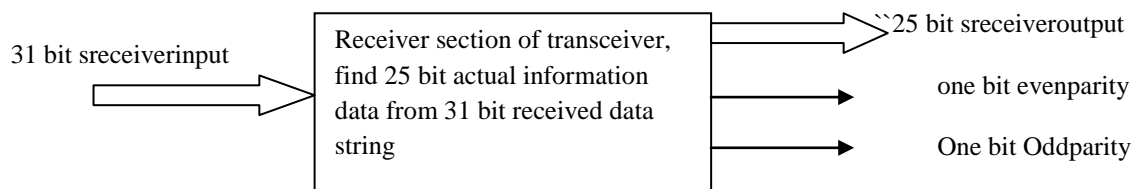


Fig. 8 –Block diagram of receiver of transceiver at source section

In this paper we have written VHDL code for transmitter and receiver for transceiver of source section. At transmitter section we have written VHDL code for generate 5 redundancy bit for 25 bit information data string to make 31 bit data string for transmission. [1][2][3][9][10][11][12][13]

At receiver section we also written VHDL code for finding error bit location and correcting that error bit by replacing 'zero' by 'one' and 'one' by 'zero'. We have written VHDL code for regenerate 25 bit information data string from 31 bit correct data string. [1][2][3][9][10][11][12][13]

In this paper, we use Xilinx ISE 10.1 simulator to simulate this VHDL code for transmitter and receiver section of transceiver. The VHDL code for transceiver shown in Xilinx ISE 10.1 project navigator window and input output wave form for transceiver at source section shown in Xilinx ISE 10.1 simulation window. [7][8]

Xilinx ISE 10.1 project navigator and Xilinx ISE 10.1 simulation window shown below.



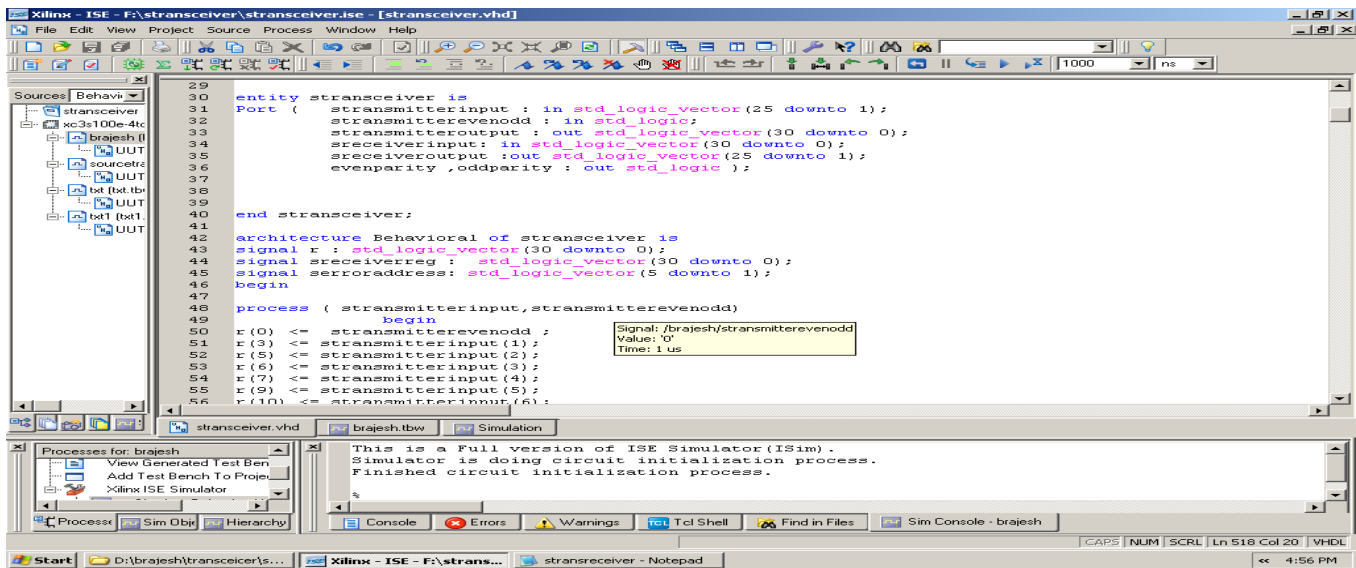


Fig .9 - Xilinx ISE 10.1 project navigator window shows VHDL code for source section

