Science Focus **Vol. 21 (1) 2016 pp. 37 - 52**
Faculty of Pure and Applies Sciences, LAUTECH, Ogbomosho, Printed in Nigeria

HAMMING (7, 4) CODING ALGORITHM FOR DATA PROTECTION

Ayeni, A.A[1], Yusuf S.O[2], *Onidare, S.O[3] and Adeyemo, Z. K[4].

[1, 2, 3,] Department of Telecommunication Science, University of Ilorin.

[4]Dapartment of Electronics and Electrical Engineering, Ladoke Akintola University of Technology

Abstract

Data is subject to corruption through the communication channel. In other words, it is prone to error as it traverses the source to destination link. Noise in the communication channel makes it difficult to reproduce a message at the destination end system. While some applications can absorb minimal amount of error, during transmission, for example, when transmitting multimedia data such as video and audio, random errors may be overlooked, but such is not allowed in some applications like text that requires high level of accuracy. For this reason, during transmission, efforts need be made towards the detection and correction of these inevitable errors. Channel coding (divided into convolutional and block codes), especially the block codes, is used to address problems of error detection and correction in data. The popular block coding techniques available are BCH code, Hamming code, Reed-solomon code and Golay code. Problems of channel coding can be addressed by automatic repeat-request (ARQ) approach and forward error correction (FEC) approach. This research work deals with forward error correcting code, specifically Hamming code under the umbrella of channel coding. This work, initially, explains the working of a block code and then Hamming coding is demonstrated both practically, using Visual basic and theoretically, using mathematical equations. The tool which was used to simulate this algorithm is Visual Basic; it was adopted because most Visual Studio IDE features are also used in the same context with Microsoft Visual C++, Microsoft Visual C# and Visual Basic. The algorithm was designed for Hamming (7, 4) which is capable of detecting and correcting one bit error in the received vector using the concept of syndrome decoding. When four bits of information (x = 4) is transmitted simultaneously over the noisy channel and if one bit is in error in the received vector, then this code can detect and correct one bit error in the receive vector. A codeword length of 7 bits (y = 7) was used and also presented were generator matrix (4×7), a parity check matrix (3×7) and a decoding matrix (4×7) to achieve the required Hamming distance.

Introduction

The Communication system is generally subject to corruption as data traverses the source to destination system. Usually, the chances of error occurring during data transmission are high, and therefore, reducing error in this system is a huge priority. To achieve an error free data, communication over a channel that is error prone such as a noisy communication channel, error needs to be detected and corrected. Error correcting codes (ECCs) are common tools of attaining reliable system communication and maintaining high data integrity because, they are capable of protecting the system against soft errors that are encountered during communication of data from the sender to the receiver. As a result, it is important to ensure error control in these applications. When transmitting and storing digital data, the performance evaluation of the system could be measured by the bit error rate (BER). The bit error rate can be obtained by dividing the number of error bits by the number of total bits. The presence of noise in the transmission medium may cause obstruction in signal, which is capable of causing data corruption (Heller and Jacobs, 1971). The value of signal -to- noise ratio (SNR), a measure of the relationship between the signal and noise power, of a communication system, usually obtained by dividing the signal power by that of the noise power, is inversely proportional to bit error rate. This indicates that, the lesser the value of the bit error rate, the higher the value of the signal-to-noise ratio and the better the quality of service (QoS) of the transmission process (Sklar, 2001). One of the most widely adopted methods of error-correction is the block coding. In coding theory, channel coding is categorized into convolutional and block codes (Viterbi, 1967). Block codes allow data to be encoded in blocks, common examples of which are, Hamming codes, Golay codes, Reed muller codes, Reed Solomon codes, Hadamar codes, Expander codes. All these types of block codes mentioned earlier are widely adopted because they can be implemented practically and also because generally speaking, block codes permit different fields such as mathematics, computer science and code theory to study error detection and correction ability and rate, bounds of different block codes parameter, etc, of all block codes in a unified manner (Sklar, 2001). During transmission, the chances of error occurring is inevitable; detection and correction of these errors are carried out by some applications while some can absorb minimal amount of error when transmitting. For example, when transmitting multimedia data such as video and audio random errors may be overlooked but such is not allowed in some application like text that requires high level of accuracy.

***Corresponding author email:** *onidaresam@yahoo.com*

ISSN 1596-7026

As mentioned earlier, different block codes are available for error detection and correction but this paper only considers the Hamming code technique, for detecting and correcting error, during data transmission. Hamming codes is needed if more than one bit changes at a time. Hamming code adopts the technique of parity and parity bits, that is, each data is assigned a bit and this serves as validity check for data when it is read or after being received, during a data communication. Assigning more than one parity bit in the hamming code method does not only identify the single bit error; the position of the error in the data unit is also identified. During data transmission, the method adopted by the receiver station in, correcting error, is referred to as forward error correction (FEC) and this approach increases the throughput of the transmission, in the case of lots of noise in the system. This method can be enabled by adding additional data known as error correction bits during the transmission process. Two-bit errors are detected in Hamming code, by assigning more than one parity bit, which are computed on a separate combination of bits during transmission process. The Hamming codeword can be generated by combining the data bits and parity bits. This can be obtained by the equation

$$\text{Codeword, } C = Z \quad \cdot \quad G_x$$

Literature Review

The Noisy-channel coding theorem stated by Shannon describes the total efficiency of error-correcting techniques versus the degree of noise interference and data corruption. Shannon theorem states that given a noisy channel with channel capacity C and information transmitted at a rate R, and then if $R < C$, there exist a code that allow the probability of error at the receiver to be made arbitrarily small. This means that, theoretically, it is possible to transmit information, almost, without error at any rate below a limiting rate, C. The maximum rate at which we can transmit information is set by the bandwidth, the signal level, and the noise level. C is therefore called the channel's information carrying Capacity.

$$C = 2B\, log_2 \left\{ \left(1 + \frac{S}{N}\right)^{\frac{1}{2}} \right\}$$

$$C = Blog_2 \left(1 + \frac{S}{N}\right)$$

Analysis and Design of the Algorithm

The Hamming error detection and correction algorithm encode the four bits data that the user enters into the source encoder. Once the encode button is clicked, the encoder uses the generator matrix and the four bits user data to compute the codeword. This algorithm is a computer-based type that makes use of a computer system to achieve the aim of the Hamming (7, 4) algorithm. The Hamming code is simple to understand, compared with the other error correcting-codes. It adopts the approach of extra redundant bits for checking errors, by making use of special check equations. This research work involves the use of matrices such as Generator matrix, Check matrix and decoding matrix. The decoding matrix is used to get back the original four bits data from the corrected received codeword. The demonstration of how this code works is as follows:

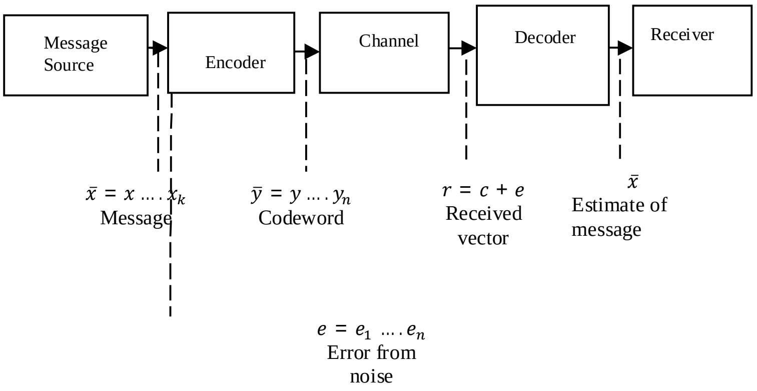


Fig.1: A system Block diagram showing how devices are connected.

