Reed-Muller Codes

An important class of linear block codes rich in algebraic and geometric structure is the class of Reed-Muller codes, which includes the Extended Hamming code. It was discovered by Muller in 1954, and the first decoding algorithm was devised by Reed also in 1954.

For each positive integer \( m \) and each integer \( r \) with \( 0 \leq r \leq m \), there is an \( r \)th order Reed-Muller Code \( \mathcal{RM}(r, m) \), which is a binary linear code of parameters

\[
\left[ 2^m, \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r}, 2^{m-r} \right].
\]

Note that \( \mathcal{RM}(1, m) \) code is a small code \((n = 2^m)\) with a large minimum distance \((n = 2^{m-1})\), so a good decoding algorithm is the most elementary: for each received word \( w \), find a codeword in \( \mathcal{RM}(1, m) \) closed to \( w \). This can be done very efficiently. In fact, \( \mathcal{RM}(1, 5) \) was used by Mariner 9 to transmit black and white pictures of Mars back to the Earth in 1972.

\[\clubsuit\] Definition. There are many ways to define the Reed-Muller codes. First we choose an inductive definition. Later, we shall explain how to construct Reed-Muller codes using the Kronecker product.

The zeroth order Reed-Muller codes \( \mathcal{RM}(0, m) \), for \( m \geq 0 \), are defined to be the repetition codes \( \{0, 1\} \) of length \( 2^m \).

\[\mathcal{RM}(0, 0) = \{0, 1\} = \mathbb{K}, \quad \mathcal{RM}(0, 1) = \{00, 11\}, \quad \mathcal{RM}(0, 2) = \{0000, 1111\}, \ldots.\]

The first order Reed-Muller codes \( \mathcal{RM}(1, m) \) are binary codes defined for all integers \( m \geq 1 \), recursively by:

(i) \( \mathcal{RM}(1, 1) = \{00, 01, 10, 11\} = \mathbb{K}^2 \).

(ii) For \( m \geq 1 \), \( \mathcal{RM}(1, m + 1) = \{ (u, u) : u \in \mathcal{RM}(1, m) \} \cup \{ (u, u + 1) : u \in \mathcal{RM}(1, m) \} \).

\[\spadesuit\] Example. \( \mathcal{RM}(1, 2) = \{ (x, y + x) : x \in \{00 01 10 11\}, \ y \in \{00 11\}\}. \]

\[\mathcal{RM}(1, 3) = \left\{ \begin{array}{c} 00 00 00 00, \ 00 00 11 11, \ 01 01 01 01, \ 01 01 11 10, \\
10 10 01 10, \ 10 10 01 01, \ 11 11 11 11, \ 11 11 00 00, \\
00 11 00 11, \ 00 11 11 00, \ 01 10 01 01, \ 01 10 10 01, \\
10 01 10 01, \ 10 01 01 10, \ 11 00 11 00, \ 11 00 00 11. \end{array} \right\}. \]

\[\spadesuit\] Dimension. For \( m \geq 1 \), the Reed-Muller code \( \mathcal{RM}(1, m) \) is a binary \([2^m, m + 1, 2^{m-1}]\) linear code, in which every codeword except \( 0 \) and \( 1 \) has weight \( 2^{m-1} \).

\( \mathcal{RM}(m - 1, m) \) consists of all binary words of length \( 2^m \) that have even weight.

California State University, East Bay
Theorem 1. The codes $\mathcal{RM}(r, m)$ and $\mathcal{RM}(m-r-1, m)$ are dual of each other.