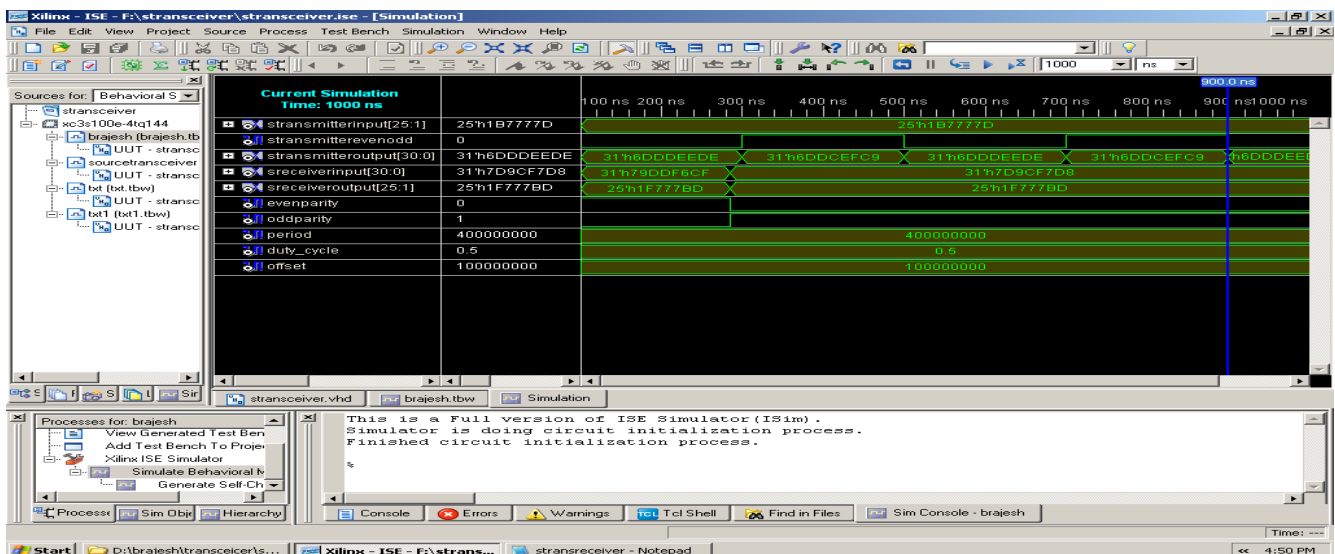


Fig .10 - Xilinx ISE 10.1 simulation window display input output waveform for source section

## VI. Working Of Transceiver At Destination Section

At destination section transceiver want to transmit '25'h1F777BD = 11110111011101110111101' 25 bit information data string. To transmit this 25 bit information data need minimum 5 redundancy bits and 'one' parity decide bit to make 31 bit data string for transmission by transceiver. [1][2][3]

In this paper, If parity decide bit is 'one' means we use even parity method to generate 5 redundancy bit else we use odd parity check method to generate 5 redundancy bit. [1][2][3][4][14][17][18].

Suppose the name of 25 bit information data is 'dtransmitterinput', one bit parity decides bit is 'dtransmitterevenodd' and 31 bit transmitted data is 'dtransmitteroutput'. [1][2][3][4][14][17][18].

In this paper at destination section we want to transmit "25'h1F777BD" 25 bit information data; to transmit 25'h1F777BD information data first we check the value of "dtransmitterevenodd" bit. If "dtransmitterevenodd" bit is 'one' use even parity method to calculate the value of redundancy bits else we use odd parity method to calculate the value of redundancy bits. [1][2][3][4][14][17][18].

The value of redundancy bits are '5'h13 = 10011 or 5'h0C = 01100' for 25'h1F777BD information data string by even parity check method or odd parity check method respectively.

