Error-Correcting Codes

Code Theory

Radu Trîmbiţaş

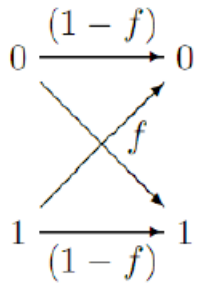
UBB

December 2013

Perfect Communication over Imperfect Channels

Many (if not all) real channels we use to send (store) data are imperfect (noisy).

- modem $\rightarrow$ phone line $\rightarrow$ modem
- Mariner (Mars) $\rightarrow$ radio waves $\rightarrow$ Earth
- comp. memory $\rightarrow$ hard drive $\rightarrow$ comp. memory



Noisy Hard Drive I

- We have an unreliable hard drive.
- Drive stores and reads the bits with $f = 10\%$ error, i.e., on average, every 10^{th} bit is read incorrectly.

- But we want the drive to be reliable with $P(\text{biterror}) \approx 10^{-15}$.
- If we have $P(\text{biterror}) \approx 10^{-15}$, then we can expect 1 wrong bit in 113TB of data. This should be enough to safely read and write 1GB per day for 10 years.
- What can we do to achieve reliable communication or data storage?

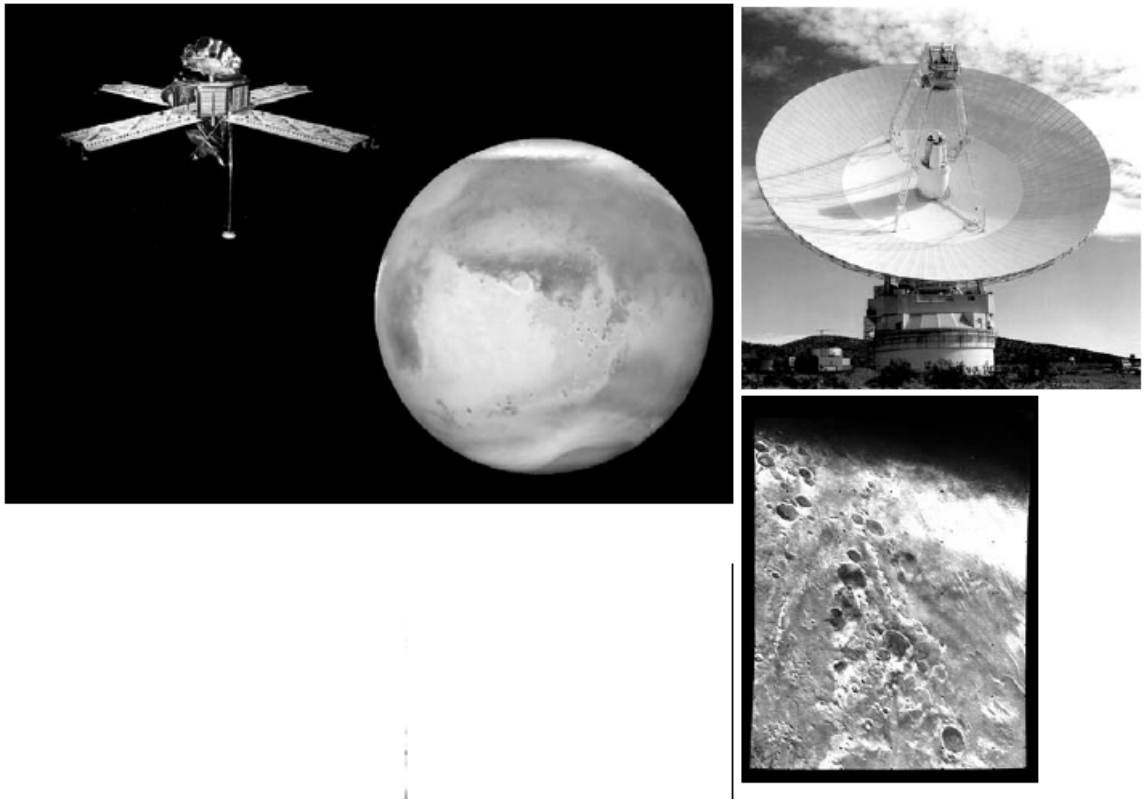
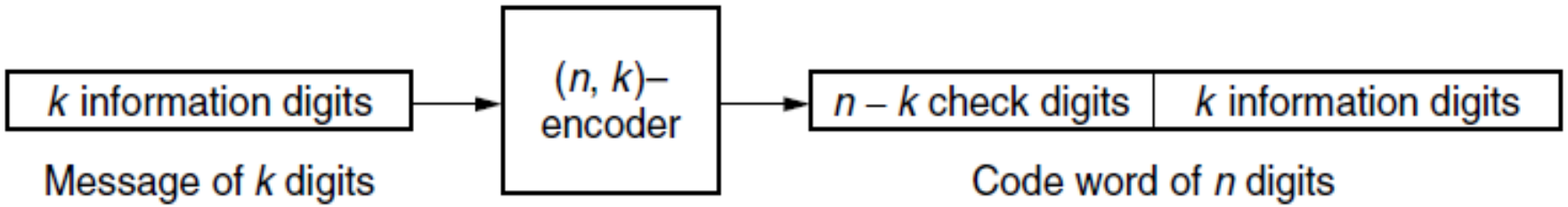


Figure : In 1969 the Mariners 6 and 7 space probes sent back over 200 close-up photographs of Mars. Each photograph was divided into 658,240 pixels and each pixel was given a brightness level ranging from 1 to 28. Therefore, each photograph required about 5 million bits of information. These bits were encoded, using an error-correcting code, and transmitted at a rate of 16,200 bits per second back to Earth, where they were received and decoded into photographs.

The Coding Problem

- We discuss only binary codes. Most of the results generalizes to codes from any finite fields.
- BSC Channels
- To transmit a message over a noisy channel, we break up the message into blocks of k digits and we encode each block by attaching $n-k$ check digits to obtain a code word consisting of n digits. Such a code is referred to as an (n, k)-code



Coding I

- The codewords transmitted and received over a noisy channel can be processed in two ways
  1. *to detect errors* by checking wether or not the received word is a code word. If yes, it is assumed to be the transmitted word. Otherwise, an error must have occured during transmission.
  2. *to correct errors* - the decoder chooses the transmitted code word that is most likely to produce the received word.
- In an (n, k)-code, the original message is k digits long and there are 2^k different possible messages and hence 2^k code words. The received words have n digits; hence there are 2^n possible words that could be received, only 2^k of which are code words.
- The extra $n - k$ check digits that are added to produce the code word are called *redundant digits* because they carry no new information but only allow the existing information to be transmitted more accurately.
- The ratio $R = k/n$ is called the *code rate* or *information rate*.

Coding II

- For each particular communications channel, it is a major problem to design a code that will transmit useful information as fast as possible and, at the same time, as reliably as possible.
- For codes to be efficient, they usually have to be very long; they may contain 2^{100} messages and many times that number of possible received words. To be able to encode and decode such long codes effectively, we look at codes that have a strong algebraic structure.

Simple Codes I

- the (3,2)-code thet attaches a single bit parity check to a message of length 2. The parity check is the sum modulo 2 of the digits in the message (Table 1)
- (3, 1)-code that repeats a message, consisting of a single digit, three times (Table 2).

