Error detection and correction using Hamming code

Aditya Iyer – 13BEC038

Vignesh Iyer – 13BEC039

ABSTRACT: In digital communication in order to transmit information in terms of bits, many techniques are used to encode the bits and to detect and correct errors. One of the techniques is the use of Hamming Code. It can at most detect two errors and correct one error. Other techniques include Block code, cyclic codes. In this report, the method of implementation of Hamming Code is explained with an example.

1. INTRODUCTION

A common error detection technique is to add a parity bit to the message being transmitted. Thus, single-bit errors and an odd number of errors can be detected. The error bits cannot be corrected as their location in message is not known.

Thus the hamming code which was developed by Richard W. Hamming developed this code in order to resolve the problem of error correction. The basic Hamming code is capable of detecting single or double errors and can correct a single error. In this method, we add parity bits i.e. error correction bits to our data in order to encode it. Even though the number of bits transmitted increases, the data is protected and this code can correct maximum up to 1-bit. The condition for code correction is given by the equation:

$$2^n \geq 2^k + 2^k \times \sum_{j=1}^{n} \binom{n}{j}$$

This equation is called as the Hamming bound condition, where n→code digits, k→data digits. A single error correcting perfect code is called Hamming Code. In the below derivation we have made the following assumptions:

n=7, k=4, so parity bits are n-k=3. On calculations the value of j=1, for n=7 and k=4. Here digits is the number of bits used.

The basic structure of this type code can be given as follows:

2. ENCODING

Consider the following:

$\mathbf{C= (C_1, C_2, C_3, \dots Cn)_n}$

$\mathbf{d= (d_1, d_2, d_3, \dots, d_k)_k}$

where C→code digits, d→data digits and **n>k**

The code matrix is given by

C=dG

Where G→generator matrix and $\mathbf{G= [I_k \ \ P]}$

I_k→Identity matrix of size k, P→parity bits

$\therefore \mathbf{C= d\,[I_k \ \ P]}$

$\therefore \mathbf{C= [d \ dP]}$

$\therefore \mathbf{C= [d \ C_p]}$, **where** C_p=dP

The above equation is the transmitted bits in form of coded words. Here the data digits are less as compared to code digits due to the addition of parity bits in the code digits and the size of parity bits is n-k.

3. DECODING

I. ERROR DETECTION

Now at the receiver side the transmitted bits are received and they are then decoded with the help of H transpose matrix as shown below:

Here $\mathbf{C= [d \ dP] = [d \ C_p]}$

$$\therefore \boldsymbol{dP} \oplus \boldsymbol{C_p} = \mathbf{0}$$

(Both are same)

$$\therefore [\boldsymbol{d} \quad \boldsymbol{C_p}] \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{I_m} \end{bmatrix} = \mathbf{0}$$

$$\therefore \boldsymbol{C}\begin{bmatrix}\boldsymbol{P}\\ \boldsymbol{I_m}\end{bmatrix} = \mathbf{0}$$

∴CH^T=0, where $\boldsymbol{H^T} = \begin{bmatrix}\boldsymbol{P}\\ \boldsymbol{I_m}\end{bmatrix}$

If any matrix is multiplied with H^T it is always zero and if there is any error the result would be non-zero. The code word on being received by the receiver is checked for error by multiplying the code word with H^T. Also the received word r is given by

$$\boldsymbol{r} = \boldsymbol{C} \oplus \boldsymbol{e}$$

Where C= code matrix, e=error matrix

Consider the following data

$$\boldsymbol{d} = [\mathbf{0\ 1\ 1\ 0}]$$

$$\boldsymbol{C} = [\mathbf{0\ 1\ 1\ 0\ 1\ 1\ 0}]$$

$$\boldsymbol{p} = \begin{bmatrix}\mathbf{1} & \mathbf{1} & \mathbf{1}\\ \mathbf{0} & \mathbf{1} & \mathbf{1}\\ \mathbf{1} & \mathbf{0} & \mathbf{1}\\ \mathbf{1} & \mathbf{1} & \mathbf{0}\end{bmatrix}$$

$$\boldsymbol{G} = \begin{bmatrix}\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1}\\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1}\\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0}\end{bmatrix}$$

II. ERROR CORRECTION

Suppose the received word is **r= [1 1 1 0 1 1 0],** which is an error. So the error detector,