Now to generate 31 bit encrypted data string for transmission, we use 25 bit information data , 5 redundancy bits and 'one' parity decide bit. By using these bits we get '31'h7DDDF6CF or 31'h7DDCF7D8' 31 bit encrypted data string for transmission by even parity check method or odd parity check method respectively.

The generation method of 31 bit data transmission for 25 bit information data at destination is same as transmitter of transceiver at source section. [1][2][3][4][14][17][18].

At destination section, transceiver receives 31 bit data string, and find is there any error or not? If any error is occurred transceiver find this error bit location and correct that error bit by replacing 'zero' by 'one' and one by zero. After correcting error bit transceiver regenerate actual 31 bit data string who's transmitted by source section transceiver. Now transceiver find actual 25 bit information data string from corrected 31 bit data string.

Suppose , source section want transmit '25'h1B7777D = 1110101110111011101111101 ' 25 bit information data by encrypted '31'h6DDCEFC9 = 110110111011100111011111001001 or 31'h6DDDEEDE = 110110111011101110111011011110' 31 bit data string is generated by even parity check or odd parity check method. Now transceiver of destination section receives '31'h6FDCEFC9 = 11011111011100111011111001001 or 31'h6DDAEDE = 110110111011010111011011110' 31 bit corrupted data string due to noise. [1][2][3][4][14][17][18].

After receiving of '31'h6FDCEFC9 or 31'h6DDAEDE' 31 bit corrupted data string , receiver section of transceiver at destination check is there any error is occurred or not ? If any error is occurred receiver find that error bit location. [1][2][3][4][14][17][18].

To find the error bit location we have written VHDL code for generate "derroraddress" bits( name of error bit address) by even parity check or odd parity check method. [1][2][3][4][14][17][18].

The calculation of finding "derroraddress" bits is same as the "serroraddress" bits at source section. By using this method we find the value of "derroraddress" is '5'h19 = 11001 or 5'h0E = 01110' by even parity check or odd parity check method respectively.

Now transceiver of destination section knows that the location of error bit for '31'h6FDCEFC9 or 31'h6DDAEDE' 31 bit received corrupted data string. After finding the address of error bit location transceiver correct this error bit by replacing one by zero or zero by one and get actual 31 bit data string is '31'h6DDCEFC9 or 31'h6DDDEEDE' whose transmitted by transceiver of source section.

After correcting error bit, we capable to generate actual 25 bit information data string from corrected 31 bit data string. we get 25'h1B7777D actual 25 bit information data string Form '31'h6DDCEFC9 or 31'h6DDDEEDE' 31 bit data string. [1][2][3][4][14][17][18].

The generation method of 25 bit information data string from received 31 bit corrupted data string is same as generation method 25 bit information data string from received corrupted 31 bit data string at source section.

Here we already described the method for generating actual 25 bit actual information data string from received 31 bit corrupted data string above in working of receiver of transceiver at source section. [1][2][3][4][14][17][18].

Here we use the term dtransmitterinput , dtransmitteroutput , dtransmitterevenodd , dreceiverinput , dreceiverout ,evenparity , oddparity for representing destination 25 bit information data input , destination 31 bit encrypted data string for transmission , destination 1 bit parity decide bit for generation 31 bit encrypted data string , destination 31 bit received corrupted data string , destination 25 bit actual information data string output , received data string is generated in even parity , received data is generated in odd parity method.

At destination section, we have written VHDL code for generating 31 bit encrypted data string form 25 bit information string for transmission. We written VHDL code for finding 25 bit information data string from received 31 bit corrupted data string. [1][2][3][4][14][17][18].

