Encoding and Decoding of Narrow-Sense binary BCH code

Kevin Selva Prasanna (EE14B028) and Srivatsan Ramesh (EE14B058)

***Abstract*— This electronic document is a documentation of the experiments performed on our implementation of the encoder and decoder of a narrow-sense Bose Chaudhuri Hocquenghem code with a code length of $n = 127$ and a designed code distance of $\delta = 15$.**

I. INTRODUCTION

Bose Chaudhuri Hocquenghem codes or **BCH codes** form a class of cyclic error-correcting codes that are constructed using polynomials over a finite field. One of the key features of BCH codes is that during code design, there is a precise control over the number of symbol errors correctable by the code. In particular, it is possible to design binary BCH codes that can correct multiple bit errors.

II. CONSTRUCTION OF CODE

A. Design Parameters

The BCH code designed has a code length $n = 127$ and the minimum distance as $\delta = 15$. The decoder can correct upto w errors and e erasures where

$$2w + e < \delta$$

B. Generator Polynomial

The generator polynomial for a narrow-sense $t-error-correcting$ BCH code is given by

$$g(x) = LCM(\phi_1(x), \phi_2(x), ..., \phi_{2t}(x))$$

where $\phi_1(x), \phi_2(x), ..., \phi_{2t}(x)$ are the minimal polynomials of $\alpha, \alpha^2, \alpha^3, ..., \alpha^{2t}$ respectively and α is a primitive element of the finite field $\mathbb{F}_{128}$ generated with the polynomial $x^7 + x^3 + 1$.
Here, $g(x)$ can be simply obtained by taking the LCM of minimal polynomials of only the odd powers of α.

$$g(x) = LCM(\phi_1(x), \phi_3(x), ..., \phi_{2t-1}(x))$$

For the case of $\mathbb{F}_{128}$ generated with the polynomial $x^7 + x^3 + 1$ we get

$$\begin{aligned} g(x) = 1 &+ x^2 + x^3 + x^6 + x^{12} + x^{13} + x^{15} + x^{19} + x^{20} \\ &+ x^{22} + x^{23} + x^{24} + x^{28} + x^{39} + x^{40} + x^{43} + x^{46} \\ &+ x^{47} + x^{49} \end{aligned}$$

C. Parity Check Matrix

The parity check matrix is given by

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{2^m-2} \\ 1 & \alpha^2 & (\alpha^2)^2 & \dots & (\alpha^2)^{2^m-2} \\ 1 & \alpha^3 & (\alpha^3)^2 & \dots & (\alpha^3)^{2^m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \dots & (\alpha^{2t})^{2^m-2} \end{bmatrix}$$

D. Systematic Encoding

Systematic encoding of the messages were done using the expression

$$c(x) = m(x)x^{n-k} + r(x)$$

where $r(x) = m(x) \bmod g(x)$

E. Decoding

The most important and interesting part of BCH codes is decoding. We get a set of equations called the $Newton\ Identities$ whose solution gives the coefficients of the error location polynomial. We use Berlekamp Massey Algorithm to solve these set of equations to get the coefficients of the error locator polynomial.
For correcting a received word which has both errors and erasures we use an easy technique. We replace all the erasures with 0 and correct the errors, then replace all the erasures with 1 and correct the errors. We pick the error with minimum weight as the maximum likelihood error and use it to correct the received vector by adding the error to the received word.
Berlekamp Massey algorithm will sometime not give a correct error locator polynomial when the number of errors goes beyond 7, more specifically when the word with error doesn't lie inside any hamming sphere of radius 7 with the codewords as center. Sometimes, the polynomial returned by the algorithm will have repeated roots which also denotes an error in the error locator polynomial.

III. WORKING EXAMPLES

All elements of GF(128) generated by the primitive polynomial x^7+x^3+1 are expressed in their exponential form as powers of their primitive element unless and otherwise specified.