Proof. Let $B_{r,m}$ and $B_{m-r-1,m}$ be orthonormal bases of $\mathcal{RM}(r, m)$ and and $\mathcal{RM}(m-r-1, m)$, respectively. Then $B_m$ contains \( \begin{pmatrix} m \\ 0 \end{pmatrix} + \begin{pmatrix} m \\ 1 \end{pmatrix} + \cdots + \begin{pmatrix} m \\ r \end{pmatrix} \) linearly independent vectors and $B_{m-r-1,m}$ contains \( \begin{pmatrix} m \\ 0 \end{pmatrix} + \begin{pmatrix} m \\ 1 \end{pmatrix} + \cdots + \begin{pmatrix} m \\ m-r-1 \end{pmatrix} \) linearly independent vectors. The facts that
\[
2^m = \left( \begin{pmatrix} m \\ 0 \end{pmatrix} + \begin{pmatrix} m \\ 1 \end{pmatrix} + \cdots + \begin{pmatrix} m \\ r \end{pmatrix} \right) + \left( \begin{pmatrix} m \\ r+1 \end{pmatrix} + \begin{pmatrix} m \\ r+2 \end{pmatrix} + \cdots + \begin{pmatrix} m \\ m \end{pmatrix} \right)
\]
and
\[
\begin{pmatrix} m \\ 0 \end{pmatrix} = \begin{pmatrix} m \\ m \end{pmatrix}, \quad \begin{pmatrix} m \\ 1 \end{pmatrix} = \begin{pmatrix} m \\ m-1 \end{pmatrix}, \quad \cdots, \quad \begin{pmatrix} m \\ m-r-1 \end{pmatrix} = \begin{pmatrix} m \\ r+1 \end{pmatrix}
\]

imply that the vector space $K^m$ is generated by the union of two orthonormal bases $B_m$ and $B_{m-r-1,m}$. Hence the code $\mathcal{RM}(m-r-1, m)$ is the dual of $\mathcal{RM}(r, m)$.

- According to this theorem, $\mathcal{RM}(r, 2r+1)$ codes are self dual. Thus $\mathcal{RM}(1, 3)$ and $\mathcal{RM}(2, 5)$ are self-dual.

☆ Generator Matrix. Rather than using the recursive description of the code, we will give a recursive construction for the generator matrix of $\mathcal{RM}(r, m)$ code, which we will denote by $G(r, m)$.

For $r = 0$, define $G(0, m) = [1 1 \ldots 1 1]$, and for $r = m$, define
\[
G(m, m) = \begin{bmatrix} G(m-1, m) \\
0 & 0 & \ldots & 0 & 1 \end{bmatrix}.
\]

If $G(r, m)$ is a generator matrix for $\mathcal{RM}(r, m)$, then a generator matrix for $\mathcal{RM}(r, m+1)$ is
\[
G(r, m+1) = \begin{bmatrix} G(r, m) & G(r, m) \\
0 & G(r-1, m) \end{bmatrix}
\]

☆ Examples.
$G(0, 1) = [1 1], \; G(0, 2) = [1 1 1 1], \; G(0, 3) = [1 1 1 1 1 1 1 1],$

\[
G(1, 1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; G(1, 2) = \begin{bmatrix} G(1, 1) & G(1, 1) \\
\theta & G(0, 1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1 \end{bmatrix},
\]

\[
G(2, 2) = \begin{bmatrix} G(1, 2) \\
0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1111 \\
0101 \\
0011 \\
0001 \end{bmatrix}, \; G(1, 3) = \begin{bmatrix} G(1, 2) & G(1, 2) \\
\theta & G(0, 2) \end{bmatrix} = \begin{bmatrix} 1111 & 1111 \\
0101 & 0101 \\
0011 & 0011 \\
0000 & 1111 \end{bmatrix}, \; G(2, 3) = \begin{bmatrix} G(2, 2) & G(2, 2) \\
\theta & G(1, 2) \end{bmatrix} = \begin{bmatrix} 11111111 \\
01010101 \\
00110011 \\
00010001 \\
00001111 \\
00000101 \\
00000011 \end{bmatrix}, \; G(3, 3) = \begin{bmatrix} G(2, 3) \\
00000001 \end{bmatrix} = \begin{bmatrix} 11111111 \\
01010101 \\
00110011 \\
00010001 \\
00001111 \\
00000101 \\
00000011 \end{bmatrix}.
\]
Corollary 1.1. The parity-check matrix of $G(r, m)$ is $G(m - r - 1, m)^t$.

