Expository Paper

Introduction to Reed-Solomon Codes

Author: Yo Sup (Joseph) Moon

Advisor: Nathan Kaplan

August 8, 2011
1 Introduction

1.1 History

Reed-Solomon Codes are error-correcting codes with applications ranging from data retrieval from bar codes and QR codes in our daily lives to sending transmissions to and from spacecrafts launched in deep-space missions. The Reed-Solomon (RS) Code was discovered by Irving Reed and Gus Solomon and was subsequently presented to the world in their paper “Polynomial Codes over Certain Finite Fields” in the Journal of the Society for Industrial and Applied Mathematics (1959). Since then, RS Codes have been an integral contributor to the telecommunications revolution that took place in the last half of the twentieth century. In particular, Reed-Solomon codes are the most frequently used digital error control codes in the world, due their usage in computer memory and non-volatile memory applications [1]. A hurried list of significant applications include the Digital Audio Disk, Deep Space Telecommunication Systems, Error Control for Systems with Feedback, Spread-Spectrum Systems, and Computer Memory.

1.2 Mathematical Background

Definition. The Hamming Distance \(d(x,y)\) between two strings \(x, y\) from a particular alphabet is defined as the metric measuring the number of differing coordinates.

Definition. A code of length \(n\) is a set containing codewords \(c_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,n}), c_{i,j} \in \mathbb{F}_q\).

Definition. A code \(C \subset \mathbb{F}_q^n\) is linear if it is a linear subspace. The dimension \(k\) of a code is the dimension of the subspace. Further, the minimum distance \(d\) is defined as the smallest Hamming distance between any two distinct codewords. A linear code of length \(n\), dimension \(k\) and minimum distance \(d\) is denoted \([n,k,d]\).

Definition. A generator matrix for an \([n,k,d]\) code \(C\) is any \(k \times n\) matrix \(G\) whose rows form a basis for \(C\). For any set of \(k\) independent columns of a generator matrix \(G\), the corresponding set of coordinates form an information set.

Definition. A linear code \(C\) is cyclic if

\[(c_0, c_1, \ldots, c_{n-1}) \in C \Rightarrow (c_1, c_2, \ldots, c_{n-1}, c_0) \in C.\]

Proposition. Singleton Bound.

Let \(d(x,y)\) be the Hamming distance between \(x\) and \(y\). Let \(A_q(n,d)\) represent the maximum number of possible codewords in a \(q\)-ary block code of length \(n\) and minimum distance \(d\).

Then,

\[A_q(n,d) \leq q^{n-d+1}.\]

If an \([n,k,d]\) code over \(\mathbb{F}_q\) exists, then \(k \leq n - d + 1\).

Definition. A code \(C\) is Maximum Distance Separable (MDS) if equality holds in the Singleton Bound.

Theorem. If \(C\) is an \([n,k,d]\) code, then every \(n - d + 1\) coordinate position contains an information set. In addition, \(d\) is the largest number with this property.

Proof. Consult [3] \(\square\)

Definition. Let \(\alpha_1, \ldots, \alpha_s\) be elements in \(\mathbb{F}\). The \(s \times s\) matrix \(V = [v_{i,j}]\) where \(v_{i,j} = \alpha_j^{i-1}\) is called a Vandermonde matrix.

Lemma. A matrix \(V\) is nonsingular if the elements \(\alpha_1, \ldots, \alpha_s\) are distinct.
2 Approaches to the RS Code

2.1 Original Conception

Reed and Solomon’s original approach in constructing and decoding their code was not done via generator polynomials, but by using finite field arithmetic.

Definition. A Galois Field is a finite field of \( q \) elements. It is denoted by \( GF(q) \).

Proposition. \( GF(q) \) contains an element \( \alpha \) such that \( \alpha^{q-1} = 1 \). This element is called primitive. In particular, there is no \( m < q - 1 \) such that \( \alpha^m = 1 \).