A. Case 1: No errors

Received Codeword 127-D vector
0001010111111001100100101010111010101100011001
0101101100111011010011110111110101000011010001
100011101100010101011111100010101000

Syndrome 14-D vector
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

The decoding table for the BCH code

k	σ	d	l_k	$2k-l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	-1	0	0
1	0 -1 -1 -1 -1 -1 -1 -1	-1	1	1
2	0 -1 -1 -1 -1 -1 -1 -1	-1	1	3
3	0 -1 -1 -1 -1 -1 -1 -1	-1	1	5
4	0 -1 -1 -1 -1 -1 -1 -1	-1	1	7
5	0 -1 -1 -1 -1 -1 -1 -1	-1	1	9
6	0 -1 -1 -1 -1 -1 -1 -1	-1	1	11
7	0 -1 -1 -1 -1 -1 -1 -1	-1	1	13

Error locations Null vector
1x0 empty row vector

Decoded codeword 127-D vector
0001010111111001100100101010111010101100011001
0101101100111011010011110111110101000011010001
1000111011000101010111111000

B. Case 2: 4 errors

Received vector 127-D vector
11100101111110011001001010101110101011000110010
10110110011101101001111011111010100001101000110
0011101100010101011111100010101000

Syndrome 14-D vector
93 59 21 118 119 42 9 109 5 111 102 84 19 18

Error locations 4-D vector
0 1 2 3

The decoding table for the BCH code

k	σ	d	l_k	$2k-l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	93	0	0
1	0 93 -1 -1 -1 -1 -1 -1	18	1	1
2	0 93 52 -1 -1 -1 -1 -1	8	2	2
3	0 93 119 83 -1 -1 -1 -1	89	3	3
4	0 93 39 96 6 -1 -1 -1	-1	4	4
5	0 93 39 96 6 -1 -1 -1	-1	4	6
6	0 93 39 96 6 -1 -1 -1	-1	4	8
7	0 93 39 96 6 -1 -1 -1	-1	4	10

Decoded codeword 127-D vector
00010101111110011001001010101110101011000110010
10110110011101101001111011111010100001101000110
0011101100010101011111000

C. Case 3: 8 errors

Received vector 127-D vector
11101010111110011001001010101110101011000110010
10110110011101101001111011111010100001101000110
0011101100010101011111100010101000

Syndrome 14-D vector
90 53 49 106 66 98 21 85 54 5 111 69 2 42

The decoding table for the BCH code

k	σ	d	l_k	$2k-l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	90	0	0
1	0 90 -1 -1 -1 -1 -1 -1	100	1	1
2	0 90 10 -1 -1 -1 -1 -1	65	2	2
3	0 90 15 55 -1 -1 -1 -1	101	3	3
4	0 90 55 23 46 -1 -1 -1	86	4	4
5	0 90 94 6 105 40 -1 -1	115	5	5
6	0 90 96 125 124 68 75 -1	69	6	6
7	0 90 17 103 91 47 47 121	-1	7	7

Result
Failed

Reported Reason for error
Number of errors is beyond the correctable limit w>t

D. Case 4: 10 erasures

Received vector 127-D vector
22222222221110011001001010101110101011000110010
10110110011101101001111011111010100001101000110
0011101100010101011111100010101000

Trial 1: Substituting Erasures with 0:

Syndrome 14-D vector
115 103 14 79 65 28 108 31 28 3 29 56 67 89

The decoding table for the BCH code

k	σ	d	l_k	$2k - l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	115	0	0
1	0 115 -1 -1 -1 -1 -1 -1	9	1	1
2	0 115 21 -1 -1 -1 -1 -1	118	2	2
3	0 115 39 97 -1 -1 -1 -1	4	3	3
4	0 115 68 98 34 -1 -1 -1	66	4	4
5	0 115 76 35 99 32 -1 -1	-1	5	5
6	0 115 76 35 99 32 -1 -1	-1	5	7
7	0 115 76 35 99 32 -1 -1	-1	5	9

Error locations 4-D vector
3 7 8 5 9

Trial 2: Substituting Erasures with 1:

Syndrome 14-D vector
76 25 60 50 24 120 41 100 40 48 52 113 99 82

The decoding table for the BCH code

k	σ	d	l_k	$2k - l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	76	0	0
1	0 76 -1 -1 -1 -1 -1 -1	126	1	1
2	0 76 50 -1 -1 -1 -1 -1	44	2	2
3	0 76 0 121 -1 -1 -1 -1	103	3	3
4	0 76 81 0 109 -1 -1 -1	122	4	4
5	0 76 21 71 119 13 -1 -1	-1	5	5
6	0 76 21 71 119 13 -1 -1	-1	5	7
7	0 76 21 71 119 13 -1 -1	-1	5	9