Proof. According to Theorem 1, the generator $G(r, m)$ of $RM(r, m)$ is orthogonal to the generator $G(m - r - 1, m)$ of $RM(m - r - 1, m)$. This clearly implies that the parity-check matrix of $G(r, m)$ is $G(m - r - 1, m)^t$.

Theorem 2. For $r > 0$, the code $RM(r, m - 1)$ is contained in $RM(r, m)$.

Proof. We prove by induction.

We start with

$$G(1, m + 1) = \begin{bmatrix} G(1, m) & G(1, m) \\ 0 & G(0, m) \end{bmatrix}.$$ 

(Since the top row entries of $[G(1, m) \ G(1, m)]$ are all ones, we conclude that $RM(0, m) = [0 \ 1]$ is contained in $RM(1, m)$.

In general, since $G(r - 1, m - 1)$ is a sub-matrix of $G(r - 1, m)$ and $G(r - 2, m - 1)$ is a sub-matrix of $G(r - 1, m - 1)$ we have obviously

$$G(r - 1, m) = \begin{bmatrix} G(r - 1, m - 1) & G(r - 1, m - 1) \\ 0 & G(r - 2, m - 1) \end{bmatrix}$$

a sub-matrix of $G(r, m)$. Thus $RM(r, m - 1)$ is contained in $RM(r, m)$.

Corollary 2.1. For any integer $m \geq 2$, the code $RM(m - 2, m)$ is an extended Hamming code and $RM(1, m)$ is its dual code.

Proof. Recall that any $[2^m, 2^m - m - 1, 3]$ code is a Hamming code and any $[2^m, 2^m - m - 1, 4]$ code is an extended Hamming code. Since $RM(m - 2, m)$ is a $[2^m, 2^m - m - 1, 4]$ code, we conclude that that it is an extended Hamming code. The remainder of the proof follows from Theorem 1.

An immediate result of this corollary is that $RM(1, 3)$ is a self dual code.

★ Kronecker Product. The Kronecker product also called tensor product or the direct product of two matrices $A = (a_{ij})$ and $B$ is defined as follows:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \ldots & a_{1n}B \\ a_{21}B & a_{22}B & \ldots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \ldots & a_{mn}B \end{pmatrix}.$$ 

For example

$$G(1, 1) \otimes G(1, 1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = G(1, 2).$$

Thus by using the Kronecker product, we may build generators for Reed-Muller codes. Here is how we define the generators:

Let $G_0 = [0 \ 1]$, $Z_1 = Z$, $G(0, 0) = [1]$, and $J = [1 \ 1]$. Then

\textit{California State University, East Bay}
(1) \( G(0, 1) = J \otimes G(0, 0) = [1, 1] \) and \( Z_2 = Z_1 \otimes Z_1 = [0, 0, 0, 1] \).

(2) \( G(0, 2) = [1, 1, 1, 1] \), \( G(1, 2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \), and \( G(22) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \).

(3) \( Z_3 = Z \otimes Z_2 = [0, 0, 0, 0, 0, 0, 1] \), \( G(0, 3) = J \otimes G(02) = [1, 1, 1, 1, 1, 1, 1, 1] \).

\[
G(1, 3) = \begin{bmatrix} J \otimes G(1, 2) \\ Z \otimes G(0, 2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.
\]

Note that the above matrix \( G(1, 3) = G(3 - 2, 3) \) generates an extended self-dual Hamming code.

