International Journal of Engineering Research and General Science Volume 4, Issue 4, July-August, 2016
ISSN 2091-2730

Design Single and Multiple Errors Correction Block Codes

Zahraa Raad Mayoof Hajiyat

Faculty of Engineering Technology & Built Environment,

Department of Electrical and Electronic Engineering,

B.Eng (Hons) Communication and Electronic Engineering

UCSI University, Kuala Lumpur, Malaysia

E-mail:glancing@ymail.com

Abstract— Channel coding is a process used in digital communications system to guarantee a transmission is received with minimal or no errors. Some coding methods that employed are achieved by involving additional binary numbers into the transmission. When decoded on the receiving finished, the transmission can be checked for errors that may have happened. Channel coding that apply this method are block codes which called forward error correction (FEC) codes that allow a limited number of errors to be detected and corrected without retransmission. Hamming codes has found a best class of single error correcting codes while Reed– Solomon codes has found a best class of multiple error correcting codes for designing of block codes and it shows how error correction and detection can be accomplished. Beside, using Matlab software to perform single and multiple errors correction design. Also, The BER display for both Hamming codes and Reed-Solomon codes.

Keywords— Communication systems, Errors correction, Channel coding, Block codes, Hamming codes, Reed-Solomon codes, BER display.

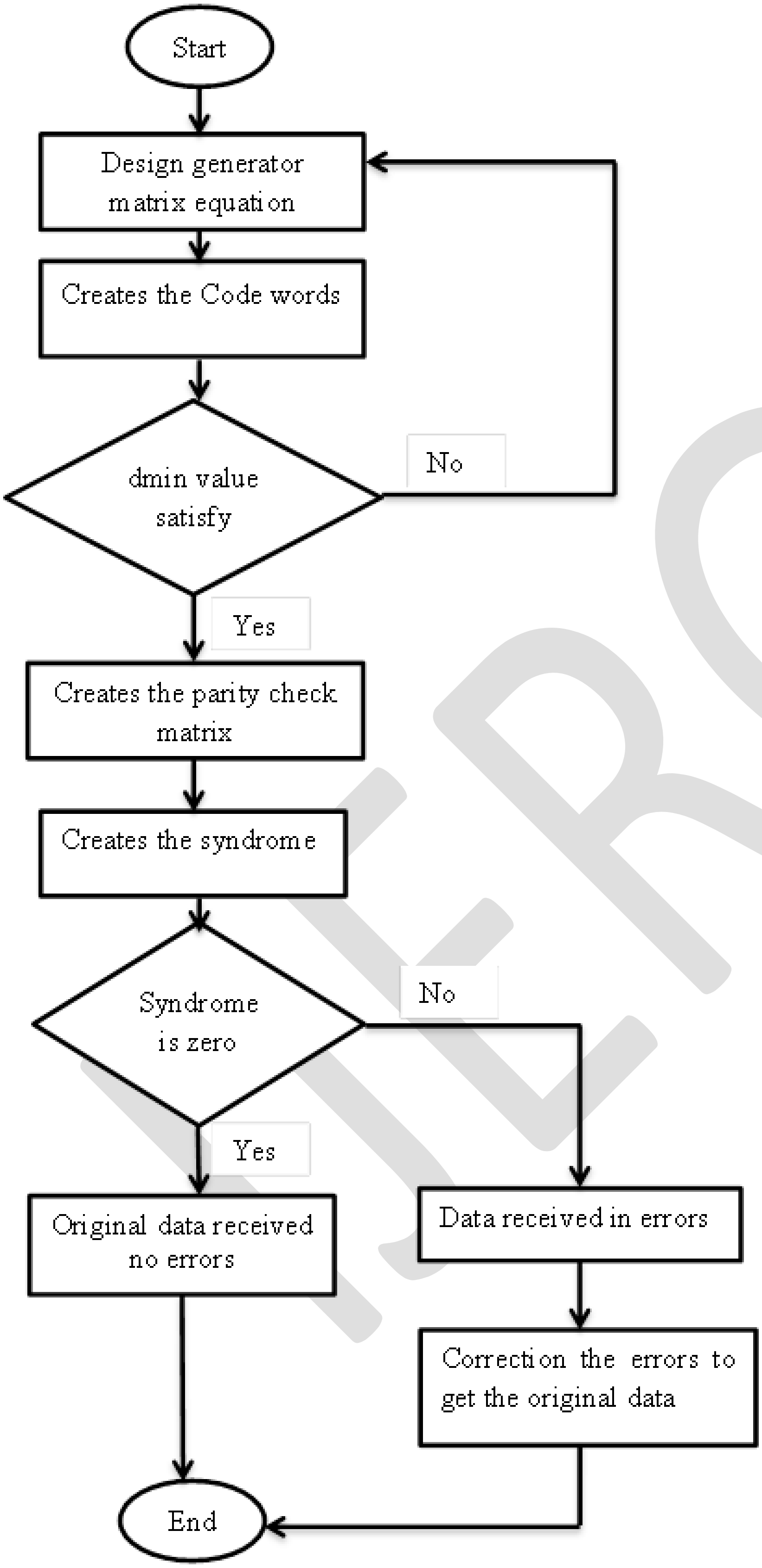
INTRODUCTION