Since \( \alpha \) is primitive, the elements \( \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\} \) must be distinct. Then it follows that \( GF(q) = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\} \).

Multiplication in the field, then, is well-defined and obeys the “obvious” rules of exponentiation:

\[
\alpha^x \cdot \alpha^y = \alpha^{x+y}.
\]

Addition works in a similar manner.

Definition. A primitive polynomial \( P(X) \) over \( GF(q) \) is an irreducible polynomial with coefficients in \( GF(q) \) and a primitive root \( \alpha \) in \( GF(q) \).

The RS Code is constructed in the following manner:

Suppose \( m := (m_0, m_1, m_2, \ldots, m_{k-1}) \) is a list of information symbols (i.e. the information one wants to encode), with each coordinate taken from \( GF(q) \). Define

\[
P(X) := m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}.
\]

Each code word \( c \) is generated by evaluating the polynomial by each member of \( GF(q) \):

\[
c = (c_0, c_1, \ldots, c_{q-1}) = (P(0), P(\alpha), \ldots, P(\alpha^{q-1})) .
\] (1)

The complete code \( C \) is constructed by choosing \( m \) over all possible values. Thus it follows that there are \( q^k \) codewords in \( C \). And since any two polynomials of degree \( (k - 1) \) is another polynomial of degree less than or equal to \( (k - 1) \), we have that \( C \) is linear. We say that this code is of length \( n = q \) and dimension \( k \) and denote it by \( RS(n, k) \).

Now, consider a particular codeword \( c \) in \( C \). From (1), we get a system of \( q \) linear equations with \( k \) variables:

\[
\begin{align*}
P(0) & = m_0 \\
P(\alpha) & = m_0 + m_1 \alpha + m_2 \alpha^2 + \cdots + m_{k-1} \alpha^{k-1} \\
P(\alpha^2) & = m_0 + m_1 \alpha^2 + m_2 \alpha^4 + \cdots + m_{k-1} \alpha^{2(k-1)} \\
\vdots \\
P(\alpha^{q-1}) & = m_0 + m_1 \alpha^{q-1} + m_2 \alpha^{2(q-1)} + \cdots + m_{k-1} \alpha^{(k-1)(q-1)}.
\end{align*}
\] (2)

Proposition. There is a unique solution \( m = (m_0, m_1, \ldots, m_{k-1}) \) for any \( k \) choices of the \( q \) linear equations.

This follows from the proposition in the Mathematical Background section, but Reed and Solomon’s original proof offers some insight in their thinking:
2.1 Original Conception

Approaches to the RS Code

Proof. Vandermonde proof

Without loss of generality, suppose we take the first $k$ equations to solve. Then we wish to solve the set of equations:

\[
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
1 & \alpha & \alpha^2 & \cdots & \alpha^{k-1} \\
1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{k-1} & \alpha^{2(k-1)} & \cdots & \alpha^{(k-1)(k-1)}
\end{bmatrix}
\begin{bmatrix}
m_0 \\
m_1 \\
m_2 \\
\vdots \\
m_{k-1}
\end{bmatrix}
= 
\begin{bmatrix}
P(0) \\
P(\alpha) \\
P(\alpha^2) \\
\vdots \\
P(\alpha^{k-1})
\end{bmatrix}.
\]

We can show that the system of equations have a unique solution for \{\(m_0, m_1, \ldots, m_{k-2}, m_{k-1}\)} by computing the determinant of the coefficient matrix

\[
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
1 & \alpha & \alpha^2 & \cdots & \alpha^{k-1} \\
1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{k-1} & \alpha^{2(k-1)} & \cdots & \alpha^{(k-1)(k-1)}
\end{bmatrix}.
\]

But the determinant of the matrix reduces to that of a Vandermonde matrix, and we know that the determinant of a Vandermonde matrix is nonsingular.