S=r.H^T

$$\therefore \mathbf{S} = (\mathbf{C}\oplus\mathbf{e}).\boldsymbol{H^T}$$

$$\therefore \mathbf{S} = \mathbf{C}.\boldsymbol{H^T}\oplus\boldsymbol{e}.\boldsymbol{H^T}$$

$$\therefore \mathbf{S} = \mathbf{0}\oplus\mathbf{e}.\boldsymbol{H^T}$$

$$\therefore \mathbf{S} = \mathbf{e}.\boldsymbol{H^T}$$

As a result for the above example, the value of S matrix can be given as follows:

$$\boldsymbol{S} = [\mathbf{1}\quad \mathbf{1}\quad \mathbf{1}\quad \mathbf{0}\quad \mathbf{1}\quad \mathbf{1}]\begin{bmatrix}\mathbf{1} & \mathbf{1} & \mathbf{1}\\ \mathbf{0} & \mathbf{1} & \mathbf{1}\\ \mathbf{1} & \mathbf{0} & \mathbf{1}\\ \mathbf{1} & \mathbf{1} & \mathbf{0}\\ \mathbf{1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix}$$

∴S= [1 1 1]

$$\therefore \boldsymbol{e}.\boldsymbol{H^T} = \boldsymbol{e}.\begin{bmatrix}\mathbf{1} & \mathbf{1} & \mathbf{1}\\ \mathbf{0} & \mathbf{1} & \mathbf{1}\\ \mathbf{1} & \mathbf{0} & \mathbf{1}\\ \mathbf{1} & \mathbf{1} & \mathbf{0}\\ \mathbf{1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix}$$

Here e is considered to be a combination of identity matrix as shown below:

$$\boldsymbol{e} = \begin{bmatrix}\boldsymbol{I} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{I}\end{bmatrix}$$

As a result X=e.H^T, so the value of S is given by

$$\boldsymbol{X} = \begin{bmatrix}\mathbf{1} & \mathbf{1} & \mathbf{1}\\ \mathbf{0} & \mathbf{1} & \mathbf{1}\\ \mathbf{1} & \mathbf{0} & \mathbf{1}\\ \mathbf{1} & \mathbf{1} & \mathbf{0}\\ \mathbf{1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix}$$

Now **S= [1 1 1]**, locating the position of S in X, it is in the first row of the matrix of X. As a result error is in the first row, therefore the value of error comes out to be **e= [1 0 0 0 0 0 0]**. So the corrected code matrix is given by

$$\boldsymbol{C} = \boldsymbol{r}\oplus\boldsymbol{e}$$

$$\boldsymbol{C} = [\mathbf{1}\quad \mathbf{1}\quad \mathbf{1}\quad \mathbf{0}\quad \mathbf{1}\quad \mathbf{1}\quad \mathbf{1}]\oplus[\mathbf{1}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}]$$

$$\therefore \boldsymbol{C} = [\mathbf{0}\quad \mathbf{1}\quad \mathbf{1}\quad \mathbf{0}\quad \mathbf{1}\quad \mathbf{1}\quad \mathbf{0}]$$

Thus the value of code word received matches the code word which is transited.

