

On Error Correcting Codes and Their Applications



By

Asma Shaheen Ansari

Department of Mathematics
Quaid-i-Azam University, Islamabad
PAKISTAN
2017

On Error Correcting Codes and Their Applications



By

Asma Shaheen Ansari

Supervised by

Prof. Dr. Tariq Shah

Department of Mathematics
Quaid-i-Azam University, Islamabad
PAKISTAN
2017

On Error Correcting Codes and Their Applications

By

Asma Shaheen Ansari

*A Thesis
Submitted in the Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
IN
MATHEMATICS*

Supervised by

Prof. Dr. Tariq Shah

Department of Mathematics
Quaid-i-Azam University, Islamabad
PAKISTAN
2017

Author's Declaration

I **Asma Shaheen Ansari** hereby state that my PhD thesis titled **On Error Correcting Codes and Their Applications** is my own work and has not been submitted previously by me for taking any degree from the Quaid-i-Azam University Islamabad, Pakistan or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my graduate the university has the right to withdraw my PhD degree.

Name of Student:  **Asma Shaheen Ansari**

Date: **14-03-2017**

Plagiarism Undertaking

I solemnly declare that research work presented in the thesis titled "**On Error Correcting Codes and Their Applications**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and **Quaid-i-Azam University** towards plagiarism. Therefore, I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even afterward of PhD degree, the University reserves the rights to withdraw/revoke my PhD degree and that HEC and the University has the right to publish my name on the HEC/University Website on which names of students are placed who submitted plagiarized thesis.

Student/Author Signature:



Name: **Asma Shaheen Ansari**

On Error Correcting Codes and Their Applications

By

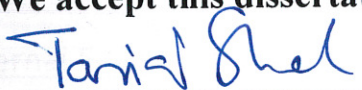
Asma Shaheen Ansari

CERTIFICATE

A DISSERTATION SUBMITTED IN THE PARTIAL FULFILMENT
OF THE REQUIRMENT FOR THE DEGREE OF DOCTOR OF
PHILOSOPHY

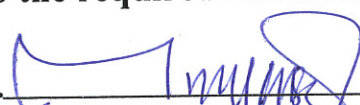
We accept this dissertation as conforming to the required standard.

1.



Prof. Dr. Tariq Shah
(Supervisor)

2.



Prof. Dr. Muhammad Yousaf Malik
(Chairman)

3.



Dr. Akbar Azam
Professor
Department of Mathematics
COMSATS CIIT
Chak Shahzad, Islamabad.
(External Examiner)

4.



Dr. Matloob Anwar
Assistant Professor
Department of Mathematics
School of Natural Sciences
NUST, H-12 Islamabad
(External Examiner)

**Department of Mathematics
Quaid-i-Azam University
Islamabad, Pakistan
2017**

Certificate of Approval

This is to certify that the research work presented in this thesis entitled **On Error Correcting Codes and Their Applications**, was conducted by **Mrs. Asma Shaheen Ansari** under the supervision of **Prof. Dr. Tariq Shah**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the Department of Mathematics, Quaid-i-Azam University, Islamabad in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Field of Mathematics from Department of Mathematics, Quaid-i-Azam University Islamabad, Pakistan.

Student Name: **Asma Shaheen Ansari**: Signature:

External committee:

a) External Examiner 1: Signature:

Name: **Dr. Akbar Azam**

Designation: Professor

Office Address: CIIT, Park Road, Chak Shahzad, Islamabad.



b) External Examiner 2: Signature:

Name: **Dr. Matloob Anwar**

Designation: Assistant Professor

Office Address: NUST, Islamabad



c) Internal Examiner: Signature:

Name: **Dr. Tariq Shah**

Designation: Professor

Office Address: Department of Mathematics, QAU, Islamabad.



d) Supervisor Name: Signature:

Prof. Dr. Tariq Shah



e) Name of Dean/ HOD Signature:

Prof. Dr. Muhammad Yousaf Malik



List of Papers

1. Asma Shaheen Ansari and Tariq Shah, An association between primitive and non-primitive BCH codes using monoid rings, EURASIP journal on wireless communications and networking, Springer, DOI 10.1186/s13638-016-0526-3.
2. Asma Shaheen Ansari and Tariq Shah, Cyclic codes as Ideals in $F_2[x; a\mathbb{Z}_{\geq 0}]_{\{n\}}$, $F_2[x]_{an}$, $F_2[x; (a/b)\mathbb{Z}_{\geq 0}]_{bn}$, and $F_2[x; (1/b)\mathbb{Z}_{\geq 0}]_{\{abn\}}$, U.P.B. Sci. Bull., Series A, 78(3), 2016.
3. Asma Shaheen Ansari and Tariq Shah, Application of BCH codes in DNA formation (Submitted).
4. Tariq Shah, Syed Azmat Hussain and Asma Shaheen Ansari, Bandwidth apportionment in Cognitive Radio: Multiple Forward Transmission via BCH-codes. (Submitted)
5. Tariq Shah, Asma Shaheen Ansari, Riffat Perveen, Anam Kazmi and Antonio Aparecido de Andrade⁵. Primitive to non-primitive BCH codes: An instantaneous path shifting scheme for data transmission. (Submitted).
6. Asma Shaheen Ansari, Tariq Shah and Zia-ur-Rahman, An algorithm bridges a primitive and a sequence of non-primitive BCH codes. (Submitted).

**Dedicated to my my beloved
mother (late), my great
father (Shaheen Iqbal), my
supporting husband and my
beloved sons Rayyan and
Yahya.**

Acknowledgement

All praises to my Lord Allah the Almighty, who has given me the opportunity to do this work and help me in every thick and thin. Endless durood on my beloved Prophet Hazrat Muhammad (S.A.A.W), whose words are an inspiration in every step of my life.

I would like to express my sincere gratitude to my supervisor Prof. Tariq Shah for the continuous support of my Ph.D study and research, for his motivation, enthusiasm, and immense knowledge. His patience and support helped me overcome many crisis situations and finish this dissertation.

I also benefited from the Indigenous Ph.D. Fellowship Program, for which I am thankful to Higher Education Commission, Islamabad.

I also want to thank Riffat Parveen, Zia-ur-Rehman and Syed Azmat Husain who have helped me in this research. I would like to show gratitude to all of my friends; Ayesha, Munazza, Shafaq, Sumaira, Rabia and especially Sadia for being so nice, helpful and supportive throughout this work. I would like to thank my colleagues, seniors and juniors for lending a helping hand when in need.

Finally, and most importantly a very special thanks to my family. I am indebted to my pious and great mother Mrs. Najma Shaheen (late), because of whom what I am. I greatly thank my dearest father Mr. Shaheen Iqbal Ansari who always stood by me and provided me with encouragement and moral support. I am really very much grateful to my beloved husband and his whole family (especially my mother in law and father in law) without their support this thesis wouldn't be accomplished. My husband's endless love, support, patience and care helped me through every thick and thin. He was always there in every time I loose hope. Special thanks to my sweet sisters (Uzma, Mahum, Anum and Maryam) and brothers (Umer and Owais) for their prayers and moral support. In the end i would like to thank my cute little sons Rayyan and Yahya for giving me their time and for being so nice to me I love u both. This thesis would not have been possible without my wonderful families and caring friends.



Asma Shaheen Ansari

2016

Abstract

To expand the use of codes and provide a form of error-correction, it is useful to extend the use of binary streams into another representation. In this work, we have used different monoid rings for the construction of a new family of error correcting codes having better error correction capability. Initially we have constructed binary cyclic codes using monoid rings instead of polynomial ring. For an n length binary cyclic code, three different binary cyclic codes of length an , bn and abn are obtain. These codes are interleaved codes capable of correcting burst of errors alongwith random error correction.

The BCH codes form a class of parameterized error-correcting codes which have been the subject of interest. Instead of primitive BCH codes we have showed the existence of non-primitive BCH codes of length bn over the fields F_2 , F_4 and finite rings \mathbb{Z}_{2^m} along with their applications. The value of b is investigated for which the existence of the non-primitive BCH code C_{bn} is assured. It is noticed that the code C_n is embedded in the code C_{bn} . Therefore, the data transmitted by the code C_n can also be transmitted by the code C_{bn} . The BCH codes C_{bn} have better error correction capability whereas the BCH code C_n has better code rate.

Abstract

To expand the use of codes and provide a form of error-correction, it is useful to extend the use of binary streams into another representation. In this work, we have used different monoid rings for the construction of a new family of error correcting codes having better error correction capability. Initially we have constructed binary cyclic codes using monoid rings instead of polynomial ring. For an n length binary cyclic code, three different binary cyclic codes of length an , bn and abn are obtain. These codes are interleaved codes capable of correcting burst of errors alongwith random error correction.

The BCH codes form a class of parameterized error-correcting codes which have been the subject of interest. Instead of primitive BCH codes we have showed the existence of non-primitive BCH codes of length bn over the fields F_2 , F_4 and finite rings \mathbb{Z}_{2^m} along with their applications. The value of b is investigated for which the existence of the non-primitive BCH code C_{bn} is assured. It is noticed that the code C_n is embedded in the code C_{bn} . Therefore, the data transmitted by the code C_n can also be transmitted by the code C_{bn} . The BCH codes C_{bn} have better error correction capability whereas the BCH code C_n has better code rate.

Contents

1	Introduction	2
1.1	Algebraic notions and algebraic codes	5
1.2	Algebraic notions	5
1.2.1	Semigroup	5
1.2.2	Ring	6
1.2.3	Rings of formal power series	8
1.2.4	Semigroup ring	10
1.2.5	Galois fields and Galois rings	10
1.3	Algebraic codes	11
1.3.1	Fundamentals of coding theory	11
1.3.2	Linear codes	13
1.3.3	Cyclic codes	15
1.3.4	Bose-Chaudhuri-Hocquenghem codes (BCH codes)	17
1.3.5	Decoding algorithms	18
2	Cyclic codes as ideals in $\mathbb{F}_2[x; a\mathbb{N}_0]_n$, $\mathbb{F}_2[x]_{an}$, $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ and $\mathbb{F}_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$	22
2.1	Cyclic codes as ideals in $\mathbb{F}_2[x; a\mathbb{N}_0]_n$	23
2.2	Cyclic codes as ideals in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$	28
2.3	Relationship among cyclic codes C_n, C_{an}, C_{bn} and C_{abn}	34
2.3.1	Relationship of C_n, C_{an}, C_{bn} and C_{abn} by interleaving	35
2.3.2	Relationship of C_n, C_{an}, C_{bn} and C_{abn} by generator and parity check matrices	36

2.4	Decoding procedure	44
3	Construction of non-primitive BCH codes using monoid rings	51
3.1	BCH code C_n as ideal in $F_2[x; a\mathbb{N}_0]_n$	52
3.2	BCH codes as ideals in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$	54
4	Family of non-primitive BCH codes	63
4.1	BCH codes as ideal in $F_2[x; \frac{a}{b^j}\mathbb{N}_0]_{b^j n (1 \leq j \leq m)}$	63
4.2	Link between primitive and a family of non-primitive BCH codes	74
4.2.1	General Decoding Principle	78
4.3	The Algorithm	83
4.3.1	Encoding of non-primitive BCH code of length $b^j n$	83
4.3.2	Error correction in received polynomial (Decoding)	88
5	Construction of non-primitive BCH codes over the field F_4	99
5.1	BCH-codes as Ideal in the ring $F_4[x; a\mathbb{N}_0]_n$	99
5.2	BCH-codes as Ideal in the ring $F_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}$	101
5.3	Primitive BCH code C_n and non-primitive BCH code C_{bn} : A link	109
5.4	General decoding principle	110
6	Non-primitive BCH codes over Galois rings	118
6.1	BCH-codes as ideals in $\mathbb{Z}_{2^m}[x; a\mathbb{N}_0]_n$ and $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]_{b^j n}$	118
6.2	Relation between the sequences $\{C_{b^j n}\}_{j \geq 0}$ and $\{C'_{b^j n}\}_{j \geq 0}$ of BCH codes	124
6.3	Decoding procedure	130
7	Applications	138
7.1	Application in cognitive radio	138
7.1.1	Bandwidth limitations	139
7.1.2	Multiple forward transmission through embedded BCH codes	140
7.1.3	CRFTM for BCH codes	147
7.2	Application in DNA formation	156
7.3	Application in data transformation	160

7.4 Summary	162
8 Conclusion	164

Chapter 1

Introduction

Error-correcting codes are one of the most effective and widely applied branch of abstract algebra over the last sixty years. It forms the basis of modern communication systems and is used in essentially all hardware level implementations of smart and intelligent machines, such as scanners, optical devices, and telecom equipment. It is due to the error-correcting codes that we are able to communicate over long distances and are able to achieve megabit bandwidth over a wireless communication channel.

One of the important class of error-correcting codes is cyclic codes. These codes were initially studied by Prange ([30], [31]). Since then, advancement in the theory of cyclic codes for correcting random as well as burst errors has been encouraged by many coding theorists. The cyclic codes were first studied over the binary field F_2 . Then were extended to the prime field F_p and its Galois field extension F_q , where p is a prime integer and q is p^m with m a positive integer. The correspondence of cyclic codes with ideals was observed independently by Peterson [28] and Kasami [19].

An important class of cyclic codes are binary Hamming codes. They were discovered by R. W. Hamming and M. J. E. Golay. Hamming represent a family of binary linear error-correcting codes that can detect up to two errors and correct one error. They have interesting properties and are easy to encode and decode.

In [16], Hocquenghem and in [7], Bose and Ray-Chaudhuri independently developed the large class of error correcting codes named as BCH codes. These codes are a remarkable generalization of the Hamming codes for correcting multiple-errors. 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. Another advantage of BCH codes is the ease with which they can be decoded via an algebraic method known as syndrome decoding.

The extension of a BCH code embedded in a semigroup ring was first discussed by Cazaran [9]. A great amount of information regarding rings construction and its corresponding polynomial codes are discussed in [24]. In [21], [20] and [22], the authors explained the extensions of BCH codes in many ring constructions where the outcomes are the special case of semigroup rings. In a series of papers [4], [34], [35], [36], [37], [38], [39] several classes of cyclic codes over a finite unitary commutative ring are constructed, through monoid rings. The purpose of these constructions is to address the error correction and the code rate trade off in a smart way.

In [40], Shah et al. showed the existence of a binary cyclic code of length $(n + 1)n$ corresponding to the n length binary BCH code using a monoid. It is established that the n length binary BCH code is embedded in it. In [38], by the use of monoid ring existence of a binary cyclic code of length $(n + 1)^{3^k} - 1$, where k is a positive integer, corresponding to a binary cyclic code of length n is explained. Both studies cannot show the existence of BCH codes corresponding to the length n binary BCH code.

Other than finite fields, linear codes over finite rings have been discussed in a series of papers initiated by Blake in [5] and [6]. He introduced the notions of the Hamming codes, Reed-Solomon codes and the BCH codes over arbitrary integer residue rings. Spiegel in [42] and [43], showed that the codes over the finite local ring \mathbb{Z}_{p^k} can be described in terms of codes over \mathbb{Z}_p and thus, are able to define codes over \mathbb{Z}_m , for any positive integer m . Shankar in [41], linked the notion of BCH codes over \mathbb{Z}_p to the class of BCH codes over the finite local ring \mathbb{Z}_{p^k} through a p reduction map. A remarkable development regarding Berlekamp-Massey decoding algorithm was given by Forney et al. in [13]. Recently Interlando, et al. in [18], have proposed a decoding procedure based on the modified Berlekamp-Massey algorithm for linear codes over the finite rings.

This thesis is organized as follows:

Chapter one describes a brief introduction to algebraic notions and algebraic coding theory. In Chapter two, the construction of n length binary cyclic codes as an ideal in the factor ring $\mathbb{F}_2[x; a\mathbb{N}_0]/((x^a)^n - 1)$ is explained. On the basis of binary cyclic code \mathcal{C}_n construction of binary

cyclic codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} , as ideals in the factor rings $\mathbb{F}_2[x]/((x^n - 1))$, $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]/((x^{\frac{a}{b}})^{bn} - 1)$ and $\mathbb{F}_2[x; \frac{1}{b}\mathbb{N}_0]/((x^{\frac{1}{b}})^{abn} - 1)$, are explained. The relationship among all of these binary cyclic codes is obtained through interleaving technique and by their generator and parity check matrices. Their error correction capability and decoding is also discussed in this Chapter.

In Chapter three, the construction of binary BCH codes using monoid ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$, where a , and b are integers such that $a, b > 1$, is given. We show the existence of non-primitive binary BCH code of length bn using an irreducible polynomial $p(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ of degree br , corresponding to a given length n binary BCH code \mathcal{C}_n generated by r degree primitive polynomial $p(x^a)$ in $\mathbb{F}_2[x; a\mathbb{N}_0]$. It is noticed that the binary BCH code \mathcal{C}_n is embedded in non-primitive BCH code \mathcal{C}_{bn} . In this way a link between primitive and non-primitive BCH codes is attained. The length of the binary BCH code \mathcal{C}_{bn} is well controlled and has better error correction capability.

Chapter four generalizes the case of Chapter three by taking the monoid ring $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ where $1 \leq j \leq m$ and m is any fix positive integer. This gives a new family of BCH codes such that the smaller length code is embedded in the larger length code. For wider range of examples and quick results we have proposed an algorithm which calculates all the BCH codes of particular length, their error correction capability, code rate and cyclotomic cosets. The simulation is carried out using computer programme *MATLAB*. It provides built in routines solely for primitive BCH codes with degree of primitive polynomial less than 16. Whereas in constructing non-primitive BCH codes, the degree of non-primitive polynomial is greater than 16. In order to lever these conditions *Generic Algorithm* is developed in *MATLAB*.

In Chapter five, we have constructed BCH codes over the field $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2 = 1 + \alpha\}$. In [10], Faria et al., showed the existence of DNA sequences which can be identified as codewords of BCH codes over the field \mathbb{F}_4 . They have proposed an algorithm capable of producing DNA sequences, associated with coding regions of genes, as codewords of error-correcting codes. Their results allow the use of efficient computer simulations in the analysis of biological processes such as polymorphism and mutation, consequently reducing time spent in laboratorial experiments. This is the main motivation to enhance the case of binary field, to the Galois field \mathbb{F}_4 . We compare the results of both the fields in this Chapter.

In Chapter six, instead of finite fields we construct BCH codes over finite rings (Galois

rings) using monoid rings. Following [41], we have constructed sequence of non-primitive BCH codes $\{C'_{b^j n}\}_{j \geq 1}$, in the factor ring $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0] / ((x^{\frac{a}{b^j}})^{b^j n} - 1) = \mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]_{b^j n}$ corresponding to n length BCH code C'_n having symbols from the local ring \mathbb{Z}_{2^m} . Thus, for a fixed m , against n length primitive BCH codes C'_n over \mathbb{Z}_{2^m} , there exist a sequence $\{C'_{b^j n}\}_{j \geq 1}$ of non-primitive BCH codes over \mathbb{Z}_{2^m} . Consequently, a link between primitive BCH codes C_n, C'_n , (over F_2 and \mathbb{Z}_{2^m}), and the sequences of non-primitive BCH codes $\{C_{b^j n}\}_{j \geq 1}$ and $\{C'_{b^j n}\}_{j \geq 1}$ is developed. For the decoding of binary BCH codes of length $b^j n$ over $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$, we modify Berlekamp-Massey decoding algorithm through which one can obtain the decoding of $b^j n$ and n length binary BCH codes over the field F_2 . Throughout this Chapter we have given comparison and connection between the codes constructed over Galois rings and Galois fields.

The Chapter 7, discusses the applications of the newly constructed BCH codes in cognitive radio, in the formation of DNA sequences and in data transmission. Lastly Chapter 8 concludes the thesis.

1.1 Algebraic notions and algebraic codes

In this chapter we provide basic concepts related to algebra and coding theory which are essential for the understanding of this thesis. It is divided into two main sections. In section 1, basic structures of algebra whereas in section 2, the fundamentals of algebraic coding theory are discussed.

1.2 Algebraic notions

The section we provide basic notions and terminologies related to semigroups, rings, fields, modules and semigroup rings.

1.2.1 Semigroup

A non-empty set S is said to be a **semigroup**, if it satisfies the closure property and associative law with respect to the binary operation $*$. An element e of a semigroup S is called the **identity element** of S , if $s * e = e * s$ for all $s \in S$. A semigroup S is said to be a **monoid** under the binary operation $*$, if identity element e exists in S . A semigroup is called a **commutative**

semigroup, if it satisfies $s * t = t * s$ for all $s, t \in S$. A non-empty subset T of a semigroup (monoid) S is a **subsemigroup (submonoid)** of S , if T itself is a semigroup (monoid) under the binary operation on S .

An element s of a monoid S is said to be **invertible**, if $s + t = e$, for some $t \in S$. A semigroup S is said to be **torsion free**, if each element of S has infinite order except the identity element. An element s of semigroup S is said to be **cancellative**, if $s + t = s + u$ implies $t = u$ for all $t, u \in S$. A semigroup S is called **cancellative semigroup** if all the elements of S are cancellative. A semigroup S is called a **cyclic semigroup**, if it is generated by a single element. A semigroup is called **totally ordered**, if there is a relation “ \sim ” on S which is reflexive, asymmetric, transitive and satisfies $s \sim t$ or $t \sim s$ for all $s, t \in S$. An order \sim on S is said to be compatible if, $s_1 \sim s_2$ implies $s_1 + s \sim s_2 + s$ for all $s_1, s_2, s \in S$. A cancellative and torsion free semigroup is totally ordered. A semigroup having a compatible total order is cancellative and torsion free.

A monoid G is said to be a **group** under the binary operation $*$, if inverse of each element of G exists in G i.e. for all $a \in G$, there exists $a' \in G$ such that $aa' = e$ and $a'a = e$. A group G under the binary operation $*$ is written as $(G, *)$. A group $(G, *)$ is called commutative or Abelian if all elements of G commute.

1.2.2 Ring

A set \mathcal{R} together with two binary operations, addition and multiplication is called a ring if \mathcal{R} is an Abelian group with respect to addition, semigroup with respect to multiplication and multiplication is distributive over addition. If \mathcal{R} is a monoid with respect to multiplication then \mathcal{R} is called a **unitary ring**. A non-empty subset U of \mathcal{R} is called a **subring** of \mathcal{R} , if U is itself a ring under the induced operations. A ring \mathcal{R} is a **commutative ring** if multiplication is commutative in \mathcal{R} .

An element r of a unitary ring \mathcal{R} is **invertible** or **unit** if, $r.r_1 = r_1.r = 1$, for some $r_1 \in \mathcal{R}$ which is an **inverse** of r in \mathcal{R} . A non-zero element z of \mathcal{R} is a **zero divisor** of \mathcal{R} , if $za = 0$ for some non-zero element a in \mathcal{R} . If \mathcal{R} has no zero divisor, then \mathcal{R} is called an **integral domain**. \mathcal{R} is **cancellative** if and only if it is an integral domain. A non-zero element p in a commutative ring \mathcal{R} is said to be **prime** if and only if p divides ab implies either p divides a

or p divides b . A non-zero element q in a ring \mathcal{R} is said to be **irreducible** (or non-factorable) if in every factorization $q = bc$, either b is invertible or c is invertible where $b, c \in \mathcal{R}$. A **ring homomorphism** is a map $\phi : \mathcal{R} \rightarrow U$, which preserves both the operations, i.e., (i) $\phi(x + y) = \phi(x) + \phi(y)$, (ii) $\phi(xy) = \phi(x)\phi(y)$, for all $x, y \in \mathcal{R}$, where \mathcal{R} and U are any rings.

If \mathcal{R} and U contains identity element, then the homomorphism of \mathcal{R} into U is called a **homomorphism of rings with identity**, which also preserve the identity element, i.e. $\phi(1_{\mathcal{R}}) = 1_U$. A ring homomorphism ϕ is said to be a **monomorphism (epimorphism)** if ϕ is one-one (**onto**). If ϕ is one-one and onto then ϕ is called a **ring isomorphism**. In this case the rings \mathcal{R} and U are said to be isomorphic and we write it as $\mathcal{R} \cong U$.

A subring I of a ring \mathcal{R} is called an **ideal** in \mathcal{R} if for each $i \in I$, $ri \in I$ for all $r \in \mathcal{R}$. Every ideal is a subring, but converse is not true. An ideal I of \mathcal{R} is called **proper ideal** if $I \cap \mathcal{R} \neq \mathcal{R}$ and is said to be **improper ideal**, if $I \cap \mathcal{R} = \mathcal{R}$. A proper ideal I of \mathcal{R} is called **prime ideal** of \mathcal{R} if, $ri \in I$ implies $r \in I$ or $i \in I$. A proper ideal of \mathcal{R} is said to be **maximal ideal**, if it is not contained in any other proper ideal of \mathcal{R} . Every ideal of a ring is contained in some maximal ideal of that ring. An ideal I is said to be **finitely generated** if it is generated by finite number of elements i.e., $I = (r_1, r_2 \dots r_n)$, $r_{i_{finite}} \in \mathcal{R}$. A finitely generated ideal is called **principal ideal** if it is generated by a single element i.e., $I = \langle a \rangle$ for some $a \in \mathcal{R}$. A ring \mathcal{R} is called **principal ideal ring (PIR)**, if all the ideals of \mathcal{R} are principal. A ring \mathcal{R} is integral domain if and only if (0) is prime ideal in \mathcal{R} . A commutative ring with identity is called a **local ring** if it has only one maximal ideal.

Let I be an ideal of the commutative ring \mathcal{R} with identity, then the **quotient ring (or factor ring)** of \mathcal{R} , denoted by \mathcal{R}/I , is the collection of all distinct equivalence classes of element of \mathcal{R} modulo I ; that is,

$$\mathcal{R}/I = \{a + I : a \in \mathcal{R}\}$$

It is easy to verify that \mathcal{R}/I is again a ring. Also, \mathcal{R}/I is commutative if \mathcal{R} is commutative.

A commutative ring \mathcal{R} with identity e is called a **field** F if every non-zero element in F is invertible. Let $(M, +)$ be an Abelian group and \mathcal{R} be a unitary commutative ring. Then, M is called an **\mathcal{R} -module**, if a product is defined between elements of the ring and elements of the module that is distributive over addition and is compatible with the ring multiplication.

If the ring \mathcal{R} is replaced by a field F , then M is called a **vector space** V **over** the field F . A module is a generalization of vector space. A module with basis is called a **free module**. Every vector space is a free module. A non-empty subset W of V is called a **subspace** of V if W itself is a vector space over the field F .

1.2.3 Rings of formal power series

Let \mathcal{R} be a commutative ring and \mathbb{N}_0 be the additive monoid of non-negative integers. The set $\mathcal{R}^{\mathbb{N}_0} = \{g : \mathbb{N}_0 \rightarrow \mathcal{R}\}$ of all infinite sequences from \mathbb{N}_0 to \mathcal{R} is called formal power series, define as: $g(0) = g_0, g(1) = g_1, \dots, g(n) = g_n$. Also: $g = (g_0, g_1, g_2, \dots, g_k, \dots)$, where $g_i \in \mathcal{R}$. The set $\mathcal{R}^{\mathbb{N}_0}$ is a ring containing \mathcal{R} as a subring. Let $g, h \in \mathcal{R}^{\mathbb{N}_0}$ be any arbitrary elements such that $g = (g_0, g_1, \dots)$ and $h = (h_0, h_1, \dots)$. The addition and multiplication of formal power series is defined as follows:

$$g + h = (g_0 + h_0, g_1 + h_1, \dots) \text{ and } gh = (k_0, k_1, \dots),$$

where for each $n \geq 0, k_n = \sum_{i+j=n} g_i h_j$. The zero element of $\mathcal{R}^{\mathbb{N}_0}$ is $(0, 0, 0, \dots)$ and the additive inverse of (g_0, g_1, \dots) is $(-g_0, -g_1, \dots)$. Hence $(\mathcal{R}^{\mathbb{N}_0}, +)$ becomes an Abelian group. Moreover, $(\mathcal{R}^{\mathbb{N}_0}, \cdot)$ is semigroup and multiplication is distributive over addition, therefore $(\mathcal{R}^{\mathbb{N}_0}, +, \cdot)$ forms a ring structure known as the ring of formal power series in one indeterminate over \mathcal{R} .

There exists an embedding $\theta : \mathcal{R} \rightarrow \mathcal{R}^{\mathbb{N}_0}$ defined by $\theta(r) = (r, 0, 0, 0, \dots)$. So, an element $r \in \mathcal{R}$ has a representation $(r, 0, 0, 0, \dots)$ in $\mathcal{R}^{\mathbb{N}_0}$. Now we define a power series in a formal way, we have

$$x = (0, 1, 0, \dots) \text{ and}$$

$$g_0 x = (0, g_0, 0, \dots), \text{ where } g_0 \in \mathcal{R}.$$

In general $g_n x^n, n \geq 1$ denotes the sequence $(0, 0, \dots, 0, g_n, 0, \dots)$, where g_n is the element at $(n+1)th$ term in this sequence. Thus $g(x) = (g_0, g_1, \dots, g_n, \dots)$ can be uniquely expressed in the form

$$\begin{aligned}
g &= g_0 + g_1x + g_2x^2 + \dots + g_nx^n + \dots \\
&= \sum g_kx^k.
\end{aligned}$$

To indicate the indeterminate x , usually we denote $\mathcal{R}^{\mathbb{N}_0}$ by $\mathcal{R}[[x]]$. If $g(x) = \sum g_kx^k$ is a non-zero power series (that is, if not all the $g_k = 0$) in $\mathcal{R}[[x]]$, then the smallest integer n such that $g_n \neq 0$ is called the order of $g(x)$ and denoted by $\text{ord}(g(x))$. Let $g(x), h(x) \in \mathcal{R}[[x]]$, with $\text{ord}(g(x)) = n$ and $\text{ord}(h(x)) = m$, then

$$g(x)h(x) = g_nh_mx^{n+m} + (g_{n+1}h_m + g_nh_{m+1})x^{n+m+1} + \dots$$

By the definition of multiplication in $\mathcal{R}[[X]]$, it can easily be seen that all the coefficients of $g(x)h(x)$ up to $(n+m)$ th are zero. If we assume that one of g_n and h_m is not a divisor of zero in \mathcal{R} , then $g_nh_m \neq 0$ and

$$\text{ord}(g(x)h(x)) = n + m = \text{ord}(g(x)) + \text{ord}(h(x)).$$

Polynomial Rings

The set of all power series in $\mathcal{R}[[x]]$, whose finite number of coefficients are nonzero is denoted by $\mathcal{R}[x]$. Therefore,

$$\mathcal{R}[x] = \{g_0 + g_1x + \dots + g_nx^n : g_n \in R, n \geq 0\}.$$

An element of $\mathcal{R}[x]$ is called polynomial in an indeterminate x over the ring \mathcal{R} . The polynomial ring $\mathcal{R}[x]$ is a subring of $\mathcal{R}[[x]]$. Given the non-zero polynomial

$$g(x) = g_0 + g_1x + \dots + g_nx^n = \sum_{k=0}^n g_kx^k \in \mathcal{R}[x],$$

the coefficient g_n is called the leading coefficient of $g(x)$ and the integer n is called the degree of the polynomial. The degree of a non-zero polynomial is therefore a non-negative integer. The zero polynomial has no degree. The non-zero constant polynomials are of zero degree. A

polynomial whose leading coefficient is 1 is called a **monic polynomial**.

1.2.4 Semigroup ring

Let $(S, *)$ be a commutative semigroup and \mathcal{R} be an arbitrary ring. The set of all finitely non-zero functions g from S into \mathcal{R} which are non-zero at finite points is denoted by $\mathcal{R}[S]$. This set $\mathcal{R}[S]$ is a ring with respect to binary operations addition and multiplication defined as:

$$(g + h)(s) = g(s) + h(s) \text{ and } (gh)(s) = \sum_{t*u=s} g(t)h(u), \quad (1.1)$$

where the symbol $\sum_{t*u=s}$ indicates that the sum is taken over all pairs (t, u) of elements of S such that $t * u = s$, and when s is not expressible in the form $t * u$ for any $t, u \in S$, then $(gh)(s) = 0$. The set $\mathcal{R}[S]$ is known as the *semigroup ring of S over \mathcal{R}* . The representation of $\mathcal{R}[S]$ will be $\mathcal{R}[x; S]$ whenever S is an additive monoid. There is an isomorphism between additive semigroup S and multiplicative semigroup $\{x^s : s \in S\}$, so a non-zero element g of $\mathcal{R}[x; S]$ is uniquely represented in the canonical form $\sum_{i=1}^n g(s_i)x^{s_i} = \sum_{i=1}^n g_i x^{s_i}$, where $g_i \neq 0$ and $s_i \neq s_j$ for $i \neq j$.

Degree and order of an element are not generally defined in monoid rings. However if S is a totally ordered monoid, degree and order of an element of monoid ring $\mathcal{R}[x; S]$ is defined in the following manner: If $g = \sum_{i=1}^n g_i x^{s_i} \in \mathcal{R}[x; S]$, where $s_1 < s_2 < \dots < s_n$, then s_n is called the degree of g written as $\deg(g) = s_n$ and s_1 is the order of g written as $\text{ord}(g) = s_1$.

The monoid ring $\mathcal{R}[x; S]$ is a **polynomial ring** in one indeterminate if $S = \mathbb{N}_0$.

1.2.5 Galois fields and Galois rings

Galois field

Polynomials over a field F modulo an irreducible polynomial $q(x)$ of degree s forms a field which is called an **extension field of degree s** over F . The extension field is obtained by adjoining a root say α , of $q(x)$ to the field F . It is denoted by $F[\alpha]$.

The residue classes of integers modulo any prime number p form a field of p elements called Galois field $GF(p)$. The field of polynomials over $GF(p)$ modulo an irreducible polynomial of degree m is called the Galois field of p^m elements denoted by $GF(p^m)$. For any number $q = p^m$,

that is a power of a prime number, there is a field $GF(q)$, which has q elements. Every finite field is isomorphic to some Galois field. They differ only in the way the elements are named.

For example x^3+x+1 the irreducible polynomial of degree 3 gives quotient ring $GF(2[x]/(x^3+x+1))$ which is isomorphic to Galois field $GF(2^3)$ of order 8. The elements of this Galois field are polynomials of degree less than 3 with coefficients belongs to $GF(2)$.

Galois ring

For positive integers m, s and p , where p is a prime, we have Galois ring of order p^{ms} denoted by $GR(p^m, s)$. It is the Galois extension of degree s of the ring $\mathbb{Z}/\mathbb{Z}_{p^m}$ of integers mod p^m . For $s = 1$, the ring $GR(p^m, 1)$ is $\mathbb{Z}/\mathbb{Z}_{p^m}$, whereas for $m = 1$, the ring $GR(p, s)$ is F_{p^s} .

Let $\phi(x) \in \mathbb{Z}[x]$ be a monic irreducible polynomial of degree s , over \mathbb{Z} , which remains irreducible modulo p , then the Galois ring $GR(p^m, s)$ is isomorphic to the quotient ring $(\mathbb{Z}/\mathbb{Z}_{p^m})[x]/(\phi(x))$, where $\phi(x)$ is a polynomial in $(\mathbb{Z}/\mathbb{Z}_{p^m})[x]$.

1.3 Algebraic codes

In modern communication systems transmission of messages through a channel or storage of massive amount of data consistently, possibility of errors is always there. Different coding and decoding schemes are used to correct errors from the received message to recover the original message. The basic concept of working behind these schemes is to add parity bits to the message bits. The codes in which the number of message bits and parity bits are kept fixed are algebraic codes, which are basically block codes. The construction of these codes is highly based on the algebra, therefore they are called algebraic codes. This section is divided into three subsections in which we discuss fundamentals of coding theory, linear codes, cyclic codes and the construction of BCH codes over the finite fields.

1.3.1 Fundamentals of coding theory

This subsection discusses some basic notions and terminologies related to coding theory which are used in all sections of this dissertation.

Consider a finite set \mathcal{S} of p elements, a p -ary **code** \mathcal{C} of length n is a subset of the set \mathcal{S}^n (the Cartesian product), where n is a positive integer. The elements of the set \mathcal{S} are called

symbols or **bits** and the set itself is referred to as **symbol set**. The elements of the code \mathcal{C} are called **codewords**. A code is said to be **binary code**, if the number of symbols in the symbol set are two and is called **ternary code**, if size of symbol set is three. The size of the set \mathcal{S}^n is p^n . A subset \mathcal{C} of \mathcal{S}^n is said to be **trivial code**, if the size of \mathcal{C} is one or p^n . If all the codewords of a code \mathcal{C} have same coordinates, i.e., $\mathcal{C} = \{aaa...a | a \in \mathcal{S}\}$, then such a code is called a **repetition code**. Size of a repetition code is equal to the size of the symbol set.

Let c_1 and c_2 be two elements of \mathcal{S}^n , $c_1 = c_{1,1}c_{1,2} \cdots c_{1,n}$, $c_2 = c_{2,1}c_{2,2} \cdots c_{2,n}$. The **Hamming distance** between c_1 and c_2 is the total number of subscripts in which the coordinates of c_1 and c_2 differ, i.e.,

$$|\{j : c_{1,j} \neq c_{2,j}\}|.$$

The Hamming distance between two elements c_1 and c_2 of \mathcal{S}^n is denoted by $d(c_1, c_2)$. For example, $d(11011, 10010) = 2$. Hamming distance satisfies the following three conditions:

- (1) $d(c_1, c_2) = 0$ if and only if $c_1 = c_2$.
- (2) $d(c_1, c_2) = d(c_2, c_1)$, for all $c_1, c_2 \in \mathcal{S}^n$.
- (3) $d(c_1, c_3) \leq d(c_1, c_2) + d(c_2, c_3)$, for all c_1, c_2 and $c_3 \in \mathcal{S}^n$.

Hence, d is a metric on the set \mathcal{S}^n .

The **minimum distance** of the code \mathcal{C} is the smallest Hamming distance between any two codewords in the code \mathcal{C} , i.e.,

$$d(\mathcal{C}) = \min\{d(c_i, c_j) : c_i, c_j \in \mathcal{C}, c_i \neq c_j\}.$$

The **minimum distance or minimum Hamming distance** of a code \mathcal{C} is denoted by d . For example, the minimum Hamming distance of an n length repetition code is n . The **Hamming weight of a vector** $c \in \mathcal{F}^n$ is the number $w(c)$ of its nonzero coordinates i.e., $w(c) = d(c, 0)$. If \mathcal{C} is a linear code, the distance $d(\mathcal{C})$ is the same as the minimum weight of nonzero words i.e.,

$$d(\mathcal{C}) = \min\{w(c) \mid c \in \mathcal{C}, c \neq 0\}.$$

A code \mathcal{C} is known as **t -error correcting code**, if it is capable of correcting t or less errors whenever t or fewer errors have been occurred during the transmission of a codeword. The error detection and correction capabilities of a code are directly related with the minimum distance of the code. Following theorem provides a relation between the minimum distance and its error detection and correction capabilities.

Theorem 1 [27, 4.1.3] *From any transmitted codeword, a code \mathcal{C} with minimum distance d can detect and correct upto $d - 1$ and $\lfloor \frac{d-1}{2} \rfloor$ errors respectively.*

A code of length n , having minimum distance d , and size M is represented by (n, M, d) -code. A code is said to be a very good, if it satisfies the following three conditions:

- (1) *Length n should be small so that fast and low cost transmission could be possible.*
- (2) *Minimum distance d should be large so that more errors can be detected and corrected.*
- (3) *Size M should be large so that a variety of messages can be sent.*

1.3.2 Linear codes

Linear codes are much accordant to algebraic treatment due to possessing many algebraic properties. In these codes, the symbol set is a finite field F . The set F^n is an n -dimensional vector space over the field F . A subset \mathcal{C} of F^n is said to be **linear code**, if it is a subspace of the vector space F^n . A subspace of a finite dimensional vector space is also finite dimensional. Hence, every linear code has a dimension. A linear code \mathcal{C} of length n and dimension k is represented by (n, k) . The size of an (n, k) -code is equal to p^n , where p is the size of the corresponding field F . Henceforth, F_p denotes a finite field of p elements and a vector c in F_p^n is represented as $c = (c_1, c_2, \dots, c_n)$.

Let \mathcal{C} be an (n, k) -code over the field F . A $k \times n$ matrix with rows forming the basis for the codes \mathcal{C} is called **generator matrix** for the code \mathcal{C} . Let G be a generator matrix for an (n, k) -code \mathcal{C} , every element of \mathcal{C} can be uniquely expressed as the linear combination of the rows of G . In other words, \mathcal{C} is the row space of the matrix G , i.e., $\mathcal{C} = \{c.G \mid c \in F^k\}$. A vector c of the space F^k is of length k , however a vector of an (n, k) -code \mathcal{C} is of length n . As, the size of both spaces is p^k . Therefore, we have a bijection δ from F^k to \mathcal{C} defined as: $\delta(c) = c.G$ for all $c \in F^k$. In these settings, the vectors of F^k are called **messages** and their images are

said to be **codewords**, i.e., $c.G$ is the codeword corresponding to the message c . As, we have already mentioned that this bijection maps a vector of length k on to a vector of length n . The $n - k$ elements attached to c are called **parity bits**. The map δ is called **encoding map**. The ratio of message length to the codeword length is called **code rate** of an (n, k) -code \mathcal{C} i.e., k/n . The dual code of \mathcal{C} can be defined as:

$$\mathcal{C}^\perp = \{c' \in \mathbb{F}^n : c \cdot c' = 0 \text{ for all } c \in \mathcal{C}\}. \quad (1.2)$$

Where the multiplication is defined as: $c \cdot c' = cc'^T$. It is well known that if \mathcal{C} is a k -dimensional subspace of an n -dimensional space, then its dual \mathcal{C}^\perp is also an $(n - k)$ -dimensional subspace of the space and hence a code. For example, the dual of a binary repetitive code $\mathcal{C} = \{0000, 1111\}$ is a $(4, 3)$ - code over \mathbb{F}_2 and is equal to:

$$\mathcal{C}^\perp = \{0000, 1100, 1010, 1001, 0110, 0101, 0011, 1111\}. \quad (1.3)$$

Moreover, we have $(\mathcal{C}^\perp)^\perp = \mathcal{C}$. Since, dual of a code \mathcal{C} is also a subspace therefore, it also has generator matrix. The generator matrix of \mathcal{C}^\perp is very important for the decoding purposes.

Let \mathcal{C} be any linear (n, k) -code and \mathcal{C}^\perp is its dual. Then the generator matrix H of the code \mathcal{C}^\perp is called the parity check matrix of the code \mathcal{C} . It is easy to prove that a vector c of the space \mathbb{F}^n is a code-word of the code \mathcal{C} , iff $c.H^T = 0 = H.c^T$. Following theorem provides the relation between the generator matrix and parity check matrix of a linear code.

Theorem 2 [27, 4.2.9] *Let \mathcal{C} be a linear (n, k) -code over the field F . Let G and H be generator and parity-check matrices of the code \mathcal{C} respectively. Then $G.H^T = 0 = H.G^T$. Conversely, if G is any $k \times n$ matrix, and H is an $(n - k) \times n$ matrix, of rank k and $n - k$ respectively, with $G.H^T = 0$. Then H is a parity-check matrix iff G is a generator matrix for the code \mathcal{C} .*

For example, a linear code $\mathcal{C} = \{000, 111\}$ has a generator matrix $[111]$. The dual of this code is: $\mathcal{C}^\perp = \{000, 110, 011, 101\}$. The parity check matrix of \mathcal{C} is:

$$H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad (1.4)$$

Suppose G is a generator matrix of an (n, k) – code \mathcal{C} with k linearly independent columns. Then by applying suitable row operations on G , the matrix G becomes equivalent to $G^* = [I_k : B]$, where I_k is the identity matrix of order k , and B is a matrix of order $k \times (n - k)$. Of course, rows of G^* are linearly independent and hence G^* is a generator matrix of the code \mathcal{C} . This generator matrix is called **canonical generator matrix** of the code \mathcal{C} . The **canonical parity check matrix** is given by: $H^* = [-B^T : I_{n-k}]$. Two (n, k) – codes \mathcal{C} and \mathcal{C}' over the same field F are said to be **equivalent codes**, if there exist a bijection from Ψ from \mathcal{C} to \mathcal{C}' such that

$$\Psi(c_1, c_2, \dots, c_n) = (\alpha_1 c_{\sigma(1)}, \alpha_2 c_{\sigma(2)}, \dots, \alpha_n c_{\sigma(n)}), \quad (1.5)$$

where $\alpha_1, \dots, \alpha_n \in F \setminus \{0\}$ and σ is a permutation on the set $\{1, 2, \dots, n\}$.

Theorem 3 [27, 4.2.18] *Let \mathcal{C} be an (n, k) – code with minimum distance d , then d is equal to the minimum number of linearly independent columns in a parity check matrix of the code \mathcal{C} and hence, $d(\mathcal{C}) \leq n - k + 1$.*

1.3.3 Cyclic codes

In this subsection, we discuss cyclic codes which are in fact linear codes. These codes are of great interest due to their strong algebraic structure.

A **cyclic shift** on F^n is a ∂ map from F^n to F^n defined as: $\partial(a_1 a_2 a_3 \dots a_n) = (a_n a_1 a_2 \dots a_{n-1})$. It is easy to prove that cyclic shift is a linear operator. An (n, k) – code \mathcal{C} is said to be **cyclic code**, if $\partial(\mathcal{C}) \subseteq \mathcal{C}$.

The code $\mathcal{C} = \{000, 110, 011, 101\}$ is a cyclic code. If G is a generator matrix of an (n, k) – code \mathcal{C} , then \mathcal{C} is cyclic iff $\partial(\{R_i\}_{i=1}^k) \subset \mathcal{C}$. If we denote the set of all polynomials of degree less than n over F by \mathcal{P}_n , then \mathcal{P}_n is an n – dimensional vector space over the field F . By linear algebra, F^n is isomorphic to the space \mathcal{P}_n . Now, consider the factor ring:

$$\frac{\mathbb{F}[x]}{(x^n - 1)} = \{f_0 + f_1\alpha + f_2\alpha^2 + \dots + f_{n-1}\alpha^{n-1} : f_{i's} \in \mathbb{F}\}, \quad (1.6)$$

where α is the root of the polynomial $x^n - 1$. Replacing α by x , the above factor ring becomes equal to the set of all polynomials of degree less than n and hence isomorphic to \mathcal{P}_n . Of course, $\frac{\mathbb{F}[x]}{(x^n - 1)}$ is a ring. The operation of multiplication in \mathcal{P}_n is defined as: $f(x) * g(x) = f(x)g(x) \pmod{(x^n - 1)}$

$(x^n - 1)$). The cyclic shift of an element c of \mathcal{C} is equal to $x * c(x)$ in the ring \mathcal{P}_n . Thus, a linear code over the field F is cyclic iff $x * c(x) \in \mathcal{C}$ for all $c(x) \in \mathcal{C}$.

Theorem 4 [27, 4.3.5] *A linear code \mathcal{C} of length n is cyclic iff \mathcal{C} is an ideal of the ring \mathcal{P}_n .*

Theorem 5 [27, 4.3.6] *Let \mathcal{C} be any non-zero ideal in a ring $F[x]_n$, then*

- (a) *there exists a unique monic polynomial $g(x)$ of least degree in \mathcal{C} ,*
- (b) *$g(x)$ divides $x^n - 1$ in $F[x]$,*
- (c) *for all $p(x) \in \mathcal{C}$, $g(x)$ divides $p(x)$ in $F[x]$,*
- (d) *$\mathcal{C} = (g(x))$.*

Conversely, assume that \mathcal{C} is an ideal generated by $a(x) \in F[x]_n$. Then $a(x)$ is a least degree polynomial in \mathcal{C} iff $a(x)$ divides $x^n - 1$ in $F[x]$.

Hence, by computing all the irreducible factors of the polynomial $x^n - 1$, we can find out all possible cyclic codes of length n . For example, if we take $n = 3$, then the only non trivial cyclic codes over F_2 are the ideals generated by $x - 1$ and $x^2 + x + 1$.

Generator polynomial

Let \mathcal{C} be a non-zero ideal in $F[x]_n$. Let $g(x)$ be the unique monic polynomial of smallest degree in \mathcal{C} . Then $g(x)$ is said to be a generator polynomial of the cyclic code \mathcal{C} .

Theorem 6 [27, 4.3.11] *Let $\mathcal{C} \subset F[x]_n$ be any cyclic code with generator polynomial*

$$g(x) = g_0 + g_1x + g_2x^2 + \dots + g_rx^r, \text{ where } g_r = 1. \quad (1.7)$$

Then the dimension of the code \mathcal{C} is $n - r$. Moreover, the $(n - r) \times n$ matrix

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{r-1} & g_r & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & g_2 & \cdots & g_{r-1} & g_r & \cdots & 0 \\ \vdots & \vdots & \vdots & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & g_r \end{bmatrix}, \quad (1.8)$$

is a generator matrix of \mathcal{C} .

A code \mathcal{C} in $\mathbb{F}[x]_n$ generated by the polynomial $g(x)$ such that a polynomial $h(x)$ in $\mathbb{F}[x]$ satisfies the relation $x^n - 1 = g(x)h(x)$. Then $h(x)$ is called a check polynomial of the code \mathcal{C} . Since, $x^n - 1$ is monic, therefore check polynomial is also monic and unique. It is easy to prove that:

$$\mathcal{C} = \{c(x) \in \mathbb{F}[x]_n | c(x) * h(x) = 0\}. \quad (1.9)$$

The check polynomial $h(x)$ is also an irreducible divisor of the polynomial $x^n - 1$, and hence $h(x)$ also generated a cyclic code. Suppose $h(x) = h_0 + h_1x + h_2x^2 + \dots + h_kx^k$, the reciprocal of $h(x)$ is given as:

$$\overline{h(x)} = h_k + h_{k-1}x + h_{k-2}x^{k-2} + \dots + h_0x^k \quad (1.10)$$

This polynomial $\overline{h(x)}$ is an irreducible divisor of the polynomial $x^n - 1$ and hence generator of a cyclic code $\mathbb{F}[X]_n$ of dimension $(n - k) \times n$.

Theorem 7 [27, 4.3.14] *Let $h(x)$ be the check polynomial of a cyclic code \mathcal{C} . The cyclic code generated by $\overline{h(x)}$ is equal to the dual of the code \mathcal{C} . Consequently, the matrix*

$$\begin{bmatrix} h_k & h_{k-1} & \dots & \dots & h_0 & 0 & 0 & 0 & 0 \\ 0 & h_k & \dots & \dots & h_1 & h_0 & 0 & \dots & 0 \\ \vdots & \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & \dots & 0 & h_k & h_{k-1} & \dots & \dots & \dots & h_0 \end{bmatrix} \quad (1.11)$$

is a parity check matrix for the code \mathcal{C} .

1.3.4 Bose-Chaudhuri-Hocquenghem codes (BCH codes)

In this subsection we discuss a very important kind of cyclic code named as BCH codes. First we discuss some properties of finite fields. Every finite field has order power of some prime p . Finite fields of order $q = p^m$ are denoted by \mathbb{F}_q or $GF(q)$, where p is a prime number and m is any positive integer. The set \mathbb{F}_q^* denotes the set of all non zero elements of \mathbb{F}_q and is a cyclic group of order $p^m - 1$ under multiplication.

If $p^m - 1$ is divisible by any number n , then there is an element b in \mathbb{F}_q^* whose order is n that is $o(b) = n$, then b is called primitive n th root of unity in \mathbb{F}_q . If $o(b) = p^m - 1$, then b

is called primitive element in F_q . For the finite field F_q there exists an irreducible polynomial $h(x)$ in $F_q[x]$ of degree r , the quotient ring $F_q[x]/(h(x))$ is a field of size q^r , denoted by F_{q^r} or $GF(q, r)$. The field F_{q^r} is called an extension field of F_q of degree r .

Let $\alpha \in F_{q^r}$, then there exist unique monic polynomial of least degree $g(x) \in F_q[x]$ such that $g(\alpha) = 0$. The $g(x)$ is an irreducible and minimal polynomial of α over F_q . If there exists another polynomial $h(x)$ in $F_q[x]$ such that $h(\alpha) = 0$ then $g(x) \mid h(x)$, also degree of $g(x)$ divides r . If α is primitive element in F_{q^r} then degree of $g(x)$ is equal to r .

Theorem 8 [27] *Let α be any element in F_{q^r} . Then $\alpha, \alpha^q, \alpha^{q^2}, \dots$ have the same minimal polynomial over the field F_q .*

Now, we explain in detail the construction of BCH codes over the field F_q .

Definition 9 *The set of the powers of α is called cyclotomic coset. The smallest entries of the cyclotomic cosets are called coset representatives.*

Consider the positive integers c, d, n, q , where $2 \leq d \leq n$, n is relatively prime to q and divides $q^s - 1$, where s is the least positive integer, such that $q^s \equiv 1 \pmod{n}$. Let β be a primitive n th root of unity in F_{q^s} and $m_i(x)$ is the minimal polynomial of β^i , for $i = c, c+1, \dots, c+d-2$. Then the generator polynomial of n length BCH code of designed distance d is obtained by taking lcm of all minimal polynomials.

Definition 10 *A BCH code over a finite field F_q of block length n and designed distance d is a cyclic code generated by a generating polynomial*

$$g(x) = \text{lcm}\{m_i(x) : l \leq i \leq c+d-2\} \in F_q[x], \quad (1.12)$$

*whose root set contains $d-1$ distinct elements $\beta, \beta^2, \dots, \beta^{c+d-2}$, where β is a primitive n^{th} root of unity and c is some integer. For $n = q^s - 1$, the BCH code is called **primitive BCH code** and for $c = 1$, it is known as **narrow sense BCH code**.*

1.3.5 Decoding algorithms

The process of estimating correct transmitted vector from the received vector is called decoding. In this section, we discuss decoding procedures of linear, cyclic and BCH codes. Decoding is

done by the principle of *Maximal likelihood decoding*, in which we have to find the codeword which is nearest to the received vector. To explain decoding algorithms, we have to introduce the concept of error vector. Let \mathcal{C} be an (n, k, d) -code over the field \mathbb{F}_q and we transmit a codeword \bar{a} but received word is \bar{b} . Then the *error vector* is given by $e = \bar{b} - \bar{a}$. So, $\bar{b} = \bar{a} + e$ implies that $d(\bar{b}, \bar{a}) = w(\bar{b} - \bar{a}) = w(e)$. To decode the vector \bar{b} , we search a codeword \bar{c} such that $d(\bar{b}, \bar{c}) = w(\bar{b} - \bar{c})$ is minimal.

Standard Array Decoding

Standard array decoding is explained with the help of cosets and coset leaders.

Definition 11 (*Cosets and Coset Leaders*) Let \mathcal{C} be an (n, k) -code over \mathbb{F}_q .

1. A coset of \mathcal{C} is a coset of subgroup of the group $(\mathbb{F}_q^n, +)$. It is a set of the form $a + \mathcal{C} = \{a + c : c \in \mathcal{C}\}$ for all $a \in \mathbb{F}_q^n$.
2. A vector of smallest weight in the coset is known as a coset leader.