For example, \( G(0, 1) = [1, 1] \), \( G(0, 2) = [1, 1, 1, 1] \), \( G(0, 3) = [1, 1, 1, 1, 1, 1, 1, 1] \).

\[
G(1, 1) = \begin{bmatrix} 11 \\ 01 \end{bmatrix}, \quad G(1, 2) = \begin{bmatrix} G(1, 1) \\ G(1, 1) \theta G(0, 1) \end{bmatrix} = \begin{bmatrix} 11 \\ 01 \\ 01 \\ 00 \end{bmatrix},
\]

\[
G(2, 2) = \begin{bmatrix} G(1, 2) \\ 0001 \end{bmatrix} = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \\ 0001 \end{bmatrix}, \quad G(1, 3) = \begin{bmatrix} G(1, 2) \\ G(2, 2) \theta G(0, 2) \end{bmatrix} = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \\ 0001 \\ 0001 \\ 0001 \\ 0000 \\ 1111 \end{bmatrix},
\]

\[
G(2, 3) = \begin{bmatrix} G(2, 2) \\ G(2, 2) \theta G(1, 2) \end{bmatrix} = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \\ 0001 \\ 0001 \\ 0001 \\ 0000 \\ 1111 \\ 0000 \\ 0001 \\ 0000 \\ 0000 \\ 0000 \\ 0000 \\ 0000 \end{bmatrix}, \quad \text{and} \quad G(3, 3) = \begin{bmatrix} G(2, 3) \\ 00000001 \end{bmatrix} = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \\ 0001 \\ 0001 \\ 0001 \\ 0000 \\ 1111 \\ 0000 \\ 0001 \\ 0000 \\ 0000 \\ 0000 \\ 0000 \\ 0000 \end{bmatrix}.
\]

\[ \blacktriangleright \textbf{Reed Decoding} \] This decoding algorithm is quite simple. We explain the algorithm with the \( RM(1, 3) \) code which is also a self-dual extended Hamming code.

Since the dimension of \( RM(1, 3) \) is four and the generator matrix \( RM(1, 3) \) has four rows, we conclude that the rows of

\[
G(1, 3) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}
\]

form a basis of the one-error-correcting code \( RM(1, 3) \).

\[ \textbf{Example 1.} \] Suppose the word \( w = [0, 1, 0, 1, 1, 1, 0] \) was received. Clearly there is at least one error involved (Why?)

\[ \textbf{Step 1.} \] We express \( w \) as a linear combination of \( v_1, v_2, v_3, v_4 \). Thus

\[
w = [0, 1, 0, 1, 1, 1, 0] = \lambda_1 v_1 + \lambda_2 v_2 + \lambda_3 v_3 + \lambda_4 v_4
= [\lambda_1, \lambda_1 + \lambda_2, \lambda_1 + \lambda_3, \lambda_1 + \lambda_2 + \lambda_3, \lambda_1 + \lambda_4, \lambda_1 + \lambda_2 + \lambda_4, \lambda_1 + \lambda_3 + \lambda_4, \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4] .
\]
Step 2. If there is no error in the first digit, then $\lambda_1 = 0$, $\lambda_2 = 1$, $\lambda_3 = 0$, $\lambda_4 = 1$. But in the sixth column we have $\lambda_2 + \lambda_4 = 1 + 1 = 1$, which is a contradiction. Thus the sixth digit of $w$ must be changed from 1 to zero. Note that the word $v = [0 1 0 1 1 0 1 0]$ is a codeword.

Example 2. Suppose the word $w = [1 0 1 1 0 0 1 1]$ was received with at most one digit-error.

Solution. If there is no error in the first digit, then $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 0$, $\lambda_4 = 1$. But we have contradictions in the sixth, seventh, and eight columns. Since the word $v = [0 0 1 1 0 0 1 1]$ is a codeword, we conclude that the error occurred in the first digit.

Exercise. Suppose the following words were received using the $\mathcal{RM}(1,3)$ code:

$$w_1 = [0 1 0 1 0 1 1 1] \quad w_2 = [0 0 1 1 1 1 0 0] \quad w_3 = [0 0 1 0 1 1 0 0] \quad w_4 = [0 0 1 1 1 1 0 1]$$

Use Reed Decoding algorithm to decode the above words, assuming that in each word, there is at most one digit-error.

\section*{Hadamard Matrices}

A Hadamard matrix of order $n$ is an $n \times n$ matrix $H_n$ with elements 1 and -1 such that $H_n H_n^t = n I_n$ (i.e., $H_n$ is orthogonal).