Message	Code word
00	00
01	101
10	110
11	011
	↑
	parity check

Table : $(3, 2)$ parity check code

Message	Code word
0	000
1	111

Table : $(3, 1)$ repeating code

Simple Codes II

- If one error occurs in the (3, 2) parity check code during transmission, say 101 is changed to 100, then this would be detected because there would be an odd number of 1's in the received word. However, this code will not correct any errors; the received word 100 is just as likely to have come from 110 or 000 as from 101. This code will not detect two errors either. If 101 was the transmitted code word and errors occurred in the first two positions, the received word would be 011, and this would be erroneously decoded as 11.
- The decoder first performs a parity check on the received word. If there are an even number of 1's in the word, the word passes the parity check, and the message is the last two digits of the word. If there are an odd number of 1's in the received work, it fails the parity check, and the decoder registers an error. (Table 3)

Simple Codes III

- The (3, 1) repeating code can be used as an error-detecting code, and it will detect one or two transmission errors but, of course, not three errors. This same code can also be used as an error-correcting code. If the received word contains more 1's than 0's, the decoder assumes that the message is 1; otherwise, it assumes that the message is 0. This will correctly decode messages containing one error, but will erroneously decode messages containing more than one error. (Table 4)

Received word	101	111	100	000	110
Parity Check	Passes	Fails	Fails	Passes	Passes
Received Message	01	Error	Error	00	10

Table : $(3, 2)$ Parity Check code used to detect errors

Simple Codes IV

Received word	111	010	011	000
Decoded Message	1	0	1	0

Table : $(3, 1)$ Repeating Code Used to Correct Errors

Hamming Distance I

Definition 1

The *Hamming distance* between two words u and v of the same length is the number of positions in which they differ.

- Notation: $d(u, v)$. Examples: $d(101, 100) = 1$, $d(101, 010) = 3$, and $d(010, 010) = 0$.
- The Hamming distance between two words is the number of single errors needed to change one word into the other. In an (n, k)-code, the 2^n received words can be thought of as placed at the vertices of an n-dimensional cube with unit sides.
- The Hamming distance between two words is the shortest distance between their corresponding vertices along the edges of the n-cube.

Hamming Distance II

- The 2^k code words form a subset of the 2^n vertices, and the code has better error-correcting /and error-detecting capabilities the farther apart these code words are.
- Figure 2 illustrates the (3,2) parity check code whose code words are at Hamming distance 2 apart.
- Figure 3 illustrates the (3,1) repeating code whose code words are at Hamming distance 3 apart.

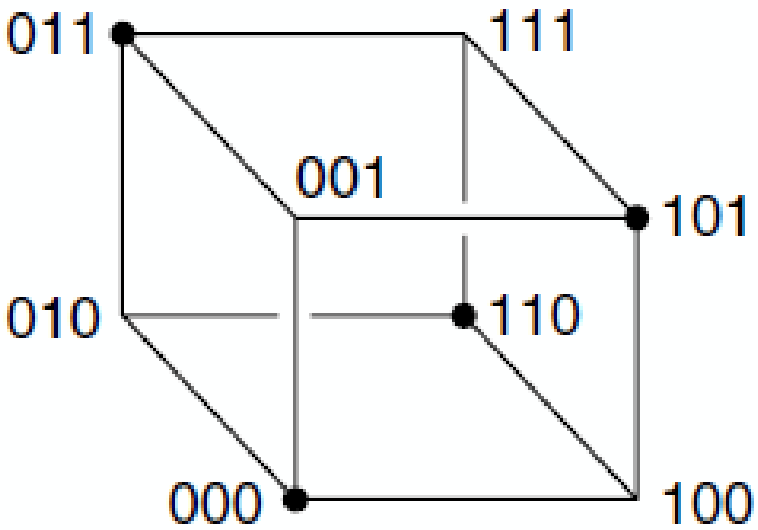


Figure : The code words of the (3,2) parity check code are shown as large dots.

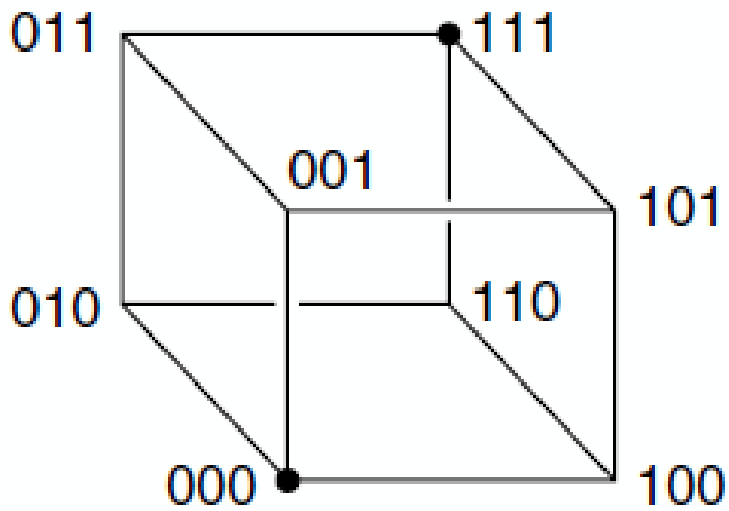


Figure : The code words of the (3,1) repeating code are shown as large dots.

Properties of Hamming distance I

Theorem 2

A code will detect all sets of t or fewer errors if and only if the minimum Hamming distance between code words is at least $t + 1$.

Proof.

If r errors occur when the code word u is transmitted, the received word v is at Hamming distance r from u. These transmission errors will be detected if and only if v is not another code word. Hence all sets of t or fewer errors in the code word u will be detected if and only if the Hamming distance of u from all the other code words is at least $t + 1$. □

Properties of Hamming distance II

Theorem 3

A code is capable of correcting all sets of t or fewer errors if and only if the minimum Hamming distance between code words is at least $2t + 1$.

Proof.

Suppose that the code contains two code words u_1 and u_2 at Hamming distance $2t$ or closer. Then there exists a received word v that differs from u_1 and u_2 in t or fewer positions. This received word v could have originated from u_1 or u_2 with t or fewer errors and hence would not be correctly decoded in both these situations.

Conversely, any code whose code words are at least $2t + 1$ apart is capable of correcting up to t errors. This can be achieved in decoding by choosing the code word that is closest to each received word. □

Properties of Hamming distance III

Conclusion: a (n, k)-code with minimum distance between code words $= d$ can detect $d - 1$ errors and correct at most $(d - 1)/2$ errors. The rate is k/n.

Polynomial Representation I

- The word $a_0 a_1 \dots a_{n-1}$ can be represented by the polynomial

$$a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \in \mathbb{Z}_2[x].$$

- We use this representation to show how codes can be constructed.

Definition 4

Let $p(x) \in \mathbb{Z}_2[x]$ a polynomial of degree $n-k$. The *polynomial code generated by* $p(x)$ is an (n,k)-code whose code words are polynomials, of degree $\leq n$, which are divisible by $p(x)$.

- A message of length k is represented by a polynomial $m(x)$, of degree $\leq k$. In order that the higher-order coefficients in a code polynomial carry the message digits, we multiply $m(x)$ by x^{n-k}. This has the effect of shifting the message $n-k$ places to the right.

Polynomial Representation II

- To encode the message polynomial $m(x)$, we divide $x^{n-k}m(x)$ by $p(x)$ and add the remainder, $r(x)$, to $x^{n-k}m(x)$ to form the code polynomial
$$v(x) = r(x) + x^{n-k}m(x).$$
- This code polynomial is always a multiple of $p(x)$ because, by the division algorithm,

  $x^{n-k}m(x) = q(x)\cdot p(x) + r(x)$ where $\deg r(x) < n-k$ or $r(x) = 0$;

  thus

  $$v(x) = r(x) + x^{n-k}m(x) = -r(x) + x^{n-k}m(x) = q(x)\cdot p(x).$$

  (In $\mathbb{Z}_2[x]$ $r(x) = -r(x)$.)

