AN INTRODUCTION TO ERROR CORRECTING CODES
Part 1

Jack Keil Wolf
ECE 154C

Spring 2008

Noisy Communications

- Noise in a communications channel can cause errors in the transmission of binary digits.

- Transmit: 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 1 0 …
- Receive: 1 1 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 …

- For some types of information, errors can be detected and corrected but not in others.

Example: Transmit: Come to my house at 17:25 …
Receive: Come tc my houzx at 14:25 …

Making Digits Redundant

- **In binary error correcting codes, only certain binary sequences (called code words) are transmitted.**

- **This is similar to having a dictionary of allowable words.**

- **After transmission over a noisy channel, we can check to see if the received binary sequence is in the dictionary of code words and if not, choose the codeword most similar to what was received.**

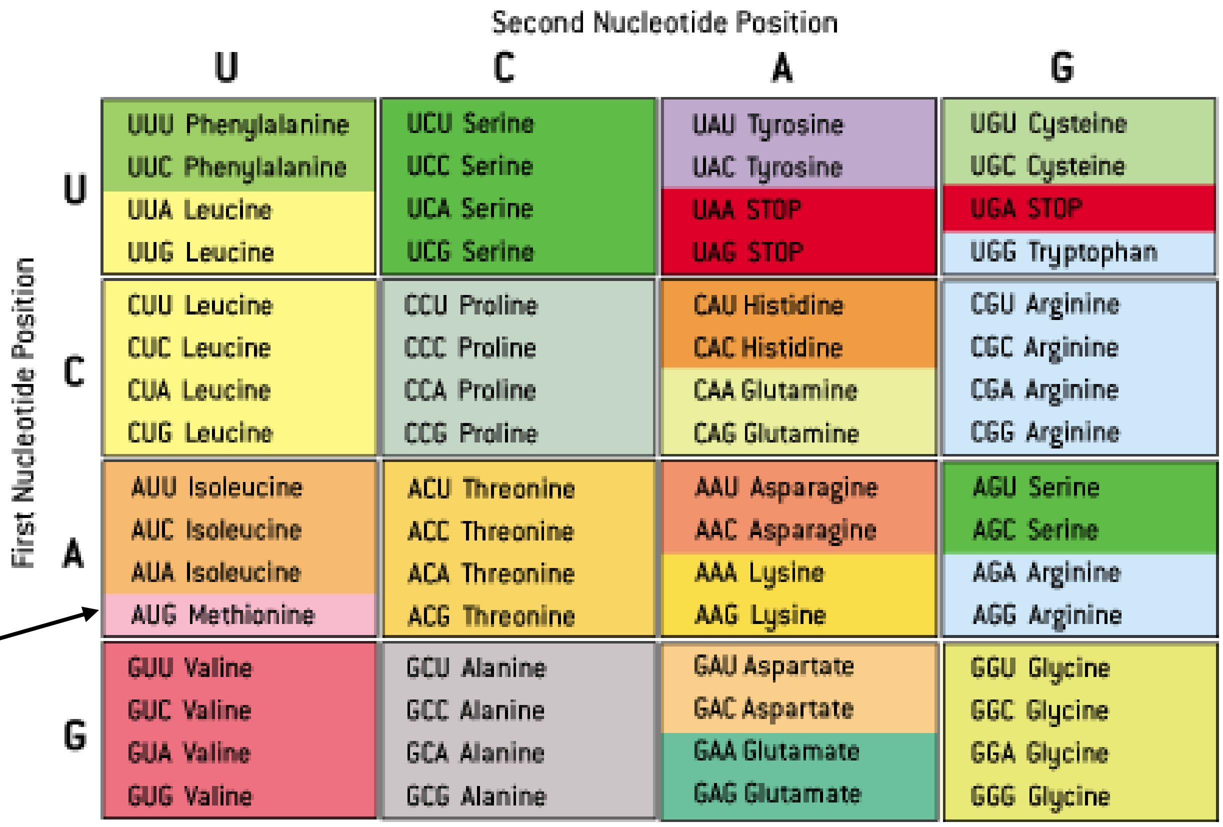
NATURE’S ERROR CONTROL CODE

- **Nature’s code is a mapping of RNA sequences to proteins.**

- **“RNA” consists of four “symbols": A, U, G, and C. “Proteins” consists of 20 “symbols": the amino acids.**

- **The genetic code is a code in which three nucleotides in RNA specify one amino acid in protein.**

NATURE'S ERROR CONTROL DECODING TABLE



	Second Nucleotide Position			
First Nucleotide Position	U	C	A	G
U	UUU Phenylalanine	UCU Serine	UAU Tyrosine	UGU Cysteine
	UUC Phenylalanine	UCC Serine	UAC Tyrosine	UGC Cysteine
	UUA Leucine	UCA Serine	UAA STOP	UGA STOP
	UUG Leucine	UCG Serine	UAG STOP	UGG Tryptophan
C	CUU Leucine	CCU Proline	CAU Histidine	CGU Arginine
	CUC Leucine	CCC Proline	CAC Histidine	CGC Arginine
	CUA Leucine	CCA Proline	CAA Glutamine	CGA Arginine
	CUG Leucine	CCG Proline	CAG Glutamine	CGG Arginine
A	AUU Isoleucine	ACU Threonine	AAU Asparagine	AGU Serine
	AUC Isoleucine	ACC Threonine	AAC Asparagine	AGC Serine
	AUA Isoleucine	ACA Threonine	AAA Lysine	AGA Arginine
	AUG Methionine	ACG Threonine	AAG Lysine	AGG Arginine
G	GUU Valine	GCU Alanine	GAU Aspartate	GGU Glycine
	GUC Valine	GCC Alanine	GAC Aspartate	GGC Glycine
	GUA Valine	GCA Alanine	GAA Glutamate	GGA Glycine
	GUG Valine	GCG Alanine	GAG Glutamate	GGG Glycine

LUCY READING

AUG starts
codon.