For example, $H_1 = [1]$, $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. An $n \times n$ Hadamard matrix with $n > 2$ exists only if 4 divides $n$. For $n > 4$, in order to construct a Hadamard matrix of order $n$ using the Kronecker product, $n$ must be divisible by 8.

\[ H_4 = H_2 \otimes H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \]

Since $H_n H_n^t = n I_n$, any two different rows of $H_n$ must be orthogonal, and the matrix obtained from the permutation of rows or columns of $H_n$ is also a Hadamard matrix, but the symmetry may be lost. Clearly $-H_n$ is also a Hadamard matrix.

\section*{Decoding Algorithm for First order Reed-Muller Codes, Using Hadamard Matrices}

The $\mathcal{RM}(1,m)$ code may be decoded using a Hadamard matrix $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and the Kronecker product.

This algorithm requires that for $j = 1, 2, \ldots, m$, the matrices

$$H_m^j = I_{2^{m-j}} \otimes H \otimes I_{2^{j-1}}$$

be constructed first. For example, for $\mathcal{RM}(1,m)$ code, we need to construct:

$$H_2^1 = I_2 \otimes H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad H_2^2 = H \otimes I_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$
The recursive nature of the construction of $\mathcal{RM}(1,m)$ codes suggests that there is a recursive approach to decoding as well.

**Fast Decoding Algorithm for RM(1,m).** The generator matrix $G(1,m)$ of the $\mathcal{RM}(1,m)$ code, changes the message $a = [a_0, a_1, a_2, \ldots, a_m]$ of zeros and ones into a codeword

$$v = [v_0, v_1, v_2, \ldots, v_{2^m-1}].$$

Suppose the word $v$ was sent and the word $w = (w_0, w_1, w_2, \ldots, w_{2^m-1})$ is received.

To decode $w$, we proceed as follows:

**Step 1.** First, we replace all the zeros by -1 in $w$ to obtain $w_0$ as follows:

$$w_0 = 2w - e, \text{ where } e = [1, 1, 1, \ldots, 1, 1].$$

**Step 2.** For $j = 2, 3, \ldots, m$, compute:

$$w_1 = w_0 H_1^m, \quad w_2 = w_1 H_2^m, \quad w_3 = w_2 H_3^m, \quad \ldots, \quad w_{m-1} = w_{m-2} H_{m-1}^m, \quad w_m = w_{m-1} H_m^m.$$

**Step 3.** Find the position $i_0$ of $w_m = (w_{m,0}, w_{m,1}, \ldots, w_{m,2^m})$, where $w_{m,i_0}$ is the largest component, in absolute value.

Let $v(i_0) \in \mathbb{F}_2^m$ be the binary representation of $i_0$

$$0 \rightarrow 000 \ldots 00, \quad 1 \rightarrow 100 \ldots 00, \quad 2 \rightarrow 010 \ldots 00, \quad 3 \rightarrow 110 \ldots 00, \ldots, 2^m-1 \rightarrow 111 \ldots 11.$$
If \( w_{m,i_0} \) is positive, then, the presumed message is \( [1, v(i_0)] \), and if it is negative, then the presumed message is \( [0, v(i_0)] \).

**Note.** In the above statement, the presumed message does not mean that it is the correct message. To make sure that the message obtained from this method is the sent message, we should encode the message with the generator matrix \( G(1, m) \) and compared it with the received word \( w \); only if the encoded word is equal to \( w \), then we may assert that the procedure was successful.