Polynomial Representation III

- The polynomial $x^{n-k}m(x)$ has zeros in the $n-k$ lowest-order terms, whereas the polynomial $r(x)$ is of degree less than $n-k$; hence the k highest-order coefficients of the code polynomial $v(x)$ are the message digits, and the $n-k$ lowest-order coefficients are the check digits. These check digits are precisely the coefficients of the remainder $r(x)$.
- For example, let $p(x) = 1 + x^2 + x^3 + x^4$ be the generator polynomial of a $(7, 3)$-code. We encode the message 101 as follows:

$$
\begin{array}{rcccccccccccc}
\textit{message} & = & 1 & & 0 & & 1 & & & & & & \\
m(x) & = & 1 & & & & +x^2 & & & & & & \\
x^4 m(x) & = & & & & & & & x^4 & & & & +x^6 \\
r(x) & = & 1 & & +x & & & & & & & & \\
v(x) = r(x) + x^4 m(x) & = & 1 & & +x & & & & +x^4 & & & & +x^6 \\
\textit{code word} & = & 1 & & 1 & & 0 & & 0 & & 1 & 0 & 1
\end{array}
$$

Polynomial Representation IV

- The generator polynomial $p(x) = a_0 + a_1x + \cdots + a_{n-k}x^{n-k}$ is always chosen so that $a_0 = 1$ and $a_{n-k} = 1$, since this avoids wasting check digits. If $a_0 = 0$, any code polynomial would be divisible by x and the first digit of the code word would always be 0; if $a_{n-k} = 0$, the coefficient of x^{n-k-1} in the code polynomial would always be 0.

Examples I

Example 5

Write down all the codewords for the code generated by the polynomial $p(x) = 1 + x + x^3$ when the message length is $k = 3$.

Solution.

$\deg p(x) = 3 \Longrightarrow 3$ check digits and message length $n = 6$. The number of messages is $2^k = 8$. Consider the message 110, which is represented by polynomial $m(x) = 1 + x$. The check digits are obtained by dividing $x^3 m(x) = x^3 + x^4$ by $p(x)$. The checkdigits are the coefficients of the remainder $r(x) = 1 + x^2$. The codeword is $v(x) = r(x) + x^3 m(x) = 1 + x^2 + x^3 + x^4$, and the codeword is 101110. Table 5 shows all the codewords. □

Examples II

- A received message can be checked for errors by testing whether it is divisible by the generator polynomial $p(x)$.
- If the remainder is nonzero when the received polynomial $u(x)$ is divided by $p(x)$, an error must have occurred during transmission.
- If the remainder is zero, the received polynomial $u(x)$ is a code word, and either no error has occurred or an undetectable error has occurred.

Example 6

If the generator polynomial is $p(x) = 1 + x + x^3$, test whether the following received words contain detectable errors: (i) 100011, (ii) 100110, (iii) 101000.

Examples III

Solution.

The received polynomials are $1 + x^4 + x^5$, $1 + x^3 + x^4$, and $1 + x^2$, respectively. These contain detectable errors if and only if they have nonzero remainders when divided by $p(x) = 1 + x + x^3$. Hence $1 + x^4 + x^5$ is divisible by $p(x)$, but $1 + x^3 + x^4$ and $1 + x^2$ are not. Therefore, errors have occurred in the latter two words but are unlikely to have occurred in the first. □

- Table 5 lists all the codewords.
- Hence, in Example 14.4 we can tell at a glance whether a word is a code word simply by noting whether it is on this list.
- However, in practice, the list of code words is usually so large that it is easier to calculate the remainder when the received polynomial is divided by the generator polynomial.

Message			Code Word: Check Digits			Code Word: Message Digits		
0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0
0	1	0	0	1	1	0	1	0
0	0	1	1	1	1	0	0	1
1	1	0	1	0	1	1	1	0
1	0	1	0	0	1	1	0	1
0	1	1	1	0	0	0	1	1
1	1	1	0	1	0	1	1	1
↑	↑	↑	↑	↑	↑	↑	↑	↑
1	x	x^2	1	x	x^2	x^3	x^4	x^5

Table : $(6, 3)$-code generated by $1 + x + x^3$

Shift registers I

- The remainder can easily be computed using shift registers. Figure 4 shows a shift register for dividing by $1 + x + x^3$. The square boxes represent unit delays, and the circle with a cross inside denotes a modulo 2 adder (or exclusive OR gate).
- The delays are initially zero, and a polynomial $u(x)$ is fed into this shift register with the high-order coefficients first. When all the coefficients of $u(x)$ have been fed in, the delays contain the remainder of $u(x)$ when divided by $1 + x + x^3$. If these are all zero, the polynomial $u(x)$ is a code word; otherwise, a detectable error has occurred. Table 14.7 illustrates this shift register in operation.

Shift registers II

- The register in Figure 4 could be modified to encode messages, because the check digits for $m(x)$ are the coefficients of the remainder when $x^3 m(x)$ is divided by $1 + x + x^3$. However, the circuit in Figure 5 is more efficient for encoding. Here the message $m(x)$ is fed simultaneously to the shift register and the output. While $m(x)$ is being fed in, the switch is in position 1 and the remainder is calculated by the register. Then the switch is changed to position 2, and the check digits are let out to immediately follow the message.
- This encoding circuit could also be used for error detection. When $u(x)$ is fed into the encoding circuit with the switch in position 1, the register calculates the remainder of $x^3 u(x)$ when divided by $p(x)$. However, $u(x)$ is divisible by $p(x)$ if and only if $x^3 u(x)$ is divisible by $p(x)$, assuming that $p(x)$ does not contain a factor x.

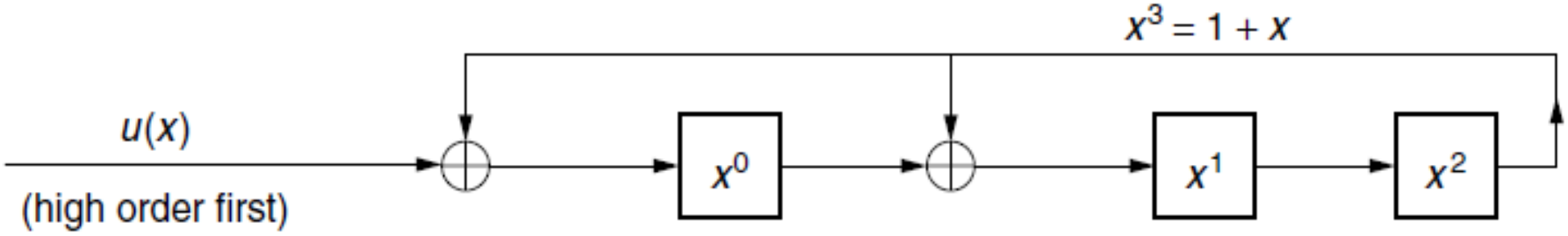


Figure : Shift register for dividing by $1 + x + x^3$

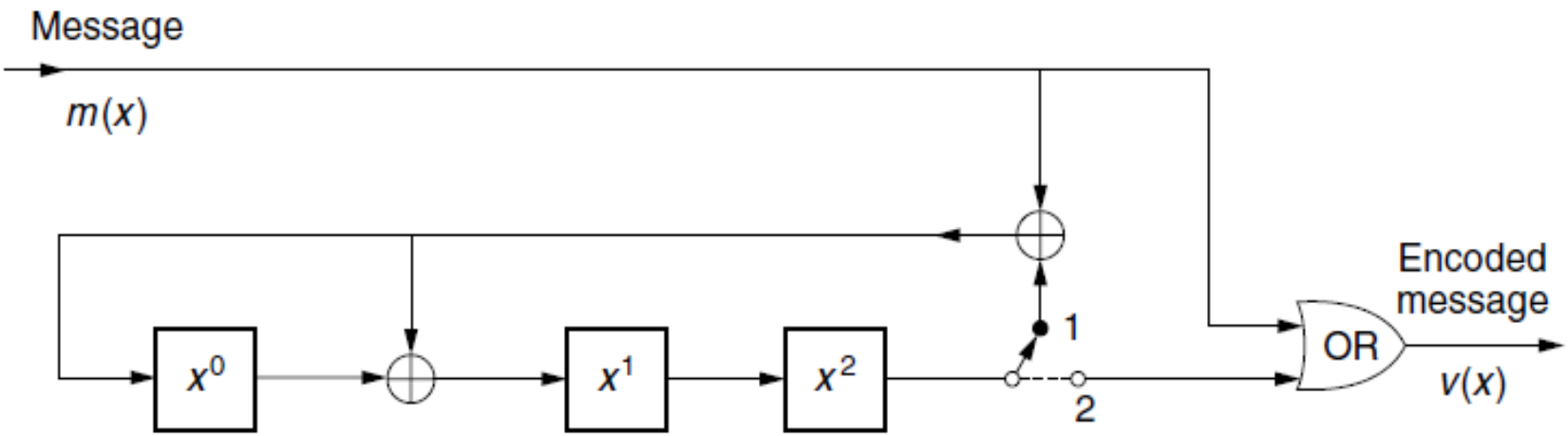


Figure : Encoding circuit for a code generated by $1 + x + x^3$

Stage	Received Polynomial Waiting to Enter Register						Register contents			
							x^0	x^1	x^2	
0	1	0	0	1	1	0	0	0	0	←register initially 0
1		1	0	0	1	1	0	0	0	
2			1	0	0	1	1	0	0	
3				1	0	0	1	1	0	
4					1	0	0	1	1	
5						1	1	1	1	
6							0	0	1	←remainder x^2