The cosets of \mathcal{C} form a partition of \mathbb{F}_q^n and all cosets are of the same size. So, for (n, k) -code \mathcal{C} over \mathbb{F}_q , every coset of \mathcal{C} has q^k vectors and \mathcal{C} contains q^{n-k} cosets.

Standard Array Decoding states the vector \bar{b} is decoded as the codeword \bar{c} if and only if the coset containing \bar{b} has a coset leader $\bar{b} - \bar{c}$. Thus, the error vector is equal to the coset leader.

Hence, we follow the following two steps for decoding \bar{b} :

1. Determine a coset leader e of the coset containing \bar{b} .
2. Decode \bar{b} as $\bar{b} - e$.

It can be done by using a standard array, which is a table having q^{n-k} rows and q^k columns. The first row has all codewords and the first column has coset leaders of each coset. The element in the i th row and j th column is the sum of coset leader of position (i th row, 1st column) and the codeword of position (1st row, j th column). So, by using a standard array, vector \bar{b} is decoded as the codeword that is in the first row and in the same column in which \bar{b} occurs.

Syndrome decoding

Standard array decoding is used only when the length of code is small. For large length codes, size of the array becomes very large so the Syndrome decoding method is used. Given the

coset leaders, the vector \bar{b} is decoded just by finding the row (coset) in which \bar{b} occurs. There is no need to determine exact position of \bar{b} in the array. So if we find the coset leader of coset containing \bar{b} , then there is no need of standard array. This is the basic idea of syndrome decoding.

Definition 12 Let \mathcal{C} be an (n, k, d) -code over F_q and H be a parity check matrix of \mathcal{C} . Then for $\bar{b} \in F_q^n$, the syndrome of \bar{b} (with respect to H) is defined as $\text{syn}(\bar{b}) = S(\bar{b}) = \bar{b}H^T$. If \mathcal{C} is a cyclic code with generator polynomial $g(x)$, then the syndrome is given by $S(\bar{b}) = \text{rem}_{g(x)}(x^{n-k}\bar{b}(x))$.

In this method, we use a syndrome table instead of standard array. For each coset, we find a coset leader and also its syndrome. Vector \bar{b} is decoded through following steps:

1. Compute the syndrome $S(\bar{b})$ of received vector \bar{b} .
2. Find a coset leader e in the table such that $S(\bar{b}) = S(e)$.
3. Decode \bar{b} as $\bar{b} - e$.

BCH codes over finite rings

Here we are given a very brief note on the construction of BCH codes over finite rings. The construction of BCH codes over finite rings was given by Priti Shankar in [41]. He constructed the codes over Z_{p^m} by the method which is same as the construction of codes over F_q . For this purpose, extension of Galois rings is used, where few conditions of extension of Galois field are lost. To explain important properties of Galois ring extension, let Z_{p^m} be the ring of polynomial and $h(x)$ be the irreducible polynomial of degree r over Z_{p^m} and also over $GF(p)$: Then $R = GR(p^m, r) = Z_{p^m}[x] / \langle h(x) \rangle$ is called Galois extension of Z_{p^m} of dimension r . For certain value of n , $x^n - 1$ can be written as into linear factors over $GR(p^m, r)$; where n is such that $\gcd(n, p^m) = \gcd(n, p) = 1$. With the help of these factors we determine the cyclic and BCH codes over Z_{p^m} . Zero divisors of $GR(p^m, r)$ form an Abelian group under addition having elements of degree $r - 1$ or less. The coefficients of these polynomials are zero divisors in Z_{p^m} . It means $GR(p^m, r)$ is a local ring. The Units of $GR(p^m, r)$ are those polynomials having atleast one coefficient unit in Z_{p^m} . The units of $GR(p^m, r)$ form a multiplicative group and it is represented by R^* . It is an Abelian group and can be written as a direct product of cyclic groups. We are looking for the maximal cyclic subgroup G_n , whose elements are the zeros of $x^n - 1$.

The following results are important for construction of G_n and BCH codes.

Theorem 13 [41, Theorem 2] *There is unique G_n of R^* of order relatively prime to p . This cyclic subgroup has order $p^r - 1$.*

Theorem 14 [41, Theorem 3] *Let α generates the cyclic subgroup having order n in R^* , such that $\gcd(n, p) = 1$. Then the polynomial $x^n - 1$ can be factorized as*

$$x^n - 1 = (x^n - \alpha)(x^n - \alpha^2) \dots (x^n - \alpha^n),$$

iff $R_p(h(x))$ has order n in K^ , which is the multiplicative subgroup of $K = GF(p^r)$.*

Theorem 15 [41, Theorem 4] *Let $\bar{\alpha} = R_p(\alpha)$ generate a subgroup which is cyclic of order n in K^* . Then α generate a cyclic subgroup of order $n.d$ in R^* , where $d \geq 1$, and $G_n = \langle d \rangle$.*

Lemma 16 [3, Lemma 3.1] *Let α primitive element of G_n . Then the differences $\alpha^{l_1} - \alpha^{l_2}$ are units in R if $0 \leq l_1 \neq l_2 \leq n - 1$.*

On the basis of above results, the generator polynomial $g(x)$ of cyclic BCH code of length n in $GR(p^m, r)$ can be calculated as

$$g(x) = lcm(M_1(x), M_2(x), \dots, M_2(x)),$$

where $M_i(x); 1 \leq i \leq 2t$ are the minimal polynomials of α^{b+i} over Z_{p^m} ; for some $b \geq 0$ and $t \geq 1$. The polynomials $M_i(x)$ over $GR(p^m, r)$ are calculated by the method similar to the calculation of minimal polynomial over the Galois fields.

BCH codes over finite rings are decoded by Berlekamp-Massey algorithm. It is a well known algorithm and very lengthy, hence we are not discussing here.

Chapter 2

Cyclic codes as ideals in $\mathbb{F}_2[x; a\mathbb{N}_0]_n$, $\mathbb{F}_2[x]_{an}$, $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ **and** $\mathbb{F}_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$

Communication channels are affected by disturbances that cause transmission errors to cluster into bursts. Random error correcting codes are not efficient for correcting burst errors and therefore, it is required to design specialized codes which can correct burst errors.

In this chapter, construction technique of cyclic codes is improved by using monoid rings $\mathbb{F}_2[x; a\mathbb{N}_0]$ and $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ instead of polynomial ring $\mathbb{F}_2[x]$. The new scheme is formulated in such a way, that, for an n length binary cyclic code \mathcal{C}_n generated by r degree polynomial $g(x^a)$ in $\mathbb{F}_2[x; a\mathbb{N}_0]$ three different binary cyclic codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} of length an , bn and abn are found. It is proved that these new binary cyclic codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} have generating polynomials $g(x)$ in $\mathbb{F}_2[x]$ of degree ar , $g(x^{\frac{a}{b}})$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ of degree br and $g(x^{\frac{1}{b}})$ in $\mathbb{F}_2[x; \frac{1}{b}\mathbb{N}_0]$ of degree abr respectively. It is shown that binary cyclic codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} are interleaved codes of depths a , b , and ab respectively. We have also established that if an initial code \mathcal{C}_n is capable of correcting t errors, then the interleaved codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} are capable of correcting t bursts of length a , b and ab or less. If \mathcal{C}_n is capable of correcting all bursts of length l or less, then the interleaved codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} are capable of correcting all bursts of length al , bl and abl or less.

Throughout this work we use the following totally ordered monoids which are

$$a\mathbb{N}_0 = \{0, a, 2a, \dots\}, \frac{a}{b}\mathbb{N}_0 = \{0, \frac{a}{b}, \frac{2a}{b}, \dots\} \text{ and } \frac{1}{b}\mathbb{N}_0 = \{0, \frac{1}{b}, \frac{2}{b}, \dots\},$$

where a and b are integers satisfying $a, b \geq 1$ with $b = a + 1$. The factor rings $\frac{\mathbb{F}_2[x; a\mathbb{N}_0]}{((x^a)^n - 1)}$, $\frac{\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]}{((x^{\frac{a}{b}})^{bn} - 1)}$ and $\frac{\mathbb{F}_2[x; \frac{1}{b}\mathbb{N}_0]}{((x^{\frac{1}{b}})^{abn} - 1)}$ are denoted by $F_2[x; a\mathbb{N}_0]_n$, $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ and $F_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$, where $((x^a)^n - 1)$, $((x^{\frac{a}{b}})^{bn} - 1)$ and $((x^{\frac{1}{b}})^{abn} - 1)$ are respectively the principal ideals in the monoid rings $F_2[x; a\mathbb{N}_0]$, $F_2[x; \frac{a}{b}\mathbb{N}_0]$ and $F_2[x; \frac{1}{b}\mathbb{N}_0]$. The arbitrary elements

$$\begin{aligned} f(x^a) &= f_0 + f_a(x^a) + f_{2a}(x^a)^2 + \cdots + f_{an}(x^a)^n \text{ in } F_2[x; a\mathbb{N}_0], \\ f(x^{\frac{a}{b}}) &= f_0 + f_{\frac{a}{b}}(x^{\frac{a}{b}}) + f_{2\frac{a}{b}}(x^{\frac{a}{b}})^2 + \cdots + f_{\frac{a}{b}n}(x^{\frac{a}{b}})^n \text{ in } F_2[x; \frac{a}{b}\mathbb{N}_0], \\ \text{and } f(x^{\frac{1}{b}}) &= f_0 + f_{\frac{1}{b}}(x^{\frac{1}{b}}) + f_{2\frac{1}{b}}(x^{\frac{1}{b}})^2 + \cdots + f_{\frac{1}{b}n}(x^{\frac{1}{b}})^n \text{ in } F_2[x; \frac{1}{b}\mathbb{N}_0] \end{aligned}$$

are known as (*generalized*) *polynomials*.

2.1 Cyclic codes as ideals in $F_2[x; a\mathbb{N}_0]_n$

A *linear code* \mathcal{C} of length n is a subspace of the vector space of all n -tuples over the binary field F_2 . A linear code \mathcal{C} over F_2 is a *cyclic code*, if $v = (v_0, v_1, \dots, v_{n-1}) \in \mathcal{C}$, then every cyclic shift $v^{(1)} = (v_{n-1}, v_0, \dots, v_{n-2}) \in \mathcal{C}$, where $v_i \in F_2$ and $0 \leq i \leq n - 1$.

Andrade and Shah has constructed cyclic codes over a local finite commutative ring \mathcal{R} , through the monoid rings $\mathcal{R}[x; \frac{1}{3}\mathbb{Z}_{\geq 0}]$, $\mathcal{R}[x; \frac{1}{2}\mathbb{Z}_{\geq 0}]$ and $\mathcal{R}[x; \frac{1}{2^2}\mathbb{Z}_{\geq 0}]$ in [4], [34] and [35] respectively. However in [33] the cyclic codes of certain types are discussed corresponding to the ascending chain of monoid rings. Due to the fact that $F_2[x] \subset F_2[x; \frac{1}{b}\mathbb{N}_0]$ mentioned in [4], [39], [38], [37], [36], [34] and [35], certain cyclic codes are discussed in such a way that the generator polynomials of cyclic codes in $\frac{\mathbb{F}_2[x]}{(x^n - 1)}$ and $\frac{\mathbb{F}_2[x; \frac{1}{b}\mathbb{N}_0]}{((x^{\frac{1}{b}})^{bn} - 1)}$ have a relationship. Whereas since $F_2[x] \not\subset F_2[x; \frac{a}{b}\mathbb{N}_0]$, this posed a hurdle to construct cyclic codes in factor ring $\frac{\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]}{((x^{\frac{a}{b}})^{bn} - 1)}$. However $F_2[x; a\mathbb{N}_0] \subset F_2[x; \frac{a}{b}\mathbb{N}_0]$, motivates us to construct cyclic codes in $F_2[x; a\mathbb{N}_0]_n$. A generalized polynomial

$$f(x^a) = f_0 + f_a(x^a) + f_{2a}(x^a)^2 + \cdots + f_{an}(x^a)^n, \quad (2.1)$$

in $F_2[x; a\mathbb{N}_0]$ of degree n has degree an in $F_2[x]$ and is known as the polynomial in indeterminate x . If $f(x^a) \in F_2[x; a\mathbb{N}_0]$ is a monic generalized polynomial of degree n then the factor ring $\frac{\mathbb{F}_2[x; a\mathbb{N}_0]}{(f(x^a))}$ is the ring of residue classes of generalized polynomials in $F_2[x; a\mathbb{N}_0]$ modulo the

principal ideal $(f(x^a))$. Thus, $f(x^a) = (x^a)^n - 1$ gives the factor ring

$$\frac{\mathbb{F}_2[x; a\mathbb{N}_0]}{((x^a)^n - 1)} = \{c_0 + c_a\alpha + c_{2a}\alpha^2 + \dots + c_{a(n-1)}\alpha^{n-1} : c_0, c_a, \dots, c_{a(n-1)} \in \mathbb{F}_2\}, \quad (2.2)$$

where α denotes the coset $x^a + ((x^a)^n - 1)$. Furthermore, $f(\alpha) = 0$, when α satisfies the relation $\alpha^n - 1 = 0$.

By writing x^a in place of α , the ring $\frac{\mathbb{F}_2[x; a\mathbb{N}_0]}{((x^a)^n - 1)}$ becomes $F_2[x; a\mathbb{N}_0]_n$ in which the relation $(x^a)^n = 1$ holds. In fact, $F_2[x; a\mathbb{N}_0]_n$ is an algebra over the field F_2 . The multiplication $*$ in the ring $F_2[x; a\mathbb{N}_0]_n$ is modulo ideal $((x^a)^n - 1)$. That is, for $c(x^a)$ in $F_2[x; a\mathbb{N}_0]_n$ the product $(x^a) * c(x^a)$ is given by

$$\begin{aligned} (x^a) * c(x^a) &= (x^a) * (c_0 + c_a(x^a) + c_{2a}(x^a)^2 + \dots + c_{a(n-1)}(x^a)^{n-1}) \\ &= c_{a(n-1)} + c_0(x^a) + c_a(x^a)^2 + \dots + c_{a(n-2)}(x^a)^{n-1}. \end{aligned} \quad (2.3)$$

In the following results a method of obtaining the generalized generator polynomial, which generates a principal ideal of the factor ring $F_2[x; a\mathbb{N}_0]_n$ is discussed.

The following Theorem shift [27, Theorem 4.3.5] to the monoid ring $F_2[x; a\mathbb{N}_0]$.

Theorem 17 *A subset \mathcal{C}_n of $F_2[x; a\mathbb{N}_0]_n$ is an n length binary cyclic code if and only if \mathcal{C}_n is an ideal in the ring $F_2[x; a\mathbb{N}_0]_n$.*

Proof. Assume that \mathcal{C}_n is an ideal in $F_2[x; a\mathbb{N}_0]_n$. Then \mathcal{C}_n can also be considered as a subspace of F_2 -space F_2^n . Thus, it is also closed under multiplication by any ring element, in particular under multiplication by x^a . Hence \mathcal{C}_n is a cyclic code. Conversely, if \mathcal{C}_n is a cyclic code, then \mathcal{C}_n is a linear code over F_2 . Hence, for all $f(x^a), g(x^a) \in \mathcal{C}_n$ and $\beta \in F_2$, it follows that $f(x^a) - g(x^a) \in \mathcal{C}_n$ and $\beta f(x^a) \in \mathcal{C}_n$. Further, since \mathcal{C}_n is cyclic, it follows that $x^a * f(x^a) \in \mathcal{C}_n$, for all $f(x^a) \in \mathcal{C}_n$. Thus, for every $r(x^a) \in F_2[x; a\mathbb{N}_0]_n$, it follows that $r(x^a) * f(x^a) \in \mathcal{C}_n$, and therefore, \mathcal{C}_n is an ideal in the ring $F_2[x; a\mathbb{N}_0]_n$. ■

The following Theorem converts [27, Theorem 4.3.6] for a monoid ring $F_2[x; a\mathbb{N}_0]$.

Theorem 18 *Let \mathcal{C}_n be a nonzero ideal in the ring $F_2[x; a\mathbb{N}_0]_n$. Then following hold.*

1. *There exists a unique monic generalized polynomial $g(x^a)$ of least degree in \mathcal{C}_n ,*

2. $g(x^a)$ divides $(x^a)^n - 1$ in $F_2[x; a\mathbb{N}_0]$,
3. For all $c(x^a) \in \mathcal{C}_n$, it follows that $g(x^a)$ divides $c(x^a)$ in $F_2[x; a\mathbb{N}_0]$, and
4. $\mathcal{C}_n = (g(x^a))$.

Conversely, if \mathcal{C}_n is the ideal generated by $p(x^a) \in F_2[x; a\mathbb{N}_0]_n$, then $p(x^a)$ is a generalized polynomial of least degree in \mathcal{C}_n if and only if $p(x^a)$ divides $(x^a)^n - 1$ in $F_2[x; a\mathbb{N}_0]$.

Proof. Proof is analogous to [27, Theorem 4.3.6]. ■

Next result for generator matrix is analogous to Theorem [27, Theorem 4.3.11].

Theorem 19 Let $\mathcal{C}_n \subset F_2[x; a\mathbb{N}_0]_n$ be a binary cyclic code with generator polynomial

$$g(x^a) = g_0 + g_a(x^a) + g_{2a}(x^a)^2 + \dots g_{ar}(x^a)^r, \quad g_{ar} = 1. \quad (2.4)$$

Then \mathcal{C}_n is of dimension $k(= n - r)$, which has a generator matrix of order $k \times n$ given by

$$G_r = \begin{bmatrix} g_0 & g_a & g_{2a} & \cdots & \cdots & g_{ar} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_a & \cdots & \cdots & g_{a(r-1)} & g_{ar} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & g_0 & g_a & \cdots & \cdots & g_{ar} \end{bmatrix}. \quad (2.5)$$

The following Corollary is a particular case of [27, Theorem 4.3.11].

Corollary 20 Let $\mathcal{C}_{an} \subset F_2[x]_{an}$ be a binary cyclic code with generator polynomial

$$g(x) = g_0 + g_1x^a + g_2x^{2a} + \dots + g_rx^{ar}, \quad g_r = 1. \quad (2.6)$$

Then \mathcal{C}_{an} is of dimension $ak = a(n - r)$, which has a generator matrix of order $ak \times an$ given

by:

$$G_{ar} = \begin{bmatrix} g_0 & 0 & \cdots & 0 & g_1 & 0 & \cdots & 0 & g_2 & \cdots & \cdots & g_r & 0 & 0 & \cdots & 0 \\ 0 & g_0 & 0 & \cdots & 0 & g_1 & 0 & \cdots & 0 & g_2 & \cdots & \cdots & g_r & 0 & \cdots & 0 \\ \vdots & \vdots & & & & & & & \vdots & & & & & & \vdots & \\ 0 & \cdots & 0 & g_0 & 0 & \cdots & 0 & g_1 & 0 & \cdots & 0 & g_2 & \cdots & \cdots & g_r & \end{bmatrix}. \quad (2.7)$$

The sequence $0 \cdots 0$ between g_i 's in G_{ar} has length $a - 1$.

Definition 21 Let $g(x^a)$ be a generator generalized polynomial of a binary cyclic code $\mathcal{C}_n \subset F_2[x; a\mathbb{N}_0]_n$. Then the k degree generalized polynomial $h(x^a)$ of $F_2[x; a\mathbb{N}_0]$, such that $(x^a)^n - 1 = g(x^a)h(x^a)$, is called the check generalized polynomial of \mathcal{C}_n .

Theorem 22 Let $\mathcal{C}_n \subset F_2[x; a\mathbb{N}_0]_n$ be a binary cyclic code with check generalized polynomial $h(x^a)$. Then $c(x^a) \in \mathcal{C}_n$ if and only if $c(x^a) * h(x^a) = 0$, where $c(x^a) \in F_2[x; a\mathbb{N}_0]_n$.

Proof. Let $g(x^a)$ be the generator generalized polynomial of \mathcal{C}_n . Then $g(x^a)h(x^a) = (x^a)^n - 1$ and $g(x^a) * h(x^a) = 0$. If $c(x^a) \in \mathcal{C}_n$ then by Theorem 18, $c(x^a) = q(x^a)g(x^a)$ for some $q(x^a)$ in $F_2[x; a\mathbb{N}_0]_n$ and so

$$c(x^a) * h(x^a) = q(x^a)g(x^a) * h(x^a) = 0.$$

Conversely, let $c(x^a) \in F_2[x; a\mathbb{N}_0]_n$ such that $c(x^a) * h(x^a) = 0$. This implies

$$c(x^a)h(x^a) = f(x^a)((x^a)^n - 1),$$

and

$$c(x^a)h(x^a) = f(x^a)g(x^a)h(x^a)$$

for some $f(x^a) \in F_2[x; a\mathbb{N}_0]_n$. Thus,

$$c(x^a) = f(x^a)g(x^a)$$

and hence $c(x^a) \in \mathcal{C}_n$. ■

For binary cyclic (n, k) code \mathcal{C}_n the following Theorem gives its parity check matrix.

Theorem 23 Let \mathcal{C}_n be a binary cyclic (n, k) code with check generalized polynomial

$$h(x^a) = h_0 + h_a(x^a) + h_{2a}(x^a)^2 + \dots + h_{a(k-1)}(x^a)^{k-1} + h_{ak}(x^a)^k, \quad h_k = 1. \quad (2.8)$$

Then the $(n - k) \times n$ matrix given by:

$$H_k = \begin{bmatrix} h_{ak} & h_{a(k-1)} & h_{a(k-2)} & \cdots & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_{ak} & h_{a(k-1)} & \cdots & \cdots & h_a & h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & & & & & & \vdots \\ 0 & 0 & \cdots & h_{ak} & h_{a(k-1)} & \cdots & \cdots & h_0 \end{bmatrix} \quad (2.9)$$

is a parity check matrix of \mathcal{C}_n .

Proof. Similar to [27, Theorem 4.3.14]. ■

The following Corollary is the particular case of [27, Theorem 4.3.14].

Corollary 24 Let \mathcal{C}_{an} be a binary cyclic (an, ak) code with check polynomial

$$h(x) = h_0 + h_1x^a + \dots + h_kx^{ak} \in \mathbb{F}_2[x], \quad h_k = 1 \quad (2.10)$$

Then the $(an - ak) \times an$ matrix

$$H_{ak} = \begin{bmatrix} h_k & 0 & \cdots & 0 & h_{k-1} & 0 & \cdots & 0 & h_{k-2} & \cdots & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_k & 0 & \cdots & 0 & h_{k-1} & 0 & \cdots & 0 & h_{k-2} & \cdots & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & & & \ddots & \vdots & & \ddots & & & \ddots & \vdots \\ 0 & \cdots & 0 & h_k & 0 & \cdots & 0 & h_{k-1} & 0 & \cdots & 0 & h_{k-2} & \cdots & \cdots & h_0 \end{bmatrix} \quad (2.11)$$

is a parity check matrix for \mathcal{C}_{an} and the sequence $0 \cdots 0$ between h_i 's in H_{ak} has length $a - 1$.

Example 25 Let $g(x^2) = 1 + (x^2) + (x^2)^2 \in F_2[x; 2\mathbb{N}_0]$ be the generalized polynomial with degree $r = 2$ and $g(x^2)$ divides $(x^2)^3 - 1$. Clearly $g(x^2)$ generates a binary cyclic $(3, 1)$ code in $F_2[x; 2\mathbb{N}_0]_3$ which has a generator matrix

$$G_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad (2.12)$$

In $F_2[x]$, the polynomial $g(x^2) = g(x) = 1 + x^2 + x^4$ has degree $4(= 2r)$ and divides $x^6 - 1$. So, $g(x)$ generates a binary cyclic $(6, 2)$ code in $F_2[x]_6$ which has a generator matrix

$$G_4 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (2.13)$$

Since $(x^2)^3 - 1 = (1 + x^2 + (x^2)^2)(1 + (x^2))$, it follows that $h(x^2) = 1 + (x^2)$ is the parity check generalized polynomial of $(3, 1)$ code in $F_2[x; 2\mathbb{N}_0]_3$. The parity check matrix is

$$H_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad (2.14)$$

In $F_2[x]$, $(x^2)^3 - 1$ becomes $x^6 - 1 = (1 + x^2 + x^4)(1 + x^2)$. Hence $h(x) = 1 + x^2$ is the parity check polynomial of $(6, 2)$ code in $F_2[x]_6$ and the corresponding parity check matrix is

$$H_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (2.15)$$

2.2 Cyclic codes as ideals in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$

Binary cyclic codes of length n are ideals in the factor ring $F_2[x]_n$. However, the fact $F_2[x] \subset F_2[x; \frac{1}{b}\mathbb{N}_0]$ supports the construction of binary cyclic codes in the factor ring $F_2[x; \frac{1}{b}\mathbb{N}_0]_{bn}$. Similarly, $F_2[x; a\mathbb{N}_0] \subset F_2[x; \frac{a}{b}\mathbb{N}_0]$ provides a justification for constructing binary cyclic codes in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ by using an n length cyclic code \mathcal{C}_n obtained from $F_2[x; a\mathbb{N}_0]_n$. Let $f(x^a) = f_0 + f_a(x^a) + f_{2a}(x^a)^2 + \dots + f_{an}(x^a)^n \in F_2[x; a\mathbb{N}_0]$ be a generalized polynomial of degree n , then $f(x^a)$ has degree bn in the monoid ring $F_2[x; \frac{a}{b}\mathbb{N}_0]$ and is represented by $f(x^{\frac{a}{b}}) = f_0 + f_{\frac{a}{b}}(x^{\frac{a}{b}})^b + f_{2\frac{a}{b}}(x^{\frac{a}{b}})^{2b} + \dots + f_{n\frac{a}{b}}(x^{\frac{a}{b}})^{bn}$. If $f(x^{\frac{a}{b}})$ is monic, then the factor ring $\frac{F_2[x; \frac{a}{b}\mathbb{N}_0]}{(f(x^{\frac{a}{b}}))}$ is the ring of residue classes of generalized polynomials in $F_2[x; \frac{a}{b}\mathbb{N}_0]$ modulo ideal $(f(x^{\frac{a}{b}}))$. Thus, if we take

$f(x^{\frac{a}{b}})$ to be $(x^{\frac{a}{b}})^{bn} - 1$, then the factor ring is

$$\frac{\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]}{((x^{\frac{a}{b}})^{bn} - 1)} = \{c_0 + c_{\frac{a}{b}}\beta + c_{2\frac{a}{b}}\beta^2 + \dots + c_{\frac{a}{b}(n-1)}\beta^{bn-1} : c_0, c_{\frac{a}{b}}, \dots, c_{\frac{a}{b}(n-1)} \in \mathbb{F}_2\}, \quad (2.16)$$

where β denotes the coset $x^{\frac{a}{b}} + ((x^{\frac{a}{b}})^{bn} - 1)$. Also, $f(\beta) = 0$, when β satisfies the relation $\beta^{bn} - 1 = 0$.

By writing $x^{\frac{a}{b}}$ in place of β , the ring $\frac{\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]}{((x^{\frac{a}{b}})^{bn} - 1)}$ becomes $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ in which the relation $(x^{\frac{a}{b}})^{bn} = 1$ holds. The factor ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ is algebra over the field \mathbb{F}_2 . The multiplication $*$ in the ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ is defined as, for $c(x^{\frac{a}{b}})$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ the product $(x^{\frac{a}{b}}) * c(x^{\frac{a}{b}})$ is given by:

$$\begin{aligned} (x^{\frac{a}{b}}) * c(x^{\frac{a}{b}}) &= (x^{\frac{a}{b}}) * (c_0 + c_{\frac{a}{b}}(x^{\frac{a}{b}}) + c_{2\frac{a}{b}}(x^{\frac{a}{b}})^2 + \dots + c_{\frac{a}{b}(n-1)}(x^{\frac{a}{b}})^{n-1}) \\ &= c_{\frac{a}{b}(n-1)} + c_0(x^{\frac{a}{b}}) + c_{\frac{a}{b}}(x^{\frac{a}{b}})^2 + \dots + c_{\frac{a}{b}(n-2)}(x^{\frac{a}{b}})^{n-1} \end{aligned}$$

Following results give a method of obtaining the generator generalized polynomial, which generates a principal ideal of the factor ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$.

Theorem 26 *A subset \mathcal{C}_{bn} in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ is a binary cyclic code if and only if \mathcal{C}_{bn} is an ideal in the ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$.*

Proof. Let \mathcal{C}_{bn} be an ideal in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. Then \mathcal{C}_{bn} can be considered as a vector subspace of \mathbb{F}_2 -space \mathbb{F}_2^{bn} . Since \mathcal{C}_{bn} is closed under multiplication defined in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, that is $x^{\frac{a}{b}} * c(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$, for all $c(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$ and $x^{\frac{a}{b}} \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, therefore \mathcal{C}_{bn} is a cyclic code. Conversely, suppose that \mathcal{C}_{bn} is a cyclic code, then \mathcal{C}_{bn} is a linear code over \mathbb{F}_2 . For all $c(x^{\frac{a}{b}}), d(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$ and $\gamma \in \mathbb{F}_2$, $c(x^{\frac{a}{b}}) - d(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$ and $\gamma c(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$. Further, \mathcal{C}_{bn} is cyclic, so $x^{\frac{a}{b}} * c(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$ for all $c(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$. This implies that $r(x^{\frac{a}{b}}) * c(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$ for every $r(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. Hence, \mathcal{C}_{bn} is an ideal in the ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. ■

The following Theorem extends Theorem 18 for the monoid ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$.

Theorem 27 *Let \mathcal{C}_{bn} be a nonzero ideal in the ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. Then the following holds.*

1. *There exists a unique monic generalized polynomial $g(x^{\frac{a}{b}})$ of least degree in \mathcal{C}_{bn} ,*
2. *$g(x^{\frac{a}{b}})$ divides $(x^{\frac{a}{b}})^{bn} - 1$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$,*

3. For all $c(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$, it follows that $g(x^{\frac{a}{b}})$ divides $c(x^{\frac{a}{b}})$ in $F_2[x; \frac{a}{b}\mathbb{N}_0]$, and

4. $\mathcal{C}_{bn} = (g(x^{\frac{a}{b}}))$.

Conversely, if \mathcal{C}_{bn} is the ideal generated by $p(x^{\frac{a}{b}}) \in F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, then $p(x^{\frac{a}{b}})$ is a generalized polynomial of least degree in \mathcal{C}_{bn} if and only if $p(x^{\frac{a}{b}})$ divides $(x^{\frac{a}{b}})^{bn} - 1$ in $F_2[x; \frac{a}{b}\mathbb{N}_0]$.

Similar to [27, Theorem 4.3.11], the following Theorem gives the generator matrix of the binary cyclic code \mathcal{C}_{bn} .

Theorem 28 Let $\mathcal{C}_{bn} \subset F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ be a binary cyclic code with generator polynomial

$$g(x^{\frac{a}{b}}) = g_0 + g_{\frac{a}{b}}(x^{\frac{a}{b}})^b + g_{2\frac{a}{b}}(x^{\frac{a}{b}})^{2b} + \cdots + g_{r\frac{a}{b}}(x^{\frac{a}{b}})^{br}, \quad g_{r\frac{a}{b}} = 1 \quad (2.17)$$

Then \mathcal{C}_{bn} is of dimension $bk = b(n - r)$, which has a generator matrix of order $bk \times bn$ given by:

$$G_{br} = \begin{bmatrix} g_0 & 0 & \cdots & 0 & g_{\frac{a}{b}} & 0 & \cdots & 0 & g_{2\frac{a}{b}} & \cdots & \cdots & g_{r\frac{a}{b}} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & 0 & \cdots & 0 & g_{\frac{a}{b}} & 0 & \cdots & 0 & g_{2\frac{a}{b}} & \cdots & \cdots & g_{r\frac{a}{b}} & 0 & \cdots & 0 \\ \vdots & \vdots & & & & & & & \vdots & & & & & & & \vdots \\ 0 & \cdots & 0 & g_0 & 0 & \cdots & 0 & g_{\frac{a}{b}} & 0 & \cdots & 0 & g_{2\frac{a}{b}} & \cdots & \cdots & g_{r\frac{a}{b}} \end{bmatrix} \quad (2.18)$$

The sequence $0 \cdots 0$ between g_i 's in G_{br} has length $b - 1$.

Corollary 29 [Theorem 19] Let $\mathcal{C}_{bn} \subset F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ be a binary cyclic code with generator polynomial

$$g(x^{\frac{a}{b}}) = g_0 + g_{\frac{a}{b}}(x^{\frac{a}{b}})^b + g_{2\frac{a}{b}}(x^{\frac{a}{b}})^{2b} + \cdots + g_{r\frac{a}{b}}(x^{\frac{a}{b}})^{br}, \quad g_{r\frac{a}{b}} = 1 \quad (2.19)$$

If $b = 1$, then $g(x^{\frac{a}{b}}) \in F_2[x; a\mathbb{N}_0]$ and generates a binary cyclic (n, k) code \mathcal{C}_n which has a $k \times n$ generator matrix

$$G_r = \begin{bmatrix} g_0 & g_a & g_{2a} & \cdots & \cdots & g_{ar} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_a & \cdots & \cdots & g_{a(r-1)} & g_{ar} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & g_0 & g_a & \cdots & \cdots & g_{ar} \end{bmatrix} \quad (2.20)$$

Definition 30 The generalized polynomial $h(x^{\frac{a}{b}})$, such that $(x^{\frac{a}{b}})^n - 1 = g(x^{\frac{a}{b}})h(x^{\frac{a}{b}})$, is called the check generalized polynomial of binary cyclic code $\mathcal{C}_{bn} \subset F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, where $g(x^{\frac{a}{b}})$ is the generator generalized polynomial of \mathcal{C}_{bn} .

Analogous to Theorem 22, the following Theorem is obtained for the binary cyclic code \mathcal{C}_{bn} in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$.

Theorem 31 Let \mathcal{C}_{bn} be a bn length binary cyclic code in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ with check generalized polynomial $h(x^{\frac{a}{b}})$. Then $a(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$, where $a(x^{\frac{a}{b}}) \in F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, if and only if $a(x^{\frac{a}{b}}) * h(x^{\frac{a}{b}}) = 0$.

Analogous to Theorem 23, the following Theorem is obtained for binary cyclic code \mathcal{C}_{bn} in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$.

Theorem 32 Let \mathcal{C}_{bn} be a binary cyclic (bn, bk) code with check generalized polynomial

$$h(x^{\frac{a}{b}}) = h_0 + h_{\frac{a}{b}}(x^{\frac{a}{b}})^b + \cdots + h_{\frac{a}{b}k}(x^{\frac{a}{b}})^{bk}, \quad h_{\frac{a}{b}k} = 1. \quad (2.21)$$

Then the $b(n-k) \times bn$ matrix given by:

$$H_{bk} = \begin{bmatrix} h_{\frac{a}{b}k} & 0 & \cdots & 0 & h_{\frac{a}{b}(k-1)} & \cdots & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_{\frac{a}{b}k} & 0 & \cdots & 0 & h_{\frac{a}{b}(k-1)} & \cdots & \cdots & h_0 & 0 & \cdots 0 \\ \vdots & \vdots & & & & & & \vdots & & & \\ 0 & \cdots & 0 & h_{\frac{a}{b}k} & 0 & \cdots & 0 & h_{\frac{a}{b}(k-1)} & \cdots & \cdots & h_0 \end{bmatrix} \quad (2.22)$$

is a parity check matrix for \mathcal{C}_{bn} and the sequence $0 \cdots 0$ in H_{bk} has length $b-1$.

Corollary 33 [Theorem 23] Let \mathcal{C}_{bn} be a binary cyclic (bn, bk) code with check generalized polynomial

$$h(x^{\frac{a}{b}}) = h_0 + h_{\frac{a}{b}}(x^{\frac{a}{b}})^b + \cdots + h_{\frac{a}{b}k}(x^{\frac{a}{b}})^{bk}, \quad h_{\frac{a}{b}k} = 1. \quad (2.23)$$

If $b = 1$, then $(n - k) \times n$ matrix

$$H_k = \begin{bmatrix} h_{ak} & h_{a(k-1)} & h_{a(k-2)} & \cdots & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_{ak} & h_{a(k-1)} & \cdots & \cdots & h_a & h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & h_{ak} & h_{a(k-1)} & \cdots & \cdots & h_0 \end{bmatrix} \quad (2.24)$$

is a parity check matrix of the binary cyclic code \mathcal{C}_n .

Now shift the generalized polynomial $f(x^{\frac{a}{b}})$ of arbitrary degree n in $F_2[x; \frac{a}{b}\mathbb{N}_0]$ to a generalized polynomial $f(x^{\frac{1}{b}})$ in $F_2[x; \frac{1}{b}\mathbb{N}_0]$ as

$$f(x^{\frac{1}{b}}) = f_0 + f_{\frac{1}{b}}(x^{\frac{1}{b}})^a + f_{\frac{2}{b}}(x^{\frac{1}{b}})^{2a} + \cdots + f_{\frac{n}{b}}(x^{\frac{1}{b}})^{an}. \quad (2.25)$$

Thus the degree of an arbitrary generalized polynomial in $F_2[x; \frac{a}{b}\mathbb{N}_0]$ has exceeds from n to an in $F_2[x; \frac{1}{b}\mathbb{N}_0]$. Consequently, the degree of generator generalized polynomial $g((x^{\frac{1}{b}}))$ also exceeds from $r' = br$ to $r'' = abr$, where $g(x^{\frac{1}{b}})$ divides $(x^{\frac{1}{b}})^{abn} - 1$ and generates a binary cyclic (abn, abk) code \mathcal{C}_{abn} in $F_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$.

Thus from the generator and parity check matrices of the code \mathcal{C}_{bn} we obtain the generator and parity check matrices of the code \mathcal{C}_{abn} .

Theorem 34 Let $\mathcal{C}_{abn} \subset F_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$ be a binary cyclic code with generator polynomial

$$g((x^{\frac{1}{b}})) = g_0 + g_{\frac{1}{b}}(x^{\frac{1}{b}})^{ab} + g_{\frac{2}{b}}(x^{\frac{1}{b}})^{2b} + \cdots + g_{\frac{r}{b}}(x^{\frac{a}{b}})^{br}, \quad g_{\frac{r}{b}} = 1 \quad (2.26)$$

Then \mathcal{C}_{abn} is of dimension $abk = ab(n - r)$, which has a generator matrix of order $abk \times abn$ given by

$$G_{abr} = \begin{bmatrix} g_0 & 0 & \cdots & 0 & g_{\frac{1}{b}} & 0 & \cdots & 0 & g_{\frac{2}{b}} & \cdots & \cdots & g_{\frac{r}{b}} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & 0 & \cdots & 0 & g_{\frac{1}{b}} & 0 & \cdots & 0 & g_{\frac{2}{b}} & \cdots & \cdots & g_{\frac{r}{b}} & 0 & \cdots & 0 \\ \vdots & \vdots & & & & & & & \vdots & & & & & & & \vdots \\ 0 & \cdots & 0 & g_0 & 0 & \cdots & 0 & g_{\frac{1}{b}} & 0 & \cdots & 0 & g_{\frac{2}{b}} & \cdots & \cdots & g_{\frac{r}{b}} \end{bmatrix} \quad (2.27)$$

where the sequence $0 \cdots 0$ between g_i 's in G_{abr} has length $ab - 1$.

Theorem 35 Let \mathcal{C}_{abn} be a binary cyclic (abn, abk) code with check generalized polynomial

$$h(x^{\frac{1}{b}}) = h_0 + h_{\frac{1}{b}}(x^{\frac{1}{b}})^{ab} + \cdots + h_{\frac{k}{b}}(x^{\frac{1}{b}})^{abk}, \quad h_{\frac{k}{b}} = 1. \quad (2.28)$$

Then the $ab(n - k) \times abn$ matrix given by

$$H_{abk} = \begin{bmatrix} h_{\frac{k}{b}} & 0 & \cdots & 0 & h_{\frac{(k-1)}{b}} & \cdots & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_{\frac{k}{b}} & 0 & \cdots & 0 & h_{\frac{(k-1)}{b}} & \cdots & \cdots & h_0 & 0 & \cdots 0 \\ \vdots & \vdots & & & & & & \vdots & & & \\ 0 & \cdots & 0 & h_{\frac{k}{b}} & 0 & \cdots & 0 & h_{\frac{(k-1)}{b}} & \cdots & \cdots & h_0 \end{bmatrix} \quad (2.29)$$

is a parity check matrix for \mathcal{C}_{abn} and the sequence $0 \cdots 0$ between h_i 's in H_{abk} has length $ab - 1$.

Example 36 Let $g(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^6$ be a generator generalized polynomial of degree $br = 6$ and divides $(x^{\frac{2}{3}})^9 - 1$, then $g(x^{\frac{2}{3}})$ generates a binary cyclic $(9, 3)$ code with generator matrix

$$G_6 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (2.30)$$

Whereas, in $F_2[x; \frac{1}{3}\mathbb{N}_0]$, $g(x^{\frac{2}{3}})$ becomes $g((x^{\frac{1}{3}})) = 1 + (x^{\frac{1}{3}})^6 + (x^{\frac{1}{3}})^{12}$ and has degree 12 and divides $(x^{\frac{1}{3}})^{18} - 1$. Thus, in generator generalized polynomial $g(x^{\frac{1}{3}})$ every exponent of the indeterminate $x^{\frac{1}{3}}$ is a multiple of 2, and it generates a cyclic $(18, 6)$ code having generator matrix

$$G_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.31)$$

Since, $(x^{\frac{2}{3}})^9 - 1 = (1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^6)(1 + (x^{\frac{2}{3}})^3)$ and $(x^{\frac{1}{3}})^{18} - 1 = (1 + (x^{\frac{1}{3}})^6 + (x^{\frac{1}{3}})^{12})(1 + (x^{\frac{1}{3}})^6)$, it

follows that the parity check generalized polynomials $h(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3$ and $h((x^{\frac{1}{3}})) = 1 + (x^{\frac{1}{3}})^6$ give the following parity check matrices

$$\begin{aligned}
 H_3 &= \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \text{ and} \\
 H_4 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.32}
 \end{aligned}$$

2.3 Relationship among cyclic codes C_n, C_{an}, C_{bn} and C_{abn}

In this section, we demonstrate the association between the binary cyclic codes C_n, C_{an}, C_{bn} and C_{abn} by two ways:

- (1) Using technique of interleaving.
- (2) Through generator and parity check matrices of binary cyclic codes C_n, C_{an}, C_{bn} and C_{abn} .

2.3.1 Relationship of C_n, C_{an}, C_{bn} and C_{abn} by interleaving

For a given (n, k) cyclic code, a $(\beta n, \beta k)$ cyclic code can be constructed by interleaving. This is done by simply arranging β code vectors in the original code into β rows of a rectangular array and then transmitting them column by column. In this way a codeword of βn digits is obtained whose two consecutive bits are now separated by $\beta - 1$ positions. The parameter β is called *interleaving degree*.

Proposition 37 *The codes C_{an}, C_{bn} and C_{abn} are interleaved codes of degree a, b and ab respectively, where the code C_n is the base code.*

Proof. Take a code vectors from the base code C_n and arrange them into a rows of an $a \times n$ array. Then by transmitting this code array column by column in serial manner we get the binary cyclic code C_{an} . Similarly, the binary cyclic code C_{bn} is obtained by taking b code vectors from the base code C_n , arranging them into b rows of an $b \times n$ array and then transmitting it column by column in serial manner. In this way codewords of an and bn digits are obtained whose two consecutive bits are now separated by $a - 1$ and $b - 1$ positions respectively. Now, by arranging ab code vectors from the code C_n and arranging them into ab rows of an $ab \times n$ array and then transmitting it column by column, the binary cyclic code C_{abn} is obtained. This gives codewords of abn digits whose two consecutive bits are separated by $ab - 1$ positions. ■

Hence, the codes C_{an}, C_{bn} and C_{abn} are interleaved codes of degree a, b and ab respectively.

Example 38 *In Examples 25 and 36, the $(3, 1)$ code C_3 acts as a base code. The code C_6 is obtained by arranging 2 codewords 111 and 000 in C_3 into 2 rows of an 2×3 array, that is:*

$$\begin{array}{ccc} 1 & 1 & 1 \\ 0 & 0 & 0 \end{array}, \quad (2.33)$$

and then by transmitting this code array column by column we get 101010, which is a codeword in C_6 . Similarly, by arranging 3 and 6 codewords in C_3 into 3 and 6 rows of an 3×3 and 6×3

arrays, that is:

$$\begin{array}{ccc}
 & & 0 \ 0 \ 0 \\
 & & 1 \ 1 \ 1 \\
 1 \ 1 \ 1 & & 0 \ 0 \ 0 \\
 0 \ 0 \ 0 & \text{and} & 0 \ 0 \ 0 \\
 1 \ 1 \ 1 & & 0 \ 0 \ 0 \\
 & & 1 \ 1 \ 1
 \end{array} . \tag{2.34}$$

Then transmitting them column by column we get codewords 101101101 and 010001010001010001 in \mathcal{C}_9 and \mathcal{C}_{18} .

2.3.2 Relationship of $\mathcal{C}_n, \mathcal{C}_{an}, \mathcal{C}_{bn}$ and \mathcal{C}_{abn} by generator and parity check matrices

Now, we explain the relationship between the codes $\mathcal{C}_n, \mathcal{C}_{an}, \mathcal{C}_{bn}$ and \mathcal{C}_{abn} through their generator and parity check matrices, using the notion of direct sum of codes.

The following definition of direct sum of the codes is taken from [17].

Definition 39 (a) Let \mathcal{C}_i be an (n_i, k_i) code, where $i \in \{1, 2\}$, both having symbols from the same Galois field F_q . Then their direct sum

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(c_1, c_2) \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\} \tag{2.35}$$

is a $(n_1 + n_2, k_1 + k_2)$ code.

(b) For $i \in \{1, 2\}$, if \mathcal{C}_i has generator matrix G_i and parity check matrix H_i , then

$$G_1 \oplus G_2 = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \text{ and } H_1 \oplus H_2 = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}, \tag{2.36}$$

respectively are the generator and parity check matrices for the code $\mathcal{C}_1 \oplus \mathcal{C}_2$.

The following result explains the relationship between the binary cyclic codes $\mathcal{C}_n, \mathcal{C}_{an}, \mathcal{C}_{bn}$ and \mathcal{C}_{abn} through their generator matrices.

Theorem 40 Let G_r, G_{ar}, G_{br} , and G_{abr} , be the generator matrices corresponding to the generator generalized polynomials

$$\begin{aligned} g(x^a) &= 1 + (x^a) + \cdots + (x^a)^r, \quad g(x) = 1 + x^a + \cdots + x^{ar}, \\ g(x^{\frac{a}{b}}) &= 1 + (x^{\frac{a}{b}})^b + \cdots + (x^{\frac{a}{b}})^{br} \quad \text{and} \quad g((x^{\frac{1}{b}})^a) = 1 + (x^{\frac{1}{b}})^{ab} + \cdots + (x^{\frac{1}{b}})^{abr}, \end{aligned}$$

of binary cyclic codes $\mathcal{C}_n, \mathcal{C}_{an}, \mathcal{C}_{bn}$ and \mathcal{C}_{abn} in $F_2[x; a\mathbb{N}_0]_n, F_2[x]_{an}, F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}, F_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$.

Then the following conditions hold.

- 1) $G_{ar} \sim \oplus_1^a G_r$,
- 2) $G_{br} \sim G_r \oplus G_{ar} \sim \oplus_1^b G_r$, and
- 3) $G_{abr} \sim \oplus_1^a G_{br} \sim \oplus_1^a G_r \oplus G_{ar} \sim \oplus_1^{ab} G_r$.

Proof. As $g(x^a) = 1 + (x^a) + \cdots + (x^a)^r$ divides $(x^a)^n - 1$ in $F_2[x; a\mathbb{N}_0]$, therefore the generator matrix G_r has order $k \times n$, where $k = n - r$. In $F_2[x]$ the generalized polynomial $g(x^a) = g(x) = 1 + x^a + \cdots + x^{ar}$ and divides $x^{an} - 1$. Consequently, a generator matrix G_{ar} of order $ak \times an$ is obtained which after some suitable column operations becomes

$$G_{ar} \sim \begin{bmatrix} G_r & 0 & \cdots & 0 \\ 0 & G_r & 0 \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & G_r \end{bmatrix}_{a(k \times n)} \quad (1)$$

This implies that G_{ar} contains a blocks of G_r at its main diagonal and hence $G_{ar} \sim \oplus_1^a G_r$. Similarly, $g(x^{\frac{a}{b}}) = 1 + (x^{\frac{a}{b}}) + \cdots + (x^{\frac{a}{b}})^{br}$ divides $x^{bn} - 1$, which have generator matrix G_{br} of order $bk \times bn$. On applying suitable column operations, blocks of G_{ar} and G_r are obtained at main diagonal of G_{br}

$$G_{br} \sim \begin{bmatrix} G_{ar} & 0 \\ 0 & G_r \end{bmatrix}_{(a+1)(k \times n)} \quad (2)$$

Putting the value of G_{ar} from (1) in (2) we get,

$$G_{br} \sim \begin{bmatrix} G_r & 0 & \cdots & 0 \\ 0 & G_r & 0 \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & G_r \end{bmatrix}_{(a+1)(k \times n)} \quad (3)$$

This shows that G_{br} contains b blocks of G_r , that is, $G_{br} \sim \oplus_1^b G_r$. Finally, $g((x^{\frac{1}{b}})^a) = 1 + (x^{\frac{1}{b}})^{ab} + \cdots + (x^{\frac{1}{b}})^{abr}$ divides $x^{abn} - 1$, which gives generator matrix G_{abr} of order $abk \times abn$ which after suitable column operations gives

$$G_{abr} \sim \begin{bmatrix} G_{br} & 0 & \cdots & 0 \\ 0 & G_{br} & 0 \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{br} \end{bmatrix}_{a(bk \times bn)} \quad (4)$$

Putting the value of G_{br} from (2) and (3), we get

$$G_{abr} \sim \begin{bmatrix} G_{ar} & 0 & 0 & \cdots & 0 \\ 0 & G_r & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \cdots & 0 \\ \vdots & 0 & \cdots & G_{ar} & \vdots \\ 0 & 0 & 0 & \cdots & G_r \end{bmatrix}_{a(bk \times bn)} \quad (5)$$

$$\sim \begin{bmatrix} G_r & 0 & \cdots & 0 \\ 0 & G_r & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & G_r \end{bmatrix}_{ab(k \times n)},$$

which shows G_{abr} contains ab blocks of G_r , that is, $G_{abr} \sim \oplus_1^{ab} G_r$. ■

The following example illustrates Theorem 40.

Example 41 Let $a = 2$, $b = 3$ and $r = 2$. From Example 25 and Example 36 we get

$$G_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.37)$$

After some suitable column operations on G_{12} we have

$$\begin{aligned} G_{12} &\sim \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \\ &\sim G_6 \oplus G_6 \end{aligned} \quad (2.38)$$

On applying suitable column operations on G_6 , it gives

$$\begin{aligned} G_6 &\sim \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \\ &\sim G_4 \oplus G_2 \end{aligned} \quad (2.39)$$

and similarly G_4 becomes

$$\begin{aligned} G_4 &\sim \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \\ &\sim G_2 \oplus G_2. \end{aligned} \quad (2.40)$$

So,

$$G_6 \sim G_2 \oplus G_2 \oplus G_2 \text{ and} \quad (2.41)$$

$$G_{12} \sim G_2 \oplus G_2 \oplus G_2 \oplus G_2 \oplus G_2 \oplus G_2.$$

Encoding: In the matrix G_{abr} , the matrices G_{br} , G_{ar} and G_r exist as block matrices and the generator generalized polynomial of the cyclic (abn, abk) code \mathcal{C}_{abn} can be used for encoding. So, a message word $u \in \mathbb{F}_2^{abk}$ is encoded as uG_{abr} . Hence the code

$$\mathcal{C}_{abr} = \{uG_{abr} : u \in \mathbb{F}_2^{abk}\}. \quad (2.42)$$

On partitioning u as

$$u = (u_{1 \times b} : u_{1 \times a} : u_{1 \times k}), \quad (2.43)$$

where $u_{1 \times b} \in \mathbb{F}_2^{bk}$, $u_{1 \times a} \in \mathbb{F}_2^{ak}$ and $u_{1 \times k} \in \mathbb{F}_2^k$, we get

$$\mathcal{C}_{abr} \sim \{u_{1 \times b}G_{br} : u_{1 \times a}G_{ar} : u_{1 \times k}G_r\}. \quad (2.44)$$

Example 42 Let $a = 2$, $b = 3$ and $r = 2$, then $u \in \mathbb{F}_2^6$ is given by

$$u = [1 \ 1 \ 0 \ 0 \ 1 \ 1]. \quad (2.45)$$

The row matrix u has order 1×6 . By partitioning the matrix u we get

$$u = [1 \ 1 \ 0]_{1 \times 3} : [0 \ 1]_{1 \times 2} : [1]_{1 \times 1} \quad (2.46)$$

$$= [u_1 : u_2 : u_3] \text{ and}$$

$$\begin{aligned} uG_{12} &= [u_1G_{6(3 \times 9)} : u_2G_{4(2 \times 6)} : u_3G_{2(1 \times 3)}] \\ &= 110110110010101111 \end{aligned} \quad (2.47)$$

Thus, the message word u is encoded as the codeword uG_{12} .

For parity check matrix, Theorem 40 doesn't hold, whereas it holds for the canonical parity check matrix.

The generator and parity check matrices of a binary cyclic code \mathcal{C}_{bn} , described above, are not in canonical forms. In general, for a linear code, a generator matrix G is transformed into canonical form by applying elementary row operations. But, in the case of a cyclic code, the canonical form can be obtained by using the generator generalized polynomial and the division algorithm in the Euclidean domain $F_2[x; \frac{a}{b}\mathbb{N}_0]$.

For any generalized polynomial $f(x^{\frac{a}{b}}) \in F_2[x; \frac{a}{b}\mathbb{N}_0]$, let $rem_{g(x^{\frac{a}{b}})}(f(x^{\frac{a}{b}}))$ denotes the remainder on dividing $f(x^{\frac{a}{b}})$ by $g(x^{\frac{a}{b}})$. For the sake of simplicity we denote it as $r(f(x^{\frac{a}{b}}))$.

