12-1990

Reed - muller codes

Michael D. Nestor

To the Graduate Council:

I am submitting herewith a thesis written by Michael D. Nestor entitled "Reed - muller codes." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Robert M. McConnel, Major Professor

We have read this thesis and recommend its acceptance:

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Michael D. Nestor entitled "Reed - Muller Codes". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Robert M. McConnel

Robert M. M^{c}Connel, Major Professor

We have read this thesis
and recommend its acceptance:

Yueh-er Kuo

Accepted for the Council:

CW Minkel

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature ______________________

Date 8/7/90

REED - MULLER CODES

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Michael D. Nestor

December 1990

ACKNOWLEDGEMENTS

The author wishes to express sincere gratitude to Dr. Robert M. M^{c}Connel for his direction, advice, and encouragement in this thesis, and to his wife Debbie, his parents, and to Rudy B. Rabbit.

ABSTRACT

The Reed - Muller codes are a class of multiple error correcting linear codes with a geometry.

This paper investigates some properties of this code, describes two decoding schemes, and looks at the efficiency of these codes according to three bounds on linear codes.

TABLE OF CONTENTS

I. BACKGROUND

Coding theory is applied to many situations which have a common feature that information coming from a source is transmitted over a noisy channel to a receiver. Examples are telephone conversations, information from a computer keyboard transmitted to the computer, or a weather satellite transmitting radar pictures to a weather station. Suppose a weather satellite needs to send a picture of the cloud cover over the United States so that local weathermen around the nation can show this on the six o'clock news. In order to send this picture a grid is placed over the picture and for each square on the grid the degree of blackness is measured, say on a scale from 1 to 127. These numbers are expressed in the binary system, that is, each square produces a string of seven 0's and 1's. The seven-tuples of 0's and 1's are transmitted to a receiving station on earth. On arrival the signal may have become garbled (that is, errors may have occurred in some of the positions or bits of a seven-tuple, say 1010101 was sent but 0110010 was received) and therefore the picture is distorted. In order to prevent this, redundancy is built into the signal, that is, the transmitted sequence consists of more than the necessary information. These redundant bits are used to help recognize the sequence that was sent. A central goal of coding theory is to correctly recover the message that was originally sent.

Definition 1: Linear Binary Code

An (n,k) linear binary code (or a linear code over the finite field F_2) is a k-dimensional subspace V of the vector space $F_2^n = F_2 \oplus F_2 \oplus \ldots . \oplus F_2$ (n copies of F_2) over F_2. Then V is a subspace of $\{(x_1,x_2,\ldots,x_n): x_i \epsilon\ F_2\}$. The ratio k/n is called the information rate.

This paper deals exclusively with linear binary codes. A general coding procedure for a linear code is illustrated below:

$\underbrace{(u_1,u_2,\ldots,u_k)}_{\underline{u}}$----->**ENCODER**----->$\underbrace{(x_1,x_2,\ldots,x_n)}_{\underline{x}}$---->**CHANNEL**

ERRORS ADDED---->$\underbrace{(e_1+x_1,e_2+x_2,\ldots,e_n+x_n)}_{\underline{e}+\underline{x}=\underline{y}}$----->**DECODER** (Most

likely $\underline{e}$ determined)----->$\underbrace{(v_1,v_2,\ldots v_k)}_{\underline{v}}$,

where $\underline{u}$ is the information vector, $\underline{x}$ is the encoded vector, $\underline{e}$ is the added error, $\underline{y}$ is the received vector, and $\underline{v}$ is the estimate of the information vector.

One can think of an (n,k) linear code over F_2 encoded in this manner as a set of n-tuples from F_2, where each n-tuple is comprised of two parts: the message part, consisting of k digits, and the redundancy part, consisting of the remaining n-k digits. For example, the set {0000000,0010111, 0101011,1001101,1100110,1011010,01111000,1110001} is the (7,3) binary linear code with rate 3/7, the first three digits of each code word are the information digits and the last four digits are the redundancy digits.

Definition 2 : Hamming Distance and Hamming weight

The Hamming distance between two vectors of a vector space is the number of components in which they differ. The Hamming distance between two vectors (or code words), say $\underline{x}$ and $\underline{y}$ is denoted $d(\underline{x},\underline{y})$, so $d(\underline{x},\underline{y}) = |i : 0 \leq i \leq n : x_i \neq y_i|$. From hereafter when we refer to distance we mean Hamming distance.

The Hamming weight of a vector (or code word) is the number of nonzero components of the vector (or code word). The Hamming weight of a vector, say $\underline{x}$ is denoted $w(\underline{x})$. We see that $w(\underline{x}) = d(\underline{x},\underline{0})$. From hereafter when we refer to weight we mean Hamming weight.

Definition 3: Minimum Distance and Minimum Weight

The minimum distance of a code C is $\min\{d(\underline{x},\underline{y}) : \underline{x},\underline{y} \in C, \underline{x} \neq \underline{y}\}$. The minimum distance of a code C is denoted by $d_{min}(C)$.

The minimum weight of C is $\min\{w(\underline{x}) : \underline{x} \in C, \underline{x} \neq \underline{0}\}$. Note that if $\underline{x}$ and $\underline{y}$ are both code words of a linear code, then $\underline{x}-\underline{y}$ must be a code word, since the set of all code words is a vector space. Therefore, the distance between any two code words equals the weight of some third code word, and the minimum distance for a linear code equals the minimum weight of the code.

Properties of Distance and Weight of Linear Codes

For any vectors $\underline{x},\underline{y}$ and $\underline{z}$ of a linear code, $d(\underline{x},\underline{y}) \leq d(\underline{x},\underline{z}) + d(\underline{z},\underline{y})$ and $d(\underline{x},\underline{y}) = w(\underline{x}-\underline{y})$.

Proof:

Recall that $d(\underline{x},\underline{y})$ is the number of positions where $\underline{x}$ and $\underline{y}$ disagree. If $\underline{x}$ and $\underline{y}$ disagree in the i^{th} position and $\underline{x}$ and $\underline{z}$ agree then $\underline{z}$ and $\underline{y}$ disagree. Hence, it follows that $d(\underline{x},\underline{y}) \leq d(\underline{x},\underline{z}) + d(\underline{z},\underline{y})$. By definition of distance and weight it follows that $d(\underline{x},\underline{y}) = w(\underline{x}-\underline{y})$.

Correcting Capability of a Linear Code

If the weight of every nonzero code word in a linear code is at least 2t + 1, then the code can correct any t or fewer errors induced on $\underline{x}$ by $\underline{e}$. Furthermore, the code can detect any 2t errors.

Proof:

We will use nearest neighbor decoding, that is, for any received vector $\underline{y}$, assume that the corresponding code word sent is a code word $\underline{x}'$ such that the distance $d(\underline{x}',\underline{y})$ is a minimum. Now suppose the transmitted code word $\underline{x}$ is received as the vector $\underline{y}$ and at most t errors were made in the transmission. Then by definition of distance, $d(\underline{x},\underline{y}) \leq t$. If $\underline{w}$ is any other code word other than $\underline{x}$, then $\underline{w}-\underline{x}$ is a nonzero code word. Therefore, $2t+1 \leq w(\underline{w}-\underline{x}) = d(\underline{w},\underline{x}) \leq d(\underline{w},\underline{y}) + d(\underline{y},\underline{x}) \leq d(\underline{w},\underline{y}) + t \to t + 1 \leq d(\underline{w},\underline{y})$. So the code word closest to the received vector $\underline{y}$ is $\underline{x}$; therefore, $\underline{y}$ is correctly decoded as $\underline{x}$, since $\underline{x} = \underline{x}'$. In order to show that the code can detect 2t errors, suppose a transmitted code word $\underline{x}$ is received as the vector $\underline{y}$ and at least one error, but no more than 2t errors were made in the transmission. Since only code words are transmitted, an error will be detected whenever a received word is not a code word. But $\underline{y}$ cannot be a code word because $d(\underline{x},\underline{y}) \leq 2t$ and the minimum distance of the code is $2t + 1$.

An (n,k) binary linear code can be easily specified.

Definition 4: Generator Matrix

Let C be an (n,k) linear code over F_2. A matrix **G** whose row space equals C is called a generator matrix for C. Conversely, if **G** is a matrix with entries from F_2, its row space is called the code generated by **G**.

By specifying a generating matrix of a code it is possible to give a description of the code. For example, consider the code that has as a generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

By looking at **G** we see that the dimension of this code is k =4 and length n = 7, so we conclude that this is a (7,4) binary linear code with rate 4/7.

The generator matrix **G** can be used as a tool for encoding. An (n,k) binary linear code has 2^k code words, and so can be used to communicate any one of 2^k distinct messages.

Assume that these messages are k-tuples and the rows of G are linearly independent. Then a simple encoding rule which maps the messages $\underline{u} = (u_1, u_2, \ldots, u_k)$ to $\underline{x} = (x_1, x_2, \ldots, x_n)$ is $\underline{u} \dashrightarrow \underline{u}G$.

It is well known that any matrix is row-equivalent to a row reduced echelon matrix, so every linear code has a unique row reduced echelon generator matrix. If the generator matrix G is in row reduced echelon form then the standard generating matrix can easily be found and has the form:

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & \ldots & 0 & a_{1,1} & \cdots & a_{1,n-k} \\ 0 & 1 & \ldots & 0 & a_{2,1} & \cdots & a_{2,n-k} \\ . & 0 & 1 & ..0 & . & . \; . \; . & . \\ 0 & . & . \; . & 1 & a_{k,1} & . \; . & a_{k,n-k} \end{array}\right]$$

We see that we can find a function G that carries F_2^k to a subspace of F^n in such a way that for any k-tuple $\underline{u}$ in F_2^k, the image vector $\underline{u}G$ will agree with $\underline{u}$ in the first k components and build some redundancy in the last n-k components. An (n,k) linear code in which k information digits occur at the beginning of each code word is called a systematic code (or a code in systematic form).

Example 1:

Consider the (6,3) binary linear code. The messages are {000,001,010,100,101,011,110,111} and a standard generating matrix is

$$\mathbf{G}= \begin{bmatrix} 1&0&0&1&1&0\\ 0&1&0&1&0&1\\ 0&0&1&1&1&1 \end{bmatrix}$$

The message 001 is encoded to the code word 001111, since

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

The (6,3) linear code generated by **G** is

<u>Message</u>	<u>Encoder **G**</u>	<u>Code word</u>
0 0 0	$\longrightarrow$	0 0 0 0 0 0
0 0 1	$\longrightarrow$	0 0 1 1 1 1
0 1 0	$\longrightarrow$	0 1 0 1 0 1
1 0 0	$\longrightarrow$	1 0 0 1 1 0
1 1 0	$\longrightarrow$	1 1 0 0 1 1
1 0 1	$\longrightarrow$	1 0 1 0 0 1
0 1 1	$\longrightarrow$	0 1 1 0 1 0
1 1 1	$\longrightarrow$	1 1 1 1 0 0

Since the minimum weight of any nonzero code word in this (6,3) linear code is three, this code is capable of correcting any single error and detecting any double error.

Suppose an (n,k) binary linear code is encoded by the map $\underline{u} \longrightarrow \underline{u}\mathbf{G}$, where **G** is the k x n standard generating matrix. We rewrite **G** as $[\mathbf{I}_k|\mathbf{P}]$, where $\mathbf{I}_k$ is the k x k identity matrix and **P** is the k x (n-k) matrix obtained from **G** by deleting the first k columns of **G**. Now consider the (n-k) x n matrix $\mathbf{H} = [\mathbf{P}^t|\mathbf{I}_{n-k}]$. Note that $\mathbf{G}\mathbf{H}^t = [\mathbf{I}_k|\mathbf{P}][\mathbf{P}^t|\mathbf{I}_{n-k}]^t = \mathbf{P} + \mathbf{P} = \underline{0}$. This implies every code word $\underline{u}\mathbf{G}$ has inner product 0 with every row of **H**, so for a linear code C, $\underline{x} \in C \leftrightarrow \underline{x}\mathbf{H}^t = \underline{0}$. **H** is called the parity check matrix of the code C.

Theorem 1:

If C is an (n,k) linear code over F_2 with parity check matrix $\mathbf{H}$, then $d_{min}(C)$ is the smallest number of columns of $\mathbf{H}$ that are linearly independent (or sum to $\underline{0}$).

Proof:

A vector $\underline{x} = (x_1, x_2, \ldots x_n)$is a code word if and only if $\underline{x}\mathbf{H}^t = \underline{0}$. The product $\underline{x}\mathbf{H}^t$ is a linear combination of the columns of $\mathbf{H}$, and thus a linear dependence relation among the columns of $\mathbf{H}$. The number of columns of $\mathbf{H}$ that appear with nonzero coefficients is the number of nonzero components of $\underline{x}$, which is the weight of $\underline{x}$. Conversely, the coefficients of any dependence relation among the columns of $\mathbf{H}$ are components of a vector that must be in the null space of $\mathbf{H}$.

If the channel has caused some errors during transmission, then the received vector $\underline{y} = \underline{x} + \underline{e}$ and $\underline{y}H^t = s$ is called the syndrome. Since $\underline{y}H^t = (\underline{x} + \underline{e})H^t = \underline{e}H^t$, the syndrome depends only on the error vector. We can use this fact for the purpose of decoding. For a fixed $s \in F_2^{n-k}$, the set of solutions to $\underline{e}H^t$ forms a coset of the linear code C; $C + \underline{e} = \{\underline{x} + \underline{e} : \underline{x} \in C\}$. There are 2^{n-k} syndromes, therefore there are 2^{n-k} cosets of C. For example, consider the (6,3) binary linear code. The syndromes are {000, 001, 010, 100, 011, 101, 110, 111}. We arrange the vectors $\underline{e}$ according to their syndromes.

Syndrome	Coset Leader							
000	000000	001111	010101	100110	011010	101001	110011	111100
001	000001	001110	010100	100111	011011	101000	110010	111101
010	000010	001101	010111	100100	011000	101011	110101	111110
011	110000	111111	100101	010110	101010	011001	000011	001100
100	000100	001011	010001	100010	011110	101101	110111	111000
101	010000	011111	000101	110110	001010	111001	100011	101100
110	100000	101111	110101	000110	111010	001001	010011	011100
111	001000	000111	011101	101110	010010	100001	111011	110100

The above rows headed by the syndrome are the cosets of C. For example, the first row headed by 000 is C itself and the second row headed by 001 are all those vectors $\underline{e}$ such that $\underline{e}H^t = \underline{y}H^t = 001$. The coset leaders are a minimum weight vectors of each coset of C.

Syndrome Decoding

The following algorithm describes syndrome decoding.

1.) For any received vector $\underline{y}$, compute the syndrome $s = \underline{y}H^t$.

2.) Find a minimum weight vector (the coset leader) $\underline{e}$ such that $\underline{e}H^t = \underline{y}H^t$, $\underline{e} = \underline{y} + C$.

3.) Output $\underline{x} = \underline{y} + \underline{e}$.

Theorem 2

Syndrome decoding is nearest neighbor decoding, that is, a received vector $\underline{y}$ is decoded as the code word $\underline{x}$ such that $d(\underline{x},\underline{y})$ is a minimum.

Proof:

Let C be a binary linear code and $\underline{y}$ a received vector. Suppose $\underline{e}$ is the coset leader for the coset $\underline{y}$ + C, then $\underline{y}$ + C = $\underline{e}$ + C. $\underline{y} = \underline{x} + \underline{e}$ for some $\underline{x} \in C$, so $\underline{y}$ is decoded as $\underline{x}$. Let $\underline{w}$ be any other code word in C, then $\underline{y} + \underline{w} \in \underline{y} + C = \underline{e} + C$, since $\underline{e}$ is a minimum weight vector of the members of the coset $w(\underline{y} + \underline{w}) \geq w(\underline{e})$ therefore, $d(\underline{y},\underline{w}) \geq w(\underline{e}) = w(\underline{x} + \underline{y}) = d(\underline{x},\underline{y})$.

<u>Example 2:</u>

Consider the (7,4) binary linear code with

$$\mathbf{G} = \left[\begin{array}{cccc|ccc} 1&0&0&0&0&1&1\\ 0&1&0&0&1&0&1\\ 0&0&1&0&1&1&0\\ 0&0&0&1&1&1&1 \end{array}\right] \Rightarrow \quad \mathbf{H} = \left[\begin{array}{cccc|ccc} 0&1&1&1&1&0&0\\ 1&0&1&1&0&1&0\\ 1&1&0&1&0&0&1 \end{array}\right] \Rightarrow$$

$$\mathbf{H}^t = \begin{bmatrix} 0&1&1\\ 1&0&1\\ 1&1&0\\ 1&1&1\\ 1&0&0\\ 0&1&0\\ 0&0&1 \end{bmatrix}$$

Observing **H**, we see that d_{min} of the (7,4) code is three; therefore, this code is capable of correcting one channel error. If one error occurs during transmission, say in the i^{th} component of the sent vector $\underline{x}$, then $\underline{x}$ is received as $\underline{y} = \underline{x} + \underline{e}_i$, where $\underline{e}_i$ has a one in the i^{th} component. Then $\underline{e}_i\mathbf{H}^t$ is the i^{th} row of $\mathbf{H}^t$. Therefore, as the syndrome for the received vector $\underline{y}$ is the i^{th} row of $\mathbf{H}^t$, then the decoder knows an error has occurred in the i^{th} position and the estimate of $\underline{x}$ is $\underline{y} + \underline{e}_i$.

For example, suppose the received vector is $\underline{y}$ = 1 0 1 0 0 1 1, then

$$\underline{v}\mathbf{H}^t = (1\ 0\ 1\ 0\ 0\ 1\ 1) \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (1\ 1\ 0)$$

(1 1 0) is the third row of $\mathbf{H}^t$, so we assume an error was made in the third position of $\underline{v}$. Therefore, the code word is estimated to be 1 0 0 0 0 1 1.

Definition 5: Binary Hamming code

Let $\mathbf{H}$ be a m x (2^m-1) matrix such that the columns of $\mathbf{H}$ are the 2^m-1 nonzero vectors of length m over F_2 in some order. Notice that the rows of $\mathbf{H}$ are linearly independent. The (n = 2^m-1, k = 2^m-1-m) linear code over F_2 whose parity check matrix is $\mathbf{H}$ is called a binary Hamming code of length 2^m-1. The (7,4) binary linear code of example 2 is a binary Hamming code.

Definition 6: The Dual code of a Code

If C is an (n,k) linear code over F_2, a parity check for C is an equation of the form $a_1x_1 + a_2x_2 + \ldots + a_nx_n = 0$ which is satisfied for all $\underline{x} = (x_1,x_2,\ldots x_n) \in C$. The set of all vectors $\underline{a} = (a_1,a_2,\ldots a_n)$ for which this equation is satisfied for all $\underline{x} \in C$ is itself a subspace of the vector space of F_2. It is denoted by $C^\perp$ and is called the dual code of C. $C^\perp$ has dimension n - dim C, that is , $C^\perp$ is an (n,n-k) linear code over F_2. A parity check matrix for C is a generator matrix for

$C^{\perp}$. If $C = C^{\perp}$ then C is called self-dual.