This result gives the RS code very significant error-correcting capabilities. Suppose that $t$ unknown coordinates of the $q$ coordinates of the received codeword was corrupted during transmission. There are \(\binom{q}{k}\) choices for the system of equations to solve, with \(\binom{t+k-1}{k}\) of them leading to an incorrect solution (because an incorrect solution will arise when at least 1 out of the $k$ choices of the equations is incorrect). When the probability of an error occurring during transmission is small, we can assume that the solution with the majority count will be the correct solution. That is, as long as \(\binom{t+k-1}{k} < \binom{q-t}{k}\), we will obtain a correct solution. Now,

\[
\binom{t+k-1}{k} < \binom{q-t}{k} \iff \frac{(t+k-1)!}{(t-1)!(k)!} < \frac{(q-t)!}{(q-t-k)!k!} \iff (t+k-1)(t+(k-1)-1)\cdots(t+(k-k)-1) < (q-t)((q-1)-t)\cdots((q-k)-t) \iff t+k-1 < q-t \iff 2t < q-k+1 \iff t < \frac{q-k+1}{2}.
\]

So the code can correct up to $t = \left\lfloor \frac{q-k+1}{2} \right\rfloor$ errors. By the Singleton Bound, this is the best possible error correction capability for any code of the same length and dimension. It follows that RS Codes are MDS codes.

The true merit of the RS Code is in its additional ability to deal with erasures. Suppose that in the decoding process, a certain threshold $\varphi_C$ is implemented such that signals below the threshold are decoded into zeros and signals above the threshold are decoded into ones. This is called the hard decoding method. In certain cases, however, it may be better to simply omit a codeword coordinate rather than to decode it to a value with significant probability of being incorrect. This is called soft decoding. Suppose then, we
allow for “erasures” of the coordinates of codewords. Since only \( k \) uncorrupted linear equations are needed to retrieve the information bits, up to \( q - k \) coordinates may be erased. Let \( v \) denote the number of erasures.

As long as \( \binom{t + k - 1}{k} < \binom{q - v - t}{k} \), errors can be corrected. Similar to before, we have:

\[
\frac{(t + l - 1)!}{k!(t - 1)!} < \frac{(q - v - t)!}{k!(q - k - v - t)!}
\]

\[
\iff 2t + v < q - k + 1.
\]

Thus, a Reed-Solomon code can correct up to \( t \) errors and \( v \) erasures, as long as the above inequality is satisfied.

### 2.2 Generator Polynomials

The generator polynomial method to the construction of the RS Code is the most commonly used approach today \([1]\).

For a cyclic code \( C \), we interpret \( c \in C \) as such:

\[
(c_0, c_1, \ldots, c_{n-2}, c_{n-1}) \mapsto c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}.
\]

**Proposition.** A cyclic code can always be defined using a generator polynomial \( g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_{n-k} x^{n-k} \). A vector \( c \) is a codeword in the code generated by \( g(x) \) if and only if the corresponding code polynomial \( c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1} \) is a multiple of \( g(x) \).

From the proposition, we have that for \( k \) information symbols \( m = (m_0, m_1, \ldots, m_{k-1}) \) and its corresponding polynomial \( m(x) \), any codeword \( c(x) \) is generated by the following relation:

\[
c(x) = m(x)g(x).
\]

The cyclic Reed-Solomon code with symbols from \( \text{GF}(q) \) have length \( q - 1 \), with one fewer coordinate than the codewords generated via Reed and Solomon’s original construction. Note that the field \( \text{GF}(256) \) is a popular choice of the finite field because each of the 256 field elements can be represented as a binary byte (8 bits).