Table : Contents of the Shift Register when $1 + x^3 + x^4$ is divided by $1 + x + x^3$

Primitive polynomials and codes I

- How is the generator polynomial chosen so that the code has useful properties without adding too many check digits?
- We now give some examples.

Proposition 7

The polynomial $p(x) = 1 + x$ generates the $(n, n-1)$ parity check code.

Proof.

A polynomial in $\mathbb{Z}_2[x]$ is divisible by $1 + x$ if and only if it contains an even number of nonzero coefficients. Hence the code words of a code generated by $1 + x$ are those words containing an even number of 1's. The check digit for the message polynomial $m(x)$ is the remainder when $xm(x)$ is divided by $1 + x$. Therefore, by the remainder theorem, the check digit is m(1), the parity of the number of 1's in the message. This code is the parity check code. □

Primitive polynomials and codes II

- The $(3, 1)$ code that repeats the single message digit three times has code words 000 and 111, and is generated by the polynomial $1 + x + x^2$.
- We now give one method, using primitive polynomials, of finding a generator for a code that will always detect single, double, or triple errors.
- Furthermore, the degree of the generator polynomial will be as small as possible so that the check digits are reduced to a minimum. Recall that an irreducible polynomial $p(x)$ of degree m over $\mathbb{Z}_2$ is *primitive* if $p(x)|(1 + x^k)$ for $k = 2^m - 1$ and for no smaller k.

Primitive polynomials and codes I

Theorem 8

If $p(x)$ is a primitive polynomial of degree m, then the $(n, n-m)$-code generated by $p(x)$ detects all single and double errors whenever $n \leq 2^m - 1$.

Primitive polynomials and codes II

Proof.

Let $v(x)$ be a transmitted code word and u(x) = v(x) + $e(x)$ be the received word. $e(x)$ – error polynomial. An error is detectable if and only if $p(x) \nmid u(x)$. Since $p(x)|v(x)$, an error $e(x)$ will be detectable $\Longleftrightarrow$ $p(x) \nmid e(x)$.

If a single error occurs, the error polynomial contains a single term, say x^i, where $0 \leq i < n$. Since $p(x)$ is irreducible, it does not have 0 as a root; therefore, $p(x) \nmid x^i$, and the error x^i is detectable.

If a double error occurs, the error polynomial $e(x)$ is of the form $x^i + x^j$, where $0 \geq i < j < n$. Hence $e(x) = x^i(1 + x^{j-i})$, where $0 < j - i < n$. Now $p(x) \nmid x^i$, and since $p(x)$ is primitive, $p(x) \nmid (1 + x^{j-i})$ if $j - i < 2^m - 1$. Since $p(x)$ is irreducible, $p(x) \nmid x^i(1 + x^{j-i})$ whenever $n \leq 2^m - 1$, and all double errors are detectable. □

Primitive polynomials and codes III

Corollary 9

If $p_1(x)$ is a primitive polynomial of degree m, the $(n, n-m-1)$-code generated by $p(x) = (1+x)p_1(x)$ detects all double errors and any odd number of errors whenever $n \leq 2^m - 1$.

Proof.

The code words in the code generated by $p(x)$ must be divisible by $p_1(x)$ and by $(1+x)$. The factor $(1+x)$ has the effect of adding an overall parity check digit to the code. All the code words have an even number of terms, and the code will detect any odd number of errors. Since the code words are divisible by the primitive polynomial $p_1(x)$, the code will detect all double errors if $n \leq 2^m - 1$. □

Primitive polynomials and codes IV

- Some primitive polynomials of low degree are given in Table 7. For example, by adding 11 check digits to a message of length 1012 or less, using the generator polynomial $(1+x)(1+x^3+x^{10}) = 1+x+x^3+x^4+x^{10}+x^{11}$, we can detect single, double, triple, and any odd number of errors.
- Furthermore, the encoding and detecting can be done by a small shift register using only 11 delay units. The number of different messages of length 1012 is 2^{1012}, an enormous figure!
- When written out in base 10, it would contain 305 digits.

Primitive Polynomial	Degree m	$2^m - 1$
$1 + x$	1	1
$1 + x + x^2$	2	3
$1 + x + x^3$	3	7
$1 + x + x^4$	4	15
$1 + x^2 + x^5$	5	31
$1 + x + x^6$	6	63
$1 + x^3 + x^7$	7	127
$1 + x^2 + x^3 + x^4 + x^8$	8	255
$1 + x^4 + x^9$	9	511
$1 + x^3 + x^{10}$	10	1023

Table : Short Table of Primitive Polynomials in $\mathbb{Z}_2[x]$

Matrix Representation I

- Another natural way to represent a word $a_1 a_2 \dots a_n$ of length n is by the element $(a_1, a_2, \dots, a_n)^T$ of the vector space $\mathbb{Z}_2^n = \mathbb{Z}_2 \times \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$ of dimension n over $\mathbb{Z}_2$. We denote the elements of our vector spaces as column vectors, and $(a_1, a_2, \dots, a_n)^T$ denotes the transpose of $(a_1, a_2, \dots, a_n)$. In an (n, k)-code, the 2^k possible messages of length k are all the elements of the vector space $\mathbb{Z}_2^k$, whereas the 2^n possible received words of length n form the vector space $\mathbb{Z}_2^n$.
- An *encoder* is an injective function

$$\gamma : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$$

that that assigns to each k-digit message an n-digit code word.

Matrix Representation II

- An (n, k)-code is called a *linear code* if the encoding function is a linear transformation from $\mathbb{Z}_2^k$ to $\mathbb{Z}_2^n$. Nearly all block codes in use are linear codes, and in particular, all polynomial codes are linear.

Proposition 10

Let $p(x)$ be a polynomial of degree $n - k$ that generates an (n, k)-code. Then this code is linear.

Matrix Representation III

Proof.

Let $\gamma : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$ be the encoding function defined by $p(x)$. Let $m_1(x)$ and $m_2(x)$ be two message polynomials of degree less than k and let $\mathbf{m}_1$ and $\mathbf{m}_2$ be the same messages considered as vectors in $\mathbb{Z}_2^k$. The code vector $\gamma(\mathbf{m}_i)$ corresponds to the code polynomial $v_i(x) = r_i(x) + x^{n-k} m_i(x)$, where $r_i(x)$ is the remainder when $x^{n-k} m_i(x)$ is divided by $p(x)$. Now

$$v_1(x) + v_2(x) = r_1(x) + r_2(x) + x^{n-k}[m_1(x) + m_2(x)],$$

and $r1(x) + r2(x)$ has degree less than $n - k$; therefore, $r_1(x) + r_2(x)$ is the remainder when $x^{n-k} m_1(x) + x^{n-k} m_2(x)$ is divided by $p(x)$. Hence $v_1(x) + v_2(x)$ corresponds to the code vector $\gamma(\mathbf{m}_1 + \mathbf{m}_2)$ and

$$\gamma(\mathbf{m}_1 + \mathbf{m}_2) = \gamma(\mathbf{m}_1) + \gamma(\mathbf{m}_2).$$

Since the only scalars are 0 and 1, this implies that γ is a linear. □

Generator Matrix I

- Let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ be the standard basis of the vector space $\mathbb{Z}_2^n$, that is, $\mathbf{e}_i$ contains a 1 in the ith position and 0's elsewhere. Let G be the $n \times k$ matrix that represents, with respect to the standard basis, the transformation $\gamma : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$, defined by an (n, k) linear code. This matrix G is called the *generator matrix* or *encoding matrix* of the code.
- If $\mathbf{m}$ is a message vector, its code word is $v = G\mathbf{m}$. The code vectors are the vectors in the image of γ, and they form a vector subspace of $\mathbb{Z}_2^n$ of dimension k.
- The columns of G are a basis for this subspace, and therefore, a vector is a code vector if and only if it is a linear combination of the columns of the generator matrix G.