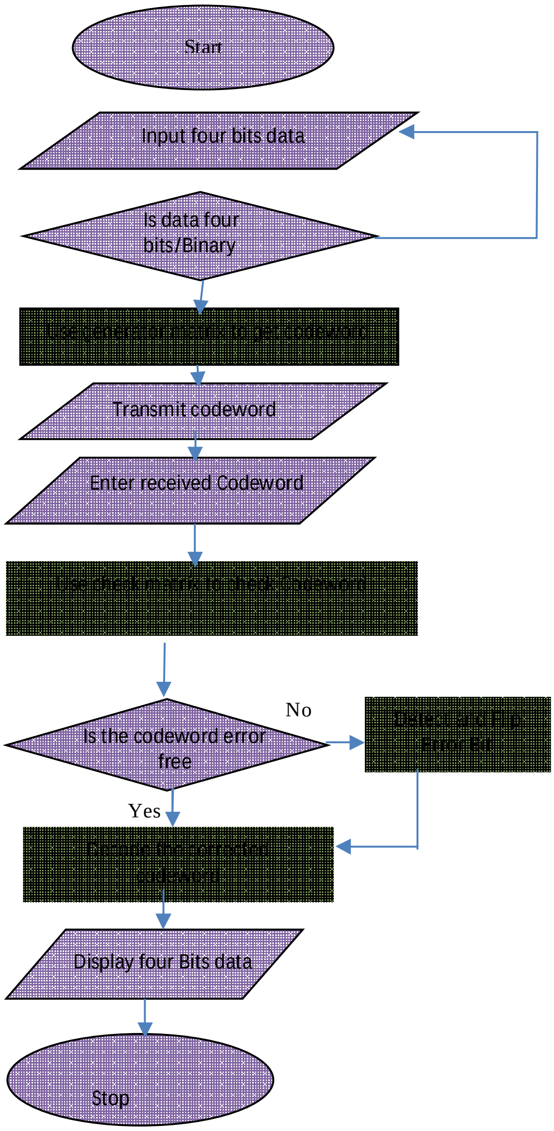

Start
Input four bits data
Is data four bits/Binary
Use generator matrix to get codeword
Transmit codeword
Enter received Codeword
Use check matrix to check Codeword
Is the codeword error free
No
Detect and Flip Error Bit
Yes
Decode the corrected codeword
Display four Bits data
Stop

Fig. 2: The program flow chart

Hamming code with matrix manipulation

Consider the four un-encoded information bits x_1 , x_2, x_3 and x_4. This 1 x 4 matrix can be demonstrated as

$$\left[\overline{X}\right] = \left[X_1 \;\; X_2 \;\; X_3 \;\; X_4\right] \qquad (1)$$

The seven-bit encoded data is represented as;

$$\left[\overline{Y}\right] = \left[Y_1 \;\; Y_2 \;\; Y_3 \;\; Y_4 \;\; Y_5 \;\; Y_6 \;\; Y_7\right] \qquad (2)$$

The encoding process of the Hamming code can be represented as modulo-2 multiplication;

$$\left[x_1 \; x_2 \; x_3 \; x_4\right] \times \left[Gx\right] = \left[y_1 \; y_2 \; y_3 \; y_4 \; y_5 \; y_6 \; y_7\right] \qquad (3)$$

Where y_1, y_2, y_3, and y_4 represent the data bits while y_5, y_6, and y_7 represents the check bits.

In the design, the modular check equations listed below were used:

$$D_1= x_2 \oplus x_3 \oplus x_4 \qquad (4)$$

$$D_2= x_1 \oplus x_3 \oplus x_4 \qquad (5)$$

$$D_3= x_1 \oplus x_2 \oplus x_4 \qquad (6)$$

$$D_4= x_1 \oplus x_2 \oplus x_3 \qquad (7)$$

$$A_1 = \begin{matrix}0\\1\\1\\1\end{matrix} \quad A_2 = \begin{matrix}1\\0\\1\\1\end{matrix} \quad A_3 = \begin{matrix}1\\1\\0\\1\end{matrix} \quad A_4 = \begin{matrix}1\\1\\1\\0\end{matrix}$$

$$B_1 = \begin{matrix}1\\0\\0\\0\end{matrix} \quad B_2 = \begin{matrix}0\\1\\0\\0\end{matrix} \quad B_3 = \begin{matrix}0\\0\\1\\0\end{matrix} \quad B_4 = \begin{matrix}0\\0\\0\\1\end{matrix}$$

From the modular check equations, that is, equation (4), (5), (6), and (7), check bits D1 to D4 can also be formed by representing, in the equations, every data bit position that was present in each check equation by a one (1). More so, B1 to B4 can also be formed by changing all bit positions in D1 to D4 that has 1's to zeros (0's) and also the position with a zero to a one (1).

$$\mathbf{G_x} = \begin{array}{ccccccc} B_1 & B_2 & B_3 & B_4 & A_1 & A_2 & A_3 \\ \left[\begin{matrix}1\\0\\0\\0\end{matrix}\right. & \begin{matrix}0\\1\\0\\0\end{matrix} & \begin{matrix}0\\0\\1\\0\end{matrix} & \begin{matrix}0\\0\\0\\1\end{matrix} & \begin{matrix}0\\1\\1\\1\end{matrix} & \begin{matrix}1\\0\\1\\1\end{matrix} & \left.\begin{matrix}1\\1\\0\\1\end{matrix}\right] \end{array}$$

From equation (3) above, substituting G_x into the equation we have:

$$[x_1\ x_2\ x_3\ x_4]\times[Gx]=[y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7] \text{........... } (3)$$

$$[x_1\ x_2\ x_3\ x_4] \times \begin{bmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{bmatrix} = [y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7]$$

Where the 4 x 7 matrix is known as a generator matrix represented by $[Gx]$. The encoding process is done by multiplying the information vector $[\overline{X}]$ by a generator matrix $[Gx]$ to obtain the encoded bit sequence $[\overline{Y}]$.

$$[\overline{Y}] = [\overline{X}]\times[Gx] \qquad (3)$$

From the generator matrix $\mathbf{G_x}$ of the Hamming code above:

$$G_x = \begin{bmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{bmatrix}$$

$$[Gx] = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

$$[Gx] = [[I]_{4\times 4}\ [P]]$$

4 x 4 identity sub-matrix $[I]$ 4 x 3 parity sub-matrix $[P]$

Fig. 3: Show the Deconstruction of generator matrix $[Gx]$

Parity check

When the data transmission is error free we have

$$[r] = \left[\bar{Y}\right]$$

Which indicates that the receiver codeword [r] is the same with $\left[\bar{Y}\right]$ which represent the transmitted codeword.

Parity Check Matrix

$$[H] = [[A]^T\ [I]_{3\times 3}]$$

Note that $[A]^T$ becomes a matrix of dimension 3 x 4.

$$[H] = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

3x4 transpose of parity matrix $[A]$ 3x3 identity submatrix $[I]$

Figure 4: deconstruction of check matrix $[H]$

Now let $[r]$ represents the received codeword, the **syndrome** vector [S] can be obtained by,

$$[S] = [r] \times [H]T$$

$$[S] = [y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7] \times \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since the multiplication involves a [1×7] and a [7×3] matrix, therefore the resulting syndrome would be a [1×3] row matrix.

$$W \oplus Z = \begin{cases} 0 \text{ } if \quad W = Z \\ 1 \text{ } if \quad W \neq Z \end{cases}$$

If the result of [S] gives an all zeros [1×3] row matrix, the receiver takes it that the transmission is error free. This agreement or disagreement is exactly what is needed for single-bit error detection and correction in the received seven-bit sequence.