**Construction**

We wish to construct a \( t \)-error correcting RS code of length \( q - 1 \) with entries in \( \text{GF}(q) \). Take a primitive element \( \alpha \in \text{GF}(q) \). We construct the generator polynomial \( g(x) \) such that \( g \) has \( 2t \) consecutive powers of \( \alpha \) as its roots:

\[
g(x) = \prod_{i=1}^{2t} (x - \alpha^i).
\]

It follows that a code polynomial can have degree \( 2t \leq d \leq q - 2 \). The dimension of the code is then \( k = q - 2t - 1 \). The Maximum Distance Separation relation applies here as well:

\[
\text{Error correction capacity} = \left\lfloor \frac{\text{length} - \text{dimension}}{2} \right\rfloor
\]

This method of construction provides a very convenient way for determining the correctness of the received word. One must simply make sure that the code polynomial has the necessary roots as dictated by the construction.
2.3 Fourier Transforms

Definition. The Fourier Transform is a map that takes a complex-valued function of a real variable to another. The domain of the map is called the time domain and the image of the map is called the frequency domain.

The notion of time domain and frequency domain is ambiguous in a finite field setting, but the operation itself is well-defined and leads to useful consequences [7].

Definition. Take $\alpha \in GF(q)$ a primitive element in the finite field.

The Galois field Fourier transform of a vector $c = (c_0, c_1, \ldots, c_{n-1})$ is the defined as follows:

$$F\{ (c_0, c_1, \ldots, c_{n-1}) \} = (C_0, C_1, \ldots, C_{n-1}),$$

where

$$C_j = \sum_{i=0}^{n-1} c_i \alpha^{ij}.$$ 

Theorem. Over GF(q), a field of characteristic $p$, a vector and its spectrum are related by

$$C_j = \sum_{i=0}^{n-1} \alpha^{ij} c_i$$

and

$$c_i = \frac{1}{n \mod p} \sum_{j=0}^{n-1} \alpha^{-ij} C_j.$$ 

Proof. In any field, we have the relation

$$x^n - 1 = (x - 1)(x^{n-1} + x^{n-2} + \cdots + 1).$$

By definition, $\alpha^r$ is a root of the polynomial on the left for $1 \leq r \leq n - 1$. Therefore, we have that $\alpha^r$ is a root of the last term for all $r \neq 0$ modulo $n$.

This is equivalent to

$$\sum_{j=0}^{n-1} \alpha^{rj} = 0 \text{ for } r \neq 0 \mod n.$$ 

and when $r = 0$,

$$\sum_{j=0}^{n-1} \alpha^{rj} = n \mod p$$

which is not zero if $n$ is not a multiple of $p$.

We thus obtain:

$$\sum_{j=0}^{n-1} \alpha^{-ij} \sum_{k=0}^{n-1} c_k = \sum_{k=0}^{n-1} c_k \sum_{j=0}^{n-1} \alpha^{(k-i)j} = (n \mod p)c_i.$$ 

In addition, if the field has characteristic $p$, there exists an integer $s$ such that $q - 1 = p^s - 1 \equiv 0 \mod n$. Consequently, $n$ is not a multiple of $p$, and $n \mod p \neq 0$.

This proves the claim. \qed

Theorem. 1. The polynomial $c(x) := c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$ has a root at $\alpha^j$ iff the $j$th spectral component $C_j = 0$. 


The polynomial $C(X) := C_0 + C_1 x + \cdots + C_{n-1} x^{n-1}$ has a root at $\alpha \iff$ the $i$th time component $c_i = 0$. 

**Proof.**

1. This follows from the equation

$$c(\alpha^i) = \sum_{i=0}^{n-1} c_i \alpha^{ij} = C_j.$$

2. Similarly, from

$$c_i = \frac{1}{n \mod p} \sum_{j=0}^{n-1} \alpha^{ij} C_j$$

the claim follows. 

**Corollary.** A word polynomial has $2t$ consecutive powers of $\alpha$ as roots iff the spectrum of the corresponding word has $2t$ consecutive zero coordinates.

From the corollary we see that the GFFT method is a dual to the generator polynomial method. The GFFT method thus leads to a series of efficient encoders and decoders not available by generator polynomials alone, using the powerful vocabulary of the Fourier Transform.

