# HAMING CODE FOR SINGLE BIT ERROR DETECTION & ERROR CORRECTION WITH EVEN PARITY USING VHDL

Pranjali Pothare, Prajakta Ambatkar, Payal Patre, Karishma Padole, Shilpa Lende, Ankita Belekar

Assistance Prof. Mrs.Samrudhi Shirgaonkar

Dept.of Electronics and Telecommunication

SRMCEW,Nagpur 08

**Abstract—Single bit error detection and correction can be done through hamming code. Hamming code can be made with either even parity or odd parity check method. The information signal can get corrupted in the process of transmission which can lead to wrong information reception. We have used even parity check method for encrypting the information signal and forming the hamming code. The information signal is first encrypted and then transmitted using some transmission media which may be corrupted in the process. The receiver receives the signal and detects and corrects the signal using even parity check method. 11 bit information signal which is encrypted to form 15 bit hamming code where, 4 bits are the redundant bit which is transmitted. And at the receiver, it is decrypted to get the original 11 bit information signal.**

**Keywords:- Hamming code, Even parity .**

## I.INTRODUCTION

In communication system, a secure data transmission from transmitter to receiver is very major issue. 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 code can detect up to two simultaneous bit error.

In communication system, the information signal is send through transmitter and is transmitted through some random media. In the process of transmission the information gets corrupted, so we get a signal at the receiver which is different than the original information signal. The general idea for achieving error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receivers can use to check consistency of the delivered message, and to recover data determined to be corrupted.

The information 'k bits' is encrypted using even parity check method to form hamming code of n bits. The information signal is added with redundant bits 'r' to form the code this code is then transmitted. The receiver receives the code and corrects it to get the original k bits information signal.

## II.HAMMING CODE

Hamming codes are the codes used in communication for error free transmission of information. Hamming codes can detect up to two and correct up to one bit errors. Hamming code is formed by adding the information and redundant bits i.e, n= k + r. The size of the redundant bit is determined by the formula:

$$2r \geq k+ r+ 1$$

The code is represented by (n, k) code where 'n' is the codeword length and 'k' is datalength. We can use this formula for calculating the number of redundancy bits. A very common hamming code is (7, 4) it contains 4 bit information and 3 bit redundancy. Redundancy is added at the encoder to form the codeword and will be removed at the decoder after getting the correct sequence and the desired information will be the final output.

We calculate the redundant bits according to the information bits and place the bits in the codeword in positions given by the sequence be $2^0, 2^1, 2^3, 2^4, 2^5 \ldots\ldots\ldots 2^{r-1}$, where r is the number of redundancy bits.

The redundant bits can be calculated either by Even parity check method or Odd parity check method. Here in this paper we have used even parity check method for the Hamming Encoder and Decoder.

**<u>Even Parity Check Method :</u>**-The value of redundancy bit can be finding by XORING of different location of information data bit for different redundancy bit. The property of XOR gate is that if number of one`s are even in input its shows the output zero else its shows output one. By using this property we can easily find the number of one`s in a given string are even or odd for a particular redundancy bit.

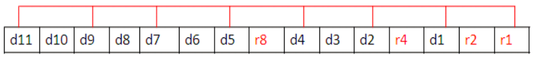
## III.CALCULATION OF REDUNDANCY AT ENCODER WITH 11BIT INFORMATION SIGNAL

We have used even parity check method for calculating redundancy. The value of redundancy bit can be finding by XORING of different location of information data bit for different redundancy bit. the property of XOR gate is that if number of one`s are even in input its shows the output zero else its shows output one. Let the number information bits is k and the corresponding redundancy is r and the codeword length n= k + r, So the bit positions of the redundancies in codeword will be $2^0, 2^1, 2^3, 2^4, 2^5 \ldots\ldots\ldots 2^{r-1}$. The values of the redundancies can be given by x-or operation of the information bits present at the positions alternately in codeword for r1 and taking two consecutive leaving next two and so for r2. Similarly we take 2t bits and left next 2t for r(t). Here for a test purpose we have taken 11 bit information signal and 4 bit redundancy is required to form a 15 bit codeword. And we have calculated the redundancy by even parity check method and will be at the $1^{st}, 2^{nd}, 4^{th}, 8^{th}$ position bit of the codewords.