Weight Enumerators of Linear Codes

The minimum distance of a linear code can tell us how many errors a received vector may contain and still be decoded correctly. Often more information about the weight of the code words is desired. For this purpose we define the weight enumerator of a linear code.

Definition 7: Weight Enumerator of a Linear Code

Let C be a linear code of length n and let A_i be the number of code words of weight i. Then $A(z) = \sum_{i=0}^{n} A_i z^i$ is called the weight enumerator of C.

For example, consider the (6,3) linear code in example 2. The code words are {000000, 001111, 010101, 100110, 110011, 101001, 011010, 111100}.

We see that the code has one word of weight zero, four words of weight three, and three words of weight four. Therefore the weight enumerator for this code is $A(z) = 1 + 4z^3 + 3z^4$.

The MacWilliams Identities

The MacWilliams identities give a relation between the weight enumerator of a linear code and the weight enumerator of its dual code.

Let A(z) be the weight enumerator of an (n,k) linear code

C over F_q and let B(z) be the weight enumerator of $C^{\perp}$. Then A(z) and B(z) are related by the formula

$$q^k B(z) = [1+(q-1)z]^n A((1-z)/[1+(q-1)z])$$

For codes over F_2 the formula reduces to

$$2^k B(z) = [1+z]^n A[(1-z)/(1+z)].$$

For a proof of the MacWilliams identities see reference 4.

Example 3:

Consider the (7,3) Hamming code. From example 2, we see that each of the nonzero code words have weight four. So the weight enumerator is $A(z) = 1 + 7z^4$. The dual code is the (7,4) linear code. Using the MacWilliams identities, the weight enumerator of the (7,4) code is

$$\begin{aligned} B(z) &= 2^{-3}[1+z]^7 A[(1-z)/(1+z)] \\ &= 1/8 \ [1+z]^7 \ [1+7(1-z)^4(1+z)^{-4}] \\ &= 1/8 \ \{[1+z]^7 + 7(1-z)^4(1+z)^3\} \\ &= 1 + 7z^3 + 7z^4 + z^8. \end{aligned}$$

From B(z) we see that the (7,4) linear code has one word of weight zero, seven words of weight three, seven words of weight four, and one word of weight eight.

Constructing Linear Codes from Other Linear Codes

Many linear codes can be constructed by modifying previously constructed linear codes. We will give one of two methods to modify a linear code to obtain another linear code.

The first method called extending a code is defined as follows: if C is a linear code of length n over F_2, we define the extended code $C_e = \{(c_1,c_2,\ldots,c_n,c_{n+1}) \mid (c_1,c_2,\ldots,c_n) \in C, \Sigma\, c_i = 0\} \cup (11 \ldots 1)$. The second method, called puncturing a code, is the inverse process of extending a code.

<u>Example 4:</u>

Consider the (7,4) Hamming code with parity check matrix

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 1 & : & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 0 & 1 \end{bmatrix}$$

If we add an overall parity check or extend the code, the parity check matrix of the extended code is

$$\mathbf{H'} = \begin{bmatrix} & & & & & & & 0 \\ & & & & \mathbf{H} & & & 0 \\ & & & & & & & 0 \\ 1 & 1 & 1 & 1 & : & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 1 & 1 & : & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & : & 1 & 1 & 1 & 1 \end{bmatrix}$$

If we put $\mathbf{H'}$ in standard form, we get

$$\mathbf{H'} = \begin{bmatrix} 0 & 1 & 1 & 1 & : & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & : & 0 & 0 & 0 & 1 \end{bmatrix}$$

and therefore the standard generating matrix is

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & : & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & : & 1 & 1 & 1 & 0 \end{bmatrix}$$

This code is an (8,4) linear code, and since it is an extension of the (7,4) Hamming code, this code is called the (8,4) extended Hamming code.

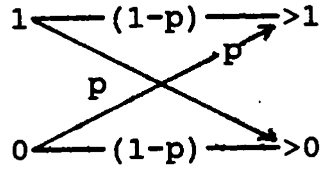
Channel Error

We have seen that a linear code is capable of correcting some errors that have occurred during transmission of the code words. But in order to predict the performance of the code, it is desirable to have some information about the source of the code and the channel through which the code is transmitted.

Our source for a linear binary code word is the binary symmetric source. The binary symmetric source emits two possible symbols " 0 " or " 1 " at a rate $R = k/n$. These symbols are called bits. The bits emitted by the source are random and a 0 is as likely to be emitted as a 1.

Our channel, the binary symmetric channel, is an object through which it is possible to transmit one bit per unit time. This channel is not completely reliable and there is a fixed probability p, called the raw bit error probability with $0 \leq p \leq 1/2$, that the output bit will not be the same as the input bit.

If p is the raw bit error probability, then 1-p is the probability that the output bit is the same as the input bit. We can represent the binary symmetric channel in the following way:



Consider the (7,4) Hamming code with parity check matrix decoding. We have seen that this code is single error correcting. In fact, syndrome decoding will fail to correctly identify the original code word if and only if two or more bit errors occur. In this code we are independently choosing seven bits for a code word. If two or more bit errors occur then the original code word is not identified. Thus, a binomial distribution describes the probability that a code word is incorrectly identified. We call this probability the block error probability, $P_E = P[\underline{v} \neq \underline{u}]$, where $\underline{u}$ is the original code word and $\underline{v}$ is its estimate. Hence the block error probability for a (7,4) Hamming code is

$$P_E = \Sigma\ C(7,k)\ p^k\ (1-p)^{7-k}$$

The block error probability P_E gives us the probability that the estimate $\underline{v}$ of the original code word $\underline{u}$ are not equal. But some of the components of $\underline{v}$ may nevertheless be correct. So we define the bit error probability $P_e^i = P(v_i \neq u_i)$. The entire bit error probability for a (n,k) linear code is given by $P_e = 1/k\ \Sigma\ P_e^i$, where $P_e^i = P(v_i \neq u_i)$ and $i = 1,2,\ldots,k$.

Recall that the rate of an (n,k) binary linear code is $R = k/n$. Figure 1 [See reference 4] shows the set of achievable (R,P_e) pairs for a binary symmetric source and channel with p

= .1.

To help understand Figure 1, consider a linear code C with rate R = 1. Suppose that C has a bit error probability, P_e = 0.3. Figure 1 tells us that there exists codes with a lower bit error probability.

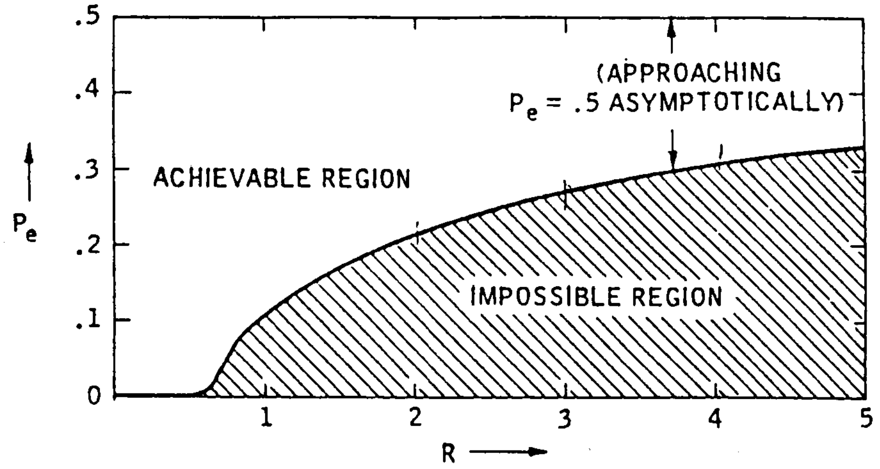


Figure 1. Achievable (R,P_e) for a B.S.C. (p = .1).

A description of the boundary between the achievable points and nonachievable points would be most helpful. In order to give the description, we need to introduce the binary entropy function:

$H_2(x) = -x\mathrm{Log}_2 x - (1-x)\mathrm{Log}_2(1-x)$, $0 < x < 1$, where $H_2(0) = H_2(1) = 0$.

The curved part of the boundary between achievable points and nonachievable points in Figure 1 is the set points (R,P_e) that satisfy:

$R = [1-H_2(0.1)]/[1-H_2(P_e)]$, where $0 \le P_e \le 1/2$

The remainder of the boundary is the segment of the R axis, from R = 0 to R = 1 - $H_2(0.1)$ = 0.531.

Figure 2 [See reference 4] is the graph of a general binary symmetric source and channel. The curved part of the boundary is the set of points (R,P_e) that satisfy:

$R = [1 - H_2(p)]/[1 - H_2(P_e)]$

The remainder of the boundary is the segment of the R axis from R = 0 to R = 1 - $H_2(p)$.

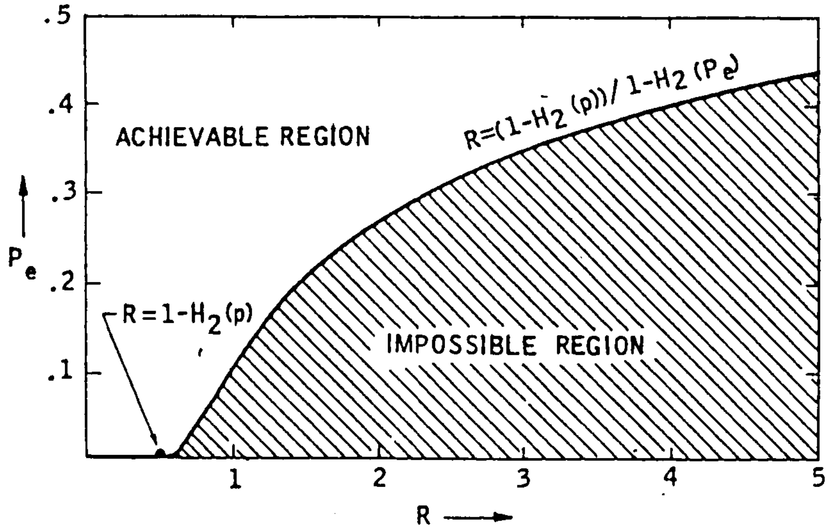


Figure 2. Achievable (R,P_e) for a general B.S.C.

A remarkable thing about this is if R < 1 - $H_2(p)$, then any positive P_e,however small, is achievable. The number 1 - $H_2(p)$ is called the capacity of the channel.

A binary linear code is called optimum for the binary symmetric channel if its probability of error is as small as for any binary linear code with the same length n and the same dimension k.

Some Bounds on Binary Linear Codes

In this section we will describe three bounds on binary linear codes. The first bound, the Plotkin bound is an upper bound on the minimum distance for the code with the maximum minimum distance. The second bound, the Varsharmov - Gilbert bound (V - G bound) is a lower bound on the minimum distance for the code with the maximum minimum distance. Finally, the third bound, the Hamming bound is another upper bound on the maximum minimum distance possible and a lower bound on the entire bit error probability for codes using a binary symmetric channel.

The Plotkin Bound

Fix n and d, and let $M_L(n,d)$ denote the maximum number of code words in a binary linear code with length n and minimum distance greater than or equal to d. The Plotkin bound states that if $n \geq 2d - 2$, then $M_L(n,d) \leq 2^{n-2d+2}d$. Furthermore, the number of parity checks, (n-k), required to achieve minimum distance is at least $2d - 2 - \log_2 d$.

In order to prove the Plotkin bound, we will use the following theorem:

Theorem 3

Let C be an (n,k) binary linear code, then $d_{min}(C) \leq n2^{k-1} / (2^k - 1)$.

Proof:

For any i $(1 \leq i \leq n)$ let $A = (\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_t)$ be the set of code words of C with a "1" in the i^{th} position. Let B = $\{\underline{b}_1,\underline{b}_2,\ldots,\underline{b}_s\}$ be the set of all code words with a "0" in the i^{th} position. We see that $A \cup B = C$ and $A \cap B = 0$. Now take a code word $\underline{b}_j \in B$ and consider the set $\underline{b}_j + A = \{\underline{b}_j+\underline{a}_1,\underline{b}_j+\underline{a}_2,\ldots,\underline{b}_j+\underline{a}_t\}$. Then $\underline{b}_j + A \subseteq B$, $|\underline{b}_j + A| = |A| \rightarrow |A| \leq |B|$. Now consider the set $\underline{a}_j + B = \{\underline{a}_j+\underline{b}_1,\underline{a}_j+\underline{b}_2,\ldots,\underline{a}_j+\underline{b}_s\}$, $a_j \in A$; then $\underline{a}_j + B \subseteq A$, $|\underline{a}_j + B| = |B| \rightarrow |B| \leq |A|$. Therefore, $|A| = |B|$. But $|C| = 2^k \rightarrow |A| = 2^{k-1} = |B|$. If we arrange the code words of C as rows of a matrix, since $|A| = 2^{k-1} = |B|$ we observe that there are 2^{k-1} 0's and 2^{k-1} 1's in each column of this matrix. The sum of all the weights of all the code words is equivalent to adding all the 1's (as real numbers) that appear in this matrix. Therefore, this sum equals $n2^{k-1}$. A minimum weight nonzero code word of a code is at most the average weight of the code words of C. Since there are $2^k - 1$ code words with nonzero weight, $d_{min}(C) \leq n2^{k-1} /(2^k-1)$.

Now we prove the Plotkin bound. Let C be an (n,k) binary linear code with $|C| = M_L(n,d)$, with $d_{min}(C) = d'$ and $d' \geq d$. Let C_n be the code obtained by selecting all code words of C that have 0 in their first digit. Because their are 2^{k-1} such code words in C, $|C_n| = 1/2\ |C|$. Now let C_{n-1} be the code obtained by selecting all code words of C_n that have a 0 in the second digit, $|C_{n-1}| = 1/2\ |C_n| = 1/4\ |C|$. We continue to obtain codes in this manner until we have obtained code C_o, which has 0's in the first n - 2d + 2 digits, $|C_o| = 1/2^{n-2d+2}$

$|C|$. We delete the first $n - 2d + 2$ digits of C_0 to obtain a code C'_0. C'_0 has length $2d - 2$, dimension k', and minimum distance equal to the minimum distance of C. Therefore, $2^{k'} = 1/2^{n-2d+2} |C| = 1/2^{n-2d+2} M_L(n,d)$. Substituting C'_0 into the result of theorem 3 we have

$d' \leq (2d-2)2^{k'-1} / (2^{k'} - 1)$,

So,

$d(2^{k'}-1) \leq (2d-2)2^{k'-1}$

$d2^{k'}-d \leq (2d-2)2^{k'-1}$

$2^{k'-1}(2d-2d+2) \leq d$

$2^{k'} \leq d$

Therefore, $1/2^{n-2d+2} M_L(n,d) \leq d$.

Since, $M_L(n,d) = 2^k$ for the binary linear code with maximum minimum distance, we have

$k \leq n - 2d + 2 + \log_2 d$ or $n - k \geq 2d - 2 - \log_2 d$.

In other words, the number of parity checks required to achieve maximum minimum distance is at least $2d - 2 - \log_2 d$.

The Varsharmov - Gilbert Bound

If $C(n-1,1) + C(n-1,2) + \ldots + C(n-1,d-2) < 2^r$, then there exists a code with length n and at most r parity check symbols , where r is the smallest integer value that satisfies the above equation and $d = d_{min}$. For proof of the V-G bound see reference 5.

The Hamming Bound

Let C be an (n,k) binary linear code with $d_{min} \geq 2t + 1$, then C must have at least

$\log_2 [1 + C(n,1) + C(n,2) + \ldots + C(n,t)]$ parity check symbols. Also if a binary code of length n is capable of correcting t errors then it contains at most $2^n / (1 + C(n,1) + \ldots + C(n,t))$ code words. For proof of the Hamming bound see reference 5.

Figure 3 [See reference 4] is a graph of the three bounds for codes with large length n and minimum distance. For large n and $d_{min}(C)$, the Plotkin bound can be written as $k/n \leq 1 - 2(d_{min}(C))/n$, the Hamming bound can be written as $k/n \leq 1 - H(t/n)$, where t is the number of errors the code is capable of correcting, and the V-G can be written as $k/n \geq 1 - H((2(d_{min}(C))-2)/n-1)$ [See reference 4]. These bounds can be interpreted as bounds on the minimum distance d possible for a (n,k) binary linear code. We should note that the greater the minimum distance of a code , the better the error correcting capability. But we sacrifice the rate of the code by increasing minimum distance. Also, if point (d/2n,k/n) of a code lies on the Hamming upper bound then it can be shown that the code is optimum [See reference 4]. If we fix the rate k/n of a code with large n, say at 0.7, the V - G lower bound tells us a (n,k) binary linear code with d/2n approximately equal to .03 exists; the Hamming upper bound tells us if the (n,k) code has d/2n equal to about .05 then the code is

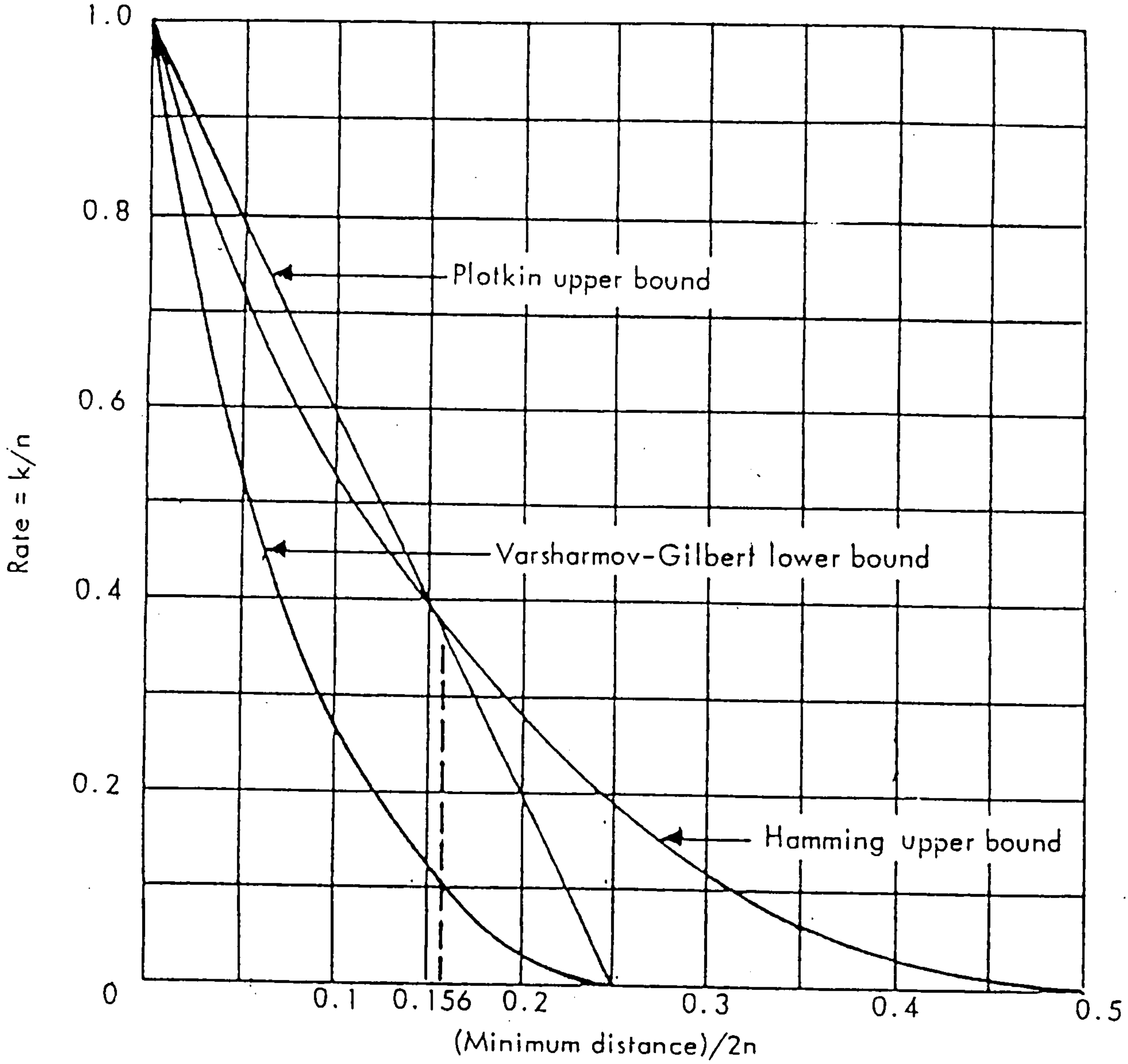


Figure 3. Some bounds on linear codes.

optimum; the Plotkin bound tells us if the code is to have maximum minimum distance the d/2n has to be about 0.1.