Block code are forward error correction (FEC) codes that allow a limited number of errors to be detected and corrected without retransmission. Thus, block codes can be used to improve the performance of a communications system when increasing the transmitter power is impossible. The block codes have several categories such as Reed- Solomon codes, Hamming codes, Hadamard codes, Expander codes, Golay codes, and Reed–Muller codes. These categories also belong to the class of linear codes, and they are called linear block codes. Mostly, these codes are known as algebraic block codes, or cyclic block codes, because they can be produced using Boolean polynomials. In 1950, Richard Wesley Hamming found a best class of single error correcting codes. The code was used for long distance telephony. Hamming code is an improvement on the parity check method. Hamming is a well-known name in the field of data encoding and decoding. In 1960, Irving Reed and Gus Solomon published a class of error-correcting codes that are now called Reed-Solomon (R-S) codes. These codes have great control and effectiveness, and currently found in many applications from compact disc players to deep-space applications.

METHODOLOGY

The research methodology of design block codes that can correct single and multiple errors such as Hamming codes and Reed-Solomon codes are required to gathering all knowledge, concepts and theories of error detection, error correction, block codes, Hamming codes and Reed-Solomon codes theories. Besides, gathering the theories of block codes principles such as hamming distance, hamming weight, mini hamming distance, linear block codes and symmetric block codes. Furthermore, the generator matrix designs, codewords, parity check matrix and syndrome. When understanding the concepts of block codes, the design will apply these theories with develop the design to able to correct the errors in the transmitting bits messages.

The flow chart for designing block codes such as Hamming codes and Reed-Solomon codes as following:



RESULTS

(11, 4) Hamming code design:

The equation design:

C1= m1 ⊕ m2 ⊕ m3 ⊕ m4

C2=m1 ⊕ m2 ⊕ m3

C3=m1 ⊕ m3 ⊕ m4

C4=m1 ⊕ m2 ⊕ m4

C5=m2 ⊕ m3 ⊕ m4

C6= m1 ⊕ m4

C7=m1 ⊕ m3

This design of (11, 4) hamming code is for d_min =5 using Matlab software.

```
Command Window
>> g=[ 1 1 1 1 0 1 1;1 1 0 1 1 0 0;1 1 1 0 1 0 1;1 0 1 1 1 1 0]

g =

     1     1     1     1     0     1     1
     1     1     0     1     1     0     0
     1     1     1     0     1     0     1
     1     0     1     1     1     1     0

>> generator=[g,eye(4)]

generator =

     1     1     1     1     0     1     1     1     0     0     0
     1     1     0     1     1     0     0     0     1     0     0
     1     1     1     0     1     0     1     0     0     1     0
     1     0     1     1     1     1     0     0     0     0     1
```

```
>> m=[0 0 0 0;0 0 0 1;0 0 1 0;0 0 1 1;0 1 0 0;0 1 0 1;0 1 1 0;0 1 1 1;1 0 0 0;1 0 0 1;1 0 1 0;1 0 1 1;1 1 0 0;1 1 0 1; 1 1 1 0; 1 1 1 1

m =

     0     0     0     0
     0     0     0     1
     0     0     1     0
     0     0     1     1
     0     1     0     0
     0     1     0     1
     0     1     1     0
     0     1     1     1
     1     0     0     0
     1     0     0     1
     1     0     1     0
     1     0     1     1
     1     1     0     0
     1     1     0     1
     1     1     1     0
     1     1     1     1
```

```
>> codeword=m*generator

codeword =

     0     0     0     0     0     0     0     0     0     0     0
     1     0     1     1     1     1     0     0     0     0     1
     1     1     1     0     1     0     1     0     0     1     0
     2     1     2     1     2     1     1     0     0     1     1
     1     1     0     1     1     0     0     0     1     0     0
     2     1     1     2     2     1     0     0     1     0     1
     2     2     1     1     2     0     1     0     1     1     0
     3     2     2     2     3     1     1     0     1     1     1
     1     1     1     1     0     1     1     1     0     0     0
     2     1     2     2     1     2     1     1     0     0     1
     2     2     2     1     1     1     2     1     0     1     0
     3     2     3     2     2     2     2     1     0     1     1
     2     2     1     2     1     1     1     1     1     0     0
     3     2     2     3     2     2     1     1     1     0     1
     3     3     2     2     2     1     2     1     1     1     0
     4     3     3     3     3     2     2     1     1     1     1
```

```
>> p=g'

p =

     1     1     1     1
     1     1     1     0
     1     0     1     1
     1     1     0     1
     0     1     1     1
     1     0     0     1
     1     0     1     0

>> paritycheck=[eye(7),p]

paritycheck =

     1     0     0     0     0     0     0     1     1     1     1
     0     1     0     0     0     0     0     1     1     1     0
     0     0     1     0     0     0     0     1     0     1     1
     0     0     0     1     0     0     0     1     1     0     1
     0     0     0     0     1     0     0     0     1     1     1
     0     0     0     0     0     1     0     1     0     0     1
```

```
>> error=eye(11)

error =

     1     0     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     0     1
```

```
>> H=paritycheck'

H =

     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
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
     1     1     1     1     0     1     1
     1     1     0     1     1     0     0
     1     1     1     0     1     0     1
     1     0     1     1     1     1     0
```

```
>> syndrome=error*H

syndrome =

     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
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
     1     1     1     1     0     1     1
     1     1     0     1     1     0     0
     1     1     1     0     1     0     1
     1     0     1     1     1     1     0
```

```
>> r=[1   1   1   0   1   0   1   0   0   1   0] % no error

r =

   1   1   1   0   1   0   1   0   0   1   0

>> syndrome=r*H

syndrome =

   2   2   2   0   2   0   2
```

```
>> r=[1   1   1   0   1   0   1   0   1   1   0] % one error

r =

   1   1   1   0   1   0   1   0   1   1   0

>> syndrome=r*H

syndrome =

   3   3   2   1   3   0   2

>> orignal=r+[0   0   0   0   0   0   0   0   1   0   0]

orignal =

   1   1   1   0   1   0   1   0   2   1   0
```

```
>> r=[1   1   1   1   1   1   0   0   0   1   1] % two error

r =

   1   1   1   1   1   1   0   0   0   1   1

>> syndrome=r*H

syndrome =

   3   2   3   2   3   2   1

>> orignal=r+[0   0   0   0   0   0   0   0   0   1   0]

orignal =

   1   1   1   1   1   1   0   0   0   2   1
```

<u>The (7, 3) Reed-Solomon code design:</u>

<u>The generator polynomial:</u>

$g(x) = (x-\alpha)(x-\alpha^2))(x-\alpha^3)(x-\alpha^4)$

$g(x) \;=(x^2-(\alpha+\alpha^2)x+(\alpha.\alpha^2))(x^2-(\alpha^3+\alpha^4)x+(\alpha^3.\alpha^4))$

$g(x)=(x^2-\alpha^4x+\alpha^3)(x^2-\alpha^6x+\alpha^7)$

$g(x)=(x^2-\alpha^4x+\alpha^3)(x^2-\alpha^6x+\alpha^0)$

$g(x)=x^4-(\alpha^4+\alpha^6)x^3\;+(\alpha^3+\alpha^{10}+\alpha^0)x^2-(\alpha^4+\alpha^9)x+\alpha^3$

$g(x)=x^4-\alpha^3\,x^3+\alpha^0\,x^2-\alpha^1\,x+\alpha^3$

$g(x) = \alpha^3+\alpha^1\,x+\alpha^0\,x^2+\alpha^3\,x^3+x^4$

Adopt the three message bits $010 \rightarrow \alpha^1$,$110 \rightarrow \alpha^3$,$111 \rightarrow \alpha^5$.

So, the message bits sequence is

: $\alpha^1+\alpha^3x+\alpha^5x^2 = x^4(\alpha^1+\alpha^3x+\alpha^5x^2) = \alpha^1x^4+\alpha^3x^5+\alpha^5x^6$