Generator Matrix II

- Most coding theorists write the elements of their vector spaces as row vectors instead of column vectors, as used here. In this case, their generator matrix is the transpose of ours, and it operates on the right of the message vector.
- In the (3,2) parity check code, a vector $\mathbf{m} = (m_1, m_2)^T$ is encoded as $v = (c, m_1, m_2)^T$, where the parity check $c = m_1 + m_2$. Hence the generator matrix is

$$G = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ because } \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} c \\ m_1 \\ m_2 \end{bmatrix}.$$

If the code word is to contain the message digits in its last k positions, the generator matrix must be of the form $G = \begin{bmatrix} P \\ I_k \end{bmatrix}$, where P is an $(n-k) \times k$ matrix and I_k is the $k \times k$ identity matrix.

Generator Matrix III

Example 11

Find the generator matrix for the $(6, 3)$-code of Example 5 that is generated by the polynomial $1 + x + x^3$.

Generator Matrix IV

Solution.

The columns of the generator matrix G are the code vectors corresponding to messages consisting of basis elements $\mathbf{e}_1 = (1,0,0)^T$, $\mathbf{e}_2 = (0,1,0)^T$, and $\mathbf{e}_3 = (0,0,1)^T$. We see from Table 5 that the generator matrix is

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

Encoding the message vector $\mathbf{m}$: compute $G\mathbf{m}$

Properties of a Generator Matrix

Theorem 12

Let $\gamma : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$ *be the encoding function for a linear* (n, k)*- code with generator matrix* $G = \begin{bmatrix} P \\ I_k \end{bmatrix}$*, where where* P *is an* $(n-k) \times k$ *matrix and* I_k *is the* $k \times k$ *identity matrix. Then the linear transformation*

$$\eta : \mathbb{Z}_2^n \to \mathbb{Z}_2^{n-k}$$

defined by the $(n-k) \times n$ *matrix* $H = (I_{n-k}|P)$ *has the following properties:*

(i) $\mathrm{Ker}\eta = \mathrm{Im}\,\gamma$.

(ii) *A received vector* $\mathbf{u}$ *is a code vector if and only if* $H\mathbf{u} = 0$.

Proof I

(i) $\eta \circ \gamma : \mathbb{Z}_2^k \to \mathbb{Z}_2^{n-k}$ is the zero transformation because

$$HG = [I_{n-k}|P] \begin{bmatrix} P \\ I_k \end{bmatrix} = I_{n-k}P + PI_k = P + P = 0$$

Hence $\operatorname{Im}\gamma \subseteq \operatorname{Ker}\eta$. Since the first $n-k$ columns of H consist of the standard basis vectors in $\mathbb{Z}_2^{n-k}$, $\operatorname{Im}\eta$ spans $\mathbb{Z}_2^{n-k}$ and contains 2^{n-k} elements. By the morphism theorem for groups,

$$|\operatorname{Ker}\eta| = \frac{|\mathbb{Z}_2^n|}{|\operatorname{Im}\eta|} = \frac{2^n}{2^{n-k}} = 2^k.$$

But $|\operatorname{Im}\gamma| = 2^k$, and $\operatorname{Im}\gamma = \operatorname{Ker}\eta$.

(ii) The code vectors form a subspace, $\operatorname{Im}\gamma$, of dimension k in $\mathbb{Z}_2^n$, generated by the columns of G. We now find a linear transformation $\eta : \mathbb{Z}_2^n \to \mathbb{Z}_2^{n-k}$ represented by a matrix H, whose kernel is precisely $\operatorname{Im}\gamma$. Hence a vector $\mathbf{u}$ will be a code vector if and only if $H\mathbf{u} = 0$.

Proof II

The $(n-k) \times n$ matrix H in Theorem 12 is called the *parity check matrix* of the (n, k)-code.

Examples I

- The parity check matrix of the $(3, 2)$ parity check code is the 1×3 matrix $H = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$. A received vector $\mathbf{u} = (u_1, u_2, u_3)^T$ is a code vector if and only if

$$H\mathbf{u} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = u_1 + u_2 + u_3 = 0.$$

- The parity check matrix of the $(3, 1)$-code that repeats the message three times is the 2×3 matrix $H = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$. A received vector $\mathbf{u} = (u_1, u_2, u_3)^T$ is a code vector if and only if $H\mathbf{u} = 0$, that is, if and only if $u_1 + u_3 = 0$ and $u_2 + u_3 = 0$. In $\mathbb{Z}_2$, this is equivalent to $u_1 = u_2 = u_3$.

Examples II

- The parity check matrix for the $(6, 3)$-code of Examples 5 and 11 is

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

  - A received vector $\mathbf{u} = (u_1, \ldots, u_6)^T$ is a code vector if and only if

$$\begin{aligned} u_1 \qquad\quad + u_4 \qquad\quad + u_6 &= 0 \\ u_2 \qquad\quad + u_4 \quad + u_5 \quad + u_6 &= 0 \\ u_3 \qquad\quad + u_5 \quad + u_6 &= 0 \end{aligned}$$

  - That is, if and only if

$$\begin{aligned} u_1 &= u_4 + u_6 \\ u_2 &= u_4 + u_5 + u_6 \\ u_3 &= u_5 + u_6 \end{aligned}$$

Examples III

- In this code, the three digits on the right, u_4, u_5, and u_6, are the message digits, whereas u_1, u_2, and u_3 are the check digits.

- For each code vector $\mathbf{u}$, the equation $H\mathbf{u} = 0$ expresses each check digit in terms of the message digits. This is why H is called the parity check matrix.

Example 13

Find the generator matrix and parity check matrix for the $(9, 4)$-code generated by $p(x) = (1 + x)(1 + x + x^4) = 1 + x^2 + x^4 + x^5$. Then use the parity check matrix to determine whether the word 110110111 is a code word.

Examples IV

- *Solution*. The check digits attached to a message polynomial $m(x)$ are the coefficients of the remainder when $x^5 m(x)$ is divided by $p(x)$. The message polynomials are linear combinations of 1, x, x^2, and x^3. We can calculate the remainders when x^5, x^6, x^7, and x^8 are divided by $p(x)$ as follows. [This is just like the action of a shift register that divides by p(x).]

$$x^5 \equiv 1 + x^2 + x^4 \bmod p(x)$$
$$x^6 \equiv x + x^3 + x^5 \equiv 1 + x + x^2 + x^3 + x^4 \bmod p(x)$$
$$x^7 \equiv x + x^2 + x^3 + x^4 + x^5 \equiv 1 + x + x^3 \bmod p(x)$$
$$x^8 \equiv x + x^2 + x^4 \bmod p(x)$$

Examples V

- Therefore, every code polynomial is a linear combination of the following basis polynomials:

$$1 + x^2 + x^4 + x^5$$
$$1 + x + x^2 + x^3 + x^4 + x^6$$
$$1 + x + x^3 + x^7$$
$$x + x^2 + x^4 + x^8.$$

- The generator matrix G is obtained from the coefficients of the polynomials above, and the parity check matrix H is obtained from G. Recall that

$$G = \begin{bmatrix} P \\ I_k \end{bmatrix}, \; H = \begin{bmatrix} I_{n-k} & P \end{bmatrix}.$$

Examples VI

Hence:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

- If the received vector is $\mathbf{u} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}^T$, $H\mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \end{bmatrix}^T$ and hence $\mathbf{u}$ is not a code vector.□

Examples VII

- **Conclusion**. Summing up, if $G = \begin{bmatrix} P \\ I_k \end{bmatrix}$ is the generator matrix of an (n, k)-code, then $H = (I_{n-k}|P)$ is the parity check matrix. We encode a message $\mathbf{m}$ by calculating $G\mathbf{m}$, and we can detect errors in a received vector $\mathbf{u}$ by calculating $H\mathbf{u}$. A linear code is determined by either giving its generator matrix or by giving its parity check matrix.