Table 1: The Syndrome generated and the bit in error

[S]	Bit received in Error
[0 0 0]	None
[0 1 1]	y_1
[1 0 1]	y_2
[1 1 0]	y_3
[1 1 1]	y_4
[1 0 0]	y_5
[0 1 0]	y_6
[0 0 1]	y_7

Error Correction

If a single bit error was encountered. Mathematically, we can write

$$[r] = \left[\bar{Y}\right] + e_i$$

Modulo 2, signifies that e_i is the i^{th} unit vector (a zero vector with a 1 in the i^{th}, counting from 1)

$$\mathbf{e_2} = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

The expression above shows a single bit error in the i^{th} place.

Now, by multiplying the vector by H:

$$[H]\,[r] = [H]\,(\bar{y} + e_i) = [H]\,[\bar{y}] + [H]\,[e_i]$$

Since the transmitted data $\bar{y}$ is error free, we have zero by multiplying [H] and $\bar{y}$. Thus

$$[H]\,[\bar{y}] + [H]\,[e_i] = 0 + [H]\,[e_i] = [H]\,[e_i]$$

Now, from [H] with the i^{th} standard basis vector if we remove the column of [H], we are certain the error occurred in column [H] location. For example, suppose a bit error on bit number 4 is introduced, where the initial data sent is 0110 then the result will be:

$$[r]= (\bar{v} \quad + e_4) = \begin{pmatrix}0\\1\\1\\0\\0\\1\\1\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}0\\1\\1\\1\\0\\1\\1\end{pmatrix} = \begin{pmatrix}0\\1\\1\\1\\0\\1\\1\end{pmatrix}$$

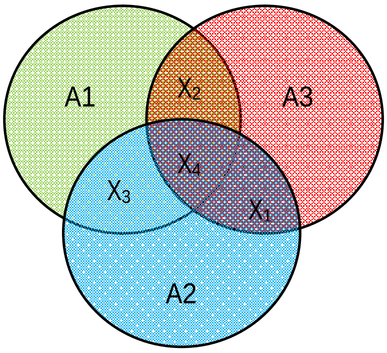


Fig. 5: Representing a codeword using a Venn diagram

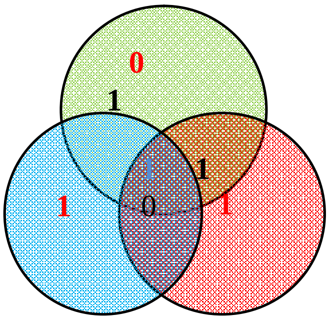


Fig. 6: A bit error invoking bad parity in all the other bits.

Fig. 6 shows how a bit error as indicated, in blue text, in the above Venn diagram invokes bad parity in the entire circle as indicated in red text. To detect bit error, the parity of the circle needs to be detected. Once we detect a faulty parity, then the location of the bit with error is the data bit that intersects, only, the faulty parity. From the Venn diagram in fig. 5 A_1, A_2 and A_3 are all infected with bad parity, so all the bits that fall within the intersection of A_1, A_2 & A_3 which is the one represented with blue text ($\mathbf{X_4}$) indicates the error bit.

$$S] = [H] \times [r] = \begin{pmatrix}0 & 1 & 1 & 1 & 1 & 0 & 0\\1 & 0 & 1 & 1 & 0 & 1 & 0\\1 & 1 & 0 & 1 & 0 & 0 & 1\end{pmatrix} \times \begin{pmatrix}0\\1\\1\\1\\0\\1\\1\end{pmatrix} = \begin{pmatrix}3\\3\\3\end{pmatrix} = \begin{pmatrix}1\\1\\1\end{pmatrix}$$

The algorithm was designed in such a manner that the syndrome [H] which is located in the 4th column is identical to the decimal value of 7, which signifies the presence of bad parity in 4th bit in this design. Furthermore, the error detected in the 4th bit can be tackled through the process of negation of its value or flip method.

$$r_{corrected} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ \overline{1} \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

From the above, the corrected value of the error detected in the 4th column corresponds the transmitted value $[\overline{Y}]$.

Decoding

In a situation where only zero or one bit of errors are encounter, in order to correct this error the received data must be converted or decoded back to its original form first by defining a decoding matrix R as shown below.

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

X_r which is the received value can be obtained by:

$$\mathbf{X_r = R \quad \times \quad r_c}$$

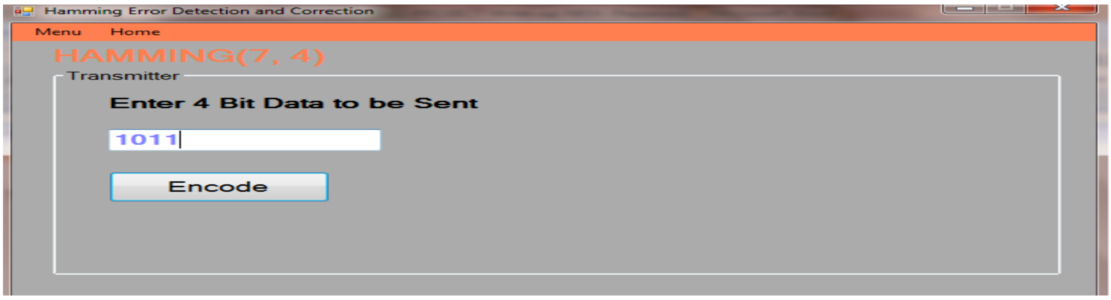
and using the example from above:

$$\mathbf{X_r} = \mathbf{X_r} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

The actual transmitted data is retrieved by the receiver after the decoding operation is carried out by the decoder on the corrected received codeword.

Testing of the Algorithm

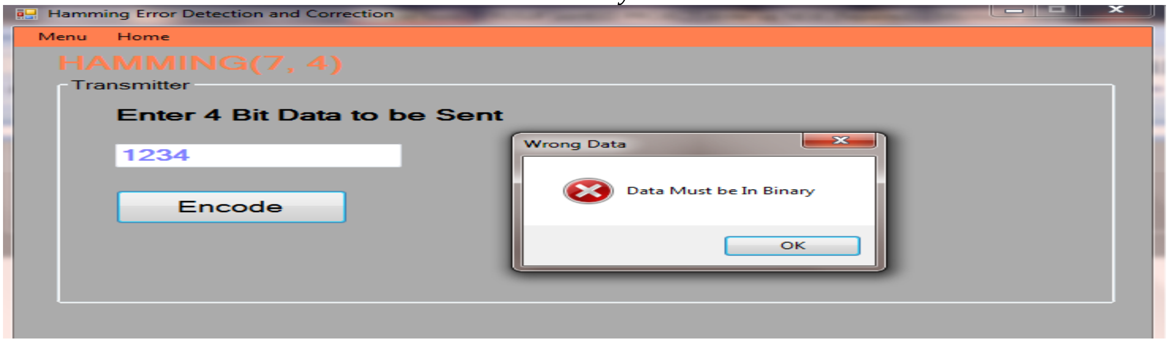
The interface of the algorithm is designed to be operated by human agents. The user of the algorithm will need to input a four bits data to be transmitted, then the operator will just need to click on interface button to generate a result. The use of the algorithm is easy as it is user friendly and self–explanatory (instructional).