Theorem 43 *Let $g(x^{\frac{a}{b}})$ be the generator generalized polynomial of a binary cyclic (bn, bk) code \mathcal{C}_{bn} in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ and A_{br} be a $bk \times b(n-k)$ matrix whose i -th row is $r((x^{\frac{a}{b}})^{b(n-k)+i-1})$, for $i = 1, \dots, k$. Then the canonical generator and parity check matrices of \mathcal{C}_{bn} respectively are*

$$G_{br} = \begin{bmatrix} I_{bk} & \vdots & A_{br} \end{bmatrix} \text{ and } H_{bk} = \begin{bmatrix} (A_{br})^T & \vdots & I_{b(n-k)} \end{bmatrix} \quad (2.48)$$

Proof. Since $deg(g(x^{\frac{a}{b}})) = b(n-k)$, it follows that $r(x^{\frac{a}{b}})^j = (x^{\frac{a}{b}})^j$ for $j < b(n-k)$. Moreover, $g(x^{\frac{a}{b}})$ divides $(x^{\frac{a}{b}})^{bn} - 1$ and $r(x^{\frac{a}{b}})^{bn+j} = r(x^{\frac{a}{b}})^j$ for all $j \geq 0$. Thus, we have to compute $r(x^{\frac{a}{b}})^j$ only for $j = b(n-k), \dots, bn-1$. Let

$$g_i(x^{\frac{a}{b}}) = (x^{\frac{a}{b}})^{i-1} - (x^{\frac{a}{b}})^{bk} r((x^{\frac{a}{b}})^{b(n-k)+i-1})$$

for $i = 1, 2, \dots, bk$. Then $deg(g_i(x^{\frac{a}{b}})) < bn$, so, $g_i(x^{\frac{a}{b}}) \in F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. Furthermore,

$$(x^{\frac{a}{b}})^{b(n-k)+i-1} - r((x^{\frac{a}{b}})^{b(n-k)+i-1}) \in \mathcal{C}_{bn},$$

therefore

$$g_i(x^{\frac{a}{b}}) = (x^{\frac{a}{b}})^{bk} * (x^{\frac{a}{b}})^{b(n-k)+i-1} - r((x^{\frac{a}{b}})^{b(n-k)+i-1}) \in \mathcal{C}_{bn}.$$

Let G_{br} be the $(bk \times bn)$ matrix whose i -th row is $g_i(x^{\frac{a}{b}})$, written as a row vector, $i = 1, \dots, bk$. Then

$$G_{br} = \begin{bmatrix} I_{bk} & \vdots & A_{br} \end{bmatrix},$$

where A_{br} is a $bk \times b(n-k)$ matrix whose i -th row is $r((x^{\frac{a}{b}})^{b(n-k)+i-1})$. Taking transpose of A_{br} we get a $b(n-k) \times bk$ matrix, whose i -th column is $r((x^{\frac{a}{b}})^{b(n-k)+i-1})$, written as a column

vector, $i = 1, \dots, bk$. Then

$$H_{bk} = \begin{bmatrix} A_{br}^T & \vdots & I_{b(n-k)} \end{bmatrix},$$

which is a $b(n-k) \times bn$ matrix. ■

Theorem 44 *Let A_r , A_{ar} , A_{br} and A_{abr} be the matrices as taken in Theorem 43 with respect to the corresponding generator (generalized) polynomials $g(x^a)$, $g(x)$, $g(x^{\frac{a}{b}})$ and $g((x^{\frac{1}{b}}))$ in $F_2[x; a\mathbb{N}_0]_n$, $F_2[x]_{an}$, $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ and $F_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$ respectively. Then*

- 1) $A_{ar} \sim \oplus_1^a A_r$,
- 2) $A_{br} \sim A_r \oplus A_{ar} \sim \oplus_1^b A_r$, and
- 3) $A_{abr} \sim \oplus_1^a A_{br} \sim \oplus_1^a A_r \oplus A_{ar} \sim \oplus_1^{ab} A_r$.

Proof. For the generator generalized polynomial

$$g(x^a) = 1 + (x^a) + \dots + (x^a)^r, \quad (2.49)$$

the remainders $r(x^a)^j$, where $n-k \leq j \leq n-1$ give the matrix A_r of order $k \times (n-k)$. Similarly, for

$$g(x) = 1 + x^a + \dots + x^{ar} \quad (2.50)$$

, the matrix A_{ar} of order $ak \times a(n-k)$ is obtained through the remainders $r(x^j)$, where $a(n-k) \leq j \leq an-1$. After applying suitable column operations on A_{ar} , it gives

$$\begin{aligned} A_{ar} &\sim \begin{bmatrix} A_r & 0 & 0 \cdots & 0 \\ 0 & A_r & 0 \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & A_r \end{bmatrix}_{a(k \times n-k)} \\ &\sim \oplus_1^a A_r. \end{aligned} \quad (2.51)$$

Corresponding to the generator generalized polynomial

$$g(x^{\frac{a}{b}}) = 1 + (x^{\frac{a}{b}})^b + \dots + (x^{\frac{a}{b}})^{br}, \quad (2.52)$$

the remainders $r((x^{\frac{a}{b}})^j)$ gives the matrix A_{br} of order $bk \times b(n-k)$, where $b(n-k) \leq j \leq b(n-1)$.

On applying suitable column operations on it, it gives submatrices of size $ak \times a(n-k)$ and $k \times n-k$, that is,

$$\begin{aligned}
A_{br} &\sim \begin{bmatrix} A_{ar} & O \\ O & A_r \end{bmatrix}_{(a+1)(k \times n-k)} \\
&\sim A_r \oplus A_{ar} \\
&\sim \oplus_1^{a+1=b} A_r.
\end{aligned} \tag{2.53}$$

Finally, for

$$g((x^{\frac{1}{b}})^a) = 1 + (x^{\frac{1}{b}})^{ab} + \dots + (x^{\frac{1}{b}})^{abr}, \tag{2.54}$$

the remainders $r((x^{\frac{1}{b}})^j)$, where $ab(n-k) \leq j \leq ab(n-1)$ gives A_{abr} of order $abk \times ab(n-k)$.

Which on applying suitable column operations gives submatrices of size $bk \times b(n-k)$, that is,

$$\begin{aligned}
A_{abr} &\sim \begin{bmatrix} A_{br} & 0 \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & A_{br} \end{bmatrix}_{a(bk \times b(n-k))} \\
&\sim \oplus_1^a A_{br} \sim \oplus_1^a A_r \oplus A_{ar} \\
&\sim \oplus_1^{ab} A_r,
\end{aligned} \tag{2.55}$$

which proves the theorem. ■

Similar results for canonical generator and parity check matrices are obtain by Theorem 44.

The following example illustrates Theorem 44.

Example 45 *To find the parity check matrix for $(18, 6)$ code obtained by the monoid ring $F_2[x; \frac{1}{3}\mathbb{N}_0]$, we first divide $(x^{\frac{1}{3}})^j$ by $g((x^{\frac{1}{3}})) = 1 + (x^{\frac{1}{3}})^6 + (x^{\frac{1}{3}})^{12}$, where $j = 12, 13, \dots, 17$, to get the remainders*

$$\begin{aligned}
r(x^{\frac{1}{3}})^{12} &= 1 + (x^{\frac{1}{3}})^6, \quad r(x^{\frac{1}{3}})^{13} = (x^{\frac{1}{3}}) + (x^{\frac{1}{3}})^7, \\
r(x^{\frac{1}{3}})^{14} &= (x^{\frac{1}{3}})^2 + (x^{\frac{1}{3}})^8, \quad r(x^{\frac{1}{3}})^{15} = (x^{\frac{1}{3}})^3 + (x^{\frac{1}{3}})^9, \\
r(x^{\frac{1}{3}})^{16} &= (x^{\frac{1}{3}})^4 + (x^{\frac{1}{3}})^{10}, \quad r(x^{\frac{1}{3}})^{17} = (x^{\frac{1}{3}})^5 + (x^{\frac{1}{3}})^{11}.
\end{aligned}$$

Therefore,

$$A_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.56)$$

Accordingly,

$$H_{12} = \begin{bmatrix} (A_{12})^T & \vdots & I_{12} \end{bmatrix}. \quad (2.57)$$

Similarly,

$$A_6 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \text{ gives} \quad (2.58)$$

$$H_6 = \begin{bmatrix} (A_6)^T & \vdots & I_6 \end{bmatrix}, \quad H_4 = \begin{bmatrix} (A_4)^T & \vdots & I_4 \end{bmatrix} \text{ and } H_2 = \begin{bmatrix} (A_2)^T & \vdots & I_2 \end{bmatrix} \quad (2.59)$$

Thus by Theorem 44,

$$H_{12} = \begin{bmatrix} \oplus_1^2 (A_6)^T & \vdots & I_{12} \end{bmatrix}, \quad H_6 = \begin{bmatrix} (A_2)^T \oplus (A_4)^T & \vdots & I_6 \end{bmatrix}, \quad H_4 = \begin{bmatrix} \oplus_1^2 A_2 & \vdots & I_4 \end{bmatrix}. \quad (2.60)$$

2.4 Decoding procedure

The codes \mathcal{C}_n , \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} have same minimum distance and hence same error correction capability along with the same code rate, but the codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} are interleaved codes of degree a, b and ab , where the base code \mathcal{C}_n is cyclic. Thus, if the initial code \mathcal{C}_n is capable of correcting t errors, then the interleaved codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} are capable of correcting t bursts of length a, b and ab or less, no matter where it starts, will affect no more than t bits in each row. This t bits error in each row will be corrected by the base code \mathcal{C}_n . If \mathcal{C}_n is capable of correcting all bursts of length l or less, then the interleaved codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} are capable

of correcting all bursts of length al , bl and abl or less.

We give decoding scheme only for the code \mathcal{C}_{bn} , through which decoding of \mathcal{C}_n and \mathcal{C}_{an} can easily be obtained. Decoding of the code \mathcal{C}_{abn} can be obtained by shifting $(x^{\frac{a}{b}})$ to $(x^{\frac{1}{b}})^a$.

The following theorem gives syndrome for binary cyclic codes \mathcal{C}_{bn} through its canonical parity check matrices H_{bk} .

Theorem 46 *Let \mathcal{C}_{bn} be a binary cyclic (bn, bk) code in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ with generator polynomial $g(x^{\frac{a}{b}})$ and the canonical parity check matrix H_{bk} . Then, for any vector $c \in F_2^{bn}$, the syndrome*

$$S(c) = r((x^{\frac{a}{b}})^{b(n-k)} c(x^{\frac{a}{b}})). \quad (2.61)$$

Proof. By Theorem 43, $H_{bk}^T = \begin{bmatrix} A_{br} \\ \dots \\ I_{b(n-k)} \end{bmatrix}$, where A_{br} is a $bk \times b(n-k)$ matrix whose i -th row is $r((x^{\frac{a}{b}})^{b(n-k)+i-1})$, for $i = 1, \dots, k$. The i -th row of the identity matrix $I_{b(n-k)}$ is $(x^{\frac{a}{b}})^{bn+i}$, for $i = 1, \dots, b(n-k)$. Hence, the relation

$$r((x^{\frac{a}{b}})^{bn+j}) = r((x^{\frac{a}{b}})^j)$$

for all $j \geq 0$, gives the i -th row of $(H_{bk})^T$ given by $r((x^{\frac{a}{b}})^{b(n-k)+i-1})$, where $i = 1, \dots, bn$. If

$$c = (c_0, c_{\frac{a}{b}}, \dots, c_{\frac{a}{b}(bn-1)}) \in \mathbb{F}_2^{bn},$$

then

$$c(x^{\frac{a}{b}}) = c_0 + c_{\frac{a}{b}}(x^{\frac{a}{b}}) + \dots + c_{\frac{a}{b}(bn-1)}(x^{\frac{a}{b}})^{(bn-1)} \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}.$$

Thus,

$$\begin{aligned} S(c) &= \begin{bmatrix} c_0 & c_{\frac{a}{b}} & \dots & c_{\frac{a}{b}(bn-1)} \end{bmatrix} H_{bk}^T \\ &= \sum_{i=1}^{bn} c_{\frac{a}{b}(i-1)} r((x^{\frac{a}{b}})^{(b(n-k)+i-1})) \\ &= r\left(\sum_{i=1}^{bn} c_{\frac{a}{b}(i-1)} (x^{\frac{a}{b}})^{(b(n-k)+i-1})\right) \\ &= r\left((x^{\frac{a}{b}})^{b(n-k)} c(x^{\frac{a}{b}})\right), \end{aligned}$$

which proves the theorem. ■

In a similar way, we get the syndromes for binary cyclic codes \mathcal{C}_{abn} and \mathcal{C}_{an} through their canonical parity check matrices H_{abk} and H_{ak} .

Theorem 47 *Let \mathcal{C}_{abn} be a binary cyclic (abn, abk) code in $F_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$ with generator polynomial $g(x^{\frac{1}{b}})$ and the canonical parity check matrix H_{abk} . Then, for any vector $c \in F_2^{bn}$, the syndrome*

$$S(c) = r((x^{\frac{1}{b}})^{ab(n-k)}c(x^{\frac{1}{b}})). \quad (2.62)$$

Theorem 48 *Let \mathcal{C}_{an} be a binary cyclic (an, ak) code in $F_2[x; \mathbb{N}_0]_{an}$ with generator polynomial $g(x)$ and the canonical parity check matrix H_{ak} . Then, for any vector $c \in F_2^{an}$, the syndrome*

$$S(c) = r((x)^{a(n-k)}c(x)). \quad (2.63)$$

In a binary cyclic code \mathcal{C}_{bn} , with generator generalized polynomial $g(x^{\frac{a}{b}})$, two vectors $c, d \in F_2^{bn}$ lie in the same coset if and only if $g(x^{\frac{a}{b}})$ divides $c(x^{\frac{a}{b}}) - d(x^{\frac{a}{b}})$, that is,

$$r(c(x^{\frac{a}{b}})) = r(d(x^{\frac{a}{b}})). \quad (2.64)$$

Let $v(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$ be a generalized code polynomial, and $u(x^{\frac{a}{b}})$ be a generalized received polynomial. Then,

$$v(x^{\frac{a}{b}}) = u(x^{\frac{a}{b}}) - e(x^{\frac{a}{b}}), \quad (2.65)$$

where $e(x^{\frac{a}{b}})$ is an error generalized polynomial. Therefore,

$$S(v) = S(u) - S(e) \text{ implies } S(u) = S(e) \text{ as } S(v) = 0. \quad (2.66)$$

Therefore, based on the previous discussion, we deduce the following decoding steps.

Decoding Steps

1. For received vector $u = (u_0, u_{\frac{a}{b}}, \dots, u_{\frac{a}{b}(bn-1)}) \in \mathbb{F}_2^{bn}$ with generalized received polynomial

$$u(x^{\frac{a}{b}}) = u_0 + u_{\frac{a}{b}}(x^{\frac{a}{b}}) + \dots + u_{\frac{a}{b}(bn-1)}(x^{\frac{a}{b}})^{(bn-1)}, \quad (2.67)$$

find the syndrome

$$S(u) = r((x^{\frac{a}{b}})^{b(n-k)}u(x^{\frac{a}{b}})). \quad (2.68)$$

2. Construct a syndrome table for generalized error polynomials.
3. Verify by the table that for which i , where $1 \leq i \leq n-1$, $S(u) = S(e_i)$. Then the generalized error polynomial $e_i(x^{\frac{a}{b}})$ for the generalized received polynomial $u(x^{\frac{a}{b}})$ is obtained.
4. Consequently, $v(x^{\frac{a}{b}}) = u(x^{\frac{a}{b}}) - e(x^{\frac{a}{b}})$ is the generalized decoded code polynomial of the binary cyclic code \mathcal{C}_{bn} .
5. The received interleaved sequence in \mathcal{C}_{bn} is de-interleaved and rearranged back to a rectangular array of b rows of the binary cyclic code \mathcal{C}_n . Then each row is decoded based on binary cyclic code \mathcal{C}_n .

Illustration

In Examples 25 and 36, the $(3, 1)$ code act as a base code capable of correcting single error.

Let $n = 9$, $k = 3$ and

$$g(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^6 \in \mathbb{F}_2[x; \frac{2}{3}\mathbb{N}_0]_{3n} \quad (2.69)$$

be the generator generalized polynomial. Let $u = 110000100$ be the received vector. Then

$$u(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}}) + (x^{\frac{2}{3}})^6 \in \mathbb{F}_2[x; \frac{2}{3}\mathbb{N}_0]_9 \quad (2.70)$$

is its corresponding generalized polynomial. The syndrome of $u(x^{\frac{2}{3}})$ is

$$S(u) = (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}}) + 1. \quad (2.71)$$

The syndrome table of error generalized polynomials $e_i(x^{\frac{2}{3}})$, where $0 \leq i \leq 8$ is given by:

Syndrome Table 1

$e_i(x^{\frac{2}{3}})$	$e(x^{\frac{2}{3}})$	$S(e)$
$e_0(x^{\frac{2}{3}})$	1	$1 + (x^{\frac{2}{3}})^3$
$e_1(x^{\frac{2}{3}})$	$x^{\frac{2}{3}}$	$(x^{\frac{2}{3}}) + (x^{\frac{2}{3}})^4$
$e_2(x^{\frac{2}{3}})$	$(x^{\frac{2}{3}})^2$	$(x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}})^5$
$e_3(x^{\frac{2}{3}})$	$(x^{\frac{2}{3}})^3$	1
$e_4(x^{\frac{2}{3}})$	$(x^{\frac{2}{3}})^4$	$(x^{\frac{2}{3}})$
$e_5(x^{\frac{2}{3}})$	$(x^{\frac{2}{3}})^5$	$(x^{\frac{2}{3}})^2$
$e_6(x^{\frac{2}{3}})$	$(x^{\frac{2}{3}})^6$	$(x^{\frac{2}{3}})^3$
$e_7(x^{\frac{2}{3}})$	$(x^{\frac{2}{3}})^7$	$(x^{\frac{2}{3}})^4$
$e_8(x^{\frac{2}{3}})$	$(x^{\frac{2}{3}})^8$	$(x^{\frac{2}{3}})^5$

From the Syndrome Table 1 we find that $S(u) = S(e_1) + S(e_3)$. So the generalized error polynomial is $e(x^{\frac{2}{3}}) = (x^{\frac{2}{3}}) + (x^{\frac{2}{3}})^3$ which has error pattern $e = 010100000$, which is a burst of length 3. Therefore,

$$v(x^{\frac{2}{3}}) = u(x^{\frac{2}{3}}) - e(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^6,$$

which is the generator generalized polynomial of the code \mathcal{C}_9 , its vector form is 100100100.

Now, on shifting the generalized received polynomial

$$u(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}}) + (x^{\frac{2}{3}})^6 \text{ to } u(x^{\frac{1}{3}}) = 1 + (x^{\frac{1}{3}})^2 + (x^{\frac{1}{3}})^{12} \in \mathbb{F}_2[x; \frac{1}{3}\mathbb{N}_0]_{18}$$

we get received word

$$u = 101000000000100000 \text{ in } \mathcal{C}_{18}.$$

The syndrome of $u(x^{\frac{1}{3}})$ is

$$S(u) = (x^{\frac{1}{3}})^8 + (x^{\frac{1}{3}})^2 + 1.$$

The syndrome table of generalized error polynomials $e_i(x^{\frac{1}{3}})$, where $0 \leq i \leq 17$ is given by:

Syndrome Table 2

$e_i(x^{\frac{1}{3}})$	$e(x^{\frac{1}{3}})$	$S(e)$
$e_{0,1}(x^{\frac{1}{3}})$	1	$1 + (x^{\frac{1}{3}})^6$
$e_{2,3}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^2$	$(x^{\frac{1}{3}})^2 + (x^{\frac{1}{3}})^8$
$e_{4,5}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^4$	$(x^{\frac{1}{3}})^4 + (x^{\frac{1}{3}})^{10}$
$e_{6,7}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^6$	1
$e_{8,9}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^8$	$(x^{\frac{1}{3}})^2$
$e_{10,11}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^{10}$	$(x^{\frac{1}{3}})^4$
$e_{12,13}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^{12}$	$(x^{\frac{1}{3}})^6$
$e_{14,15}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^{14}$	$(x^{\frac{1}{3}})^8$
$e_{16,17}(x^{\frac{1}{3}})$	$(x^{\frac{1}{3}})^{16}$	$(x^{\frac{1}{3}})^{10}$

From the Syndrome Table 2 we get

$$S(u) = S(e_{2,3}(x^{\frac{1}{3}})) + S(e_{6,7}(x^{\frac{1}{3}})). \quad (2.72)$$

This gives the generalized error polynomial $e(x^{\frac{1}{3}}) = (x^{\frac{1}{3}}) + (x^{\frac{1}{3}})^6$ which has error pattern

$$e = 001000100000000000, \quad (2.73)$$

which is a burst of length 5. Therefore,

$$v(x^{\frac{1}{3}}) = u(x^{\frac{1}{3}}) - e(x^{\frac{1}{3}}) = 1 + (x^{\frac{1}{3}})^6 + (x^{\frac{1}{3}})^{12}, \quad (2.74)$$

the generator generalized polynomial of binary cyclic code \mathcal{C}_{18} , and its vector form is

$$100000100000100000. \quad (2.75)$$

The vector u in \mathcal{C}_9 is formed by interleaving 3 rows $u_1 = 101$, $u_2 = 100$ and $u_3 = 000$ in \mathcal{C}_3 which have respectively the error vectors $e_1 = 010$, $e_2 = 100$ and $e_3 = 000$. On interleaving the

vectors $u_1 = 101$ and $u_2 = 100$ in \mathcal{C}_3 , we get a received vector $u = 110010$ in \mathcal{C}_6 . Its decoding gives the error vector $e = 011000$ which is a burst of length 2.

Hence, the interleaved codes $(18, 6)$, $(9, 3)$ and $(6, 2)$ are capable of correcting single burst of length 6, 3 and 2 or less.

In this study, a new technique of constructing binary cyclic codes is introduced using monoid rings $F_2[x; a\mathbb{N}_0]$, $F_2[x; \frac{a}{b}\mathbb{N}_0]$ and $F_2[x; \frac{1}{b}\mathbb{N}_0]$ instead of polynomial ring $F_2[x]$. So, a scheme is articulated in such a manner that; for an n length binary cyclic code \mathcal{C}_n , an ideal in the factor ring $F_2[x; a\mathbb{N}_0]_n$; there exists binary cyclic codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} of lengths an , bn and abn which are respectively ideals in the factor rings $F_2[x]_{an}$, $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ and $F_2[x; \frac{1}{b}\mathbb{N}_0]_{abn}$.

The pronouncements of this chapter are as follows:

1. The generator and parity check matrix of binary cyclic code \mathcal{C}_{abn} contains blocks of generator and parity check matrices of binary cyclic codes $\mathcal{C}_n, \mathcal{C}_{an}$ and \mathcal{C}_{bn} . Hence, encoding and decoding of all the binary cyclic codes $\mathcal{C}_n, \mathcal{C}_{an}$ and \mathcal{C}_{bn} can be done simultaneously by the encoding and decoding of binary cyclic code \mathcal{C}_{abn} .
2. The constructed binary cyclic codes $\mathcal{C}_{an}, \mathcal{C}_{bn}$ and \mathcal{C}_{abn} are found to be interleaved codes of degree a , b and ab , respectively, where the binary cyclic code \mathcal{C}_n is the base code. Therefore, if the base code \mathcal{C}_n corrects t errors, then the interleaved codes $\mathcal{C}_{an}, \mathcal{C}_{bn}$ and \mathcal{C}_{abn} are capable of correcting t bursts of length a, b and ab or less. If \mathcal{C}_n is capable of correcting all bursts of length l or less, then the interleaved codes $\mathcal{C}_{an}, \mathcal{C}_{bn}$ and \mathcal{C}_{abn} are capable of correcting all bursts of length al, bl and abl or less.

Chapter 3

Construction of non-primitive BCH codes using monoid rings

BCH codes are one of the most important classes of cyclic codes for error correction. In this chapter, we have generalized BCH codes using monoid rings instead of a polynomial ring over the binary field \mathbb{F}_2 . We show the existence of non-primitive binary BCH code \mathcal{C}_{bn} of length bn , corresponding to a given n length binary BCH code \mathcal{C}_n . The value of b is investigated for which the existence of non-primitive BCH code \mathcal{C}_{bn} is assured. It is noticed that the code \mathcal{C}_n is embedded in the code \mathcal{C}_{bn} . Therefore, encoding and decoding of the codes \mathcal{C}_n and \mathcal{C}_{bn} can be done simultaneously. The data transmitted by \mathcal{C}_n can also be transmitted by \mathcal{C}_{bn} . The BCH code \mathcal{C}_{bn} has better error correction capability whereas the BCH code \mathcal{C}_n has better code rate, hence both gains are achieved at the same time.

Through monoid rings, in a sequence of papers [4], [36], [37], [39], [38], [34], [35] several classes of cyclic codes over a finite unitary commutative ring are constructed. The purpose of these constructions is to address the error correction and the code rate trade off in a better way. However, for a particular interest in [40] it is established that, there does not exist a binary BCH code of length $(n+1)n$ in the factor ring $\mathbb{F}_2[x; \frac{1}{2}\mathbb{N}_0]/((x^{\frac{1}{2}})^{(n+1)n} - 1)$ generated by generalized polynomial $g(x^{\frac{1}{2}}) \in \mathbb{F}_2[x; \frac{1}{2}\mathbb{N}_0]$ of degree $2r$ corresponding to the length n binary BCH code in $\mathbb{F}_2[x]/(x^n - 1)$ having generator polynomial $g(x) \in \mathbb{F}_2[x]$ of degree r . But, there do exist a binary cyclic code of length $(n+1)n$ such that the length n binary BCH code is embedded in it. Besides

this, the existence of a binary cyclic $((n+1)^{3^k} - 1, (n+1)^{3^k} - 1 - 3^k r)$ code, where k is a positive integer, corresponding to a binary cyclic $(n, n-r)$ code is established in [38] by the use of monoid ring $\mathbb{F}_2[x; \frac{1}{3^k}\mathbb{N}_0]$. In both papers [40] and [38], the authors cannot show the existence of binary BCH codes corresponding to the length n binary BCH code in $\mathbb{F}_2[x]/(x^n - 1)$. In this study, we address this issue and construct a binary BCH code using monoid ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$, where a, b are integers such that $a, b > 1$. We show the existence of non-primitive binary BCH code C_{bn} of length bn using an irreducible polynomial $p(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ of degree br , corresponding to a given length n binary BCH code C_n generated by r degree primitive polynomial $p(x^a)$ in $\mathbb{F}_2[x; a\mathbb{N}_0]$.

3.1 BCH code C_n as ideal in $\mathbb{F}_2[x; a\mathbb{N}_0]_n$

A polynomial ring $\mathbb{F}[x]$ is initially a monoid ring $\mathbb{F}_2[x; S]$, where S is the additive monoid \mathbb{N}_0 , the non-negative integers. It can be observed that $\mathbb{F}_2[x] \subset \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ only when $a = 1$. This force us to first define cyclic codes using monoid ring $\mathbb{F}_2[x; a\mathbb{N}_0]$ and then define cyclic codes using monoid ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$. As $\mathbb{F}_2[x; a\mathbb{N}_0] \subset \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$, also $\mathbb{F}_2[x; a\mathbb{N}_0] \subset \mathbb{F}_2[x]$ for all $a \geq 1$. Where both the monoids $a\mathbb{N}_0$ and $\frac{a}{b}\mathbb{N}_0$ are totally ordered, so degree and order of elements in $\mathbb{F}_2[x; a\mathbb{N}_0]$ and $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ are defined. The indeterminate of polynomials in monoid rings $\mathbb{F}_2[x; a\mathbb{N}_0]$ and $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ are respectively given by x^a and $x^{\frac{a}{b}}$, and they behave like an indeterminate x in $\mathbb{F}_2[x]$. The arbitrary elements in $\mathbb{F}_2[x; a\mathbb{N}_0]$ and $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ are

$$\begin{aligned} f(x^a) &= 1 + (x^a) + (x^a)^2 + \dots (x^a)^n \text{ and} \\ f(x^{\frac{a}{b}}) &= 1 + (x^{\frac{a}{b}}) + (x^{\frac{a}{b}})^2 + \dots (x^{\frac{a}{b}})^n \end{aligned}$$

and we call them generalized polynomials.

The construction of a BCH code in the factor ring $\mathbb{F}_2[x; a\mathbb{N}_0]_n$ is similar to that of a BCH code in $\mathbb{F}_2[x]_n$, as $\mathbb{F}_2[x; a\mathbb{N}_0] \subset \mathbb{F}_2[x]$. For this, let C_n be a binary BCH code based on the positive integers $c, d, q = 2$ and n such that $2 \leq d \leq n$ with $\gcd(n, 2) = 1$ and $n = 2^s - 1$, where s is the degree of a primitive irreducible polynomial in $\mathbb{F}_2[x; a\mathbb{N}_0]$. Consequently, the n length

binary BCH code \mathcal{C}_n has generator polynomial of degree r given by

$$g(x^a) = \text{lcm}\{m_i(x^a) : i = c, c+1, \dots, c+d-2\}, \quad (3.1)$$

where $m_i(x^a)$ are minimal polynomials of ξ^i for $i = c, c+1, \dots, c+d-2$. Where ξ is the primitive n th root of unity in \mathbb{F}_{2^s} , an s degree Galois field extension of \mathbb{F}_2 . Since $m_i(x^a)$ divides $(x^a)^n - 1$ for each i , it follows that $g(x^a)$ divides $(x^a)^n - 1$. This implies $\mathcal{C}_n = (g(x^a))$ is a principal ideal in the factor ring $\mathbb{F}_2[x; a\mathbb{N}_0]_n$.

In the following example primitive BCH code of length 15 is discussed using monoid ring $\mathbb{F}_2[x; 2\mathbb{N}_0]$.

Example 49 Let $p(x^2) = (x^2)^4 + (x^2) + 1$ be a primitive polynomial in $\mathbb{F}_2[x; 2\mathbb{Z}_0]$, then we have a primitive BCH code of length $n = 2^4 - 1 = 15$. Let ξ be a primitive root in $GF(2^4)$, satisfying the relation $\xi^4 + \xi + 1 = 0$. Using this relation we have $\xi^{15} = 1$, that is ξ is the primitive 15th root of unity. Since

$$g(x^2) = \text{lcm}\{m_i(x^2), i = c, c+1, \dots, c+d-2\}, \quad (3.2)$$

therefore first we calculate $m_i(x^2)$. By [27, Theorem 4.4.2], ξ, ξ^2, ξ^4, ξ^8 have same minimal polynomial $m_1(x^2) = p(x^2)$. Similarly we get

$$\begin{aligned} m_3(x^2) &= (x^2)^4 + (x^2)^3 + (x^2)^2 + (x^2) + 1, \\ m_5(x^2) &= (x^2)^2 + (x^2) + 1 \text{ and} \\ m_7(x^2) &= (x^2)^4 + (x^2)^3 + 1. \end{aligned}$$

The BCH code with designed distance $d = 3$ has generator polynomial

$$g(x^2) = m_1(x^2) = (x^2)^4 + (x^2) + 1.$$

It has minimum distance at least 3 and corrects up to 1 error. Since the generator polynomial is of degree 4, therefore it is a $(15, 11)$ code having code rate $R = 0.733$. BCH codes of length

15 with different design distances are discussed in Table 3:

Table 3: BCH codes of length 15

d	(n, k)	t	R
3	(15, 11)	1	0.733
5	(15, 7)	2	0.466
7	(15, 5)	3	0.333
15	(15, 1)	7	0.066

3.2 BCH codes as ideals in $F_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$

In this section, we investigate the values of b for which there exists a bn length BCH code in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, corresponding to an n length BCH code \mathcal{C}_n in $\mathbb{F}_2[x; a\mathbb{N}_0]_n$. For this, let \mathcal{C}_n be a binary BCH code in $\mathbb{F}_2[x; a\mathbb{N}_0]_n$ constructed in previous section. Now using the following map

$$p(x^a) = p_0 + p_1x^a + \dots + p_s(x^a)^s \mapsto p_0 + p_1(x^{\frac{a}{b}})^b + \dots + p_s(x^{\frac{a}{b}})^{bs} = p(x^{\frac{a}{b}}), \quad (3.3)$$

we convert the s degree primitive polynomial $p(x^a)$ in $\mathbb{F}_2[x; a\mathbb{N}_0]$ to a bs degree polynomial $p(x^{\frac{a}{b}})$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$. This converted polynomial is never primitive; therefore, the corresponding BCH code will also be non-primitive. However, the non-primitive BCH code can be constructed only when $p(x^{\frac{a}{b}})$ is irreducible. Hence, for the construction of a non-primitive BCH code in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, we choose only such a primitive irreducible polynomial $p(x^a)$ in $\mathbb{F}_2[x; a\mathbb{N}_0]$ for which there is an irreducible polynomial $p(x^{\frac{a}{b}})$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$.

Particularly, for $b = 2$ or $2l$ there neither exist a primitive BCH code nor a non-primitive BCH code, since we know that $p(x^2) = (p(x))^2$ in $\mathbb{F}_2[x]$, the same result holds in $\mathbb{F}_2[x; \frac{a}{2}\mathbb{N}_0]$. Similarly, for $s = 5, 7, 11, 13, 17, \dots$ and there multiples we don't find any b for which we have an irreducible polynomial in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$.

For instance see Table 4 for the list of irreducible polynomials of degree bs in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ corresponding to primitive irreducible polynomial of degree s in $\mathbb{F}_2[x; a\mathbb{N}_0]$.

Table 4: Irreducible polynomials $p(x^{\frac{a}{b}})$ against primitive polynomials $p(x^a)$

deg	$p(x^a)$	$p(x^{\frac{a}{b}})$
3	$(x^a)^3 + (x^a) + 1$	$(x^{\frac{a}{7}})^{21} + (x^{\frac{a}{7}})^7 + 1$
4	$(x^a)^4 + (x^a) + 1$	$(x^{\frac{a}{3}})^{12} + (x^{\frac{a}{3}})^3 + 1, (x^{\frac{a}{5}})^{20} + (x^{\frac{a}{5}})^5 + 1$
6	$(x^a)^6 + (x^a) + 1$	$(x^{\frac{a}{3}})^{18} + (x^{\frac{a}{3}})^3 + 1, (x^{\frac{a}{7}})^{42} + (x^{\frac{a}{7}})^7 + 1$
8	$(x^a)^8 + (x^a)^4 + (x^a)^3 + (x^a)^2 + 1$	$(x^{\frac{a}{3}})^{24} + (x^{\frac{a}{3}})^{12} + (x^{\frac{a}{3}})^9 + (x^{\frac{a}{3}})^6 + 1,$
	$(x^a)^8 + (x^a)^4 + (x^a)^3 + (x^a)^2 + 1$	$(x^{\frac{a}{5}})^{40} + (x^{\frac{a}{5}})^{20} + (x^{\frac{a}{5}})^{15} + (x^{\frac{a}{5}})^{10} + 1$
9	$(x^a)^9 + (x^a)^4 + 1$	$(x^{\frac{a}{7}})^{63} + (x^{\frac{a}{7}})^{28} + 1$
10	$(x^a)^{10} + (x^a)^3 + 1$	$(x^{\frac{a}{3}})^{30} + (x^{\frac{a}{3}})^9 + 1$
\vdots	\vdots	\vdots

Table 4 explains that for $s = 2$ and 3 we have $b = 3$ and 7 and for $s = 4$ and 6 we have $b = (3, 5)$ and $(3, 7)$ respectively and similarly we have for their multiples. From this we have the list of BCH codes of length n and bn , where bn divides $2^{bs} - 1$, mentioned in Table 5.

Table 5: BCH codes of length n and bn

s	n	bn
2	3	9
3	7	49
4	15	45, 75
6	63	189, 441
8	255	765, 1275
9	511	3577
10	1023	1023
\vdots	\vdots	\vdots

The above discussion can be sum up with the following result.

Proposition 50 *Let $p(x^a) \in \mathbb{F}_2[x; a\mathbb{N}_0]$ be a primitive irreducible polynomial of degree $s \in$*

$\{2l, 3l, 4l, 6l\}$, where $l \in \mathbb{Z}^+$. Then the corresponding bs degree generalized polynomial $p(x^{\frac{a}{b}})$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ is non-primitive irreducible polynomial for $b \in \{3, 7, \{3, 5\}, \{3, 7\}\}$ respectively.

Proof. Let $p(x^a) = 1 + x^a + \dots + (x^a)^s$ be a primitive irreducible polynomial in $\mathbb{F}_2[x; a\mathbb{N}_0]$, where $s \in \{2l, 3l, 4l, 6l\}$, where $l \in \mathbb{Z}^+$ such that α is its root and $\alpha^{2^s-1} = 1$. Then the corresponding generalized polynomial $p(x^{\frac{a}{b}}) = 1 + (x^{\frac{a}{b}})^b + \dots + (x^{\frac{a}{b}})^{bs}$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ has root $\beta = (p_{si})^M \in \mathbb{F}_2^{bs}$, where p_{si} is a primitive element in \mathbb{F}_2^{bs} and M is a positive integer such that

$$M(b(2^s - 1)) = 2^{bs} - 1.$$

This implies $\beta^{b(2^s-1)} = 1$. Hence $p(x^{\frac{a}{b}})$ is not primitive. But $p(x^{\frac{a}{b}})$ is irreducible over \mathbb{F}_2 for $b \in \{3, 7, \{3, 5\}, \{3, 7\}\}$ respectively by [14, Theorem 5.1 and Example 5.4], where the indeterminate $x^{\frac{a}{b}}$ behaves as indeterminate x . ■

Definition 51 A code \mathcal{C} generated by a non-primitive element β of a Galois field $GF(q^m)$, such that the length of the code is the order of β , is called a **non-primitive BCH code**.

Theorem 52 Let $n = 2^s - 1$ be the length of primitive BCH code \mathcal{C}_n , where $p(x^a) \in \mathbb{F}_2[x; a\mathbb{N}_0]$ is a primitive irreducible polynomial of degree s such that $p(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ is irreducible polynomial of degree bs .

1) Then for positive integers c_1, d_1, bn such that $2 \leq d_1 \leq bn$ and bn is relatively prime to 2, there exists a non-primitive binary BCH code \mathcal{C}_{bn} of length bn , where bn is order of an element $\alpha \in \mathbb{F}_{2^{bs}}$.

2) The non-primitive BCH code \mathcal{C}_{bn} of length bn is defined as

$$\mathcal{C}_{bn} = \{v(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn} : v(\alpha^i) = 0 \text{ for all } i = c_1, c_1 + 1, \dots, c_1 + d_1 - 2.\}$$

Equivalently, \mathcal{C}_{bn} is the null space of the matrix

$$H = \begin{bmatrix} 1 & \alpha^{c_1} & \alpha^{2c_1} & \dots & \alpha^{(bn-1)c_1} \\ 1 & \alpha^{c_1+1} & \alpha^{2(c_1+1)} & \dots & \alpha^{(bn-1)(c_1+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{c_1+d_1-2} & \alpha^{2(c_1+d_1-2)} & \dots & \alpha^{(bn-1)(c_1+d_1-2)} \end{bmatrix}. \quad (3.4)$$

Proof. 1) Since it is given that the bs degree polynomial $p(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$ is not primitive, so the BCH code constructed through it is also not primitive. Hence the length of the code $n \neq 2^{bs} - 1$. However, there is an element $\alpha \in \mathbb{F}_{2^{bs}}$ of order bn vanishing $p(x^{\frac{a}{b}})$. Let $m_i(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{Z}_0]$ denotes the minimal polynomial of α^i and $g(x^{\frac{a}{b}})$ be the lcm of distinct polynomials among $m_i(x^{\frac{a}{b}})$, $i = c_1, c_1 + 1, \dots, c_1 + d_1 - 2$; that is,

$$g(x^{\frac{a}{b}}) = \text{lcm}\{m_i(x^{\frac{a}{b}}) : i = c_1, c_1 + 1, \dots, c_1 + d_1 - 2\}.$$

As $m_i(x^{\frac{a}{b}})$ divides $(x^{\frac{a}{b}})^{bn} - 1$ for each i , therefore $g(x^{\frac{a}{b}})$ also divides $(x^{\frac{a}{b}})^{bn} - 1$. This implies that \mathcal{C}_{bn} is a principal ideal generated by $g(x^{\frac{a}{b}})$ in the factor ring $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. Hence \mathcal{C}_{bn} is a non-primitive BCH code of length bn over \mathbb{F}_2 with designed distance d_1 .

2) Let $v(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$, then

$$v(x^{\frac{a}{b}}) = g(x^{\frac{a}{b}})q(x^{\frac{a}{b}})$$

for some $q(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$, where $g(x^{\frac{a}{b}})$ is the generator polynomial of \mathcal{C}_{bn} . Hence $v(\alpha^i) = 0$ for all $i = c_1, c_1 + 1, \dots, c_1 + d_1 - 2$. Conversely, let $v(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ such that $v(\alpha^i) = 0$ for all $i = c_1, c_1 + 1, \dots, c_1 + d_1 - 2$. Then $m_i(x^{\frac{a}{b}})$ divides $v(x^{\frac{a}{b}})$ for all $i = c_1, c_1 + 1, \dots, c_1 + d_1 - 2$. Hence $g(x^{\frac{a}{b}})$ divides $v(x^{\frac{a}{b}})$, so $v(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$.

For second part, let

$$v(x^{\frac{a}{b}}) = v_0 + v_1(x^{\frac{a}{b}}) + \dots + v_{bn-1}(x^{\frac{a}{b}})^{bn-1} \in \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]_{bn}.$$

Then $v(\alpha^i) = 0$ for all $i = c_1, c_1 + 1, \dots, c_1 + d_1 - 2$ if and only if $Hv^T = 0$, where $v = (v_0, v_1, \dots, v_{bn-1}) \in \mathbb{F}_2^{bn}$. This proves that \mathcal{C}_{bn} is the null space of H . ■

Remark 53 Corresponding to the (n, k) BCH code \mathcal{C}_n with generator polynomial $g(x^a) = p(x^a)$ in $\mathbb{F}_2[x; a\mathbb{N}_0]$, we have a (bn, bk) BCH code \mathcal{C}_{bn} with generating polynomial $g(x^{\frac{a}{b}}) = p(x^{\frac{a}{b}})$ in $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$. This (bn, bk) BCH code \mathcal{C}_{bn} is an interleaved code of degree b , capable of correcting a single error burst of length b or less (see [29, Theorem 11.1]).

The following example illustrates the construction of a non-primitive BCH code of length bn through $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$.

Example 54 For a primitive polynomial $p(x^2) = 1 + (x^2) + (x^2)^4$ in $\mathbb{F}_2[x; 2\mathbb{N}_0]$, there is a non-primitive irreducible polynomial $p(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^{12}$ in $\mathbb{F}_2[x; \frac{2}{3}\mathbb{N}_0]$. Let $\alpha \in \mathbb{F}_{2^{12}}$, satisfies the relation $\alpha^{12} + \alpha^3 + 1 = 0$. Using this relation we can compute all the distinct powers of α in $GF(2^{12})$, see Table 6 (it is clear that α has order 45).

Table 6: Distinct powers of α in $GF(2^{12})$

$\alpha^{12} = \alpha^3 + 1$	$\alpha^{21} = 1 + \alpha^3 + \alpha^9$	$\alpha^{30} = 1 + \alpha^3 + \alpha^6$	$\alpha^{39} = 1 + \alpha^6 + \alpha^9$
$\alpha^{13} = \alpha + \alpha^4$	$\alpha^{22} = \alpha + \alpha^4 + \alpha^{10}$	$\alpha^{31} = \alpha + \alpha^4 + \alpha^7$	$\alpha^{40} = \alpha + \alpha^7 + \alpha^{10}$
$\alpha^{14} = \alpha^2 + \alpha^5$	$\alpha^{23} = \alpha^2 + \alpha^5 + \alpha^{11}$	$\alpha^{32} = \alpha^2 + \alpha^5 + \alpha^8$	$\alpha^{41} = \alpha^2 + \alpha^8 + \alpha^{11}$
$\alpha^{15} = \alpha^3 + \alpha^6$	$\alpha^{24} = 1 + \alpha^6$	$\alpha^{33} = \alpha^3 + \alpha^6 + \alpha^9$	$\alpha^{42} = 1 + \alpha^9$
$\alpha^{16} = \alpha^4 + \alpha^7$	$\alpha^{25} = \alpha + \alpha^7$	$\alpha^{34} = \alpha^4 + \alpha^7 + \alpha^{10}$	$\alpha^{43} = \alpha + \alpha^{10}$
$\alpha^{17} = \alpha^5 + \alpha^8$	$\alpha^{26} = \alpha^8 + \alpha^2$	$\alpha^{35} = \alpha^5 + \alpha^8 + \alpha^{11}$	$\alpha^{44} = \alpha^2 + \alpha^{11}$
$\alpha^{18} = \alpha^6 + \alpha^9$	$\alpha^{27} = \alpha^3 + \alpha^9$	$\alpha^{36} = 1 + \alpha^3 + \alpha^6 + \alpha^9$	$\alpha^{45} = 1$
$\alpha^{19} = \alpha^7 + \alpha^{10}$	$\alpha^{28} = \alpha^4 + \alpha^{10}$	$\alpha^{37} = \alpha + \alpha^4 + \alpha^7 + \alpha^{10}$	
$\alpha^{20} = \alpha^8 + \alpha^{11}$	$\alpha^{29} = \alpha^5 + \alpha^{11}$	$\alpha^{38} = \alpha^2 + \alpha^5 + \alpha^8 + \alpha^{11}$	

Here we have $bn = n' = 3 \times 15 = 45$. To calculate the generating polynomial $g(x^{\frac{2}{3}})$ we first calculate the minimal polynomials which are :

$$\begin{aligned}
m'_1(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^3 + 1, \\
m'_3(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}}) + 1, \\
m'_5(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^3 + 1, \\
m'_7(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^9 + 1, \\
m'_9(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + 1, \\
m'_{15}(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + 1, \\
m'_{21}(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}})^3 + 1.
\end{aligned}$$

Which gives the following generating polynomials of BCH codes of length 45 with design distance

$d_1 = 3, 5, 7, 9, 15, 21$ and 45.

$$\begin{aligned}
g(x^{\frac{2}{3}}) &= 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^{12}, \quad g(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}}) + 1 \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{22} + (x^{\frac{2}{3}})^{18} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^9 + (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}}) + 1 \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{34} + (x^{\frac{2}{3}})^{31} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{19} + (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}}) + 1 \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{37} + (x^{\frac{2}{3}})^{36} + (x^{\frac{2}{3}})^{34} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{22} + (x^{\frac{2}{3}})^{21} \\
&\quad + (x^{\frac{2}{3}})^{19} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^4 + 1 \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{40} + (x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{35} + (x^{\frac{2}{3}})^{34} + (x^{\frac{2}{3}})^{32} + (x^{\frac{2}{3}})^{31} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{25} \\
&\quad + (x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{20} + (x^{\frac{2}{3}})^{19} + (x^{\frac{2}{3}})^{17} + (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^8 \\
&\quad + (x^{\frac{2}{3}})^5 + (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + 1 \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{44} + (x^{\frac{2}{3}})^{43} + (x^{\frac{2}{3}})^{42} + \dots + (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + 1
\end{aligned}$$

Which generates $(45, 33), (45, 29), (45, 23), (45, 11), (45, 7), (45, 5)$ and $(45, 1)$ codes and corrects up to 1, 2, 3, 4, 7, 10 and 22 errors having code rate 0.733, 0.644, 0.511, 0.244, 0.155, 0.11, 0.022 respectively. Where the code $(45, 33)$ is also capable of correcting any single error burst of length 3 or less by Remark 53.

Table 7, gives a comparison between minimum distance, code rate and error correction capability of codes $\mathcal{C}_{15}, \mathcal{C}_{45}$ in $\mathbb{F}_2[x; 2\mathbb{N}_0], \mathbb{F}_2[x; \frac{2}{3}\mathbb{N}_0]$ respectively.

Table 7: Comparison between \mathcal{C}_{15} and \mathcal{C}_{45}

(n, k)	d	t	R	(n, k)	d'	t_1	R_1
(15, 11)	3	1	0.733	(45, 33)	3	1	0.733
(15, 7)	5	2	0.466	(45, 29)	5	2	0.644
(15, 5)	7	3	0.333	(45, 23)	7	3	0.511
(15, 1)	15	7	0.066	(45, 11)	9	4	0.244
				(45, 7)	15	7	0.155
				(45, 5)	21	10	0.11
				(45, 1)	45	22	0.022

As bn divides $n' = 2^{bs} - 1$, so $(x^{\frac{a}{b}})^{bn} - 1$ divides $(x^{\frac{a}{b}})^{n'} - 1$ in $F_2[x; \frac{a}{b}\mathbb{N}_0]$. It follows that $((x^{\frac{a}{b}})^{n'} - 1) \subset ((x^{\frac{a}{b}})^{bn} - 1)$. Consequently, third isomorphism theorem for rings gives

$$\frac{F_2[x; \frac{a}{b}\mathbb{N}_0]/((x^{\frac{a}{b}})^{n'} - 1)}{((x^{\frac{a}{b}})^{bn} - 1)/((x^{\frac{a}{b}})^{n'} - 1)} \simeq \frac{F_2[x; \frac{a}{b}\mathbb{N}_0]}{((x^{\frac{a}{b}})^{bn} - 1)} \simeq \frac{F_2[x; a\mathbb{N}_0]}{((x^a)^n - 1)}.$$

Thus, there is embedding $C_n \hookrightarrow C_{bn} \hookrightarrow C_{n'}$ of codes, whereas C_n, C_{bn} and $C_{n'}$ are respectively primitive BCH, non-primitive BCH and primitive BCH codes. Whereas the embedding $C_n \hookrightarrow C_{bn}$ is defined as:

$$a(x^a) = a_0 + a_1(x^a) + \dots + a_{n-1}(x^a)^{n-1} \mapsto a_0 + a_1(x^{\frac{a}{b}})^b + \dots + a_{n-1}(x^{\frac{a}{b}})^{b(n-1)} = a(x^{\frac{a}{b}}).$$

Where $a(x^a) \in C_n$ and $a(x^{\frac{a}{b}}) \in C_{bn}$.

The above discussion shapes the following.

Theorem 55 *Let C_n be a primitive binary BCH code of length $n = 2^s - 1$ generated by r degree polynomial $g(x^a)$ in $F_2[x; a\mathbb{N}_0]$, then:*

- 1) *There exists a bn length binary non-primitive BCH code C_{bn} generated by br degree polynomial $g(x^{\frac{a}{b}})$ in $F_2[x; \frac{a}{b}\mathbb{N}_0]$; and*
- 2) *The binary primitive BCH code C_n is embedded in the binary non-primitive BCH code C_{bn} .*

Also we can deduce $g(x^a)$ from $g(x^{\frac{a}{b}})$ by substituting x^a for y^b .

Example 56 *Following Examples 49 and 54:*

The BCH codes with designed distance $d = 3$ have generator polynomials $g(x^2) = m_1(x^2) = 1 + (x^2) + (x^2)^4$ and $g(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^{12}$ with same error correction capability and code rate. The only difference is; the degree, data bits, code length and check sum of the code C_{45} is three times that of code C_{15} .

Whereas, on letting $(x^{\frac{2}{3}}) = y$ in the generating polynomial of $(45, 29)$ code, that is $x^2 = y^3$,

we get

$$\begin{aligned}
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}}) + 1 \\
g(y) &= (y)^{16} + (y)^{13} + (y)^{12} + (y)^7 + (y)^3 + (y) + 1 \\
g(y^3) &= (y^3)^{16} + (y^3)^{13} + (y^3)^{12} + (y^3)^7 + (y^3)^3 + (y^3) + 1 \\
g(x^2) &= (x^2)^{16} + (x^2)^{13} + (x^2)^{12} + (x^2)^7 + (x^2)^3 + (x^2) + 1 \\
&= (x^2)^{13} + (x^2)^{12} + (x^2)^7 + (x^2)^3 + 1 \in F_2[x; 2\mathbb{N}_0]_{15}.
\end{aligned}$$

Where the generating polynomial $(x^2)^4 + (x^2) + 1$ divides $(x^2)^{13} + (x^2)^{12} + (x^2)^7 + (x^2)^3 + 1$. Hence the corresponding vector is in $(15, 11)$. So $(15, 11)$ code is embedded in $(45, 29)$ code.

Similarly, in Table 6, we have shown that which code in $F_2[x; 2\mathbb{N}_0]_{15}$ with designed distance d is embedded in a code in $F_2[x; \frac{2}{3}\mathbb{N}_0]_{45}$ with designed distance d' .

The corresponding code vectors of the generating polynomials

$$\begin{aligned}
g(x^2) &= (x^2)^8 + (x^2)^7 + (x^2)^6 + (x^2)^4 + 1 \text{ and} \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{37} + (x^{\frac{2}{3}})^{36} + (x^{\frac{2}{3}})^{34} + (x^{\frac{2}{3}})^{30} \\
&\quad + (x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{22} + (x^{\frac{2}{3}})^{21} + (x^{\frac{2}{3}})^{19} + (x^{\frac{2}{3}})^{15} \\
&\quad + (x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^4 + 1 \text{ are}
\end{aligned}$$

$$\begin{aligned}
v &= (1000101111000000) \\
v' &= (1000101111000000 \\
&\quad 1000101111000000 \\
&\quad 1000101111000000).
\end{aligned}$$

Clearly v is properly contain in v' , in fact it is repeating three time after a particular pattern. Hence the generating matrix G' of $g(x^{\frac{2}{3}})$ will contain the generating matrix G of $g(x^2)$ such that $G' = \oplus_1^3 G$.

The next chapter is the generalization of this chapter. Therefore a generalized decoding

procedure is given there and hence omitted in this chapter.

Chapter 4

Family of non-primitive BCH codes

In this chapter, we have developed a relation between a primitive and a family of non-primitive BCH codes. We show the existence of a family of non-primitive binary BCH codes $\{\mathcal{C}_{b^j n}^j\}_{1 \leq j \leq m}$, where $b^j n$ is the length of the code $\mathcal{C}_{b^j n}^j$, using an n length binary primitive BCH code \mathcal{C}_n . Furthermore a decoding procedure is introduced, such that a codeword in the binary BCH code \mathcal{C}_n can be transmitted with high code rate and decoded through codeword of any of the binary BCH code of the family $\{\mathcal{C}_{b^j n}^j\}_{1 \leq j \leq m}$. Moreover it is observed that, for each $1 \leq j \leq m$, the binary BCH code $\mathcal{C}_{b^{j-1} n}^{j-1}$ is embedded in the binary BCH $\mathcal{C}_{b^j n}^j$.

Encoding and decoding algorithms are also introduced for a binary non-primitive BCH code of length $b^j n$ against an n length binary primitive BCH code. The algorithms have been simulated in Matlab. Matlab provides a built in routines for primitive BCH code, but impose several constraints, like degree of primitive polynomial that is s should be lesser than 16. This work focuses on non-primitive polynomials, where s changes to bs and go far more than 16. In order to lever these conditions we have developed generic algorithm.

4.1 BCH codes as ideal in $F_2[x; \frac{a}{b^j} \mathbb{N}_0]_{b^j n(1 \leq j \leq m)}$

In the previous chapter we have shown the construction of binary BCH code of length bn in the monoid ring $\mathbb{F}_2[x; \frac{a}{b} \mathbb{N}_0]_{bn}$, in this section we will show the existence of family of BCH codes $\{\mathcal{C}_{b^j n}^j\}_{1 \leq j \leq m}$ in the monoid ring $\mathbb{F}_2[x; \frac{a}{b^j} \mathbb{N}_0]_{b^j n(1 \leq j \leq m)}$. For this we use the same technique discussed in the last chapter. Hence we first investigate the values of b for which there exists a

$b^j n$ length BCH code in $\mathbb{F}_2[x; \frac{a}{b^j} \mathbb{N}_0]_{b^j n}$, corresponding to an n length BCH code \mathcal{C}_n in $\mathbb{F}_2[x; a \mathbb{N}_0]_n$.