Error Correcting and Decoding I

- We would like to find an efficient method for correcting errors and decoding. One crude method would be to calculate the Hamming distance between a received word and each code word. The code word closest to the received word would be assumed to be the most likely transmitted word. However, the magnitude of this task becomes enormous as soon as the message length is quite large.
- Consider an (n, k) linear code with encoding function $\gamma : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$. Let $V = \operatorname{Im} \gamma$ be the subspace of code vectors.
- If the code vector $\mathbf{v} \in V$ is sent through a channel and an error $\mathbf{e} \in \mathbb{Z}_2^n$ occurs during transmission, the received vector will be $\mathbf{u} = \mathbf{v} + \mathbf{e}$. The decoder receives the vector $\mathbf{u}$ and has to determine the most likely transmitted code vector $\mathbf{v}$ by finding the most likely error pattern $\mathbf{e}$. This error is $\mathbf{e} = -\mathbf{v} + \mathbf{u} = \mathbf{v} + \mathbf{u}$.
- The decoder does not know what the code vector $\mathbf{v}$ is, but knows that the error $\mathbf{e}$ lies in the coset $V + \mathbf{u}$.

Error Correcting and Decoding II

- The most likely error pattern in each coset of $\mathbb{Z}_2^n$ by V is called the *coset leader*.
- The coset leader will usually be the element of the coset containing the smallest number of 1's. If two or more error patterns are equally likely, one is chosen arbitrarily. In many transmission channels, errors such as those caused by a stroke of lightning tend to come in bursts that affect several adjacent digits. In these cases, the coset leaders are chosen so that the 1's in each error pattern are bunched together as much as possible.
- The cosets of $\mathbb{Z}_2^n$ by the subspace V can be characterized by means of the parity check matrix H. The subspace V is the kernel of the transformation $\eta : \mathbb{Z}_2^n \to \mathbb{Z}_2^{n-k}$; therefore, by the morphism theorem, the set of cosets $\mathbb{Z}_2^n / V$ is isomorphic to $\operatorname{Im} \eta$, where the isomorphism sends the coset $V + \mathbf{u}$ to $\eta(u) = H\mathbf{u}$. Hence the coset $V + \mathbf{u}$ is characterized by the vector $H\mathbf{u}$.

Error Correcting and Decoding III

- If H is an $(n-k) \times n$ parity check matrix and $u \in \mathbb{Z}_2^n$, then the $(n-k)$-dimensional vector $H\mathbf{u}$ is called the syndrome of u. (Syndrome is a medical term meaning a pattern of symptoms that characterizes a condition or disease.)
- Every element of $\mathbb{Z}_2^{n-k}$ is a syndrome; thus there are 2^{n-k} different cosets and 2^{n-k} different syndromes.

Theorem 14

Two vectors are in the same coset of $\mathbb{Z}_2^n$ by V if and only if they have the same syndrome.

Error Correcting and Decoding IV

Proof.

If $u_1, u_2 \in \mathbb{Z}_2^n$, then the following statements are equivalent:

(i) $V + \mathbf{u}_1 = V + \mathbf{u}_2$,

(ii) $\mathbf{u}_1 - \mathbf{u}_2 \in V$,

(iii) $H(\mathbf{u}_1 - \mathbf{u}_2) = 0$,

(iv) $H\mathbf{u}_1 = H\mathbf{u}_2$.

Syndrome Decoding

- **We can decode received words to correct errors** by using the following procedure:
  1. Calculate the syndrome of the received word.
  2. Find the coset leader in the coset corresponding to this syndrome.
  3. Subtract the coset leader from the received word to obtain the most likely transmitted word.
  4. Drop the check digits to obtain the most likely message.
- For a polynomial code generated by $p(x)$, the syndrome of a received polynomial $u(x)$ is the remainder obtained by dividing $u(x)$ by $p(x)$.
- This is because the jth column of H is the remainder obtained by dividing x^{j-1} by $p(x)$.
- Hence the syndrome of elements in a polynomial code can easily be calculated by means of a shift register that divides by the generator polynomial.

Examples I

Example 15

Write out the cosets and syndromes for the $(6, 3)$-code with parity check matrix

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

Solution.

- Each of the rows in Table 8 forms a coset with its corresponding syndrome. The top row is the set of code words.

Examples II

- The element in each coset that is most likely to occur as an error pattern is chosen as coset leader and placed at the front of each row. In the top row 000000 is clearly the most likely error pattern to occur. This means that any received word in this row is assumed to contain no errors.
- In each of the next six rows, there is one element containing precisely one nonzero digit; these are chosen as coset leaders. Any received word in one of these rows is assumed to have one error corresponding to the nonzero digit in its coset leader.
- In the last row, every word contains at least two nonzero digits. We choose 000110 as coset leader.

Examples III

- We could have chosen 101000 or 010001, since these also contain two nonzero digits; however, if the errors occur in bursts, then 000110 is a more likely error pattern. Any received word in this last row must contain at least two errors. In decoding with 000110 as coset leader, we are assuming that the two errors occur in the fourth and fifth digits.
- Each word in Table 8 can be constructed by adding its coset leader to the code word at the top of its column.

Examples IV

Synd rome	Coset Leader				Words			
000	000000	110100	011010	111001	101110	001101	100011	010111
100	100000	010100	111010	011001	001110	101101	000011	110111
010	010000	100100	001010	101001	111110	011101	110011	000111
001	001000	111100	010010	110001	100110	000101	101011	011111
110	000100	110000	011110	111101	101010	001001	100111	010011
011	000010	110110	011000	111011	101100	001111	100001	010101
111	000001	110101	011011	111000	101111	001100	100010	010110
101	000110	110010	011100	111111	101000	001011	100101	010001

Table : Syndromes and All Words of a $(6, 3)$-Code

- A word could be decoded by looking it up in the table and taking the code word at the top of the column in which it appears.

Examples V

- When the code is large, this decoding table is enormous, and it would be impossible to store it in a computer.
- However, in order to decode, all we really need is the parity check matrix to calculate the syndromes, and the coset leaders corresponding to each syndrome.

Example 16

Decode 111001, 011100, 000001, 100011, and 101011 using Table 9, which contains the syndromes and coset leaders. The parity check matrix is

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

Examples VI

Solution.

Table 10 shows the calculation of the syndromes and the decoding of the received words. □

Syndrome	Coset Leader
000	000000
100	100000
010	010000
001	001000
110	000100
011	000010
111	000001
101	000110

Table : Syndromes and Coset Leaders for a $(6, 3)$ Code

Examples VII

Word received $\mathbf{u}$	111001	011100	000001	100011	101011
Syndrome $H\mathbf{u}$	000	101	111	000	001
Coset leader $\mathbf{e}$	000000	000110	000001	000000	001000
Code word $\mathbf{u}+\mathbf{e}$	111001	011010	000000	100011	100011
Message	001	010	000	011	011

Table : Decoding Using Syndromes and Coset Leaders

Examples - Continuation I

Example 17

Calculate the table of coset leaders and syndromes for the $(9, 4)$ polynomial code of Example 13, which is generated by $p(x) = 1 + x^2 + x^4 + x^5$.