Enter 4 Bit Data to be Sent

1011

Fig. 7: The first stage of using the algorithm



Enter 4 Bit Data to be Sent

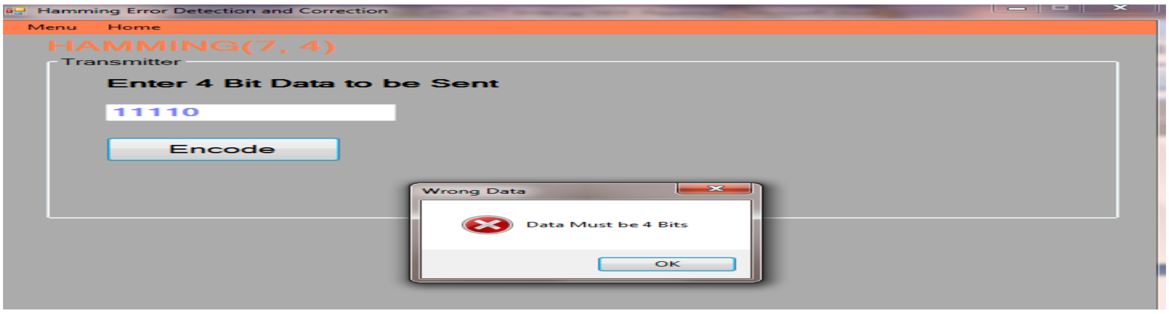
1234

Wrong Data

Data Must be In Binary

Fig. 8: Error message stating that binary data must be input.

Fig. 8 shows the error message that will be displayed, if anything other than four-bit binary data is inputted into the algorithm.



Enter 4 Bit Data to be Sent

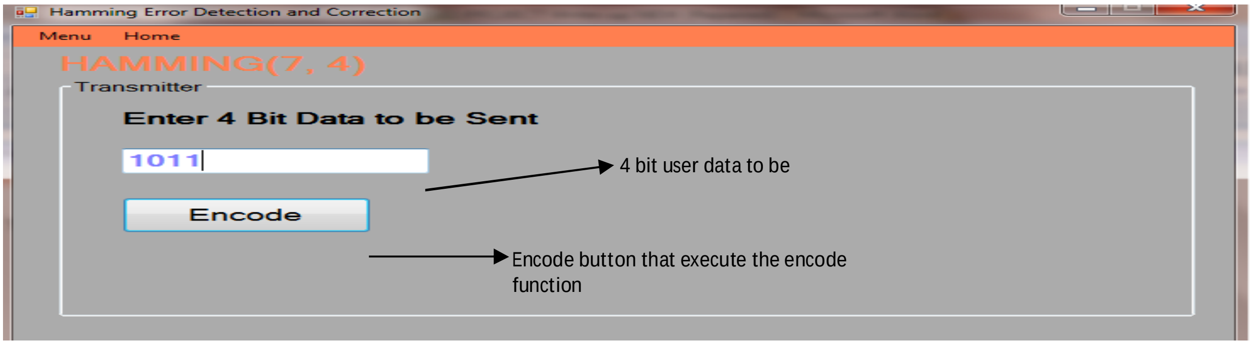
11110

Wrong Data

Data Must be 4 Bits

Fig. 9: Error message, indicating that the length of information data must be 4 bits.

From fig.9, it is shown that the data this algorithm can process must be 4-bit but if otherwise, an error message will be displayed saying "data must be 4 bits".


Hamming Error Detection and Correction
Menu
Home
HAMMING(7, 4)
Transmitter
Encode
4 bit user data to be
Encode button that execute the encode function

Enter 4 Bit Data to be Sent

1011

Fig. 10: Show the four-bit data to be encoded

Analysis of the Result

In Hamming codes, each integer $k \geq 2$ has a code with:

$$L = 2^k - 1,$$

$$F = 2^k - k - 1, \text{ and}$$

$$R = F/L = 1 - k/(2^k - 1),$$

Which is the highest possible for codes with distance 3.

Since, in this algorithm, the redundant bit $k = 3$. Therefore, L, given by the formula: $L = 2^k - 1$ equals:

$$L = 2^k - 1 = 2^3 - 1 = 8 - 1 = 7\text{bits}$$

Where L, F and R indicate the block length, message length and rate of Hamming code respectively.

Table 2: Using the value of k to generate all realistic Hamming codes.

Parity bits	Total bits	Data bits	Name	Rate
2	3	1	Hamming(3,1) (Triple repetitive code)	1/3≈0.333
3	7	4	Hamming(7,4)	4/7≈0.571
4	15	11	Hamming(15,11)	11/15≈0.733
5	31	26	Hamming(31,26)	26/31≈0.839
K	2^k-1	2^k-k-1		$\mathbf{1-k}$ /($\mathbf{2^k-k}$)

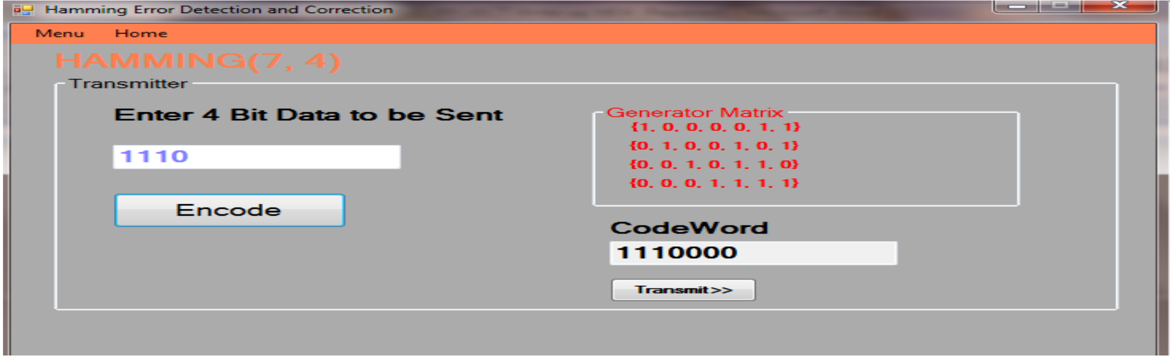
The detection/correction capacity depends on the minimum distance of a code, which was demonstrated in the theorem below:

Theorem 1: Let E be a code with minimum distance r. Then E can detect any r – 1 errors, and can correct any $\left[\frac{\mathbf{r-1}}{\mathbf{2}}\right]$ errors. Moreover, there are some r errors E cannot detect and there are $\left[\frac{\mathbf{k-1}}{\mathbf{2}}\right]$ + 1 errors E cannot correct. Also since 4 bits is used for the data bits in this algorithm, the expected levels will be:

$$\text{Levels} = \mathbf{2^4} = 16 \text{ levels}$$

Table 3: Show the expected levels in a system using four-bits to represent data and the codeword for each input.

	Information bits				Parity bits			Codeword
Levels	x_1	x_2	x_3	x_4	A_1	A_2	A_3	$\bar{Y} = y_1\ y_2\ y_3\ y_4\ v_5\ y_6\ y_7$
0	0	0	0	0	0	0	0	0 0 0 0 0 0 0
1	0	0	0	1	1	1	1	0 0 0 1 1 1 1
2	0	0	1	0	1	1	0	0 0 1 0 1 1 0
3	0	0	1	1	0	0	1	0 0 1 1 0 0 1
4	0	1	0	0	1	0	1	0 1 0 0 1 0 1
5	0	1	0	1	0	1	0	0 1 0 1 0 1 0
6	0	1	1	0	0	1	1	0 1 1 0 0 1 1
7	0	1	1	1	1	0	0	0 1 1 1 1 0 0
8	1	0	0	0	0	1	1	1 0 0 0 0 1 1
9	1	0	0	1	1	0	0	1 0 0 1 1 0 0
10	1	0	1	0	1	0	1	1 0 1 0 1 0 1
11	1	0	1	1	0	1	0	1 0 1 1 0 1 0
12	1	1	0	0	1	1	0	1 1 0 0 1 1 0
13	1	1	0	1	0	0	1	1 1 0 1 0 0 1
14	1	1	1	0	0	0	0	1 1 1 0 0 0 0
15	1	1	1	1	1	1	1	1 1 1 1 1 1 1