Here we use Xilinx ISE 10.1 simulator for simulate VHDL code of transceiver at destination section. The VHDL code for transceiver at destination section shown in Xilinx ISE 10.1 project navigator window and Input output time wave form for transceiver at destination shown in Xilinx ISE 10.1 simulation window below.[7][8]



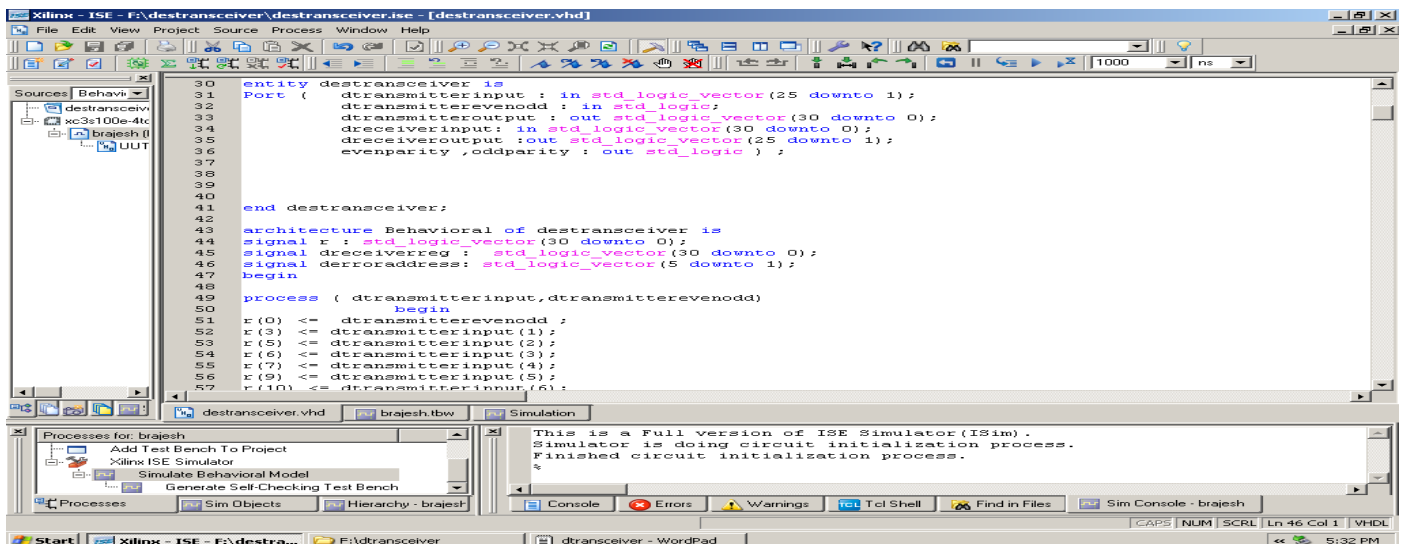


Fig. 11 - Xilinx ISE 10.1 project navigator window display VHDL code for destination section