Using the following map

$$p_0 + p_1 x^a + \dots + p_s (x^a)^s \mapsto p_0 + p_1 (x^{\frac{a}{b^j}})^{b^j} + \dots + p_{s-1} (x^{\frac{a}{b^j}})^{b^j s},$$

we convert the s degree primitive polynomial $p(x^a)$ in $\mathbb{F}_2[x; a \mathbb{N}_0]$ to a $b^j s$ degree polynomial $p(x^{\frac{a}{b^j}})$ in $\mathbb{F}_2[x; \frac{a}{b^j} \mathbb{N}_0]$. We will consider only such a primitive irreducible polynomial $p(x^a)$ in $\mathbb{F}_2[x; a \mathbb{N}_0]$ for which there is an irreducible polynomial $p(x^{\frac{a}{b^j}})$ in $\mathbb{F}_2[x; \frac{a}{b^j} \mathbb{N}_0]$. Following table gives a list of few irreducible polynomials of degree $b^j s$ in $\mathbb{F}_2[x; \frac{a}{b^j} \mathbb{N}_0]$ corresponding to primitive polynomial of degree s in $\mathbb{F}_2[x; a \mathbb{N}_0]$. For $p(x^a) \in \mathbb{F}_2[x; a \mathbb{N}_0]$, $p(x^{\frac{a}{b}}) \in \mathbb{F}_2[x; \frac{a}{b} \mathbb{N}_0]$, $p(x^{\frac{a}{b^2}}) \in \mathbb{F}_2[x; \frac{a}{b^2} \mathbb{N}_0]$, replace x^a , $x^{\frac{a}{b}}$, $x^{\frac{a}{b^2}}$ by x , y , z respectively.

Table 8 : Irreducible polynomials corresponding to primitive polynomials

deg	$p(x)$	$p(y)$	$p(z)...$
3	$1 + x + x^3$	$1 + y^7 + y^{21}$	$1 + z^{49} + z^{147}$
4	$1 + x + x^4$	$1 + y^3 + y^{12},$ $1 + y^5 + y^{20}$	$1 + z^9 + z^{36},$ $1 + z^{25} + z^{100}$
6	$1 + x + x^6$	$1 + y^3 + y^{18}$ $1 + y^7 + y^{42}$	$1 + z^9 + z^{54}$ $1 + z^{49} + z^{294}$
8	$1 + x + x^3$ $+ x^5 + x^8$	$1 + y^3 + y^9$ $+ y^{15} + y^{24},$ $1 + y^5 + y^{15}$ $+ y^{25} + y^{40}$	$1 + z^9 + z^{27}$ $+ z^{45} + z^{72},$ $1 + z^{25} + z^{75}$ $+ z^{125} + z^{200}$
9	$1 + x^4 + x^9$	$1 + y^{28} + y^{63}$	$1 + z^{196} + z^{441}$
10	$1 + x^3 + x^{10}$	$1 + y^9 + y^{30}$	
\vdots	\vdots	\vdots	\vdots

By Table 9 we deduce the following Theorem.

Theorem 57 *Let $p(x^a) \in \mathbb{F}_2[x; a \mathbb{N}_0]$ be a primitive irreducible polynomial of degree $s \in \{2l, 3l, 4l, 6l\}$, where $l \in \mathbb{Z}^+$. Then the corresponding $b^j s$ degree generalized polynomial $p(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j} \mathbb{N}_0]$ is*

non-primitive irreducible for $b \in \{3, 7, \{3, 5\}, \{3, 7\}\}$ respectively.

Proof is same as that of Theorem 50.

The existence of non-primitive BCH code of length $b^j n$ is shown in the following Theorem.

Theorem 58 *Let $n = 2^s - 1$ be the length of primitive BCH code \mathcal{C}_n , where $p(x^a) \in \mathbb{F}_2[x; a\mathbb{N}_0]$ is a primitive irreducible polynomial of degree s such that $p(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ is non-primitive irreducible polynomial of degree $b^j s$.*

1) *Then for positive integers $c_j, d_j, b^j n$ such that $2 \leq d_j \leq b^j n$ and $b^j n$ is relatively prime to 2, there exist a non-primitive binary BCH code $\mathcal{C}_{b^j n}$ of length $b^j n$, where $b^j n$ is order of an element $\alpha \in \mathbb{F}_{2^{b^j s}}$.*

2) *The non-primitive BCH code $\mathcal{C}_{b^j n}$ of length $b^j n$ is defined as*

$$\mathcal{C}_{b^j n} = \{v(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]_{b^j n} : v(\alpha^i) = 0 \text{ for all } i = c_j, c_j + 1, \dots, c_j + d_j - 2\}$$

Equivalently, $\mathcal{C}_{b^j n}$ is the null space of the matrix

$$H = \begin{bmatrix} 1 & \alpha^{c_j} & \alpha^{2c_j} & \dots & \alpha^{(b^j n - 1)c_j} \\ 1 & \alpha^{c_j + 1} & \alpha^{2(c_j + 1)} & \dots & \alpha^{(b^j n - 1)(c_j + 1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{c_j + d_j - 2} & \alpha^{2(c_j + d_j - 2)} & \dots & \alpha^{(b^j n - 1)(c_j + d_j - 2)} \end{bmatrix}$$

Proof. 1) Since $b^j s$ degree polynomial $p(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ is irreducible but not primitive, so there does not exist $n_j = 2^{b^j s} - 1$ length primitive BCH code. However, there is an element α of order $b^j n$ vanishes $p(x^{\frac{a}{b^j}})$. Now, since $2^s - 1 \mid 2^{b^j s} - 1$ then $b^j(2^s - 1)$ also divide $2^{b^j s} - 1$. Hence $\alpha^{b^j n} = \alpha^{n_j} = 1$, implies $\alpha \in \mathbb{F}_{2^{b^j s}}$. Let $m'_i(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{Z}_0]$ denotes the minimal polynomial of α^i and $g(x^{\frac{a}{b^j}})$ be the lcm of distinct polynomials among $m'_i(x^{\frac{a}{b^j}})$, $i = c_j, c_j + 1, \dots, c_j + d_j - 2$; that is,

$$g(x^{\frac{a}{b^j}}) = \text{lcm}\{m'_i(x^{\frac{a}{b^j}}) : i = c_j, c_j + 1, \dots, c_j + d_j - 2\}$$

As $m'_i(x^{\frac{a}{b^j}})$ divides $(x^{\frac{a}{b^j}})^{b^j n} - 1$ for each i , therefore $g(x^{\frac{a}{b^j}})$ also divides $(x^{\frac{a}{b^j}})^{b^j n} - 1$. This implies that $\mathcal{C}_{b^j n}$ is a principal ideal generated by $g(x^{\frac{a}{b^j}})$ in the factor ring $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]_{b^j n}$. Hence $\mathcal{C}_{b^j n}$ is a non-primitive BCH code of length $b^j n$ over \mathbb{F}_2 with designed distance d_j .

2) Let $v(x^{\frac{a}{b^j}}) \in \mathcal{C}_{b^j n}$, then

$$v(x^{\frac{a}{b^j}}) = g(x^{\frac{a}{b^j}})q(x^{\frac{a}{b^j}})$$

for some $q(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$, where $g(x^{\frac{a}{b^j}})$ is the generator polynomial of $\mathcal{C}_{b^j n}$. Hence $v(\alpha^i) = 0$ for all $i = c_j, c_j + 1, \dots, c_j + d_j - 2$. Conversely, let $v(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]_{b^j n}$ such that $v(\alpha^i) = 0$ for all $i = c_j, c_j + 1, \dots, c_j + d_j - 2$. Then $m_i(x^{\frac{a}{b^j}})$ divides $v(x^{\frac{a}{b^j}})$ for all $i = c_j, c_j + 1, \dots, c_j + d_j - 2$. Hence $g(x^{\frac{a}{b^j}})$ divides $v(x^{\frac{a}{b^j}})$, so $v(x^{\frac{a}{b^j}}) \in \mathcal{C}_{b^j n}$.

For second part, let

$$v(x^{\frac{a}{b^j}}) = v_0 + v_1(x^{\frac{a}{b^j}}) + \dots + v_{b^j n - 1}(x^{\frac{a}{b^j}})^{b^j n - 1} \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]_{b^j n}.$$

Then $v(\alpha^i) = 0$ for all $i = c_j, c_j + 1, \dots, c_j + d_j - 2$ if and only if $Hv^T = 0$, where

$$v = (v_0, v_1, \dots, v_{b^j n - 1}) \in \mathbb{F}_2^{b^j n}.$$

This proves that $\mathcal{C}_{b^j n}$ is the null space of H . ■

Following are the examples of the construction of family of non-primitive BCH code.

Example 59 For a primitive polynomial $p(x^2) = 1 + (x^2) + (x^2)^3$ in $\mathbb{F}_2[x; 2\mathbb{N}_0]$ there is a non-primitive irreducible polynomial $p(x^{\frac{2}{7}}) = (x^{\frac{2}{7}})^{21} + (x^{\frac{2}{7}})^7 + 1$ in $\mathbb{F}_2[x; \frac{2}{7}\mathbb{N}_0]$ by Table 8. Let

$\alpha \in \mathbb{F}_{2^{21}}$, satisfies the relation $\alpha^{21} + \alpha^7 + 1 = 0$. Using this relation we have the following table.

Table 9: Distinct powers of α in $GF(2^{21})$

$\alpha^{21} = \alpha^7 + 1$	$\alpha^{31} = \alpha^{17} + \alpha^{10}$	$\alpha^{41} = \alpha^{20} + \alpha^{13} + \alpha^6$
$\alpha^{22} = \alpha^8 + \alpha$	$\alpha^{32} = \alpha^{18} + \alpha^{11}$	$\alpha^{42} = \alpha^{14} + 1$
$\alpha^{23} = \alpha^9 + \alpha^2$	$\alpha^{33} = \alpha^{19} + \alpha^{12}$	$\alpha^{43} = \alpha^{15} + \alpha$
$\alpha^{24} = \alpha^{10} + \alpha^3$	$\alpha^{34} = \alpha^{20} + \alpha^{13}$	$\alpha^{44} = \alpha^{16} + \alpha^2$
$\alpha^{25} = \alpha^{11} + \alpha^4$	$\alpha^{35} = \alpha^{14} + \alpha^7 + 1$	$\alpha^{45} = \alpha^{17} + \alpha^3$
$\alpha^{26} = \alpha^{12} + \alpha^5$	$\alpha^{36} = \alpha^{15} + \alpha^8 + \alpha$	$\alpha^{46} = \alpha^{18} + \alpha^4$
$\alpha^{27} = \alpha^{13} + \alpha^6$	$\alpha^{37} = \alpha^{16} + \alpha^9 + \alpha^2$	$\alpha^{47} = \alpha^{19} + \alpha^5$
$\alpha^{28} = \alpha^{14} + \alpha^7$	$\alpha^{38} = \alpha^{17} + \alpha^{10} + \alpha^3$	$\alpha^{48} = \alpha^{20} + \alpha^6$
$\alpha^{29} = \alpha^{15} + \alpha^8$	$\alpha^{39} = \alpha^{18} + \alpha^{11} + \alpha^4$	$\alpha^{49} = 1$
$\alpha^{30} = \alpha^{16} + \alpha^9$	$\alpha^{40} = \alpha^{19} + \alpha^{12} + \alpha^5$	

Hence length of the code is $bn = n' = 7 \times 7 = 49$. Now, to calculate generating polynomial $g(x^{\frac{2}{7}})$ we first calculate the minimal polynomials. By [27, Theorem 4.4.2], $\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}, \alpha^{15}, \alpha^{30}, \alpha^{11}, \alpha^{22}, \alpha^{44}, \alpha^{39}, \alpha^{29}, \alpha^9, \alpha^{18}, \alpha^{36}, \alpha^{23}, \alpha^{46}, \alpha^{43}, \alpha^{37}, \alpha^{25}$ all have same minimal polynomial $m'_1(x^{\frac{2}{7}}) = p(x^{\frac{2}{7}})$. The set of powers of these α collectively form a set which is called a set of cyclotomic cosets. Let $m'_3(x^{\frac{2}{7}})$ be the minimal polynomial for α^3 , then $\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}, \alpha^{47}, \alpha^{45}, \alpha^{41}, \alpha^{33}, \alpha^{17}, \alpha^{34}, \alpha^{19}, \alpha^{38}, \alpha^{27}, \alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{31}, \alpha^{13}, \alpha^{26}$ all are roots for $m'_3(x^{\frac{2}{7}})$. Therefore by using Table 9 we have $m'_3(x^{\frac{2}{7}}) = (x^{\frac{2}{7}})^{21} + (x^{\frac{2}{7}})^{14} + 1$. Similarly we get,

$$m'_7(x^{\frac{2}{7}}) = (x^{\frac{2}{7}})^3 + (x^{\frac{2}{7}}) + 1 \text{ and } m'_{21}(x^{\frac{2}{7}}) = (x^{\frac{2}{7}})^3 + (x^{\frac{2}{7}})^2 + 1.$$

Which gives the following generating polynomials of BCH code with design distance $d' = 3, 7, 21$

and 49.

$$\begin{aligned}
g(x^{\frac{2}{7}}) &= (x^{\frac{2}{7}})^{21} + (x^{\frac{2}{7}})^7 + 1 \\
g(x^{\frac{2}{7}}) &= (x^{\frac{2}{7}})^{42} + (x^{\frac{2}{7}})^{35} + (x^{\frac{2}{7}})^{28} + (x^{\frac{2}{7}})^{21} + (x^{\frac{2}{7}})^{14} + (x^{\frac{2}{7}})^7 + 1 \\
g(x^{\frac{2}{7}}) &= (x^{\frac{2}{7}})^{45} + (x^{\frac{2}{7}})^{43} + (x^{\frac{2}{7}})^{42} + (x^{\frac{2}{7}})^{38} + (x^{\frac{2}{7}})^{36} + (x^{\frac{2}{7}})^{35} + \\
&\quad (x^{\frac{2}{7}})^{31} + (x^{\frac{2}{7}})^{29} + (x^{\frac{2}{7}})^{28} + (x^{\frac{2}{7}})^{24} + (x^{\frac{2}{7}})^{22} + (x^{\frac{2}{7}})^{21} + \\
&\quad (x^{\frac{2}{7}})^{17} + (x^{\frac{2}{7}})^{15} + (x^{\frac{2}{7}})^{14} + (x^{\frac{2}{7}})^{10} + (x^{\frac{2}{7}})^8 + (x^{\frac{2}{7}})^7 + \\
&\quad (x^{\frac{2}{7}})^3 + (x^{\frac{2}{7}}) + 1 \\
g(x^{\frac{2}{7}}) &= (x^{\frac{2}{7}})^{48} + (x^{\frac{2}{7}})^{47} + (x^{\frac{2}{7}})^{46} + \dots + (x^{\frac{2}{7}})^2 + (x^{\frac{2}{7}}) + 1
\end{aligned}$$

Which generates $(49, 28)$, $(49, 7)$, $(49, 4)$ and $(49, 1)$ codes which corrects up to 1, 3, 10 and 24 errors having code rate 0.571, 0.143, 0.081 and 0.020 respectively. Following tables give comparison between minimum distances, code rate and error correction capability of codes constructed through $\mathbb{F}_2[x; 2\mathbb{N}_0]$, $\mathbb{F}_2[x; \frac{2}{7}\mathbb{N}_0]$, $\mathbb{F}_2[x; \frac{2}{7^2}\mathbb{N}_0]$ of length 7, 49 and 343 respectively.

Table 10: Comparison between C_7 , C_{49} and C_{343}

(n, k)	d	t	R	$(3n, k_1)$	d_1	t_1	R_1	$(3^2n, k_2)$	d_2	t_2	R_2
$(7, 4)$	3	1	0.571	$(49, 28)$	3	1	0.571	$(343, 196)$	3	1	0.571
$(7, 1)$	5	2	0.143	$(49, 7)$	7	3	0.143	$(343, 49)$	7	3	0.143
				$(49, 4)$	21	10	0.081	$(343, 28)$	21	10	0.081
				$(49, 1)$	49	24	0.021	$(343, 7)$	49	24	0.021
								$(343, 4)$	147	73	0.012
								$(343, 1)$	343	171	0.002

Example 60 For a primitive polynomial $p(x^2) = 1 + (x^2) + (x^2)^4$ in $\mathbb{F}_2[x; 2\mathbb{Z}_0]$, we have non-primitive irreducible polynomials $p(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^{12}$ in $\mathbb{F}_2[x; \frac{2}{3}\mathbb{Z}_0]$ and $p(x^{\frac{2}{3^2}}) = 1 + (x^{\frac{2}{9}})^9 + (x^{\frac{2}{9}})^{36}$ in $\mathbb{F}_2[x; \frac{2}{3^2}\mathbb{Z}_0]$ (see Table 8), through which we get non-primitive BCH codes of length 45 and 135. Let $\alpha \in GF(2^{36})$, satisfies the relation $\alpha^{36} + \alpha^9 + 1 = 0$. Using this relation

we obtain the distinct powers of α in $GF(2^{36})$

Table 11: Distinct powers of α in $GF(2^{36})$

$\alpha^{36} = 1 + \alpha^9$	$\alpha^{52} = \alpha^{16} + \alpha^{25}$	$\alpha^{68} = \alpha^5 + \alpha^{14} + \alpha^{32}$	$\alpha^{84} = \alpha^{12} + \alpha^{30}$
$\alpha^{37} = \alpha + \alpha^{10}$	$\alpha^{53} = \alpha^{17} + \alpha^{26}$	$\alpha^{69} = \alpha^6 + \alpha^{15} + \alpha^{33}$	$\alpha^{85} = \alpha^{13} + \alpha^{31}$
$\alpha^{38} = \alpha^2 + \alpha^{11}$	$\alpha^{54} = \alpha^{18} + \alpha^{27}$	$\alpha^{70} = \alpha^7 + \alpha^{16} + \alpha^{34}$	$\alpha^{86} = \alpha^{14} + \alpha^{32}$
$\alpha^{39} = \alpha^3 + \alpha^{12}$	$\alpha^{55} = \alpha^{19} + \alpha^{28}$	$\alpha^{71} = \alpha^8 + \alpha^{17} + \alpha^{35}$	$\alpha^{87} = \alpha^{15} + \alpha^{33}$
$\alpha^{40} = \alpha^4 + \alpha^{13}$	$\alpha^{56} = \alpha^{20} + \alpha^{29}$	$\alpha^{72} = 1 + \alpha^{18}$	$\alpha^{88} = \alpha^{16} + \alpha^{34}$
$\alpha^{41} = \alpha^5 + \alpha^{14}$	$\alpha^{57} = \alpha^{21} + \alpha^{30}$	$\alpha^{73} = \alpha + \alpha^{19}$	$\alpha^{89} = \alpha^{17} + \alpha^{35}$
$\alpha^{42} = \alpha^6 + \alpha^{15}$	$\alpha^{58} = \alpha^{22} + \alpha^{31}$	$\alpha^{74} = \alpha^2 + \alpha^{20}$	$\alpha^{90} = 1 + \alpha^9 + \alpha^{18}$
$\alpha^{43} = \alpha^7 + \alpha^{16}$	$\alpha^{59} = \alpha^{23} + \alpha^{32}$	$\alpha^{75} = \alpha^3 + \alpha^{21}$	$\alpha^{91} = \alpha + \alpha^{10} + \alpha^{19}$
$\alpha^{44} = \alpha^8 + \alpha^{17}$	$\alpha^{60} = \alpha^{24} + \alpha^{33}$	$\alpha^{76} = \alpha^4 + \alpha^{22}$	$\alpha^{92} = \alpha^2 + \alpha^{11} + \alpha^{20}$
$\alpha^{45} = \alpha^9 + \alpha^{18}$	$\alpha^{61} = \alpha^{25} + \alpha^{34}$	$\alpha^{77} = \alpha^5 + \alpha^{23}$	$\alpha^{93} = \alpha^3 + \alpha^{12} + \alpha^{21}$
$\alpha^{46} = \alpha^{10} + \alpha^{19}$	$\alpha^{62} = \alpha^{26} + \alpha^{35}$	$\alpha^{78} = \alpha^6 + \alpha^{24}$	$\alpha^{94} = \alpha^4 + \alpha^{13} + \alpha^{22}$
$\alpha^{47} = \alpha^{11} + \alpha^{20}$	$\alpha^{63} = 1 + \alpha^9 + \alpha^{27}$	$\alpha^{79} = \alpha^7 + \alpha^{25}$	$\alpha^{95} = \alpha^5 + \alpha^{14} + \alpha^{23}$
$\alpha^{48} = \alpha^{12} + \alpha^{21}$	$\alpha^{64} = \alpha + \alpha^{10} + \alpha^{28}$	$\alpha^{80} = \alpha^8 + \alpha^{26}$	$\alpha^{96} = \alpha^6 + \alpha^{15} + \alpha^{24}$
$\alpha^{49} = \alpha^{13} + \alpha^{22}$	$\alpha^{65} = \alpha^2 + \alpha^{11} + \alpha^{29}$	$\alpha^{81} = \alpha^9 + \alpha^{27}$	$\alpha^{97} = \alpha^7 + \alpha^{16} + \alpha^{25}$
$\alpha^{50} = \alpha^{14} + \alpha^{23}$	$\alpha^{66} = \alpha^3 + \alpha^{12} + \alpha^{30}$	$\alpha^{82} = \alpha^{10} + \alpha^{28}$	$\alpha^{98} = \alpha^8 + \alpha^{17} + \alpha^{26}$
$\alpha^{51} = \alpha^{15} + \alpha^{24}$	$\alpha^{67} = \alpha^4 + \alpha^{13} + \alpha^{31}$	$\alpha^{83} = \alpha^{11} + \alpha^{29}$	$\alpha^{99} = \alpha^9 + \alpha^{18} + \alpha^{27}$

$\alpha^{100} = \alpha^{10} + \alpha^{19} + \alpha^{28}$	$\alpha^{112} = \alpha^4 + \alpha^{13} + \alpha^{22} + \alpha^{31}$	$\alpha^{124} = \alpha^7 + \alpha^{25} + \alpha^{34}$
$\alpha^{101} = \alpha^{11} + \alpha^{20} + \alpha^{29}$	$\alpha^{113} = \alpha^5 + \alpha^{14} + \alpha^{23} + \alpha^{32}$	$\alpha^{125} = \alpha^8 + \alpha^{26} + \alpha^{35}$
$\alpha^{102} = \alpha^{12} + \alpha^{21} + \alpha^{30}$	$\alpha^{114} = \alpha^6 + \alpha^{15} + \alpha^{24} + \alpha^{33}$	$\alpha^{126} = 1 + \alpha^{27}$
$\alpha^{103} = \alpha^{13} + \alpha^{22} + \alpha^{31}$	$\alpha^{115} = \alpha^7 + \alpha^{16} + \alpha^{25} + \alpha^{34}$	$\alpha^{127} = \alpha + \alpha^{28}$
$\alpha^{104} = \alpha^{14} + \alpha^{23} + \alpha^{32}$	$\alpha^{116} = \alpha^8 + \alpha^{17} + \alpha^{26} + \alpha^{35}$	$\alpha^{128} = \alpha^2 + \alpha^{29}$
$\alpha^{105} = \alpha^{15} + \alpha^{24} + \alpha^{33}$	$\alpha^{117} = 1 + \alpha^{18} + \alpha^{27}$	$\alpha^{129} = \alpha^3 + \alpha^{30}$
$\alpha^{106} = \alpha^{16} + \alpha^{25} + \alpha^{34}$	$\alpha^{118} = \alpha + \alpha^{19} + \alpha^{28}$	$\alpha^{130} = \alpha^4 + \alpha^{31}$
$\alpha^{107} = \alpha^{17} + \alpha^{26} + \alpha^{35}$	$\alpha^{119} = \alpha^2 + \alpha^{20} + \alpha^{29}$	$\alpha^{131} = \alpha^5 + \alpha^{32}$
$\alpha^{108} = 1 + \alpha^9 + \alpha^{18} + \alpha^{27}$	$\alpha^{120} = \alpha^3 + \alpha^{21} + \alpha^{30}$	$\alpha^{132} = \alpha^6 + \alpha^{33}$
$\alpha^{109} = \alpha + \alpha^{10} + \alpha^{19} + \alpha^{28}$	$\alpha^{121} = \alpha^4 + \alpha^{22} + \alpha^{31}$	$\alpha^{133} = \alpha^7 + \alpha^{34}$
$\alpha^{110} = \alpha^2 + \alpha^{11} + \alpha^{20} + \alpha^{29}$	$\alpha^{122} = \alpha^5 + \alpha^{23} + \alpha^{32}$	$\alpha^{134} = \alpha^8 + \alpha^{35}$
$\alpha^{111} = \alpha^3 + \alpha^{12} + \alpha^{21} + \alpha^{30}$	$\alpha^{123} = \alpha^6 + \alpha^{24} + \alpha^{33}$	$\alpha^{135} = 1$

Now, we calculate minimal polynomials to find the generating polynomial $g(x^{\frac{2}{9}})$. By [27, Theorem 4.4.2],

$$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}, \alpha^{64}, \alpha^{128}, \alpha^{121}, \alpha^{107}, \alpha^{79}, \alpha^{23}, \alpha^{46}, \alpha^{92}, \alpha^{49}, \alpha^{98}, \alpha^{61}, \alpha^{122},$$

$$\alpha^{109}, \alpha^{83}, \alpha^{31}, \alpha^{62}, \alpha^{124}, \alpha^{113}, \alpha^{91}, \alpha^{47}, \alpha^{94}, \alpha^{53}, \alpha^{106}, \alpha^{77}, \alpha^{19}, \alpha^{38}, \alpha^{76}, \alpha^{17}, \alpha^{34}, \alpha^{68}$$

all have same minimal polynomial $m'_1(x^{\frac{2}{9}}) = p(x^{\frac{2}{9}}) = 1 + (x^{\frac{2}{9}})^9 + (x^{\frac{2}{9}})^{36}$. Let $m'_3(x^{\frac{2}{9}})$ be the minimal polynomial for α^3 , then $\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}, \alpha^{96}, \alpha^{57}, \alpha^{114}, \alpha^{93}, \alpha^{51}, \alpha^{102}, \alpha^{69}$ all

are roots for $m'_3(x^{\frac{2}{9}})$. Therefore we get $m'_3(x^{\frac{2}{9}}) = (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^3 + 1$. Similarly we obtain

$$\begin{aligned}
m'_5(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{18} + (x^{\frac{2}{9}})^9 + 1 \\
m'_7(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{36} + (x^{\frac{2}{9}})^{27} + 1 \\
m'_9(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^4 + (x^{\frac{2}{9}}) + 1 \\
m'_{15}(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^6 + (x^{\frac{2}{9}})^3 + 1 \\
m'_{21}(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^9 + 1 \\
m'_{27}(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^4 + (x^{\frac{2}{9}})^3 + (x^{\frac{2}{9}})^2 + (x^{\frac{2}{9}}) + 1 \\
m'_{45}(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^2 + (x^{\frac{2}{9}}) + 1 \\
m'_{63}(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^4 + (x^{\frac{2}{9}})^3 + 1.
\end{aligned}$$

The BCH code with $d_2 = 3$ has generator polynomial

$$g(x^{\frac{2}{9}}) = 1 + (x^{\frac{2}{9}})^9 + (x^{\frac{2}{9}})^{36}.$$

It has minimum distance 3 and corrects up to 1 error. Since the generator polynomial is of degree 36, its code rate is $\frac{99}{135} = 0.733$. The BCH code with $d_2 = 5$ has generator polynomial

$$g(x^{\frac{2}{9}}) = (x^{\frac{2}{9}})^{48} + (x^{\frac{2}{9}})^{39} + (x^{\frac{2}{9}})^{36} + (x^{\frac{2}{9}})^{21} + (x^{\frac{2}{9}})^9 + (x^{\frac{2}{9}})^3 + 1.$$

It corrects up to 2 errors with $\frac{87}{135} = 0.644$ code rate.

The BCH code with $d_2 = 7$ has generator polynomial

$$\begin{aligned}
g(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{66} + (x^{\frac{2}{9}})^{54} + (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{36} + (x^{\frac{2}{9}})^{30} + (x^{\frac{2}{9}})^{27} \\
&\quad + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^3 + 1.
\end{aligned}$$

It corrects up to 3 errors and has code rate $R_2 = \frac{69}{135} = 0.5111$. The BCH code with $d_2 = 9$ has

generator polynomial

$$g(x^{\frac{2}{9}}) = (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{93} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{48} + (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^3 + 1.$$

It corrects up to 4 errors and has code rate $R_2 = \frac{33}{135} = 0.244$.

Similarly, BCH codes with $d_2 = 11, 17, 23, 29, 47$ and 65 has generator polynomials

$$\begin{aligned} g(x^{\frac{2}{9}}) = & (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{103} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{97} + (x^{\frac{2}{9}})^{93} + (x^{\frac{2}{9}})^{91} \\ & + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{61} + (x^{\frac{2}{9}})^{58} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{52} + (x^{\frac{2}{9}})^{48} \\ & + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{16} + (x^{\frac{2}{9}})^{13} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^7 \\ & + (x^{\frac{2}{9}})^3 + (x^{\frac{2}{9}}) + 1 \end{aligned}$$

$$\begin{aligned} g(x^{\frac{2}{9}}) = & (x^{\frac{2}{9}})^{112} + (x^{\frac{2}{9}})^{108} + (x^{\frac{2}{9}})^{105} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{100} + (x^{\frac{2}{9}})^{99} \\ & + (x^{\frac{2}{9}})^{94} + (x^{\frac{2}{9}})^{91} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{67} + (x^{\frac{2}{9}})^{63} + (x^{\frac{2}{9}})^{60} \\ & + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{55} + (x^{\frac{2}{9}})^{54} + (x^{\frac{2}{9}})^{49} + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} \\ & + (x^{\frac{2}{9}})^{22} + (x^{\frac{2}{9}})^{18} + (x^{\frac{2}{9}})^{15} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^{10} + (x^{\frac{2}{9}})^9 \\ & + (x^{\frac{2}{9}})^4 + (x^{\frac{2}{9}}) + 1 \end{aligned}$$

$$\begin{aligned} g(x^{\frac{2}{9}}) = & (x^{\frac{2}{9}})^{124} + (x^{\frac{2}{9}})^{121} + (x^{\frac{2}{9}})^{120} + (x^{\frac{2}{9}})^{109} + (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{105} \\ & + (x^{\frac{2}{9}})^{94} + (x^{\frac{2}{9}})^{91} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{79} + (x^{\frac{2}{9}})^{76} + (x^{\frac{2}{9}})^{75} \\ & + (x^{\frac{2}{9}})^{64} + (x^{\frac{2}{9}})^{61} + (x^{\frac{2}{9}})^{60} + (x^{\frac{2}{9}})^{49} + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} \\ & + (x^{\frac{2}{9}})^{34} + (x^{\frac{2}{9}})^{31} + (x^{\frac{2}{9}})^{30} + (x^{\frac{2}{9}})^{19} + (x^{\frac{2}{9}})^{16} + (x^{\frac{2}{9}})^{15} \\ & + (x^{\frac{2}{9}})^4 + (x^{\frac{2}{9}}) + 1 \end{aligned}$$

$$\begin{aligned}
g(x^{\frac{2}{9}}) = & (x^{\frac{2}{9}})^{128} + (x^{\frac{2}{9}})^{127} + (x^{\frac{2}{9}})^{126} + (x^{\frac{2}{9}})^{124} + (x^{\frac{2}{9}})^{120} + (x^{\frac{2}{9}})^{113} \\
& + (x^{\frac{2}{9}})^{112} + (x^{\frac{2}{9}})^{111} + (x^{\frac{2}{9}})^{109} + (x^{\frac{2}{9}})^{105} + (x^{\frac{2}{9}})^{98} + (x^{\frac{2}{9}})^{97} \\
& + (x^{\frac{2}{9}})^{96} + (x^{\frac{2}{9}})^{94} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{83} + (x^{\frac{2}{9}})^{82} + (x^{\frac{2}{9}})^{81} \\
& + (x^{\frac{2}{9}})^{79} + (x^{\frac{2}{9}})^{75} + (x^{\frac{2}{9}})^{68} + (x^{\frac{2}{9}})^{67} + (x^{\frac{2}{9}})^{66} + (x^{\frac{2}{9}})^{64} \\
& + (x^{\frac{2}{9}})^{60} + (x^{\frac{2}{9}})^{53} + (x^{\frac{2}{9}})^{52} + (x^{\frac{2}{9}})^{51} + (x^{\frac{2}{9}})^{49} + (x^{\frac{2}{9}})^{45} \\
& + (x^{\frac{2}{9}})^{38} + (x^{\frac{2}{9}})^{37} + (x^{\frac{2}{9}})^{36} + (x^{\frac{2}{9}})^{34} + (x^{\frac{2}{9}})^{30} + (x^{\frac{2}{9}})^{23} \\
& + (x^{\frac{2}{9}})^{22} + (x^{\frac{2}{9}})^{21} + (x^{\frac{2}{9}})^{19} + (x^{\frac{2}{9}})^{15} + (x^{\frac{2}{9}})^8 + (x^{\frac{2}{9}})^7 \\
& + (x^{\frac{2}{9}})^6 + (x^{\frac{2}{9}})^4 + 1
\end{aligned}$$

$$\begin{aligned}
g(x^{\frac{2}{9}}) = & (x^{\frac{2}{9}})^{130} + (x^{\frac{2}{9}})^{128} + (x^{\frac{2}{9}})^{125} + (x^{\frac{2}{9}})^{124} + (x^{\frac{2}{9}})^{122} + (x^{\frac{2}{9}})^{121} \\
& + (x^{\frac{2}{9}})^{120} + (x^{\frac{2}{9}})^{115} + (x^{\frac{2}{9}})^{113} + (x^{\frac{2}{9}})^{110} + (x^{\frac{2}{9}})^{109} + (x^{\frac{2}{9}})^{107} \\
& + (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{105} + (x^{\frac{2}{9}})^{100} + (x^{\frac{2}{9}})^{98} + (x^{\frac{2}{9}})^{95} + (x^{\frac{2}{9}})^{94} \\
& + (x^{\frac{2}{9}})^{92} + (x^{\frac{2}{9}})^{91} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{85} + (x^{\frac{2}{9}})^{83} + (x^{\frac{2}{9}})^{80} + (x^{\frac{2}{9}})^{79} \\
& + (x^{\frac{2}{9}})^{77} + (x^{\frac{2}{9}})^{76} + (x^{\frac{2}{9}})^{75} + (x^{\frac{2}{9}})^{70} + (x^{\frac{2}{9}})^{68} + (x^{\frac{2}{9}})^{65} + (x^{\frac{2}{9}})^{64} \\
& + (x^{\frac{2}{9}})^{62} + (x^{\frac{2}{9}})^{61} + (x^{\frac{2}{9}})^{60} + (x^{\frac{2}{9}})^{55} + (x^{\frac{2}{9}})^{53} + (x^{\frac{2}{9}})^{50} \\
& + (x^{\frac{2}{9}})^{49} + (x^{\frac{2}{9}})^{47} + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{40} + (x^{\frac{2}{9}})^{38} \\
& + (x^{\frac{2}{9}})^{35} + (x^{\frac{2}{9}})^{34} + (x^{\frac{2}{9}})^{32} + (x^{\frac{2}{9}})^{31} + (x^{\frac{2}{9}})^{30} + (x^{\frac{2}{9}})^{25} \\
& + (x^{\frac{2}{9}})^{23} + (x^{\frac{2}{9}})^{20} + (x^{\frac{2}{9}})^{19} + (x^{\frac{2}{9}})^{17} + (x^{\frac{2}{9}})^{16} + (x^{\frac{2}{9}})^{15} \\
& + (x^{\frac{2}{9}})^{10} + (x^{\frac{2}{9}})^8 + (x^{\frac{2}{9}})^5 + (x^{\frac{2}{9}})^4 + (x^{\frac{2}{9}})^2 + (x^{\frac{2}{9}}) + 1 \text{ and} \\
g(x^{\frac{2}{9}}) = & (x^{\frac{2}{9}})^{134} + (x^{\frac{2}{9}})^{133} + \dots + (x^{\frac{2}{9}})^2 + (x^{\frac{2}{9}}) + 1.
\end{aligned}$$

They corrects upto 7, 10, 13, 22, 31 and 67 errors with code rates 0.215, 0.170, 0.0814, 0.0518, 0.0370 and 0.007 respectively.

Following are the tables of primitive and non-primitive BCH codes of length 15, 45 and 135.

Table 12: BCH codes of length 15, 45 and 135

(n, k)	d	t	R	(n, k)	d_1	t_1	R_1	(n, k)	d_2	t_2	R_2
(15, 11)	3	1	0.733	(45, 33)	3	1	0.733	(135, 99)	3	1	0.733
(15, 7)	5	2	0.466	(45, 29)	5	2	0.644	(135, 87)	5	2	0.644
(15, 5)	7	3	0.333	(45, 23)	7	3	0.511	(135, 69)	7	3	0.511
(15, 1)	15	7	0.066	(45, 11)	9	4	0.244	(135, 33)	9	4	0.244
				(45, 7)	15	7	0.155	(135, 29)	15	7	0.215
				(45, 5)	21	10	0.11	(135, 23)	21	10	0.170
				(45, 1)	45	22	0.022	(135, 11)	27	13	0.0814
								(135, 7)	45	22	0.0518
								(135, 5)	63	31	0.0370
								(135, 1)	135	67	0.007

From example 59, it is clear that the code generated through $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ corrects more errors and has better code rate than the code generated through $\mathbb{F}_2[x; a\mathbb{N}_0]$.

4.2 Link between primitive and a family of non-primitive BCH codes

Now we are in position to develop a link between a primitive $(n, n - r)$ binary BCH code \mathcal{C}_n and a non-primitive $(b^j n, b^j n - r_j)$ binary BCH code $\mathcal{C}_{b^j n}$, where r and r_j are respectively the degrees of their generating polynomials $g(x^a)$ and $g(x^{\frac{a}{b^j}})$. From Theorem 52(1), it follows that the generalized polynomial $g(x^{\frac{a}{b^j}}) \in \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ divides $(x^{\frac{a}{b^j}})^{b^j n} - 1$ in $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$. So, there is a non-primitive BCH code $\mathcal{C}_{b^j n}$ generated by $g(x^{\frac{a}{b^j}})$ in $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]_{b^j n}$. By the same argument, as $b^j n$ divides $n_j = 2^{b^j s} - 1$, so $(x^{\frac{a}{b^j}})^{b^j n} - 1$ divides $(x^{\frac{a}{b^j}})^{n_j} - 1$ in $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$. It follows that $((x^{\frac{a}{b^j}})^{n_j} - 1) \subset ((x^{\frac{a}{b^j}})^{b^j n} - 1)$. Consequently, third isomorphism theorem for rings gives

$$\frac{\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]/((x^{\frac{a}{b^j}})^{n_j} - 1)}{((x^{\frac{a}{b^j}})^{b^j n} - 1)/((x^{\frac{a}{b^j}})^{n_j} - 1)} \simeq \frac{\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]}{((x^{\frac{a}{b^j}})^{b^j n} - 1)} \simeq \frac{\mathbb{F}_2[x; a\mathbb{N}_0]}{(x^a)^n - 1}$$

Thus there are embeddings $\mathcal{C}_n \hookrightarrow \mathcal{C}_{b^j n} \hookrightarrow \mathcal{C}_{n_j}$ of codes, whereas $\mathcal{C}_n, \mathcal{C}_{b^j n}, \mathcal{C}_{n_j}$ are respectively primitive BCH, non-primitive BCH and primitive BCH codes. Whereas the embeddings $\mathcal{C}_n \hookrightarrow \mathcal{C}_{b^j n}$ are defined as:

$$a(x^a) = a_0 + a_1(x^a) + \dots + a_{n-1}(x^a)^{n-1} \mapsto a_0 + a_1(x^{\frac{a}{b^j}})^{b^j} + \dots + a_{n-1}(x^{\frac{a}{b^j}})^{b^j(n-1)} = a(x^{\frac{a}{b^j}}),$$

where $a(x^a) \in \mathcal{C}_n$ and $a(x^{\frac{a}{b^j}}) \in \mathcal{C}_{b^j n}$.

Also, if $g(x^{\frac{a}{b^{j-1}}})$ is the generator polynomial of the binary non-primitive BCH code $\mathcal{C}_{b^{j-1}n}^{j-1}$ in $\mathbb{F}_2[x; \frac{a}{b^{j-1}}\mathbb{Z}_{\geq 0}]_{b^{j-1}n}$, then $g(x^{\frac{a}{b^j}})$ is the generator polynomial of the binary non-primitive BCH code $\mathcal{C}_{b^j n}^j$ in the monoid ring $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{Z}_{\geq 0}]_{b^j n}$. Thus the non-primitive BCH code $\mathcal{C}_{b^{j-1}n}^{j-1}$ is embedded in non-primitive BCH code $\mathcal{C}_{b^j n}^j$ under the monomorphism defined as; $a(x^{\frac{a}{b^{j-1}}}) \mapsto a(x^{\frac{a}{b^j}})$.

The above discussion shape the following.

Theorem 61 *Let \mathcal{C}_n be a primitive binary BCH code of length $n = 2^s - 1$ generated by r degree polynomial $g(x^a)$ in $\mathbb{F}_2[x; a\mathbb{N}_0]$. Then*

1) *there exist a $b^j n$ length binary non-primitive BCH code $\mathcal{C}_{b^j n}$ generated by $b^j r$ degree polynomial $g(x^{\frac{a}{b^j}})$ in $\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]$;*

2) *the binary primitive BCH code \mathcal{C}_n is embedded in the binary non-primitive BCH code $\mathcal{C}_{b^j n}$, for each $j \geq 1$,*

3) *the binary BCH codes of the sequence $\{\mathcal{C}_{b^j n}^j\}_{j \geq 1}$ have the following embedding $\mathcal{C}_{b^n}^1 \hookrightarrow \dots \hookrightarrow \mathcal{C}_{b^j n}^j \hookrightarrow \dots$.*

Hence we have the following relationships

$$\begin{array}{ccccccc} \mathbb{F}_2[x; a\mathbb{N}_0] & \subset & \mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0] & \subset & \mathbb{F}_2[x; \frac{a}{b^2}\mathbb{N}_0] & \subset & \dots & \mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0] \\ \frac{\mathbb{F}_2[x; a\mathbb{N}_0]}{((x^a)^n - 1)} & \simeq & \frac{\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]}{((x^{\frac{a}{b}})^{bn} - 1)} & \simeq & \frac{\mathbb{F}_2[x; \frac{a}{b^2}\mathbb{N}_0]}{((x^{\frac{a}{b^2}})^{b^2 n} - 1)} & \simeq & \dots & \frac{\mathbb{F}_2[x; \frac{a}{b^j}\mathbb{N}_0]}{((x^{\frac{a}{b^j}})^{b^j n} - 1)} \\ \cup & & \cup & & \cup & & \dots & \cup \\ \mathcal{C}_n & \hookrightarrow & \mathcal{C}_{bn}^1 & \hookrightarrow & \mathcal{C}_{b^2 n}^2 & \hookrightarrow & \dots & \mathcal{C}_{b^j n}^j \end{array}$$

Remark 62 $g(x^{\frac{a}{b}})$ can be deduced from $g(x^{\frac{a}{b^j}})$ by substituting $x^{\frac{a}{b^j}} = y$ and then replacing y by $y^{b^{j-1}} = x^{\frac{a}{b}}$.

Example 63 In Example 59, on letting $(x^{\frac{2}{9}}) = y$, that is $x^{\frac{2}{3}} = y^3$, we get

$$\begin{aligned}
g(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{103} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{97} + (x^{\frac{2}{9}})^{93} + (x^{\frac{2}{9}})^{91} + (x^{\frac{2}{9}})^{90} + \\
&\quad (x^{\frac{2}{9}})^{61} + (x^{\frac{2}{9}})^{58} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{52} + (x^{\frac{2}{9}})^{48} + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} + \\
&\quad (x^{\frac{2}{9}})^{16} + (x^{\frac{2}{9}})^{13} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^7 + (x^{\frac{2}{9}})^3 + (x^{\frac{2}{9}}) + 1 \\
g(y^3) &= (y^3)^{106} + (y^3)^{103} + (y^3)^{102} + (y^3)^{97} + (y^3)^{93} + (y^3)^{91} + (y^3)^{90} + \\
&\quad (y^3)^{61} + (y^3)^{58} + (y^3)^{57} + (y^3)^{52} + (y^3)^{48} + (y^3)^{46} + (y^3)^{45} + \\
&\quad (y^3)^{16} + (y^3)^{13} + (y^3)^{12} + (y^3)^7 + (y^3)^3 + (y^3) + 1 \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{106} + (x^{\frac{2}{3}})^{103} + (x^{\frac{2}{3}})^{102} + (x^{\frac{2}{3}})^{97} + (x^{\frac{2}{3}})^{93} + (x^{\frac{2}{3}})^{91} + (x^{\frac{2}{3}})^{90} \\
&\quad + (x^{\frac{2}{3}})^{61} + (x^{\frac{2}{3}})^{58} + (x^{\frac{2}{3}})^{57} + (x^{\frac{2}{3}})^{52} + (x^{\frac{2}{3}})^{48} + (x^{\frac{2}{3}})^{46} + (x^{\frac{2}{3}})^{45} \\
&\quad + (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} \\
&\quad + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}}) + 1 \\
&= (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}}) + 1 \in \mathbb{F}_2[x; \frac{2}{3}\mathbb{N}_0]_{45}
\end{aligned}$$

Similarly, for

$$\begin{aligned}
g(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{112} + (x^{\frac{2}{9}})^{108} + (x^{\frac{2}{9}})^{105} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{100} + (x^{\frac{2}{9}})^{99} + (x^{\frac{2}{9}})^{94} + \\
&\quad (x^{\frac{2}{9}})^{91} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{67} + (x^{\frac{2}{9}})^{63} + (x^{\frac{2}{9}})^{60} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{55} + \\
&\quad (x^{\frac{2}{9}})^{54} + (x^{\frac{2}{9}})^{49} + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{22} + (x^{\frac{2}{9}})^{18} + (x^{\frac{2}{9}})^{15} + \\
&\quad (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^{10} + (x^{\frac{2}{9}})^9 + (x^{\frac{2}{9}})^4 + (x^{\frac{2}{9}}) + 1
\end{aligned}$$

we have

$$g(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{22} + (x^{\frac{2}{3}})^{18} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^9 + (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}}) + 1,$$

which is the generating polynomial of BCH code $(45, 23)$ having design distance $d_2 = 6, 7$.

In this way we can obtain the non-primitive binary BCH code \mathcal{C}_{45} from non-primitive binary BCH code \mathcal{C}_{135} .

From Examples 59 and 60 we deduce the following lemma that explains the relationship

Clearly the binary bits of v^1 are properly overlapped on the bits of v^2 , in fact it is repeating three time after a particular pattern. Hence the generating matrix G_2 of $g(x^{\frac{2}{3}})$ will contain the generating matrix G_1 of $g(x^{\frac{2}{3}})$ such that $G_2 = \oplus_1^3 G_1$.

4.2.1 General Decoding Principle

The binary BCH code C_n is embedded in the binary non-primitive BCH code $C_{b^j n}$ for any positive integer $1 \leq j \leq m$. So description of the decoding procedure of the code $C_{b^j n}$ for any fixed positive integer $1 \leq j \leq m$ is given. We use the decoding procedure which follows the same principle as of the primitive binary BCH code.

Take $a^j \in F_2^{b^j n}$ as a received vector. We obtain the syndrome matrix of a^j , $S(a^j) = a^j H^T$. In this way, we calculate a table of syndromes which is useful in determining the error vector e such that $S(a^j) = S(e)$. So the decoding of received vector a^j has done as the transmitted vector $v^j = a^j - e$. We adopt the algebraic method for finding e from the syndrome vector $S(a^j)$.

Let $C_{b^j n}$ be the binary non-primitive BCH code with length $b^j n$ and designed distance d^j . Let H be the $(d^j - 1) \times b^j n$ matrix over $F_{2^{b^j s}}$. The syndrome of $a^j \in F_2^{b^j n}$ as $S(a^j) = a^j H^T$. The polynomial form of $a^j = (a_0^j, a_1^j, \dots, a_{b^j n-1}^j)$ is $a^j(x^{\frac{a}{b^j}}) = a_0^j + a_1^j(x^{\frac{a}{b^j}}) + a_2^j(x^{\frac{a}{b^j}})^2 + \dots + a_{b^j n-1}^j(x^{\frac{a}{b^j}})^{b^j n-1}$. So

$$S(a^j) = \begin{bmatrix} a_0^j & a_1^j & \dots & a_{b^j n-1}^j \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha^{c^j} & \alpha^{c^j+1} & \dots & \alpha^{c^j+d^j-2} \\ \alpha^{2c^j} & \alpha^{2(c^j+1)} & \dots & \alpha^{2(c^j+d^j-2)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(b^j n-1)c^j} & \alpha^{(b^j n-1)(c^j+1)} & \dots & \alpha^{(b^j n-1)(c^j+d^j-2)} \end{bmatrix} \quad (4.1)$$

$$S(a^j) = \begin{bmatrix} S_{c^j} & S_{c^j+1} & \dots & S_{c^j+d^j-2} \end{bmatrix},$$

where $S_k = a_0^j + a_1^j \alpha^k + \dots + a_{b^j n-1}^j \alpha^{(b^j n-1)k} = a^j(\alpha^k)$ for $k = c^j, c^j + 1, \dots, c^j + d^j - 2$.

Now, let a codeword $v \in C_{b^j n}$ is transmitted and the vector received is $a^j = v^j + e$, where e is the error vector. Then $S(e) = S(a^j)$. Let $e(x^{\frac{a}{b^j}}) = e_0 + e_1(x^{\frac{a}{b^j}}) + e_2(x^{\frac{a}{b^j}})^2 + \dots + e_{b^j n-1}(x^{\frac{a}{b^j}})^{b^j n-1}$

be the error polynomial. Suppose i_1, \dots, i_l be the positions where an error has occurred. Then $e_i \neq 0$ if and only if $i \in I = \{i_1, \dots, i_l\}$. Hence $e(x^{\frac{a}{b^j}}) = \sum_{i \in I} e_i (x^{\frac{a}{b^j}})^i$. Since the code corrects upto t errors, where $t = \left\lfloor \frac{d^j - 1}{2} \right\rfloor$. So we assume $l \leq t$, that is $2l < d^j$. Since $S(e) = S(a^j)$, we have $e(\alpha^k) = S_k$ for $k = c^j, c^j + 1, \dots, c^j + d^j - 2$. Thus the $2l$ unknowns i_1, \dots, i_l and e_{i_1}, \dots, e_{i_l} satisfy the following system of $d^j - 1$ linear equations in e_{i_1}, \dots, e_{i_l} :

$$\sum_{i \in I} e_i \alpha^{ji} = S_j, j = c^j, c^j + 1, \dots, c^j + d^j - 2. \quad ((i))$$

We first obtain a solution for the error positions i_1, \dots, i_l . We define the error locator polynomial $f(x^{\frac{a}{b^j}}) = f_0 + f_1(x^{\frac{a}{b^j}}) + f_2(x^{\frac{a}{b^j}})^2 + \dots + f_{l-1}(x^{\frac{a}{b^j}})^{l-1} + (x^{\frac{a}{b^j}})^l$. Since $f(\alpha^i) = 0$ for each $i \in I$, we have

$$f_0 + f_1(\alpha^i) + \dots f_{l-1}(\alpha^i)^{l-1} + (\alpha^i)^l = 0. \quad (4.2)$$

On multiplying this equation by $e_i \alpha^{ki}$, we get

$$f_0 e_i \alpha^{ki} + f_1 e_i \alpha^{(k+1)i} + \dots f_{l-1} e_i \alpha^{(k+l-1)i} + e_i \alpha^{(k+l)i} = 0, \quad (4.3)$$

for each $i \in I$. Summing these l equations for $i = i_1, \dots, i_l$ and using the relations (i), we have

$$f_0 S_k + f_1 S_{k+1} + \dots f_{l-1} S_{k+l-1} + S_{k+l} = 0, \quad (4.4)$$

for each $j = c^j, c^j + 1, \dots, c^j + l - 1$. Thus the l unknowns f_0, f_1, \dots, f_{l-1} satisfy the following $l \times l$ system of linear equations:

$$\begin{bmatrix} S_{c^j} & S_{c^j+1} & \dots & S_{c^j+l-1} \\ S_{c^j+1} & S_{c^j+2} & \dots & S_{c^j+l} \\ \vdots & \vdots & \ddots & \vdots \\ S_{c^j+l-1} & S_{c^j+l} & \dots & S_{c^j+2l-2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{l-1} \end{bmatrix} = \begin{bmatrix} S_{c^j+l} \\ S_{c^j+l+1} \\ \vdots \\ S_{c^j+2l-1} \end{bmatrix}. \quad ((ii))$$

Let S denotes the coefficient matrix in the above linear system. It can be verified by direct computation that $S = VDV^T$, where

$$V = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \dots & \alpha^{i_l} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{i_1(l-1)} & \alpha^{i_2(l-1)} & \dots & \alpha^{i_l(l-1)} \end{bmatrix}, \quad D = \begin{bmatrix} e_{i_1} \alpha^{i_1 c} & 0 & \dots & 0 \\ 0 & e_{i_2} \alpha^{i_2 c} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e_{i_l} \alpha^{i_l c} \end{bmatrix}.$$

The matrix V is a Vandermonde matrix. Since α is a non-primitive $b^j n$ th root of unity in $F_{2^{bj_s}}$ and i_1, \dots, i_l are distinct integers in $\{0, \dots, b^j n - 1\}$, we have $\alpha^{i_1}, \dots, \alpha^{i_l}$ are all distinct. Hence $\det V \neq 0$. Further, e_{i_1}, \dots, e_{i_j} are all nonzero and hence $\det D \neq 0$. Therefore $\det S \neq 0$, and linear system (ii) has a unique solution.

We have assumed that the number of positions where an error has occurred is $z \leq t$. If the actual number of error positions is less than z , then for any choice of distinct positions i_1, \dots, i_z , the coefficients e_{i_1}, \dots, e_{i_z} cannot be all zero. So $\det D \neq 0$. Hence z is the greatest positive integer $\leq t$ such that system (ii) has a unique solution. Therefore we find the value of z by taking successively $z = t, t-1, \dots$ in system (ii) until we have a value for which system (ii) has a unique solution, which gives us the error locator polynomial $f(x^{\frac{a}{b^j}}) = f_0 + f_1(x^{\frac{a}{b^j}}) + f_2(x^{\frac{a}{b^j}})^2 + \dots + f_{z-1}(x^{\frac{a}{b^j}})^{z-1} + (x^{\frac{a}{b^j}})^z$. Now to find the roots of $f(x^{\frac{a}{b^j}})$, we put $x^{\frac{a}{b^j}} = \alpha^i$, $i = 0, 1, \dots$. By the definition of $f(x^{\frac{a}{b^j}})$, these roots are $\alpha^{i_1}, \dots, \alpha^{i_z}$. Thus we find the unique solution for the unknowns i_1, \dots, i_z . Having thus found the error vector e , we decode the received vector a as the codeword $v^j = a^j - e$.