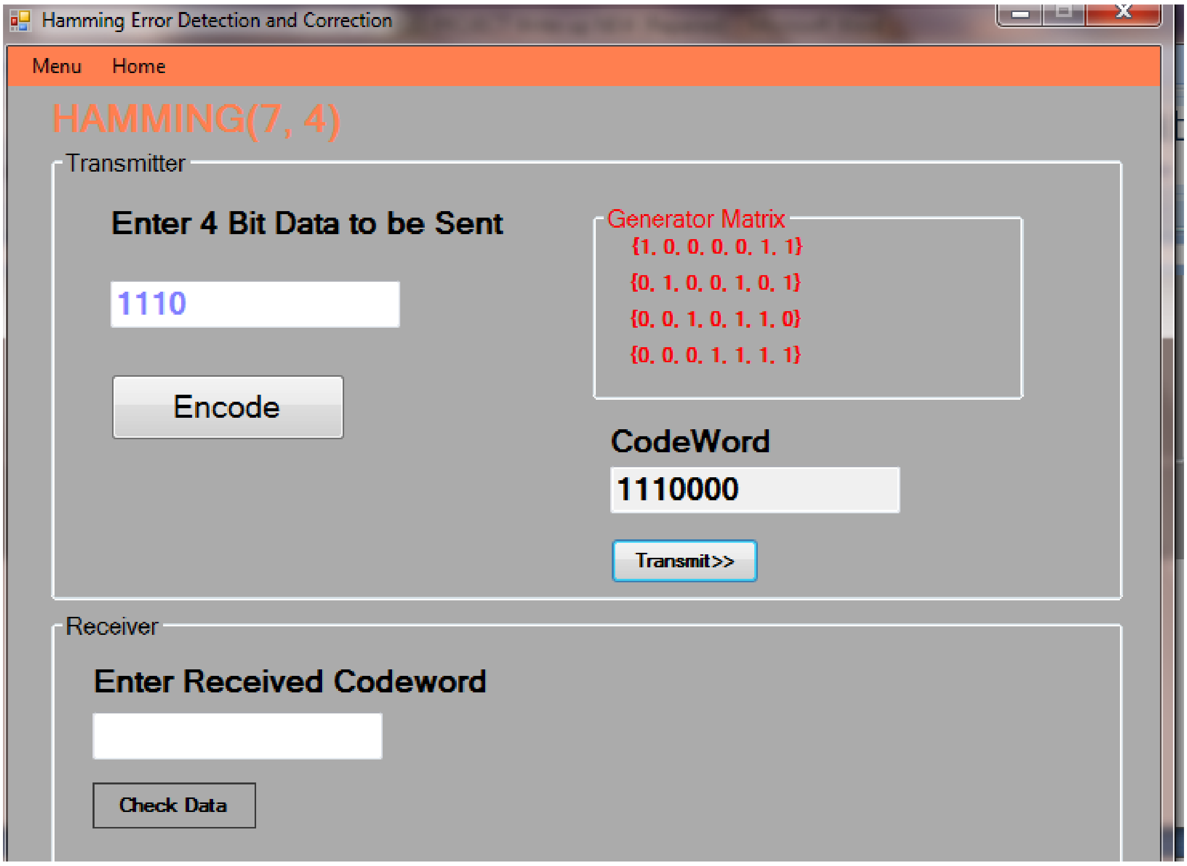
Enter 4 Bit Data to be Sent
1110

Generator Matrix
{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}

CodeWord
1110000

Fig. 11: the result of the encode function for a 4-bits data.

From fig. 11, it is shown that when a four-bit data from the message source is fed into an encoder, the algorithm is designed to append redundant bits to the data so that if the channel is noisy, data corruption can easily be detected and corrected. By clicking the encode button, the algorithm generate a codeword of length n=7 where the check-bit m= 7 - 4 = 3 bits



Enter 4 Bit Data to be Sent
1110

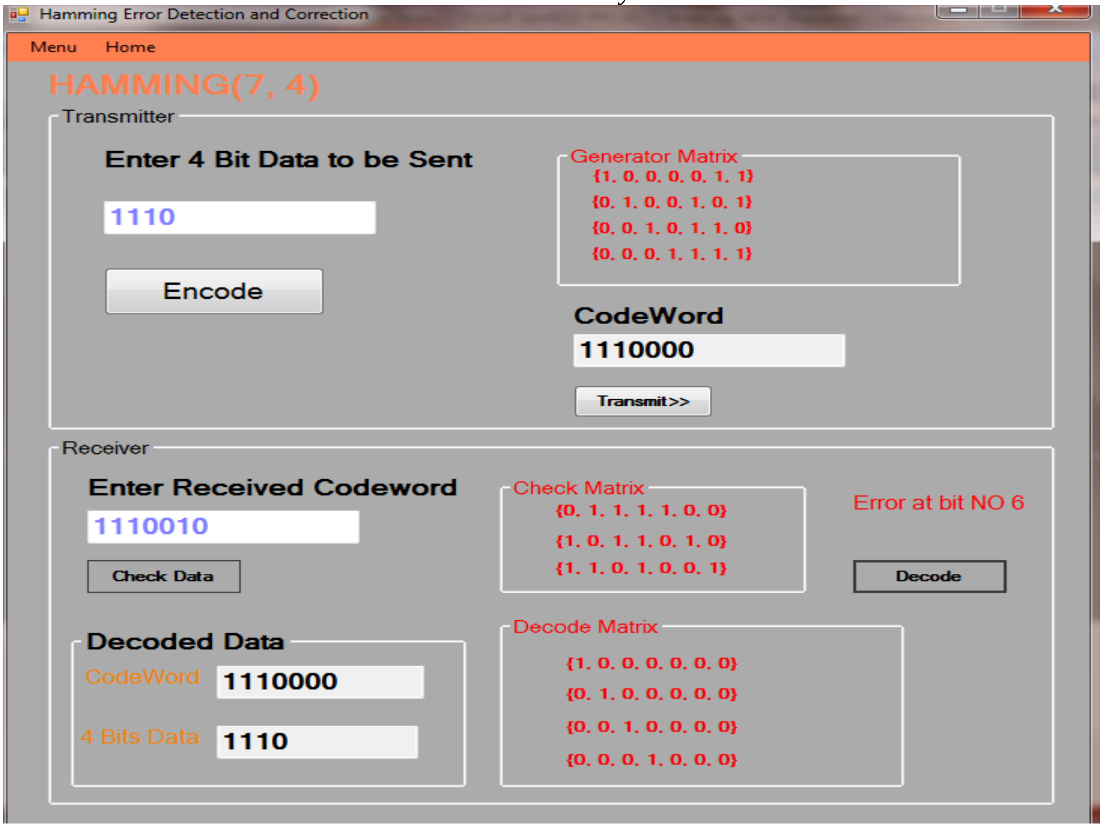
Generator Matrix
{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}

CodeWord
1110000

Enter Received Codeword

Fig. 12: The receiver interface asking for the codeword received.

Fig. 12 shows the screenshot from the algorithm, when the transmit button is clicked. As the user clicks the transmit button, the algorithm will bring out the field for the user to enter the codeword received. At this stage, the user can decide to flip any of the 7-bits and the algorithm will detect the error.