Geometry

Definition 8: Affine Geometry

The affine geometry of dimension m over the field F_q is the vector space F_q^m. We use the notation AG(m,q) for the geometry.

Definition 9: K-Flat

A k-dimensional affine subspace or a k-flat is a coset of a k-dimensional linear subspace. If k=m-1 we call the flat a hyperplane.

II. REED - MULLER CODES

Introduction

The Reed-Muller class of codes (R-M codes) were developed by D.E. Muller and I.S. Reed. Reed went further and developed an easily applied multiple error correcting decoding scheme. The R-M codes are binary linear codes and are connected with finite geometries.

Construction

There are several ways of constructing the R-M codes. Reed developed the code words as lists of values which are taken by a Boolean function on F_2^m. Later these codes were represented in terms of characteristic functions of subsets in the affine geometry of dimension m over F_2, that is, AG(m,2), and as coefficients of binary expansions of polynomials. We will represent the R-M codes in terms of characteristic functions of subsets in AG(m,2), and as coefficients of binary expansions of polynomials. But first we will define the basis vectors for this code, use them as rows of a generator matrix, and thus easily define the code words. We need the following notation to define the basis vectors. Let the standard basis of F_2^m be denoted by $\underline{u}_0, \underline{u}_1, \ldots \underline{u}_{m-1}$, let the binary representation of a number j $(0 \le j < 2^m)$ be $j = \sum_{i=0}^{m-1} \beta_{i,j} 2^i$, where $\beta_{i,j} \in F_2$. Then $\underline{x}_j = \sum_{i=0}^{m-1} \beta_{i,j} \underline{u}_i$ is a point in AG(m,2) and all points of AG(m,2) are obtained in this way. Let **E** be the m x n, where

$n = 2^m$, matrix with columns $\underline{x}_j$. Hence $\mathbf{E}$ is a list of the points of AG(m,2).

$$\mathbf{E} = \begin{array}{c}
\begin{array}{cccc} \underline{x}_0 & \underline{x}_1 & & \underline{x}_n \end{array} \\
\left[\begin{array}{cccc}
\beta_{0,0} & \beta_{0,1} & \cdots\cdots & \beta_{0,n-1} \\
\beta_{1,0} & \beta_{1,1} & \cdots\cdots & \cdot \\
\vdots & \vdots & \ddots & \vdots \\
\beta_{m-1,0} & \cdot & \cdot & \beta_{m-1,n-1}
\end{array}\right]
\end{array}$$

Example 4:

Let $m = 3$, then we have $0 \le j < 8$.

$j = 0 = 0\cdot2^0 + 0\cdot2^1 + 0\cdot2^2 \rightarrow \underline{x}_0 = (000)$

$j = 1 = 1\cdot2^0 + 0\cdot2^1 + 0\cdot2^2 \rightarrow \underline{x}_1 = (100)$

$j = 2 = 0\cdot2^0 + 1\cdot2^1 + 0\cdot2^2 \rightarrow \underline{x}_2 = (010)$

$j = 3 = 1\cdot2^0 + 1\cdot2^1 + 0\cdot2^2 \rightarrow \underline{x}_3 = (110)$

$j = 4 = 0\cdot2^0 + 0\cdot2^1 + 1\cdot2^2 \rightarrow \underline{x}_4 = (001)$

$j = 5 = 1\cdot2^0 + 0\cdot2^1 + 1\cdot2^2 \rightarrow \underline{x}_5 = (101)$

$j = 6 = 0\cdot2^0 + 1\cdot2^1 + 1\cdot2^2 \rightarrow \underline{x}_6 = (011)$

$j = 7 = 1\cdot2^0 + 1\cdot2^1 + 1\cdot2^2 \rightarrow \underline{x}_7 = (111)$

Then the 3x8 matrix $\mathbf{E}$ is

$$\begin{array}{c}
\begin{array}{cccccccc} \underline{x}_0 & \underline{x}_1 & \underline{x}_2 & \underline{x}_3 & \underline{x}_4 & \underline{x}_5 & \underline{x}_6 & \underline{x}_7 \end{array} \\
\left[\begin{array}{cccccccc}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}\right]
\end{array}$$

At this point we can construct the R-M codes using the rows of the matrix $\mathbf{E}$ and the 2^m-length vector $\underline{I} = (11...1)$. We label the rows of $\mathbf{E}$, starting with the first row, $\underline{v}_0$, $\underline{v}_1$,

$\underline{v}_2$, ..., and $\underline{v}_{m-1}$, respectively. The rth order R-M code is formed by using the vectors $\underline{I}$, $\underline{v}_0$, $\underline{v}_1$, $\underline{v}_1$, ..., $\underline{v}_{m-1}$ and all products of these vectors r or fewer times as a basis. If $\underline{v}_i = (a_0, a_1, \ldots, a_n)$ and $\underline{v}_j = (b_0, b_1, \ldots, b_n)$ then vector multiplication is defined by $\underline{v}_i\underline{v}_j = (a_0b_0, a_1b_1, \ldots, a_nb_n)$. The rth order R-M code of length $n = 2^m$ is denoted by R(r,m). A generator matrix **G** (not in standard form) is

$$\begin{bmatrix} 1\ 1\ \ldots\ 1 \\ \underline{v}_0 \\ \underline{v}_1 \\ \cdot \\ \underline{v}_0\underline{v}_1 \\ \cdot \\ \underline{v}_0\underline{v}_1 \cdot\cdot \underline{v}_r \end{bmatrix}$$

The code words are of R(r,m) are formed by taking all combinations of sums of the basis vectors.

Example 5:

Let m = 3. We use matrix **E** of example 4 to get

$\underline{I} = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$

$\underline{v}_0 = (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)$

$\underline{v}_1 = (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$

$\underline{v}_2 = (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)$

The basis vectors for R(1,3) are $\underline{I}$, $\underline{v}_0$, $\underline{v}_1$, and $\underline{v}_2$.

The basis vectors for R(2,3) are $\underline{I}$, $\underline{v}_0$, $\underline{v}_1$, $\underline{v}_2$, $\underline{v}_0\underline{v}_1$, $\underline{v}_0\underline{v}_2$, and $\underline{v}_1\underline{v}_2$.

The basis vectors for R(3,3) are $\underline{I}$, $\underline{v}_0$, $\underline{v}_1$,$\underline{v}_2$, $\underline{v}_0\underline{v}_1$, $\underline{v}_0\underline{v}_2$, $\underline{v}_1\underline{v}_2$, and $\underline{v}_0\underline{v}_1\underline{v}_2$.

The code words for R(1,3) are

$\underline{v}_0 = (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)$

$\underline{v}_1 = (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$

$\underline{v}_2 = (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)$

$\underline{v}_0 + \underline{v}_1 = (0\ 1\ 1\ 0\ 0\ 1\ 1\ 0)$

$\underline{v}_0 + \underline{v}_2 = (0\ 1\ 0\ 1\ 1\ 0\ 1\ 0)$

$\underline{v}_1 + \underline{v}_2 = (0\ 0\ 1\ 1\ 1\ 1\ 0\ 0)$

$\underline{v}_0 + \underline{v}_1 + \underline{v}_2 = (0\ 1\ 1\ 0\ 1\ 0\ 0\ 1)$

$\underline{I} = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$

$\underline{I} + \underline{v}_0 = (1\ 0\ 1\ 0\ 1\ 0\ 1\ 0)$

$\underline{I} + \underline{v}_1 = (1\ 1\ 0\ 0\ 1\ 1\ 0\ 0)$

$\underline{I} + \underline{v}_2 = (1\ 1\ 1\ 1\ 0\ 0\ 0\ 0)$

$\underline{I} + \underline{v}_0 + \underline{v}_1 = (1\ 0\ 0\ 1\ 1\ 0\ 0\ 1)$

$\underline{I} + \underline{v}_0 + \underline{v}_2 = (1\ 0\ 1\ 0\ 0\ 1\ 0\ 1)$

$\underline{I} + \underline{v}_1 + \underline{v}_2 = (1\ 1\ 0\ 0\ 0\ 0\ 1\ 1)$

$\underline{I} + \underline{v}_0 + \underline{v}_1 + \underline{v}_2 = (1\ 0\ 0\ 1\ 0\ 1\ 1\ 0)$

$(0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$

R(1,3) has 2^4 code words and is an (8,4) binary linear code.

The number of basis vectors for R(1,m) is 1 (for $\underline{I}$) + m. The number of basis vectors for R(2,m) is 1 + m + C(m,2), since we choose 2 out of the m vectors for products. In general the number of basis vectors for R(r,m) is 1 + C(m,1) + C(m,2) + ... + C(m,r). As will be shown in this chapter, these basis vectors are linearly independent so the dimension k of R(r,m) is k = 1 + C(m,1) + C(m,2) + C(m,3) + ... + C(m,r).

Now we represent R-M codes in terms of characteristic functions of subsets in AG(m,2) and as coefficients of binary expansions of polynomials. The first representation presented here shows the geometry of the codes. But in order to do this we will introduce the following definitions and lemmas.

Definitions

1.) $A_i = \{\underline{X}_j \in AG(m,2) \mid \beta_{i,j} = 1\}$; A_i is an (m-1)-dimensional affine subspace.

2.) If $\underline{v}_i$ is the i^{th} row of **E** ; $\underline{v}_i$ has a one in every position where $\beta_{ij} = 1$, $0 \le j < 2^m$. Therefore, $\underline{v}_i$ is the characteristic function of A_i, $\underline{I}$ is the characteristic function of AG(m,2).

3.) If $\underline{a} = (a_0, a_1, \ldots, a_n)$ and $\underline{b} = (b_0, b_1, \ldots b_n)$ are vectors in F_2^n, then $\underline{a}\,\underline{b} = (a_0b_0, a_1b_1, \ldots, a_nb_n)$.

4.) If $S \subset \{0,1,2,\ldots,m-1\}$ define

$$T(S) = \{j = \sum_{i=0}^{m-1} \beta_{i,j} 2^i \mid i \notin S \rightarrow \beta_{i,j} = 0,\ 0 \le i < m\}$$

Lemma 1:

Let $t = \sum_{i=0}^{m-1} \beta_{i,t} 2^i$ and let i(1),i(2),...i(s) be the values of i for which $\beta_{i,t} = 0$. If $\underline{v}_{i(1)}\underline{v}_{i(2)}\cdots\underline{v}_{i(s)} = (a_{t,o}, a_{t,1}, \ldots a_{t,n-1})$, then $(x+1)^t = \sum_{j=0}^{n-1} a_{t,j} x^{n-1-j}$. If there are no values of i for which $\beta_{i,t} = 0$ then the vacuous product of $\underline{v}_{i(s)}$'s is the vector $\underline{I}$.

In order to prove Lemma 1 we will use Lucas's Theorem from number theory.

Lucas's Theorem

Let p be a prime number and let $n = \sum_{i=0}^{L} n_i p^i$ and $k = \sum_{i=0}^{L} k_i p^i$ be representations of n and k in base p. Then $C(n,k) \equiv \prod_{i=0}^{L} C(n_i,k_i) \pmod p$

Proof:

We use the fact that $(1 + x)^p \equiv 1 + x^p \pmod p$. Write $n = ap + r$ and $k = bp + s$, where $0 \le r < p$ and $n \ge k$. Then $(1 + x)^{ap+r} \equiv (1 + x^p)^a (1 + x)^r \pmod p$.

Comparing coefficients of x^{bp+s} on both sides of the congruence yields

$C(ap+r,bp+s) \equiv C(a,b)\ C(r,s) \pmod p$.

The result now follows from induction, that is, write $n = \sum_{i=0}^{L} \alpha_i p^i = \sum_{i=1}^{L} \alpha_i p^i + \alpha_0$ and $k = \sum_{i=0}^{L} \beta_i p^i = \sum_{i=1}^{L} \beta_i p^i + \beta_0$, where $\alpha_i \ge \beta_i$.

Proof of Lemma 1:

In Lucas's theorem, note that when p = 2, the $C(n_i,k_i)$ are one of C(0,0) = 1, C(0,1) = 0, C(1,0) = 1, or C(1,1) = 1. Thus for p = 2, we have that $C(n_i,k_i) = 1$ if and only if $k_i = 0$ whenever $n_i = 0$.

Now $(x + 1)^t = C(t,0)x^t + C(t,1)x^{t-1} + \ldots + 1$. Rewrite each C(t,k), where k = 0,1,2,...,t as C(t,n-1-j), where $0 \le j \le n-1$. By Lucas's theorem, $C(t,n-1-j) \equiv C(\beta_{0,t},\beta_{0,n-1}-\beta_{0,j})\ C(\beta_{1,t},\beta_{1,n-1}-\beta_{1,j}) \ldots C(\beta_{m,t},\beta_{m,n-1}-\beta_{m,j}) \pmod 2$. Because the 2-adic expansion of n-1 has all coefficients one, $\beta_{i,n-1} = 1$, for all i. And since $\beta_{i,n-1-j} = \beta_{i,n-1} - \beta_{i,j}$, $C(t,n-1-j) \equiv 1 \pmod 2$ if and only if $\beta_{i,j} = 1$ for every i for which $\beta_{i,t} = 0$.

The values for i for which $\beta_{i,t} = 0$ are i(1), i(2) ,..., i(s). Also $v_{i(1)}v_{i(2)} \cdots v_{i(s)} = (a_{t,0}, a_{t,1}, \ldots, a_{t,n-1})$ is the product of the rows i(1),i(2),...,i(s) of the matrix $\mathbf{E}$. In this product we will have a one in the j^th position if and only if $\beta_{i,j} = 1$, for i = i(1),i(2),...,1(s). Since i(1), i(2),..., i(s) are all values of i such that $\beta_{i,t} = 0$, then for all i, $\beta_{i,j} = 1$ whenever $\beta_{i,t} = 0$ if and only if $\beta_{i,j} = 1$ for all i = i(1), i(2),...,i(s). Therefore, $C(t,n-1-j) = a_{t,j}$ and $(x + 1)^t = \sum_{j=0}^{n-1} a_{t,j}\, x^{n-i-j}$.

<u>Lemma 2</u>

If i(1),i(2),...,i(s) are different then

1.) $\underline{v}_{i(1)}\underline{v}_{i(2)} \cdots \underline{v}_{i(s)}$ is the characteristic function of the (m-s)-flat $A_{i(1)} \cap A_{i(2)} \cap \ldots \cap A_{i(s)}$. We call these (m-s)-flats basis flats.

2.) The weight $w(\underline{v}_{i(1)}\underline{v}_{i(2)} \cdots \underline{v}_{i(s)})$ in F_2^n is 2^{m-s}.

3.) Let e_j $(0 \leq j \leq n - 1)$ be the j^th basis vector in a standard basis for F_2^n. Then $e_j = \prod_{i=0}^{m-1} (\underline{v}_i + (1 + \beta_{i,j})\, \underline{I})$.

4.) The products $\underline{v}_{i(1)}\underline{v}_{i(2)} \cdots \underline{v}_{i(s)}$ $(0 \leq s \leq m)$ are a basis of F_2^n.

Proofs:

1.) $A_{i(k)} \cap A_{i(t)} = \{\underline{x}_j \in AG(m,2) \mid \beta_{i(k),j} = 1 \text{ and } \beta_{i(t),j} = 1\}$. $\underline{v}_{i(k)} = (a_0, a_1, \ldots a_{n-1})$ and $\underline{v}_{i(t)} = (b_0, b_1, \ldots b_{n-1})$ are the characteristic functions of $A_{i(k)}$ and $A_{i(t)}$ respectively. Therefore every component a_h of $\underline{v}_{i(k)}$ $(h = 0,1,\ldots,n-1)$is 1 whenever $\beta_{i(k),j} = 1$, likewise every component b_h of $\underline{v}_{i(t)}$

is 1 whenever $\beta_{i(t),j} = 1$. The product $\underline{v}_{i(k)}\underline{v}_{i(t)}$ has a 1 in every component $a_h b_h$ where both a_h and b_h are equal to one. Therefore, $\underline{v}_{i(k)}\underline{v}_{i(t)}$ is the characteristic function of $A_{i(k)} \cap A_{i(t)}$.

2.) The weight of $\underline{v}_{i(1)}\underline{v}_{i(2)}\cdots\underline{v}_{i(s)}$ is the cardinality of an (m-s)-flat = 2^{m-s}.