To compute the syndrome of a binary BCH code we have $S_2 = (S_1)^2$, $S_6 = (S_3)^2$ and so on. We can compute the syndrome more easily by using the division algorithm. If $m(x^{\frac{a}{b^j}})$ is the minimal polynomial of α , then $S_1 = a^j(\alpha)$ can be obtained by finding the remainder on dividing $a^j(x^{\frac{a}{b^j}})$ by $m(x^{\frac{a}{b^j}})$ and then putting $x^{\frac{a}{b^j}} = \alpha$ in it. In general, to find S_k , we divide $a^j(x^{\frac{a}{b^j}})$ by $m(x^{\frac{a}{b^j}})$ and find the remainder.

The decoding of the code C_{bn} from the decoding of the code $C_{b^j n}$ can be obtain as; take $x^{\frac{a}{b^j}} = y$, which gives $x^{\frac{a}{b}} = y^{b^{j-1}}$. In this way the code polynomial $v^j(x^{\frac{a}{b^j}})$ in $F_2[x; \frac{a}{b^j} \mathbb{N}_0]_{b^j n}$ becomes $v^j(y)$. Again on replacing y by $y^{b^{j-1}}$, we get $v^j(y^{b^{j-1}}) = v^j(x^{\frac{a}{b}})$. The remainder after dividing $v^j(x^{\frac{a}{b}})$ by $(x^{\frac{a}{b}})^{bn} - 1$, will be the decoded vector of $F_2[x; \frac{a}{b} \mathbb{N}_0]_{bn}$ and the generator polynomial $g(x^{\frac{a}{b}})$ divides $v^j(x^{\frac{a}{b}})$.

The above discussion can be sum up in the following steps.

Step I: For binary non-primitive BCH code C_{b^j-1n} with designed distance d^j , let $a^j(x^{\frac{a}{b^j}})$ be the received polynomial with l errors, where $l \leq t_j$.

Step II: Compute the syndromes and find the value of l , such that the system (2) has a unique solution.

Step III: Step II gives us the error locator polynomial $f(x^{\frac{a}{b^j}})$. Now find the roots of $f(x^{\frac{a}{b^j}})$ through which we obtain the error polynomial $e(x^{\frac{a}{b^j}})$.

Step IV: We decode the received polynomial $a^j(x^{\frac{a}{b^j}})$ as $v^j(x^{\frac{a}{b^j}}) = a^j(x^{\frac{a}{b^j}}) - e(x^{\frac{a}{b^j}})$.

Step V: The code vector v^{j-1} in $C_{b^{j-1}n}$ can be drag out from the decoded code vector v^j in $C_{b^j n}$ by putting $x^{\frac{a}{b^j}} = y$ in corresponding code polynomial $v^j(x^{\frac{a}{b^j}})$. This gives $v^j(x^{\frac{a}{b^j}}) = v^j(y)$. Again by replacing y by y^b we get $v^j(y) = v^j(y^b) = v^j(x^{\frac{a}{b^{j-1}}})$.

Step VI: Divide $v^j(x^{\frac{a}{b^{j-1}}})$ by $(x^{\frac{a}{b^{j-1}}} - 1)^{b^{j-1}n-1}$, the remainder $v^j(x^{\frac{a}{b^{j-1}}})$ will be in $F_2[x; \frac{a}{b^{j-1}}\mathbb{N}_0]_{b^{j-1}n}$, and the generator polynomial $g(x^{\frac{a}{b^{j-1}}})$ divides $v^j(x^{\frac{a}{b^{j-1}}})$. Then its corresponding vector $v^j \in C_{b^{j-1}n}$.

Step VII: If we replace y by y^{b^j} we get $v^j(y) = v^j(y^{b^j}) = v^j(x^a)$. So on dividing $v^j(x^a)$ by $(x^a)^n - 1$, the remainder $v^j(x^a)$ will be in $F_2[x; a\mathbb{N}_0]_n$, and the generator polynomial $g(x^a)$ divides $v^j(x^a)$. Then its corresponding vector $v^j \in C_n$.

Illustration

Let C_{135} be a (135, 29) binary non-primitive BCH code with designed distance $d = 4$. Assume that

$$\begin{aligned} a^2(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{103} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{97} + (x^{\frac{2}{9}})^{93} + (x^{\frac{2}{9}})^{91} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{61} \\ &\quad + (x^{\frac{2}{9}})^{58} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{52} + (x^{\frac{2}{9}})^{48} + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{38} + (x^{\frac{2}{9}})^{16} \\ &\quad + (x^{\frac{2}{9}})^{13} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^7 + (x^{\frac{2}{9}})^3 + (x^{\frac{2}{9}})^2 + (x^{\frac{2}{9}}) + 1 \end{aligned}$$

is the received polynomial. The error position $l = 2$ and the syndromes are $S_1 = a^2(\alpha) = \alpha^{11}$, $S_2 = (S_1)^2 = \alpha^{22}$, $S_3 = a^2(\alpha^3) = \alpha^{105}$ and $S_4 = (S_2)^2 = \alpha^{44}$. The error locator polynomial is given by $f(x^{\frac{2}{9}}) = f_0 + f_1(x^{\frac{2}{9}}) + (x^{\frac{2}{9}})^2$. Then we have the following system of equations for f_0 , f_1 .

$$\begin{bmatrix} \alpha^{11} & \alpha^{22} \\ \alpha^{22} & \alpha^{105} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} \alpha^{105} \\ \alpha^{44} \end{bmatrix}$$

$$\begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} \frac{\alpha^{105}}{\alpha^{62}} & \frac{\alpha^{22}}{\alpha^{62}} \\ \frac{\alpha^{22}}{\alpha^{14}} & \frac{\alpha^{11}}{\alpha^{14}} \end{bmatrix} \begin{bmatrix} \alpha^{105} \\ \alpha^{44} \end{bmatrix} = \begin{bmatrix} \alpha^{40} \\ \alpha^{11} \end{bmatrix}.$$

Hence the error locator polynomial is $f(x^{\frac{2}{9}}) = \alpha^{40} + \alpha^{11}(x^{\frac{2}{9}}) + (x^{\frac{2}{9}})^2$. Trying successively $x = 1, \alpha, \alpha^2, \dots$, we find that α^2 and α^{38} are the roots. Hence the error polynomial is $e(x^{\frac{2}{9}}) = (x^{\frac{2}{9}})^2 + (x^{\frac{2}{9}})^{38}$. Thus we decode $a^2(x^{\frac{2}{9}})$ as

$$\begin{aligned} v^2(x^{\frac{2}{9}}) &= a''(x^{\frac{2}{9}}) + e(x^{\frac{2}{9}}) = (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{103} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{97} + (x^{\frac{2}{9}})^{93} + \\ &\quad (x^{\frac{2}{9}})^{91} + (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{61} + (x^{\frac{2}{9}})^{58} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{52} + (x^{\frac{2}{9}})^{48} + (x^{\frac{2}{9}})^{46} \\ &\quad + (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{16} + (x^{\frac{2}{9}})^{13} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^7 + (x^{\frac{2}{9}})^3 + (x^{\frac{2}{9}}) + 1. \end{aligned}$$

Now letting $x^{\frac{2}{9}} = y$, this gives $y^3 = x^{\frac{2}{3}}$, we get

$$\begin{aligned} v^2(y^3) &= (y^3)^{106} + (y^3)^{103} + (y^3)^{102} + (y^3)^{97} + (y^3)^{93} + (y^3)^{91} + (y^3)^{90} + (y^3)^{61} \\ &\quad + (y^3)^{58} + (y^3)^{57} + (y^3)^{52} + (y^3)^{48} + (y^3)^{46} + (y^3)^{45} + (y^3)^{16} + (y^3)^{13} \\ &\quad + (y^3)^{12} + (y^3)^7 + (y^3)^3 + (y^3) + 1 \\ v^2(y^3) &= v^2(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{106} + (x^{\frac{2}{3}})^{103} + (x^{\frac{2}{3}})^{102} + (x^{\frac{2}{3}})^{97} + (x^{\frac{2}{3}})^{93} + (x^{\frac{2}{3}})^{91} + \\ &\quad (x^{\frac{2}{3}})^{90} + (x^{\frac{2}{3}})^{61} + (x^{\frac{2}{3}})^{58} + (x^{\frac{2}{3}})^{57} + (x^{\frac{2}{3}})^{52} + (x^{\frac{2}{3}})^{48} + (x^{\frac{2}{3}})^{46} + \\ &\quad (x^{\frac{2}{3}})^{45} + (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}}) + 1. \end{aligned}$$

Where $v^2(x^{\frac{2}{3}}) \in F_2[x; \frac{2}{3}\mathbb{N}_0]_{45}$.

Now, after dividing $v^2(x^{\frac{2}{3}})$ by $(x^{\frac{2}{3}})^{45} - 1$, we obtain the remainder $v^2(x^{\frac{2}{3}})$ as

$$v^2(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}}) + 1 \in C_{45},$$

where $v^2(x^{\frac{2}{3}})$ is the generator polynomial of non-primitive binary BCH code (45, 29). Again on letting $x^{\frac{2}{3}} = y$, this gives $y^3 = x^2$, we get $v^2(x^2) = (x^2)^{13} + (x^2)^{12} + (x^2)^7 + (x^2)^3 + 1 \in C_{15}$, where C_{15} is primitive binary BCH code (15, 11), it is due to the reason that the generator polynomial $g(x^2) = (x^2)^4 + (x^2) + 1$ divides $v(x^2)$.

4.3 The Algorithm

In this section we propose algorithm in order to calculate non primitive BCH codes of length $b^j n$ using primitive BCH code of length n . Both of the algorithm that is: encoding and decoding of codes is carried out in Matlab. The process of developing algorithm is divided into two major steps, i.e., Encoding of non-primitive BCH code of length $b^j n$ and its decoding.

4.3.1 Encoding of non-primitive BCH code of length $b^j n$

In encoding we first calculate primitive polynomial of degree s by invoking Matlab's built in command "*bchgenpoly*". After this operation, non-primitive polynomial of degree bs is calculated. With the help of its roots say α' , elements of Galois field $GF(2^{bn})$ are calculated such that the unity is reached that is $(\alpha')^{b^j n} = 1$. Through these elements cyclotomic cosets are determined which gives all the minimal polynomials. Finally we get non-primitive polynomial with the help of these minimal polynomials. Then design distances are calculated through which number of errors that can be corrected in each BCH codes is determined.

Various modules are developed in order to achieve specified result. Table 13 shows list of these methods and its description.

Table 13: Encoding modulation description

	Module	Input	Output
a	Elements of Galois Field	$b * n$	$\alpha'[i]$
b	Bchgenpoly	n, k	$g[i]$
c	Cyclotomic_cosets	$\alpha'[i], b, k$	$c[i]$
d	Design_distance	$c[i]$	d
e	error	d	t

Algorithm routines are explained as follows:

a. Elements of Galois field (alpha array $\alpha'[i]$)

This module calculates all the elements of Galois field $GF(2^{bn})$ using a root say α' of non-primitive polynomial of degree bs , such that it gives identity at power bn . We call them alpha array $\alpha'[i]$ and index array $A[index]$. Given input is $b * n$, the non-primitive polynomial of degree bs gives the first element of the array $\alpha'[i]$. By increasing its power each element of the array $\alpha'[i]$ is calculated in outer loop. Then in nested while loop their corresponding values are determined in such a way that if the value of any element in the array exceed bs we take the remainder $rem(index, bs)$ as mentioned in line 9 of the algorithm. The loop breaks when the condition for identity is met.

BEGIN

1 INPUT b and n

2 $bn \leftarrow b \times n$

3 Initialize $A[index]$ from 1 to $bn - 1$

4 Initialize $\alpha'[i] \leftarrow 0$

5 Initialize index $\leftarrow 1$

6 WHILE index $\neq 0$

7 mark $A[index] \leftarrow 0$

8 Initialize $v \leftarrow index$

9 Calculate next candidate, p_i , by $rem(index, bs)$

10 $\alpha'[i] \leftarrow v$

```

11      WHILE current position of  $A[index] = 0$ 
12          IF current position  $< A[index]$  size
13              increment  $i$ 
14          ELSEIF
15               $i \leftarrow 0$ 
16          BREAK
17      ENDIF
18  ENDWHILE
19  ENDWHILE
END

```

b. bchgenpoly ($g[i]$):

Its a Matlab built in function and its complete documentation is found under <http://www.mathworks.com/help>

In this module we get the generator polynomial of primitive BCH code of length n .

```

1 INPUT  $n, k$ 
2 OUTPUT  $g[i]$ 

```

c. Cyclotomic_cosets($c[i]$):

Given $\alpha'[i]$, dimension of code k and positive integer b , cyclotomic cosets $c[i]$ are calculated. Length of $c[i]$ is initialized to at max $b * k$, in short all elements should not exceed the maximum length. The loop start from 2 to max length and calculate unique values in given $\alpha'[i]$. The process stops when we get sum of two elements = 2. Once cyclotomic cosets are calculated we can calculate minimal polynomials for BCH code.

```

BEGIN
1  INPUT  $\alpha'[i], b, k$ 
2  Initialize cal_coset  $\leftarrow 0$ 
3  Initialize len_cal_coset  $\leftarrow 0$ 
4  Initialize code_length  $\leftarrow b * k$ 
5  Initialize len_coset  $\leftarrow$  length of  $c[i]$ 
6  Initialize code  $\leftarrow b * k$ 
7  FOR  $i \leftarrow 2$  to len_coset
8      cal_coset  $\leftarrow c[i]$  at position  $i$ 

```

```

9          IF  $i \neq 2$  THEN
10             code_length  $\leftarrow bk$  cal_coset
11          ENDIF
12          PRINT  $c[i]$ 
13      END FOR
END

```

d. non_prim_gen_poly($g'[i]$)

Given primitive polynomial $p[i]$ and b . First find the highest degree of $p[i]$ i.e., s . Initial degree of non-primitive polynomial i.e., bs . Initialize each element of $p'[i]$ of length bs to 0. Iterate loop from 2 to bs in order to initialize coefficient array to 1. Finally iterate each element of p' and modify the value if $p'[i]$ at position i to the value of p at i . Finally insert 1 at position 0 of $p'[i]$ when its first element is 0. The output of this module play an integral role for calculating non primitive BCH generating polynomial. It is denoted by g' in our algorithm. We are interested in rows of obtained matrix. Using the matrix we obtained, the code for non-primitive generating polynomial of length b . Finally these values are printed out and saved in file for further usage.

```

BEGIN
1 INPUT  $p[i], b$ 
2 Initialize len  $\leftarrow$  length of  $p[i]$ 
3 Initialize size_array  $\leftarrow$  len  $\times b$ 
4 Initialize  $p'[i]$  of length size_array i.e.  $p'[i] \leftarrow 0$ 
5 Initialize index  $\leftarrow 1$ 
6 Initialize len_coef_array  $\leftarrow 0$ 
7   FOR  $i$  taking values from 2 to len
8     IF  $p'[i]$  at  $i \neq 0$  THEN
9       Initialize coef_array  $\leftarrow b * (i - 1)$ 
10      increment index
11    ENDIF
12  ENDFOR
13  len_coef_array  $\leftarrow$  length of coef_array

```

```

14   FOR  $j$  taking values from 1 to len_coef_array
15        $g'(i)$  at position  $j$  of coef_array  $\leftarrow 1$ 
16   ENDFOR
17    $g'[i] \leftarrow 1$  to  $p'[i]$ 
18   PRINT  $g'[i]$ 
END

```

e. designed_distance (d):

Here design distance is calculated from $g'[i]$. The length of coset array cl is determined and then iterate index from 2 to cl , calculate next index ni by increment current index. If $ni \leq cl$ then next element of coset is calculated to the value of coset array at position of next index. Otherwise the bn is assigned to next coset. The whole process iterates to lencoset coset and finally stops at the last coset. Design distance d is calculated from the last value of coset array at position 1.

```

BEGIN
1   INPUT  $c[i]$  i.e. coset_array
2   Initialize lencoset  $\leftarrow$  length of coset_array
3   Initialize next_index  $\leftarrow 0$ 
4   Initialize next_coset  $\leftarrow 0$ 
5   FOR index taking values from 2 to lencoset
6       next_index  $\leftarrow$  increment index
7       IF next_index  $\leq$  lencoset THEN
8           next_coset  $\leftarrow$  coset_array at position next_index
9       ELSEIF
10          next_coset  $\leftarrow bn$ 
11      ENDIF
12  ENDFOR
13   $d \leftarrow$  next_coset at position 1.
END

```

f. error :

For given designed distance d , the error correction capability of a code t is calculated.

```

BEGIN
1   INPUT  $d$ .
2       error  $t \leftarrow (d - 1)/2$ 
END

```

4.3.2 Error correction in received polynomial (Decoding)

In decoding step for a received polynomial, we first calculate the syndrome matrix $S[i]$. Then the D_matrix is calculated that should be invertible. After this error locator polynomial is determined whose roots give the exact position of errors in the received polynomial. Finally the received polynomial is corrected.

To find the error vector and obtain the corrected codeword following scheme is used. Table 14 shows list of the following steps for error correction.

Table 14: Decoding modulation description

	Module	Input	Output
a	Syndrome_Matrix	$d, bn, bk, \alpha', r[i]$	$S'[i]$
b	Calculate D_matrix	$t, S'[i]$	D_matrix
c	Is D invertible	t, D_matrix	$ D \neq 0$
d	error_locator_poly	$t, D_matrix, S'[i]$	$f[i]$
e	error_position	$f[i]$	$e[i]$
f	error_values	$t, S[i], D_matrix, e[i], bn, pm[i]$	$ev[i]$
h	Correct_recieved polynomial	$t, bn, e[i], ev[i], r[i], \alpha'$	$v[i]$

a. Syndrome_matix ($S'[i]$):

Given design distance d , bn , message length bk and recieved polynomail $r[i]$, syndrome matrix $S'[i]$ can be calculated. The length of $S'[i]$ is initialize to $bn - bk$. Furthermore $S'[i]$ is initializes by $GF[i]$. Nested loop are used to calculate $S'[i]$. Upper loop is limited to the length of $bn + d - 2$, where $S'[i]$ equalizes to power of α' and in nested loop $S'[i]$ equalize to power

of $S'[i]$ and the iterator. Once the values of $S'[i]$ are filled with the above values of $S'[i]$, then these $S'[i]$ can be calculated as product of $r[i]$ and $(S'[i])^t$.

```

BEGIN
1   INPUT  $d, bn, bk, \alpha, r[i]$ 
2   Initialize lenSyndrome  $\leftarrow bn - bk$ 
3   Initialize  $S'[i] \leftarrow GF$  of len Syndrome
4   Initialize valueSynd  $\leftarrow 0$ 
5   Initialize valueSynd  $\leftarrow GF$  length  $bn$ 
6   Initialize loopLimit  $\leftarrow bn + d - 2$ 
7   FOR  $i \leftarrow bn$  to loop limit
8       valueSynd  $\leftarrow \alpha^i$ 
9       FOR  $j \leftarrow 1$  to  $bn$ 
10          evalSynd  $\leftarrow$  valueSynd powers  $j$ 
11      ENDFOR
12   $S'[i] \leftarrow$  received_poly * transpose of evalSynd
13  ENDFOR
14  PRINT  $S'[i]$ 
END

```

b. D_matrix ($D[i]$):

Given error t and $S'[i]$, D_matrix is calculated and then double loops operation on syndrome matrix is carried out and suitable values from syndrome matrix is scanned out. The process is as follows. First calculate $GF[i]$ of length t and D -matrix is initialized to that value. $D[i]$ in nested loop for loops. Both loops iterates from 1 to t . $D[i]$ values are $S'[i]$ values at position i of sum of loops iterators to 1.

```

BEGIN
1   INPUT  $t, S'[i]$ 
2   Calculate Galois field length of  $t$ 
3   Initialize  $D\_matrix \leftarrow$  Galois field calculated in previous step
4   FOR index  $i1$  taking from 1 to  $t$ 
5       For index  $i2$  taking from 1 to  $t$ 

```



```

6            $D\_matrix \leftarrow S'[i]$  at position  $i1 + i2 - 1$ 
7       ENDFOR
8   ENDFOR
END

```

c. Is D invertible:

This module check if $D[i]$ is invertible. If $D[i]$ is invertible, then it is fine otherwise error, t , is decremented and D_matrix is again calculated. These operations are carried out till error t becomes 0. If t becomes 0, then algorithm will exit.

```

BEGIN
1   INPUT  $t, D\_matrix$ 
2   IF  $D\_matrix$  is invertible THEN
3       continue algo // goto step 5.
4   ELSEIF
5       decrement  $t$ ;
6       IF  $t$  equals 0
7           goto STEP 3
8       ENDIF
9   ENDIF
10  // Panic Condition
11  IF  $t$  equals 0
12      Print ERROR cannot be corrected.
13  EXIT algo
14  ENDIF
END

```

d. error_locator_polynomial ($f[i]$):

Given input t, D_matrix and $S'[i]$, $f[i]$ is calculated. First initialize product matrix $pm[i]$ and $f[i]$ to $GF[i]$. Then iterate the loop from 1 to t , $pm[i]$ is filled with the value of $S'[i]$ at $t + i$. After the loops ends, the value of $(S'[i] * pm[i])^{-1}$ get equals to temporary matrix. Once the temporary matrix is acheived $f[i]$ is taken as the transpose of that temporary matrix.

```

BEGIN

```

```

1   INPUT  $t, D[i], S[i]$ 
2   create  $\alpha^t$ 
3   Initialize product_matrix  $pm[i] \leftarrow \alpha^t$ 
4   Initialize  $f[i] \leftarrow \alpha^t$ 
5   Initialize temporary_matrix  $T''[i]$  of size  $t \leftarrow 0$ 
6   FOR index  $i$  taking values from 1 to  $t$ 
7        $pm[i] \leftarrow S[i]$  at position  $t + i$ 
8   ENDFOR
9    $T''[i] \leftarrow (S[i])^{-1} * pm[i]$ 
10   $f[i] \leftarrow (T''[i])^t$ 
11   $f[t + 1] \leftarrow 1$  // coefficient of  $f[t + 1] = 1$ 
END

```

e. Error position matrix ($e[i]$):

Based on $f[i]$ we can determine error position. First the roots of $f[i]$ is calculated and then we take its inverse. The values we obtain are in matrix form and these manifest error position.

```

BEGIN
1   INPUT  $f[i]$ 
2   Initialize error_pos_matrix  $e[i] \leftarrow 0$ 
3   Initialize root_matrix  $\leftarrow 0$ 
4   root_matrix  $\leftarrow$  roots of  $f[i]$ 
5    $e[i] \leftarrow$  inverse of elements of root_matrix
6   PRINT  $e[i]$ 
END

```

f. Error values ($ev[i]$):

In the previous step we have calculated error position, so once error position is determined we can easily calculate their respective values. The nested for loops are used to determine error values. Both of the loops iterate from 1 to t , in the first loop values from D_matrix can be taken while in the next loop value of $pm[i]$ can be taken along with $S'[i]$. Finally $ev[i]$ get equated to $(D_matrix * pm[i])^{-1}$.

```

BEGIN

```

```

1  INPUT  $t, S[i], D\_matrix, e[i], bn, pm[i]$ 
2  Initialize error_value_matrix  $ev[i] \leftarrow 0$ 
3  FOR  $1 \leq i1 \leq t$ 
4      FOR  $1 \leq i2 \leq t$ 
5           $D\_matrix$  elements at  $i1$  and  $i2 \leftarrow e[i]$  at  $i2 * (i1 + bn - 1)$ 
6      ENDFOR
7       $pm[i]$  elements at position  $i1 \leftarrow S[i1]$ 
8  ENDFOR
9   $ev[i] \leftarrow (D\_matrix * pm[i])^{-1}$ 
10 PRINT  $ev[i]$ 
END

```

g. Correct_received_polynomial:

Once we have calculated error positions and error values the received polynomial can be corrected. Here input parameters are $e[i], ev[i], r[i]$ and bn . First estimated codeword denoted by est_code is calculated by various operations i.e., taking loops to t, bn , and power of Galois field. The received polynomial is corrected by subtraction of error polynomial and we get the corrected codeword $v[i]$.

```

BEGIN
1  INPUT  $t, bn, e[i], ev[i], r[i], \alpha'$ 
2  calculate  $GF[i]$  of length  $bn$ .
3  Initialize  $est\_error \leftarrow GF[i]$ 
4  Initialize  $est\_code \leftarrow GF[i]$ 
5  Initialize  $alpha\_val \leftarrow 0$ 
6  FOR  $1 \leq i \leq t$ 
7      FOR  $1 \leq j \leq bn$ 
8           $alpha\_val \leftarrow (\alpha')^{j-1}$ 
9          IF  $alpha\_val = \text{element } e[i] \text{ at } i$  THEN
10              $est\_error$  position  $j \leftarrow est\_error$  at position  $j + ev[i]$  at  $i$ 
11         ENDIF
12     ENDFOR

```

```

13         est_code  $\leftarrow$  r[i] + est_error
14         PRINT est_code
15         elements of pm[i] at i1  $\leftarrow$  S'[i] elements at i1
16     ENDFOR
17     ev[i]  $\leftarrow$  D_matrix * pm[i]
18     PRINT ev[i]
END

```

Example 67 For the code of length 45 simulation is carried out as follows: in this case $b = 3$ and $n = 15$. Using $n = 15$ and $k = 11$, Matlab's build in function **genpoly** is invoked in order to find primitive polynomial, i.e., $p(i) = x^4 + x + 1$, as explain in Table 1. With $b = 3$ and $p(i) = x^4 + x + 1$, **non_primitive_poly** function is invoked, as described in Table 1 step 4, here non primitive polynomial named as $p'(i)$ is obtained. Output for $p'(i)$ is $x^{12} + x^9 + 1$.

Cyclotomic cosets i.e., **coset_array** values are also calculated. First non_primitive_sequence in step 1 of Table 1, is invoked to find the power of alpha till $\alpha^{45} = 1$. With coset_array in hand, the designed distance d can be calculated, which is the first element of next coset_array. Last but not the least error t is calculated against the given designed distance d . Code rate R is also calculated against each k_1 and bn but is not mentioned in previous section.

The output are as follows:

Cyclotomic cosets for $(45, 33) = [1\ 2\ 4\ 8\ 16\ 32\ 19\ 38\ 31\ 17\ 34\ 23]$,

$t_1 = 1$ and $R_1 = (0.73333)$.

Cyclotomic cosets for $(45, 29) = [3\ 6\ 12\ 24]$,

$t_1 = 2$ and $R_1 = (0.64444)$.

Cyclotomic cosets for $(45, 23) = [5\ 10\ 20\ 40\ 35\ 25]$,

$t_1 = 3$ and $R_1 = (0.51111)$.

Cyclotomic cosets for $(45, 11) = [7\ 14\ 28\ 11\ 22\ 44\ 43\ 41\ 37\ 29\ 13\ 26]$,

$t_1 = 4$ and $R_1 = (0.24444)$.

Cyclotomic cosets for $(45, 5) = [15\ 30]$,

$t_1 = 10$ and $R_1 = (0.11111)$.

Now comes error correction in received polynomial. In this the code $(45, 29)$ is taken under

consideration, with designed distance $d_1 = 5$ and $t_1 = 2$. Let the received polynomial be

$$x^{44} + x^{16} + x^{13} + x^{12} + x^{11} + x^7 + x^3 + x + 1.$$

With the given values d_1 , k_1 , and received polynomial, syndrome_matrix is calculated. The output for syndromes are: $S_1 = \alpha^2$, $S_2 = \alpha^4$, $S_3 = \alpha^{30}$, $S_4 = \alpha^8$. Next we arrange syndrome values in linear equation form that is $Ax = B$. Where $A = [S_1, S_2; S_2, S_3]$ and $B = [S_3, S_4]$. Matrix A is named as t_matrix of $t \times t$ dimension. Then we find the whether the t_matrix is singular or not. If the determinant of t_matrix is non zero then error locator polynomial is calculated. Next error position is calculated from sigma_matrix which is obtained from the coefficients of error locator polynomial. For the given values, error_positions are 44 and 11. Hence the error polynomial is $x^{44} + x^{11}$. On subtracting error polynomial from received polynomial the following code polynomial $x^{16} + x^{13} + x^{12} + x^7 + x^3 + x + 1$ is obtained. All of the above equations are obtained by using Matlab symbolic toolbox.

With the help of the above discussed algorithm many examples on non-primitive BCH codes of length bn , b^2n , b^3n are constructed corresponding to primitive BCH code of length n . The parameters for all binary non-primitive BCH codes of length bn , b^2n , b^3n where $n \leq 2^6 - 1$ and b is either 3 or 7 are given in Table 15.

Table 15: BCH codes of length $n \leq 2^6 - 1$

n	bn	k₁	t₁	R₁	b²n	k₂	t₂	R₂	b³n	k₃	t₃	R₃
7	49	28	1	0.571	343	196	1	0.571	2401	1372	1	0.571
		7	3	0.142		49	3	0.142		343	3	0.142
		4	10	0.081		28	10	0.081		196	10	0.081
		1	24	0.020		7	24	0.020		49	24	0.020
						4	73	0.011		28	73	0.011
						1	171	0.002		7	171	0.002
										4	541	0.001
										1	1200	0.000

n	bn	k₁	t₁	R₁	b²n	k₂	t₂	R₂	b³n	k₃	t₃	R₃
15	45	33	1	0.733	135	99	1	0.733	405	297	1	0.733
		29	2	0.644		87	2	0.644		261	2	0.644
		23	3	0.511		69	3	0.511		207	3	0.511
		11	4	0.244		33	4	0.244		99	4	0.244
		7	7	0.155		29	7	0.214		87	7	0.214
		5	10	0.111		23	10	0.170		69	10	0.170
		1	22	0.022		11	13	0.081		33	13	0.081
						7	22	0.051		29	22	0.071
						5	31	0.037		23	31	0.056
						1	67	0.007		11	40	0.027
										7	67	0.017
										5	94	0.012
										1	202	0.002
63	189	171	1	0.904	567	513	1	0.904	1701	1539	1	0.904
		165	2	0.873		495	2	0.873		1485	2	0.873
		147	3	0.777		441	3	0.777		1323	3	0.777
		129	4	0.682		387	4	0.682		1161	4	0.682
		123	5	0.650		381	5	0.672		1143	5	0.671
		105	6	0.555		327	6	0.576		981	6	0.576
		87	7	0.460		273	7	0.481		819	7	0.481
		81	10	0.428		255	10	0.449		765	10	0.449

n	bn	k₁	t₁	R₁	b²n	k₂	t₂	R₂	b³n	k₃	t₃	R₃
63	189	75	11	0.3968	567	237	11	0.4179	1701	711	11	0.417
		57	13	0.3015		183	13	0.3227		549	13	0.322
		54	15	0.2857		177	15	0.3121		543	15	0.319
		36	16	0.1904		123	16	0.2169		381	16	0.224
		30	19	0.1587		105	19	0.1851		327	19	0.192
		24	22	0.1269		87	22	0.1534		273	22	0.160
		18	31	0.0952		81	31	0.1428		255	31	0.149
		16	34	0.0846		75	34	0.1322		237	34	0.1393
		10	40	0.0529		57	40	0.1005		183	40	0.1075
		7	46	0.0370		54	46	0.0952		177	46	0.1040
		1	94	0.0053		36	49	0.0634		123	49	0.0723
						30	58	0.0529		105	58	0.0617
						24	67	0.0423		87	67	0.0511
						18	94	0.0317		81	94	0.0476
						16	103	0.0282		75	103	0.0440
						10	121	0.0176		57	121	0.0335
						7	139	0.0123		54	139	0.0317
						1	283	0.0017		36	148	0.0211
										30	175	0.0176
										10	364	0.0058
										1	850	0.0005

Table 16 manifests the error and code rate values against some selected codes which we have obtained after simulating our algorithm. These codes are of length bn , b^2n and b^3n , where $n \leq 2^6 - 1$, and $b = 3, 7$. k_1, k_2 and k_3 are dimensions of the codes $\mathcal{C}_{bn}^1, \mathcal{C}_{b^2n}^2$ and $\mathcal{C}_{b^3n}^3$ respectively.

Interleaved Codes

From Table 16, it is observed that corresponding to a primitive (n, k) code there are (bn, bk) , (b^2n, b^2k) , (b^3n, b^3k) codes with same error correction capability and code rate. These

codes are found to be *interleaved codes* (*Interleaving* is a periodic and reversible reordering of codes of L transmitted bits) of depth b , b^2 and b^3 . Hence along with random error correction capability these codes can correct burst of error of length b , b^2 and b^3 respectively. The term burst of error means that two or more bits in the received word has changed from 1 to 0 or from 0 to 1. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Similarly for the code (bn, bk) the codes (b^2n, b^2k) and (b^3n, b^3k) are interleaved codes of depth b^2 and b^3 respectively.

The code $(49, 28)$ is interleaved code of depth 7, which is formed by interleaving the following 7 codewords from $(7, 4)$ code that is,

$$\begin{aligned} & (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) , (1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0) , (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) , \\ & (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) , (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) , (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) , \\ & (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) , \text{on writing them column by column it gives} \end{aligned}$$

$$v^1 = (0100000010000000000000100000000000000000000000000000000) \in \mathcal{C}_{49}^2.$$

In a similar way, codeword of $(343, 196)$ and $(2401, 1372)$ are obtained by writing column by column 7 codeword of \mathcal{C}_{49}^2 and \mathcal{C}_{343}^3 . Therefore for decoding a received polynomial in \mathcal{C}_{343}^3 one can easily reverse the process and correct errors in the codeword of either \mathcal{C}_{49}^2 or \mathcal{C}_7^1 .

We conclude this chapter as follows:

1) Existence of a sequence of binary non-primitive BCH codes of lengths $b^m n$, where m is a positive integer, is ensured against a given n length primitive binary BCH code.

2) The sequence of embeddings of the binary BCH codes is obtained and the binary BCH code of length $b^j n$ has higher code rate and error correction capability than binary BCH code of length $b^{j-1} n$.

3) The data configured through length $b^{j-1} n$ can be transmitted and decoded under binary BCH code of length $b^j n$.

4) A method is devised by which we can improve the data/information transfer and receiving with better trade off.study.

5) An algorithm for the construction of non-primitive BCH codes $\mathcal{C}_{b^j n}^j$ of length $b^j n$, where

j is a positive integer and n is length of primitive binary BCH code C_n , is given.

6) Corresponding to a primitive (n, k) binary BCH code C_n there are $(b^j n, b^j k)$ codes with same error correction capability and code rate. These codes are found to be interleaved codes of depth b^j . Therefore along with random error correction capability these codes can also correct error burst of length b^j .

This work is further extended over the Galois field \mathbb{F}_4 .

Chapter 5

Construction of non-primitive BCH codes over the field F_4

In this chapter, construction of BCH codes over the field \mathbb{F}_4 is introduced. For this initially a primitive BCH code \mathcal{C}_n of length n an ideal in the monoid ring $\mathbb{F}_4[x; a\mathbb{N}_0]_n$ is constructed and based on it existence of a non-primitive BCH code \mathcal{C}_{bn} of length bn is investigated, such that the code \mathcal{C}_n is embedded in \mathcal{C}_{bn} . Furthermore we have compared the efficiency of BCH codes constructed over fields \mathbb{F}_2 and \mathbb{F}_4 .

5.1 BCH-codes as Ideal in the ring $F_4[x; a\mathbb{N}_0]_n$

Let $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2 = 1 + \alpha\}$ be the finite field. The construction of BCH codes in monoid ring $\mathbb{F}_4[x; a\mathbb{N}_0]_n$ is similar to the construction of BCH codes in $\mathbb{F}_4[x]_n$. For this, let $c, d, q = 4$ and n be the positive integers such that $4 \leq d \leq n$ with $\gcd(n, 4) = 1$ and $n = (4)^s - 1$, where s is the degree of a primitive irreducible polynomial in $\mathbb{F}_4[x; a\mathbb{N}_0]$. Consequently, the n length binary BCH code \mathcal{C}_n has generator polynomial of degree r given by $g(x^a) = \text{lcm}\{m_i(x^a) : i = c, c+1, \dots, c+d-2\}$, where $m_i(x^a)$ are minimal polynomials of γ^i for $i = c, c+1, \dots, c+d-2$. Where γ is the primitive n th root of unity in \mathbb{F}_{4^s} , an s degree Galois field extension of \mathbb{F}_4 . Since $m_i(x^a)$ divides $(x^a)^n - 1$, it follows that $g(x^a)$ divides $(x^a)^n - 1$. This implies $\mathcal{C}_n = (g(x^a))$ is a principal ideal in the factor ring $\mathbb{F}_4[x; a\mathbb{N}_0]_n$.

In the following example, primitive BCH codes in $\mathbb{F}_4[x; a\mathbb{N}_0]_{15}$ are considered and a variation

in error correction capability and code rates have been noticed.

Example 68 Let $p(x^2) = (x^2)^2 + (x^2) + \alpha \in \mathbb{F}_4[x; 2\mathbb{Z}_0]$ be the primitive polynomial. Then $n = 4^2 - 1 = 15$. Let $\xi \in \mathbb{F}_{4^2}$, satisfy the relation $\xi^2 = \xi + \alpha$. Using this relation we get $\xi^{15} = 1$. Hence, ξ is the primitive 15th root of unity and $p(x^2)$ is the minimal polynomial of ξ . Since $g(x^2) = \text{lcm}\{m_i(x^2), i = c, c+1, \dots, c+d-2\}$, therefore we first calculate $m_i(x^2)$. By [27, Theorem 4.4.2], ξ, ξ^4 , have same minimal polynomial $m_1(x^2) = p(x^2)$. Let $m_2(x^2)$ be the minimal polynomial for ξ^2 and ξ^8 . Using above relations we get $m_2(x^2) = (x^2)^2 + (x^2) + \alpha^2$. Similarly we get

$$\begin{aligned} m_3(x^2) &= (x^2)^2 + \alpha^2(x^2) + 1, \quad m_5(x^2) = (x^2) + \alpha m_6(x^2) = (x^2)^2 + \alpha(x^2) + 1, \\ m_7(x^2) &= (x^2)^2 + \alpha(x^2) + \alpha, \quad m_{10}(x^2) = (x^2)^2 + \alpha^2, \quad m_{11}(x^2) = (x^2)^2 + \alpha^2(x^2) + \alpha^2. \end{aligned}$$

The BCH code with designed distance $d = 2$ has generator polynomial $g(x^2) = m_1(x^2) = (x^2)^2 + (x^2) + \alpha$. On writing its coefficients in ascending order with respect to power of (x^2) we get $\alpha 11$. The following table discuss BCH codes for different designed distances, coefficients of generator polynomials, error correction capability and code rate.

Table 16: BCH codes of length 15

(n, k)	d	$\text{coeff}(g(x^2))$	t	R
(15, 11)	3	11001	1	0.73
(15, 9)	5	1 α α 11 α^2 1	2	0.6
(15, 6)	7	α 10 α 1 α α^2 α^2 1	3	0.4
(15, 3)	11	α 01 α^2 α^2 0 α^2 1 α α α^2 α 1	5	0.2
(15, 1)	15	111111111111111	7	0.06.

Where d, t and R denote the designed distance, error correction capability and code rate of the code of length 15 over the field \mathbb{F}_4 respectively, and $\text{coeff}(g(x^2))$ denotes the coefficient of the generating polynomial $g(x^2)$.

5.2 BCH-codes as Ideal in the ring $F_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}$

For the construction of a non-primitive BCH code in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}$, we choose only such a s degree primitive irreducible polynomial $p(x^a) \in \mathbb{F}_4[x; a\mathbb{N}_0]$ for which there is a bs degree irreducible polynomial $p(x^{\frac{a}{b}})$ in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$, whereas it is not true in general. For illustration see the following list of few irreducible polynomials of degree bs in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$ corresponding to primitive irreducible polynomial of degree s in $\mathbb{F}_4[x; a\mathbb{N}_0]$. For $p(x^a) \in \mathbb{F}_4[x; a\mathbb{N}_0]$, $p(x^{\frac{a}{3}}) \in \mathbb{F}_2[x; \frac{a}{3}\mathbb{N}_0]$, $p(x^{\frac{a}{5}}) \in \mathbb{F}_4[x; \frac{a}{5}\mathbb{N}_0]$, $p(x^{\frac{a}{7}}) \in \mathbb{F}_4[x; \frac{a}{7}\mathbb{N}_0]$. For the sake of convenience replace $x^a, x^{\frac{a}{3}}, x^{\frac{a}{5}}, x^{\frac{a}{7}}$ by x, y, z, w respectively.

Table 17: Non primitive irreducible polynomials against primitive irreducible polynomials

deg	$p(x)$	b	$p(y), p(z), p(w)$
2	$x^2 + x + \alpha$	3, 5	$y^6 + y^3 + \alpha, z^{10} + z^5 + \alpha$
3	$x^3 + x^2 + x + \alpha$	3, 7	$y^9 + y^6 + y^3 + \alpha, w^{21} + w^{14} + w^7 + \alpha$
4	$x^4 + x^3 + x + \alpha$	3, 5	$y^{12} + y^9 + y^3 + \alpha, z^{20} + z^{15} + z^5 + \alpha$
5	$x^5 + x + \alpha$	3	$y^{15} + y^3 + \alpha$
6	$x^6 + x^2 + x + \alpha$	3, 5,	$y^{18} + y^6 + y^3 + \alpha, z^{30} + z^{10} + z^5 + \alpha,$
		7	$w^{42} + w^{14} + w^7 + \alpha$
7	$x^7 + \alpha^2 x^5 + \alpha^2 x + \alpha^2$	3	$y^{21} + \alpha^2 y^{15} + \alpha^2 y^3 + \alpha^2$
8	$x^8 + x^6 + x^4 + x^2 + x + \alpha^2$	3, 5	$y^{24} + y^{18} + y^{12} + y^6 + y^3 + \alpha^2,$
			$z^{40} + z^{30} + z^{20} + z^{10} + z^5 + \alpha^2$
9	$x^9 + \alpha x^5 + x^4 + \alpha^2$	3, 7	$y^{27} + \alpha y^{15} + y^{12} + \alpha^2, w^{27} + \alpha w^{15} + w^{12} + \alpha^2$
10	$x^{10} + \alpha^2 x^5 + \alpha^2 x + a$	3, 5	$y^{30} + \alpha^2 y^{15} + \alpha^2 y^3 + a, z^{50} + \alpha^2 z^{25} + \alpha^2 z^5 + a$
\vdots	\vdots	\vdots	\vdots

One can easily verify Table 18 with the help of GAP4. Table 18 shape the following proposition.

Proposition 69 *Let $p(x^a) \in \mathbb{F}_4[x; a\mathbb{N}_0]$ be a primitive irreducible polynomial of degree $s \in \{2l, 3l, 5l, 6l\}$, where $l \in \mathbb{Z}^+$. Then the corresponding bs degree generalized polynomial $p(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$ is non-primitive irreducible polynomial for $b \in \{(3, 5), (3, 7), \{3\}, (3, 5, 7)\}$ respectively.*

As a consequence of Table 17, the primitive and non-primitive BCH codes of lengths n and bn respectively are obtained.

Table 18: BCH codes of lengths n and bn

s	n	bn
2	3	9, 15
3	7	21, 49
4	15	45, 75
6	63	189, 315, 441
7	128	384
8	255	765, 1275
9	511	1533, 3577
10	1023	3069, 5115
12	4094	12282, 20470, 28658

Theorem 70 *Let $n = 4^s - 1$ be the length of primitive BCH code \mathcal{C}_n , where $p(x^a) \in \mathbb{F}_4[x; a\mathbb{N}_0]$ is a primitive irreducible polynomial of degree s such that $p(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$ is a non-primitive irreducible polynomial of degree bs . Then*

1) *for positive integers c^1, d_1, bn such that $2 \leq d_1 \leq bn$ and bn is relatively prime to 4, there exists a non-primitive BCH code \mathcal{C}_{bn} of length bn , where bn is order of an element $\alpha \in \mathbb{F}_{4^{bs}}$.*

2) *the non-primitive BCH code \mathcal{C}_{bn} is defined as*

$$\mathcal{C}_{bn} = \{v(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]_{bn} : v(\alpha^i) = 0 \text{ for all } i = c^1, c^1 + 1, \dots, c^1 + d_1 - 2.\}$$

Equivalently, \mathcal{C}_{bn} is the null space of the matrix

$$H = \begin{bmatrix} 1 & \alpha^{c^1} & \alpha^{2c^1} & \dots & \alpha^{(bn-1)c^1} \\ 1 & \alpha^{c^1+1} & \alpha^{2(c^1+1)} & \dots & \alpha^{(bn-1)(c^1+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{c^1+d_1-2} & \alpha^{2(c^1+d_1-2)} & \dots & \alpha^{(bn-1)(c^1+d_1-2)} \end{bmatrix}.$$

Proof. 1) Since the polynomial $p(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$ is irreducible but not primitive, it follows that the code constructed through it is also non primitive. However, there is an element $\alpha \in \mathbb{F}_{4^{bs}}$ of order bn vanishes $p(x^{\frac{a}{b}})$. Therefore, bn divides $4^{bs} - 1$. Now, let $m_i(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$ denote the minimal polynomial of α^i and $g(x^{\frac{a}{b}})$ be the *lcm* of distinct polynomials among $m_i(x^{\frac{a}{b}})$, $i = c^1, c^1 + 1, \dots, c^1 + d_1 - 2$; that is,

$$g(x^{\frac{a}{b}}) = \text{lcm}\{m_i(x^{\frac{a}{b}}) : i = c^1, c^1 + 1, \dots, c^1 + d_1 - 2\}.$$

As $m_i(x^{\frac{a}{b}})$ divides $(x^{\frac{a}{b}})^{bn} - 1$ for each i , therefore $g(x^{\frac{a}{b}})$ also divides $(x^{\frac{a}{b}})^{bn} - 1$. This implies that \mathcal{C}_{bn} is a principal ideal generated by $g(x^{\frac{a}{b}})$ in the factor ring $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. Hence \mathcal{C}_{bn} is a non-primitive BCH code of length bn over \mathbb{F}_4 with designed distance d_1 .

2) Let $v(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$. Then $v(x^{\frac{a}{b}}) = g(x^{\frac{a}{b}})q(x^{\frac{a}{b}})$ for some $q(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$, where $g(x^{\frac{a}{b}})$ is the generator polynomial of \mathcal{C}_{bn} . Hence $v(\alpha^i) = 0$ for all $i = c^1, c^1 + 1, \dots, c^1 + d_1 - 2$. Conversely, take $v(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ such that $v(\alpha^i) = 0$ for all $i = c^1, c^1 + 1, \dots, c^1 + d_1 - 2$. This implies $m_i(x^{\frac{a}{b}})$ divides $v(x^{\frac{a}{b}})$ for all $i = c^1, c^1 + 1, \dots, c^1 + d_1 - 2$ and therefore $g(x^{\frac{a}{b}})$ divides $v(x^{\frac{a}{b}})$. Thus $v(x^{\frac{a}{b}}) \in \mathcal{C}_{bn}$. For the second part, we let

$$v(x^{\frac{a}{b}}) = v_0 + v_1(x^{\frac{a}{b}}) + \dots + v_{bn-1}(x^{\frac{a}{b}})^{bn-1} \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}.$$

Then $v(\alpha^i) = 0$ for all $i = c^1, c^1 + 1, \dots, c^1 + d_1 - 2$ if and only if $Hv^T = 0$, where $v = (v_0, v_1, \dots, v_{bn-1}) \in \mathbb{F}_4^{bn}$. This proves that \mathcal{C}_{bn} is the null space of the matrix H . ■

The following example illustrates the construction of a non primitive BCH code of length $3n$ in $\mathbb{F}_4[x; \frac{2}{3}\mathbb{N}_0]_{3n}$.

Example 71 Corresponding to a primitive polynomial $p(x^2) = (x^2)^2 + (x^2) + \alpha$ in $\mathbb{F}_4[x; 2\mathbb{N}_0]$

there is a non-primitive irreducible polynomial $p(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^3 + \alpha$ in $\mathbb{F}_4[x; \frac{2}{3}\mathbb{N}_0]$ (see Table 18). Let $\beta \in GF(4^6)$, satisfy the relation $\beta^6 + \beta^3 + 1 = 0$. Using this relation we obtain the following list of elements.

Table 19: Distinct powers of β in $GF(4^6)$

$\beta^6 = \beta^3 + \alpha$	$\beta^{20} = \alpha\beta^5$	$\beta^{34} = \beta^4 + \alpha\beta^4$
$\beta^7 = \beta^4 + \alpha\beta$	$\beta^{21} = 1 + \alpha + \alpha\beta^3$	$\beta^{35} = \beta^5 + \alpha\beta^5$
$\beta^8 = \beta^5 + \alpha\beta^2$	$\beta^{22} = \beta + \alpha\beta + \alpha\beta^4$	$\beta^{36} = 1 + \beta^3 + \alpha\beta^3$
$\beta^9 = \alpha + \beta^3 + \alpha\beta^3$	$\beta^{23} = \beta^2 + \alpha\beta^2 + \alpha\beta^5$	$\beta^{37} = \beta + \beta^4 + \alpha\beta^4$
$\beta^{10} = \beta^4 + \alpha\beta + \alpha\beta^4$	$\beta^{24} = 1 + \alpha + \beta^3$	$\beta^{38} = \beta^2 + \beta^5 + \alpha\beta^5$
$\beta^{11} = \beta^5 + \alpha\beta^2 + \alpha\beta^5$	$\beta^{25} = \beta + \alpha\beta + \beta^4$	$\beta^{39} = 1 + \alpha\beta^3$
$\beta^{12} = 1 + \beta^3$	$\beta^{26} = \beta^2 + \alpha\beta^2 + \beta^5$	$\beta^{40} = \beta + \alpha\beta^4$
$\beta^{13} = \beta + \beta^4$	$\beta^{27} = \alpha + \alpha\beta^3$	$\beta^{41} = \beta^2 + \alpha\beta^5$
$\beta^{14} = \beta^2 + \beta^5$	$\beta^{28} = \alpha\beta + \alpha\beta^4$	$\beta^{42} = 1 + \alpha + \beta^3 + \alpha\beta^3$
$\beta^{15} = \alpha$	$\beta^{29} = \alpha\beta^2 + \alpha\beta^5$	$\beta^{43} = \beta + \alpha\beta + \beta^4 + \alpha\beta^4$
$\beta^{16} = \alpha\beta$	$\beta^{30} = 1 + \alpha$	$\beta^{44} = \beta^2 + \alpha\beta^2 + \beta^5 + \alpha\beta^5$
$\beta^{17} = \alpha\beta^2$	$\beta^{31} = \beta + \alpha\beta$	$\beta^{45} = 1$
$\beta^{18} = \alpha\beta^3$	$\beta^{32} = \beta^2 + \alpha\beta^2$	
$\beta^{19} = \alpha\beta^4$	$\beta^{33} = \beta^3 + \alpha\beta^3$	

Thus $bn = 3 \times 15 = 45$. Now, to calculate generating polynomial $g(x^{\frac{2}{3}})$, we first calculate the minimal polynomials. By [27, Theorem 4.4.2], $\beta, \beta^4, \beta^{16}, \beta^{19}, \beta^{31}, \beta^{34}$ have same minimal polynomial $m_1(x^{\frac{2}{3}}) = p(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^3 + \alpha$. Let $m_2(x^{\frac{2}{3}})$ be the minimal polynomial for β^2 , then $\beta^2, \beta^8, \beta^{32}, \beta^{38}, \beta^{17}, \beta^{23}$ all are roots for $m_2(x^{\frac{2}{3}})$. Therefore by using Table II we get $m_2(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^3 + \alpha^2$. Similarly, we obtain

$$\begin{aligned}
m_3(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + \alpha, \quad m_5(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^3 + \alpha, \quad m_6(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + \alpha^2, \\
m_7(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^6 + \alpha(x^{\frac{2}{3}})^3 + \alpha, \quad m_9(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^2 + \alpha^2(x^{\frac{2}{3}}) + 1, \quad m_{10}(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^3 + \alpha^2, \\
m_{11}(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^6 + \alpha^2(x^{\frac{2}{3}})^3 + \alpha^2, \quad m_{15}(x^{\frac{2}{3}}) = (x^{\frac{2}{3}}) + \alpha, \quad m_{18}(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^2 + \alpha(x^{\frac{2}{3}}) + 1, \\
m_{21}(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^2 + \alpha(x^{\frac{2}{3}}) + \alpha, \quad m_{30}(x^{\frac{2}{3}}) = (x^{\frac{2}{3}}) + \alpha^2, \quad m_{33}(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^2 + \alpha^2(x^{\frac{2}{3}}) + \alpha^2.
\end{aligned}$$

The BCH code with designed distance $d_1 = 2$ has generator polynomial $g(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^3 + \alpha$. On writing its coefficients in ascending order with respect to power of $(x^{\frac{2}{3}})$ we get $\alpha 001001$. Following is the table of BCH codes for different designed distances, coefficients of generator polynomials, error correction capability and code rate.

Table 20: BCH codes of length 45 over the field \mathbb{F}_4

(bn, k_1)	d_1	$\text{coef}g(x^{\frac{2}{3}})$	t_1	R_1
(45, 33)	3	1001000000001	1	0.73
(45, 31)	5	$\alpha 11\alpha 11000000\alpha 11$	2	0.68
(45, 28)	6	$\alpha^2\alpha 1\alpha^2\alpha^2\alpha 11000\alpha^2\alpha\alpha 11$	3	0.62
(45, 26)	7	$\alpha\alpha 0\alpha^2 101\alpha 0010\alpha\alpha 01\alpha^2 001$	3	0.57
(45, 20)	9	$\alpha^2\alpha^2 0\alpha 101\alpha^2 0100\alpha\alpha^2 01\alpha^2 001011001$	4	0.44
(45, 18)	10	$\alpha^2 1110100\alpha^2\alpha\alpha^2 1\alpha\alpha 0\alpha 0\alpha^2\alpha^2 1\alpha^2 0\alpha\alpha 11\alpha^2 1$	5	0.4
(45, 15)	11	$\alpha\alpha^2\alpha^2 01\alpha 10\alpha^2 1\alpha 0\alpha^2\alpha 1\alpha^2\alpha\alpha 0\alpha^2 1\alpha^2 0\alpha\alpha^2 10\alpha 1\alpha^2 1$	5	0.33
(45, 9)	15	$1\alpha\alpha 11\alpha^2 1000000001\alpha\alpha 11\alpha^2 1000000001\alpha\alpha 11\alpha^2 1$	7	0.2
(45, 8)	18	$\alpha\alpha 10\alpha^2 0110000000\alpha\alpha 10\alpha^2 0110000000\alpha\alpha 10\alpha^2 011$	9	0.17
(45, 6)	21	$\alpha 10\alpha 1\alpha\alpha^2\alpha^2 100000\alpha 10\alpha 1\alpha\alpha^2\alpha^2 100000\alpha 10\alpha 1\alpha\alpha^2\alpha^2 1$	10	0.13
(45, 4)	22	$\alpha^2 101\alpha^2 1\alpha^2\alpha^2\alpha 011000\alpha^2 101\alpha^2 1\alpha^2\alpha^2\alpha 011000\alpha^2 101\alpha^2 1\alpha^2\alpha^2\alpha 011$	11	0.08

Where d_1, t_1 , and R_1 denote the designed distance, error correction capability and the code rate of the code of length 45 over the field \mathbb{F}_4 respectively, k_1 denotes the dimension of the BCH code, and $\text{coeff}(g(x^{\frac{2}{3}}))$ denote the coefficients of the generating polynomial $g(x^{\frac{2}{3}})$.