HAMMING(7, 4)

Transmitter

Enter 4 Bit Data to be Sent

1110

Generator Matrix

{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}

CodeWord

1110000

Receiver

Enter Received Codeword

1110010

Check Matrix

{0, 1, 1, 1, 1, 0, 0}
{1, 0, 1, 1, 0, 1, 0}
{1, 1, 0, 1, 0, 0, 1}

Error at bit NO 6

Decoded Data

CodeWord 1110000

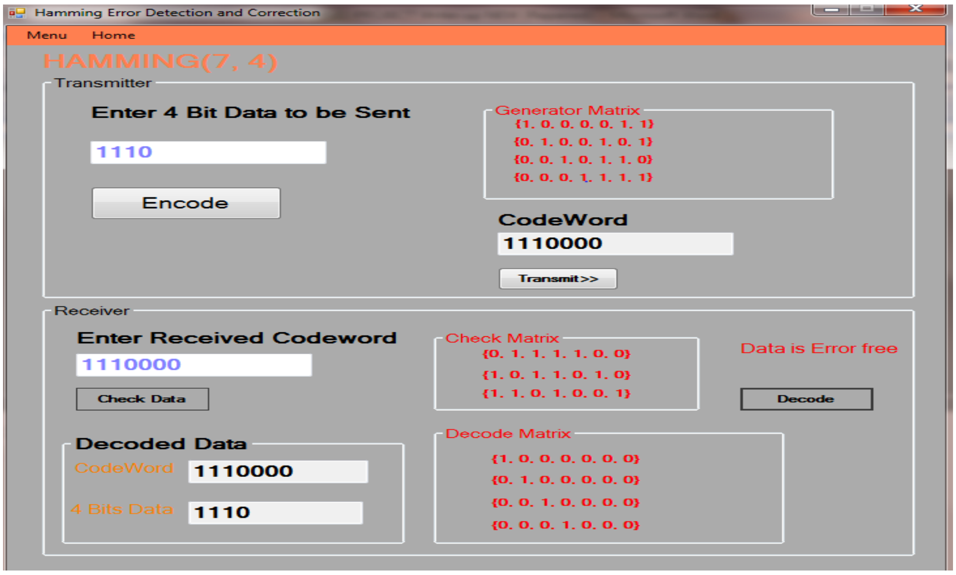
4 Bits Data 1110

Decode Matrix

{1, 0, 0, 0, 0, 0, 0}
{0, 1, 0, 0, 0, 0, 0}
{0, 0, 1, 0, 0, 0, 0}
{0, 0, 0, 1, 0, 0, 0}

Fig. 13: The algorithm message when the bit position six is in error.

From fig. 13, when the check data button is clicked, the algorithm uses the received codeword and check matrix to compute the syndrome then the syndrome is compared with the column of the check matrix to determine the bit in error. For example the screenshot above show that bit six of the codeword is in error.



HAMMING(7, 4)

Transmitter

Enter 4 Bit Data to be Sent

1110

Generator Matrix

{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}

CodeWord

1110000

Receiver

Enter Received Codeword

1110000

Check Matrix

{0, 1, 1, 1, 1, 0, 0}
{1, 0, 1, 1, 0, 1, 0}
{1, 1, 0, 1, 0, 0, 1}

Data is Error free

Decoded Data

CodeWord 1110000

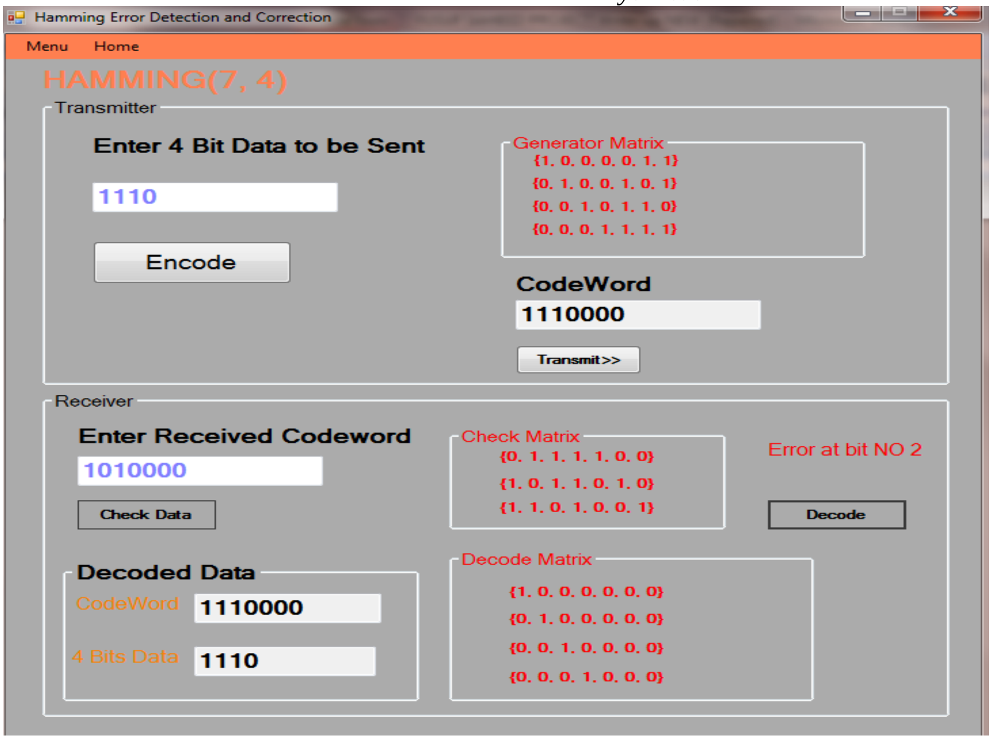
4 Bits Data 1110

Decode Matrix

{1, 0, 0, 0, 0, 0, 0}
{0, 1, 0, 0, 0, 0, 0}
{0, 0, 1, 0, 0, 0, 0}
{0, 0, 0, 1, 0, 0, 0}

Fig. 14: The error free message when the transmitted codeword is the same as the received vector.

It has been explained earlier that when the transmitted codeword is the same as that of the received vector, the algorithm will generate a (1×3) all zero's row matrix which, when traced with all columns of the check matrix, will find no match.



HAMMING(7, 4)

Transmitter

Enter 4 Bit Data to be Sent: 1110

Generator Matrix

{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}

CodeWord: 1110000

Receiver

Enter Received Codeword: 1010000

Check Matrix

{0, 1, 1, 1, 1, 0, 0}
{1, 0, 1, 1, 0, 1, 0}
{1, 1, 0, 1, 0, 0, 1}

Error at bit NO 2

Decoded Data

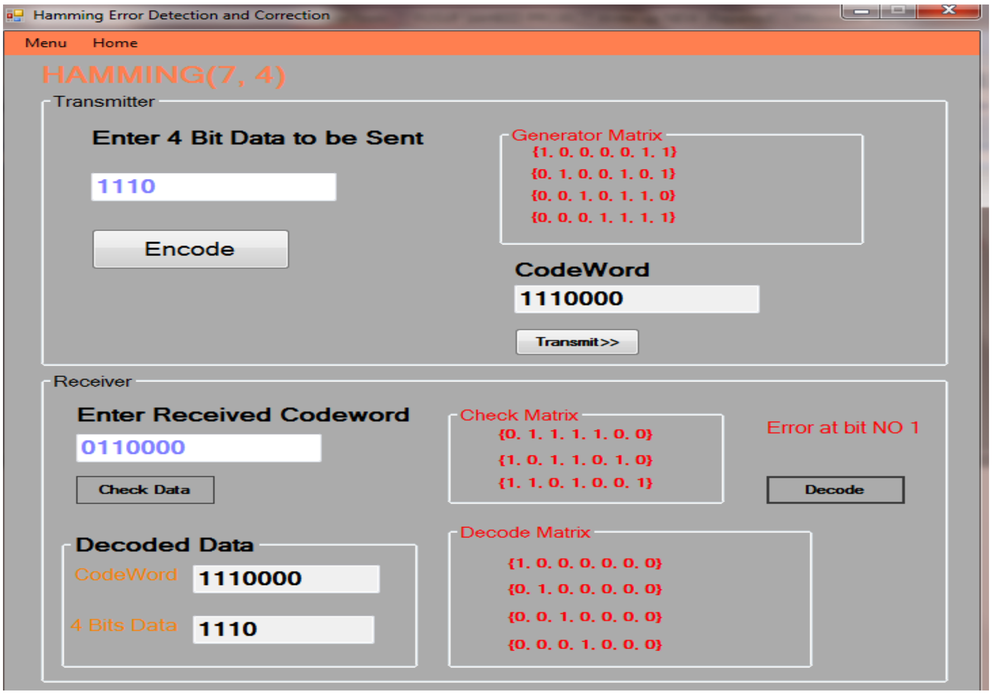
CodeWord: 1110000

4 Bits Data: 1110

Decode Matrix

{1, 0, 0, 0, 0, 0, 0}
{0, 1, 0, 0, 0, 0, 0}
{0, 0, 1, 0, 0, 0, 0}
{0, 0, 0, 1, 0, 0, 0}