Solution.

- There is no simple algorithm for finding all the coset leaders. One method of finding them is as follows.
- We write down, in Table 11, the 25 possible syndromes and try to find their corresponding coset leaders. We start filling in the table by first entering the error patterns, with zero or one errors, next to their syndromes. These will be the most likely errors to occur.

Examples - Continuation II

- The error pattern with one error in the jth position is the jth standard basis vector in $\mathbb{Z}_2^9$ and its syndrome is the jth column of the parity check matrix H, given in Example 13. So, for instance, H(000000001) = 01101, the last column of H.
- The next most likely errors to occur are those with two adjacent errors. We enter all these in the table. For example,

$$\begin{aligned} H(000000011) &= H(000000010) + H(000000001) \\ &= 11010 + 01101, \text{ the last two columns of } H \\ &= 10111. \end{aligned}$$

- This still does not fill the table. We now look at each syndrome without a coset leader and find the simplest way the syndrome can be constructed from the columns of H.
- Most of them come from adding two columns, but some have to be obtained by adding three columns.

Examples - Continuation III

Syndrome	Coset Leader	Syndrome	Coset Leader	Syndrome	Coset Leader
00000	000000000	01011	000011100	10110	000111000
00001	000010000	01100	011000000	10111	000000011
00010	000100000	01101	000000001	11000	110000000
00011	000110000	01110	011100000	11001	110010000
00100	001000000	01111	000001010	11010	000000010
00101	000000110	10000	100000000	11011	000010010
00110	001100000	10001	001001000	11100	111000000
00111	001110000	10010	000000101	11101	000100100
01000	010000000	10011	001101000	11110	000010100
01001	010010000	10100	000011000	11111	000000100
01010	000001100	10101	000001000		

Table : Syndromes and Their Coset Leaders for a (9,4) Code

Examples - Continuation IV

- The $(9, 4)$-code in Example 17 will, by Corollary 9, detect single, double, and triple errors. Hence it will correct any single error.
- It will not detect all errors involving four digits or correct all double errors, because 000000000 and 100001110 are two code words of Hamming distance 4 apart.
- For example, if the received word is 100001000, whose syndrome is 00101, Table 11 would decode this as 100001110 rather than 000000000; both these code words differ from the received word by a double error.

Example 18

Decode 100110010, 100100101, 111101100, and 000111110 using the parity check matrix in Example 13 and the coset leaders in Table 11.

Examples - Continuation V

Solution.

Table 12 illustrates the decoding process. □

Word received $\mathbf{u}$	100110010	100100101	111101100	000111110
Syndrome $H\mathbf{u}$	01000	00000	10111	10011
Coset leader $\mathbf{e}$	010000000	000000000	000000011	001101000
Code word $\mathbf{u}+\mathbf{e}$	110110010	100100101	111101111	001010110
Message	0010	0101	1111	0110

Table : Decoding Using Syndromes and Coset Leaders

BCH Codes I

- The most powerful class of error-correcting codes known to date were discovered around 1960 by Hocquenghem and independently by Bose and Chaudhuri.
- For any positive integers m and t , with $t < 2^{m-1}$, there exists a Bose–Chaudhuri–Hocquenghem (BCH) code of length $n = 2^m - 1$ that will correct any combination of t or fewer errors.
- These codes are polynomial codes with a generator $p(x)$ of degree mt and have message length at least $n - mt$.
- A t-*error-correcting BCH code* of length $n = 2^m - 1$ has a generator polynomial $p(x)$ that is constructed as follows.

BCH Codes II

- Take a primitive element α in the Galois field $GF(2^m)$. Let $p_i(x) \in \mathbb{Z}_2[x]$ be the irreducible polynomial with α_i as a root, and define

$$p(x) = \text{lcm}\left(p_1(x), p_2(x), \ldots, p_{2t}(x)\right).$$

  It is clear that α, α^2, α^3, $\ldots$, α^{2t} are all roots of $p(x)$. It can be shown that $[p_i(x)]^2 = p_i(x^2)$ and hence α^{2i} is a root of $p_i(x)$. Therefore,

$$p(x) = \text{lcm}(p_1(x), p_3(x), ..., p_{2t-1}(x)).$$

- Since $GF(2^m)$ is a vector space of degree m over $\mathbb{Z}_2$, for any $\beta = \alpha^i$, the elements 1, β, β^2, $\ldots$, β^m are linearly dependent. Hence β satisfies a polynomial of degree at most m in $\mathbb{Z}_2[x]$, and the irreducible polynomial $p_i(x)$ must also have degree at most m. Therefore,

$$\deg p(x) \leq \deg p_1(x) \cdot \deg p_3(x) \cdots \deg p_{2t-1}(x) \leq mt.$$

An Example I

Example 19

Find the generator polynomials of the t-error-correcting BCH codes of length $n = 15$ for each value of $t < 8$.

Solution.

- Let α be a primitive element of $GF(16)$, where $\alpha^4 + \alpha + 1 = 0$.
- We repeatedly refer back to the elements of $GF(16)$ given in Table 14 when performing arithmetic operations in $GF(16) = \mathbb{Z}_2(\alpha)$.
- We first calculate the irreducible polynomials $p_i(x)$ that have α_i as roots. We only need to look at the odd powers of α. The element α itself is the root of $x^4 + x + 1$. Therefore, $p_1(x) = x^4 + x + 1$.
- If the polynomial $p_3(x)$ contains α^3 as a root, it also contains

$$(\alpha^3)^2 = \alpha^6, (\alpha^6)^2 = \alpha^{12}, (\alpha^{12})^2 = \alpha^{24} = \alpha^9, (\alpha^9)^2 = \alpha^{18} = \alpha^3.$$

An Example II

- Hence

$$\begin{aligned} p_3(x) &= (x-\alpha^3)(x-\alpha^6)(x-\alpha^{12})(x-\alpha^9) \\ &= \left(x^2+(\alpha^3+\alpha^6)x+\alpha^9\right)\left(x^2+(\alpha^{12}+\alpha^9)x+\alpha^{21}\right) \\ &= x^4+(\alpha^2+\alpha^8)x^3+(\alpha^9+\alpha^{10}+\alpha^6)x^2+(\alpha^{17}+\alpha^8)x+\alpha^{15} \\ &= x^4+x^3+x^2+x+1. \end{aligned}$$

- The polynomial $p_5(x)$ has roots α^5, α^{10}, and $\alpha^{20}=\alpha^5$. Hence

$$\begin{aligned} p_5(x) &= (x-\alpha^5)(x-\alpha^{10}) \\ &= x^2+x+1. \end{aligned}$$

An Example III

- The polynomial $p_7(x)$ has roots α^7, α^{14}, $\alpha^{28} = \alpha^{13}$, $\alpha^{26} = \alpha^{11}$, and $\alpha^{22} = \alpha^7$. Hence

$$\begin{aligned} p_7(x) &= (x - \alpha^7)(x - \alpha^{14})(x - \alpha^{13})(x - \alpha^{11}) \\ &= (x^2 + \alpha x + \alpha^6)(x^2 + \alpha^4 x + \alpha^9) \\ &= x^4 + x^3 + 1. \end{aligned}$$

- Now every power of α is a root of one of the polynomials $p_1(x)$, $p_3(x)$, $p_5(x)$, or $p_7(x)$. For example, $p_9(x)$ contains α^9 as a root, and therefore, $p_9(x) = p_3(x)$.
- The BCH code that corrects one error is generated by $p(x) = p_1(x) = x^4 + x + 1$.

An Example IV

- The BCH code that corrects two errors is generated by

$$p(x) = \text{lcm}(p_1(x), p_3(x)) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1).$$

This least common multiple is the product because $p_1(x)$ and $p_3(x)$ are different irreducible polynomials. Hence $p(x) = x^8 + x^7 + x^6 + x^4 + 1$.