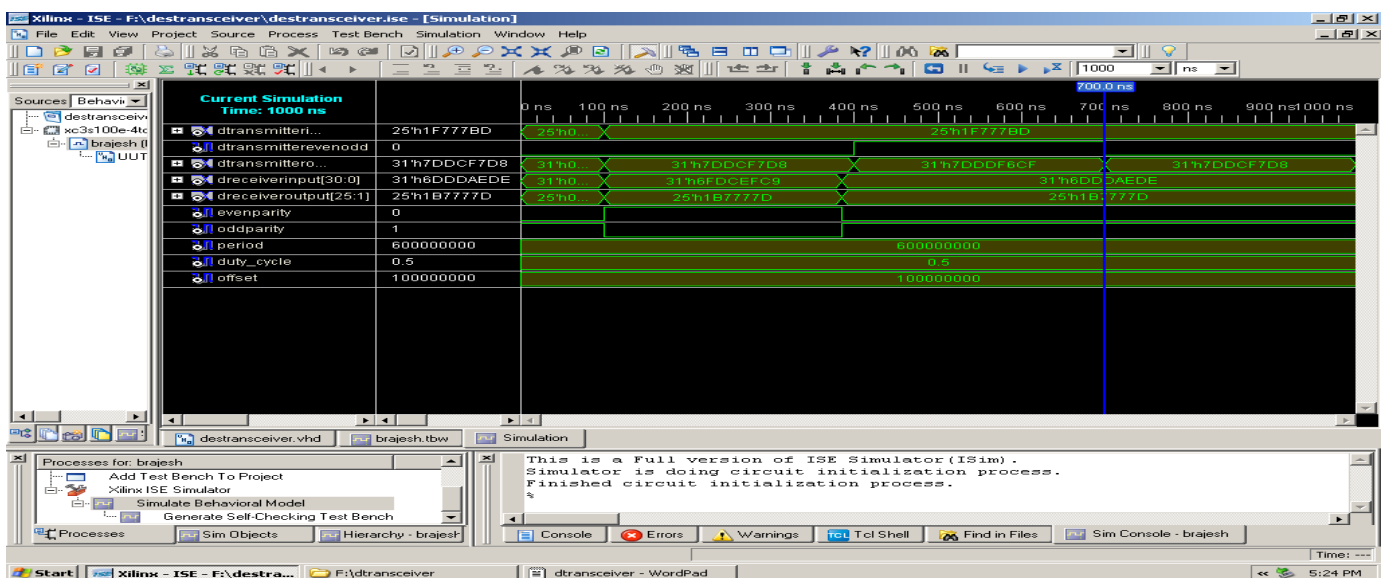


Fig. 12 - Xilinx ISE 10.1 simulation window display simulated results for destination section

## VII. Application

Hamming code methodologies is capable for detecting 2 bit error and correcting single bit error. When we use Hamming code methodology for communication, if single bit error is occurred due to noisy channel no need to retransmit data string again for proper communication because it is able to correct single bit error.

In this paper we design system, to be able to communicate in full duplex mode with 25 bit information data string by even parity and odd parity check method.

The application of this system is that now we can communicate with 25 bit information data string in full duplex mode.

Error detection and correction codes are used in many common systems including: storage devices (CD, DVD, and DRAM), mobile communication (cellular telephones, wireless, and microwave links), digital television, and high-speed modems (ADSL, xDSL).

## VIII. Advantage

Before publication of my research paper, communication is possible by 7 bit information data string only. After publication of my some research papers communication is possible by 25 bit information data string. But till now, communication is possible in simplex mode only by transmitting 30 bit data string with even parity and odd parity check method for 25 bit information data.

Till now, source section is capable only for generating 30 bit data string for transmission and destination is capable for regenerate actual 25 bit information data string from received 30 bit corrupted data string.

In this paper, we use transceiver at source section and destination to make both sections capable for receiving and transmitting data string.

Here we transmit 31 bit data string for transmit 25 bit information data with 5 redundancy bits and '1' extra bit for parity decide.

The advantage of this paper is that now communication is possible in full duplex mode (means both section source and destination can transmit as well as receives data string simultaneously).

Speed of communication system also depends on the number of frame (combination of number of bit is called frame) that can be transmitted in a second. To increase the speed of communication system increases 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 31 bits (25 bit information data with 5 redundancy bit and 'one' extra bit for parity decide) in a single frame.

## **IX. Conclusion**

The overall conclusion is that, now communication is possible in full duplex mode with 25 bit information data string without retransmit data string if any single bit error is occurred.

By using this paper both sections are capable to generate 31 bit data string for transmit 25 bit information data with even parity and odd parity check method. Both sections are also capable to receive 31 bit corrupted data string and finding actual 25 bit information data string from received 31 bit corrupted data string.

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 increases 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 31 bit (25 bit information data with 5 redundancy bit and one parity decide bit) in a single frame.