Bit position

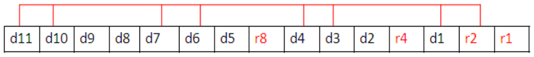
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| d11 | d10 | d9 | d8 | d7 | d6 | d5 | r8 | d4 | d3 | d2 | r4 | d1 | r2 | r1 |

<u>Adding r1</u>:-



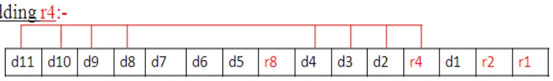
r1= (d1 xor d2 xor d4 xor d5 xor d7 xor d9 xor d11)

<u>Adding r2</u> :-



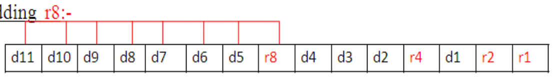
r2=(d1 xor d3 xor d4 xor d6 xor d7 xor d10 xor d11)

<u>Adding r4</u>:-



r4= (d2 xor d3 xor d4 xor d8 xor d8 xor d9 xor d11)

<u>Adding r8</u>:-



r8= (d5 xor d6 xor d7 xor d8 xor d9 xor d10 xor d11)

Let the 11 bit information signal be "**10011100101**", so the value of redundancies become,

**r1:** d1 x-or d2 x-or d4 x-or d5 x-or d7 x-or d9 x-or d11

**r2:** d1 x-or d3 x-or d4 x-or d6 x-or d7 x-or d10 x-or d11

**r4:** d2 x-or d3 x-or d4 x-or d8 x-or d9 x-or d10 x-or d11

**r8:** d5 x-or d6 x-or d7 x-or d8 x-or d9 x-or d10 x-or d11

The value of r1 is 1(the number of one`s are 5)
The value of r2 is 0(the number of one`s are 4)
The value of r4 is 1(the number of one`s are 3)
The value of r8 is 0(the number of one`s are 4)

Therefore the final output sequence of the encoder becomes "101001101001101".

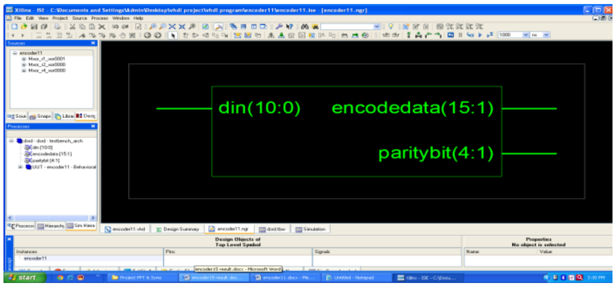
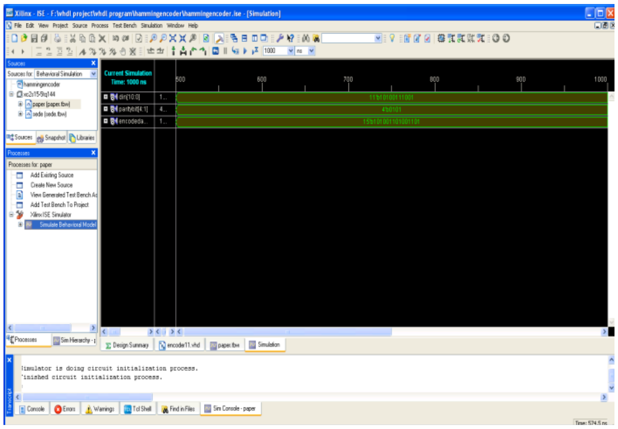