Fig. 15: Showing that bit position two is in error



HAMMING(7, 4)

Transmitter

Enter 4 Bit Data to be Sent: 1110

Generator Matrix

{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}

CodeWord: 1110000

Receiver

Enter Received Codeword: 0110000

Check Matrix

{0, 1, 1, 1, 1, 0, 0}
{1, 0, 1, 1, 0, 1, 0}
{1, 1, 0, 1, 0, 0, 1}

Error at bit NO 1

Decoded Data

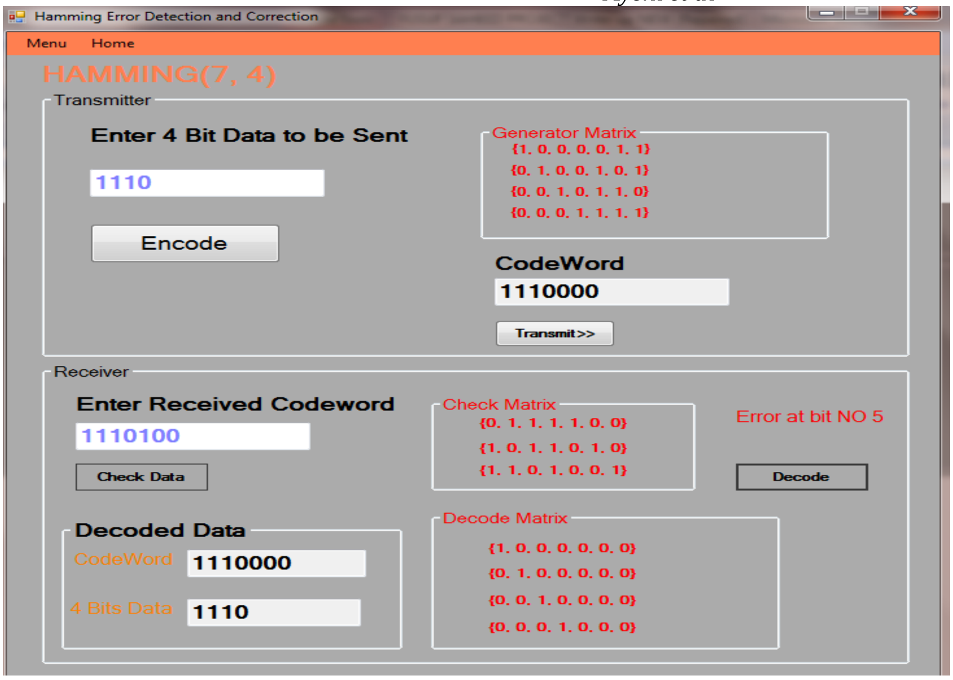
CodeWord: 1110000

4 Bits Data: 1110

Decode Matrix

{1, 0, 0, 0, 0, 0, 0}
{0, 1, 0, 0, 0, 0, 0}
{0, 0, 1, 0, 0, 0, 0}
{0, 0, 0, 1, 0, 0, 0}

Fig. 16: Showing that bit position 1 is in error.



Enter 4 Bit Data to be Sent: 1110

Generator Matrix:

```
{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}
```

CodeWord: 1110000

Enter Received Codeword: 1110100

Check Matrix:

```
{0, 1, 1, 1, 1, 0, 0}
{1, 0, 1, 1, 0, 1, 0}
{1, 1, 0, 1, 0, 0, 1}
```

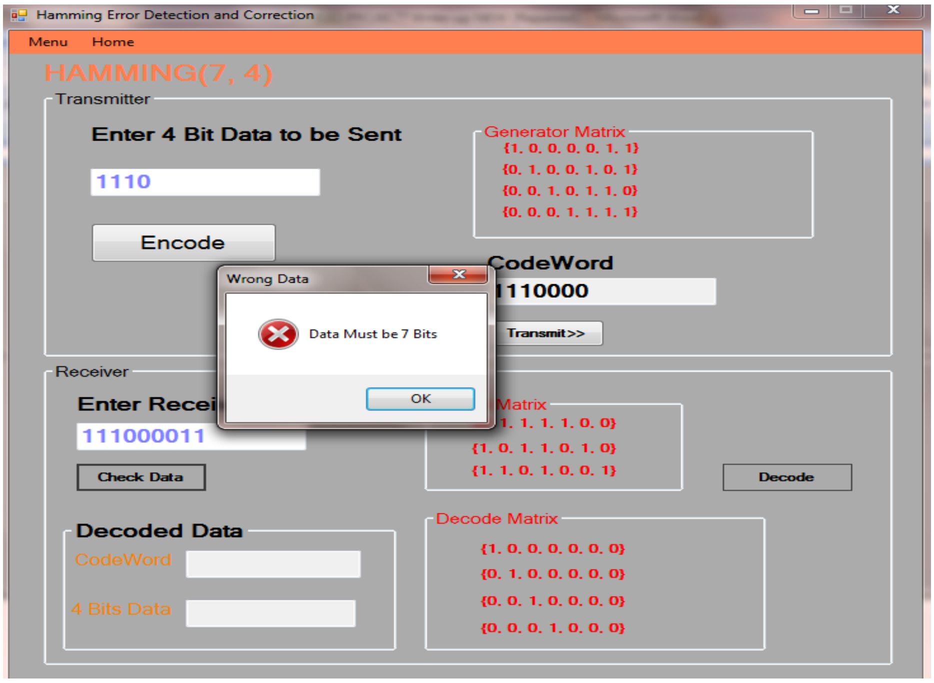
Error at bit NO 5

Decoded Data — CodeWord: 1110000; 4 Bits Data: 1110

Decode Matrix:

```
{1, 0, 0, 0, 0, 0, 0}
{0, 1, 0, 0, 0, 0, 0}
{0, 0, 1, 0, 0, 0, 0}
{0, 0, 0, 1, 0, 0, 0}
```

Fig. 17: There is error in bit number 5



Enter 4 Bit Data to be Sent: 1110

Generator Matrix:

```
{1, 0, 0, 0, 0, 1, 1}
{0, 1, 0, 0, 1, 0, 1}
{0, 0, 1, 0, 1, 1, 0}
{0, 0, 0, 1, 1, 1, 1}
```

CodeWord: 1110000

Enter Received Codeword: 111000011

Wrong Data: Data Must be 7 Bits

```
1, 1, 1, 1, 0, 0}
{1, 0, 1, 1, 0, 1, 0}
{1, 1, 0, 1, 0, 0, 1}
```

Decode Matrix:

```
{1, 0, 0, 0, 0, 0, 0}
{0, 1, 0, 0, 0, 0, 0}
{0, 0, 1, 0, 0, 0, 0}
{0, 0, 0, 1, 0, 0, 0}
```

Fig. 18: An error message is displayed when the user enter more than 7-binary data.