Error locations 4-D vector
0 1 2 4 6

Decoded codeword 127-D vector
00010101111110011001001010101110101011000110010
10110110011101101001111011111010100001101000110
0011101100010101011111100010101000

E. Case 5: 16 erasures

Received vector 127-D vector
22222222222222222201001010101110101011000110010
10110110011101101001111011111010100001101000110
0011101100010101011111100010101000

Trial 1: Substituting Erasures with 0:

Syndrome 14-D vector
108 89 43 51 58 86 107 102 21 116 4 45 76 87

The decoding table for the BCH code

k	σ	d	l_k	$2k - l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	108	0	0
1	0 108 -1 -1 -1 -1 -1 -1	33	1	1
2	0 108 52 -1 -1 -1 -1 -1	81	2	2
3	0 108 45 29 -1 -1 -1 -1	48	3	3
4	0 108 81 7 19 -1 -1 -1	14	4	4
5	0 108 109 72 5 122 -1 -1	72	5	5
6	0 108 25 123 8 104 77 -1	26	6	6
7	0 108 49 37 34 13 121 76	-1	7	7

Result
Trial Failed

Trial 2: Substituting Erasures with 1:

Syndrome 14-D vector
24 48 82 96 39 37 69 65 94 78 43 74 82 11

The decoding table for the BCH code

k	σ	d	l_k	$2k - l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	24	0	0
1	0 24 -1 -1 -1 -1 -1 -1	109	1	1
2	0 24 85 -1 -1 -1 -1 -1	115	2	2
3	0 24 70 30 -1 -1 -1 -1	48	3	3
4	0 24 97 10 18 -1 -1 -1	94	4	4
5	0 24 13 8 90 76 -1 -1	97	5	5
6	0 24 40 114 0 28 21 -1	76	6	6
7	0 24 81 102 113 53 6 55	-1	7	7

Result
Trial Failed

Reported reason for error
Number of errors and erasures is beyond correctable limit 2w+e>2t

F. *Case 6: 6 errors and 8 erasures*

Received vector 127-D vector
222222220000001011001001010101110101011000110010
1011011001110110100111101111101010000110100 0110
0011101100010101011111100010101000

Trial 1: Substituting Erasures with 0:

Syndrome 14-D vector
4 8 24 16 31 48 53 32 93 62 53 96 72 106

The decoding table for the BCH code

k	σ	d	l_k	$2k - l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	4	0	0
1	0 4 -1 -1 -1 -1 -1 -1	40	1	1
2	0 4 36 -1 -1 -1 -1 -1	68	2	2
3	0 4 22 32 -1 -1 -1 -1	45	3	3
4	0 4 27 75 13 -1 -1 -1	17	4	4
5	0 4 125 87 100 4 -1 -1	99	5	5
6	0 4 31 117 17 59 95 -1	7	6	6
7	0 4 28 26 5 74 116 39	-1	7	7

Result
Trial Failed

Trial 2: Substituting Erasures with 1:

Syndrome 14-D vector
29 58 110 116 22 93 124 105 111 44 1 59 111 121

The decoding table for the BCH code

k	σ	d	l_k	$2k - l_k$
-0.5	0 -1 -1 -1 -1 -1 -1 -1	0	0	-1
0	0 -1 -1 -1 -1 -1 -1 -1	29	0	0
1	0 29 -1 -1 -1 -1 -1 -1	76	1	1
2	0 29 47 -1 -1 -1 -1 -1	78	2	2
3	0 29 52 31 -1 -1 -1 -1	39	3	3
4	0 29 101 56 8 -1 -1 -1	85	4	4
5	0 29 72 35 108 77 -1 -1	104	5	5
6	0 29 92 126 9 10 27 -1	72	6	6
7	0 29 99 59 10 6 63 45	-1	7	7

Result
Trial Failed

Reported reason for error
Number of errors and erasures is beyond correctable limit 2w+e>2t

IV. RESULTS

The code thus successfully performs encoding and decoding of BCH codes in all possible cases. It is capable of correcting errors and erasures whenever theoretically possible and reports decoding error and the reason when the number of erasures and errors exceeds the decodable limit.

V. CONCLUSION

BCH codes are very powerful. They provide us a handle on the number of errors we want to be able to correct and are a very important class of codes.

REFERENCES

[1] Channel Codes - Classical and Modern WILLIAM E. RYAN University of Arizona, SHU LIN University of California, Davis