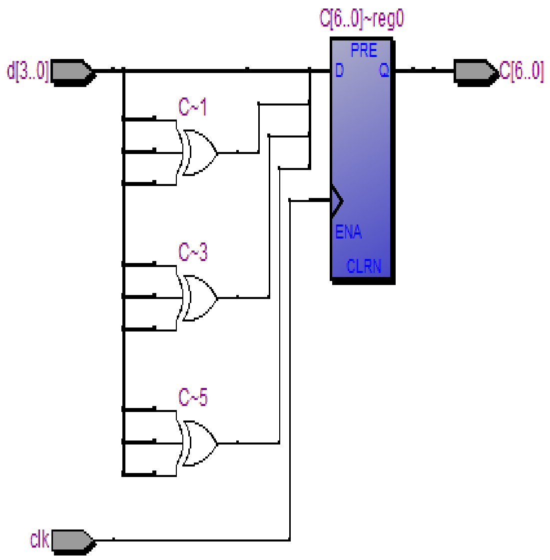
Appendix

A. Hamming Encoder

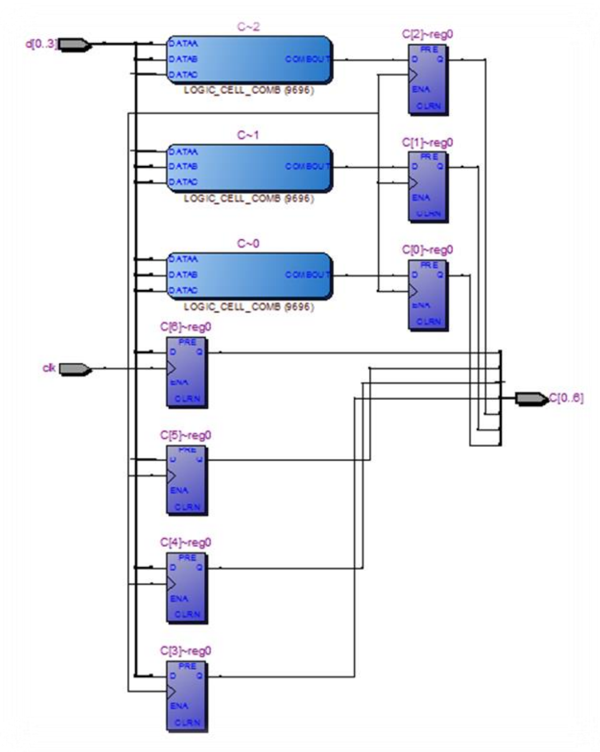
a. The code for hamming encoder can be given as follows:

```
module hammingenc(clk,d,C);
output reg [6:0]C;              //Code digits
input [3:0]d;                   //Data digits
input clk;
always@(posedge clk)
begin
C[6]=d[3];
C[5]=d[2];
C[4]=d[1];
C[3]=d[0];
C[2]=d[0]^d[1]^d[3];            //Parity bits
C[1]=d[0]^d[2]^d[3];            //Parity bits
C[0]=d[1]^d[2]^d[3];            //Parity bits
end
endmodule
```

b. RTL View:



c. TECHNOLOGY View:



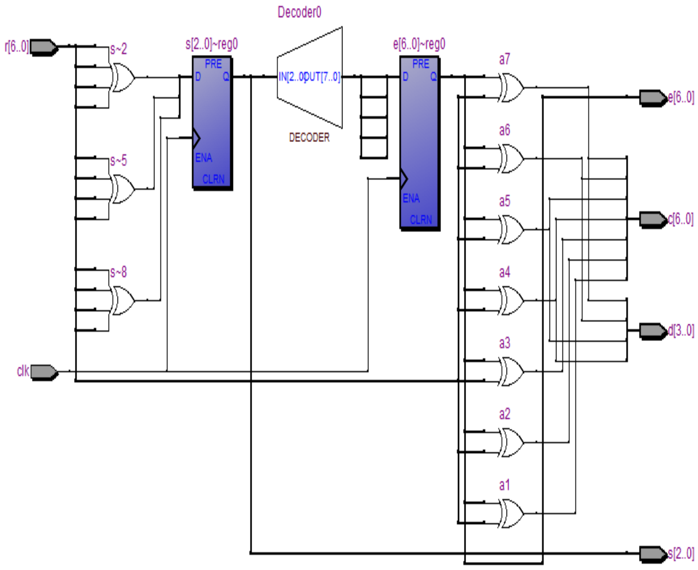
B. HAMMING DECODER

a. The code for hamming decoder for given bits is as follows:

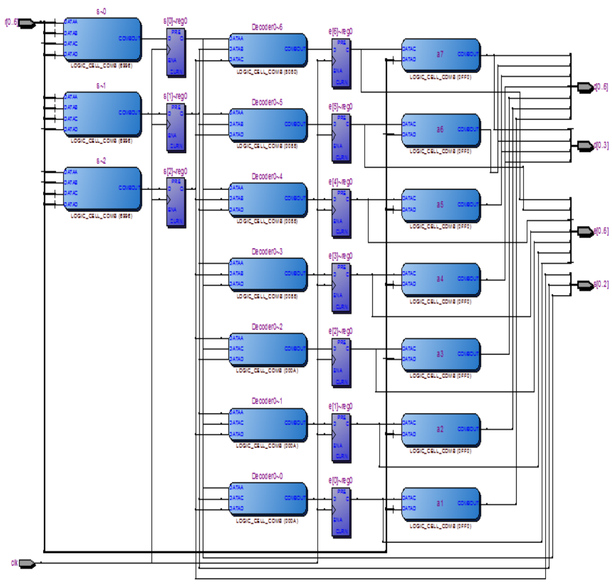
```
module hammingdec(clk,r,s,c,d,e);

input clk;

assign a=7'b1111111; // to be used later to eliminate error

output [6:0]c;

input [6:0]r;  // Port Assignments r--> received sequence c--> error free and actual encoded codeword sequence

output reg [6:0]e;   // error matrix [](1X7)

output reg [3:0]d;   // d--> actual data sequence d=[](1X3) without parity bits, here four

output reg [2:0]s;

always@(posedge clk) // clk--> clock

begin

s[2]=r[6]^r[2]^r[3]^r[4]; // Syndrome Matrix Value Computation S= [](1X3)

s[1]=r[6]^r[1]^r[3]^r[5];

s[0]=r[0]^r[4]^r[5]^r[6];

end

always@(posedge clk)

begin

case(s)                 // Maximum Likelihood Error Detection
```

```
            0 : e= 7'b0000000;        // No error if Syndrome Matrix is a null matrix i.e., [000]
            1 : e= 7'b0000001;
            2 : e= 7'b0000010;
            3 : e= 7'b0100000;
            4 : e= 7'b0000100;
            5 : e= 7'b0010000;
            6 : e= 7'b0001000;
            7 : e= 7'b1000000;
default: e=7'bzzzzzzz;
endcase
end
xor a1(c[0],e[0],r[0]); // xor of error matrix and a 7-bit unity sequence to eliminate errors, actually only one error.
xor a2(c[1],e[1],r[1]);
xor a3(c[2],e[2],r[2]);
xor a4(c[3],e[3],r[3]);
xor a5(c[4],e[4],r[4]);
xor a6(c[5],e[5],r[5]);
xor a7(c[6],e[6],r[6]);
always@(c)
begin
d[0]=c[3];
d[1]=c[4];  // [3:0]d is the actual transmitted data using hamming code
d[2]=c[5];
d[3]=c[6];
end
endmodule
```

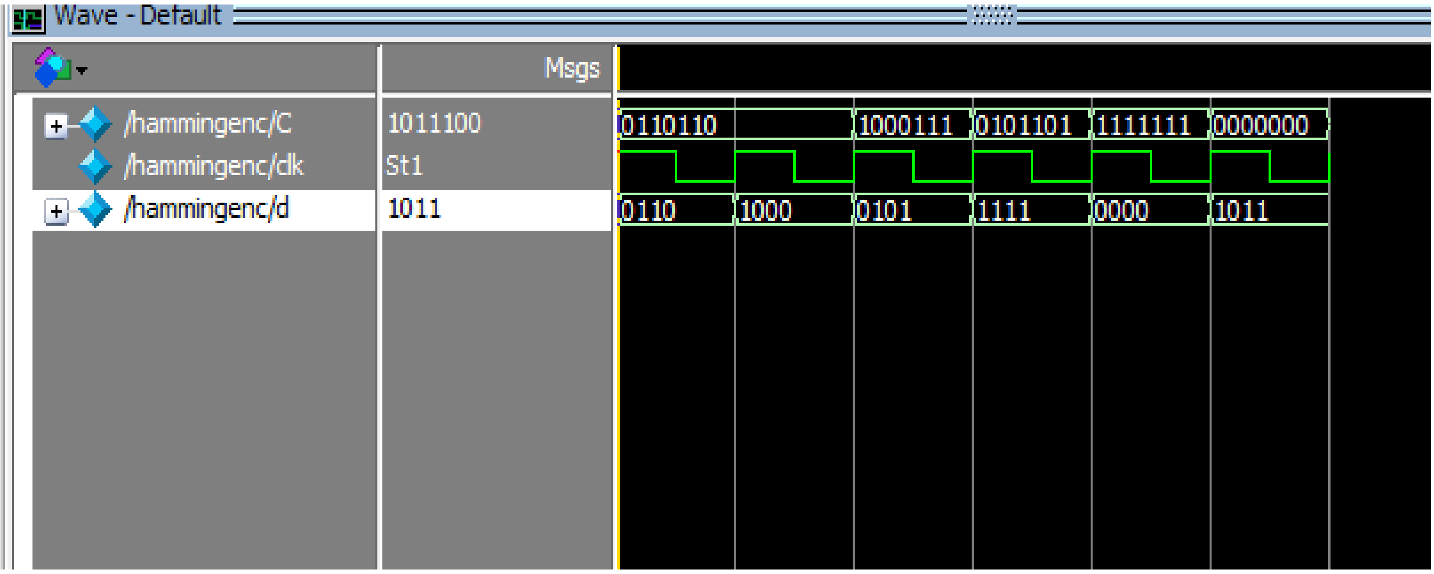
b. RTL View:



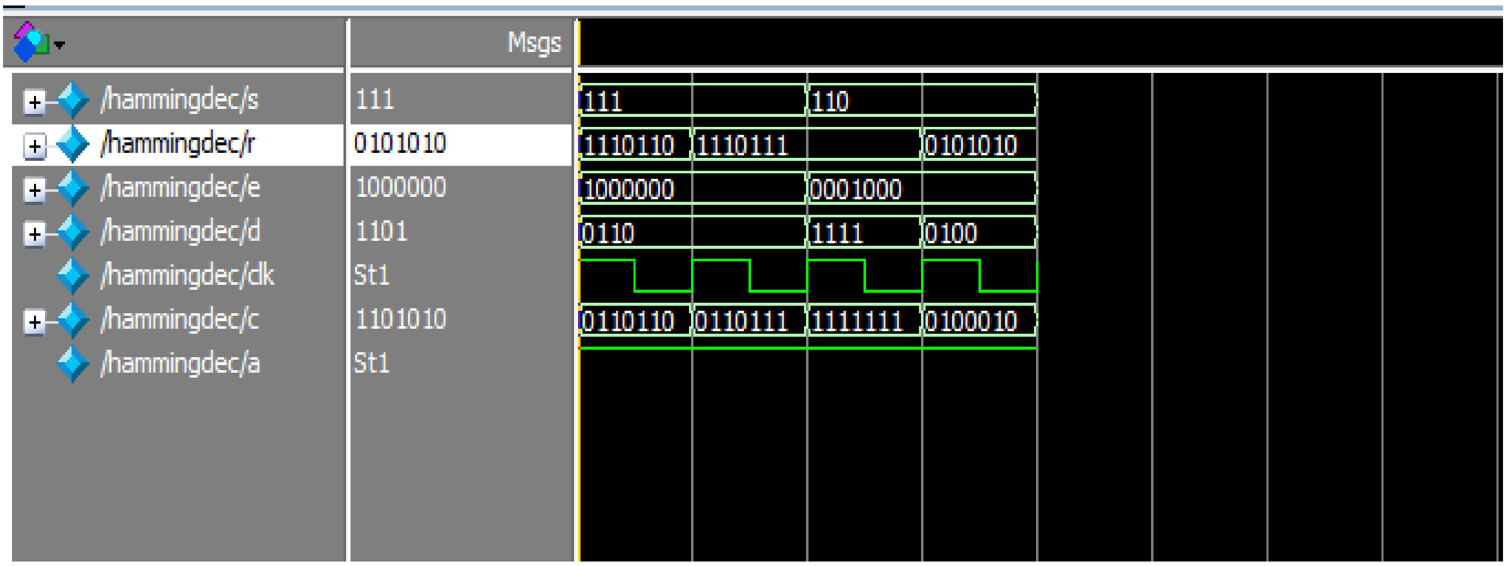
c. **TECHNOLOGY View**



A. SIMULATION FOR ENCODER



B. SIMULATION FOR DECODER



4. CONCLUSION

Hamming Encoder and Decoder were implemented using behavioural style of modelling and successfully implemented on DE-2 FPGA Board. Simulations were also observed and matched appropriately with the hardware output.

ACKNOWLEDGEMENT

This is to thank Prof. Amit Degada for his valuable help and guidance regarding this topic of special assignment.

REFERENCES

[1] Simon Haykin, Communication System, 4^{th} edition and 5^{th} edition, John Wiley and sons, 2001

[2] Simon Haykin, Digital Communication, John Wiley and sons, 2001