- The BCH code that corrects three errors is generated by

$$\begin{aligned} p(x) &= \text{lcm}(p_1(x), p_3(x), p_5(x)) \\ &= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) \\ &= x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1. \end{aligned}$$

An Example V

- The BCH code that corrects four errors is generated by

$$\begin{aligned} p(x) &= \operatorname{lcm}(p_1(x), p_3(x), p_5(x), p_7(x)) \\ &= p_1(x) \cdot p_3(x) \cdot p_5(x) \cdot p_7(x) \\ &= \frac{x^{15}+1}{x+1} = \sum_{i=0}^{14} x^i. \end{aligned}$$

  This polynomial contains all the elements of $GF(16)$ as roots, except for 0 and 1.

- Since $p_9(x) = p_3(x)$, the five-error-correcting BCH code is generated by

$$\begin{aligned} p(x) &= \operatorname{lcm}(p_1(x), p_3(x), p_5(x), p_7(x), p_9(x)) \\ &= \frac{x^{15}+1}{x+1}. \end{aligned}$$

An Example VI

- This is also the generator of the six- and seven-error-correcting BCH codes.
- These results are summarized in Table 13.□
- For example, the two-error-correcting BCH code is a (15, 7)-code with generator polynomial $x^8 + x^7 + x^6 + x^4 + 1$. It contains seven message digits and eight check digits.
- The seven-error-correcting code generated by $(x^{15} + 1)/(x + 1)$ has message length 1, and the two code words are the sequence of 15 zeros and the sequence of 15 ones. Each received word can be decoded by majority rule to give the message 1, if the word contains more 1's than 0's, and to give the message 0 otherwise. It is clear that this will correct up to seven errors.

t	Roots of $p_{2t-1}(x)$	Degree $p_{2t-1}(x)$	$p(x)$	$\deg p(x) = 15 - k$	Mess. length, k
1	$\alpha, \alpha^2, \alpha^4, \alpha^8$	4	$p_1(x)$	4	11
2	$\alpha^3, \alpha^6, \alpha^{12}, \alpha^9$	4	$p_1(x)p_3(x)$	8	7
3	α^5, α^{10}	2	$p_1(x)p_3(x)p_5(x)$	10	5
4	$\alpha^7, \alpha^{14}, \alpha^{13}, \alpha^{11}$	4	$(x^{15}+1)/(x+1)$	14	1
5	$\alpha^9, \alpha^3, \alpha^6, \alpha^{12}$	4	$(x^{15}+1)/(x+1)$	14	1
6	$\alpha^{11}, \alpha^7, \alpha^{14}, \alpha^{13}$	4	$(x^{15}+1)/(x+1)$	14	1
7	$\alpha^{13}, \alpha^{11}, \alpha^7, \alpha^{14}$	4	$(x^{15}+1)/(x+1)$	14	1

Table : Construction of t-Error-Correcting BCH Codes of Length 15

Element						α^0	α^1	α^2	α^3
0	$=$	0				0	0	0	0
α^0	$=$	1				1	0	0	0
α^1	$=$		α			0	1	0	0
α^2	$=$			α^2		0	0	1	0
α^3	$=$				α^3	0	0	0	1
α^4	$=$	1	$+\alpha$			1	1	0	0
α^5	$=$		α	$+\alpha^2$		0	1	1	0
α^6	$=$			α^2	$+\alpha^3$	0	0	1	1
α^7	$=$	1	$+\alpha$		$+\alpha^3$	1	1	0	1
α^8	$=$	1		$+\alpha^2$		1	0	1	0
α^9	$=$		α		$+\alpha^3$	0	1	0	1
α^{10}	$=$	1	$+\alpha$	$+\alpha^2$		1	1	1	0
α^{11}	$=$		α	$+\alpha^2$	$+\alpha^3$	0	1	1	1
α^{12}	$=$	1	$+\alpha$	$+\alpha^2$	$+\alpha^3$	1	1	1	1
α^{13}	$=$	1		$+\alpha^2$	$+\alpha^3$	1	0	1	1
α^{14}	$=$	1			$+\alpha^3$	1	0	0	1
α^{15}	$=$	1							

Theoretical Results I

We now show that the BCH code given at the beginning of this section does indeed correct t errors.

Lemma 20

The minimum Hamming distance between code words of a linear code is the minimum number of ones in the nonzero code words.

Proof.

If v_1 and v_2 are code words, then, since the code is linear, $v_1 - v_2$ is also a code word. The Hamming distance between v_1 and v_2 is equal to the number of 1's in $v_1 - v_2$. The result now follows because the zero word is always a code word, and its Hamming distance from any other word is the number of 1's in that word. □

Theoretical Results II

Theorem 21

If $t < 2^{m-1}$, the minimum distance between code words in the BCH code given in at the beginning of this section is at least $2t + 1$, and hence this code corrects t or fewer errors.

Theoretical Results III

Proof of Theorem 21.

Suppose that the code contains a code polynomial with fewer than $2t+1$ nonzero terms,

$$v(x) = v_1 x^{r_1} + \cdots + v_{2t} x^{r_{2t}} \text{ where } r_1 < \cdots < r_{2t}.$$

This code polynomial is divisible by the generator polynomial $p(x)$ and hence has roots α, α^2, α^3, $\ldots$, α^{2t} . Therefore, if $1 \leq i \leq 2t$,

$$\begin{aligned} v(\alpha^i) &= v_1 \alpha^{ir_1} + \cdots + v_{2t} \alpha^{ir_{2t}} \\ &= \alpha^{ir_1}(v_1 + \cdots + v_{2t} \alpha^{ir_{2t} - ir_1}). \end{aligned}$$

. . .

Theoretical Results IV

Proof of Theorem 21 - Continuation.

Put $s_i = r_i - r_1$; the elements $v_1, \ldots, v_{2t}$ satisfy the linear system

$$\begin{aligned} v_1 + v_2\alpha^{s_2} + \cdots + v_{2t}\alpha^{s_{2t}} &= 0 \\ v_1 + v_2\alpha^{2s_2} + \cdots + v_{2t}\alpha^{2s_{2t}} &= 0 \\ &\vdots \\ v_1 + v_2\alpha^{2ts_2} + \cdots + v_{2t}\alpha^{2ts_{2t}} &= 0 \end{aligned}$$

The coefficient matrix is nonsingular because it is Vandermonde and α, α^2, $\ldots$, α^{2t} are all different if $t < 2^{m-1}$.
The linear system must have the unique solution $v_i = 0$, $i = 0, \ldots, 2t$.
Therefore, there are no nonzero code words with fewer than $2t + 1$ ones, and, by Lemma 20 and Proposition 3, the code will correct t or fewer errors. □

Theoretical Results V

- There is, for example, a BCH (127,92)-code that will correct up to five errors. This code adds 35 check digits to the 92 information digits and hence contains 2^{35} syndromes.
- It would be impossible to store all these syndromes and their coset leaders in a computer, so decoding has to be done by other methods.
- The errors in BCH codes can be found by algebraic means without listing the table of syndromes and coset leaders.

References I

- Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, 2nd edition, Wiley, 2006.
- David J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.
- Robert M. Gray, *Entropy and Information Theory*, Springer, 2009
- John C. Bowman, Coding Theory, University of Alberta, Edmonton, Canada, 2003
- D. G. Hoffmann, Coding Theory. The Essential, Marcel Deker, 1991
- W. J. Gilbert, W. K. Nicholson, Modern Algebra with Applications, 2nd edition, Wiley, 2004
- C. E. Shannon, A mathematical theory of communication, *Bell System Technical Journal*, 1948.

References II


R. V. Hamming, Error detecting and error correcting codes, *Bell System Technical Journal*, 29: 147-160, 1950


Reed, Irving S.; Solomon, Gustave, Polynomial Codes over Certain Finite Fields, *Journal of the Society for Industrial and Applied Mathematics* (SIAM) 8 (2): 300–304, 1960