**Examples.** Consider the Reed-Muller code \( RM(1, 3) \) generated by the matrix \( G(1, 3) \).

a. Suppose the word \( w = 10101011 \) is received, then:

**Step 1.** \( w_0 = 2w - e = (1, -1, 1, -1, 1, -1, 1, 1) \).

**Step 2.** Compute:

\[
\begin{align*}
w_1 &= w_0H_3^1 = [0, 2, 0, 2, 0, 2, 2, 0] \\
w_2 &= w_1H_3^2 = [0, 4, 0, 0, 2, 2, -2, 2] \\
w_3 &= w_2H_3^3 = [2, 6, -2, 2, -2, 2, -2]
\end{align*}
\]

**Step 3.** The second component of \( w_3 \) is 6 occurring in position 1. Since \( v(1) = 100 \) and \( 6 > 0 \), then the presumed message is \( a = 1100 \).

b. Suppose \( w = 10001111 \) is the received word, then

**Step 1.** \( w_0 = 2w - e = [1, -1, -1, -1, 1, 1, 1, 1] \).

**Step 2.** Compute:

\[
\begin{align*}
w_1 &= w_0H_3^1 = [0, 2, -2, 2, 0, 0, 2, 0] \\
w_2 &= w_1H_3^2 = [-2, 2, 2, 2, 4, 0, 0, 0] \\
w_3 &= w_2H_3^3 = [2, 2, 2, 2, -6, 2, 2, 2]
\end{align*}
\]

**Step 3.** The fifth component of \( w_3 \) is -6 occurring in position 4.

Since \( v(4) = 001 \) and \( -6 < 0 \), then the presumed message is \( a = 0001 \).

\( \heartsuit \) Matlab.

% Choose \( m \), then define the Hadamard matrix \( H_2 \). >> \( m = 3 \); \( H = \text{hadamard}(2) \);
% Use \( H_2 \) to define \( H_3^1, H_3^2, \ldots, H_3^m \).
>> \( H31 = \text{kron} (\text{eye}(4), H) \); \( H32 = \text{kron} (\text{kron} (\text{eye}(2), H), \text{eye}(2)) \);
>> \( H33 = \text{kron}(H, \text{eye}(4)) \)

**Example a.** Suppose the received word is:

>> \( wa = [10101011] \)

\( wa = \\
\begin{array}{cccc}
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 1
\end{array}
\)

**STEP 1.** Replace all the zero components of \( wa \) by -1.

>> \( xa = wa - \text{ones}(1,8) \); \( wxa = wa + xa \)
wxa = 
    1 -1 1 -1 1 -1 1 1

STEP 2. Now we evaluate:

\[
\begin{align*}
    \text{>> } wxa_1 &= wxa \ast H31, \quad wxa_2 = wxa_1 \ast H32, \quad wxa_3 = wxa_2 \ast H33 \\
    wxa_1 &= \begin{bmatrix} 0 & 2 & 0 & 2 & 2 & 0 \end{bmatrix} \\
    wxa_2 &= \begin{bmatrix} 0 & 4 & 0 & 0 & 2 & 2 & -2 & 2 \end{bmatrix} \\
    wxa_3 &= \begin{bmatrix} 2 & 6 & -2 & 2 & -2 & 2 & 2 & -2 \end{bmatrix}
\end{align*}
\]

STEP 3. The largest component of \( wxa_3 \) (in absolute value) is 6 occurring in position 1 (the first position is position zero). The binary position of 6 is 1 0 0.

\[
\begin{align*}
    va_1 &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\
    va &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}
\end{align*}
\]
Since 6 is positive, the presumed message is

\[
\begin{align*}
    va &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \\
    va &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}
\end{align*}
\]

The first position is position zero, so the binary position of 2 is 000.

Example b. Suppose the received word is:

\[
\begin{align*}
    wb &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \\
    wb &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{align*}
\]

STEP 1. Replace all the zero components of \( wa \) by -1.

\[
\begin{align*}
    \text{>> } xb &= wb - \text{ones}(1, 8); wxb &= wb + xb \\
    wxb &= \begin{bmatrix} 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{align*}
\]

STEP 2. Now we evaluate:

\[
\begin{align*}
    \text{>> } wxb_1 &= wxb \ast H31, \quad wxb_2 = wxb_1 \ast H32, \quad wxb_3 = wxb_2 \ast H33 \\
    wxb_1 &= \begin{bmatrix} 0 & 2 & -2 & 0 & 2 & 0 & 2 & 0 \end{bmatrix} \\
    wxb_2 &= \begin{bmatrix} -2 & 2 & 2 & 4 & 0 & 0 & 0 \end{bmatrix} \\
    wxb_3 &= \begin{bmatrix} 2 & 2 & 2 & -6 & 2 & 2 & 2 \end{bmatrix}
\end{align*}
\]