3.) Consider the matrix **E**. For each i for which $\beta_{i,j} = 0$, we replace $\underline{v}_i$ by $\underline{I} + \underline{v}_i$ in the matrix **E'**. Notice that $\prod (\underline{v}_i + (1 + \beta_{i,j})\ \underline{I})$ is the product of the rows of the matrix **E'**. Now we multiply the rows of this matrix **E'**. Consider the k^{th} position in this n-tuple. If $k = j$, then the j^{th} position in each row of **E'**is one. If $k \neq j$, then as the rows of **E** are distinct, there is an i such that $\beta_{i,k} \neq \beta_{i,j}$. If $\beta_{i,j} = 0$, then $\beta_{i,k} = 1$ and the i^{th} row of **E'**is $\underline{I} + \underline{v}_i$ and thus has a zero in the k^{th} position. If $\beta_{i,j} = 1$, then $\beta_{i,k} = 0$ and the i^{th} row of **E'** is $\underline{v}_i$ and thus has a zero in the k^{th} position. In either case there is a row of **E'** that has a zero in the k^{th} position.

4.) There are $1 + C(m,1) + C(m,2) + \ldots + C(m,s) = 2^m$ products $\underline{v}_{i(1)}\cdots\underline{v}_{i(s)}$. Since $\{\underline{e}_j : 0 \leq j \leq n\}$ is a basis of $F_2{}^n$ and each $\underline{e}_j$ is a linear combination of the products $\underline{v}_{i(1)} \cdots \underline{v}_{(s)}$, the products $\underline{v}_{i(1)} \cdots \underline{v}_{i(s)}$ are a basis of $F_2{}^n$.

Example 6 illustrates Lemma 2 part 3.

Example 6

The third basis vector in a standard basis for F_2^8 is $\underline{e}_3$ = (0 0 0 1 0 0 0 0). Observe (0 0 0 1 0 0 0 0) = $\prod (\underline{v}_i + (1+\beta_{i,j})\underline{I}$ = $\underline{v}_0\underline{v}_1(\underline{v}_2+\underline{I})$ = $\underline{v}_0\underline{v}_1\underline{v}_2 + \underline{v}_0\underline{v}_1$ = (0 0 0 0 0 0 0 1) + (0 0 0 1 0 0 0 1) = (0 0 0 1 0 0 0 0).

From 4.) this set of vectors $\underline{v}_{i(1)} \cdot \cdot \cdot \underline{v}_{(s)}$ is linearly independent and hence forms a basis for the R - M code. Now we define R-M codes in terms of characteristic functions of subsets of AG(m,2).

Definition 10: Reed - Muller Code

Let $0 \leq r < m$. The linear code of length $n = 2^m$ which has products $\underline{v}_{i(1)}\underline{v}_{i(2)}\cdots\underline{v}_{i(s)}$ with $s \leq r$ factors and $\underline{I}$ as a basis is called the rth order Reed - Muller Code (R(r,m)).

Example 7:

Let m = 3 . We find the same matrix **E** as in example 4:

$$\begin{array}{cc} & \begin{array}{cccccccc} \underline{x}_0 & \underline{x}_1 & \underline{x}_2 & \underline{x}_3 & \underline{x}_4 & \underline{x}_5 & \underline{x}_6 & \underline{x}_7 \end{array} \\ \mathbf{E} = & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \end{array}$$

Also R(1,3), R(2,3), and R(3,3) have the same basis vectors as in example 5. But we note that

$\underline{v}_0$ is the characteristic function of $A_0 = \{\underline{x}_1,\underline{x}_3,\underline{x}_5,\underline{x}_7\}$,

$\underline{v}_1$ is the characteristic function of $A_1 = \{\underline{x}_2,\underline{x}_3,\underline{x}_6,\underline{x}_7\}$,

$\underline{v}_2$ is the characteristic function of $A_2 = \{\underline{x}_4,\underline{x}_5,\underline{x}_6,\underline{x}_7\}$,

$\underline{v}_0\underline{v}_1$ is the characteristic function of $A_0 \cap A_1 = \{\underline{x}_3,\underline{x}_7\}$,

$\underline{v}_0\underline{v}_2$ is the characteristic function of $A_0 \cap A_2 = \{\underline{x}_5,\underline{x}_7\}$,

$\underline{v}_1\underline{v}_2$ is the characteristic function of $A_1 \cap A_2 = \{\underline{x}_6, \underline{x}_7\}$,
$\underline{v}_0\underline{v}_1\underline{v}_2$ is the characteristic function of $A_0 \cap A_1 \cap A_2 = \{\underline{x}_7\}$,
and $\underline{I}$ is the characteristic function of AG(m,2).

Lemma 1 shows how the basis vectors of R - M codes can be represented by coefficients of the polynomial expansion of $(x + 1)^t$. For example, $\underline{v}_0 = (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)$ tells us that t is the number for which $\beta_{0,t} = 0$, $\beta_{1,t} = 1$, and $\beta_{2,t} = 1$. So $t = 0\cdot 2^0 + 1\cdot 2^1 + 1\cdot 2^2 = 6$; therefore, $(x + 1)^6 = x^6 + x^4 + x^2 + 1$ corresponds to $\underline{v}_0$. Likewise, for $\underline{v}_1 = (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$, t is the number for which $\beta_{0,t} = 1$, $\beta_{1,t} = 0$, and $\beta_{2,t} = 1$. So $t = 1\cdot 2^0 + 0\cdot 2^1 + 1\cdot 2^2 = 5$; therefore, $(x + 1)^5 = x^5 + x^4 + x + 1$. The remaining basis vectors are represented as follows:

$\underline{v}_2$ corresponds to $(x + 1)^3 = x^3 + x^2 + x + 1$
$\underline{v}_0\underline{v}_1$ corresponds to $(x + 1)^4 = x^4 + 1$
$\underline{v}_0\underline{v}_2$ corresponds to $(x + 1)^2 = x^2 + 1$
$\underline{v}_1\underline{v}_2$ corresponds to $(x + 1) = x + 1$
$\underline{v}_0\underline{v}_1\underline{v}_2$ corresponds to $(x + 1)^0 = 1$
$\underline{I}$ corresponds to $(x + 1)^7 = x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$

The basis vectors for the first representation of R(r,m) are the same as the basis vectors for the representation in terms of characteristic functions of subsets (the basis flats) of AG(m,2). The dimension of the basis flats of AG(m,2) of R(r,m) $\leq m - r$. Therefore, the code words of R(r,m) are sums of characteristic functions of basis flats of dimension $\leq m - r$. In fact, a code word is in R(r,m) if and only if it is the sum of characteristic functions of the basis flats of dimension $\leq$ m - r.

Some Properties of R - M Codes

R(r,m) has minimum distance 2^{m-r}.

Proof:

In order to show this property, we need the following theorem from Massey et. al.

Theorem 4:

Let $P(x) = \sum_{i=0}^{t} b_i(x+c)^i$; where $b_t \neq 0$, and let i(0) be the smallest index i for which $b_i \neq 0$. Then $w(P(x)) \geq w((x+c)^{i(0)})$, where w(P(x)) is the number of nonzero coefficients of P(x). For a proof of theorem 4 see reference 2.

Now let P(x) be a nonzero code word in R(r,m). $P(x) = \sum_{i=0}^{n-1} b_i\ (x+1)^i$; where b_i = 0 or 1 and $(x+1)^i$ is a basis vector. By theorem 4 $w(P(x)) \geq w((x+1)^{i(0)}$. As $(x+1)^{i(0)}$ is a basis vector , therefore, by Lemma 2 and by definition of R(r,m), $w((x+1))^{i(0)} = 2^{m-s}$, where $0 \leq s \leq r$. Hence $w(P(x)) \geq 2^{m-s} \geq 2^{m-r}$; therefore, $d_{min}R(r,m) \geq 2^{m-r}$. By Lemma 2, $w(v_{i(1)}\underline{v}_{i(2)}\cdots\underline{v}_{i(s)})$ in F_2^n is 2^{m-r}. Therefore, $d_{min}R(r,m) = 2^{m-r}$.

The dual of R(r,m) is R(m-r-1,m)

Proof:

We need to show that the dimension of R(r,m) and the dimension of R(m-r-1,m) sum to 2^m and that the basis vectors of R(r,m) and R(m-r-1,m) are orthogonal. The dimension of R(r,m) = k = 1 + C(m,1) + C(m,2) + ... + C(m,r). The dimension of R(m-r-1,m) = L = 1 + C(m,1) + C(m,2) + ... + C(m,m-r-1). 2^m = 1 + C(m,1) + C(m,2) + ...+ C(m,m). Therefore, 2^m - k = 1 + C(m,1) + C(m,2) + ... + C(m,m) - 1 C(m,1) - C(m,2) - ... - C(m,r) = C(m,m) + C(m,m-1) + C(m,m-2) + ...+ C(m,r+1) = 1 + C(m,1) + C(m,2) + ... + C(m,m-r-1) = L. Hence k + L = 2^m. Now let $\underline{v}_{i(1)}\underline{v}_{i(2)}\cdots\underline{v}_{i(s)}$ and $\underline{v}_{j(1)}\underline{v}_{j(2)}\cdots\underline{v}_{j(t)}$ be basis vectors of R(r,m) and R(m-r-1,m) respectively. Thus $s + t \leq r + m - r - 1 = m - 1 < m$. The standard product of these two basis vectors has the form $\underline{v}_{k(1)},\underline{v}_{k(2)},\cdots,\underline{v}_{k(u)}$ where $u < m$. By Lemma 2, $w(\underline{v}_{k(1)}\underline{v}_{k(2)}\cdots\underline{v}_{k(u)})$ is even; thus the inner product is zero and therefore the original basis vectors are orthogonal.

R(m-2,m) is the (n,n-m-1) extended Hamming code

The dimension of R(m-2,m) is 1 + C(m,1) + C(m,2) + ... + C(m,m-2) = 2^m - m - 1 = n - m - 1, the dimension of the (n,n-m-1) extended Hamming code. By definition , the Hamming code over F_2 has the n x (2^m - 1) parity check matrix **H** such that the columns of **H** are the 2^m - 1 nonzero vectors of AG(m,2) in some order. If we extend the Hamming code as per Example 4, we see that the parity check matrix of the extended Hamming code has rows $\underline{v}_0$, $\underline{v}_1$, ... , $\underline{v}_m$, $\underline{I}$ and is a generator matrix of R(1,m). But the dual of R(1,m) is R(m-2,m). Hence the parity check matrix for the (n,n-m-1) extended Hamming code is a parity check matrix for R(m-2,m). Therefore, R(m-2,m) is the (n,n-m-1) extended Hamming code.

Some Weight Enumerators of R - M Codes

The weight enumerators of all of R - M codes have not yet been found. But we can use the MacWilliams Identities to find some of these weight enumerators.

Consider R(0,m), the repetition code. R(0,m) has two words $\underline{0}$ and $\underline{I}$, so its weight is $A(z) = 1 + z^n$. The dual of R(0,m) is R(m-1,m). Using the MacWilliams Identities the weight enumerator of R(m-1,m) is B(z) =

$$1/2\ [(1+z)^n\ (1+((1-z)/(1+z))^n] = 1/2\ [(1+z)^n+(1-z)^n]$$

$$= 1/2\ [2\ (z^n+C(n,2)z^{n-2}+C(n,4)z^{n-4}+\ldots+C(n,n-2)z^2+1)]$$

$$= z^n + C(n,2)z^{n-2} + C(n,4)z^{n-4} + \ldots + C(n,n-2)z^2 + 1$$

Next we consider R(1,m). R(1,m) has 2^{m+1} code words; $\underline{0}$ and $\underline{I}$ are two of the code words, so we are left with $2^{m+1} - 2$ code words to account for. The basis vectors of R(1,m) are $\underline{v}_0, \underline{v}_1, \ldots, \underline{v}_{m-1}$, and $\underline{I}$. First we consider code words formed by all possible sums of the set $\{\underline{v}_0, \underline{v}_1, \ldots, \underline{v}_{m-1}\}$. Let $\underline{v}_i = (a_0, a_1 \ldots, a_{n-1})$ and $\underline{v}_k = (b_0, b_1 \ldots, b_{n-1})$ be two different elements of this set of basis vectors. Let A be the set of indices j of the components a_j of $\underline{v}_i$ such that $a_j = 1$, $j \in \{0,1,\ldots, n-1\}$, and B be the set of indices j of the components b_j of $\underline{v}_k$ such that $b_j = 1$. Since $w(\underline{v}_i) = w(\underline{v}_k) = 2^{m-1}$, then $|A| = |B| = 2^{m-1}$. The set $A \cap B$ is the set of indices j such that $a_j = 1$ and $b_j = 1$. The product $\underline{v}_i\underline{v}_k$ has a 1 in every component a_jb_j such that $a_j = 1$ and $b_j = 1$. Therefore $|A \cap B| = w(\underline{v}_i\underline{v}_k) = 2^{m-2}$ by Lemma 2. The symmetric difference between the sets A and B, $(A\backslash A\cap B) \cup (B\backslash A\cap B)$, is the set of indices j such that a_j

$= 1$ and $b_j = 0$ or $a_j = 0$ and $b_j = 1$. The sum $\underline{v}_i + \underline{v}_k$ has a 1 in every component $a_j + b_j$ whenever $a_j = 1$ and $b_j = 0$ or $a_j = 0$ and $b_j = 1$. Therefore $w(\underline{v}_i+\underline{v}_k) = |(A\backslash A\cap B) \cup (B\backslash A\cap B)| = |A| + |B| - 2|A\cap B| = 2^{m-1} + 2^{m-1} - 2(2^{m-2}) = 2^{m-1}$.

Now we add a third basis vector $\underline{v}_h = (c_0, c_1, \ldots c_{n-1})$ to $\underline{v}_i + \underline{v}_k$ and we wish to determine $w(\underline{v}_h + \underline{v}_i + \underline{v}_k)$. Using the same technique as above, we first determine $w(\underline{v}_h(\underline{v}_i + \underline{v}_k)) = w(\underline{v}_h\underline{v}_i + \underline{v}_h\underline{v}_k)$. Let the set A' be the set of indices j of the components c_ja_j of $\underline{v}_h\underline{v}_i$ such that $c_ja_j = 1$ and let the set B' be the set of indices j of the components c_jb_j of $\underline{v}_h\underline{v}_k$ such that $c_jb_j = 1$. Since $w(\underline{v}_h\underline{v}_i) = w(\underline{v}_h\underline{v}_k) = 2^{m-2}$, $|A'| = |B'| = 2^{m-2}$. The set $A' \cap B'$ is the set of indices j such that $c_ja_j = 1$ and $c_jb_j = 1$. Therefore $|A' \cap B'| = w(\underline{v}_h\underline{v}_i \cdot \underline{v}_h\underline{v}_k) = w(\underline{v}_h\underline{v}_i\underline{v}_k) = 2^{m-3}$. The set $(A'\backslash A'\cap B') \cup (B'\backslash A'\cap B')$ is the set of indices j such that $c_ja_j = 1$ and $c_jb_j = 0$ or $c_ja_j = 0$ and $c_ja_j = 1$. The sum $\underline{v}_h\underline{v}_i + \underline{v}_h\underline{v}_k$ has a 1 in every component $c_ja_j + c_jb_j$ such that $c_ja_j = 1$ and $c_jb_j = 0$ or $c_ja_j = 0$ and $c_jb_j = 1$. Therefore $w(\underline{v}_h(\underline{v}_i + \underline{v}_k)) = 2^{m-2} + 2^{m-2} - 2(2^{m-3}) = 2^{m-2}$. Now we can determine $w(\underline{v}_h + (\underline{v}_i + \underline{v}_k)) = w(\underline{v}_h + \underline{v}_i + \underline{v}_k)$. Let the set A" be the set of indices j of the components c_j of $\underline{v}_h$ such that $c_j = 1$ and let the set B" be the set of indices j of the components (a_j+b_j) of $(\underline{v}_i+\underline{v}_k)$ such that $(a_j+b_j) = 1$. Since $w(\underline{v}_h) = w(\underline{v}_i + \underline{v}_k) = 2^{m-1}$, $|A"| = |B"| = 2^{m-1}$. The set $A" \cap B"$ is the set of indices j such that $c_j = 1$ and $a_j + b_j = 1$. Therefore $|A" \cap B"| = w(\underline{v}_h(\underline{v}_i+\underline{v}_k)) = 2^{m-2}$. The set $(A"\backslash A"\cap B") \cup (B"\backslash A"\cap B")$ is the set of indices j such that $c_j = 1$ and $(a_j+b_j) = 0$ or $c_j = 0$ and

$(a_j+b_j) = 1$. The sum $\underline{v}_h + (\underline{v}_i + \underline{v}_k)$ has a 1 in every component $c_j + (a_j+b_j)$ such that $c_j = 1$ and $(a_j+b_j) = 0$ or $c_j = 0$ and $(a_j+b_j) = 1$. Therefore $w(\underline{v}_h + \underline{v}_i + \underline{v}_k) = 2^{m-1} + 2^{m-1} - 2(2^{m-2}) = 2^{m-1}$.