Remark 72 These are the following two observations obtained from different examples:

1) the non-primitive BCH codes (bn, bk) are interleaved codes (Interleaving is a periodic and reversible reordering of codes of l transmitted bits) of degree b with same code rate and error correction capability as that of primitive BCH code (n, k) . They are capable of correcting burst of length b or less. By burst of error we means that two or more bits in the received word has changed from 1 to 0, α, α^2 or from α to 0, 1, α^2 and so on. For example the non-primitive BCH code (45, 33) is an interleaved code of degree 3 and is capable of correcting burst of length 3. For instance let $r = 10010000000010000000000001\alpha\alpha^2 0000000000000000$ be

2) the primitive BCH code repeats b times in non-primitive BCH codes whenever both have same code dimension that is $k = k_1$. For example on writing the corresponding code vectors of the generating polynomials of $(15, 9)$ and $(45, 9)$ codes that is:

we have $v = (1\alpha\alpha11\alpha^21000000000)$ and $v^1 = (1\alpha\alpha11\alpha^21000000001\alpha\alpha11\alpha^21000000001\alpha\alpha11\alpha^2100000000)$. Therefore the corresponding generating matrix G_1 of $g(x^{\frac{2}{3}})$ contains the generating matrix G of $g(x^2)$ such that $G_1 = \oplus_1^3 G$.

106

Table 21: Comparison of BCH codes of length 15

BCH code through $\mathbb{F}_2[x; a\mathbb{N}_0]$				BCH code through $\mathbb{F}_4[x; a\mathbb{N}_0]$			
d	$d_{(\min)}$	t	R	d	$d_{(\min)_1}$	t_1	R_1
2	3	1	0.733	2	3	1	0.866
4	5	2	0.466	3	3	1	0.733
6	7	3	0.333	4	7	3	0.6
8	15	7	0.066	6	6	2	0.533
				7	8	3	0.4
				8	10	4	0.266
				11	11	5	0.2
				12	15	7	0.066

Table 22: Comparison of BCH codes of length 45

BCH code through $\mathbb{F}_2[x; \frac{a}{b}\mathbb{N}_0]$				BCH code through $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$			
d'	$d_{(\min)_1}$	t_1	R_1	d'	$d_{(\min)_2}$	t_2	R_2
2	3	1	0.733	2	3	1	0.866
4	7	3	0.644	3	3	1	0.733
6	9	4	0.511	4	9	4	0.688
8	9	4	0.244	6	15	7	0.622
10	15	7	0.155	7	12	5	0.577
16	21	10	0.11	8	15	7	0.44
22	45	22	0.022	10	22	10	0.4
				11	25	12	0.33
				12	21	10	0.2
				16	18	8	0.177
				19	24	11	0.133
				22	30	14	0.088
				31	33	16	0.066
				34	45	22	0.022

From above tables we notice that: The possible choices of codes with different design distances are more over the field \mathbb{F}_4 as compared with the codes over the binary field. Secondly, it is clear that the code rate is better over the field \mathbb{F}_4 as for example in \mathcal{C}_{15} primitive BCH code with design distance 5 and 7 has code rate 0.466 and 0.333 over the binary field whereas over \mathbb{F}_4 we are getting 0.6 and 0.4. In non-primitive BCH codes \mathcal{C}_{45} with design distances 5, 7, 9, 15 and 21 we get code rates 0.644, 0.511, 0.244, 0.155, 0.11 over \mathbb{F}_2 and 0.688, 0.577, 0.44, 0.2, 0.133 over \mathbb{F}_4 respectively.

5.3 Primitive BCH code C_n and non-primitive BCH code C_{bn} :

A link

Now we formulate a link between a primitive BCH code C_n , and a non-primitive BCH code C_{bn} over the field \mathbb{F}_4 , where r and r' are the degrees of their generating polynomials $g(x^a)$ and $g(x^{\frac{a}{b}})$ respectively. From Theorem 70(1), it follows that the generator polynomial $g(x^{\frac{a}{b}}) \in \mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$ divides $(x^{\frac{a}{b}})^{bn} - 1$ in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$. So, there is a non-primitive BCH code C_{bn} generated by $g(x^{\frac{a}{b}})$ in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}$. Since, bn divides $n = 4^{bs} - 1$, so $(x^{\frac{a}{b}})^{bn} - 1$ divides $(x^{\frac{a}{b}})^n - 1$ in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$. It follows that $((x^{\frac{a}{b}})^n - 1) \subset ((x^{\frac{a}{b}})^{bn} - 1)$. Consequently, by the third isomorphism theorem for rings we have

$$\frac{\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]/((x^{\frac{a}{b}})^n - 1)}{((x^{\frac{a}{b}})^{bn} - 1)/((x^{\frac{a}{b}})^n - 1)} \simeq \frac{\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]}{((x^{\frac{a}{b}})^{bn} - 1)} \simeq \frac{\mathbb{F}_4[x; a\mathbb{N}_0]}{((x^a)^n - 1)}.$$

This gives the embedding $C_n \hookrightarrow C_{bn} \hookrightarrow C_{n'}$ of codes, where C_n, C_{bn} and $C_{n'}$ are respectively primitive BCH, non-primitive BCH and primitive BCH codes. The embedding $C_n \hookrightarrow C_{bn}$ is defined as: $a(x^a) = a_0 + a_1(x^a) + \dots + a_{n-1}(x^a)^{n-1} \mapsto a_0 + a_1(x^{\frac{a}{b}})^b + \dots + a_{n-1}(x^{\frac{a}{b}})^{b(n-1)} = a(x^{\frac{a}{b}})$, where $a(x^a) \in C_n$ and $a(x^{\frac{a}{b}}) \in C_{bn}$.

The above discussion formulates the following theorem.

Theorem 73 *Let C_n be a primitive BCH code of length $n = 4^s - 1$ generated by $g(x^a)$ in $\mathbb{F}_4[x; a\mathbb{N}_0]$ of degree r . Then*

- 1) *there exists a non-primitive BCH code C_{bn} of length bn generated by $g(x^{\frac{a}{b}})$ in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]$ of degree br ,*
- 2) *the BCH code C_n is embedded in the BCH code C_{bn} .*

Remark 74 $g(x^a)$ can be deduced from $g(x^{\frac{a}{b}})$ by substituting $x^{\frac{a}{b}} = y$ and then replacing y by $y^b = x^a$.

Example 75 *The following example is deduced by Example 68 and 71.*

The BCH codes having bits from the Galois field \mathbb{F}_4 with designed distances d , $d_1 \geq 4$ have

generator polynomials $g(x^2)$ and $g(x^{\frac{2}{3}})$, on letting $(x^{\frac{2}{3}}) = y$, that is $x^2 = y^3$, we get

$$\begin{aligned}
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{17} + (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{15}\alpha + (x^{\frac{2}{3}})^{14}\alpha + (x^{\frac{2}{3}})^{13}\alpha + \alpha^2(x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^8 + \\
&\quad (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^6\alpha + \alpha^2(x^{\frac{2}{3}})^5 + \alpha^2(x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^2\alpha + (x^{\frac{2}{3}})\alpha + \alpha^2 \\
g(y^3) &= (y^3)^{17} + (y^3)^{16} + (y^3)^{15}\alpha + (y^3)^{14}\alpha + (y^3)^{13}\alpha + \alpha^2(y^3)^{12} + (y^3)^8 + \\
&\quad (y^3)^7 + (y^3)^6\alpha + \alpha^2(y^3)^5 + \alpha^2(y^3)^4 + (y^3)^3 + (y^3)^2\alpha + (y^3)\alpha + \alpha^2 \\
g(x^2) &= (x^2)^{14}\alpha + (x^2)^{13}\alpha + (x^2)^{12}\alpha^2 + (x^2)^8 + (x^2)^7 + (x^2)^6\alpha + (x^2)^5\alpha^2 + \\
&\quad (x^2)^4\alpha^2 + (x^2)^3 + (x^2)^2\alpha^2 + (x^2)\alpha^2 + 1.
\end{aligned}$$

Where $g(x^2) \in \mathbb{F}_4[x; 2\mathbb{Z}_0]$ and is divisible by $(x^2)^2 + (x^2) + \alpha$, the generator polynomial of the BCH code $(15, 13)$ with designed distance $d = 2$. Table 23 shows that for a code in $\mathbb{F}_4[x; \frac{2}{3}\mathbb{N}_0]_{45}$ with designed distance d_1 we have a code in $\mathbb{F}_4[x; 2\mathbb{N}_0]_{15}$ with designed distance d embedded in it.

Table 23: Embedding of C_{15} in C_{45}

d'	(bn, k_2)	t_1	d	R_1	(bn, k_1)	t	R
3	(45, 33)	1	3	0.73	(15, 11)	1	0.733
11	(45, 15)	5	3	0.333	(15, 11)	1	0.733
15	(45, 9)	7	4	0.2	(15, 9)	2	0.6
18	(45, 8)	9	6	0.177	(15, 8)	3	0.53
21	(45, 6)	10	7	0.133	(15, 6)	4	0.4
22	(45, 4)	11	10	0.088	(15, 4)	5	0.266
31	(45, 3)	15	11	0.066	(15, 3)	5	0.2
45	(45, 1)	22	115	0.022	(15, 1)	7	0.066

5.4 General decoding principle

As the BCH code \mathcal{C}_n is embedded in the non-primitive BCH code \mathcal{C}_{bn} , so only decoding principal for the code \mathcal{C}_{bn} is explained. We use the decoding procedure which follows the same principle as of the primitive binary BCH code.

Take $a^1 \in \mathbb{F}_4^{bn}$ as a received vector. Now obtain the syndrome matrix of a^1 , and $S(a^1) =$

$a^1 H^T$. In this way, table of syndromes which is useful in determining the error vector e such that $S(a^1) = S(e)$ is calculated. So the decoding of received vector a^1 has done as the transmitted vector $v^1 = a^1 - e$. We acquire algebraic method for finding e from the syndrome vector $S(a^1)$.

Let \mathcal{C}_{bn} be the non-primitive BCH code with length bn and designed distance d_1 . Let H be the $(d_1 - 1) \times bn$ matrix over $\mathbb{F}_{4^{bs}}$. Writing $a^1 = (a_0^1, a_1^1, \dots, a_{bn-1}^1)$ in the polynomial form $a^1(x^{\frac{a}{b}}) = a_0^1 + a_1^1(x^{\frac{a}{b}}) + a_2^1(x^{\frac{a}{b}})^2 + \dots + a_{bn-1}^1(x^{\frac{a}{b}})^{bn-1}$. So the syndrome of the vector a^1 is

$$S(a^1) = \begin{bmatrix} a_0^1 & a_1^1 & \dots & a_{bn-1}^1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha^{c^1} & \alpha^{c^1+1} & \dots & \alpha^{c^1+d_1-2} \\ \alpha^{2c^1} & \alpha^{2(c^1+1)} & \dots & \alpha^{2(c^1+d_1-2)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(bn-1)c^1} & \alpha^{(bn-1)(c^1+1)} & \dots & \alpha^{(bn-1)(c^1+d_1-2)} \end{bmatrix}$$

and hence

$$S(a^1) = \begin{bmatrix} S_{c^1} & S_{c^1+1} & \dots & S_{c^1+d_1-2} \end{bmatrix},$$

where $S_j = a_0^1 + a_1^1 \alpha^j + \dots a_{bn-1}^1 \alpha^{(bn-1)j} = a^1(\alpha^j)$ for $j = c^1, c^1 + 1, \dots, c^1 + d_1 - 2$.

Now, let a codeword $v \in \mathcal{C}_{bn}$ is transmitted and the vector received is $a^1 = v^1 + e$, where e is the error vector. Then $S(e) = S(a^1)$. Let $e(x^{\frac{a}{b}}) = e_0 + e_1(x^{\frac{a}{b}}) + e_2(x^{\frac{a}{b}})^2 + \dots + e_{bn-1}(x^{\frac{a}{b}})^{bn-1}$ be the error polynomial. Suppose i_1, \dots, i_m be the positions where an error has occurred. Then $e_i \neq 0$ if and only if $i \in I = \{i_1, \dots, i_m\}$. Hence $e(x^{\frac{a}{b}}) = \sum_{i \in I} e_i(x^{\frac{a}{b}})^i$. As the code corrects up to t_1 errors, where $t_1 = \left\lfloor \frac{d_1-1}{2} \right\rfloor$. So we assume $m \leq t_1$, that is $2m < d_1$. Since $S(e) = S(a^1)$, we have $e(\alpha^j) = S_j$ for $j = c^1, c^1 + 1, \dots, c^1 + d_1 - 2$. Thus the $2m$ unknowns i_1, \dots, i_m and e_{i_1}, \dots, e_{i_m} satisfy the following system of $d_1 - 1$ linear equations in e_{i_1}, \dots, e_{i_m} :

$$\sum_{i \in I} e_i \alpha^{ji} = S_j, \quad j = c^1, c^1 + 1, \dots, c^1 + d_1 - 2 \dots (1).$$

We first obtain a solution for the error positions i_1, \dots, i_m . We define the error locator polynomial

$$f(x^{\frac{a}{b}}) = f_0 + f_1(x^{\frac{a}{b}}) + f_2(x^{\frac{a}{b}})^2 + \dots + f_{m-1}(x^{\frac{a}{b}})^{m-1} + (x^{\frac{a}{b}})^m.$$

Since $f(\alpha^i) = 0$ for each $i \in I$, we have

$$f_0 + f_1(\alpha^i) + \dots f_{m-1}(\alpha^i)^{m-1} + (\alpha^i)^m = 0.$$

On multiplying this equation by $e_i \alpha^{ji}$, we get

$$f_0 e_i \alpha^{ji} + f_1 e_i \alpha^{(j+1)i} + \dots f_{m-1} e_i \alpha^{(j+m-1)i} + e_i \alpha^{(j+m)i} = 0.$$

for each $i \in I$. Summing these m equations for $i = i_1, \dots, i_r$ and using the relations (1), we have

$$f_0 S_j + f_1 S_{j+1} + \dots f_{m-1} S_{j+m-1} + S_{j+m} = 0.$$

for each $j = c^1, c^1 + 1, \dots, c^1 + m - 1$. Thus the m unknowns f_0, f_1, \dots, f_{m-1} satisfy the following $m \times m$ system of linear equations:

$$\begin{bmatrix} S_{c^1} & S_{c^1+1} & \dots & S_{c^1+m-1} \\ S_{c^1+1} & S_{c^1+2} & \dots & S_{c^1+m} \\ \vdots & \vdots & \ddots & \vdots \\ S_{c^1+m-1} & S_{c^1+m} & \dots & S_{c^1+2m-2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{m-1} \end{bmatrix} = \begin{bmatrix} S_{c^1+m} \\ S_{c^1+m+1} \\ \vdots \\ S_{c^1+2m-1} \end{bmatrix} \dots\dots(2).$$

Let S denote the coefficient matrix in the above linear system. It can be verified by direct computation that $S = V D V^T$, where

$$V^1 = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \dots & \alpha^{i_m} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{i_1(m-1)} & \alpha^{i_2(m-1)} & \dots & \alpha^{i_m(m-1)} \end{bmatrix},$$

$$D^1 = \begin{bmatrix} e_{i_1} \alpha^{i_1 c^1} & 0 & \dots & 0 \\ 0 & e_{i_2} \alpha^{i_2 c^1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e_{i_m} \alpha^{i_m c^1} \end{bmatrix}.$$

V^1 is a Vandermonde matrix. Since α is a non-primitive bn th root of unity in $\mathbb{F}_{4^{bs}}$ and i_1, \dots, i_m are distinct integers in $\{0, \dots, bn - 1\}$, we have $\alpha^{i_1}, \dots, \alpha^{i_m}$ are all distinct. Hence $\det V \neq 0$. Further, e_{i_1}, \dots, e_{i_m} are all nonzero and hence $\det D \neq 0$. Therefore $\det S \neq 0$, and linear system (2) has a unique solution.

Let the number of positions where an error has occurred is $m \leq t_1$. If the actual number of error positions is less than m , then for any choice of distinct positions i_1, \dots, i_m , the coefficients e_{i_1}, \dots, e_{i_m} cannot be all zero. So $\det D = 0$. Hence m is the greatest positive integer $\leq t_1$ such that system (2) has a unique solution. Therefore we find the value of m by taking successively $m = t_1, t_1 - 1, \dots$ in system (2) until we have a value for which system (2) has a unique solution, which gives us the error locator polynomial

$$f(x^{\frac{a}{b}}) = f_0 + f_1(x^{\frac{a}{b}}) + f_2(x^{\frac{a}{b}})^2 + \dots + f_{m-1}(x^{\frac{a}{b}})^{m-1} + (x^{\frac{a}{b}})^m.$$

Now to find the roots of $f(x^{\frac{a}{b}})$, we put $x^{\frac{a}{b}} = \alpha^i$, $i = 0, 1, \dots$. By the definition of $f(x^{\frac{a}{b}})$, these roots are $\alpha^{i_1}, \dots, \alpha^{i_m}$. Thus we find the unique solution for the unknowns i_1, \dots, i_m . Having thus found the error vector e , we decode the received vector a as the codeword $v^1 = a^1 - e$.

To compute the syndrome of a BCH code we have $S_4 = (S_1)^4$, $S_8 = (S_4)^2$ and so on. If $m(x^{\frac{a}{b}})$ is the minimal polynomial of α , then $S_1 = a^1(\alpha)$ can be obtained by finding the remainder on dividing $a^1(x^{\frac{a}{b}})$ by $m(x^{\frac{a}{b}})$ and then putting $x^{\frac{a}{b}} = \alpha$ in it. In general, to find S_j , we divide $a^1(x^{\frac{a}{b}})$ by $m(x^{\frac{a}{b}})$ and find the remainder.

Decoding of the code \mathcal{C}_n from the decoding of the code \mathcal{C}_{bn} can be obtain as; take $x^{\frac{a}{b}} = y$, which gives $x^a = y^b$. In this way the code polynomial $v(x^{\frac{a}{b}})$ in $\mathbb{F}_4[x; \frac{a}{b}\mathbb{N}_0]_{bn}$ becomes $v^1(y)$. Again on replacing y by y^b , we get $v^1(y^b) = v^1(x^a)$. The remainder after dividing $v(x^a)$ by $(x^a)^n - 1$, will be the decoded vector of $\mathbb{F}_4[x; a\mathbb{N}_0]_n$ and the generator polynomial $g(x^a)$ divides $v(x^a)$.

The above discussion can be sum up in the following steps.

Step I: For a non-primitive BCH code \mathcal{C}_{bn} with designed distance d_1 , let $a(x^{\frac{a}{b}})$ be the received polynomial with m errors, where $m \leq t_1$.

Step II: Find the value of m by computing the syndromes, such that the system (2) has a unique solution.

Step III: Step II gives us the error locator polynomial $f(x^{\frac{a}{b}})$. Now find the roots of $f(x^{\frac{a}{b}})$ through which we obtain the error polynomial $e(x^{\frac{a}{b}})$.

Step IV: The received polynomial $a^1(x^{\frac{a}{b}})$ is decoded as $v^1(x^{\frac{a}{b}}) = a^1(x^{\frac{a}{b}}) - e(x^{\frac{a}{b}})$.

Step V: The code vector v in \mathcal{C}_n can be drag out from the decoded code vector v^1 in \mathcal{C}_{bn} by putting $x^{\frac{a}{b}} = y$ in corresponding code polynomial $v^1(x^{\frac{a}{b}})$. This gives $v^1(x^{\frac{a}{b}}) = v^1(y)$, which on replacing y by y^b becomes $v^1(y) = v^1(y^b) = v^1(x^a)$.

Step VI: Divide $v^1(x^a)$ obtain in Step V by $(x^a)^n - 1$, the remainder is the generator polynomial $g(x^a)$ or its multiple of the code \mathcal{C}_n . Hence its corresponding vector $v \in \mathcal{C}_n$.

Illustration

Let \mathcal{C}_{45} be a (45, 31) non-primitive BCH code with designed distance $d_1 = 5$. Assume that

$$\begin{aligned} a^1(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^{14} + (x^{\frac{2}{3}})^{13} + \alpha(x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^5 + (x^{\frac{2}{3}})^4 \\ &\quad + \alpha(x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + \alpha, \end{aligned}$$

is the received polynomial. The error position $l = 2$ and the syndromes are $S_1 = a^1(\beta) = \beta^3$, $S_2 = a^1(\beta^2) = \beta^6$, $S_3 = a^1(\beta^3) = \beta^{39}$ and $S_4 = (S_1)^4 = \beta^{12}$. The error locator polynomial is given by $f(x^{\frac{2}{3}}) = f_0 + f_1(x^{\frac{2}{3}}) + (x^{\frac{2}{3}})^2$. Then we have the following system of equations for f_0 , f_1 .

$$\begin{aligned} \begin{bmatrix} \beta^3 & \beta^6 \\ \beta^6 & \beta^{39} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix} &= \begin{bmatrix} \beta^{39} \\ \beta^{12} \end{bmatrix} \\ \begin{bmatrix} f_0 \\ f_1 \end{bmatrix} &= \begin{bmatrix} \frac{\beta^{39}}{\beta^{27}} & \frac{\beta^6}{\beta^{27}} \\ \frac{\beta^6}{\beta^{27}} & \frac{\beta^3}{\beta^{27}} \end{bmatrix} \begin{bmatrix} \beta^{39} \\ \beta^{12} \end{bmatrix} = \begin{bmatrix} \beta^{21} \\ \beta^3 \end{bmatrix}. \end{aligned}$$

Hence the error locator polynomial is $f(x^{\frac{2}{3}}) = \beta^{21} + \beta^3(x^{\frac{2}{3}}) + (x^{\frac{2}{3}})^2$. Trying successively $x = 1, \beta, \beta^2, \dots$, we find that β^6 and β^{15} are the roots. Hence the error polynomial is $e(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^{15}$. Thus we decode $a^1(x^{\frac{2}{3}})$ as $v^1(x^{\frac{2}{3}}) = a^1(x^{\frac{2}{3}}) + e(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{14} + (x^{\frac{2}{3}})^{13} + \alpha(x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^5 + (x^{\frac{2}{3}})^4 + \alpha(x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + \alpha$.

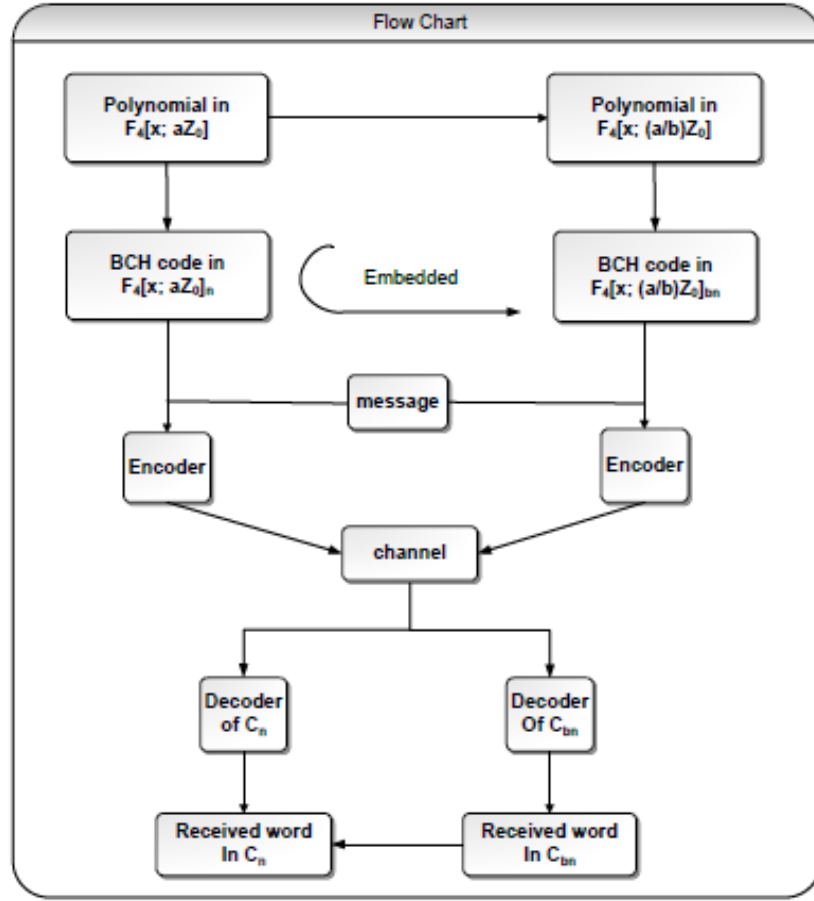
Now letting $x^{\frac{2}{3}} = y$, this gives $y^3 = x^2$, we get

$$\begin{aligned}
v^1(y^3) &= (y^3)^{14} + (y^3)^{13} + \alpha(y^3)^{12} + (y^3)^5 + (y^3)^4 + \alpha(y^3)^3 \\
&\quad + (y^3)^2 + (y^3) + \alpha \\
v^1(y^3) &= v^1(x^2) = (x^2)^{14} + (x^2)^{13} + \alpha(x^2)^{12} + (x^2)^5 + (x^2)^4 \\
&\quad + \alpha(x^2)^3 + (x^2)^2 + (x^2) + \alpha.
\end{aligned}$$

Where $v^1(x^2) \in \mathbb{F}_4[x; 2\mathbb{N}_0]_{15}$ and is completely divisible by $(x^2)^2 + (x^2) + \alpha$ the generator polynomial of non-primitive BCH code (15, 13).

Following is the flow chart of the complete scheme in which encoding and decoding of the

non primitive BCH code \mathcal{C}_{bn} and the primitive BCH code \mathcal{C}_n is occurring simultaneously.



Flow chart: Simultaneous encoding and decoding of the BCH codes \mathcal{C}_{bn} and \mathcal{C}_n

The following are the most significant outcomes of this chapter.

- 1) Over the four elements Galois field \mathbb{F}_4 , the existence of a non-primitive BCH code \mathcal{C}_{bn} of length bn based on a primitive n length BCH code \mathcal{C}_n , is ensured.
- 2) Embedding of the BCH code \mathcal{C}_n in the BCH code \mathcal{C}_{bn} is obtained, through which encoding and decoding of the BCH code \mathcal{C}_n is obtain via the BCH code \mathcal{C}_{bn} .
- 3) The BCH code \mathcal{C}_{bn} has greater error correction capability than of the BCH code \mathcal{C}_n with a small deprivation in the code rate.

4) In a BCH code of length n or bn , the code rate is better over the field \mathbb{F}_4 as compared to the codes obtained over the field \mathbb{F}_2 .

5) The possible choices of the BCH codes of a given length over \mathbb{F}_4 are more as compare to the BCH codes over \mathbb{F}_2 .

6) Among the non-primitive BCH codes of length bn there exists an interleaved code having same code rate and error correction capability, but is capable of correcting burst of length b .

This work is further generalized using Galois rings.

Chapter 6

Non-primitive BCH codes over Galois rings

In error-correcting codes, the code rate and error correction trade-off is one of the fundamental questions. In this chapter, a smart and novel approach is introduced to lever this matter. With the usage of a monoid ring a construction method of primitive and non-primitive BCH codes over Galois ring \mathbb{Z}_q , where $q = 2^m$ with $m > 1$, is given. Consequently, for a fixed m_0 , against n length primitive BCH codes \mathcal{C}_n and \mathcal{C}'_n (over \mathbb{Z}_2 and $\mathbb{Z}_{2^{m_0}}$ respectively), there exist two sequences $\{\mathcal{C}_{bj_n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{bj_n}\}_{j \geq 1}$ of non-primitive BCH codes (over \mathbb{Z}_2 and $\mathbb{Z}_{2^{m_0}}$ respectively). Through embedding and the 2 reduction map, relations intra and across, these two sequences are established. Thus, a data can be transmitted via any of coding scheme of \mathcal{C}_n , \mathcal{C}'_n , $\{\mathcal{C}_{bj_n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{bj_n}\}_{j \geq 1}$. This selection of scheme is based on the choice of better code rate or better error correction capability of the chosen code. A modified Berlekamp-Massey decoding algorithm is given for codes over Galois rings, which is also used for decoding BCH codes over Galois field.

6.1 BCH-codes as ideals in $\mathbb{Z}_{2^m}[x; a\mathbb{N}_0]_n$ and $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]_{bj_n}$

Throughout \mathbb{Z}_{2^m} , is the ring of integers modulo 2^m . the construction of BCH codes in $\mathbb{Z}_{2^m}[x; a\mathbb{N}_0]_n$ and $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]_{bj_n}$ as: The residue field of local ring \mathbb{Z}_{2^m} is \mathbb{Z}_2 . So, there is a natural projection $\mu : \mathbb{Z}_{2^m} \rightarrow \mathbb{Z}_2$ and it is extended as $\mu' : \mathbb{Z}_{2^m}[x; a\mathbb{N}_0] \rightarrow \mathbb{Z}_2[x; a\mathbb{N}_0]$ and $\mu' : \mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0] \rightarrow$

$\mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ defined as: $\mu'(f(x^a) = f_0 + f_1x^a + \dots + f_nx^{na}) = \mu(f_0) + \mu(f_1)x^a + \dots + \mu(f_n)x^{na}$ and $\mu'(f(x^{\frac{a}{b^j}}) = f_0 + f_1x^{\frac{a}{b^j}} + \dots + f_nx^{\frac{na}{b^j}}) = \mu(f_0) + \mu(f_1)x^{\frac{a}{b^j}} + \dots + \mu(f_n)x^{\frac{na}{b^j}}$.

Accordingly, an irreducible polynomial $f(x^a) \in \mathbb{Z}_{2^m}[x; a\mathbb{N}_0]$ is said to be basic irreducible polynomial if $\mu'(f(x^a) \in \mathbb{Z}_2[x; a\mathbb{N}_0]$ is irreducible. In a similar fashion an irreducible polynomial $f(x^{\frac{a}{b^j}}) \in \mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$ is said to be basic irreducible polynomial if $\mu'f(x^{\frac{a}{b^j}}) \in \mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ is irreducible.

The construction of a BCH code in the factor ring $\mathbb{Z}_{2^m}[x; a\mathbb{N}_0]_n$ is similar to that of a BCH code in $\mathbb{Z}_{2^m}[x]$ explained in [41], as $\mathbb{Z}_{2^m}[x; a\mathbb{N}_0] \subset \mathbb{Z}_{2^m}[x]$. For this, let C_n be a binary BCH code based on the positive integers $c, d, q = 2$ and n such that $2 \leq d \leq n$ with $\gcd(n, 2) = 1$ and $n = 2^s - 1$, where s is the degree of primitive irreducible polynomial $p(x^a)$ in $\mathbb{Z}_2[x; a\mathbb{N}_0]$. Let $\bar{\alpha} = \mu(\alpha)$ be primitive element in $GF(2, s)$. Then the corresponding element α has order $d(2^s - 1)$ in R^* , the group of unit elements of the Galois ring $GR(2^m, s)$ for some integer $d \geq 1$. Then element α^d generates the maximal cyclic subgroup G_{2^s-1} of R^* . Let $\xi = \alpha^d$ be a primitive element of G_{2^s-1} . Then if $\xi^{e_1}, \xi^{e_2}, \dots, \xi^{e_j}$ are roots of the polynomial $g(x^a)$, we can generate a BCH code over $GR(2^m, s)$ through this polynomial which is:

$$g(x^a) = \text{lcm}(M_{e_1}(x^a), M_{e_2}(x^a), \dots, M_{e_j}(x^a)),$$

where $M_{e_i}(x^a)$ is minimal polynomial of ξ^{e_i} . We call it a primitive BCH code over $GR(2^m, s)$. Furthermore,

$$\overline{g(x^a)} = \mu'(g(x^a)) = \text{lcm}(m_{e_1}(x^a), m_{e_2}(x^a), \dots, m_{e_j}(x^a)),$$

where $m_{e_i}(x^a)$ is minimal polynomial of $\mu'(\xi^{e_i})$, generates a BCH code over $GF(2)$.

BCH codes in which a code has length $b^j n$, $j > 1$, via the monoid ring $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$ for a fixed $m > 1$. These codes are obtained with the help of irreducible polynomial $p'(x^{\frac{a}{b^j}})$ of degree $b^j s$ over \mathbb{Z}_{2^m} . The irreducible polynomial $p'(x^{\frac{a}{b^j}})$ is taken such that $\mu'(p'(x^{\frac{a}{b^j}})) = p(x^{\frac{a}{b^j}})$. So,

$$\frac{\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]}{(p'(x^{\frac{a}{b^j}}))} = \{p_0 + p_1(x^{\frac{a}{b^j}}) + \dots + p_{b^j s-1}(x^{\frac{a}{b^j}})^{b^j s-1} : p_0, p_1, \dots, p_{b^j s-1} \in \mathbb{Z}_{2^m}\},$$

represents the set of residue classes of polynomials in indeterminate $x^{\frac{a}{b^j}}$ over \mathbb{Z}_{2^m} , modulo the polynomial $p'(x^{\frac{a}{b^j}})$. This is a unitary commutative ring, denoted by $\overline{R} = GR(2^m, b^j s)$ and it is

called the Galois extension ring of integers modulo ring \mathbb{Z}_{2^m} , having 2^{mb^js} number of elements for $j \geq 0$ and it is projected to a Galois field extension $\overline{K} = GF(2^{b^js})$,

$$\frac{\mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]}{(p(x^{\frac{a}{b^j}}))} = \{p_0 + p_1(x^{\frac{a}{b^j}}) + \dots + p_{b^js-1}(x^{\frac{a}{b^j}})^{b^js-1} : p_0, p_1, \dots, p_{b^js-1} \in \mathbb{Z}_2\},$$

of prime field \mathbb{Z}_2 having 2^{b^js} number of elements. We denote $GR(2^m, s)$ and $GF(2^s)$ by R and K , and their corresponding multiplicative groups of units by R^* and K^* respectively.

From [26, Theorem XIII.7], it follows that a polynomial irreducible over \mathbb{Z}_2 is also irreducible over \mathbb{Z}_{2^m} . Therefore, the irreducible polynomials in $\mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ listed in Chapter 3, Proposition 50, are also irreducible in $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$. The values of b are chosen in such a way that, for an s degree primitive irreducible polynomial in $\mathbb{Z}_2[x; a\mathbb{N}_0]$ we have a b^js degree irreducible polynomial in $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$ for $m \geq 1$. Now, after this selection the elements of the Galois ring $GR(2^m, b^js)$ are calculated with the help of irreducible polynomial $p'(x^{\frac{a}{b^j}})$, such that $\mu'(p'(x^{\frac{a}{b^j}})) = p(x^{\frac{a}{b^j}})$.

The following theorem extends [26, Lemma (XV.1)], from the case of polynomial ring to the monoid ring, where the coefficients are from \mathbb{Z}_{2^m} .

Theorem 76 *For $j \geq 0$ and $a \geq 1$, the Galois ring $GR(2^m, b^js)$, let $p'(x^{\frac{a}{b^j}})$ be a regular polynomial in $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$, such that $p(x^{\frac{a}{b^j}}) \in \mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ has a simple (i.e., non multiple) zero $\overline{\alpha}$ in \mathbb{Z}_2 . Then $p'(x^{\frac{a}{b^j}})$ has one and only one zero α in \mathbb{Z}_{2^m} with $\mu(\alpha) = \overline{\alpha}$.*

Let \overline{R}^* and \overline{K}^* be the multiplicative groups of units in $\overline{R} = GR(2^m, b^js)$ and $\overline{K} = GF(2^{b^js})$, respectively. Then \overline{R}^* is an Abelian group and can be written in the direct product of its cyclic sub-groups. The following Theorem extend [41, Theorem 2], for monoid rings.

Theorem 77 *\overline{R}^* and R^* has one and only one maximal cyclic subgroup. These cyclic subgroups have order b^jn and n respectively.*

The cyclic subgroups of \overline{R}^* and R^* are, respectively, generated by some powers of generators of the cyclic groups \overline{K}^* and K^* . These cyclic subgroups are, respectively, denoted by G_{b^jn} and G_n . Since, order of K^* is same as of G_n and both are cyclic, hence isomorphic to each other. Similarly, \overline{K}^* is isomorphic to G_{b^jn} . The following lemma is an extension of [3, Lemma 3.1], for the case of monoid rings.

Lemma 78 Let α be an element of $G_{b^j n}$ of order $b^j n$. Then the difference $\alpha^{l_1} - \alpha^{l_2}$ are units in $GR(2^m, b^j s)$ if $0 \leq l_1 \neq l_2 \leq b^j n - 1$ for $j \geq 0$ and $a \geq 1$.

Proof. As $\alpha^{l_1} - \alpha^{l_2}$ can be written as $-\alpha^{l_2} (1 - \alpha^{l_1 - l_2})$, where 1 denotes the unity of $GR(2^m, b^j s)$. The first term $-\alpha^{l_2}$ of product is a unit. The second term can be written as $1 - \alpha^k$ for some integer k in the interval $[1, b^j n - 1]$. Now if the element $1 - \alpha^k$ is not a unit for some $1 \leq k \leq b^j n - 1$, in $GR(2^m, b^j s)$, then $1 - \alpha^k$ will be in the maximal ideal M of the ring $GR(2^m, b^j s)$. Hence $\mu(\alpha^k) = \mu(1)$ and therefore $\bar{\alpha}$ has order k , a contradiction. ■

The following theorem is the extension of [3, Theorem 3.1].

Theorem 79 The minimum Hamming distance of BCH codes $\mathcal{C}_{b^j n}$ and $\mathcal{C}'_{b^j n}$ is d and satisfy $d \geq 2t + 1$ for $j \geq 0$ and $a \geq 1$.

Proof. Let c be a nonzero codeword in $\mathcal{C}_{b^j n}$ or $\mathcal{C}'_{b^j n}$ such that $w_H(c) \leq 2t$. Then $cH^T = 0$. Deleting $b^j n - 2t$ columns of matrix H corresponding to zeros of the codeword, the new matrix H' is Vandermonde. By lemma 78, it follows that the determinant is unit in R or \bar{R} . Hence, the only possibility for c is the zero codeword. ■

The following extend [41, Theorem 3].

Theorem 80 [41, Theorem] Let α generate a cyclic group of order $b^j n$ in \bar{R}^* , where $n = p^s - 1$. Then the polynomial $(x^{\frac{a}{b^j}})^{b^j n} - 1$ can be factored as

$$(x^{\frac{a}{b^j}})^{b^j n} - 1 = (x^{\frac{a}{b^j}} - \alpha)(x^{\frac{a}{b^j}} - \alpha^2) \dots (x^{\frac{a}{b^j}} - \alpha^{b^j n}),$$

if and only if $\mu(\alpha)$ has order $b^j n$ in \bar{K}^* for $j \geq 0$ and $a \geq 1$.

Proof. Let $\mu(\alpha) = \bar{\alpha}$ has order $b^j n$ in \bar{K}^* . Then $(x^{\frac{a}{b^j}})^{b^j n} - 1 = (x^{\frac{a}{b^j}} - \alpha)(x^{\frac{a}{b^j}} - \alpha^2) \dots (x^{\frac{a}{b^j}} - \alpha^{b^j n})$. Let $\bar{\mathbb{F}} = \{\bar{\alpha}, \bar{\alpha}^2, \dots, \bar{\alpha}^{b^j n}\}$. Since $(x^{\frac{a}{b^j}})^{b^j n} - 1$ has no multiple zeros in \bar{K}^* , from Theorem 76, it is concluded that; corresponding to each $\bar{\alpha}^l$ there is a unique element say α_l in \bar{R}^* such that $\mu(\alpha_l) = \bar{\alpha}^l$ and α_l is a root of $(x^{\frac{a}{b^j}})^{b^j n} - 1$ in \bar{R}^* . In general, factorization over ring with zero divisor is not unique. To show that $(x^{\frac{a}{b^j}})^{b^j n} - 1$ can be factored as $(x^{\frac{a}{b^j}} - \alpha)(x^{\frac{a}{b^j}} - \alpha^2) \dots (x^{\frac{a}{b^j}} - \alpha^{b^j n})$, where $\alpha_l = \alpha^l$ for $l = 1, 2, 3, \dots, n$, we have to show that the set $\mathbb{F} = \{\alpha, \alpha^2, \dots, \alpha^{b^j n}\}$ exhausts all roots of $(x^{\frac{a}{b^j}})^{b^j n} - 1$ in \bar{R}^* . Let α^* be a root of $(x^{\frac{a}{b^j}})^{b^j n} - 1$ not in \mathbb{F} . Then $\mu(\alpha^*)$

is a root of $(x^{\frac{a}{b^j}})^{b^j n} - 1$ over \overline{K}^* and it cannot coincide with any element of $\overline{\mathbb{F}}$. Also $\mu(\alpha^*)$ cannot be a new distinct element of $\overline{\mathbb{F}}$, because $(x^{\frac{a}{b^j}})^{b^j n} - 1$ has exactly $b^j n$ roots in \overline{K}^* . Thus, $(x^{\frac{a}{b^j}} - \alpha)(x^{\frac{a}{b^j}} - \alpha^2) \dots (x^{\frac{a}{b^j}} - \alpha^{b^j n})$ is the only possible factorization of $(x^{\frac{a}{b^j}})^{b^j n} - 1$ over \overline{R}^* . Now, suppose that $(x^{\frac{a}{b^j}})^{b^j n} - 1$ can be factored over \overline{R}^* as above, as $(p, b^j n) = 1$, then from Theorem 77, $b^j n$ must be divisor of $p^{b^j s} - 1$. This yields

$$(x^{\frac{a}{b^j}})^{b^j n} - 1 = (x^{\frac{a}{b^j}} - \overline{\alpha})(x^{\frac{a}{b^j}} - \overline{\alpha}^2) \dots (x^{\frac{a}{b^j}} - \overline{\alpha}^{b^j n}) \text{ over } \overline{K}^*.$$

Thus, $\overline{\alpha}$ has order $b^j n$ because $(x^{\frac{a}{b^j}})^{b^j n} - 1$ has no multiple zeros in \overline{K}^* . ■

Extending [41, Lemma 1], for monoid ring we get the following results.

Lemma 81 *Let $\overline{\alpha}$ generates a cyclic subgroup of order $b^j n$ in \overline{K}^* . Then α generates a cyclic subgroup of order $(b^j n)d$ in \overline{R}^* , where d is an integer greater than or equal to 1, and α^d generates a cyclic subgroup $G_{b^j n}$ in \overline{R}^* for $j \geq 0$.*

The following remark is of great importance, as it gives the exact value of the power of element through which the maximal cyclic subgroup $G_{b^j n}$ in $\overline{R} = GR(2^m, b^j s)$ is generated.

Remark 82 *In $\overline{R} = GR(2^m, b^j s)$, the maximal cyclic subgroup $G_{b^j n}$ is generated by $\alpha^{2^{m-1}}$ and satisfies the relation $\alpha^{(2^{m-1})b^j n} = 1$ for all $m \geq 1$. The element α is selected in such a way that, $\overline{\alpha}$ generates a cyclic subgroup of order $b^j n$ in \overline{K}^* .*

Definition 83 *Let α be a non-primitive element of $G_{b^j n}$, then cyclic BCH code of length $b^j n$ over R is called **non-primitive BCH code** generated by generating polynomial $g'(x^{\frac{a}{b^j}})$ having roots $\alpha^{b'+1}, \alpha^{b'+2}, \dots, \alpha^{b'+2t}$, where $b', j \geq 0$ and $t \geq 1$, i.e.,*

$$g'(x^{\frac{a}{b^j}}) = \text{lcm}\{M_1(x^{\frac{a}{b^j}}), M_2(x^{\frac{a}{b^j}}), \dots, M_{2t}(x^{\frac{a}{b^j}})\},$$

where $M_i(x^{\frac{a}{b^j}})$ is the minimal polynomial of $\alpha^{b'+i}$ for $1 \leq i \leq 2t$.

Locator vector in this case is given by $\eta = (\alpha^0, \alpha^{b'+1}, \alpha^{2(b'+1)}, \dots, \alpha^{(b^j n - 1)(b'+1)})$ and parity check matrix takes the form

$$H = \begin{bmatrix} 1 & \alpha^{b'+1} & \alpha^{2(b'+1)} & \dots & \alpha^{(b'n-1)(b'+1)} \\ 1 & \alpha^{b'+2} & \alpha^{2(b'+2)} & \dots & \alpha^{(b'n-1)(b'+2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b'+2t} & \alpha^{2(b'+2t)} & \dots & \alpha^{(b'n-1)(b'+2t)} \end{bmatrix}. \quad (6.1)$$

The following theorem is an extension of [41, Theorem 4].

Theorem 84 *Let $g'(x^{\frac{a}{b^j}})$ be a generator polynomial of cyclic BCH code over $GR(2^m, b^j s)$ with length $b^j n$ and $\alpha^{e_1}, \alpha^{e_2}, \dots, \alpha^{e_{b^j n - m}}$ be roots of $g'(x^{\frac{a}{b^j}})$ in $G_{b^j n}$. Then the minimum Hamming distance of the code is greater than the largest number of consecutive integers modulo $b^j n$ in the set $E = \{e_1, e_2, \dots, e_{b^j n - m}\}$.*

Proof. Let \mathcal{C} be the BCH code generated by $g'(x^{\frac{a}{b^j}}) \in \mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$ and \mathcal{C}_1 be the code generated by $g(x^{\frac{a}{b^j}}) = \mu(g'(x^{\frac{a}{b^j}})) \in \mathbb{Z}_2[x; \frac{a}{b^j} \mathbb{N}_0]$. Then for any $v'(x^{\frac{a}{b^j}}) \in \mathcal{C}$, $v(x^{\frac{a}{b^j}}) \in \mathcal{C}_1$. Let d be the number of consecutive integers modulo $b^j n$ in the set E . Suppose that the minimum distance of \mathcal{C} is less than $d + 1$. Let $r'(x^{\frac{a}{b^j}}) \in \mathcal{C}$ be such that $b^j n$ -tuple r has Hamming weight less than $d + 1$. Then if $r(x^{\frac{a}{b^j}}) = \mu'(r'(x^{\frac{a}{b^j}}))$, the Hamming weight of the vector r is less than $d + 1$. But $g(x^{\frac{a}{b^j}})$ has roots d consecutive powers of $\mu(\alpha)$. Therefore by BCH bound the code \mathcal{C}_1 has minimum distance at least $d + 1$. Hence the minimum Hamming distance of \mathcal{C} must be at least $d + 1$. ■

The algorithm for constructing a non-primitive BCH code over $GR(2^m, b^j s)$ is as follows:

1. Choose an irreducible polynomial $p(x^{\frac{a}{b^j}})$ over $GF(2^{b^j s})$ such that $p'(x^{\frac{a}{b^j}})$ is irreducible over \mathbb{Z}_{2^m} and forms the extension ring $\overline{R} = GR(2^m, b^j s)$.
2. Suppose $\overline{\alpha} = \mu(\alpha)$ is in \overline{R}^* . If α has order $d \cdot b^j n$ in \overline{R}^* for some integer d , then α^d generates $G_{b^j n}$.
3. If $\alpha^{e_1}, \alpha^{e_2}, \dots, \alpha^{e_{b^j n - m}}$ are selected to be roots of $g'(x^{\frac{a}{b^j}})$, find $M_{e_i}(x^{\frac{a}{b^j}})$ for $i = 1, 2, \dots, b^j n - m$. Thus, $g'(x^{\frac{a}{b^j}})$ is the lcm of these $M_{e_i}(x^{\frac{a}{b^j}})$. The length of code is lcm of orders of α^{e_i} and minimum distance is greater than largest number of consecutive integers in the set $e = \{e_1, e_2, \dots, e_{b^j n - m}\}$.

6.2 Relation between the sequences $\{C_{b^j n}\}_{j \geq 0}$ and $\{C'_{b^j n}\}_{j \geq 0}$ of BCH codes

In this section, we discuss the relation between the BCH codes constructed through monoid rings $\mathbb{Z}_2[x; a\mathbb{N}_0]$, $\mathbb{Z}_{2^m}[x; a\mathbb{N}_0]$, $\mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ and $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$. Accordingly, these are the codes \mathcal{C}_n , \mathcal{C}'_n , $\{\mathcal{C}_{b^j n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ over $GF(2^s)$, $GR(2^m, s)$, $GF(2^{b^j s})$ and $GR(2^m, b^j s)$ respectively.

Their relation are explained in the following steps:

1. Primitive polynomials in $\mathbb{Z}_2[x; a\mathbb{N}_0]$ gives irreducible polynomials in $\mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ and $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$. Therefore, corresponding to n length primitive BCH codes \mathcal{C}_n and \mathcal{C}'_n (over \mathbb{Z}_2 and \mathbb{Z}_{2^m} respectively), there exist two sequences $\{\mathcal{C}_{b^j n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ of non-primitive BCH codes (over \mathbb{Z}_2 and \mathbb{Z}_{2^m}).

2. The generator polynomial $g'(x^{\frac{a}{b^j}})$ of any $b^j n$ length code in the sequence $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ is transformed to generator polynomial $g(x^{\frac{a}{b^j}})$ of same $b^j n$ length code in $\{\mathcal{C}_{b^j n}\}_{j \geq 1}$ by the reduction map μ for $j \geq 1$, $a > 1$.

3. The generator polynomials of BCH codes $\{\mathcal{C}_{b^j n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$, after reduction modulo $(x^{\frac{a}{b^j}})^n - 1$, gives b^j times repeated pattern of the generator polynomials of BCH codes \mathcal{C}_n and \mathcal{C}'_n respectively, when both n and $b^j n$ lengths BCH codes have same dimension k .

4. The generator polynomials $g(x^{\frac{a}{b^j}})$ of $b^j n$ length BCH codes in $\mathbb{Z}_2[x; \frac{a}{b^j}\mathbb{N}_0]$ are obtained from the generator polynomials $g'(x^{\frac{a}{b^j}})$ of same length BCH codes in $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{N}_0]$ by reducing the coefficient modulo 2.

5. In last, we observe that for a fixed value of m the generator polynomial of a non primitive BCH code $(b^j n, k)$ code obtained through the monoid ring $\mathbb{Z}_{2^m}[x; \frac{a}{b^j}\mathbb{Z}_{\geq 0}]$ can be obtained from a generator polynomial of primitive BCH (n, k) code constructed through the monoid ring $\mathbb{Z}_{2^m}[x; a\mathbb{Z}_{\geq 0}]$, a subring of polynomial ring $\mathbb{Z}_{2^m}[x]$.

The conversion of these BCH codes is as follows.

$$\begin{array}{ccccccc} \mathcal{C}'_n & \hookrightarrow & \mathcal{C}'_{bn} & \hookrightarrow & \mathcal{C}'_{b^2 n} & \hookrightarrow & \dots \hookrightarrow \mathcal{C}'_{b^j n} \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \cdot \\ \mathcal{C}_n & \hookrightarrow & \mathcal{C}_{bn} & \hookrightarrow & \mathcal{C}_{b^2 n} & \hookrightarrow & \dots \hookrightarrow \mathcal{C}_{b^j n} \end{array}$$

From \mathcal{C}'_n or from any $b^j n$ length code $\mathcal{C}'_{b^j n}$ we obtain all other codes by some simple steps.

Corresponding to \mathcal{C}_n or \mathcal{C}_{bn} codes we have a great variety of codes in $\mathbb{Z}_{2^m}[x; a\mathbb{N}_0]$ or $\mathbb{Z}_{2^m}[x; \frac{a}{b}\mathbb{N}_0]$ for different values of m . By increasing value of m , the number of codewords in these codes also increases.

The following example illustrates the construction of non-primitive BCH codes when $b > a > 1$ and for $m = 2, 3$.

Example 85 For a primitive polynomial $p(x^2) = 1 + (x^2) + (x^2)^4$ in $\mathbb{Z}_2[x; 2\mathbb{N}_0]$, there is an irreducible polynomial $p(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^3 + 1$ in $\mathbb{Z}_2[x; \frac{2}{3}\mathbb{N}_0]$, by Table 4, which is also irreducible in $\mathbb{Z}_{2^2}[x; \frac{2}{3}\mathbb{N}_0]$. Here $n = 15$ and $b = 3$, it follows that $bn = n' = 45$. Let u be an element in $GF(2, 12)$, satisfies the relation $u^{12} + u^3 + 1 = 0$. Then $u^{90} = 1$ in $GR(2^2, 12)$ shows that the elements of the cyclic subgroup G_{45} are generated by $\alpha = u^2$ by 82. Now, to calculate generator polynomials $g'(x^{\frac{2}{3}})$ we first calculate the minimal polynomials. By [27, Theorem 4.4.2], $\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}, \alpha^{19}, \alpha^{38}, \alpha^{31}, \alpha^{17}, \alpha^{34}, \alpha^{23}$ all have same minimal polynomial $M_1(x^{\frac{2}{3}})$ which is the generating polynomial of BCH code with designed distances $d' = 3$.

$$M_1(x^{\frac{2}{3}}) = g(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{12} + 2(x^{\frac{2}{3}})^6 + 3(x^{\frac{2}{3}})^3 + 1.$$

Similarly, other minimal polynomials can be calculated through which we get the following generating polynomials with designed distance $d' = 5, 7, 9, 11$, and 45 respectively.

$$\begin{aligned} g'(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{16} + 2(x^{\frac{2}{3}})^{14} + 3(x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} + 2(x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^7 + \\ &\quad 2(x^{\frac{2}{3}})^6 + 2(x^{\frac{2}{3}})^5 + 2(x^{\frac{2}{3}})^4 + 3(x^{\frac{2}{3}})^3 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1 \end{aligned}$$

$$\begin{aligned} g'(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{22} + 2(x^{\frac{2}{3}})^{20} + (x^{\frac{2}{3}})^{18} + 2(x^{\frac{2}{3}})^{17} + 2(x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{15} + \\ &\quad 2(x^{\frac{2}{3}})^{14} + 2(x^{\frac{2}{3}})^{13} + 3(x^{\frac{2}{3}})^{12} + 2(x^{\frac{2}{3}})^{11} + (x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^9 + \\ &\quad 2(x^{\frac{2}{3}})^7 + 2(x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1 \end{aligned}$$

$$\begin{aligned} g'(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{34} + 2(x^{\frac{2}{3}})^{32} + 3(x^{\frac{2}{3}})^{31} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{19} + 2(x^{\frac{2}{3}})^{17} + \\ &\quad 3(x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1 \end{aligned}$$

$$\begin{aligned}
g'(x^{\frac{2}{3}}) = & (x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{37} + 3(x^{\frac{2}{3}})^{36} + 2(x^{\frac{2}{3}})^{35} + 3(x^{\frac{2}{3}})^{34} + 2(x^{\frac{2}{3}})^{33} + \\
& 2(x^{\frac{2}{3}})^{32} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{22} + 3(x^{\frac{2}{3}})^{21} + 2(x^{\frac{2}{3}})^{20} + \\
& 3(x^{\frac{2}{3}})^{19} + 2(x^{\frac{2}{3}})^{18} + 2(x^{\frac{2}{3}})^{17} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^7 + \\
& 3(x^{\frac{2}{3}})^6 + 2(x^{\frac{2}{3}})^5 + 3(x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^3 + 2(x^{\frac{2}{3}})^2 + 1
\end{aligned}$$

$$g'(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{44} + (x^{\frac{2}{3}})^{43} + (x^{\frac{2}{3}})^{42} + (x^{\frac{2}{3}})^{41} + \dots + (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + 1.$$

Now, $g(x^{\frac{2}{3}}) = \mu(g'(x^{\frac{2}{3}}))$ gives the generator polynomials of BCH code C_{45} with symbols from $GF(2, 12)$. We have generator polynomials in $\mathbb{Z}_{2^2}[x; \frac{2}{3}\mathbb{N}_0]$ by reducing the coefficients modulo 2^2 of generator polynomials in $\mathbb{Z}_{2^2}[x; \frac{2}{3}\mathbb{N}_0]$. To drive primitive BCH code C'_{15} from non-primitive BCH code C'_{45} , among the generator polynomials of C'_{45} over $\mathbb{Z}_{2^2}[x; \frac{2}{3}\mathbb{N}_0]$, the generator polynomials of $(45, 11)$, $(45, 7)$, $(45, 5)$, $(45, 1)$ codes are transformed to generator polynomials of $(15, 11)$, $(15, 7)$, $(15, 5)$, $(15, 1)$ codes over $\mathbb{Z}_{2^2}[x; 2\mathbb{N}_0]$. For example $(45, 11)$ code has generator polynomial

$$\begin{aligned}
g'(x^{\frac{2}{3}}) = & (x^{\frac{2}{3}})^{34} + 2(x^{\frac{2}{3}})^{32} + 3(x^{\frac{2}{3}})^{31} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{19} + 2(x^{\frac{2}{3}})^{17} \\
& + 3(x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1,
\end{aligned}$$

by reduction modulo $(x^{\frac{2}{3}})^{15} - 1$, we have

$$\begin{aligned}
g'(x^{\frac{2}{3}}) = & (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1 + (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + \\
& 3(x^{\frac{2}{3}}) + 1 + (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1,
\end{aligned}$$

it is seen that the pattern $(x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1$ is repeated three times, which is the generator polynomial of $(15, 11)$ code over $\mathbb{Z}_{2^2}[x; 2\mathbb{N}_0]$ by considering discriminant x^2 instead of $x^{\frac{2}{3}}$. By the similar method we obtain other generator polynomials.