International Journal of Engineering Research and General Science Volume 4, Issue 4, July-August, 2016
ISSN 2091-2730

$$\alpha^5x^2 + \alpha^0x + \alpha^4$$

$$\alpha^3 + \alpha^1 x + \alpha^0 x^2 + \alpha^3 x^3 + x^4 \;\big)\; \alpha^5x^6 + \alpha^3x^5 + \alpha^1x^4$$

$$\alpha^5x^6 + \alpha^1x^5 + \alpha^5x^4 + \alpha^6x^3 + \alpha^1x^2$$

$$\alpha^0x^5 + \alpha^6x^4 + \alpha^6x^3 + \alpha^1x^2$$

$$\alpha^0x^5 + \alpha^3x^4 + \alpha^0x^3 + \alpha^1x^2 + \alpha^3x$$

$$\alpha^4x^4 + \alpha^2x^3 + \alpha^3x$$

$$\alpha^4x^4 + \alpha^0x^3 + \alpha^4x^2 + \alpha^5x + \alpha^0$$

$$\alpha^6x^3 + \alpha^4x^2 + \alpha^2x + \alpha^0$$

The codeword polynomial is:

$$U(X) = \alpha^0 + \alpha^2x + \alpha^4x^2 + \alpha^6x^3 + \alpha^1x^4 + \alpha^3x^5 + \alpha^5x^6$$

$$U(X) = (100) + (001)x + (011)x^2 + (101)x^3 + (010)x^4 + (110)x^5 + (111)x^6$$

Through transmission, this codeword becomes ruined so that assumes two symbols are received in error.

$$e(X) = 0 + 0x + 0x^2 + \alpha^2x^3 + \alpha^5x^4 + 0x^5 + 0x^6$$

$$e(X) = (000) + (000)x + (000)x^2 + (001)x^3 + (111)x^4 + (000)x^5 + (000)x^6$$

The received codeword with errors:

$$U(X) = (100) + (001)x + (011)x^2 + (101)x^3 + (010)x^4 + (110)x^5 + (111)x^6$$

$$+$$

$$e(X) = (000) + (000)x + (000)x^2 + (001)x^3 + (111)x^4 + (000)x^5 + (000)x^6$$

$$r(x) = (100) + (001)x + (011)x^2 + (100)x^3 + (101)x^4 + (110)x^5 + (111)x^6$$

$$r(x) = \alpha^0 + \alpha^2x + \alpha^4x^2 + \alpha^0x^3 + \alpha^6x^4 + \alpha^3x^5 + \alpha^5x^6$$

Checking the syndrome values of received codeword:

$$s_1 = e(\alpha^1) \qquad s_2 = e(\alpha^2)$$

$$s_1 = \alpha^2(\alpha^1)^3 + \alpha^5(\alpha^1)^4 \qquad s_2 = \alpha^2(\alpha^2)^3 + \alpha^5(\alpha^2)^4$$

$$s_1 = \alpha^5 + \alpha^9 \qquad s_2 = \alpha^8 + \alpha^{13}$$

$$s_1 = \alpha^5 + \alpha^2 \qquad s_2 = \alpha^1 + \alpha^6$$

$$s_1 = \alpha^3 \qquad s_2 = \alpha^5$$

$s_3 = e(\alpha^3)$

$s_3 = \alpha^2(\alpha^3)^3 + \alpha^5(\alpha^3)^4$

$s_3 = \alpha^{11} + \alpha^{17}$

$s_3 = \alpha^4 + \alpha^3$

$s_3 = \alpha^6$

$s_4 = e(\alpha^3)$

$s_4 = \alpha^2(\alpha^4)^3 + \alpha^5(\alpha^4)^4$

$s_4 = \alpha^{14} + \alpha^{21}$

$s_4 = \alpha^0 + \alpha^0$

$s_4 = 0$

The syndrome s$\neq$ 0.

The correction errors of received codeword:

$$r(x) = (100) + (001)x + (011)x^2 + (100)x^3 + (101)x^4 + (110)x^5 + (111)x^6$$

$$+$$

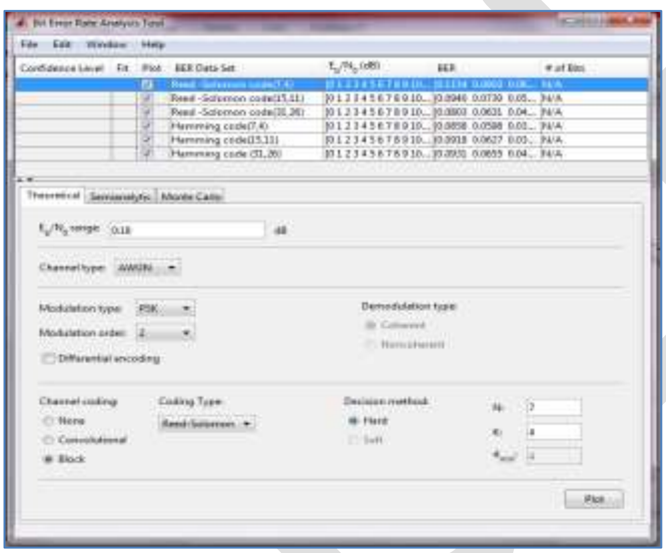
$$e(X) = (000) + (000)x + (000)x^2 + (001)x^3 + (111)x^4 + (000)x^5 + (000)x^6$$