Fig. 18 shows the complete process of encoding and decoding data, using the designed algorithm. To encode a data, after the algorithm has been launched, the user enters a four bits data and click the encode button. Once the encode button is clicked, the algorithm used the encoding matrix to manipulate the four bits data and it generates a codeword, as the output of the encoding process.

The next step is to transmit the generated codeword. To do this, the user will click the transmit button and the algorithm will bring out the decoder side of the algorithm (note: the assumption is that a visual channel is used, that is, an imaginary transmission medium) so that as the transmit button is clicked, the decoder input interface is prompt up. At the receiver's end, the input of the decoder was designed in such a way that the user will have to type the received codeword and can decide to flip any of the seven bits sequence. Once the user clicks the “check data button”, the algorithm will use the 3 × 7 matrix called the H-matrix (check matrix) to compute the syndrome, this syndrome is then compared with H-matrix column to detect which bit is in error. Once the error has been detected and corrected in the seven bits received sequence, the decoder will make use of another matrix called the decoding matrix (4 × 7 matrix) to get back the actual four bits original data.

Conclusion

The objective of this project is to design and develop an algorithm, capable of detecting and correcting error, in the received sequence, provided it is a bit different from the transmitted codeword and this was achieved. This is just a simulation design to be used on a computer system. The algorithm is capable of processing four bits data to generate seven bits codeword with three bit added as redundancy Hamming (7, 4) was believed to have the highest efficiency of about 33-percent compare to other Hamming codes such as H(15, 11), H(31, 26).

References

A. Feinstein. “A New basic theorem of information theory.” IEEE Transactions on Information Theory, 4(4):2-22, 1954.

A. J Viterbi. “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.” IEEE Transactions on Information Theory, vol.IT-13, pp. 260-269, April, 1967.

B. K. Gupta and R. L. Dua. (2011,). “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 Published by Foundation of Computer Science, New York, USA [Online]. Available: www.ijcaonline.org [Mar. 27, 2013].

B. Sklar “Digital Communications Fundamentals and Applications,” 2nd edition, Prentice Hall, Upper Saddle River, New Jersey, 2001.

C. E. Shannon. A Mathematical Theory of Communication Urbana, IL:University of Illinois Press, 1949 (reprinted 1998).

D. J. C. MacKay. (2004, August 25,). Information Theory, Inference, and Learning Algorithms. (third printing). [on-line]. Available: www.inference.phy.cam.ac.uk/mackay/itila/ [May 21, 2013].

D. J. Costello, Joachim Hagenauer, Hideki Imai and Stephen B. Wicker. (1998, Oct). “Applications of Error-Control Coding.” IEEE TRANSACTIONS ON INFORMATION THEORY. [Online]. VOL. 44, NO. 6, Available: www.engr.usask.ca/classes/EE/814/Papers/Applications-ECC.pdf [Mar. 26, 2013].

E. Abuelyaman, A. Al-Sehibani. “Optimization of the Hamming Code for Error Prone Media.” IJCSNS International Journal of Computer Science and Network Security. [online]. VOL.8 No.3, March 2008. Available: paper.ijcsns.org/07_book/200803/20080340.pdf [April 23, 2013].

G. C Clark Jr., J. B. Cain, (1981). Error-Correction Coding for Digital Communications. New York: Plenum Press. ISBN 0-306-40615-2.

H. L. Kumawat and S. Sharma. “Forward error correction (FEC) coding technique using convolutional encoding with Viterbi decoding”. International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-5, November 2012.

H. L. Kumawat and Sandhya Sharma. “An Implementation of a Forward Error Correction Technique using Convolution Encoding with Viterbi Decoding,” International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-5, November 2012.

"IEEE Richard W. Hamming Medal". IEEE. Retrieved Mar. 25, 2013.

J.A Heller. and I. M. Jacobs “Viterbi Decoding for Satellite and Space Communications,” IEEE Transactions on Communication Technology, vol. com-19, pp. 835-848, October 1971.

J. Gomes and B. K. Mishra. “double error correcting long code.” The International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.5,September 2010.

J. Rumánek And J. Šebesta. “New progressive channel coding methods for short message transmission via satellite transponder using predetermined length of frame,” Purkyňova Czech Republic VOL. 19, NO. 1, APRIL 2010

Larsen K.J. “Short Convolutional Codes with Maximal Free Distance for Rates1/2, 1/3, and 1/4,” IEEE Transactions on Information Theory, vol. IT-19, pp. 371-372, May 1973.

M. Sudan. “*Coding Theory* Meets *Theoretical Computer Science.*” SIAM News. [Online]. Volume 34, Number 10, Available: www.msri.org/people/members/sara/articles/coding [Mar. 26, 2013].

N. Zivic and C. Ruland. “Channel Coding as a Cryptography Enhancer.” [on-line]. Available: www.wseas.us/e-library/transactions/communications/.../30-578N.pdf [May 23, 2013].

NAE Members Directory - Dr. Richard W. Hamming". National Academy of Engineering. Retrieved February 5, 2011. Visited 26-03-2013 around 3:11am

Q. Zhao, P. Cosman and L. B. Milstein. “Tradeoffs of source coding, channel coding and spreading in cdma systems.” Available: code.ucsd.edu/cosman/conf-paper-34.pdf [Mar. 26, 2013].

R. Hentschke, F. Marques, F. Lima, L. Carro “Analyzing Area and Performance Penalty of Protecting Different Digital Modules with Hamming Code and Triple Modular Redundancy.” Proceedings of the 15th Symposium on Integrated Circuits and Systems Design (SBCCI’02) 0-7695-1807-9/02 $17.00 © 2002 IEEE.

R. W. Hamming. "Error Detection and Error Correction Codes" Bell Systems Tech. Journal, vol 29, pp 147-160, April, 1950

R. Hamming. (1998, Aug - Sep). Mathematics on a Distant Planet". The American Mathematical Monthly 105 (7): 640–650.

R W Hamming. “Calculus and Discrete Mathematics,” The College Mathematics Journal 15 (5) (1984), 388-389.

S. B. Wicker, (1995). Error Control Systems for Digital Communication and Storage. Englewood Cliffs NJ: Prentice-Hall. ISBN 0-13-200809-2.

S. Ghosh, S. Basu, and N. A. Touba. “Selecting Error Correcting Codes to Minimize Power in Memory Checker Circuits”, June 2013.

S. G. Wilson, (1996). Digital Modulation and Coding. Englewood Cliffs NJ: Prentice-Hall. ISBN 0-13-210071-1.

T. K. Moon,. (2005). Error Correction Coding. New Jersey: John Wiley & Sons. ISBN 978-0-471-64800-0.

Microsoft. "C++ and Visual Basic - a Short (and Partial) History. Internet: www.msdn.microsoft.com/en-us/library/aa227231%28v=vs.60%29.aspx [June 26, 2013].

W. E Ryan, S. Lin (2009). Channel Codes: Classical and Modern. Cambridge University Press. ISBN 978-0-521-84868-8.

Web site information from:

- https://secure.wikimedia.org/wikipedia/en/wiki/Forward_error_correction Visited on the [March 15, 2013] around 3:37am.
- .http://en.wikipedia.org/wiki/Richard_Hamming [May 10, 2013].
- http://biobio.loc.edu/chu/web/Courses/Cosi460/hamming_codes.htm [Mar. 26, 2013].
- en.wikipedia.org/wiki/Channel capacity
- en.wikipedia.org/wiki/Binary symmetric_channel
- en.wikipedia.org/wiki/Noisy-channel coding_theorem
- en.wikipedia.org/wiki/Shannon–Hartley_theorem
- en.pudn.com/downloads366/doc/project/detail1587620_en.html
- www.princeton.edu/~achaney/tmve/wiki100k/docs/Line_code.html
- www.andreas-schwope.de/ASIC_s/Schnittstellen/...Lines/line_codes.html