## 3 Encoding and Decoding of RS Code

The current view of the RS codes considers the RS codes to be a special case of the BCH codes\(^1\) of length $n = q - 1$. This perspective coupled with the perspectives previously introduced lead to a series of efficient encoding and decoding algorithms, which the author will discuss here.

### Encoding

Let the polynomial $d(x) = d_0 + d_1 x + \cdots + d_{k-1} x^{k-1}$ be a data polynomial with each $d_i \in \text{GF}(q)$ (each codeword $c \in C$ is constructed from each choice of $d$, and the entire code is constructed by varying over all $d$). One way of encoding RS codes is to compute $c(x) = d(x)g(x)$, where $g$ is the generator polynomial; however, this is not a systematic method since the $k$ data symbols are not explicitly present in the codeword. Hence this leads to an extra step being required in the decoding process to extract information from the codeword. A way to improve this inadequacy is to encode by using the Cauchy Matrix.

Let $d(x)$ be as above. A codeword can be systematically encoded by adding $n - k$ parity check symbols to the data symbols:

$$c = (c_0, c_1, \ldots, c_{n-1}) = (p_0, p_1, \ldots, p_{n-k-1}, d_0, d_1, \ldots, d_{k-1})$$

Define the systematic generator matrix $G$:

$$G = (I \ A).$$

where $I$ is the $k \times k$ identity matrix and $A$ is a $k \times (n - k)$ matrix, known as a Cauchy Matrix. $A$ is constrained by these definitions:

$$A_{i,j} = \frac{u_i v_j}{x_i + y_j}, \ 0 \leq i \leq k - 1, \ 0 \leq j \leq n - k - 1$$

\(^1\)see [2]
The parity check vector \( p = (p_0, \ldots, p_{n-k-1}) \) is
\[
p = dA
\]
\[
p_j = \sum_{i=0}^{k-1} d_i A_{i,j}
\]
\[
= v_j \sum_{i=0}^{k-1} d_i u_i, \quad 0 \leq j \leq n - k - 1.
\]

Decoding

A commonly used decoding process for polynomial-generated RS codes is *Syndrome-Based Decoding*.

Define an error polynomial as \( e(x) = e_0 + e_1 x + \cdots + e_{n-1} x^{n-1} \).

An input polynomial for the decoder is then:
\[
v(x) = c(x) + e(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1},
\]
where the \( v_i \)'s are the \( i \)th components of the received word. We evaluate the polynomial at the roots of the generator polynomial \( \alpha, \ldots, \alpha^{2t} \):
\[
v(\alpha^j) = c(\alpha^j) + e(\alpha^j)
\]
\[
= e(\alpha^j) = \sum_{i=0}^{n-1} e_i \alpha^{ij}, \quad j = 1, 2, \ldots, 2t.
\]

We define the \( j \)th *syndrome* of \( v \) as the Fourier transformation:
\[
S_j := v(\alpha^j) = \sum_{i=0}^{n-1} v_i \alpha^{ij}, \quad j = 1, 2, \ldots, 2t.
\]

Suppose that \( 0 \leq \nu \leq t \) errors occur in unknown locations \( i_1, i_2, \ldots, i_\nu \). Then, \( e(x) = e_{i_1} x^{i_1} + \cdots + e_{i_\nu} x^{i_\nu} \). We wish to find the \( i \)’s and the \( e_i \)’s.

Define \( Y_l := e_{i_l} \) for \( l = 1, 2, \ldots, \nu \) and \( X_l = \alpha^{i_l} \) for \( l = 1, 2, \ldots, \nu \). From evaluating the syndromes we obtain a system of equations with \( \nu \) unknowns:
\[
S_1 = Y_1 X_1 + Y_2 X_2 + \cdots + Y_\nu X_\nu
\]
\[
\vdots
\]
\[
S_j = Y_1 X_1^j + Y_2 X_2^j + \cdots + Y_\nu X_\nu^j
\]
\[
\vdots
\]
\[
S_{2t} = Y_1 X_1^{2t} + Y_2 X_2^{2t} + \cdots + Y_\nu X_\nu^{2t}.
\]
By the construction of the syndromes, at least 1 solution exists. The decoding process is completed by finding a solution with the least amount of errors.