Next we add a fourth basis vector $\underline{v}_l$ to $\underline{v}_h + \underline{v}_i + \underline{v}_k$ and we wish to determine $w(\underline{v}_l + \underline{v}_h + \underline{v}_i + \underline{v}_k)$. Note that when determining $w(\underline{v}_i + \underline{v}_k)$ we defined the sets A and B of indices j in $\underline{v}_i$ and $\underline{v}_k$ such that $|A| = |B| = w(\underline{v}_i) = w(\underline{v}_k)$ and $|A \cap B| = w(\underline{v}_i\underline{v}_k)$. Then $|(A\backslash A\cap B) \cup (B\backslash A\cap B)| = w(\underline{v}_i + \underline{v}_k)$. When determining $w(\underline{v}_h + \underline{v}_i + \underline{v}_k)$ we defined the sets A' and B' of indices j in $\underline{v}_h\underline{v}_i$ and $\underline{v}_h\underline{v}_k$ such that $|A'| = |B'| = w(\underline{v}_h\underline{v}_i) = w(\underline{v}_h\underline{v}_k)$ and $|A' \cap B'| = w(\underline{v}_h\underline{v}_i\underline{v}_k)$. Then $|(A'\backslash A'\cap B') \cup (B'\backslash A'\cap B')| = w(\underline{v}_h(\underline{v}_i + \underline{v}_k))$. Thereafter we defined the sets A" and B" of indices j in $\underline{v}_h$ and $(\underline{v}_i + \underline{v}_k)$ such that $|A''| = |B''| = w(\underline{v}_h) = w(\underline{v}_i + \underline{v}_k)$ and $|A'' \cap B''| = w(\underline{v}_h(\underline{v}_i + \underline{v}_k))$. Then $|(A''\backslash A''\cap B'') \cup (B''\backslash A''\cap B'')| = w(\underline{v}_h + (\underline{v}_i + \underline{v}_k)) = w(\underline{v}_h + \underline{v}_i + \underline{v}_k)$. We use the same technique to find $w(\underline{v}_l + \underline{v}_h + \underline{v}_i + \underline{v}_k)$. In order to determine $w(\underline{v}_l + \underline{v}_h + \underline{v}_i + \underline{v}_k)$, we need $w(\underline{v}_l(\underline{v}_h+\underline{v}_i+\underline{v}_k))$. In order to determine $w(\underline{v}_l(\underline{v}_h+\underline{v}_i+\underline{v}_k))$, we need $w(\underline{v}_l\underline{v}_h(\underline{v}_l\underline{v}_i+\underline{v}_l\underline{v}_k))$. In order to determine $w(\underline{v}_l\underline{v}_h(\underline{v}_l\underline{v}_i+\underline{v}_l\underline{v}_k))$, we need $w(\underline{v}_l\underline{v}_h\underline{v}_i \cdot \underline{v}_l\underline{v}_h\underline{v}_k) = w(\underline{v}_l\underline{v}_h\underline{v}_i\underline{v}_k) = 2^{m-4}$ by Lemma 2. So $w(\underline{v}_l\underline{v}_h(\underline{v}_l\underline{v}_i+\underline{v}_l\underline{v}_k)) = w(\underline{v}_l\underline{v}_h\underline{v}_i \pm \underline{v}_l\underline{v}_h\underline{v}_k)$, by Lemma 2 $w(\underline{v}_l\underline{v}_h\underline{v}_i) = w(\underline{v}_l\underline{v}_h\underline{v}_k) = 2^{m-3}$. Therefore, $w(\underline{v}_l\underline{v}_h(\underline{v}_l\underline{v}_i + \underline{v}_l\underline{v}_k)) = 2^{m-3} + 2^{m-3} - 2(2^{m-4}) = 2^{m-3}$. So $w(\underline{v}_l(\underline{v}_h + \underline{v}_i + \underline{v}_k)) = w(\underline{v}_l\underline{v}_h + \underline{v}_l\underline{v}_i + \underline{v}_l\underline{v}_k)$, $w(\underline{v}_l\underline{v}_h) = w(\underline{v}_l\underline{v}_i) = w(\underline{v}_l\underline{v}_k) = 2^{m-2}$. Therefore, $w(\underline{v}_l(\underline{v}_h + \underline{v}_i + \underline{v}_k) = 2^{m-2} + 2^{m-2} - 2(2^{m-3}) = 2^{m-2}$. Finally, $w(\underline{v}_l + \underline{v}_h + \underline{v}_i + \underline{v}_k) = w(\underline{v}_l + (\underline{v}_h + \underline{v}_i +$

$\underline{v}_k$)), w($\underline{v}_l$) = w($\underline{v}_h + \underline{v}_i + \underline{v}_k$) = 2^{m-1}. Therefore, w($\underline{v}_l$ + ($\underline{v}_h$ +$\underline{v}_i$ + $\underline{v}_k$)) = $2^{m-1} + 2^{m-1} - 2(2^{m-2}) = 2^{m-1}$.

In general, to find w($\sum_{i=0}^{s-1}\underline{v}_i$), that is , the weight of the sum of any s basis vectors, we need to determine w($\sum_{i=0}^{s-2}\underline{v}_i$) and use Lemma 2 to determine w($\prod_{i=0}^{s-1}\underline{v}_i$) = 2^{m-s}. Then we work through s steps. If we assume that w($\sum_{i=0}^{s-2}\underline{v}_i$) = 2^{m-1} and that w($\underline{v}_s$ ($\sum_{i=0}^{s-2}\underline{v}_i$)) = 2^{m-2} from doing s - 1 steps, it follows that w($\underline{v}_s$ + ($\sum_{i=0}^{s-2}\underline{v}_i$)) = w($\sum_{i=0}^{s-1}\underline{v}_i$) = $2^{m-1} + 2^{m-1} - 2(2^{m-2}) = 2^{m-1}$. Therefore, the weight of any code word formed by the sums of basis vectors {$\underline{v}_0, \underline{v}_1, \ldots, \underline{v}_{m-1}$} is 2^{m-1}. There are C(m,1) + C(m,2) + ... + C(m,m) = 2^m - 1 such code words in R(1,m).

The remaining code words of R(1,m) are formed by adding the vector $\underline{I}$ to each of the 2^m - 1 code words formed by the sums of {$\underline{v}_0, \underline{v}_1, \ldots \underline{v}_{m-1}$}. Since the length of each of these code words is 2^m it follows that the remaining code words have weight 2^{m-1}. Also, it follows that there are 2^m - 1 such code words. Therefore, R(1,m) has one code word $\underline{0}$ of weight 0, one code word $\underline{I}$ of weight 2^m, and 2^{m+1} -2 code words of weight 2^{m-1}. Hence the weight enumerator of R(1,m) is A(z) = 1 + $(2^{m+1}-2)z^t + z^n$, where t = 2^{m-1}. The dual of R(1,m) is R(m-2,m). Using the MacWilliams Identities the weight enumerator of R(m-2,m) is B(z) =

$$1/2^{m+1}\ [(1+z)^n\ [1+(2^{m+1}-2)[(1-z)/(1+z)]^t+[(1-z)/(1+z)]^n]$$

$$= 1/2^{m+1}\ [(1+z)^n+(1-z)^n+(2^{m+1}-2)(1-z)^t(1+z)^t]$$

$$= 1/2^m\ [z^n + C(n,2)z^{n-2} + C(n,4)z^{n-4} +\ldots + C(n,n-2)z^2 + 1 + (2^m-1)(1-z^2)^t].$$

The Reed Method of Decoding R - M Codes

Consider $R(r,m)$, Let $\underline{u}$ be the information vector $(u_1,u_2,\ldots,u_k)$, where $k = 1 + C(m,1) + C(m,2) + \ldots + C(m,r)$. Let G be the $k \times 2^m$, where $n = 2^m$, matrix whose rows are the basis vectors for $R(r,m)$, that is, G is the generator matrix of $R(r,m)$. Then similarly to the parity check matrix method of encoding, $\underline{u}$ is encoded into the vector $\underline{b}$ as follows:

$$\underline{u}G = \underline{b} = (b_0,b_1,b_3,\ldots b_{n-1}) = \text{the sent vector.}$$

The problem of the decoder is to determine the information vector $\underline{u}$ from the received vector even though some errors may have occurred during transmission. It is best to first see an example of the Reed method of decoding. Consider $R(2,3)$, $k = 1 + C(3,1) + C(3,2) = 7$, so we have seven information symbols, say $\underline{u} =(u_I,u_0,u_1,u_2,u_{01},u_{02},u_{12})$ (we relabeled the components of $\underline{u}$ for simplicity). The basis vectors for $R(2,3)$ are as follows

$\underline{I} = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$

$\underline{v}_0 = (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)$

$\underline{v}_1 = (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$

$\underline{v}_2 = (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)$

$\underline{v}_0\underline{v}_1 = (0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$

$\underline{v}_0\underline{v}_2 = (0\ 0\ 0\ 0\ 0\ 1\ 0\ 1)$

$\underline{v}_1\underline{v}_2 = (0\ 0\ 0\ 0\ 0\ 0\ 1\ 1)$

We encode $\underline{u}$ as $u_I\underline{I} + u_0\underline{v}_0 + u_1\underline{v}_1 + u_2\underline{v}_2 + u_{01}\underline{v}_0\underline{v}_1 + u_{02}\underline{v}_0\underline{v}_2 + u_{12}\underline{v}_1\underline{v}_2$ $= (b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7) = \underline{b}$, the sent vector. Then $\underline{b} =$

$u_I(1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$

$+\ u_0(0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)$

$+\ u_1(0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$

$+\ u_2(0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)$

$+\ u_{01}(0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$

$+\ u_{02}(0\ 0\ 0\ 0\ 0\ 1\ 0\ 1)$

$\underline{+\ u_{12}(0\ 0\ 0\ 0\ 0\ 0\ 1\ 1)}$

$(b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7)$

So, $b_0 = u_I$

$b_1 = u_I + u_0$

$b_2 = u_I + u_1$

$b_3 = u_I + u_0 + u_1 + u_{01}$

$b_4 = u_I + \quad + u_2$

$b_5 = u_I + u_0 + u_2 + u_{02}$

$b_6 = u_I + u_1 + u_2 + u_{12}$

$b_7 = u_I + u_0 + u_1 + u_2 + u_{01} + u_{02} + u_{12}$

In absence of errors in $\underline{b}$ we see that $b_0 + b_1 + b_2 + b_3 = b_4 + b_5 + b_6 + b_7 = u_{01}$. Hence we have two independent determinations for u_{01}. Likewise, $b_0 + b_1 + b_4 + b_5 = b_2 + b_3 + b_6 + b_7 = u_{02}$ and $b_0 + b_2 + b_4 + b_6 = b_1 + b_3 + b_5 + b_7 = u_{12}$.

After u_{01}, u_{02}, and u_{12} have been determined, we subtract $u_{01}\underline{v}_0\underline{v}_1 + u_{02}\underline{v}_0\underline{v}_2 + u_{12}\underline{v}_1\underline{v}_2$ from $\underline{b}$ and we are left with $u_I\underline{I} + u_0\underline{v}_0 + u_1\underline{v}_1 + u_2\underline{v}_2 = \underline{b}' = (b_0',b_1',b_2',b_3',b_4',b_5',b_6',b_7')$

$$
\begin{aligned}
&= u_I(1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)\\
&+ u_0(0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)\\
&+ u_1(0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)\\
&\underline{+ u_2(0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)\quad}
\end{aligned}
$$

So,
$$
\begin{aligned}
b_0' &= u_I\\
b_1' &= u_I + u_0\\
b_2' &= u_I \qquad\quad + u_1\\
b_3' &= u_I + u_0 + u_1\\
b_4' &= u_I + \qquad\qquad + u_2\\
b_5' &= u_I + u_0 \qquad + u_2\\
b_6' &= u_I \qquad\quad + u_1 + u_2\\
b_7' &= u_I + u_0 + u_1 + u_2
\end{aligned}
$$

In absence of errors,we see that

$b_0' + b_1' = b_2' + b_3' = b_4' + b_5' = b_6' + b_7' = u_0$,

$b_0' + b_2' = b_1' + b_3' = b_4' + b_6' = b_5' + b_6' = u_1$,

$b_0' + b_4' = b_1' + b_5' = b_2' + b_6' = b_3' + b_7' = u_2$,

there are four independent determinations for each u_0, u_1, and u_2. After u_0, u_1, and u_2 have been determined then $u_I\underline{I} + u_0\underline{v}_0 + u_1\underline{v}_1 + u_2\underline{v}_2$ is subtracted from $\underline{b}'$. Without errors we are left with $u_I\underline{I} = b'' = (b_0'',b_1'',b_2'',b_3'',b_4'',b_5'',b_6'',b_7'')$ and $u_I = b_0'' = b_1'' = b_2'' = b_3'' = b_4'' = b_5 = b_6'' = b_7''$.

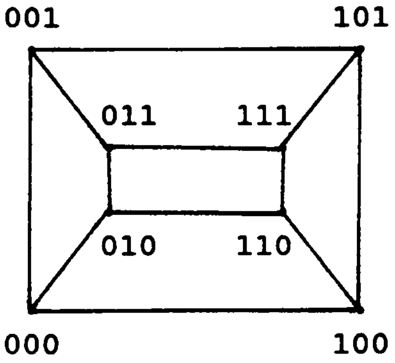
The method for determining which sums of symbols in the received vector should equal a given information symbol can be described as follows. Arrange the basis vectors in rows as above, that is, $\underline{I}$ is the first row, $\underline{v}_0$ is the second row, $\underline{v}_1$ is the third row, continue until you have k rows. Call the component corresponding to the j^th^ 0 in $\underline{v}_i$ and the component corresponding to the j^th^ 1 in $\underline{v}_i$ matching components for $\underline{v}_i$. Then the 2^{m-1} sums of matching components ($w(\underline{v}_i) = 2^{m-1}$) of $\underline{v}_i$ are used in determining u_i. The 2^{m-2} sums of components used to determine u_{ij} are found by taking a matching pair of components for $\underline{v}_i$, and with each of them the component that matches for $\underline{v}_j$. The process can be continued in a similar manner.

To see this consider R(2,3), $\underline{v}_1 = (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$ and $\underline{v}_2 = (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)$. $\underline{v}_1$'s first zero component is its first component and its first 1 component is its third component, therefore $b_0 + b_2 = u_1$ if no errors have occurred. Likewise, $b_1 + b_3 = b_4 + b_6 = b_5 + b_8 = u_1$. $\underline{v}_2$'s first 0 is its first component and its first 1 is its fourth component, therefore $b_0 + b_4 = u_2$. Likewise, $b_1 + b_5 = b_2 + b_6 = b_3 + b_7 = u_2$. The two sums used to determine u_{12} are $b_0 + b_4 + b_2 + b_6 = b_0 + b_2 + b_4 + b_6$ and $b_1 + b_5 + b_3 + b_7 = b_1 + b_3 + b_5 + b_7$.

In general, the decoder knows the determinations or sums to be used to recover the information vector $\underline{u}$. It first determines the last C(m,r) components of $\underline{u}$. There are 2^{m-r} independent sums ($w(\underline{v}_0 \cdots \underline{v}_r)$) for each of these components. The value occurring most frequently in these sums is taken as the value for the component. After the last C(m,r) components of $\underline{u}$ have been determined, the basis vectors corresponding to the components that equal one are subtracted from the received vector $\underline{b}$ to get $\underline{b}'$. The next C(m,r-1) components are similarly determined. There are 2^{m-r-1} independent sums for each of these components. The basis vectors corresponding to these components that equal one are then subtracted from $\underline{b}'$to get $\underline{b}''$. This process continues until all of the components of $\underline{u}$ have been determined.

Note that if at least one of the last C(m,r) components are incorrectly determined in the first step, then $\underline{u}$ is incorrectly determined, that is, the last $r - 1$ steps are dependent on the first step. Therefore, this decoding scheme can correct $2^{m-r-1} - 1$ or fewer errors in transmission. Recall that if the weight of every nonzero code word in a linear code is at least $2t + 1$, then the code can correct any t or fewer errors. The minimum weight of every nonzero code of R(r,m) is $2^{m-r} = 2(2^{m-r-1} - 1/2) + 1 > 2(2^{m-r-1} - 1) + 1$. Therefore, the Reed decoding scheme can correct the maximum number of errors that this code is capable of correcting.

The geometric interpretation of these codes can be used to determine which sums of symbols in the received vector should equal a given information symbol. As stated earlier $\mathbf{E}$ is the matrix with columns $\underline{x}_j \in AG(m,2)$. Place these points of AG(m,2) at the corresponding points in m-dimensional space. There are 2^m points and each column $\underline{x}_j$ of $\mathbf{E}$ corresponds to one of these points. For example, if $m = 3$, we have



The vector $\underline{v}_i$ is the characteristic function of the (m-1) dimensional flat A_i, so $\underline{v}_i$ has a one in every component a_j that corresponds to a point $\underline{x}_j$ of A_i. The vector $\underline{v}_i\underline{v}_k$ is the characteristic function of the (m-2) - flat $A_i \cap A_k$, so $\underline{v}_i\underline{v}_k$ has a one in every component c_j that corresponds to a point $\underline{x}_j$ of $A_i \cap A_k$. The vector $\underline{v}_{i(1)} \cdot \cdot \cdot \underline{v}_{i(s)}$ is the characteristic function of the (m-s) - flat $A_{i(1)} \cap \ldots \cap A_{i(s)}$, so $\underline{v}_{i(1)} \cdot \cdot \cdot \underline{v}_{i(s)}$ has a one in every component d_j that corresponds to a point $\underline{x}_j$ of $A_{i(1)} \cap \ldots \cap A_{i(s)}$. Every basis vector of an rth order code corresponds to a flat of AG(m,2). The dimension of the flat is the weight of the characteristic function of the flat.

The dual of R(r,m) is R(m-r-1,m), that is, the generator matrix for R(m-r-1,m) is the parity check matrix for R(r,m); therefore, each product of m - r - 1 or fewer basis vectors $\underline{v}_i$ defines a parity check rule. For example, if m = 4 and r = 1, then m - r - 1 = 2, that is, R(1,4) is the dual of R(2,4). A generator matrix for R(1,4) has rows $\underline{I}$, $\underline{v}_0$, $\underline{v}_1$, $\underline{v}_2$, $\underline{v}_3$. This matrix is a parity check matrix for R(2,4). If $\underline{b}$ = $(b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8,b_9,b_{10},b_{11},b_{12},b_{13},b_{14},b_{15})$ is a code word of R(2,4), then

$$\begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \cdot \\ \cdot \\ b_{15} \end{bmatrix} = \underline{0} \ ,$$

and $b_0 + b_2 + \ldots\ldots\ldots\ldots + b_{15} = 0$,

$b_1 + b_3 + b_5 + b_7 + b_9 + b_{11} + b_{13} + b_{15} = 0$,

$b_2 + b_3 + b_6 + b_7 + b_{10} + b_{11} + b_{14} + b_{15} = 0$,

$b_4 + b_5 + b_6 + b_7 + b_{12} + b_{13} + b_{14} + b_{15} = 0$,

$b_8 + b_9 + b_{10} + b_{11} + b_{12} + b_{13} + b_{14} + b_{15} = 0$.

If the symbols of R(2,4), that is, the components $\underline{b}$, are placed at the corresponding points $\underline{x}_j$ in 4 - dimensional space, each parity check rule is a parity check on the symbols associated with the points on a 3 - flat or a 4 - flat of AG(4,2).

The weight of a basis vector of R(m-r-1,m) is 2^{m-r-1} or greater. Therefore, if the symbols of $\underline{b}$ of R(r,m) are placed at the corresponding points $b_j = \underline{x}_j$ in AG(m,2), each parity check rule is a parity check on the symbols associated with the points on a m-r-1 or greater dimensional flat.

Now that we have associated the points of AG(m,2) with the code words of R(r,m) we wish to find a set of points that correspond to the sums used to determine the information symbols $\underline{u}$. Such a set of points will serve this purpose if its characteristic function has an odd number of 1's in common with the basis vector whose coefficient is to be determined but an even number of 1's in common with all other basis vectors that are a product of r or fewer vectors, because then in the sum the desired coefficient will not cancel but all other coefficients will. For example, earlier we found that for R(2,3) that $u_{01} = b_0 + b_1 + b_2 + b_3 = b_4 + b_5 + b_6 + b_7$. The set $\{b_0,b_1,b_2,b_3\}$ and the set $\{b_4,b_5,b_6,b_7\}$ correspond to the 2 - flats $\{\underline{x}_0,\underline{x}_1,\underline{x}_2,\underline{x}_3\}$ and $\{\underline{x}_4,\underline{x}_5,\underline{x}_6,\underline{x}_7\}$ respectively. The characteristic functions of these 2 - flats are (1 1 1 1 0 0 0 0) and (0 0 0 0 1 1 1 1) respectively. The basis vector $\underline{v}_0\underline{v}_1$ = (0 0 0 1 0 0 0 1) has one 1 in common with the first characteristic function and one 1 in common with the second characteristic function but all other basis vectors $\underline{v}_0\underline{v}_2$, $\underline{v}_1\underline{v}_2$, $\underline{v}_0$, $\underline{v}_1$, $\underline{v}_2$ and $\underline{I}$ have an even number of 1's in common with these two characteristic functions. Now consider $b_0 + b_1 + b_2 + b_3$ = (1 1 1 1 0 0 0 0) • $(b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7)$ = (1 1 1 1 0 0 0 0) • [u_I(1 1 1 1 1 1 1 1) + u_0(0 1 0 1 0 1 0 1) + u_1(0 0 1 1 0 0 1 1) + u_2(0 0 0 0 1 1 1 1) + u_{01}(0 0 0 1 0 0 0 1) + u_{02}(0 0 0 0 0 1 0 1) + u_{12}(0 0 0 0 0 0 1 1)] (here • indicates the dot product). Since all basis vectors except $\underline{v}_0\underline{v}_1$ have an even number of 1's in common with (1 1 1 1 0 0 0 0), this dot product is equal to u_{01}.