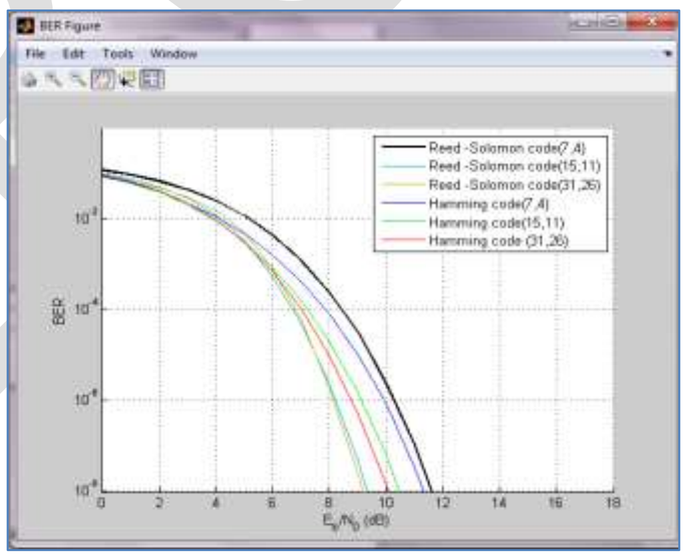
$$U(X) = (100) + (001)x + (011)x^2 + (101)x^3 + (010)x^4 + (110)x^5 + (111)x^6$$

$$U(X) = \alpha^0 + \alpha^2 x + \alpha^4 x^2 + \alpha^6 x^3 + \alpha^1 x^4 + \alpha^3 x^5 + \alpha^5 x^6$$

The BER display for Both Hamming codes and Reed –Solomon codes:

The plot results of BER of Hamming codes and Reed-Solomon codes as following:





CONCLUSION

In conclusion, the coding theory was introduced for the detection and correction of error in data. Hamming introduced a coding system for the error detection as well as correction of that error at the receiver side. In this procedure, Hamming added some extra bits to the data bits for error detection as well as correction of that error. Also, Reed-Solomon codes methods generally apply same procedure for detection and correction with using the algorithm of Finite field. Currently, interleaving method used to increase the performance of block codes such as Reed-Solomon codes and Hamming codes for detection and correction the errors. Also, the lower probability of bit error rate or BER when the input data is ruined by noise is preferred for receiver side for most of application. As results, the Reed-Solomon code is achieved lower BER compared to Hamming code for same (n, k) block code. In the end, Hamming codes and Reed-Solomon codes is a famous designation in the field of data encoding and decoding in the communication systems.

REFERENCES:

[1] Gallager, R.G., *Information theory and reliable communication*. Vol. 2. 1968: Springer.

[2] Wicker, S.B., *Error control systems for digital communication and storage*. Vol. 1. 1995: Prentice hall Englewood Cliffs.

[3] Rappaport, T.S., *Wireless communications: principles and practice*. Vol. 2. 1996: prentice hall PTR New Jersey.

[4] Stallings, W., *Data and computer communications*. 2007: Pearson/Prentice Hall.

[5] Costello, D. and S. Lin, *Error control coding.* New Jersey, 2004.

[6] Reed, I.S. and X. Chen, *Error-control coding for data networks.* Vol. 508. 2012: Springer Science & Business Media.

[7] Tarokh, V., H. Jafarkhani, and A.R. Calderbank, *Space-time block codes from orthogonal designs.* Information Theory, IEEE Transactions on, 1999. 45(5): p. 1456-1467.

[8] Vanstone, S.A. and P.C. Van Oorschot, *An introduction to error correcting codes with applications*. Vol. 71. 2013: Springer Science & Business Media.

[9] Sklar, B., *Digital communications:Fundamentals and Applications*, ed. S. Edition. 2001: Prentice Hall

[10] Furutani, K., et al., *A built-in Hamming code ECC circuit for DRAMs.* Solid-State Circuits, IEEE Journal of, 1989. 24(1): p. 50-56.

[11] Peterson, W.W. and E.J. Weldon, *Error-correcting codes*. 1972: MIT press.

[12] Reed, I.S. and G. Solomon, *Polynomial codes over certain finite fields.* Journal of the society for industrial and applied mathematics, 1960. 8(2): p. 300-304.