STEP 3. The largest component of \( wxb_3 \) (in absolute value) is 6, occurring in position 4. The binary position of 6 is 1 0 0.

\[
\begin{align*}
    vb_4 &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \\
    vb_4 &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}
\end{align*}
\]
Since -6 is negative, the presumed message is:
\[ \mathbf{vb} = [0 \ 0 \ 0 \ 1] \]
\[ \mathbf{vb} = 0 \ 0 \ 0 \ 1 \]
The first position is position zero, so the binary position of 2 is 0 0 0.

\% Suppose the vector \( \hat{w} = 1000 \ 1111 \) is received.
Then \( w_0 = 2w - e = (1, -1, -1, -1, 1, 1, 1, 1) \).

\[ \begin{align*}
\text{>> } \mathbf{w} &= [1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1] ; \\
\mathbf{e} &= \text{ones}(1, 2^m) ; \\
\mathbf{w}_0 &= 2 \mathbf{w} - \mathbf{e} \\
\mathbf{w}_0 &= \begin{bmatrix} 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{align*} \]
\% Define \( \mathbf{w}_1 = \mathbf{w}_0 \mathbf{H}_3^1 \), \( \mathbf{w}_2 = \mathbf{w}_1 \mathbf{H}_3^2 \), and \( \mathbf{w}_3 = \mathbf{w}_2 \mathbf{H}_3^3 \)

\[ \begin{align*}
\text{>> } \mathbf{w}_1 &= \mathbf{w}_0 \mathbf{H}_3^1 , \\
\mathbf{w}_2 &= \mathbf{w}_1 \mathbf{H}_3^2 , \\
\mathbf{w}_3 &= \mathbf{w}_2 \mathbf{H}_3^3 ; \\
\mathbf{w}_1 &= \begin{bmatrix} 0 & 2 & -2 & 0 & 2 & 0 & 2 & 0 \end{bmatrix} \\
\mathbf{w}_2 &= \begin{bmatrix} -2 & 2 & 2 & 2 & 4 & 0 & 0 & 0 \end{bmatrix} \\
\mathbf{w}_3 &= \begin{bmatrix} 2 & 2 & 2 & -6 & 2 & 2 & 2 \end{bmatrix}
\end{align*} \]
\% 4 \rightarrow \mathbf{v}(4) = 0 \ 0 \ 1 \ and \ -6 < 0, \ so \ a = 0 \ 0 \ 0 \ 1 .

\[ \begin{align*}
\text{>> } \mathbf{a} &= 0 \ 0 \ 0 \ 1 \\
\mathbf{a} &= 0 \ 0 \ 0 \ 1
\end{align*} \]

**Exercises.** Consider the Reed-Muller code \( \mathcal{R}M(1, 3) \), then decode the following received words:

- (a) \( \mathbf{w} = 0110 \ 0111 \)
- (b) \( \mathbf{w} = 0001 \ 0100 \)
- (c) \( \mathbf{w} = 1100 \ 1110 \).

\diamond **Mariner 9 and Reed-Muller \( \mathcal{R}M(1, 5) \) Code.** Mariner 9 was the first spaceship to be put into orbit around Mars. As it was mentioned previously, the main reason for selecting the Hadamard [Reed-Muller(32,64,16)] code \( \mathcal{R}M(1, 5) \), for transmitting black and white pictures of the planet was its fast decoding algorithm.

Since \( \mathcal{R}M(1, 5) \) is a 6 dimensional code, there is a basis with 6 elements (any linear combination of which gives a code word). The data type 6-tuple is used to provide the coefficients for the linear combination of the basis vectors. Thus associating a unique codeword to each data type. This simple computation can be hard wired and requires no memory.