Similarly, by taking $j = 2$, non-primitive BCH codes of length 135 are obtained. Non-primitive BCH codes in $\mathbb{Z}_{2^3}[x; \frac{2}{3}\mathbb{N}_0]$ are obtained from same irreducible polynomial $p'(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^3 + 1$. In this case, u satisfying the relation $u^{12} + u^3 + 1 = 0$ which gives $u^{180} = 1$ in $GR(2^3, 12)$, thus, the elements of G_{45} are generated by $\alpha = u^4$. Following is the generating polynomials of BCH code with design distance $d' = 3$.

$$g'(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{12} + 4(x^{\frac{2}{3}})^9 + 6(x^{\frac{2}{3}})^6 + 3(x^{\frac{2}{3}})^3 + 1.$$

Following table gives the comparison between minimum distances, code rate and error correction capability of codes in $\mathbb{Z}_2[x; 2\mathbb{N}_0]_{15}$, $\mathbb{Z}_{2^2}[x; \frac{2}{3}\mathbb{N}_0]_{45}$ and $\mathbb{Z}_{2^2}[x; \frac{2}{3^2}\mathbb{N}_0]_{135}$.

Table 24: Comparison between C_{15} , C_{45} , C_{135}

(n, k)	d	t	R	(n, k)	d_1	t_1	R_1	(n, k)	d_2	t_2	R_2
(15, 11)	3	1	0.733	(45, 33)	3	1	0.733	(135, 99)	3	1	0.733
(15, 7)	5	2	0.466	(45, 29)	5	2	0.644	(135, 87)	5	2	0.644
(15, 5)	7	3	0.333	(45, 23)	7	3	0.511	(135, 69)	7	3	0.511
(15, 1)	15	7	0.066	(45, 11)	9	4	0.244	(135, 33)	9	4	0.244
				(45, 7)	15	7	0.155	(135, 29)	15	7	0.215
				(45, 5)	21	10	0.11	(135, 23)	21	10	0.170
				(45, 1)	45	22	0.022	(135, 11)	27	13	0.0814
								(135, 7)	45	22	0.0518
								(135, 5)	63	31	0.0370
								(135, 1)	135	67	0.007

From Table 25, it is clear that the BCH codes in $\mathbb{Z}_{2^2}[x; \frac{2}{3}\mathbb{N}_0]_{3n}$ has better error correction capability but has less code rate than the BCH code in $\mathbb{Z}_2[x; 2\mathbb{N}_0]$. Therefore, during data transmission if more error correction capability is required, then choose 45 or 135 length BCH codes and if better code rate is required, then use 15 length BCH codes. The minimum distance of BCH codes having dimension less than 15 via $\mathbb{Z}_2[x; \frac{2}{3}\mathbb{N}_0]$ is 3 times the minimum distance of BCH codes in $\mathbb{Z}_2[x; 2\mathbb{N}_0]_{15}$. The BCH codes (15, 11), (15, 7), (15, 5) and (15, 1) have minimum distance 3, 5, 7 and 15 respectively. And BCH codes (45, 11), (45, 7), (45, 5) and (45, 1) have minimum distance 9, 15, 21 and 45 respectively.

Remark 86 BCH codes over Galois field $GF(2^{b^j s})$ and Galois ring $GR(2^m, b^j s)$, for $j \geq 0$, have same code rates and error correction capability.

Remark 87 Number of codewords in BCH codes over Galois ring $GR(2^m, b^j s)$ is greater than

number of codewords in BCH codes over its corresponding Galois field $GF(2^{b^j s})$ for $j \geq 0$.

The generalized polynomial $g'(x^{\frac{a}{b^j}})$ over $G_{b^j n}$ divides $(x^{\frac{a}{b^j}})^{b^j n} - 1$ in $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$. So, there is a BCH code $\mathcal{C}'_{b^j n}$ in the family $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ generated by $g'(x^{\frac{a}{b^j}})$ in $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]_{b^j n}$. By the same argument, as $b^j n$ divides $n_j = 2^{b^j s} - 1$, so $(x^{\frac{a}{b^j}})^{b^j n} - 1$ divides $(x^{\frac{a}{b^j}})^{n_j} - 1$ in $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$. It follows that $((x^{\frac{a}{b^j}})^{n_j} - 1) \subset ((x^{\frac{a}{b^j}})^{b^j n} - 1)$. Consequently, third isomorphism theorem for rings gives

$$\frac{\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0] / ((x^{\frac{a}{b^j}})^{n_j} - 1)}{((x^{\frac{a}{b^j}})^{b^j n} - 1) / ((x^{\frac{a}{b^j}})^{n_j} - 1)} \simeq \frac{\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]}{((x^{\frac{a}{b^j}})^{b^j n} - 1)} \simeq \frac{\mathbb{Z}_{2^m}[x; a \mathbb{N}_0]}{((x^a)^n - 1)}.$$

Thus, there is embedding $\mathcal{C}'_n \hookrightarrow \mathcal{C}'_{b^j n} \hookrightarrow \mathcal{C}'_{n_j}$ of BCH codes, whereas $\mathcal{C}'_n, \mathcal{C}'_{b^j n}, \mathcal{C}'_{n_j}$ are respectively primitive BCH, non-primitive BCH and primitive BCH codes. Whereas, the embedding $\mathcal{C}'_n \hookrightarrow \mathcal{C}'_{b^j n}$ is defined as:

$$a(x^a) = a_0 + a_1(x^a) + \dots + a_{n-1}(x^a)^{n-1} \mapsto a_0 + a_1(x^{\frac{a}{b^j}})^{b^j} + \dots + a_{n-1}(x^{\frac{a}{b^j}})^{b^j(n-1)} = a(x^{\frac{a}{b^j}}), \quad (6.2)$$

where $a(x^a) \in \mathcal{C}'_n$ and $a(x^{\frac{a}{b^j}}) \in \mathcal{C}'_{b^j n}$. Also, if $g'(x^{\frac{a}{b^{j-1}}})$ is the generator polynomial of the binary non-primitive $b^{j-1}n$ length BCH code in $\mathbb{Z}_{2^m}[x; \frac{a}{b^{j-1}} \mathbb{Z}_{\geq 0}]_{b^{j-1}n}$, then $g'(x^{\frac{a}{b^j}})$ is the generator polynomial of the binary non-primitive $b^j n$ length BCH code in the ring $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{Z}_{\geq 0}]_{b^j n}$. Thus $\mathcal{C}'_{b^{j-1}n}$ is embedded in $\mathcal{C}'_{b^j n}$ under the monomorphism defined as; $a(x^{\frac{a}{b^{j-1}}}) \mapsto a(x^{\frac{a}{b^j}})$. This sort of relationship also holds among $\mathcal{C}_n, \mathcal{C}_{b^j n}$ and \mathcal{C}_{n_j} .

The above discussion shapes the following.

Theorem 88 *Let \mathcal{C}_n and \mathcal{C}'_n be primitive BCH codes of length $n = 2^s - 1$ respectively obtained by monoid rings $\mathbb{Z}_2[x; a \mathbb{N}_0]$ and $\mathbb{Z}_{2^m}[x; a \mathbb{N}_0]$. Then following hold.*

1) *There exist the sequences $\{\mathcal{C}_{b^j n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ of non-primitive BCH codes such that $\mathcal{C}_{b^j n}$ and $\mathcal{C}'_{b^j n}$ are respectively obtained by $\mathbb{Z}_2[x; \frac{a}{b^j} \mathbb{N}_0]$ and $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$ with $b^j n$ length for $j \geq 1$.*

2) *The primitive BCH codes $\mathcal{C}_n, \mathcal{C}'_n$ are respectively embedded in the non-primitive BCH codes of the sequences $\{\mathcal{C}_{b^j n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ for each value of j .*

3) *The non-primitive BCH codes of the sequences $\{\mathcal{C}_{b^j n}\}_{j \geq 1}$ and $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ have following embeddings $\mathcal{C}_{bn} \hookrightarrow \mathcal{C}_{b^2 n} \hookrightarrow \dots \hookrightarrow \mathcal{C}_{b^j n} \hookrightarrow \dots$ and*

$$\mathcal{C}'_{bn} \hookrightarrow \mathcal{C}'_{b^2 n} \hookrightarrow \dots \hookrightarrow \mathcal{C}'_{b^j n} \hookrightarrow \dots \text{ respectively.}$$

Conversion of generating polynomial $g(x^a)$ of \mathcal{C}'_n to $g(x^{\frac{a}{b^j}})$ of $\mathcal{C}'_{b^j n}$

A generator polynomial of primitive (n, k) BCH code over \mathbb{Z}_{2^m} , can be converted to the generator polynomial of non-primitive $(b^j n, k)$ BCH code over \mathbb{Z}_{2^m} by some simple steps.

First change the indeterminate x^a to $x^{\frac{a}{b^j}}$. Then write the generator polynomial of (n, k) BCH code b^j times. In last multiply the 1^{st} pattern of generator polynomial of (n, k) BCH code with $(x^{\frac{a}{b^j}})^{n(b^j-1)}$, 2^{nd} pattern with $(x^{\frac{a}{b^j}})^{n(b^j-2)}$, 3^{rd} pattern with $(x^{\frac{a}{b^j}})^{n(b^j-3)}$ and so on.

The generator polynomial of $(15, 11)$ BCH code is given by

$$g'(x^2) = (x^2)^4 + 2(x^2)^2 + 3(x^2) + 1.$$

To find generator polynomial of $(45, 11)$ BCH code change the indeterminate x^2 to $x^{\frac{2}{3}}$ as

$$g'(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1.$$

Now, by writing this pattern 3 times and by multiplying 1^{st} pattern with $(x^{\frac{2}{3}})^{30}$, 2^{nd} pattern with $(x^{\frac{2}{3}})^{15}$, 3^{rd} pattern with 1, we have

$$\begin{aligned} g'(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{34} + 2(x^{\frac{2}{3}})^{32} + 3(x^{\frac{2}{3}})^{31} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{19} + 2(x^{\frac{2}{3}})^{17} + \\ &3(x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^2 + 3(x^{\frac{2}{3}}) + 1, \end{aligned}$$

which is the generator polynomial of $(45, 11)$ BCH code. By this method, we can transform primitive BCH codes of length 15 to following non-primitive BCH codes of length 45 and 135.

Table 25: Codes in which smaller codes repeats 3 times

(n, k)	d	t	R	(bn, k)	d_1	t_1	R_1	(b^2n, k)	d_2	t_2	R_2
$(15, 11)$	3	1	0.733	$(45, 11)$	9	4	0.244	$(135, 11)$	27	13	0.0814
$(15, 7)$	5	2	0.466	$(45, 7)$	15	7	0.155	$(135, 7)$	45	22	0.0518
$(15, 5)$	7	3	0.333	$(45, 5)$	21	10	0.11	$(135, 5)$	63	31	0.0370
$(15, 1)$	15	7	0.066	$(45, 1)$	45	22	0.022	$(135, 1)$	135	67	0.007

The overview of above discussion is shown in the following.

	Code rate \uparrow	Error	corrtn.	capacity \uparrow			
Code length \Rightarrow	n	$<$	bn	$<$	b^2n	$<$	$<$ $b^l n$
# of codwrds. \uparrow	\mathcal{C}'_n	\hookrightarrow	\mathcal{C}'_{bn}	\hookrightarrow	\mathcal{C}'_{b^2n}	\hookrightarrow ... \hookrightarrow	$\mathcal{C}'_{b^l n}$
	\downarrow		\downarrow		\downarrow		\downarrow
# of codwrds. \downarrow	\mathcal{C}_n	\hookrightarrow	\mathcal{C}_{bn}	\hookrightarrow	\mathcal{C}_{b^2n}	\hookrightarrow ... \hookrightarrow	$\mathcal{C}_{b^l n}$

6.3 Decoding procedure

Interlando, Palazzo and Elia [18] proposed a decoding procedure based on Berlekamp-Massey algorithm for Reed-Solomon and BCH codes over finite ring \mathbb{Z}_q , where q is the power of some prime p . Further, in [3] Andrade and Palazzo explained the decoding procedure of BCH code $\mathcal{C}(n, \eta)$ with same algorithm given in [18] that can correct all errors up to designed distance t , i.e., whose designed distance is greater than or equal to $2t + 1$.

In this study, we address decoding procedure of non-primitive BCH codes of length $b^j n$ from the sequence $\{\mathcal{C}'_{b^j n}\}_{j \geq 1}$ obtained through the monoid ring $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$. Here the given decoding procedure is similar to that of Berlekamp-Massey algorithm but with some modifications. Interestingly it is established that this algorithm is also applied to primitive BCH codes of length n by taking $a > 1$ and $j = 0$. To describe the steps of algorithm, we first consider the monoid ring $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$ and then α as a non-primitive element of maximal cyclic subgroup $G_{b^j n}$. The parity check matrix for non-primitive BCH codes over $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$ is given in (6.1), where t represents the number of errors that can be corrected. Let $\mathbf{c} = (c_1, c_2, \dots, c_{b^j n})$ be the codeword and $\mathbf{r} = (r_1, r_2, \dots, r_{b^j n})$ be the received vector. Then error vector is given by $\mathbf{e} = (e_1, e_2, \dots, e_{b^j n}) = \mathbf{r} - \mathbf{c}$.

The proposed decoding procedure consists of four major steps.

Step 1: Calculation of the syndromes $S_i = \mathbf{r}H^T$, where $i = 1, 2, \dots, 2t$.

Step 2: Calculation of the symmetric functions $\sigma_1, \sigma_2, \dots, \sigma_v$ from S_i .

Step 3: Calculation of the error location numbers x_1, x_2, \dots, x_v from $\sigma_1, \sigma_2, \dots, \sigma_v$.

Step 4: Calculation of the error magnitudes y_1, y_2, \dots, y_v from x_1, x_2, \dots, x_v and S_i .

Since the calculation of syndromes is straightforward, so there is no need to analyze the

first step. The possible error location numbers consists of the elements $\alpha^0, \alpha^1, \dots, \alpha^{b^j n - 1}$. The elementary symmetric function $\sigma_1, \sigma_2, \dots, \sigma_v$ are defined as coefficients of polynomial

$$(x^{\frac{a}{b^j}} - x_1)(x^{\frac{a}{b^j}} - x_2) \dots (x^{\frac{a}{b^j}} - x_v) = (x^{\frac{a}{b^j}})^v + \sigma_1(x^{\frac{a}{b^j}})^{v-1} + \dots + \sigma_{v-1}(x^{\frac{a}{b^j}}) + \sigma_v, \quad (6.3)$$

where v represents number of errors. These functions are obtained by finding a solution $\sigma_1, \sigma_2, \dots, \sigma_v$ with minimum possible v to the following set of linear equations over $\mathbb{Z}_{2^m}[x; \frac{a}{b^j} \mathbb{N}_0]$.

$$S_{\lambda+v} + S_{\lambda+v-1}\sigma_1 + \dots + S_{\lambda-1}\sigma_{v-1} + S_{\lambda}\sigma_v = 0, \quad \lambda = 1, 2, \dots, 2t - v, \quad (6.4)$$

where $S_1, S_2, S_3, \dots, S_{2t}$ is the sequence of syndromes. The solution to (6.4) is obtained by modified Berlekamp-Massey algorithm which holds for commutative rings with identity. It is an iterative algorithm because at \mathbf{n} th step, we have to determine $l_{\mathbf{n}}$ values $\sigma_i^{(\mathbf{n})}$ such that the following $\mathbf{n} - l_{\mathbf{n}}$ equations hold with $l_{\mathbf{n}}$ as small as possible and $\sigma_0^{(\mathbf{n})} = 1$.

$$S_{\mathbf{n}}\sigma_0^{(\mathbf{n})} + S_{\mathbf{n}-1}\sigma_1^{(\mathbf{n})} + \dots + S_{\mathbf{n}-l_{\mathbf{n}}}\sigma_{l_{\mathbf{n}}}^{(\mathbf{n})} = 0 \quad (6.5)$$

$$S_{\mathbf{n}-1}\sigma_0^{(\mathbf{n})} + S_{\mathbf{n}-2}\sigma_1^{(\mathbf{n})} + \dots + S_{\mathbf{n}-l_{\mathbf{n}}-1}\sigma_{l_{\mathbf{n}}}^{(\mathbf{n})} = 0 \quad (6.6)$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad (6.7)$$

$$S_{l_{\mathbf{n}}+1}\sigma_0^{(\mathbf{n})} + S_{l_{\mathbf{n}}}\sigma_1^{(\mathbf{n})} + \dots + S_1\sigma_{l_{\mathbf{n}}}^{(\mathbf{n})} = 0. \quad (6.8)$$

At \mathbf{n} th stage the solution is represented by the generalized polynomial $\sigma^{(\mathbf{n})}(x^{\frac{a}{b^j}}) = \sigma_0^{(\mathbf{n})} + \sigma_1^{(\mathbf{n})}(x^{\frac{a}{b^j}}) + \dots + \sigma_{l_{\mathbf{n}}}^{(\mathbf{n})}(x^{\frac{a}{b^j}})^{l_{\mathbf{n}}}$ and \mathbf{n} th discrepancy ($d_{\mathbf{n}}$) is defined by $d_{\mathbf{n}} = S_{\mathbf{n}+1}\sigma_0^{(\mathbf{n})} + S_{\mathbf{n}}\sigma_1^{(\mathbf{n})} + \dots + S_{\mathbf{n}+1-l_{\mathbf{n}}}\sigma_{l_{\mathbf{n}}}^{(\mathbf{n})}$.

The input of algorithm is the values of syndromes S_1, S_2, \dots, S_{2t} which belong to $GR(2^m, b^j s)$ and its output is the set of values $\sigma_1, \sigma_2, \dots, \sigma_v$, such that the equations in (6.4) hold for minimum value of v . Some initial conditions to start an algorithm are

$$\begin{aligned} \sigma^{(-1)}(x^{\frac{a}{b^j}}) &= 1 & l_{-1} &= 0 & d_{-1} &= 1 \\ \sigma^{(0)}(x^{\frac{a}{b^j}}) &= 1 & l_0 &= 0 & d_0 &= S_1 \end{aligned} .$$

Then the further steps are performed as follow.

1. $0 \rightarrow \mathbf{n}$.
2. If $d_{\mathbf{n}} = 0$, then $\sigma^{(\mathbf{n})}(x^{\frac{a}{b^j}}) \rightarrow \sigma^{(\mathbf{n}+1)}(x^{\frac{a}{b^j}})$ and $l_{\mathbf{n}} \rightarrow l_{\mathbf{n}+1}$ and go to 5.
3. If $d_{\mathbf{n}} \neq 0$, then find an $m' \leq \mathbf{n} - 1$ such that $d_{\mathbf{n}} - yd_{m'} = 0$ has a solution in y and $m' - l_{\mathbf{n}}$ has the largest value. Then $\sigma^{(\mathbf{n})}(x^{\frac{a}{b^j}}) - y \cdot (x^{\frac{a}{b^j}})^{\mathbf{n}-m'} \cdot \sigma^{(m')}(x^{\frac{a}{b^j}}) \rightarrow \sigma^{(\mathbf{n}+1)}(x^{\frac{a}{b^j}})$ and $\max \{l_{\mathbf{n}}, l_{m'} + \mathbf{n} - m'\} \rightarrow l_{\mathbf{n}+1}$.
4. If $l_{\mathbf{n}+1} = \max \{l_{\mathbf{n}}, \mathbf{n} + 1 - l_{\mathbf{n}}\}$ then go to step 5, else find the solution $D^{(\mathbf{n}+1)}(x^{\frac{a}{b^j}})$ with minimum degree l in the range $\max \{l_{\mathbf{n}}, \mathbf{n} + 1 - l_{\mathbf{n}}\} \leq l \leq l_{\mathbf{n}+1}$ such that $\sigma^{(m')}(x^{\frac{a}{b^j}})$ defined by $(x^{\frac{a}{b^j}})^{\mathbf{n}-m'} \cdot \sigma^{(m')}(x^{\frac{a}{b^j}}) = D^{(\mathbf{n}+1)}(x^{\frac{a}{b^j}}) - \sigma^{(\mathbf{n})}(x^{\frac{a}{b^j}})$ is a solution to first m' power sums, $d_{m'} = -d_{\mathbf{n}}$, with $\sigma_0^{(m')}$ a zero divisor in $GR(2^m, b^j s)$. If such a solution is found, then $D^{(\mathbf{n}+1)}(x^{\frac{a}{b^j}}) \rightarrow \sigma^{(\mathbf{n}+1)}(x^{\frac{a}{b^j}})$ and $l \rightarrow l_{\mathbf{n}+1}$.
5. If $\mathbf{n} < 2t - 1$, then $S_{\mathbf{n}+2} + S_{\mathbf{n}+1}\sigma_1^{(\mathbf{n}+1)} + \dots + S_{\mathbf{n}+2-l_{\mathbf{n}+1}}\sigma_{l_{\mathbf{n}+1}}^{(\mathbf{n}+1)} \rightarrow d_{\mathbf{n}+1}$.
6. For $\mathbf{n} + 1 \rightarrow \mathbf{n}$, if $\mathbf{n} < 2t$, then go to step 2, else stop.

The coefficients $\sigma_1^{2t}, \sigma_2^{2t}, \dots, \sigma_{l_{\mathbf{n}}}^{2t}$ of $\sigma^{(2t)}(x^{\frac{a}{b^j}})$ satisfy equation (6.4).

In the next step, the calculation of error location numbers requires one step on $GF(2, b^j s)$ because in $GR(2^m, b^j s)$ the solution to (6.4) is not unique and the reciprocal to the polynomial $\sigma^{2t}(z^{\frac{a}{b^j}})$ denoted by $\rho(z^{\frac{a}{b^j}})$ may not be right error locator polynomial

$$(z^{\frac{a}{b^j}} - x_1)(z^{\frac{a}{b^j}} - x_2) \dots (z^{\frac{a}{b^j}} - x_v), \quad (6.9)$$

where $x_i = \alpha^{j'}$ are correct error location numbers, j' is an integer such that $1 \leq j' \leq b^j n - 1$ and it indicates the position of i th errors in codeword. Error location numbers are calculated by first computing the roots z_1, z_2, \dots, z_v of $\rho(z^{\frac{a}{b^j}})$ and then selecting x_1, \dots, x_v among $x_i = \alpha^{j'}$ such that $x_i - z_i$ are zero divisors in the Galois ring $GR(2^m, b^j s)$, these x_1, \dots, x_v are correct error location numbers.

In last step, error magnitudes y_1, y_2, \dots, y_v are calculated as

$$y_j = \frac{\sum_{l=0}^{v-1} \sigma_{j,l} S_{v-l}}{\sum_{l=0}^{v-1} \sigma_{j,l} x_j^{v-l}}, \quad j = 1, 2, \dots, v, \quad (6.10)$$

and the coefficients $\sigma_{j,l}$ are given by $\sigma_{j,i} = \sigma_i + x_j \cdot \sigma_{j,i-1}$, where $i = 0, 1, \dots, v - 1$, starting with

$\sigma_0 = \sigma_{j,0} = 1$. Here the important point is to show that the denominator in the expression of y_j must be invertible, i.e., unit in $GR(2^m, b^j s)$. From [13], the denominator is given by product

$$x_j \prod_{i=1, i \neq j}^{v-1} (x_j - x_i),$$

where each factor is of the form $\alpha^i - \alpha^j$ for $0 \leq i \neq j \leq b^j n - 1$, and they all are units in $GR(2^m, b^j s)$.

In the following example, we first apply this decoding algorithm on non-primitive BCH code \mathcal{C}_{45} over \mathbb{Z}_4 and then over \mathbb{Z}_2 .

Example 89 Consider $(15, 7)$ BCH code in $\mathbb{Z}_4[x; 2\mathbb{N}_0]_{15}$ with generator polynomial

$$g'(x^2) = (x^2)^8 + (x^2)^7 + 3(x^2)^6 + 2(x^2)^5 + 3(x^2)^4 + 2(x^2)^3 + 2(x^2)^2 + 1.$$

Now, convert it to generating polynomial of $(45, 7)$ BCH code in $\mathbb{Z}_4[x; \frac{2}{3}\mathbb{N}_0]_{45}$ which is:

$$\begin{aligned} g'(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{37} + 3(x^{\frac{2}{3}})^{36} + 2(x^{\frac{2}{3}})^{35} + 3(x^{\frac{2}{3}})^{34} + 2(x^{\frac{2}{3}})^{33} + \\ &2(x^{\frac{2}{3}})^{32} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{22} + 3(x^{\frac{2}{3}})^{21} + 2(x^{\frac{2}{3}})^{20} + \\ &3(x^{\frac{2}{3}})^{19} + 2(x^{\frac{2}{3}})^{18} + 2(x^{\frac{2}{3}})^{17} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^7 + \\ &3(x^{\frac{2}{3}})^6 + 2(x^{\frac{2}{3}})^5 + 3(x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^3 + 2(x^{\frac{2}{3}})^2 + 1. \end{aligned}$$

Its design distance $d' = 15$, so the error correction capability t' equals to 7. For the sake of convenience we correct only 2 errors here. Suppose that the received word is

$$\begin{aligned} \mathbf{r}(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^2 + 2(x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^5 + 2(x^{\frac{2}{3}})^7 + 3(x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^9 + \\ &(x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^{17} + 2(x^{\frac{2}{3}})^{19} + 2(x^{\frac{2}{3}})^{20} + 3(x^{\frac{2}{3}})^{21} + 2(x^{\frac{2}{3}})^{22} + \\ &3(x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{25} + (x^{\frac{2}{3}})^{32} + 2(x^{\frac{2}{3}})^{34} + 2(x^{\frac{2}{3}})^{35} + 3(x^{\frac{2}{3}})^{36} + \\ &2(x^{\frac{2}{3}})^{37} + 3(x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{39} + (x^{\frac{2}{3}})^{40}. \end{aligned}$$

The syndromes from the received word are given by

$$S_1 = 2u^6 + 3, S_2 = 2u^3 + 1, S_3 = u^6 + 3u^3 + 1, S_4 = 2u^6 + 1.$$

By applying modified Berlekamp-Massey algorithm, we have

Table 26: Values of decoding steps

\mathbf{n}	$\sigma^{(\mathbf{n})}(z^{\frac{2}{3}})$	$d_{\mathbf{n}}$	$l_{\mathbf{n}}$	$\mathbf{n} - l_{\mathbf{n}}$
-1	1	1	0	-1
0	1	S_1	0	0
1	$1 + (2u^6 + 1) z^{\frac{2}{3}}$	$2u^3$	1	0
2	$1 + (2u^6 + 2u^3 + 1) z^{\frac{2}{3}}$	$3u^6 + 3u^3 + 2$	1	1
3	$1 + (2u^6 + 2u^3 + 1) z^{\frac{2}{3}} + (2u^9 + 3u^6 + u^3) \left(z^{\frac{2}{3}}\right)^2$	0	2	1
4	$1 + (2u^6 + 2u^3 + 1) z^{\frac{2}{3}} + (2u^9 + 3u^6 + u^3) \left(z^{\frac{2}{3}}\right)^2$	—	2	2

The roots of $\rho(z^{\frac{2}{3}}) = (z^{\frac{2}{3}})^2 + (2u^6 + 2u^3 + 1) (z^{\frac{2}{3}}) + (2u^9 + 3u^6 + u^3)$ (reciprocal of $\sigma^{(4)}(z^{\frac{2}{3}})$) are $z_1 = 3u^3 + 3$ and $z_2 = 2u^6 + 3u^3$. Among the elements $\alpha^0, \alpha^1, \alpha^2, \alpha^3, \dots, \alpha^{45}$, we have $x_1 = \alpha^6$ and $x_2 = \alpha^{24}$ is such that $x_1 - z_1 = 0$ and $x_2 - z_2 = 0$. It indicates that two errors have occurred, one at position 6 and other at position 24 in the codeword. Correct elementary symmetric functions are obtained as

$$(x^{\frac{2}{3}} - x_1)(x^{\frac{2}{3}} - x_2) = (x^{\frac{2}{3}})^2 + (2u^6 + 2u^3 + 1) (x^{\frac{2}{3}}) + (2u^9 + 3u^6 + u^3)$$

Thus, $\sigma_1 = 2u^6 + 2u^3 + 1$ and $\sigma_2 = 2u^9 + 3u^6 + u^3$. Now, by using equation (6.10), error magnitudes are $y_1 = 1$ and $y_2 = 3$, hence the error vector is $\mathbf{e}(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^6 + 3(x^{\frac{2}{3}})^{24}$. Therefore the codeword is

$$\begin{aligned} \mathbf{c}(x^{\frac{2}{3}}) = & (x^{\frac{2}{3}})^2 + 2(x^{\frac{2}{3}})^4 + 2(x^{\frac{2}{3}})^5 + 3(x^{\frac{2}{3}})^6 + 2(x^{\frac{2}{3}})^7 + 3(x^{\frac{2}{3}})^8 + \\ & (x^{\frac{2}{3}})^9 + (x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^{17} + 2(x^{\frac{2}{3}})^{19} + 2(x^{\frac{2}{3}})^{20} + 3(x^{\frac{2}{3}})^{21} + \\ & 2(x^{\frac{2}{3}})^{22} + 3(x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{24} + (x^{\frac{2}{3}})^{25} + (x^{\frac{2}{3}})^{32} + 2(x^{\frac{2}{3}})^{34} + \\ & 2(x^{\frac{2}{3}})^{35} + 3(x^{\frac{2}{3}})^{36} + 2(x^{\frac{2}{3}})^{37} + 3(x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{39} + (x^{\frac{2}{3}})^{40}. \end{aligned}$$

Now, by reducing the coefficients of above generator polynomial modulo 2, we obtain generator polynomial of $(45, 7)$ primitive BCH code in $\mathbb{Z}_2[x; \frac{2}{3}\mathbb{N}_0]$, which is

$$\begin{aligned} g(x^{\frac{2}{3}}) = & (x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{37} + (x^{\frac{2}{3}})^{36} + (x^{\frac{2}{3}})^{34} + (x^{\frac{2}{3}})^{30} + (x^{\frac{2}{3}})^{23} + \\ & (x^{\frac{2}{3}})^{22} + (x^{\frac{2}{3}})^{21} + (x^{\frac{2}{3}})^{19} + (x^{\frac{2}{3}})^{15} + (x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^7 + \\ & (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^4 + 1. \end{aligned}$$

Suppose that the received word is

$$\begin{aligned} \mathbf{r}(x^{\frac{2}{3}}) = & (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^9 + (x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^{17} + (x^{\frac{2}{3}})^{21} + \\ & (x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{25} + (x^{\frac{2}{3}})^{32} + (x^{\frac{2}{3}})^{36} + (x^{\frac{2}{3}})^{38} + \\ & (x^{\frac{2}{3}})^{39} + (x^{\frac{2}{3}})^{40}. \end{aligned}$$

The syndromes are given by $S_1 = u^{45}$, $S_2 = u^{45}$, $S_3 = u^{15}$, $S_4 = u^{45}$. Using modified Berlekamp-Massey algorithm, we have

Table 27: Values of decoding steps

\mathbf{n}	$\sigma^{(\mathbf{n})}(z^{\frac{2}{3}})$	$d_{\mathbf{n}}$	$l_{\mathbf{n}}$	$\mathbf{n} - l_{\mathbf{n}}$
-1	1	1	0	-1
0	1	S_1	0	0
1	$1 + (u^{45}) z^{\frac{2}{3}}$	0	1	0
2	$1 + (u^{45}) z^{\frac{2}{3}}$	u^{30}	1	1
3	$1 + (u^{45}) z^{\frac{2}{3}} + (u^{30}) (z^{\frac{2}{3}})^2$	0	2	1
4	$1 + (u^{45}) z^{\frac{2}{3}} + (u^{30}) (z^{\frac{2}{3}})^2$	-	2	2

The roots of $\rho(z^{\frac{2}{3}}) = (z^{\frac{2}{3}})^2 + (u^{45}) z^{\frac{2}{3}} + u^{30}$ (reciprocal of $\sigma^{(4)}(z^{\frac{2}{3}})$) are $z_1 = u^6$ and $z_2 = 1 + u^6$. Among the elements $u^0, u^1, u^2, u^3, \dots, u^{45}$, we have $x_1 = u^6$ and $x_2 = u^{24}$ is such that $x_1 - z_1 = 0$ and $x_2 - z_2 = 0$. It indicates that two errors have occurred, one at position 6 and other at position 24 in the codeword. Correct elementary symmetric functions are obtained from equation

$$(x^{\frac{2}{3}} - x_1)(x^{\frac{2}{3}} - x_2) = (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}}) + (1 + u^3 + u^6).$$

Thus, $\sigma_1 = 1$ and $\sigma_2 = 1 + u^3 + u^6$. Now, the error magnitudes are $y_1 = 1$ and $y_2 = 1$, hence the error vector is $\mathbf{e} = (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^{24}$ and the codeword is

$$\begin{aligned} \mathbf{c}(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^8 + (x^{\frac{2}{3}})^9 + (x^{\frac{2}{3}})^{10} + (x^{\frac{2}{3}})^{17} + \\ &\quad (x^{\frac{2}{3}})^{21} + (x^{\frac{2}{3}})^{23} + (x^{\frac{2}{3}})^{24} + (x^{\frac{2}{3}})^{25} + (x^{\frac{2}{3}})^{32} + \\ &\quad (x^{\frac{2}{3}})^{36} + (x^{\frac{2}{3}})^{38} + (x^{\frac{2}{3}})^{39} + (x^{\frac{2}{3}})^{40}. \end{aligned}$$

This gives message polynomial $\mathbf{m}(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^2 + (x^{\frac{2}{3}})^6$.

Corresponding to 2-error correcting $(15, 7)$ primitive BCH code in $\mathbb{Z}_2[x; \frac{2}{3}\mathbb{N}_0]$, we have 7-error correcting $(45, 7)$ non-primitive BCH code in $\mathbb{Z}_4[x; \frac{2}{3}\mathbb{N}_0]$ and $\mathbb{Z}_2[x; \frac{2}{3}\mathbb{N}_0]$.

During communication, the codewords of any of the sequences $\{\mathcal{C}_{bjn}\}_{j \geq 0}$ and $\{\mathcal{C}'_{bjn}\}_{j \geq 0}$ of BCH codes can be decoded by using modified Berlekamp-Massey algorithm. In decoder we use this single algorithm to deal all four types of codewords.

Remark 90 For b^jn length BCH codes with same code dimension it requires more time and lengthy calculations with increasing values of j . A method through which the decoding of b^jn length codes is done with the help of n length BCH codes. If in each pattern of b^jn length BCH code the errors are not more than the error correction capability of n length BCH codes, then with the help of generator polynomial of n length BCH code we decode each pattern separately and in last combine them to get b^jn length codeword as shown in following example. Consider transmitted vector

$$\mathbf{c} = 001022323110000001022323110000001022323110000,$$

of BCH code $(45, 7)$ in $\mathbb{Z}_4[x; \frac{2}{3}\mathbb{N}_0]_{45}$ as taken in example 89 and received vector is

$$\mathbf{r} = 001022023010000001002323110030001022023100000.$$

By splitting the received vector into patterns of length 15, we have

$$\begin{aligned}\mathbf{r}_1 &= 001022023010000, \mathbf{r}_2 = 001002323110030 \\ \mathbf{r}_3 &= 001022023100000.\end{aligned}$$

As number of errors in each pattern are two, so, we decode them with the help of generator polynomial of 15 length BCH code over $GR(2^2, 4)$ and in last combine them to get the transmitted vector.

However, if the number of errors in some pattern of $b^j n$ length BCH code exceeds the error correction capability of n length BCH codes, then we have no way to deal it with the help of n length BCH code.

Chapter 7

Applications

Cyclic codes and particularly BCH codes have a wide range of applications in information theory, wireless communication and cryptography. Recently application in the formation of DNA sequences is considered as codewords of BCH codes over the field \mathbb{F}_4 . This chapter is divided in three sections. Section 1 discuss the application of BCH codes over the field \mathbb{F}_2 in cognitive radio, Section 2 discusses the application of BCH codes over the field \mathbb{F}_4 in the formation of DNA sequences and Section 4 discusses the application of BCH codes over finite rings and finite fields instantaneously in data transmission.

7.1 Application in cognitive radio

The cognitive radios constantly try to regulate modulation scheme, bandwidth, code rate, power, and carrier frequency in an exertion to consume unused spectrum and elude interference to the primary user. A clever design of error correcting codes might offer generous gains in interweave multiple transmission cognitive radio arrangement. We have introduced a novel coding scheme to manage data for primary as well as secondary user by including similar (or dissimilar) binary BCH codes and a corresponding ascending sequence of embedded distinct binary BCH codes with increasing error correction capability and varying code rates. In a similar approach to an interweave model, initially the information capacity is utilized by the primary users, while, in the existence of spectrum holes, the secondary user uses binary BCH codes to competently utilize the vacant spectrum.

7.1.1 Bandwidth limitations

Let S_i be the signal set M_i be the number of signals in the signal set. Suppose $v^{(T)} = (v_0^{(T)}, \dots, v_{n_i-1}^{(T)}) \in \mathbb{F}_2^n$ is the codeword of an (n_i, k_i) -code against a message $u^{(T)} = (u_0^{(T)}, \dots, u_{k_i-1}^{(T)}) \in \mathbb{F}_2^{k_i}$ at time T and we divide each $v^{(T)}$ into n_i/m_i blocks, where $m_i = \log_2 M$, $M = 2^{m_i}$. Then modulation is a map $M : \mathbb{F}_2^m \rightarrow S_i$ defined as $s_i^{(T)} = s_i(v^{(T)})$, where $s_i^{(T)} \in S_i$ and S_i is a subset of N -dimensional real Euclidean space, that is, $S_i \subset \mathbb{R}^N$ [23, Chapter 7].

Following [32], the bandwidth required for an (n, k) code is $W = \frac{R_u}{m}(\frac{1}{R})$, where $m = \log_2 M$, R_u is the source data (transmission) rate and $R = \frac{k}{n}$, the code rate.

The bandwidth may be maximize and minimize, depends upon the minimum and the maximum value of the ratio $n/k = 1/R$ and the value of m bits for the selection of modulation scheme for different modulation types. These bits may be minimum and maximum for maximum and minimum bandwidth. It can be seen as; $W_{\max} = \frac{R_u}{m_{\min}}(\frac{1}{R})_{\max}$ and $W_{\min} = \frac{R_u}{m_{\max}}(\frac{1}{R})_{\min}$. Thus there are possibilities; (i) m is fixed but $\frac{1}{R}$ is varying, (ii) m and $\frac{1}{R}$ both are varying.

For Cognitive radio multiple forward transformation under the interweave model we may get spectrum corresponding to the given set of sequences $\{C_{bjn_i}^j\}_{j=1}^{j_0}$, $1 \leq i \leq i_0$ of binary cyclic codes for data transfer of the primary users. Now, the setup allow the secondary users having the binary BCH code $C_{n_i}^0$, $1 \leq i \leq i_0$ mod for their data transfer. Accordingly the secondary users obtain high speed data transfer as compare to its own scheme of the BCH code $C_{n_i}^0$. Furthermore since for each $1 \leq i \leq i_0$ there are sets of embeddings $C_{n_i}^0 \hookrightarrow C_{b^1 n_i}^1 \hookrightarrow \dots \hookrightarrow C_{b^{j_0} n_i}^{j_0}$ of binary BCH codes of the sequences $\{C_{bjn_i}^j\}_{j=1}^{j_0}$ and the binary BCH code $C_{n_i}^0$ is embedded in each of binary cyclic codes $C_{bjn_i}^j$ for $1 \leq j \leq j_0$. It is also observed that corresponding to the code rate $R_{n_i}^0 = \frac{k_i}{n_i}$ of binary BCH code $C_{n_i}^0$, the code rate of binary BCH code $C_{bjn_i}^j$ is $R_{bjn_i}^j = \frac{bjn_i - bj r_i}{bj n_i}$, for each $1 \leq j \leq j_0$. Consequently $R_{n_i}^0 \leq R_{b^1 n_i}^1 \leq R_{b^2 n_i}^2 \leq \dots \leq R_{bjn_i}^j \leq \dots$ and thus $\dots \leq \frac{1}{R_{bjn_i}^j} \leq \dots \leq \frac{1}{R_{b^2 n_i}^2} \leq \frac{1}{R_{b^1 n_i}^1} \leq \frac{1}{R_{n_i}^0}$. This implies $\dots \leq W_{bjn_i}^j = \frac{R_u}{m_i}(\frac{1}{R_{bjn_i}^j}) \leq \dots \leq W_{b^1 n_i}^1 = \frac{R_u}{m_i}(\frac{1}{R_{b^1 n_i}^1}) \leq W_{n_i}^0 = \frac{R_u}{m_i}(\frac{1}{R_{n_i}^0})$. Thus, if we transmit data through any of the code in the sequence $\{C_{bjn_i}^j\}_{j=1}^{j_0}$, the bandwidth $W_{bjn_i}^j$ for each $j \geq 1$ will be lesser the bandwidth $W_{n_i}^0$ required for data transmitted through the binary BCH code $C_{n_i}^0$.

7.1.2 Multiple forward transmission through embedded BCH codes

The secondary user has an opportunistic access to the spectrum, when the primary user is absent and withdraw when the primary user wishes to transmit once another time, this is due to the interweave model. Accordingly the codes constructed in Chapter 3, could deliver an excellent pattern for wireless communication in which interference issue is controlled amicably. We offer a multiple forward transmission model for Cognitive radio based on error correcting codes which guarantees the noninterference among the users.

A plan of the multiple forward transformation model is offered bellow.

It is supposed that a primary users family $\{\mathcal{P}_{bjn_i}^j\}_{j=0}^{j_0}$ from the set $\{\{\mathcal{P}_{bjn_i}^j\}_{j=0}^{j_0} : 1 \leq i \leq i_0\}$, use the family $\{C_{bjn_i}^j\}_{j=0}^{j_0}$ of binary BCH codes from $\{\{C_{bjn_i}^j\}_{j=0}^{j_0} : 1 \leq i \leq i_0\}$ for its data transmission. For each $i \in \{1, 2, \dots, i_0\}$ there are embeddings $C_{n_i}^0 \hookrightarrow C_{b^1n_i}^1 \dots \hookrightarrow C_{b^{j_0}n_i}^{j_0}$ of binary BCH codes.

The binary BCH codes in the sequence $\{C_{bjn_i}^j\}_{j=0}^{j_0}$ are used for data transmission of the sequence $\{\mathcal{P}_{bjn_i}^j\}_{j=1}^{j_0}$ of primary users with corresponding bandwidths $\{W_{bjn_i}^j\}_{j=0}^{j_0}$ such that $W_{b^{j_0}n_i}^{j_0} \leq \dots \leq W_{b^1n_i}^1 \leq W_{n_i}^0$ and the total bandwidth $\sum_{j=0}^{j_0} W_{bjn_i}^j$ is required for simultaneous transmission.

Whenever all users $\{\mathcal{P}_{bjn_i}^j\}_{j=0}^{j_0}$ transmitting their data at a glance considered to be the primary users. However, any of the user $\mathcal{P}_{bj'n_i}^{j'}$, where $j' \in \{1, 2, \dots, j_0\}$, enter as a secondary user and opportunistically can use any of path of the sequence $\{\mathcal{P}_{bjn_i}^j\}_{j=j'}^{j_0}$ of primary users whenever any of them is not using its allotted bandwidth. Here it is noticed that the data of the secondary user $\mathcal{P}_{bj'n_i}^{j'}$ is configured with the binary BCH code $C_{bj'n_i}^{j'}$ and it requires bandwidth higher than any of the bandwidth required for the data of any primary user of the sequence $\{\mathcal{P}_{bjn_i}^j\}_{j=j'}^{j_0}$. Consequently with high code rate, improved error correction capability and with less bandwidth secondary user $\mathcal{P}_{bj'n_i}^{j'}$ can transmit its data. Thus any primary user $\mathcal{P}_{b^ln_i}^l$ of the sequence $\{\mathcal{P}_{bjn_i}^j\}_{j=0}^{j_0}$ can change its status as a secondary user $\mathcal{S}_{b^ln_i}^l$ whenever any of the user $\mathcal{P}_{b^mn_i}^m$ with $l < m$, is not using its path.

Functioning of the model

Notions

$0 \leq j \leq b^{j_0}n_i$. And $1 \leq i \leq i_0$.

$\mathcal{P}_{n_i}^0$: Primary user corresponding to the binary primitive BCH code $C_{n_i}^0$.

$\mathcal{P}_{bjn_i}^j$: j th Primary user corresponding to the binary non primitive BCH code $C_{bjn_i}^j$.

m_i^j : information symbols for j th user.

E_i^j : j th encoder for m_i^j .

$\mathcal{M}_{\mathcal{P}_i^j}$: Modulation for E^j

$W_{bjn_i}^j$: Bandwidth required for user $\mathcal{P}_{bjn_i}^j$ for each j .

$D\mathcal{M}_{\mathcal{P}_i^j}$: j th Demodulation

D_i^j : j th decoder

The data of \mathcal{P}_i^j , for each j , for each i , is modulated through $\mathcal{M}_{\mathcal{P}_i^j}$, where $\mathcal{M}_{\mathcal{P}_i^j}$ is a modulation map, i.e., $\mathcal{M}_{\mathcal{P}_i^j} : \mathbb{F}_q^m \rightarrow S_{\mathcal{P}_i^j}$, where $S_{\mathcal{P}_i^j}$ is the signal set, $j_{\mathcal{P}_i^j}$ is the number of signals in the signal sets $S_{\mathcal{P}_i^j}$. However for $q = 2$, $j_{\mathcal{P}_i^j} = 2^{m_i}$, m_i is a positive integer.

Multiple Cognitive Radio Forward Transmission Model (MCRFTM) or Sheet assortments

The functionality of **MCRFTM** is as follows. We call S_i , $1 \leq i \leq i_0$, a sheet and $\prod_{i=1}^{i_0} S_i$, the sheet assortments.

Table 28: Values of S_i , $1 \leq i \leq i_0$

S_1 :	$C_{n_1}^0 \hookrightarrow C_{bn_1}^1 \hookrightarrow \dots \hookrightarrow C_{bj_0 n_1}^{j_0}$
S_2 :	$C_{n_2}^0 \hookrightarrow C_{bn_2}^1 \hookrightarrow \dots \hookrightarrow C_{bj_0 n_2}^{j_0}$
	.
	.
S_{i_0} :	$C_{n_{i_0}}^0 \hookrightarrow C_{bn_{i_0}}^1 \hookrightarrow \dots \hookrightarrow C_{bj_0 n_{i_0}}^{j_0}$
	.
	.
	.

Table 29: CRFTM

$P_{n_i}^0$	$P_{bn_i}^1$	$P_{b^2n_i}^2 \dots$	$P_{b^l n_i}^l \dots$	$P_{b^m n_i}^m \dots$	$P_{b^{j_0} n_i}^{j_0}$
\hookrightarrow	\hookrightarrow	\hookrightarrow	\hookrightarrow	$S_{n_i}^l = P_{n_i}^l \hookrightarrow \downarrow$	\downarrow
\downarrow	\downarrow	\downarrow	\downarrow		
m_i^0	m_i^1	m_i^2	m_i^l	$m_i^0 \hookrightarrow m_i^m$	$m_i^{j_0}$
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
E_i^0	E_i^1	E_i^2	E_i^l	$E_i^0 \hookrightarrow E_i^m$	$E_i^{j_0}$
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
$M_{\mathcal{P}_i^0}$	$M_{\mathcal{P}_i^1}$	$M_{\mathcal{P}_i^2}$	$M_{\mathcal{P}_i^l}$	$M_{S_i^l} = M_{\mathcal{P}_i^l} \leftrightarrow M_{\mathcal{P}_i^m}$	$M_{\mathcal{P}_i^{j_0}}$
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
$W_{n_i}^0$	$W_{b^1 n_i}^1$	$W_{b^2 n_i}^2$	$W_{b^3 n_i}^l$	$W_{b^m n_i}^m$	$W_{b^{j_0} n_i}^{j_0}$
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
$DM_{\mathcal{P}_i^0}$	$DM_{\mathcal{P}_i^1}$	$DM_{\mathcal{P}_i^2}$	$DM_{\mathcal{P}_i^l}$	$DM_{S_i^l} = DM_{\mathcal{P}_i^l} \leftrightarrow DM_{\mathcal{P}_i^m}$	$DM_{\mathcal{P}_i^{j_0}}$
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
D_i^0	D_i^1	D_i^2	D_i^l	$D_i^l \hookrightarrow D_i^m$	$D_i^{j_0}$
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
P_{i,n_i}^0	P_{i,bn_i}^1	$P_{i,b^2n_i}^2$	$P_{i,b^3n_i}^l \dots$	$S_{i,b^l n_i}^l = P_{i,b^l n_i}^l \leftrightarrow P_{i,b^m n_i}^m \dots$	$P_{i,b^{j_0} n_i}^{j_0}$

Transmission steps for an arbitrary page of the book

$$0 \leq j \leq j_0, 1 \leq i \leq i_0$$

I. All users are Primary users

1. Data of the \mathcal{P}_i^j , for each j , users transform into the set m_i^j of message bits.
2. For each j , the set m_i^j of message bits encoded through encoder E_i^j .
3. For each j , the set E_i^j of encoded messages modulated through $\mathcal{M}_{\mathcal{P}_i^j}$
4. For each j , the set $\mathcal{M}_{\mathcal{P}_i^j}$ of modulated codewords passing through the channel having bandwidth W_i^j .
5. For each j , the corresponding transmitted signals of $\mathcal{M}_{\mathcal{P}_i^j}$ are demodulated.
6. For each j , the received signals corresponding to $\mathcal{M}_{\mathcal{P}_i^j}$ are decoded through decoder D_i^j .

7. The end of whole process is the destination of data of all users.

II. All users are not Primary users

Almost all steps of data transmission are same as I. However, the user $\mathcal{P}_{b^l n_i}^l$ enter as a secondary user and opportunistically can use any of the path of the sequence $\{\mathcal{P}_{b^j n_i}^j\}_{j=l}^{j_0}$ of primary users whenever any of them is not using its allotted bandwidth. For instance, if the primary user $\mathcal{P}_{b^m n_i}^m$, where $l \leq m$, is not in, then the user $\mathcal{P}_{b^l n_i}^l$ transmitted its data configured by the binary BCH code $C_{b^l n_i}^l$ through the binary BCH code $C_{b^m n_i}^m$ used for data of primary user $\mathcal{P}_{b^m n_i}^m$ and it is now considered as the secondary user $\mathcal{S}_{b^l n_i}^l$.

A Similar Multiple Cognitive Radio Forward Transmission Model (SMCRFTM) or repeated pages of the book

This is a particular case of **MCRFTM** and it works as: For any fixed $i_l \in \{1, 2, \dots, i_0\}$ one can choose the transmission scheme of multiple data transmission in such a way that there is a binary BCH code of length n_{i_l} with corresponding sequence $\{C_{b^j n_{i_l}}^j\}_{j=0}^{j_0}$ of binary BCH codes. Furthermore all of the i_0 transmissions are consisting on this same sequence of binary BCH codes.

Table 30

$S_{i_l} : C_{n_{i_l}}^0 \hookrightarrow C_{b n_{i_l}}^1 \hookrightarrow \dots \hookrightarrow C_{b^{j_0} n_{i_l}}^{j_0}$
$S_{i_l} : C_{n_{i_l}}^0 \hookrightarrow C_{b n_{i_l}}^1 \hookrightarrow \dots \hookrightarrow C_{b^{j_0} n_{i_l}}^{j_0}$
.
.
.
$S_{i_l} : C_{n_{i_l}}^0 \hookrightarrow C_{b n_{i_l}}^1 \hookrightarrow \dots \hookrightarrow C_{b^{j_0} n_{i_l}}^{j_0}$
.
.
.

A Constant Multiple Cognitive Radio Forward and Backward Transmission Model (CMCRFBTM)

This is a case of **MCRFTM** and it works as: For any fixed $i \in \{1, 2, \dots, i_0\}$ one can choose

the transmission scheme of multiple data transmission in such a way that there is a fixed binary BCH code C_n^0 of length n with corresponding sequence $\{C_{n_i}^j\}_{j=1}^{j_0}$ of binary BCH codes, where $C_{n_i}^j = C_{n_i}^0$ for each j . Furthermore all of the transmissions are consisting on this same sequence of binary BCH codes.