4  RS Codes in Standard Usage

The author will conclude the paper with an extended overview of a single application of the Reed-Solomon Codes.

Compact Discs

RS Codes are extensively used in the error-correcting used in reading and writing Compact Discs (CDs). When sound is recorded, sound-waves are converted from an analog source to a digital medium by a process called “sampling”. The most commonly used sampling rate is 44.1 Khz, a rate chosen to be compatible with an already existing standard for video recording. The amplitude of a waveform is sampled at an instance in time and is is assigned to a binary string of length 16, one for the left and one for the right channel of sound. Each vector of length 16 is cut in half, thereby producing 4 bytes of data per second. When the sound is encoded, a combination of two RS codes called the Cross-Intervealed Reed Solomon Code (CIRC) is used, with the design to correct long burst errors. Now we are ready to describe the process of encoding.

4.0.1 Encoding

Note that the encoding method was designed to combat burst errors. Six samples of each of the four bytes are grouped into a frame consisting of 24 bytes. Denote the ith pair of bytes in the left channel as $L_i$ and its right channel counterpart as $R_i$. Then a frame consists of the following pattern:

$$L_1R_1L_2R_2\cdots L_6R_6.$$ 

The frame goes through a series of permutations before it is encoded to a master CD.

1. The Spread

The odd-numbered samples $L_1R_1, L_3R_3, L_5R_5$ are grouped with the even-numbered samples $\hat{L}_2\hat{R}_2, \hat{L}_4\hat{R}_4, \hat{L}_6\hat{R}_6$ from two frames ahead. Thus the new arrangement consists of the following pattern:

$$L_1R_1\hat{L}_2\hat{R}_2L_3R_3\hat{L}_4\hat{R}_4L_5R_5\hat{L}_6\hat{R}_6.$$ 

2. Internal Rearrangement

The new frame is then arranged so that the even and odd bytes are separated:

$$L_1L_3L_5R_1R_3R_5\hat{L}_2\hat{R}_2\hat{L}_4\hat{R}_4\hat{L}_6\hat{R}_6.$$ 

The motivation behind this is to separate samples as far apart as possible inside the new frame.

3. Adding redundancy via $C_1$

This 24-byte message, an element in $\mathbb{F}_{256}^{24}$ is then encoded by an encoder using a $[28,24,5]$ Reed-Solomon Code, which we will henceforth call $C_1$. The encoder adds four bytes of redundancy, namely $P_1$ and $P_2$ such that the new frame consists of the following pattern:

$$L_1L_3L_5R_1R_3R_5\hat{L}_2\hat{R}_2\hat{L}_4\hat{R}_4\hat{L}_6\hat{R}_6.$$ 

We see that this further separates the odd-numbered samples from the even-numbered samples.
4. 4-frame delay interleaving

Each code $c_i \in C_1$ is interleaved into a matrix with 28 rows and a large but determined number of columns in the following manner:

Row 1: The first byte of $c_1$ is placed in column 1, the first byte of $c_2$ in column 2, etc.

Row 2: Begin with four bytes of 0. Second byte of $c_1$ is placed in column 5, second byte of $c_2$ is placed in column 6, etc.

...;

Row 28: Begin with $4 \times (28 - 1) = 112$ bytes of 0. 28th byte of $c_1$ is placed in column 113, 28th byte of $c_2$ is placed in column 114, etc.