Fig1:Fig shows the black box of encoder.



```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Fig2: Simulation window shows waveform of hamming encoder in binary format.

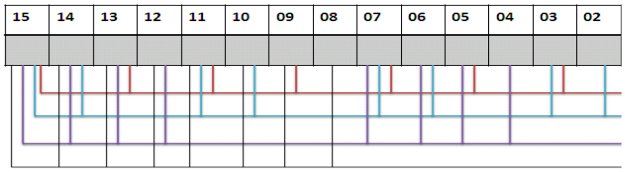
## CALCULATION FOR ADDING ERROR IN NOISE MIXER

Here, in this paper we also design error injector for the 15bit information signal to complete the whole process.In the system, when the transmitted signal of 15bit is received as a input by the noise mixer, it get xor with another input of 15bit which is given to the noise mixer.

For example, the encoded signal “101001101001101” is XOR with another input signal “000001000000000” to the noise injector, we get the corrupted output from mixer i.e.101000101001101. This corrupted 15bit is given input to the decoder i.e. the last stage of input.

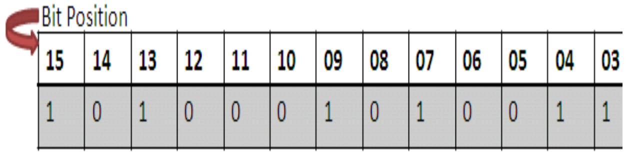
## CALCULATION FOR 11 BIT ORIGINAL SIGNAL OBTAIN AT DECODER

In the decoder, the original 11bit information signal is decoded from 15bit received signal. First, the decoder find out the redundancy values and from these 4 redundancy values the position of corrupted bit is obtained. Decoder invert that bit and remove the 4 bit redundancy and then give the output of 11bit original information signal.



r1 = 1 xor 3 xor 5 xor 7 xor 9 xor 11 xor 13 xor 15
r2 = 2 xor 3 xor 6 xor 7 xor 10 xor 11 xor 14 xor 15
r4 = 4 xor 5 xor 6 xor 7 xor 12 xor 13 xor 14 xor 15
r8 = 8 xor 9 xor 10 xor 11 xor 12 xor 13 xor14 xor 15

As before, we get the 15bit data from noise mixer i.e. “101000101001101”.Now,at the decoder we find out 4bit redundancy and get the error bit position.


Bit Position

| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |

The value of r1 = 0 ( the no. of 1’s are 6)
The value of r2 = 1 ( the no. of 1’s are 3)
The value of r4 = 0 ( the no. of 1’s are 4)
The value of r8 = 1 (the no. of 1’s are 3)

Therefore, the value “1010” shows the binary bit position of corrupted bit i.e $\mathbf{10^{th}}$ position. Now invert that bit means ‘0’ invert into ‘1’ and we get “101001101001101”.Now remove the redundancy and the corrected codeward is obtained of 11bit i.e. **“10011100101”**

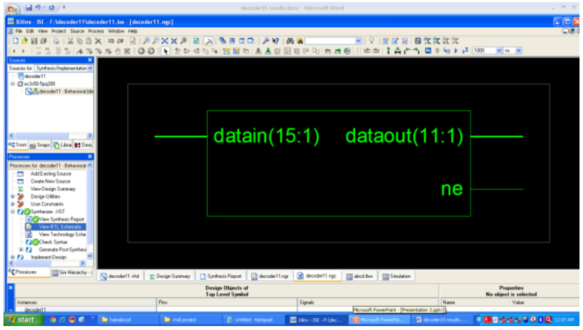
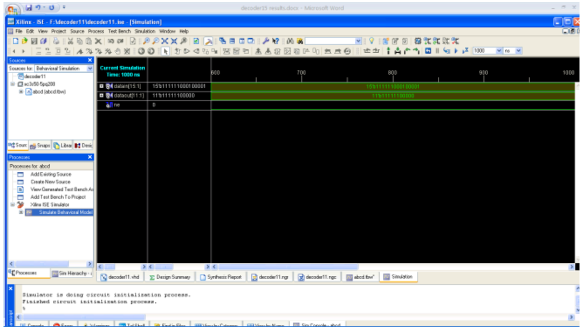


Fig3: figure shows the black box of docoder.



```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Fig4: Simulation window shows the waveform of hamming decoder in binary format.

## HAMMING SYSTEM

We have design the encoder, decoder & noise mixer individually. Encoder having 11 bit input & after adding redundancy bit we get 15 bit encoded data. The encoded data is transfer into the noise mixer, where with the help of PN sequence xoring of the input 15 bit encoded data is done and we get the corrupted 15 bit data. Which then transfer into the decoder block where we detect and correct the position of error and thus we get finally 11 bit original data.