## **X. Reference –**

- [1]. Brajesh Kumar Gupta, Rajeshwar Lal Dua, "30 bit Hamming code for Error Detection and correction using VHDL" National Journal of Engineering Science And Management (ISSN : 2249-0264) volume number I issue II .
- [2]. Brajesh Kumar Gupta, Rajeshwar Lal Dua, B.Surya Narayana Raju "30 bit Hamming code for Error Detection and correction with Odd parity Method using VHDL" International Journal of Computer Science And Communication (ISSN : 0973-7931) volume III Number I .
- [3]. Brajesh Kumar Gupta, Rajeshwar Lal 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 (ISBN : 978-93-80865-96-2 ) Volume 35 Number 13.
- [4]. Data communication and networking , Behrouz A. Forouzan , 2<sup>nd</sup> edition Tata McGrawHill publication.
- [5]. <http://www.pragsoft.com/books/CommNetwork.pdf>
- [6]. [http://www.eng.uwaterloo.ca/~tnaqvi/downloads/DOC/sd192/ISE8\\_li\\_manuals.pdf](http://www.eng.uwaterloo.ca/~tnaqvi/downloads/DOC/sd192/ISE8_li_manuals.pdf)
- [7]. <http://www.xilinx.com/training/xilinx-training-courses.pdf>
- [8]. <http://www.xilinx.com/itp/xilinx10/books/docs/qst/qst.pdf>
- [9]. <http://en.wikipedia.org/wiki/VHDL>
- [10]. <http://www.vhdl-online.de/tutorial/>
- [11]. [http://www.doulos.com/knowhow/vhdl\\_designers\\_guide/](http://www.doulos.com/knowhow/vhdl_designers_guide/)
- [12]. A VHDL Primer , J. Bhasker , 3<sup>rd</sup> edition PHI publication.
- [13]. Digital Logic Design with VHDL , Stephen Brown & Zvonko Vranesic , 2<sup>nd</sup> edition TMH publication
- [14]. Hamming r.w error detection and correction code, bell sys. Tech. J.29:147-60 1950 bell telephone laboratories ,murray hill
- [15]. <http://www.britannica.com/ebchecked/topic/585799/telecommunication/76275/repetition-codes#ref608200->
- [16]. <http://www.britannica.com/ebchecked/topic/253662/richard-wesley-hamming#ref1073410>
- [17]. <http://www.cs.princeton.edu/courses/archive/spring12/cos126/assignments/hamming.html>
- [18]. logic and computer design fundamental 4<sup>th</sup> edition error detection and correction
- [19]. computer architecture and interfacing by dr. T itagaki -- <http://people.brunel.ac.uk/~eesttti>
- [20]. data and computer communication , chapter 6 digital data communication technique

**Professor Rajeshwar Lal Dua** a Fellow Life Member of IETE and also a Life member of: I.V.S & I.P.A, former “Scientist F” of the Central Electronics Engineering Research Institute (CEERI), Pilani has been one of the most well known scientists in India in the field of Vacuum Electronic Devices for over three and half decades. His professional achievements span a wide area of vacuum microwave devices ranging from crossed-field and linear-beam devices to present-day gyrotrons.



He was awarded a degree of M.Sc (Physics) and M.Sc Tech (Electronics) from BITS Pilani. He started his professional carrier in 1966 at Central Electronics Engineering Research Institute (CEERI), Pilani. During this period he designed and developed a specific high power Magnetron for defence and batch produced about 100 tubes for their use. Trained the Engineers of Industries with know how transfer for further production of the

same.

In 1979 he visited department of Electrical and Electronics Engineering at the University of Sheffield (UK) in the capacity of independent research worker, and Engineering Department of Cambridge University Cambridge (UK) as a visiting scientist. After having an experience of about 38 years in area of research and development in Microwave field with several papers and a patent to his credit. In 2003 retired as scientist from CEERI, PILANI & shifted to aipur and joined the profession of teaching. From last eight years he is working as professor and head of electronics department in various engineering colleges. At present he is working as head and Professor in the department of Electronics and communication engineering at JNU, Jaipur. He has guided several thesis of M.tech .of many Universities.



**Mr. Brajesh Kumar Gupta** , student of Mtech IV sem of Jaipur National University, Jaipur , I have completed my B.E from “R.K.D.F institute of science & Technology “,Bhopal in 2007 in Electronics & Communication Engineering under the university of R.G.P.V Bhopal . I also done VLSI design Advanced post Graduation Diploma in DEC- 2007 from VEDANT , SCL ( semiconductor of laboratory ) Mohali , Chandigarh,it comes under Department of Space government of India (ISRO).