Since we are seeking a set of points or a flat of AG(m,2) whose characteristic function has an odd number of 1's in common with the basis vector whose coefficient is desired and an even number of 1's in common with all other basis vectors of r or fewer vectors $\underline{v}_i$, this flat must intersect the (m-r) - basis flat associated with the basis vector whose coefficient is desired at only one point and intersect all other (m-r) or greater dimensional basis flat at an even number of points.

We start decoding by finding the coefficients of the last C(m,r) basis vectors that are products of r vectors $\underline{v}_i$. A product of r basis vectors has 2^{m-r} 1's. So we wish to find a flat of AG(m,2) (the set of points that correspond to a sum) that intersects the (m-r) - basis flat at one point, but intersects all other (m-r) or greater dimensional basis flats at an even number of points. A flat that has this property is the flat perpendicular to the (m-r) - basis flat whose characteristic functions' coefficient is desired. Therefore, for each (m-r) - basis flat there are 2^{m-r} perpendicular flats. Hence 2^{m-r} sums will determine the desired coefficient.

First we look at an example to see how we find these perpendicular flats. Consider R(2,3) and the 1 - flat $A_0 \cap A_1$ $= \{\underline{x}_j \in AG(3,2) \mid \beta_{0,j} = 1 \text{ and } \beta_{1,j} = 1\} = \{\underline{x}_3, \underline{x}_7\} = \{(110)^t, (111)^t\}$. $A_0 \cap A_1$ consists of all 3 - tuples $\underline{x}_j$ of AG(3,2) such that their first and second components are 1 and the third component is a 0 or a 1. We desire to find all 2 - flats perpendicular to $A_0 \cap A_1$. First consider $\underline{x}_3 = (110)^t$, let $S_3 = \{\underline{x}_j \in AG(3,2) \mid \beta_{2,j} = 0 \text{ and } \beta_{0,j} = 0 \text{ or } 1 \text{ and } \beta_{1,j} = 0 \text{ or } 1\}$, that is, S_3 is the set of all points $\underline{x}_j$ of AG(2,3) that have a 0 in their third component and have all combinations of 0 or 1 in their first and second components. We find that S_3 $= \{(000)^t, (100)^t, (010)^t, (110)^t\} = \{\underline{x}_0, \underline{x}_1, \underline{x}_2, \underline{x}_3\}$. S_3 has four points since we are varying two components of the 3 - tuples, and $S_3 \cap (A_0 \cap A_1) = \{\underline{x}_3\}$. Since S_3 contains no points $\underline{x}_j$ that have a 1 in their third component, $|S_3 \cap (A_0 \cap A_2)| = 0$, $|S_3 \cap (A_1 \cap A_2)| = 0$ and $|S_3 \cap A_2| = 0$. S_3 contains all points of AG(3,2) that have a 1 in their first or second components; therefore, $|S_3 \cap A_0| = 2$ and $|S_3 \cap A_1| = 2$. Hence, S_3 is the 2 - flat perpendicular to the basis flat $A_0 \cap A_1$ that passes through the point $\underline{x}_3$. Now let $S_7 = \{\underline{x}_j \in AG(3,2) \mid \beta_{2,j} = 1$ and $\beta_{0,j} = 0$ or 1 and $\beta_{1,j} = 0$ or $1\}$. So $S_7 = \{(001)^t,(101)^t,(011)^t,(111)^t\} = \{\underline{x}_4,\underline{x}_5,\underline{x}_6,\underline{x}_7\}$. Since we are varying two components of the 3 -tuples $\underline{x}_j$, S_7 contains four points, and $S_7 \cap (A_0 \cap A_1) = \{\underline{x}_7\}$. As S_7 contains all points of AG(3,2) that have a 1 in their third component and all combinations of 0 or 1 in their first and second components, $|S_7 \cap (A_0 \cap A_2)| = 2$, $|S_7 \cap (A_1 \cap A_2)| = 2$, $|S_7 \cap A_0| = 2$ and $|S_7 \cap A_1| = 2$, and $|S_7 \cap A_2| = 4$. Therefore, S_7 is the 2 - flat perpendicular to the basis flat $A_0 \cap A_1$ that passes through the point $\underline{x}_7$. Similarly we can find the 2 - flats perpendicular to the basis flats $A_0 \cap A_2$ and $A_1 \cap A_2$. For each point of $A_0 \cap A_2$, with second component $\beta_{1,j}$, fix the point's second component to be $\beta_{1,j}$ and vary its first and third component over 0 or 1 to find the perpendicular flat that passes though chosen point of $A_0 \cap A_2$. For each point of $A_1 \cap A_2$, with first component $\beta_{0,j}$, fix the point's first component to be $\beta_{0,j}$, and vary its second and third components over 0 or 1 to find the perpendicular flat that passes through the chosen point of $A_1 \cap A_2$. We use the same technique to find the flats perpendicular to the 2 -dimensional basis flats A_0, A_1, and A_2. Consider $A_0 = \{\underline{x}_j \in AG(3,2) \mid \beta_{0,j} = 1\} = \{(100)^t,(110)^t,(101)^t,(111)^t\} = \{\underline{x}_1,\underline{x}_3,\underline{x}_5,\underline{x}_7\}$. Choose $\underline{x}_1 = (100)^t$, let $S_1 = \{ \underline{x}_j \in AG(3,2) \mid \beta_{1,j} = 0$ and $\beta_{2,j} = 0$ and $\beta_{0,j} = 0$ or $1\}$. So $S_1 = \{(000)^t,(100)^t\} = \{\underline{x}_0,\underline{x}_1\}$. Here we are varying one component of the 3 - tuple $\underline{x}_1$ and fixing two components, therefore $|S_1| = 2$. Because S_1 contains no points that have a 1 in their second or third components $|S_1 \cap A_1| = 0$ and $|S_1 \cap A_2| = 0$. Hence S_1 is the 1 -flat perpendicular to the basis flat A_0 that passes through the point $\underline{x}_1$. We will find just one more flat perpendicular to A_0. Choose $\underline{x}_7 = (111)^t$, let $S_7 = \{\underline{x}_j \in AG(3,2) \mid \beta_{1,j} = 1$ and $\beta_{2,j} = 1$ and $\beta_{0,j} = 0$ or $1\}$. So $S_7 = \{(011)^t,(111)^t\} = \{\underline{x}_6,\underline{x}_7\}$. Since S_7 contains all points of AG(3,2) that have a one in their second and third component $|S_7 \cap A_1| = 2$ and $|S_7 \cap A_2| = 2$. Hence S_7 is the 1 - flat perpendicular to A_0 that passes through the point $\underline{x}_7$. $S_3 = \{\underline{x}_j \in AG(3,2) \mid \beta_{1,j} = 1$ and $\beta_{2,j} = 0$ and $\beta_{0,j} = 0$ or $1\}$ $= \{010)^t,(110)^t\} = \{\underline{x}_2,\underline{x}_3\}$ is the 1 - flat perpendicular to the basis flat A_0 that passes through $\underline{x}_3$. $S_5 = \{\underline{x}_j \in AG(3,2) \mid \beta_{1,j} = 0$ and $\beta_{2,j} = 1$ and $\beta_{0,j} = 0$ or $1\} = \{(001)^t,(101)^t\} = \{\underline{x}_4,\underline{x}_5\}$ is the 1 - flat perpendicular to the basis flat A_0 that passes through the point $\underline{x}_5$. For each point of A_1, with first and third components $\beta_{0,j}$ and $\beta_{2,j}$, fix the point's first and third components to be $\beta_{0,j}$ and $\beta_{2,j}$ and vary its second component over 0 or 1 to find the perpendicular flat that passes through the chosen point of A_1. For each point of A_2, with first and second components $\beta_{0,j}$ and $\beta_{1,j}$, fix the point's first and second components to be $\beta_{0,j}$ and $\beta_{1,j}$ and vary its third component over 0 or 1 to find the perpendicular flat that passes through the chosen point of A_2.

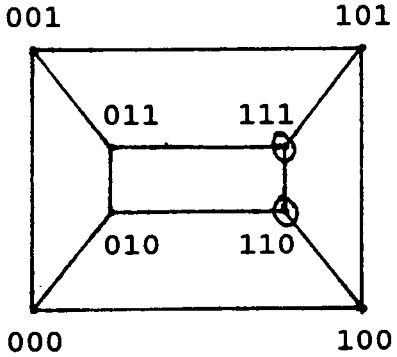
Now we want to show that for every (m-s) - basis flat ($0 < s \leq r$) of R(r,m) there is a perpendicular s - flat, that is, an s - flat which intersects the given (m-s) - basis flat at only one point and intersects every other (m-s) or fewer dimensional basis flat at an even number of points.

It is sufficient to show this for the (m-s) - basis flat $A_0 \cap A_1 \cap \ldots \cap A_{s-1} = \{\underline{x}_j \in AG(m,2) \mid \beta_{0,j} = \beta_{1,j} = \ldots = \beta_{s-1,j} = 1\}$. If the m -tuple $\underline{x}_k \in A_0 \cap \ldots \cap A_{s-1}$, then $\underline{x}_k$ has its first s components equal to 1 and its remaining m-s components equal to 0 or 1. If $\underline{x}_m \neq \underline{x}_k$ is also contained in $A_0 \cap \ldots \cap A_{s-1}$, then $\underline{x}_m$ has its first s components equal to 1 but a different sequence of 0's and 1's in its last m-s components than $\underline{x}_k$. Otherwise $\underline{x}_m = \underline{x}_k$. Let S_k be the set of points $\underline{x}_j$ of AG(m,2) whose last m-s components exactly equal the last m-s components of $\underline{x}_k$ and whose first r components are 0 or 1. If we removed the last m-s components of every point of S_k, we would be left with a set of s - tuples that have every combination of 0 or 1 in their s components. Therefore $|S_k| = 2^s$ or S_k is an s - flat of AG(m,2). First we note that $S_k \cap (A_0 \cap \ldots \cap A_{s-1}) = \underline{x}_k$, because of the uniqueness of its last m-s components. Let $A_{i(1)} \cap A_{i(2)} \cap \ldots \cap A_{i(L)}$ be any basis flat of m-s or greater dimensions. Let $N_1 = \{i(j) \mid 1 \leq j \leq L,$

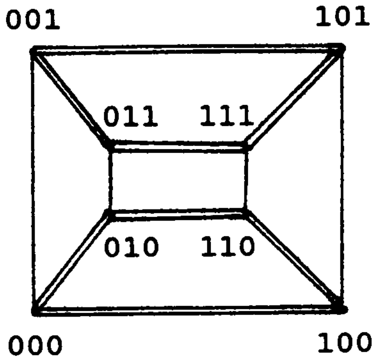
$i(j) \in \{0,1,2,\ldots s-1\}\}$ and $N_2 = \{i(j) \mid 1 \leq j \leq L,\ i(j) \in \{s,\ldots,m-s-1\}\}$. Each point of $A_{i(1)} \cap \ldots \cap A_{i(L)}$ has a 1 in each component with subscripts in $N_1 \cup N_2$. If the set of subscripts of the last m-s components of $\underline{x}_k$ which are 1 is not a subset of N_2, then $|(A_{i(1)} \cap \ldots \cap A_{i(L)}) \cap S_k| = 0$. Otherwise the intersection is not empty and there are $s - |N_1|$ components which vary arbitrarily in both S_k and $A_{i(1)} \cap \ldots \cap A_{i(L)}$. Hence there are an even number of points in $S_k \cap (A_{i(1)} \cap \ldots \cap A_{i(L)})$ when $s - |N_1| > 0$ and when $s - |N_1| = 0$ there is exactly one point $\underline{x}_k$ in the intersection.

Example 8

Consider R(2,3). $\underline{v}_0\underline{v}_1 = (00010001)$ is the characteristic function for $A_0 \cap A_1 = \{\underline{x}_3,\underline{x}_7\} = \{(110)^t,(111)^t\}$. Below these points are circled.



The perpendicular 2-flats are $\{(000)^t,(100)^t,(010)^t,(110)^t\} = \{\underline{x}_0,\underline{x}_1,\underline{x}_2,\underline{x}_3\}$ and $\{(001)^t,(101)^t,(011)^t,(111)^t\} = \{\underline{x}_4,\underline{x}_5,\underline{x}_6,\underline{x}_7\}$. Below these 2-flats are double lined.

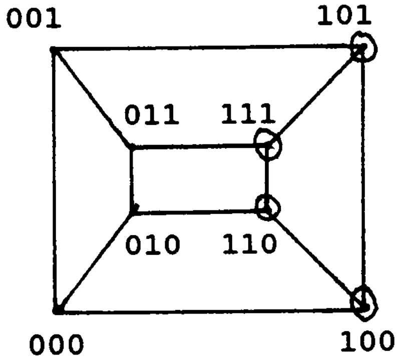


Therefore, $u_{01} = b_0+b_1+b_2+b_3 = b_4+b_5+b_6+b_7$.

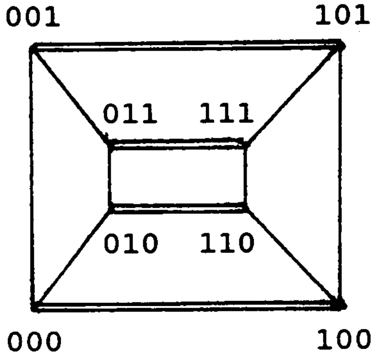
$\underline{v}_0\underline{v}_2$ = (00000101) is the characteristic function of $A_0 \cap A_2 = \{\underline{x}_5,\underline{x}_7\} = \{(101)^t,(111)^t\}$. The perpendicular 2-flats are $\{(000)^t,(100)^t,(001)^t,(101)^t\} = \{\underline{x}_0,\underline{x}_1,\underline{x}_4,\underline{x}_5\}$ and $\{(010)^t,(110)^t,(011)^t,(111)^t\} = \{\underline{x}_2,\underline{x}_3,\underline{x}_6,\underline{x}_7\}$. Therefore, $u_{02} = b_0+b_1+b_4+b_5 = b_2+b_3+b_6+b_7$.

$\underline{v}_1\underline{v}_2$ = (00000011) is the characteristic function of $A_1 \cap A_2 = \{\underline{x}_6,\underline{x}_7\} = \{(011)^t,(111)^t\}$. The perpendicular 2-flats are $\{(000)^t,(010)^t,(001)^t,(011)^t\} = \{\underline{x}_0,\underline{x}_2,\underline{x}_4,\underline{x}_6\}$ and $\{(100)^t,(110)^t,(101)^t,(111)^t\} = \{\underline{x}_1,\underline{x}_3,\underline{x}_5,\underline{x}_7\}$. Therefore, $u_{12} = b_0+b_2+b_4+b_6 = b_1+b_3+b_5+b_7$.

$\underline{v}_0$ = (01010101) is the characteristic function of $A_0 = \{\underline{x}_1,\underline{x}_3,\underline{x}_5,\underline{x}_7\} = \{(100)^t,(110)^t,(101)^t,(111)^t\}$. These points are circled below.



The perpendicular 1-flats are $\{(000)^t,(100)^t\} = \{\underline{x}_0,\underline{x}_1\},\{(001)^t,(101)^t\} = \{\underline{x}_4,\underline{x}_5\}$, $\{(010)^t,(110)^t\} = \{\underline{x}_2,\underline{x}_3\}$, and $\{(011)^t,(111)^t\} = \{\underline{x}_6,\underline{x}_7\}$. These 1-flats are double lined below.



Therefore, $u_0 = b_0'+b_1' = b_2'+b_3' = b_4'+b_5' = b_6'+b_7'$. Similarly $u_1 = b_0'+b_2' = b_1'+b_3' = b_4+b_6' = b_5'+b_7'$ and $u_2 = b_0'+b_4' = b_1'+b_5' = b_2'+b_6' = b_3'+b_7'$.

Example 9

Consider R(2,3), k = 1 + C(3,1) + C(3,2) = 7, therefore, we can send an information vector with 7 information symbols. Suppose we want to send 1 0 1 1 0 1 1. 1 0 1 1 0 1 1 is encoded as $1(\underline{I})+0(\underline{v}_0)+1(\underline{v}_1)+1(\underline{v}_2)+0(\underline{v}_0\underline{v}_1)+1(\underline{v}_0\underline{v}_2)+1(\underline{v}_1\underline{v}_2)=$

```
  1 1 1 1 1 1 1 1   (I)
+ 0 0 1 1 0 0 1 1   (v1)
+ 0 0 0 0 1 1 1 1   (v2)
+ 0 0 0 0 0 1 0 1   (v0v2)
+ 0 0 0 0 0 0 1 1   (v1v2)
  _______________
```

$= 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1 = (b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7) = \underline{b}$ the transmitted vector. Suppose that no errors occur in transmission, hence

$\underline{b}$ is the received vector. Now we decode.

$u_{01} = b_0 + b_1 + b_2 + b_3 = 1 + 1 + 0 + 0 = 0$

$u_{01} = b_4 + b_5 + b_6 + b_7 = 0 + 1 + 0 + 1 = 0$

We conclude $u_{01} = 0$.

$u_{02} = b_0 + b_1 + b_4 + b_5 = 1 + 1 + 0 + 1 = 1$

$u_{02} = b_2 + b_3 + b_6 + b_7 = 0 + 0 + 0 + 1 = 1$

We conclude $u_{02} = 1$.

$u_{12} = b_0 + b_2 + b_4 + b_6 = 1 + 0 + 0 + 0 = 1$

$u_{12} = b_1 + b_3 + b_5 + b_7 = 1 + 0 + 1 + 1 = 1$

We conclude $u_{12} = 1$.