Table 31

S_1 :	$C_{n_1}^0 = C_{n_1}^1 = \dots = C_{n_1}^{j_0}$
S_2 :	$C_{n_2}^0 = C_{n_2}^1 = \dots = C_{n_2}^{j_0}$
	.
	.
S_i :	$C_{n_i}^0 = C_{n_i}^1 = \dots = C_{n_i}^{j_0}$
	.
	.
	.

Almost all steps of data transmission are same as I. However, any user $\mathcal{P}_{n_i}^l$ can enter as a secondary user and opportunistically can use any of the path of the sequence $\{\mathcal{P}_{n_i}^j\}_{j=l}^{j_0}$ of primary users whenever any of them is not using its allotted bandwidth.

Illustration of MCRFTM

The irreducible non-primitive polynomials $p(x^{\frac{2}{3}}) = 1 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}})^{12} \in \mathbb{F}_2[x; \frac{2}{3}\mathbb{Z}_0]$ and $p(x^{\frac{2}{3^2}}) = 1 + (x^{\frac{2}{9}})^9 + (x^{\frac{2}{9}})^{36} \in \mathbb{F}_2[x; \frac{2}{9}\mathbb{Z}_0]$ are obtained through the primitive irreducible polynomial $p(x^2) = 1 + (x^2) + (x^2)^4 \in \mathbb{F}_2[x; 2\mathbb{Z}_0]$. Consequently we get $GF(2^4) \subset GF(2^{12}) \subset GF(2^{36})$, the ascending chain of Galois field extensions.

Table 32

n_i	15	45	135
s_i	4	12	36
	$\underline{[C_{n_i}^0, d], R_{15}}$	$\underline{[C_{3n_i}^1, d], R_{45}}$	$\underline{[C_{3^2n_i}^2, d], R_{135}}$
			$[(135, 99), 2, 3], 0.733$
		$[(45, 33), 2, 3], 0.733$	$[(135, 87), 4, 5], 0.644$
		$\underline{[(45, 29), 4, 5], 0.644}$	$[(135, 69), 6, 7], 0.5111$
	$\underline{[(15, 11), 2, 3], 0.733}$	$[(45, 23), 6, 7], 0.5111$	$[(135, 33), 8, 9], 0.244$
	$[(15, 7), 4, 5], 0.466$	$[(45, 11), 8, 9], 0.244$	$\underline{[(135, 29), 10], 0.215}$
	$[(15, 5), 6, 7], 0.333$	$[(45, 7), 10], 0.1555$	$[(135, 23), 16], 0.170$
	$[(15, 1), 8], .0667$	$[(45, 5), 16], 0.111$	$[(135, 11), 22], 0.0814$
		$[(45, 1), 22], 0.0222$	$[(135, 7), 28], 0.0518$
			$[(135, 5), 46], 0.0370$
			$[(135, 1), 64], 0.007$

For fixed $m_i = 2$, the relation between bandwidth and code rate is given as; $W_i = w_i(R_u/2)(1/R) = w_i R_u / 2R$, where w_i is the bandwidth expansion, R_u is the transmission rate and $R = k/n$ is the code rate. The bandwidth with different code rates is given in the following tables.

For $w_i = 1.2$ and $R_u = 64 \text{ kbps}$. Thus $w_i R_u = 76.8$ and $W_i^j = 76.8 / 2R_{bjn}^j = 38.4 / R_{bjn}^j \text{ kHz}$, where integer $j \geq 0$.

Table 33 (a)

n_i	7	49	343
s_i	3	21	147
	$\underline{[C_{n_i}^0, d], R_7^0, W_7^0 \text{ kHz}}$	$\underline{[C_{7n_i}^1, d], R_{49}^1, W_{49}^1 \text{ kHz}}$	$\underline{[C_{7^2n_i}^2, d], R_{343}^2, W_{343}^2 \text{ kHz}}$
	$\underline{[(7, 4), 2], 0.571, 67.250}$	$\underline{[(49, 4), 8], 0.081, 474.074}$	$\underline{[(343, 4), 50], 0.020, 192}$

Table 33 (b)

n_i	15	45	135
s_i	4	12	36
	$\underline{[C_{n_i}^0, d], R_{15}^0, W_{15}^0 \text{ kHz}]}$	$\underline{[C_{3n_i}^1, d], R_{45}^1, W_{45}^1 \text{ kHz}]}$	$\underline{[C_{3^2n_i}^2, d], R_{135}^2, W_{135}^2 \text{ kHz}]}$
			$[(135, 99), 2, 3], 0.733, 52.387$
		$[(45, 33), 2, 3], 0.733, 52.387$	$[(135, 87), 4, 5], 0.644, 59.627$
		$\underline{[(45, 29), 4, 5], 0.644, 59.627}$	$[(135, 69), 6, 7], 0.511, 75.122$
	$\underline{[(15, 11), 2, 3], 0.733, 52.387}$	$[(45, 23), 6, 7], 0.511, 75.122$	$[(135, 33), 8, 9], 0.244, 157.377$
	$[(15, 7), 4, 5], 0.466, 82.403$	$[(45, 11), 8, 9], 0.244, 157.377$	$\underline{[(135, 29), 10], 0.215, 178.604}$
	$[(15, 5), 6, 7], 0.333, 115.315$	$[(45, 7), 10], 0.155, 246.945$	$[(135, 23), 16], 0.170, 225.882$
	$[(15, 1), 8], 0.066, 575.712$	$[(45, 5), 16], 0.111, 345.945$	$[(135, 11), 22], 0.081, 471.744$
		$[(45, 1), 22], 0.022, 1729.729$	$[(135, 7), 28], 0.0518, 741.312$
			$[(135, 5), 46], 0.037, 1037.837$
			$[(135, 1), 64], 0.007, 5485.714$

Table 34 (c)

n_i	63	189	567
s_i	6	18	54
	$\underline{[C_{n_i}^0, d], R_{63}^0, W_{63}^0 \text{ kHz}]}$	$\underline{[C_{3n_i}^1, d], R_{189}^1, W_{189}^1 \text{ kHz}]}$	$\underline{[C_{3^2n_i}^2, d], R_{567}^2, W_{567}^2 \text{ kHz}]}$
	$\underline{[(63, 51), 4], 0.809, 47.436}$	$\underline{[(189, 123), 10], 0.650, 59.013}$	$\underline{[(567, 123), 32], 0.216, 177.040}$

7.1.3 CRFTM for BCH codes

From example 63, the (135, 29) binary non-primitive BCH code C_{135} with designed distance $d = 4$ has bandwidth $W_{135} = 178.6046$. The received polynomial

$$\begin{aligned} a^2(x^{\frac{2}{9}}) &= (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{103} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{97} + (x^{\frac{2}{9}})^{93} + (x^{\frac{2}{9}})^{91} + \\ &\quad (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{61} + (x^{\frac{2}{9}})^{58} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{52} + (x^{\frac{2}{9}})^{48} + (x^{\frac{2}{9}})^{46} + \\ &\quad (x^{\frac{2}{9}})^{45} + (x^{\frac{2}{9}})^{38} + (x^{\frac{2}{9}})^{16} + (x^{\frac{2}{9}})^{13} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^7 + (x^{\frac{2}{9}})^3 + (x^{\frac{2}{9}})^2 + (x^{\frac{2}{9}}) + 1 \end{aligned}$$

is decoded as

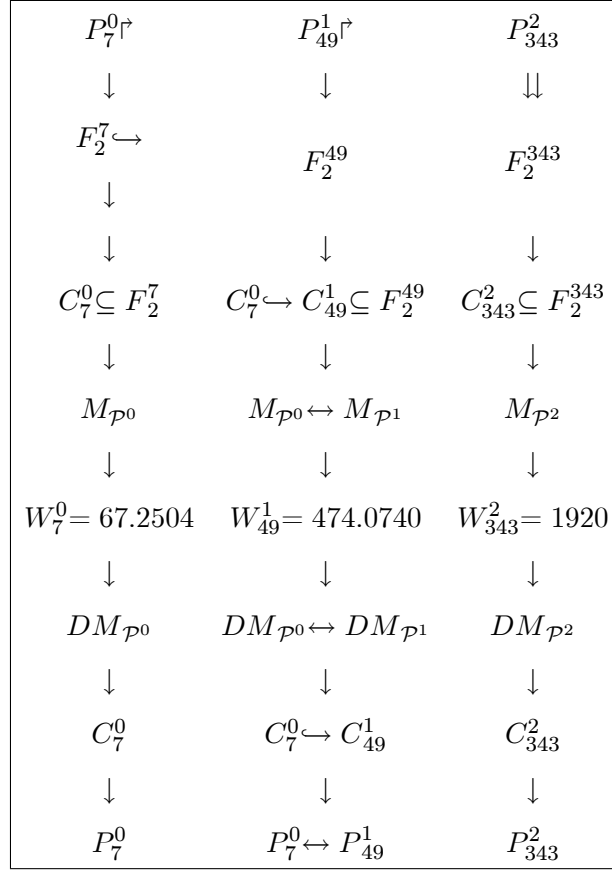
$$\begin{aligned} v^2(x^{\frac{2}{9}}) &= a^2(x^{\frac{2}{9}}) + e(x^{\frac{2}{9}}) = (x^{\frac{2}{9}})^{106} + (x^{\frac{2}{9}})^{103} + (x^{\frac{2}{9}})^{102} + (x^{\frac{2}{9}})^{97} + (x^{\frac{2}{9}})^{93} + (x^{\frac{2}{9}})^{91} + \\ &\quad (x^{\frac{2}{9}})^{90} + (x^{\frac{2}{9}})^{61} + (x^{\frac{2}{9}})^{58} + (x^{\frac{2}{9}})^{57} + (x^{\frac{2}{9}})^{52} + (x^{\frac{2}{9}})^{48} + (x^{\frac{2}{9}})^{46} + (x^{\frac{2}{9}})^{45} + \\ &\quad (x^{\frac{2}{9}})^{16} + (x^{\frac{2}{9}})^{13} + (x^{\frac{2}{9}})^{12} + (x^{\frac{2}{9}})^7 + (x^{\frac{2}{9}})^3 + (x^{\frac{2}{9}}) + 1 \end{aligned}$$

On replacing $x^{\frac{2}{9}} = y$, this gives $y^3 = x^{\frac{2}{3}}$, which gives

$$v^2(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^{16} + (x^{\frac{2}{3}})^{13} + (x^{\frac{2}{3}})^{12} + (x^{\frac{2}{3}})^7 + (x^{\frac{2}{3}})^3 + (x^{\frac{2}{3}}) + 1 \in \mathcal{C}_{45},$$

where $v^2(x^{\frac{2}{3}})$ is the generator polynomial of non-primitive binary BCH code (45, 29) with designed distance 4, and bandwidth $W_{45} = 59.6273$. Again on letting $x^{\frac{2}{3}} = y$, this gives $y^3 = x^2$, we get $v^2(x^2) = (x^2)^{13} + (x^2)^{12} + (x^2)^7 + (x^2)^3 + 1 \in C_{15}$, where C_{15} is primitive binary BCH code (15, 11) having bandwidth $W_{15} = 52.3874$, it is due to the reason that the generator polynomial

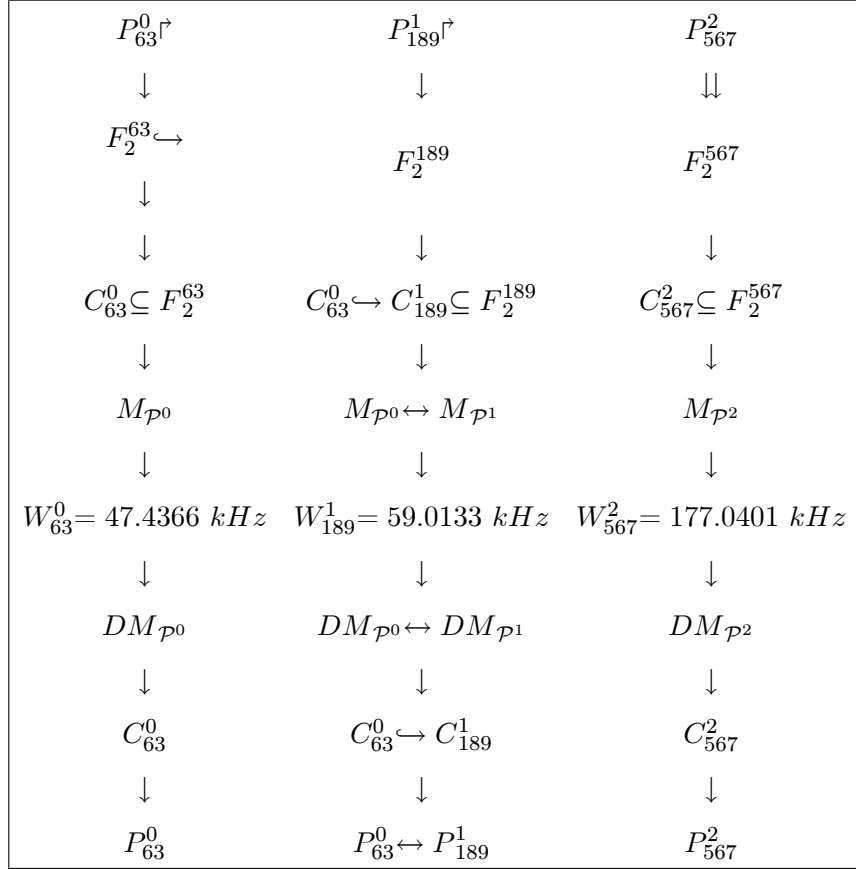
$g(x^2) = (x^2)^4 + (x^2) + 1$ divides $v^2(x^2)$.



CRFTM for BCH codes $(7, 4)$, $(49, 4)$, $(343, 4)$

$P_{15}^0 \uparrow$	$P_{45}^1 \uparrow$	P_{135}^2
\downarrow	\downarrow	\Downarrow
$F_2^{15} \hookrightarrow$	F_2^{45}	F_2^{135}
\downarrow		
\downarrow	\downarrow	\downarrow
$C_{15}^0 \subseteq F_2^{15}$	$C_{15}^0 \hookrightarrow C_{45}^1 \subseteq F_2^{45}$	$C_{135}^2 \subseteq F_2^{135}$
\downarrow	\downarrow	\downarrow
$M_{\mathcal{P}^0}$	$M_{\mathcal{P}^0} \leftrightarrow M_{\mathcal{P}^1}$	$M_{\mathcal{P}^2}$
\downarrow	\downarrow	\downarrow
$W_{15} = 52.3874$	$W_{189} = 59.6273$	$W_{135} = 178.6046$
\downarrow	\downarrow	\downarrow
$DM_{\mathcal{P}^0}$	$DM_{\mathcal{P}^0} \leftrightarrow DM_{\mathcal{P}^1}$	$DM_{\mathcal{P}^2}$
\downarrow	\downarrow	\downarrow
C_{15}^0	$C_{15}^0 \hookrightarrow C_{45}^1$	C_{135}^2
\downarrow	\downarrow	\downarrow
P_{15}^0	$P_{15}^0 \leftrightarrow P_{45}^1$	P_{135}^2

CRFTM for BCH codes $(15, 11)$, $(45, 29)$, $(135, 29)$



CRFTM for BCH codes (63, 51), (189, 123), (567, 123)

By Table 5a, 5b and 5c it is observed that corresponding to the set of BCH codes (15, 11), (45, 29) and (135, 29) (respectively (7, 4), (49, 4) and (343, 4); (63, 51), (189, 123), (567, 123)) the required bandwidths respectively are 52.3874 kHz, 59.6273 kHz and 178.6046 kHz (respectively 67.2504 kHz, 474.0740 kHz and 1920 kHz; 47.4366 kHz, 59.0133 kHz and 177.0401 kHz).

For the sequences of binary BCH codes (7, 4), (49, 4), (343, 4); (15, 11), (45, 29), (135, 29) and (63, 51) (189, 123), (567, 123)) from Tables 33a, 33b and 33c with their corresponding code rates and using the 4psk modulation schemes, we realize the symbol error rate (SER) verses signal to noise ratio (SNR) (see Fig -1, Fig -2, Fig -3). It is apparent from the Fig 3 that SER verses SNR of the sequence of binary BCH codes (63, 51) (189, 123), (567, 123) is convergent as compare to the other two sequences (7, 4), (49, 4), (343, 4) and (15, 11), (45, 29), (135, 29) of binary BCH

codes.

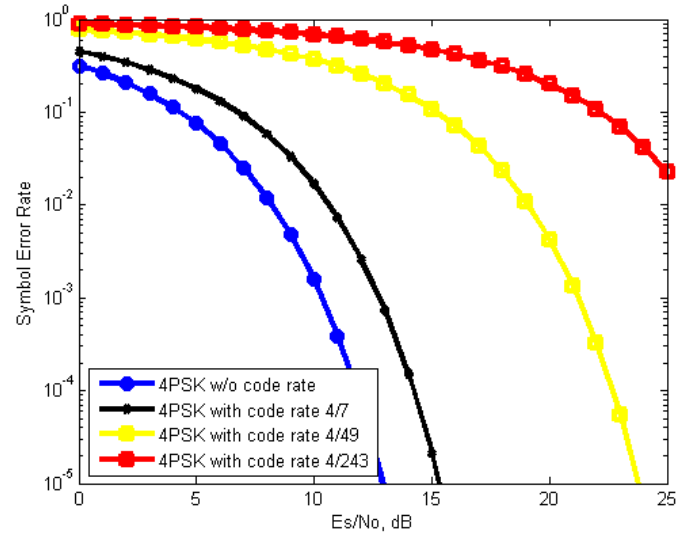


Fig-1 E_s/N_0 versus SER with different code rate

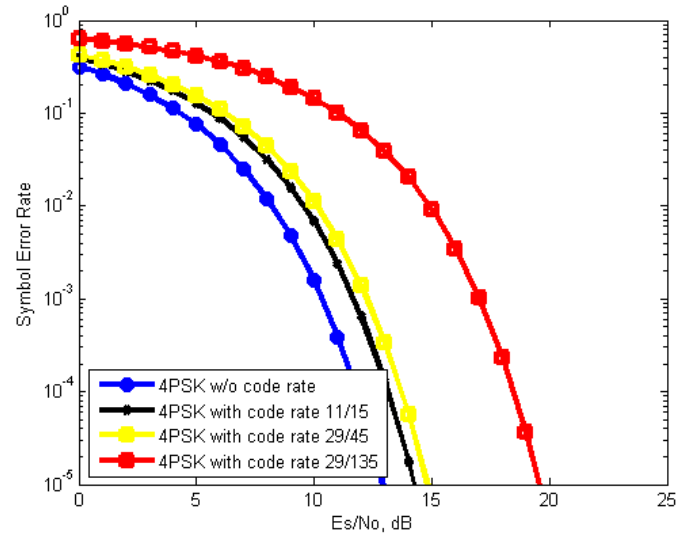


Fig-2 E_s/N_0 versus SER with different code rate

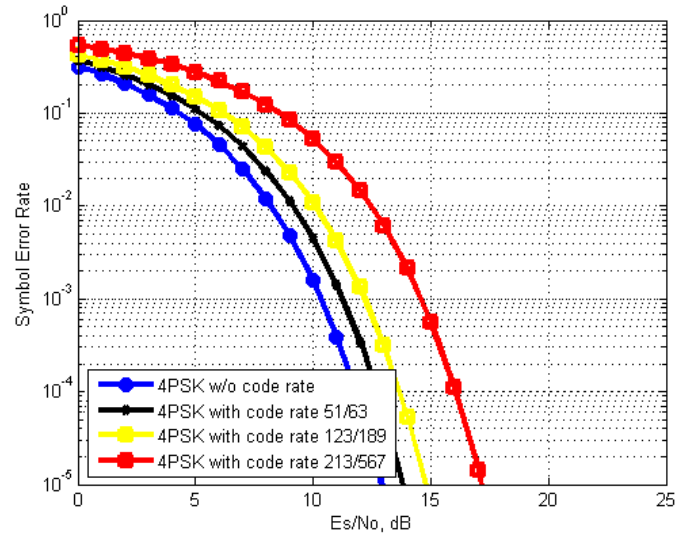


Fig-3 Es/No versus SER with different code rate

Remark 91 In CRFTM, the BCH codes are with larger lengths and higher error correction capability whenever sequence is going ahead.

- (i) User of any lesser spectrum hole can be shifted to larger spectrum hole in case it is vacant;
- (ii) Error correction capability will be enhanced in case of shifting the holes;
- (iii) Accumulative bandwidth is supporting in getting higher efficiency.

CMCRFBTM

Table 34 (a)

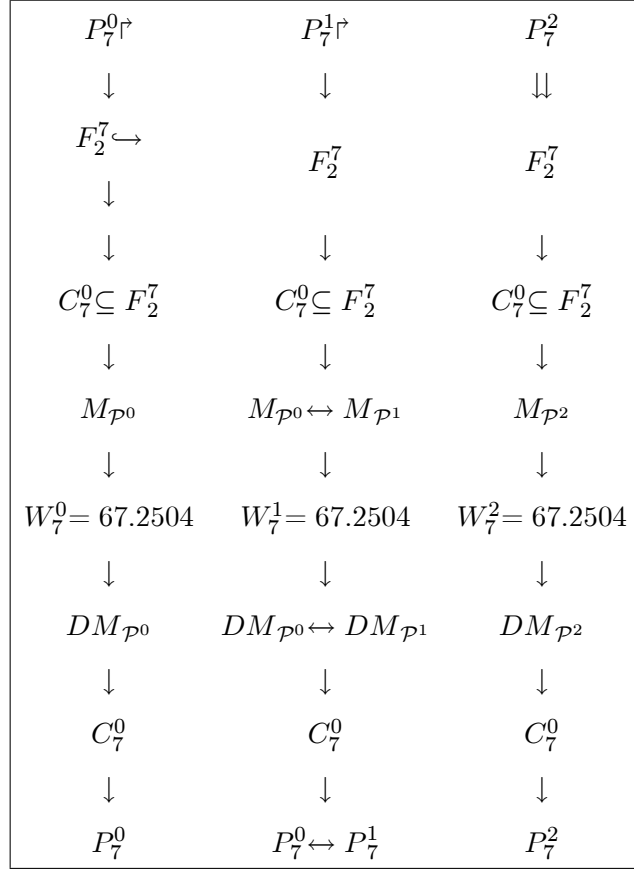
n	7	7	7
s	3	3	3
	$[C_{n_i}^0, d], R_7^0, W_7^0 \text{ kHz}$	$[C_{n_i}^0, d], R_7^0, W_7^0 \text{ kHz}$	$[C_{n_i}^0, d], R_7^0, W_7^0 \text{ kHz}$
	$[(7,4),2], 0.571, 67.2504$	$[(7,4),2], 0.571, 67.2504$	$[(7,4),2], 0.571, 67.2504$

Table 34 (b)

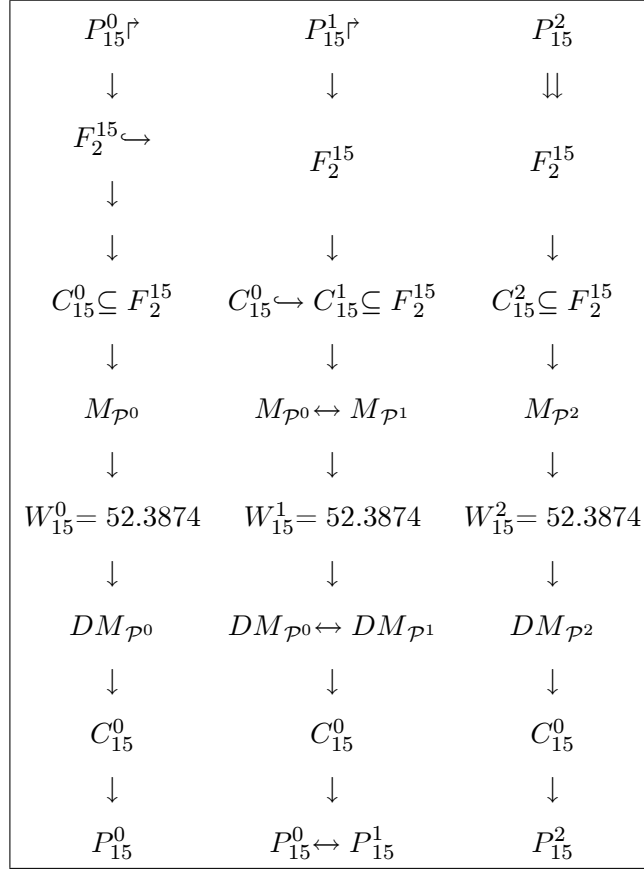
n	15	15	15
s	4	4	4
	$\underline{[C_{n_i}^0, d], R_{15}^0, W_{15}^0 \text{ kHz}}$	$\underline{[C_{n_i}^0, d], R_{15}^0, W_{15}^0 \text{ kHz}}$	$\underline{[C_{n_i}^0, d], R_{15}^0, W_{15}^0 \text{ kHz}}$
	$\underline{\underline{[(15,11),2,3],0.733,52.3874}}$	$\underline{\underline{[(15,11),2,3],0.733,52.3874}}$	$\underline{\underline{[(15,11),2,3],0.733,52.3874}}$

Table 34 (c)

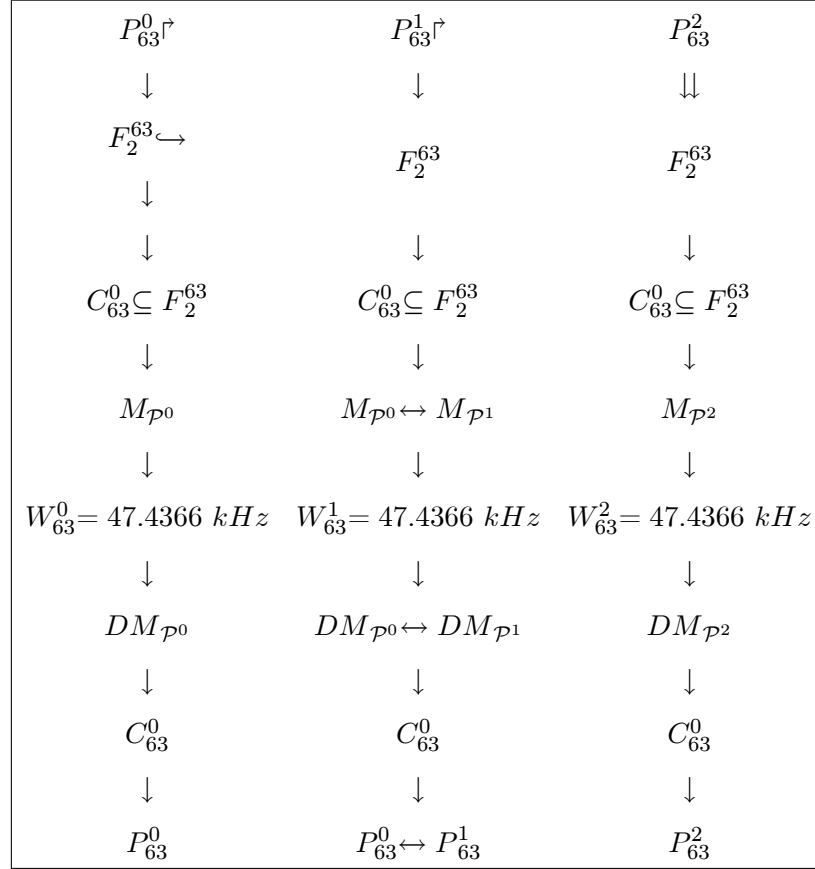
n	63	63	63
s	6	6	6
	$\underline{[C_{n_i}^0, d], R_{63}^0, W_{63}^0 \text{ kHz}}$	$\underline{[C_{n_i}^0, d], R_{63}^0, W_{63}^0 \text{ kHz}}$	$\underline{[C_{n_i}^0, d], R_{63}^0, W_{63}^0 \text{ kHz}}$
	$\underline{\underline{[(63,51),4],0.8095,47.4366}}$	$\underline{\underline{[(63,51),4],0.8095,47.4366}}$	$\underline{\underline{[(63,51),4],0.8095,47.4366}}$



CRFTM for BCH codes $(7, 4), (7, 4), (7, 4)$



CRFTM for BCH codes $(15, 11)$, $(15, 11)$, $(15, 11)$



CRFTM for BCH codes (63, 51), (63, 51), (63, 51)

Remark 92 In CMCRFBTM, same BCH code is repeated for any spectrum hole and thus we obtain the following outcomes.

- (i) Any user can move to any of the hole in case it is vacant.
- (ii) Error correction capability is fixed for all BCH codes configured with spectrum holes.
- (iii) Accumulative bandwidth is not supporting in higher efficiency.

7.2 Application in DNA formation

DNA (or deoxyribonucleic acid) is an inherited material in humans and almost all other organisms. Nearly every cell in a human's body has the same DNA. Mostly DNA is located in the cell nucleus. The information in DNA is stored as a code made up of four chemical bases:

adenine (**A**), guanine (**G**), cytosine (**C**), and thymine (**T**). Human DNA consists of about 3 billion bases, and more than 99 percent of those bases are the same in all people. The order, or sequence, of these bases determines the information available for building and maintaining an organism, similar to the way in which letters of the alphabet appear in a certain order to form words and sentences.

DNA bases pair up with each other, A with T and C with G, to form units called base pairs. Each base is also attached to a sugar molecule and a phosphate molecule. Together, a base, sugar, and phosphate are called a **nucleotide**. The order, or sequence, of these bases determines what biological instructions are contained in a strand of DNA. For example, the sequence ATCGTT might instruct for blue eyes, while ATCGCT might instruct for brown. DNA contains the instructions needed for an organism to develop, survive and reproduce. To carry out these functions, DNA sequences must be converted into messages that can be used to produce proteins, which are the complex molecules that do most of the work in our bodies. Each DNA sequence that contains instructions to make a protein is known as a **gene**. An important property of DNA is that it can replicate, or make copies of itself. Each strand of DNA can serve as a pattern for duplicating the sequence of bases. Researchers refer to DNA found in the cell's nucleus as nuclear DNA. An organism's complete set of nuclear DNA is called its **genome**.

In [25], the authors showed that the DNA also contain an error correcting code. They proposed that if a linear block error correcting code is present in DNA then some bases would be a linear function of the other bases in each set of bases. An efficient procedure is given to determine if such an error correcting code is present in the base sequence. Furthermore in [11], Faria et al. confirmed that there are DNA sequences that can be identified as codewords for error correcting codes. In [1], Abualrub et al. proposed a theory for constructing linear and additive cyclic codes of odd length over $F_4 = \{0, 1, \alpha, \alpha^2 = 1 + \alpha\}$ that are suitable for DNA computing. In [10], Faria et al. have showed the existence of DNA sequences which can be identified as codewords of BCH codes over the field F_4 . They have proposed an algorithm capable of producing DNA sequences, associated with coding regions of genes, as codewords of error-correcting codes. Their results allow the use of efficient computer simulations in the analysis of biological processes such as polymorphism and mutation, consequently reducing time

spent in laboratorial experiment. Much work has done in this area see ([12] and [8]).

In this section, we discuss the application of newly constructed non-primitive BCH codes over the field \mathbb{F}_4 , explain in chapter 5, in the formation of DNA sequences followed by the process discussed in [10, Table 1]. In [10, Table 1], the authors proposed an algorithm in which they first consider a DNA sequence as; if it were a codeword and make the conversion of the 24 permutations between the set of nucleotides $N = \{A, C, G, T\}$ and the code alphabet from the field $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2 = 1 + \alpha\}$. To check whether each one of the 24 possibilities is in fact a codeword, they use the relation $v.H^T = 0$, where v is a possible codeword. To analyze the difference between the DNA sequence and the codeword, three other possibilities for nucleotides in each position in the DNA sequence is analyzed, for each permutation, and again the relation $v.H^T = 0$ is used to verify whether v is a codeword.

Following the same strategy our construction gives repeated DNA sequences whenever both the primitive and non-primitive BCH codes have same dimension. Repeated sequences (repetitive elements, or repeats) are patterns of nucleic acids (DNA or RNA) that occur in multiple copies throughout the genome. There are 3 major categories of repeated sequence: 1) Terminal repeats, 2) Tandem repeats, 3) Interspersed repeats. We are getting tandem repeating sequence. Repetition of a pattern of one or more nucleotides in DNA such that repetition is directly adjacent to each other is called **Tandem repeats**. Several protein domains also form tandem repeats within their amino acid primary structure. For example in ATTCG ATTCG ATTCG the sequence ATTCG is repeated three times. Tandem repeat describes a pattern that helps determine an individual's inherited traits. Tandem repeats can be very useful in determining parentage.

In the following example we show how a DNA sequence associated with a generator polynomial of a BCH code tandem repeats.

Example 93 *The following example deals with a non primitive BCH code of length 189 based on the primitive BCH code of length 63 using same primitive polynomial discussed in [10, Table 1], for *Triticum aestivum* with GI number 78096542. For a primitive polynomial $p(x^2) = (x^2)^3 + b(x^2)^2 + (x^2) + \alpha$ in $\mathbb{F}_4[x; 2\mathbb{N}_0]$ there is a non-primitive irreducible polynomial $p(x^{\frac{2}{3}}) = (x^{\frac{2}{3}})^9 + b(x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^3 + \alpha$ in $\mathbb{F}_4[x; \frac{2}{3}\mathbb{N}_0]$. Let $\beta \in GF(4^6)$, satisfying the relation $\beta^9 + b\beta^6 + \beta^3 + \alpha = 0$. Using this relation we get $\beta^{189} = 1$. Following the above construction we get the following*

tables of BCH codes \mathcal{C}_{63} and \mathcal{C}_{189} over $\mathbb{F}_4[x; 2\mathbb{N}_0]$ and $\mathbb{F}_4[x; \frac{2}{3}\mathbb{N}_0]$.

Table 35: BCH codes \mathcal{C}_{63} and \mathcal{C}_{189} over $\mathbb{F}_4[x; 2\mathbb{N}_0]$ and $\mathbb{F}_4[x; \frac{2}{3}\mathbb{N}_0]$.

(n, k)	d	t	R	(bn, k_1)	d_1	t_1	R_1
(63, 60)	3	1	0.952	(189, 180)	3	1	0.952
(63, 57)	4	1	0.904	(189, 168)	5	2	0.888
(63, 54)	5	2	0.857	(189, 156)	7	3	0.825
(63, 51)	7	3	0.809	(189, 147)	9	4	0.777
(63, 48)	8	3	0.761	(189, 135)	11	5	0.714
(63, 45)	9	4	0.714	(189, 117)	15	7	0.619
(63, 42)	11	5	0.666	(189, 102)	21	10	0.539
(63, 39)	12	5	0.619	(189, 81)	27	13	0.428
(63, 36)	13	6	0.571	(189, 57)	33	16	0.301
(63, 33)	15	7	0.523	(189, 54)	39	19	0.285

The generating polynomial of the code (63, 57) repeats three times in the generating polynomial of the code (189, 57) that is:

$$\begin{aligned}
g(x^2) &= (x^2)^6 + (x^2)^5 + (x^2)^4 + (x^2) + 1 \\
g(x^{\frac{2}{3}}) &= (x^{\frac{2}{3}})^{132} + (x^{\frac{2}{3}})^{131} + (x^{\frac{2}{3}})^{130} + (x^{\frac{2}{3}})^{127} + (x^{\frac{2}{3}})^{126} + (x^{\frac{2}{3}})^{69} + (x^{\frac{2}{3}})^{68} \\
&\quad + (x^{\frac{2}{3}})^{67} + (x^{\frac{2}{3}})^{64} + (x^{\frac{2}{3}})^{63} + (x^{\frac{2}{3}})^6 + (x^{\frac{2}{3}})^5 + (x^{\frac{2}{3}})^4 + (x^{\frac{2}{3}}) + 1.
\end{aligned}$$

Therefore by Remark 72 the codewords in (63, 57) also repeats 3 times in the codewords of (189, 57). This means that the whole DNA sequence generated by (63, 57) BCH code over \mathbb{F}_4 in [10, Fig 1] tandem repeats three times to form a DNA sequence associated with a codeword in

(189, 57). Hence we get:

```

ATGGCCGCACGCCTCGCGCTGGTGGCGGCGCTCCTGTGCTCCGGTGCCACGGCCGCCGCGGCG
ATGGCCGCACGCCTCGCGCTGGTGGCGGCGCTCCTGTGCTCCGGTGCCACGGCCGCCGCGGCG
ATGGCCGCACGCCTCGCGCTGGTGGCGGCGCTCCTGTGCTCCGGTGCCACGGCCGCCGCGGCG
0a2aa11a101a11a21a1a1a2aa2aa1aa1a1a211a2aa2a1a211aa2a1101aa11a11a1aa1a
0a2aa11a101a11a21a1a1a2aa2aa1aa1a1a211a2aa2a1a211aa2a1101aa11a11a1aa1a
0a2aa11a101a11a21a1a1a2aa2aa1aa1a1a211a2aa2a1a211aa2a1101aa11a11a1aa1a

```

Fig 4: DNA sequence corresponding to a codeword of length 189 such that a DNA sequence associated with a codeword of length 63 is tandem repeating in it.

In this sequence we are getting triple nucleotide polymorphism occurring after a fix interval. Similarly for other sequences discussed in [10, Fig 1 and Fig 2], we get triple nucleotide polymorphism.

7.3 Application in data transformation

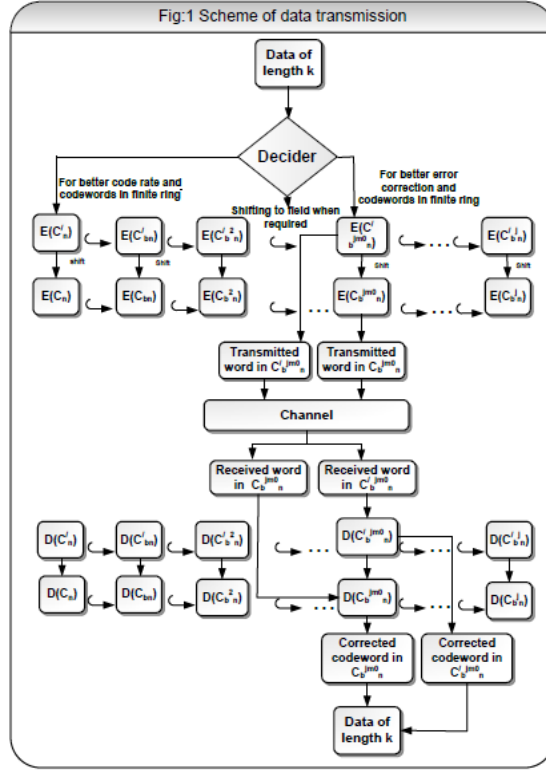
In coding theory the noisier the channel, the longer the codeword has to be to ensure perfect communication. But the longer the codeword, the longer it takes to transmit the message. Therefore, a good communication requires precisely matching codeword length to the level of noise in the channel. Wireless devices, such as cell phones or Wi-Fi transmitters, regularly send out test messages to estimate noise levels, so they can adjust their codes accordingly. However, as in cell phone reception quality can vary at locations just a few feet apart or even at a single location. Noise measurements quickly become outdated, and wireless devices routinely end up with codewords that are too long, wasting bandwidth, or too short, making accurate decoding impossible.

In this work, our newly constructed codes are linked in such a way that we may easily shift the data from smaller code of length n to any of larger code of length $b^j n$ in the sequences $\{C'_{b^j n}\}_{j \geq 1}$, and $\{C_{b^j n}\}_{j \geq 1}$, according to requirement of noise levels and obtain the benefit of

better error correction. Similarly, for fast transmission of code we can shift data from $b^j n$ length code to n length code. Furthermore, the bandwidth is maximally utilized by the code C_n and C'_n because they are embedded in codes of the sequences $\{C_{b^j n}\}_{j \geq 1}$ and $\{C'_{b^j n}\}_{j \geq 1}$ respectively. Therefore for a fixed m , against n length primitive BCH codes C_n and C'_n over \mathbb{Z}_2 and \mathbb{Z}_{2^m} , there exist two sequences $\{C_{b^j n}\}_{j \geq 1}$ and $\{C'_{b^j n}\}_{j \geq 1}$ of non-primitive BCH codes over \mathbb{Z}_2 and \mathbb{Z}_{2^m} respectively. Through embedding and the p reduction map these two sequences are established. Hence a data can be transmitted via any of the coding schemes $C_n, C'_n, \{C_{b^j n}\}_{j \geq 1}$ and $\{C'_{b^j n}\}_{j \geq 1}$. This selection of scheme is based on the choice of better code rate or better error correction capability of the chosen code.

A non-primitive BCH code in the family $\{C'_{b^j n}\}_{j \geq 1}$ or $\{C_{b^j n}\}_{j \geq 1}$ has larger code length if j is larger. However, if $C'_{b^l n}$ is a code from the sequence $\{C'_{b^j n}\}_{j \geq 1}$ and $C_{b^l n}$ is a code from the sequence $\{C_{b^j n}\}_{j \geq 1}$. They have same code length, code rate and error correction capability but $C'_{b^l n}$ have more codewords than $C_{b^l n}$. By increasing codewords maximum information per unit time is transferred, thus, the code $C'_{b^l n}$ has more benefit than the code $C_{b^l n}$. Besides this, the long length non-primitive BCH codes $C'_{b^l n}$ and $C_{b^l n}$ contain the information of smaller primitive BCH codes C'_n and C_n .

For any j_{m_0} , where $1 \leq 2 \leq \dots \leq j_{m_0}$, the synchronized encoding and decoding of the BCH codes $C_{b^{j_{m_0}} n}$ and $C'_{b^{j_{m_0}} n}$ are considered. Accordingly, any chosen BCH codes $C_{b^j n}$ and $C'_{b^j n}$ with $0 \leq j < j_{m_0}$, for data transmission, can be replace by BCH codes $C_{b^{j_{m_0}} n}$ and $C'_{b^{j_{m_0}} n}$, respectively, to decode simultaneously the codewords of BCH codes C_n and C'_n . The whole scheme is described in the following figure 1, where $E(C'_{b^j n})$, $D(C'_{b^j n})$, $E(C_{b^j n})$ and $D(C_{b^j n})$ stands the encoder and decoder of $C'_{b^j n}$ and $C_{b^j n}$ respectively, for all $0 \leq j < j_{m_0}$. EC and CR are use for error correction and code rate of the code and the arrow " \hookrightarrow " shows the embedding of codes.



7.4 Summary

In short these are the following applications of non-primitive BCH codes constructed on the base of primitive BCH codes over the Galois field \mathbb{F}_2 , the Galois field \mathbb{F}_4 and the finite Galois rings \mathbb{F}_{2^m} .

The BCH codes over the Galois field \mathbb{F}_2 , gives a novel interweave multiple transmission model for cognitive radios. The data of set $\{\mathcal{P}_{bj_{n_i}}^j, 1 \leq i \leq i_0\}_{j=0}^{j_0}$ of primary users is configured and transmitted through the set $\{C_{bj_{n_i}}^j, 1 \leq i \leq i_0\}_{j=0}^{j_0}$ of binary BCH codes. However, corresponding to each primary user $\mathcal{P}_{bj_{n_i}}^j$, the data of the family $\{\mathcal{P}_{bj_{n_i}}^j\}_{j=1}^{j_0}$ of primary users is configured by the family $\{C_{bj_{n_i}}^j\}_{j=1}^{j_0}$ of binary BCH codes having sequentially increasing code lengths and error correcting capabilities. Due to the choice of the modulation scheme, every member of the family $\{C_{bj_{n_i}}^j\}_{j=0}^{j_0}$ requires sequentially increasing but different bandwidths. A multiple transmission pattern is planned in the spirit of interweave model in such a way that the user

$\mathcal{P}_{b^l n_i}^l$ opportunistically avail the channel path (spectrum hole) of any of the primary users in the family $\{\mathcal{P}_{b^j n_i}^j\}_{j=l}^{j_0}$, which is not utilizing its allotted spectrum hole.

Whereas in BCH codes over Galois field \mathbb{F}_4 , the codewords of non-primitive BCH codes of length bn , having same code dimension as that of BCH code of length n , contains codewords of C_n which repeats b times in it. Consequently, the DNA sequence associated with the codeword of the code C_n tandem repeats b times in the DNA sequence associated with the codeword in C_{bn} .

Finally, the BCH codes over finite Galois ring are linked with the BCH codes over Galois field \mathbb{F}_2 in such a way that: one can easily shift the data from code of length n to any of the code of length $b^j n$, where $j \geq 1$, in order to obtain the benefit of better error correction. Whereas, for fast transmission of code data can be shifted from the $b^j n$ length code to the n length code. The bandwidth is maximally utilized by the code of length n as they are embedded in codes of length $b^j n$. The selection of a code is based on its length, code rate and error correction capability.

Chapter 8

Conclusion

In current times, there has been an increasing demand for digital transmission and storage systems. For this a digital system must be fully reliable, as a single error may collapse the whole system, or cause undesirable corruption of data. In such situations error correcting codes must be employed so that an error may be detected and subsequently corrected. In this work we have constructed cyclic and particularly BCH codes using monoid rings instead of polynomial ring. Through monoid rings the length of the polynomials is increased which increases the code length. Hence it is required to construct such codes which can correct more errors.

Initially we have constructed binary cyclic codes, using monoid rings. A technique is given in such a manner that for an n length binary cyclic code \mathcal{C}_n , there exists binary cyclic codes \mathcal{C}_{an} , \mathcal{C}_{bn} and \mathcal{C}_{abn} of lengths an , bn and abn . These codes are found to be interleaved codes and are linked together in a special way. Therefore, they are capable of correcting random as well as burst of errors. Afterwards we have constructed non-primitive BCH codes over Galois field \mathbb{F}_2 using monoid ring instead of polynomial rings. This construction is based on a primitive BCH code, which gives an association between primitive and non-primitive BCH codes. The non-primitive BCH codes gives better error correction capability with a little deprivation in code rate.

Moreover, we have constructed non-primitive BCH codes over the four elements Galois field \mathbb{F}_4 . These codes have better code rate as compare to the codes obtained over the field \mathbb{F}_2 . Also the possible choices of BCH codes over \mathbb{F}_4 are more as compare to the BCH codes over \mathbb{F}_2 .

In this work BCH codes over monoid rings are constructed, other codes like Reed Solomon,

Golay and Fire codes can also be constructed using the same monoid rings. Another new family of cyclic as well as BCH codes can be constructed by taking some other similar monoids. We have given an algorithm to calculate non-primitive BCH codes over the field \mathbb{F}_2 , which can further be enhanced for other Galois fields and Galois rings as well. Further by considering a varying positive integer m for local ring \mathbb{Z}_{2^m} , we will obtain family of sequences $\{\mathcal{C}_{b^j n}\}_{j \geq 1, m \geq 2}$ and $\mathcal{C}'_{b^l n} \in \{\mathcal{C}'_{b^j n}\}_{j \geq 1, m \geq 2}$ of non-primitive BCH codes. This will serve the purpose at large scale and for multiple uses.

Other than the given applications this work can be implemented in cryptography. In [15], the authors have given a notion of cryptcoding, it is a procedure through which they have joined together, encryption and error-correction in one step. Following [15], this work can easily be implemented in cryptography for error free secure network.

Bibliography

- [1] T. Abualrub, A. Ghrayeb, X. N. Zeng, Construction of cyclic codes over $GF(4)$ for DNA computing, J. Franklin Inst., 343(4), 448-457 (2006).
- [2] A. A. Andrade and R. Palazzo Jr., Linear codes over finite rings, TEMA-Tend. Mat. Apl. Comput., 6(2), 207-217 (2005).
- [3] A. A. Andrade and R. Palazzo Jr., Construction and decoding of BCH codes over finite rings, Linear Algebra Applic., 286, 69-85 (1999).
- [4] A. A. Andrade, T. Shah and A. Khan, A note on linear codes over semigroup rings. TEMA Tend. Mat. Apl. Comput. 12(2), 79-89 (2011).
- [5] I. F. Blake, Codes over certain rings, Inform. Contr., 20(4), 396-404 (1972).
- [6] I. F. Blake, Codes over integer residue rings, Inform. Contr., 29(4), 295-300 (1975).
- [7] R. C. Bose, D. K. Ray-Chaudary, On a Class of Error Correcting Binary Group Codes", Information and Control 3 (1): 68-79, doi:10.1016/s0019-9958(60)90287-4, (1960).
- [8] M. M. Brandao, L. Spoladore, L. C. B. Faria, A. S. L. Rocha, M. C. Silva-Filho and R. Plazzo, Ancient DNA sequence revealed by error-correcting codes, Scientific Reports 5, (2015), DOI:10.1038/srep12051.
- [9] J. Cazaran, A. V. Kelarev, S. J. Quinn, D. Vertigan, An algorithm for computing the minimum distances of extensions of BCH codes embedded in semigroup rings. Semigroup Forum. 73(3), 317-329 (2006).

- [10] L. C. B Faria, A. S. L Rocha, J. H Kleinschmidt, R. Palazzo Jr. and M.C. Silva-Filho, DNA sequences generated by BCH codes over $GF(4)$, Electronics letters, 46(3), 202-203 (2010).
- [11] L.C.B. Faria, A.S.L. Rocha, J.H. Kleinschmidt, M.C. Silva-Filho, E. Bim , R.H. Herai , M.E.B. Yamagishi, and R. Palazzo Jr., Is a Genome a Codeword of an Error-Correcting Code? PLoS ONE (2012). 7(5): e36644.
- [12] L. C. B. Faria, A. S. L. Rocha and R. Palazzo Jr., Transmission of intra-cellular genetic information: A system proposal, Journal of theoretical biology 358, 208-231 (2014).
- [13] G. D. Forney Jr., On decoding BCH codes, IEEE Trans. Inform. Theory, 11(4), 549-557 (1965).
- [14] S. Gao and D. Panario, Tests and Constructions of Irreducible Polynomials over Finite Fields, Foundations of Computational Mathematics, Springer, 346-361 (1997).
- [15] D. Gligoroski, S.J. Knapskog, S. Andova, Cryptcoding : encryption and error-correction coding in a single step, Proceedings of the International Conference on Security & Management (SAM 2006) 26-29 June, ISBN 1-60132-011-9,145-151 (2006).
- [16] A. Hocquenghem, Codes correcteurs d'erreurs, Chiffres (in French) (Paris) 2, 147-156 (1959).
- [17] W. C. Huffman, Vera Pless, Fundamentals of error-correcting codes, Cambridge University Press (2003).
- [18] J. C. Interlando, R. Palazzo, Jr., M. Elia, On the decoding of Reed-Solomon and BCH codes over integer residue rings, IEEE Trans. Inform. Theory, IT-43, 1013-1021 (1997).
- [19] T. Kasami, Systematic codes using binary shift register sequences, J. info. Processing Soc. Japan, 1, 198-200 (1960).
- [20] A. V. Kelarev, Ring constructions and applications, World Scientific, River Edge, New York (2002).

- [21] A. V. Kelarev, An algorithm for BCH codes extended with finite state automata. *Fundamenta Informaticae*, 84(1), 51-60 (2008).
- [22] A. V. Kelarev, Algorithms for computing parameters of graph-based extensions of BCH codes. *Journal of Discrete Algorithms*, 5(6), 553-563 (2007).
- [23] E. Krouk and S. Semenov, *Modulation and coding techniques in wireless communications*, John Wiley & Sons, Ltd, (2011).
- [24] H. Kim and K. G. Shin, In-band spectrum sensing in cognitive radio networks: energy detection or featur detection, In *Proceedings of the 14th ACM international conference on mobile computing and networking*, 14-25 (2008).
- [25] L. S. Liebovitch, Y. Tao, A. T. Todorov, and L. Levine, Is there an error correcting code in the base sequence in DNA, *Biophysical Journal*, 71, 1539-1544 (1996).
- [26] B. R. McDonlad, *Finite rings with identity*, Marcel Dekker, New York, (1974).
- [27] S. R. Nagpaul, S. K. Jain, *Topics in Applied Abstract Algebra*, Thomson, Brooks/Cole, U.S.A. (2005).
- [28] W. W. Peterson, Encoding and error-correction procedures for the Bose-Chaudhuri codes. *IRE Trans.*, IT-6, 459-470 (1960).
- [29] W. W. Peterson and E. J. Weddon, Jr., *Error correcting codes*, 2nd edittion, MIT Press Cambridge, Mass. (1972).
- [30] E. Prange, *Cyclic error-correcting codes in two symbols (AFCRC-TN-57-103*, Air force Cambridge research center, Cambridge, Mass. (1957)).
- [31] E. Prange, *The use of coset equivalence in the analysis and decoding of group codes. AFCRC-TR-59-164*, Air force Cambridge research center, Cambridge, Mass. (1959).
- [32] Rahman Doost-Mohammady, *Cognitive radio design: An SDR Approach*, Wireless and mobile communications Mekelweg 4, 2628 CD Delft.

- [33] T. Shah, A. Khan and A. A. Andrade, Ascending chains of semigroup rings and encoding: Proceedings of ITS, International Telecommunication Symposium), September 06-09, Manaus, AM, Brazil, 1-5 (2010).
- [34] T. Shah, A. Khan and A. A. Andrade, Encoding through generalized polynomial codes. *Comp. Appl. Math.*, 30(2), 349-366 (2011).
- [35] T. Shah, A. Khan and A. A. Andrade, Constructions of codes through semigroup ring $B[X; \frac{1}{2^2}\mathbb{N}_0]$ and encoding, *Comput. Math. Appl.* 62, 1645-1654 (2011).
- [36] T. Shah and A. A. Andrade, Cyclic codes through $B[X; \frac{a}{b}\mathbb{N}_0](\frac{a}{b} \in Q^+, b = a + 1)$ and Encoding, *Discrete Mathematics, Algorithms and Applications*, 4(4) (2012). DOI: 10.1142/S1793830912500590
- [37] T. Shah and A. A. Andrade, Cyclic codes through $B[X]$, $B[X; \frac{1}{kp}\mathbb{N}_0]$ and $B[X; \frac{1}{p^k}\mathbb{N}_0]$: A comparison. *Journal of Algebra and its Applications*, 11(4) (2012). DOI: 10.1142/S0219498812500788.
- [38] T. Shah, Amanullah, A. A. Andrade, A decoding procedure which improves code rate and error corrections, *JARAM*, 4(4), 37-50 (2012).
- [39] T. Shah, Amanullah, A. A. Andrade, A method for improving the code rate and error correction capability of a cyclic code, *Computational and Applied Mathematics*, 32(2), 261-274 (2013).
- [40] T. Shah, M. Khan and A. A. Andrade, A decoding method of an n length binary BCH code through $(n+1)n$ length binary cyclic code, *Anais da Academia Brasileira de Ciências* 85(3), 863-872 (2013).
- [41] P. Shankar, On BCH codes over arbitrary integer rings, *IEEE Trans. Inform. Theory*, IT-25(4), (1979), 480-483.
- [42] E. Spiegel, Codes over \mathbb{Z}_m , *Inform. Control*, 35, 48-51 (1977).
- [43] E. Spiegel, Codes over \mathbb{Z}_m , Reviseted, *Inform. Control*, 37, 100-104 (1978).al Processing Magazine, 79-89 (2007).