In addition, we pad all the rows except for the last with 0’s so that the number of columns are consistent:

Example.

\[
\begin{array}{cccccccccccc}
  c_{1,1} & c_{2,1} & c_{3,1} & c_{4,1} & c_{5,1} & c_{6,1} & c_{7,1} & c_{8,1} & c_{9,1} & c_{10,1} & c_{11,1} & c_{12,1} & c_{13,1} & \cdots \\
  0 & 0 & 0 & 0 & c_{1,2} & c_{2,2} & c_{3,2} & c_{4,2} & c_{5,2} & c_{6,2} & c_{7,2} & c_{8,2} & c_{9,2} & \cdots \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{1,3} & \cdots \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{1,4} & \cdots \\
  \vdots
\end{array}
\]

This will leave the original codeword diagonally on the matrix with slope $-\frac{1}{4}$.

5. $C_2$ and interleaving

Each column in the matrix $c \in \mathbb{F}_2^{28}$ is then encoded by a $[32, 28, 5]$ RS code, denoted by $C_2$. This thusly generates a group of codewords, each consisting of 32 bytes.

The codewords are then rearranged so that the odd-numbered symbols are matched up with the even-numbered symbols from the next codeword, and so on. The motivation for this interleaving again comes from the desire to break up other short bursts that may be present even after the 4-delay interleaving. After the rearrangement, the codewords are written consecutively in one long stream, which is then redivided into a segment of 32 bytes; at this point a 33rd byte is added that includes control and display information that the CD player requires to display the playing time, composer, and the title of the piece.

6. Imprinting

A pit is an indentation in the CD that the laser reads and interprets as data. A land is the space between the pits. Each bit is of length 0.3 $\mu$m when it is imprinted on the CD. A land-to-pit or pit-to-land transition is marked with a 1, and the pit and land itself is marked with a string of 0’s. There are physical limitations to CD writing, so each land or pit must be between 0.9 and 3.3 $\mu$m in length. Hence a pair of 1’s must be separated by at least two 0’s and at most ten 0’s.

For this reason, a look-up map from bytes to 14-length strings called the eight-to-fourteen modulation (EFM) map is implemented for the actual imprinting process.
Example. A part of the EFM Table

<table>
<thead>
<tr>
<th>Data</th>
<th>Code</th>
<th>Data</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>01000100100010</td>
<td>114</td>
<td>10010010000100</td>
</tr>
<tr>
<td>101</td>
<td>00000000100001</td>
<td>115</td>
<td>01000000100010</td>
</tr>
<tr>
<td>102</td>
<td>01000000100010</td>
<td>116</td>
<td>01000010000010</td>
</tr>
<tr>
<td>103</td>
<td>00100100100001</td>
<td>117</td>
<td>00000010000010</td>
</tr>
<tr>
<td>104</td>
<td>01001001000001</td>
<td>118</td>
<td>00010001000010</td>
</tr>
<tr>
<td>105</td>
<td>10000001000001</td>
<td>119</td>
<td>01000010000010</td>
</tr>
<tr>
<td>106</td>
<td>10010001000001</td>
<td>120</td>
<td>01001000100001</td>
</tr>
<tr>
<td>107</td>
<td>10001001000001</td>
<td>121</td>
<td>00001001001000</td>
</tr>
<tr>
<td>108</td>
<td>01000001000001</td>
<td>122</td>
<td>10010000000010</td>
</tr>
<tr>
<td>109</td>
<td>00000001000001</td>
<td>123</td>
<td>10001000000010</td>
</tr>
<tr>
<td>110</td>
<td>00010010000001</td>
<td>124</td>
<td>01000000000001</td>
</tr>
<tr>
<td>111</td>
<td>00100001000001</td>
<td>125</td>
<td>00001000000001</td>
</tr>
<tr>
<td>112</td>
<td>10000000100001</td>
<td>126</td>
<td>00010000000010</td>
</tr>
<tr>
<td>113</td>
<td>10000100000001</td>
<td>127</td>
<td>00100000000001</td>
</tr>
</tbody>
</table>

The strings are imprinted on the CD in succession; therefore there may be instances when two strings satisfying the criterion defined by the proposition do not in conjunction satisfy the criterion. For instance, 10010000000100 and 00000000010001 are allowable strings, but 10010000000100000000010001 is not. Therefore, merge bits defined as 001 must be appended to the end of the 14-length strings, resulting in a string of length 17.