Next we subtract $\underline{v}_1\underline{v}_2$ and $\underline{v}_0\underline{v}_2$ from the received vector $\underline{b}$ because u_{12} and u_{02} equal 1.

$$\begin{array}{rl} 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1 & (\underline{b}) \\ -\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 & (\underline{v}_1\underline{v}_2) \\ -\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 & (\underline{v}_0\underline{v}_2) \\ \hline \end{array}$$

$= 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1 = (b_0', b_1', b_2', b_3', b_4', b_5', b_6', b_7') = \underline{b}'$

$u_0 = b_0' + b_1' = 1 + 1 = 0$

$u_0 = b_2' + b_3' = 0 + 0 = 0$

$u_0 = b_4' + b_5' = 0 + 0 = 0$

$u_0 = b_6' + b_7' = 1 + 1 = 0$

We conclude $u_0 = 0$.

$u_1 = b_0' + b_2' = 1 + 0 = 1$

$u_1 = b_1' + b_3' = 1 + 0 = 1$

$u_1 = b_4' + b_6' = 0 + 1 = 1$

$u_1 = b_5' + b_7' = 0 + 1 = 1$

We conclude $u_1 = 1$.

$u_2 = b_0' + b_4' = 1 + 0 = 1$

$u_2 = b_1' + b_5' = 1 + 0 = 1$

$u_2 = b_2' + b_6' = 0 + 1 = 1$

$u_2 = b_3' + b_7' = 0 + 1 = 1$

We conclude $u_2 = 1$.

Now we subtract $\underline{v}_2$ and $\underline{v}_1$ from $\underline{b}'$,because u_2 and u_1 equal one.

$$\begin{array}{l} 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1 = \underline{b}' \\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 = \underline{v}_2 \\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 = \underline{v}_1 \\ \hline 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \end{array}$$

We conclude $u_I = 1$.

The received vector $\underline{b}$ is decoded as 1 0 1 1 0 1 1 which equals $\underline{u}$ the information vector.

Suppose one error had occurred in the received vector during transmission. We would not be able to determine u_{01},u_{02},and u_{12} by majority vote. If between two and six errors had occurred at least one of these components of $\underline{u}$ would be determined. But the case of seven errors occurring in the received vector is exactly the case of one error occurring, that is, all three of these components could not be determined. Therefore, one error could not be detected for this code. Of course, we could have calculated that R(2,3) is capable of correcting $2^{3-2-1} - 1 = 0$ errors and is capable of detecting 0 errors.

For an example to see how the Reed decoding scheme corrects an error during transmission consider R(1,3). R(1,3) can correct $2^{3-1-1} - 1 = 1$ error and can detect 2 errors. There are four information symbols for R(1,3); (u_I, u_0, u_1, u_2). Suppose we want to send 0 1 1 1. We encode 0 1 1 1 as follows: $0(\underline{I}) + 1(\underline{v}_0) + 1(\underline{v}_1) + 1(\underline{v}_2) = \underline{b} =$

$$
\begin{array}{rl}
\;\; 0\;1\;0\;1\;0\;1\;0\;1 & (\underline{v}_0) \\
+\; 0\;0\;1\;1\;0\;0\;1\;1 & (\underline{v}_1) \\
+\; 0\;0\;0\;0\;1\;1\;1\;1 & (\underline{v}_2) \\
\hline
\end{array}
$$

$= 0\;1\;1\;0\;1\;0\;0\;1 = (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7)$ = the sent vector.

Now suppose 0 1 0 0 1 0 0 1 is the received vector, that is, an error has occurred in b_2. We begin to decode the received vector as follows:

$u_0 = b_0 + b_1 = 1$

$u_0 = b_2 + b_3 = 0$

$u_0 = b_4 + b_5 = 1$

$u_0 = b_6 + b_7 = 1$

We conclude $u_0 = 1$.

$u_1 = b_0 + b_2 = 0$

$u_1 = b_1 + b_3 = 1$

$u_1 = b_4 + b_6 = 1$

$u_1 = b_5 + b_7 = 1$

We conclude $u_1 = 1$.

$u_2 = b_0 + b_4 = 1$

$u_2 = b_1 + b_5 = 1$

$u_2 = b_2 + b_6 = 0$

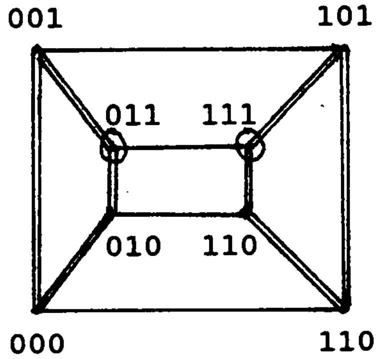
$u_2 = b_3 + b_7 = 1$

We conclude $u_2 = 1$.

Now we subtract $\underline{v}_2, \underline{v}_1$, and $\underline{v}_0$ from the received vector.

$$\begin{array}{rcccccccc}
 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
- & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
- & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
- & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
\hline
= & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}$$

We conclude $u_I = 0$ and the estimation of the information vector is 0 1 1 1, which is correct. If two errors had occurred none of the components of $\underline{u}$ would have been determined. If between three to seven errors had occurred then at least one of the components correctly or incorrectly would have been determined. And if eight errors had occurred then all the components would have been correctly determined. Therefore, two errors are detected in R(2,3) by finding no determinations for the components of $\underline{u}$. We can look at the geometry of this decoding scheme. Consider R(2,3) in example 6. Suppose no errors occurred during transmission and 1 1 0 0 1 0 1 was the received vector. 1 1 0 0 1 0 1 is the characteristic function of the 2-flat $A = \{(000)^t,(100)^t,(101)^t,(111)^t\} = \{\underline{x}_0,\underline{x}_1,\underline{x}_5,\underline{x}_7\}$. Suppose we want to recover u_{12}. u_{12} is the coefficient for the basis vector $\underline{v}_1\underline{v}_2 = (0\ 0\ 0\ 0\ 0\ 0\ 1\ 1)$ in the encoding process. $\underline{v}_1\underline{v}_2$ is also the characteristic function of the 1-flat $A_1 \cap A_2 = \{(011)^t,(111)^t\}$. In the diagram below, the circled points represent $A_1 \cap A_2$ and the points connected with double lines represent the two 2-flats perpendicular to $A_1 \cap A_2$.



If the characteristic function of the received vector has an odd number of points in common with each 2-flat perpendicular to $A_1 \cap A_2$ then $u_{12} = 1$. On the other hand, if there is an even number of points in common, then $u_{12} = 0$. Here we see that A has one point (000) in common with one 2-flat perpendicular to $A_1 \cap A_2$ and three points (100), (101), and (111) in common with the other 2-flat. Therefore, $u_{12} = 1$.

If one error occurs in transmission, one of the 2-flats perpendicular to $A_1 \cap A_2$ would have an even number of points in common with the characteristic function of the received vector and the other perpendicular 2-flat would have an odd number of points in common. And no determination of u_{12} could be made.

Threshold Decoding

First order R - M codes can also be decoded using threshold decoding. Threshold decoding was developed by J. L. Massey in 1963. Massey's primary interest in threshold decoding was for decoding and developing convolution codes.

He also showed that threshold decoding could be used for linear block codes and first order R - M codes.

Threshold decoding is similar to the Reed decoding scheme because it is a type of majority logic decoder. But the parity checks for threshold decoding are a set "orthogonal" on the components of a noise or error vector. The linear code must initially be in systematic form, so that the set of code words are n-tuples $(t_1,t_2,\ldots,t_n)$; where $t_1,t_2,\ldots,t_k$ are the information symbols and the remaining n - k symbols are the parity symbols. The received vector is of the form $(t_1+e_1,t_2+e_2,\ldots,t_n+e_n)$; where $(e_1,e_2,\ldots,e_n)$ is the noise vector. The goal of threshold decoding is to find the noise vector and then subtract it from the received vector and thus find the information vector $(t_1,t_2,\ldots t_k)$.

The following notation and definitions define threshold decoding for a binary code in systematic form. The set of code words for a binary code in systematic form is a subset of the set of n-tuples $(t_1,t_2,\ldots,t_n)$; where $t_i \epsilon F_2$. The symbols $t_1,t_2,\ldots,t_k$ are the information symbols and the remaining n - k symbols are the parity symbols determined by $t_j = \sum_{i=1}^{k} c_{ij}t_i$; where $j = k+1,k+2,\ldots,n$ and $c_{ij} \epsilon F_2$ is determined by the particular code. After a word $\underline{t} = (t_1,t_2,\ldots,t_n)$ is transmitted a received vector $\underline{r} = (r_1,r_2,\ldots r_n)$ is obtained which may differ from $\underline{t}$ by a noise sequence $\underline{e} = (e_1,e_2,\ldots e_n)$, that is, $\underline{r} = (t_1+e_1,t_2+e_2,\ldots t_n+e_n)$.

Each equation $t_j = \sum_{i=1}^{k} c_{ij}t_i$ can be rewritten as $\sum_{i=1}^{k} c_{ij}t_i - t_j$

= 0. These n - k equations form a parity set for this code. A parity check is a sum formed at the receiver defined by $s_j = \sum_{i=1}^{K} c_{ij}r_i - r_j = \sum_{i=1}^{K} c_{ij}e_i - e_j$; where j = k+1,k+2,...n. We define a composite parity check A_i to be any linear combination of $\{s_j\}$. Thus $A_i = \sum_{j=K+1}^{n} b_{ij}s_j$, where $b_{ij} \in F_2$, can be rewritten as $A_i = \sum_{j=K+1}^{n} a_{ij}e_j$; where $a_{ij} = \sum_{h=K+1}^{n} b_{ih}c_{jh}$ for j = 1,2,...k or $a_{ij} = b_{ij}$ for j = k+1,k+2,...n.

Definition 11: Orthogonal Parity Checks

A set of J $(1 \leq J \leq k)$ composite parity checks $\{A_i\}$ is said to be orthogonal on a component e_m of the noise vector if $a_{im} = 1$ for i = 1,2,...,J and $a_{ij} = 0$ for all, but at most one index i for every fixed $j \neq m$. Equivalently, a set of J composite parity checks is called orthogonal on e_m if e_m is checked by each member of the set, but no other noise component is checked by more than one member of the set.

The next theorem by Massey is used to determine e_m from J orthogonal parity checks.

Theorem 3

Provided there are $\lfloor J/2 \rfloor$ or fewer errors in the corresponding received vector, then e_m is given correctly as that value of F_2 which is assumed by the greatest fraction $\{A_i\}$, where r_i's, c_{ij}'s are known and a_{ij}'s are determined by the orthogonal parity checks, (that is, e_m is taken as that value 0 or 1 that occurs more frequently in the parity checks.). Decoding performed according to this theorem is called threshold decoding.

Note that if J =2t is even, then according to theorem 3 e_m can be correctly determined if t or fewer errors occur in the received vector. If J = 2t + 1 then e_m can again be correctly determined if t or fewer errors occur.

These definitions and theorem 2 will become clearer as we interpret this method for linear codes. We assume (n,k) is a binary linear code in systematic form with the first k symbols identical to the information symbols. The parity symbols $t_{k+1}, t_{k+2}, \ldots, t_n$ are determined from the information symbols $t_1, t_2, \ldots, t_k$ by $t_j = \Sigma\, c_{ij} t_i$; where j = k+1,k+2,...,n and $c_{ji} \in F_2$. In matrix form this becomes

$$\begin{bmatrix} t_{k+1} \\ t_{k+2} \\ \cdot \\ \cdot \\ t_n \end{bmatrix}^t = \begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ \cdot \\ t_k \end{bmatrix}^t \begin{bmatrix} c_{1,k+1} & \cdots & c_{1,n} \\ c_{2,k+1} & \cdots & c_{2,n} \\ \cdot & \cdots & \cdot \\ \cdot & \cdots & \cdot \\ c_{k,k+1} & \cdots & c_{k,n} \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ \cdot \\ t_k \end{bmatrix}^t \mathbf{P} ,$$

where **P** is the k x (n-k) matrix of coefficients c_{ij}, so

$$\begin{bmatrix} t_{k+1} \\ \cdot \\ \cdot \\ t_n \end{bmatrix} = \mathbf{P}^t \begin{bmatrix} t_1 \\ \cdot \\ \cdot \\ t_k \end{bmatrix} .$$

So the parity checks s_j are

$$\begin{bmatrix} s_{k+1} \\ s_{k+2} \\ \cdot \\ \cdot \\ s_n \end{bmatrix} = [\mathbf{P}^t:\mathbf{I}] \begin{bmatrix} e_1 \\ e_2 \\ \cdot \\ \cdot \\ e_n \end{bmatrix} = [\mathbf{P}^t:\mathbf{I}] \begin{bmatrix} r_1 \\ \cdot \\ \cdot \\ \cdot \\ r_n \end{bmatrix} ,$$

where **I** is the identity matrix of size n - k. Then $[\mathbf{P}^t:\mathbf{I}]$ = **H** is the parity check matrix of the code. Also $e_1, \ldots, e_k$ are the information noise digits and $e_{k+1},\ldots,e_n$ are the parity noise digits. The rows of $\mathbf{P}^t$ give the coefficients of the information digits appearing in the parity checks and each parity noise digit is checked by only one parity check.

Generally, in threshold decoding our task is to find a set of a sufficient number of parity checks orthogonal on a given e_j, which reduces to finding a set of rows of $\mathbf{P}^t$ which satisfy the orthogonality definition. In using the decoding scheme for R(1,m) codes, the task changes to finding linear combinations of rows of $[\mathbf{P}^t:\mathbf{I}]$ which give parity checks which are orthogonal on linear combinations of the e_j's.

In example 7 we show how to use threshold decoding to decode R(1,3).

Example 10:

First we need to put R(1,3) in systematic form, the generator matrix $\mathbf{G}$ for R(1,3) is

$$\begin{bmatrix} 1&1&1&1&1&1&1&1\\ 0&1&0&1&0&1&0&1\\ 0&0&1&1&0&0&1&1\\ 0&0&0&0&1&1&1&1 \end{bmatrix}.$$

We add the last three rows to the first row and obtain the following:

$$\begin{bmatrix} 1&0&0&1&0&1&1&0\\ 0&1&0&1&0&1&0&1\\ 0&0&1&1&0&0&1&1\\ 0&0&0&0&1&1&1&1 \end{bmatrix}.$$

Now we rearrange the columns of this matrix to obtain $\mathbf{G}$ in standard form:

$$\mathbf{G} = \begin{bmatrix} 1&0&0&0&:&1&1&1&0\\ 0&1&0&0&:&1&1&0&1\\ 0&0&1&0&:&1&0&1&1\\ 0&0&0&1&:&0&1&1&1 \end{bmatrix} = [\mathbf{I}:\mathbf{P}] .$$

The parity check matrix $\mathbf{H}$ is obtained from $\mathbf{G}$. Thus

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & : & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & : & 0 & 0 & 0 & 1 \end{bmatrix} = [\mathbf{P}^t : \mathbf{I}] \ .$$

The matrix $\mathbf{P}^t$ whose entries are the coefficients c_{ji} is

$$\mathbf{P}^t = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \ .$$

The rows of $\mathbf{P}$ correspond to the parity checks s_5, s_6, s_7, s_8. In fact,

$s_5 = r_1 + r_2 + r_3 - r_5 = e_1 + e_2 + e_3 - e_5$,

$s_6 = r_1 + r_2 + r_4 - r_6 = e_1 + e_2 + e_4 - e_6$,

$s_7 = r_1 + r_3 + r_4 - r_7 = e_1 + e_3 + e_4 - e_7$,

$s_8 = r_2 + r_3 + r_4 - r_8 = e_2 + e_3 + e_4 - e_8$.

The symbol s_j is used to denote both a parity check equation and the value of the parity check equation. For $s_7 + s_8 = r_1 + r_2 - r_7 - r_8 = e_1 + e_2 - e_7 - e_8$, $s_7 + s_8$ is a linear combination of the s_j's, that is, it gives the b_{ij}'s for this parity check, the linear combination of r_j's gives the value of this parity check, and the linear combination of the e_j's gives the a_{ij}'s which determine the orthogonality of this parity check. Hence, $s_7 + s_8$ is orthogonal on $e_1 + e_2$. The set $\{s_5, s_6\}$ is also orthogonal on $e_1 + e_2$ as $s_5 = e_1 + e_2 + e_3 - e_5$ and $s_6 = e_1 + e_2 + e_4 - e_6$. Note that one error in any of r_3 through r_8 effects only one parity check, so by majority vote $e_1 + e_2 = 0$. But if an error occurs in r_1 or r_2, then all parity checks equal 1 and $e_1 + e_2$ is determined to be one.

We can write

$$\left.\begin{matrix}(s_7+s_8)\\ s_5\\ s_6\end{matrix}\right\} = (e_1 + e_2)^* ,$$

where * indicates the majority vote of the three parity checks. Likewise the set $\{(s_6+s_7), s_5, s_8\}$ is orthogonal on $e_2 + e_3$, the set $\{(s_5+s_6), s_7, s_8\}$ is orthogonal on $e_3 + e_4$, and the set $\{(s_5+s_7), s_6, s_8\}$ is orthogonal on $e_2 + e_4$. We have

$$\left.\begin{matrix}(s_6+s_7)\\ s_5\\ s_8\end{matrix}\right\} = (e_2 + e_3)^* ,$$

$$\left.\begin{matrix}(s_5+s_6)\\ s_7\\ s_8\end{matrix}\right\} = (e_3 + e_4)^* ,$$

$$\left.\begin{matrix}(s_5+s_7)\\ s_6\\ s_8\end{matrix}\right\} = (e_2 + e_4)^* .$$

Let $s_5' = s_5 + (e_2 + e_3)^* = e_1 + (e_2 + e_3) + (e_2+e_3)^* - e_5$. Then s_5' is orthogonal on e_1. Likewise $s_6' = s_6 + (e_2 + e_4)^* = e_1 + (e_2 + e_4) + (e_2 + e_4)^* - e_6$ is orthogonal on e_1 and $s_7' = s_7 + (e_3+e_4)^* = e_1 + (e_3 + e_4) + (e_3 + e_4)^* - e_7$ is orthogonal on e_1. The matrix whose rows are determined of s_5', s_6', and s_7 is called a modified parity check matrix and is,

$$\begin{bmatrix} 1\,0\,0\,0 & : & 1 & 0 & 0 & 0 \\ 1\,0\,0\,0 & : & 0 & 1 & 0 & 0 \\ 1\,0\,0\,0 & : & 0 & 0 & 1 & 0 \end{bmatrix} .$$

Hence

$$\left.\begin{matrix} s_5' \\ s_6' \\ s_7' \end{matrix}\right\} = e_1^* \quad .$$

After these estimations have been made we can estimate the rest of the information error vector as follows:

$$e_2^* = e_1^* + (e_1 + e_2)^* ,$$

$$e_3^* = e_1^* + (e_1 + e_2)^* + (e_2 + e_3)^* ,$$

$$e_4^* = e_1^* + (e_1 + e_2)^* + (e_2 + e_3)^* + (e_3 + e_4)^* .$$

Finally we add the estimate of the information error vector $(e_1^*, e_2^*, e_3^*, e_4^*)$ to the received information digits (r_1, r_2, r_3, r_4) to obtain the estimate of the sent information.