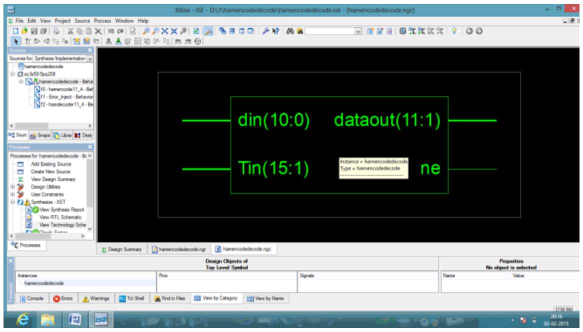
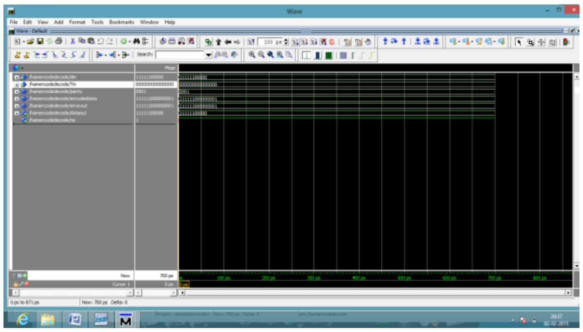
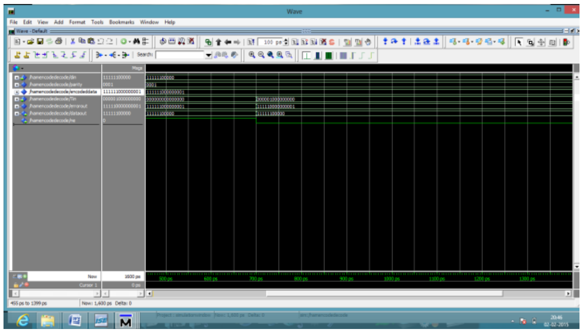


Fig5: Fig shows the black box of hammencodedecode.



```
/hamencodedecode/din          11111100000
/hamencodedecode/Tin          0000000000000000
/hamencodedecode/parity       0001
/hamencodedecode/encodeddata  1111110000000001
/hamencodedecode/errorout     1111110000000001
/hamencodedecode/dataout      11111100000
/hamencodedecode/he           1
```

Fig6: Simulation window shows the waveform of hammencodedecode.



```
/hamencodedecode/din          11111100000
/hamencodedecode/parity       0001
/hamencodedecode/encodeddata  1111110000000001
/hamencodedecode/Tin          0000010000000000
/hamencodedecode/errorout     1111100000000001
/hamencodedecode/dataout      11111100000
/hamencodedecode/he           0
```

Fig7: Simulation window shows the waveform of hammencodedecode with error.

## ADVANTAGES & APPLICATIONS

If we use the correcting codes no resending of data is required as it will automatically detect the error and correct it. We can use transmitter of higher bit capacity no need to break the data into smaller section. So it speeds up the communication. Application of the VHDL is very effective as it is much easier to write the codes. These error correcting techniques are used in many applications as in mobile communication, storage devices, television, satellite communications and all other communications.

## CONCLUSION

We see from this paper how we can use the even parity check method for any sequence of information. And we have shown it for 11 bit information signal. It speeds up the communication as we can encode the total information as a whole and send as one, there is no need for splitting. And using the same parity method at the destination we can successfully recover the original information sequence. The circuitry for this is also reduced as it makes the detection and correction becomes easier and using the VLSI makes it much more efficient.

## REFFERENCE

1] Vivek Singh, Rahul Kumar, Manish Kumar Upadhyay , “VHDL Code For Single Bit Error Detection And Correction With Even Parity Check Method Using Xilinx 9.2i” IJARSE, Vol1, Issue1,2013.

2] Brajesh Kumar Gupta , Prof. Rajeshwar Lal Dua , “30 BIT Hamming Code for Error Detection and Correction with Even Parity and Odd Parity Check Method by using VHDL” published in International Journal of Computer Applications (0975 – 8887),Volume 35– No.13, December 2011.