At the end of the $33 \times 17 = 561$ bits, 24 synchronization and 3 merge bits are appended. The result is that for each frame of six samples of information 588 bits are written onto the CD.

### 4.0.2 Decoding

Decoding takes place as to exactly reverse the process of encoding. It is implemented in two main steps:

1. Decoding with $C_2$

   The first step entails reading the disc, removing synchronization, control and display bits, and demodulating the 17-bit codes into their respective byte formats. The data is presented as a stream of bytes. The stream is divided up into a segment of 32-bytes. Recall that by construction each 32-byte segment contains 16 bytes of the odd-indexed bytes from a codeword (possibly with errors) and 16 bytes of the even-indexed bytes from the next codeword. Because $C_2$ is a $[32, 28, 5]$ code, it is able to correct up to 2 errors. But in this case, correction is only made when a single error exists; otherwise $C_2$ is used to only detect multiple errors. The reasoning behind this can be summarized by evaluating the probability of error.

   **Claim.** When correcting for 1 error, $C_2$ can detect 2 or 3 errors. The probability that $C_2$ will not detect greater or equal to four errors when using it to correct for single errors is approximately $1.9 \times 10^{-6}$.

   **Proof.** Assume that all codewords are equally likely. The probability that $C_2$ will not detect $\geq 4$ errors is approximately the number of codewords in spheres of radius 1 about each codeword divided by the total number of codewords. Hence we have that the probability is:

   $$\frac{256^{28}[1 + 32(256 - 1)]}{256^{32}} = \frac{8161}{256^4} \approx 1.9 \times 10^{-6}.$$
Claim. When correcting for 2 errors, the probability that three or more errors will be undetected by $C_2$ is approximately $7.5 \times 10^{-3}$.

Proof. Assume again that all codewords are equally likely to occur. The probability that $C_2$ will not detect three or more errors is approximately the number of codewords in spheres of radius 2 about each codeword divided by the total number of codewords. Hence we have that the probability is:

$$\frac{256^{28}[1 + \binom{32}{1}(256 - 1) + \binom{32}{2}(256 - 1)^2]}{256^{32}} \approx 7.5 \times 10^{-3}.$$ 

2. Decoding with $C_1$

There are three possibilities that can result from step 1 of the decoding process, each one resulting from the determination of the decoder.

(a) No Errors
   If the decoder determines that there are no errors, the 28-byte vector is extracted from the 32-byte vector and is passed on to step 2.

(b) Single Error
   Error is corrected, and the 28-byte vector is passed on to step 2.

(c) Two or more errors
   It passes on a 28-byte vector with every component marked as erasures.

From the array as described in (4), the diagonal elements along the line of slope $-1/4$ are extracted. There are two schemas that the decoding algorithm uses:

Schema 1: In schema 1, $C_1$ is used only to correct erasures. $C_1$ can correct up to four erasures.

Schema 2: In schema 2, $C_1$ is used to correct one error and two erasures.

Decoding methods for this step have not been standardized and the choice of method is largely dependant on the manufacturer of the decoding device. The choice may depend upon many considerations such as power consumption and price.

3. Interpolation and Silencing

Even after the error-detection and correction algorithms implemented in the decoding process, errors may still remain in the code. One additional step better suits the data for output. This step is conditional on the existence of burst errors.

Case 0: If there are no burst errors in a neighborhood of a sample, then linear interpolation is used to “conceal” the error.

Case 1: If there are burst errors in a neighborhood of a sample, the neighbors of a sample are unreliable for linear interpolation. In this case, a procedure called “muting” is used. This procedure entails gradually weakening reliable samples 32 samples before the burst and gradually strengthening the next 32 reliable samples after the burst, with the burst altered to have zero values. This muting process happens over a few milliseconds and it is essentially inaudible [1].
References