Suppose we want to send 1 1 0 1. 1 1 0 1 is encoded as (1 1 0 1) [I:P] = (1 1 0 1 0 1 0 0) = $(r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8)$. Now suppose a bit error has occurred during transmissionin the third component, so that 1 1 1 1 0 1 0 0 is the received vector. Then the parity checks are

$$s_5 = r_1 + r_2 + r_3 - r_5 = 1 + 1 + 1 - 0 = 1 ,$$

$$s_6 = r_1 + r_2 + r_4 - r_6 = 1 + 1 + 1 - 1 = 0 ,$$

$$s_7 = r_1 + r_3 + r_4 - r_7 = 1 + 1 + 1 - 0 = 1 ,$$

$$s_8 = r_2 + r_3 + r_4 + r_8 = 1 + 1 + 1 - 0 = 1 .$$

Therefore,

$$\left.\begin{aligned}(s_7 + s_8) &= 0\\ s_5 &= 1\\ s_6 &= 0\end{aligned}\right\} \rightarrow (e_1 + e_2)^* = 0 \ ,$$

$$\left.\begin{aligned}(s_6 + s_7) &= 1\\ s_5 &= 1\\ s_8 &= 1\end{aligned}\right\} \rightarrow (e_2 + e_3)^* = 1 \ ,$$

$$\left.\begin{aligned}(s_5 + s_6) &= 1\\ s_7 &= 1\\ s_8 &= 1\end{aligned}\right\} \rightarrow (e_3 + e_4)^* = 1 \ ,$$

$$\left.\begin{aligned}(s_5 + s_7) &= 0\\ s_6 &= 0\\ s_8 &= 1\end{aligned}\right\} \rightarrow (e_2 + e_4)^* = 0 \ ,$$

$$\left.\begin{aligned}s_5' &= s_5 + (e_2 + e_3)^* = 1 + 0 = 1\\ s_6' &= s_6 + (e_2 + e_4)^* = 0 + 0 = 0\\ s_7' &= s_7 + (e_3 + e_4)^* = 1 + 1 = 0\end{aligned}\right\} \rightarrow e_1^* = 0 \ ,$$

$e_2^* = e_1^* + (e_1 + e_2)^* = 0 + 0 = 0$,

$e_3^* = e_1^* + (e_1 + e_2)^* + (e_2 + e_3)^* = 0 + 0 + 1 = 1$,

$e_4^* = e_1^* + (e_1 + e_2)^* + (e_2 + e_3)^* + (e_3 + e_4)^* = 0 + 0$
$+ 1 + 1 = 0$.

The estimate of the information error vector is (0 0 1 0). Now we add (0 0 1 0) to the received information digits 1 1 1 1 to obtain 1 1 0 1 which was the original information sent.

Now we consider any first order R - M code. R(1,m) is a $(2^m,m+1)$ linear code with minimum distance 2^{m-1}. The parity check matrix $H = [P^t:I]$ is the generator matrix of R(m-2,m), the dual code of R(1,m). The weight enumerator of R(m-2,m) tells us that all rows of its generator matrix have even weight ≥ 4, in fact, all nonzero code words of R(m-2,m) have weight ≥ 4. Therefore, the $(2^m-m-1) \times (m+1)$ matrix P^t has rows of odd weight ≥ 3. The number of (m + 1) -tuples of odd weight ≥ 3 is $C(m+1,3) + C(m+1,5) + \ldots + C(m+1,m+1) = 2^{m+1}/2 - m - 1 = 2^m - 1$ for m even and $C(m+1,3) + C(m+1,5) + \ldots + C(m+1,m) = 2^{m+1}/2 - m - 1 = 2^m - m - 1$ for m odd. Also, we note that no two rows of P^t are alike, for if so, then R(m-2,m) contains a code word of weight 2. Thus P^t's rows contains all (m + 1) - tuples of odd weight three or greater. For example, if m = 4 we have ten 5-tuples of weight three and one 5-tuple of weight five.

Consider the number of parity checks orthogonal on $e_1 + e_2$ that can be formed. For any row of P^t beginning with "0 1" there is another row beginning with "1 0" that is otherwise the same. The sum of these rows forms a parity check orthogonal on $e_1 + e_2$. Also, for any row beginning with "1 1" of weight five or greater there is a row beginning with "0 0" that is otherwise the same. The sum of these rows forms a parity check orthogonal on $e_1 + e_2$. Finally, the rows beginning with "1 1" having weight three have a "1" in another position, therefore they form a set orthogonal on $e_1 + e_2$ and another distinct position in P^t. Thus we can form as many parity checks orthogonal on $e_1 + e_2$ as there are rows of P^t beginning with 1. From the weight enumerator of R(1,m), we see that the nonzero code words not equal $\underline{I}$ of R(1,m) have weight 2^{m-1}. Therefore, the weight of a row of [I:P] is 2^{m-1} and the weight of a row of P is $2^{m-1} - 1$. Hence a column of P^t has $2^{m-1} - 1$ ones. So there are $2^{m-1} - 1$ parity checks orthogonal on $e_1 + e_2$.

For $m > 2$, R(1,m) is capable of correcting $2^{m-2} - 1$ errors. Since the $2^{m-1} - 1$ parity checks are orthogonal on $e_1 + e_2$, errors in at most $2^{m-2} - 1$ of the noise digits e_3 through e_n can effect at most $2^{m-2} - 1$ of these parity checks. Also, an error in e_1 or e_2 will effect every parity check equation. Since for $m > 2$, $(2^m - 1)/2 > 2^{m-2} - 1$, $(e_1 + e_2)^*$ is determined by majority rule and is correct if at most one error occurred in e_1 and e_2. If errors occur in both the noise digits e_1 and e_2, then $(e_1 + e_2)^*$ is zero. The same argument applies for any two e_j's and hence $2^{m-1} - 1$ parity checks orthogonal on any two sums of information noise bits can be formed and $(e_i + e_j)^*$ is estimated. Once $e_1 + e_2, .., e_i + e_j, .. , e_{k-1} + e_k$ are estimated we can form a modified parity check matrix to eliminate variables from the original parity check equations. This modified parity check matrix is obtained by using the original parity check equations given by the rows of $[P^t:I]$

that begin with a one, for which there are $2^{m-1} - 1$ in number. Any sum of an even number of variables $e_2, e_3, \ldots, e_k$ can be formed from the estimated sums of two variables and, since all rows of **P** have odd weight , all parity checks on e_1 check an even number of variables $e_2, \ldots, e_k$, and these can be eliminated using the known sums and a new system of $2^{m-1} - 1$ parity check equations is obtained. Therefore, we can form a modified parity check matrix whose rows are all orthogonal on e_1. Once e_1 is found, $e_2, e_3, \ldots, e_k$ can be found using the estimated sums of two variables (as in example 7). The error correcting capability of this decoding scheme depends on determining the sums $e_i + e_j$, and hence this decoding scheme will correct $2^{m-2} - 1$ errors.

Bounds and Reed - Muller codes

Earlier we discussed three bounds on the linear codes, the Plotkin upper bound, the V - G lower bound and the Hamming upper bound. In this section we will investigate the efficiency of some R - M codes (compared to other R - M codes) using these three bounds. Recall that figure 3 is a graph of these bounds for codes with relatively large minimum distance and length. A point on this graph is $(d_{min}/2n, k/n)$. In order to see where R - M codes fit on this graph, we will first consider R - M codes with large minimum distance and length. Then we will investigate other R - M codes that do not satisfy this condition.

Consider R(r,m), the minimum distance for this code is 2^{m-r}. Therefore, $d_{min}/2n = 2^{m-r}/2^{m+1} = 2^{-(r+1)}$. Now we calculate some values for $d_{min}/2n$, given values for r.

$r = 1 \rightarrow d_{min}/2n = .250$

$r = 2 \rightarrow d_{min}/2n = .125$

$r = 3 \rightarrow d_{min}/2n = .0625$

$r = 4 \rightarrow d_{min}/2n = .03125$

$r = 5 \rightarrow d_{min}/2n = .0156$

We note that as r increases that $d_{min}/2n$ approaches 0 for any value of m. We will only consider these values for r because $d_{min}/2n$ becomes too small for our graph. Now we look at the bounds.

The Plotkin bound can be rearranged to read $k/n \leq 1 - 2(d_{min})/n = 1 - 2(2^{m-r})/2^m = 1 - 2^{1-r}$. So

$r = 1 \rightarrow k/n \leq 1 - 1 = 0$

$r = 2 \rightarrow k/n \leq .5000$

$r = 3 \rightarrow k/n \leq .7500$

$r = 4 \rightarrow k/n \leq .8750$

$r = 5 \rightarrow k/n \leq .9375$

The Hamming bound can be written as $k/n \leq 1 - H((2^{m-r-1} - 1)/2^m) = 1 - H(2^{-(r+1)} - 2^{-m}) \approx 1 - H(2^{-(r+1)})$. So

$r = 1 \rightarrow k/n \leq .189$

$r = 2 \rightarrow k/n \leq .456$

$r = 3 \rightarrow k/n \leq .663$

$r = 4 \rightarrow k/n \leq .799$

$r = 5 \rightarrow k/n \leq .884$

The V - G bound can be written as $k/n \geq 1 - H((d - 2)/(n - 1) = 1 - H((2^{m-r} - 2)/(2^m - 1) \approx 1 - H(2^{-r})$. So

$r = 1 \rightarrow k/n \geq 0$

$r = 2 \rightarrow k/n \geq .189$

$r = 3 \rightarrow k/n \geq .456$

$r = 4 \rightarrow k/n \geq .663$

$r = 5 \rightarrow k/n \geq .799$

Now we will choose some values of m for R(r,m) so that the minimum distance and length are large enough to apply to figure 3. Consider R(1,m), $d_{min} / 2n = .25$, we choose $m \geq 5$.

$m = 5 \rightarrow k/n = .1880$

$m = 6 \rightarrow k/n = .1090$

$m = 7 \rightarrow k/n = .0625$

$m = 8 \rightarrow k/n = .0352$

We note that R(1,5) meets the Hamming bound and is therefore optimum for codes of this minimum distance and length. But as m increases R(1,m) gets further away from the Hamming bound. Also, R(1,m) satisfies the V - G lower bound for all m. Consider R(2,m), $d_{min}/2n = .125$, we choose $m \geq 5$.

$m = 5 \rightarrow k/n = .500$

$m = 6 \rightarrow k/n = .344$

$m = 7 \rightarrow k/n = .227$

$m = 8 \rightarrow k/n = .146$

We see that R(2,5) meets the Plotkin upper bound and therefore has maximum minimum distance for a code of this length. R(2,6) and R(2,7) satisfy the V - G lower bound but are not as efficient as $R(2,5)$ because they do not meet the Plotkin bound. For $m \geq 8$, $R(2,m)$ falls below the V - G lower bound. Consider $R(3,m)$, $d_{min}/2n = .0625$, we choose $m \geq 6$.

$m = 6 \rightarrow k/n = .656$

$m = 7 \rightarrow k/n = .500$

$m = 7 \rightarrow k/n = .363$

The code $R(3,6)$ meets the Hamming upper bound and therefore is optimal, but it falls below the Plotkin upper, so there may exist a code with the same $d_{min}/2n$ with greater rate but with less error correcting capability. For $m \geq 8$, $R(3,m)$ falls below the V - G bound. Consider $R(4,m)$, $d_{min}/2n = .03125$, we choose $m \geq 7$.

$m = 7 \rightarrow k/n = .7734$

$m = 8 \rightarrow k/n = .6367$

The code $R(4,7)$ meets the V - G lower bound, but falls below the Plotkin upper bound and the Hamming upper bound. For $m \geq 8$ $R(4,m)$ falls below the V G lower bound. Consider $R(5,m)$, $d_{min}/2n = .0156$, we choose $m \geq 7$.

$m = 7 \rightarrow k/n = .9375$

$m = 8 \rightarrow k/n = .8550$

$m = 9 \rightarrow k/n = .748$

The code $R(5,7)$ meets the Plotkin bound, and therefore has maximum minimum distance, but it is above the Hamming bound so it is not optimal. For $m \geq 9$ $R(5,m)$ falls below the V - G lower bound.

Figure 4 is a graph of the three bounds with the above R

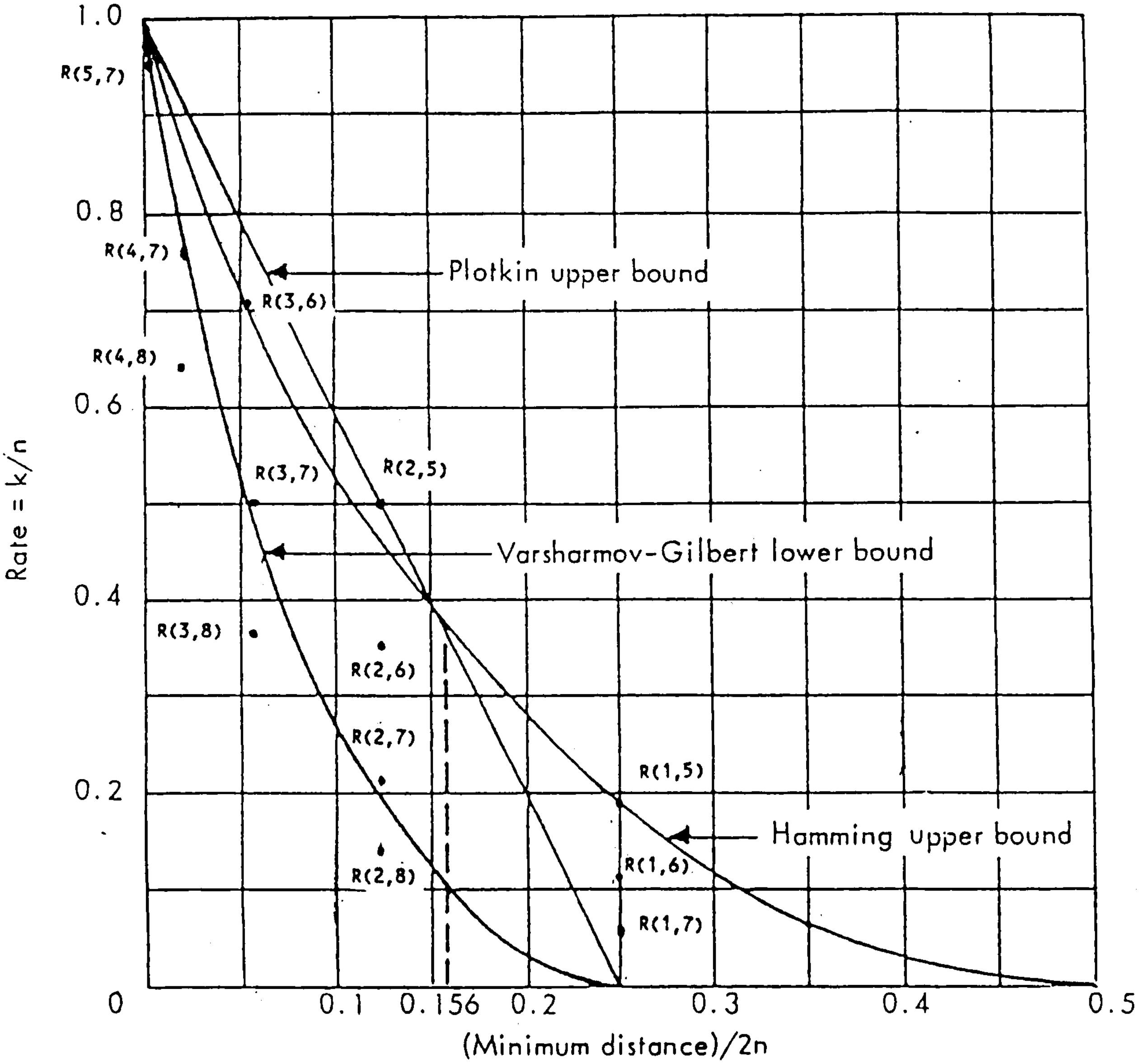


Figure 4. Bounds and Reed - Muller Codes

- M codes plotted.

Now we will discuss some R - M codes that have small minimum distance or length. Consider R(1,3), k/n = .5 and $d_{min}/2n = .25$. K = 4, so according to Hamming bound n - k = 4 ≥ greatest integer value of $[\log_2(1 + C(8,1))] = 4$. Therefore, R(1,3) meets the Hamming bound. Also, k = 4 = 8 - (2(4)-1) + 1 + $\log_2$ 4, so R(1,3) meets the Plotkin bound. Likewise R(2,4) (with $d_{min}/2n = .125$) and R(4,6) (with $d_{min}/2n = .0625$) meet the Hamming bound, but they fall below the Plotkin bound.

If we compare R(1,3) and R(1,5), both of which have $d_{min}/2n = .25$, we see that both codes are optimal,in addition, R(1,3) has max min distance and a higher rate (.5), but R(1,5) has more information symbols and more error correcting capability.

If we compare R(2,4) and R(2,5), both of which have $d_{min}/2n = .125$, we see that R(2,5) is more efficient. Because it has a higher rate (.500 compared with .310), more information symbols, meets the Plotkin bound and nearly meets the Hamming bound.

Further, we see that R(3,6) is the most efficient R - M code with $d_{min}/2n = .0625$; R(4,6) is the most efficient R - M code with $d_{min}/2n = .0312$; and R(5,7) is the most efficient R - M code for $d_{min}/2n = .0156$.

LIST OF REFERENCES

LIST OF REFERENCES

1. Blake, Ian F. and Ronald C. Mullin. The Mathematical Theory of Coding. New York: Academic Press, 1975.

2. Lint, J.H. Van. Introduction to Coding Theory. New York: Springer - Verlag, 1982.

3. Massey, James L. Threshold Decoding. Cambridge: The M.I.T. Press, 1963.

4. M^{c}Eliece, Robert J. The Theory of Information and Coding. Reading: Addison - Wesley Publishing Co., 1977.

5. Peterson, W. Wesley. Error Correcting Codes. New York: The M.I.T. Press and John Wesley and Sons Inc.,

6. Reed, I.S. A Class of Error Multiple Correcting Codes. I.R.E. Trans. Inform. Theory, 1954.

VITA

Michael D. Nestor was born in Pittsburgh, Pennsylvania on June 3, 1960. After graduating from South Allegheny High School in 1978, he served six years in the United States Navy, four of those years aboard the nuclear submarine the U.S.S. Richard B. Russell SSN 687. In August 1987, he received a Bachelor of Science in Mathematics from The Pennsylvania State University.

In August 1987, he accepted a teaching assistantship at The University of Tennessee, Knoxville and began study towards a Master of Science degree. He received this degree with a major in Mathematics.

He is married to Deborah T. Vaughn of Iowa.