Sometimes one
or more of the
RNA symbols
Is changed.

Hopefully, the
resultant triplet
still decodes to
the same protein.

RNA-Amino Acid Coding

OUTLINE

- Types of Error Correction Codes
- Block Codes:
  - Example: (7,4) Hamming Codes
  - General Theory of Binary Group Codes
  - Low Density Parity Check (LDPC) Codes
  - Reed Solomon (RS) Codes
- Convolutional Codes & Viterbi Decoding
  - Example: Rate ½ 4 State Code
  - General Description of Convolutional Codes
  - Punctured Codes
  - Decoding and the Viterbi Algorithm
  - Turbo codes

BINARY ERROR CORRECTING CODES: (ECC)

- 2^k equally likely messages can be represented by k binary digits.

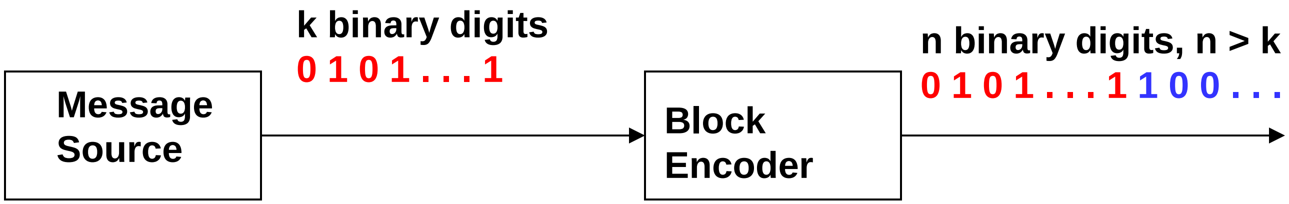
- If these k digits are not coded, an error in one or more of the k binary digits will result in the wrong message being received.

- Error correcting codes is a technique whereby more than the minimum number of binary digits are used to represent the messages.

- The aim of the extra digits, called redundant or parity digits, is to detect and hopefully correct any errors that occurred in transmission.

TWO TYPES OF BINARY CODES

- Block Codes



Message Source

k binary digits

0 1 0 1 . . . 1

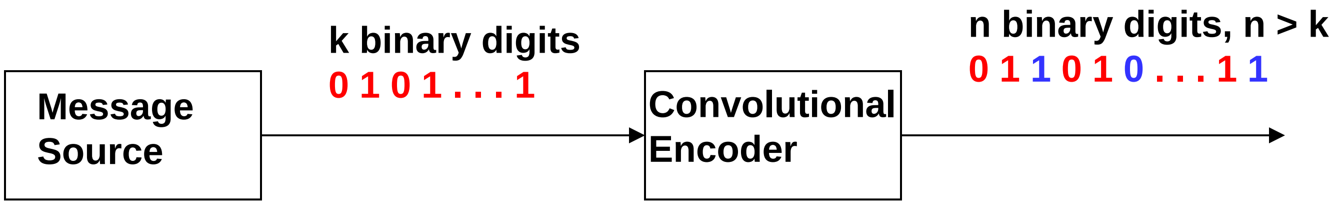
Block Encoder

n binary digits, n > k

0 1 0 1 . . . 1 1 0 0 . . . 1

Rate = k / n

- Convolutional Codes



Message Source

k binary digits

0 1 0 1 . . . 1

Convolutional Encoder

n binary digits, n > k

0 1 1 0 1 0 . . . 1 1

Rate = k / n

TYPES OF ECC

- Binary Codes
  - Encoder and decoder works on a bit basis.
- Nonbinary Codes
  - Encoder and decoder works on a byte or symbol basis.
  - Bytes usually are 8 bits but can be any number of bits.
  - Galois field arithmetic is used.
  - Example is a Reed Solomon Code
  - More generally, we can have codes where the number of symbols is a prime or a power of a prime.

TYPES OF DECODERS – BINARY CASE

- **Hard input decoders**
  - **Input to decoders are 0’s and 1’s.**

- **Soft input decoders**
  - **Input to decoders are probabilities of 0’s and 1’s.**

- **Hard output decoders**
  - **Output of decoders are 0’s and 1’s.**

- **Soft output decoders**
  - **Output of decoders are probabilities of 0’s and 1’s.**

Error Correcting and Detecting Codes

- **Binary block codes are easy to understand.**

- **Block code example:**

Information	Codeword
00	000101
01	010010
10	101101
11	111010

Which codeword was transmitted?

(a) Receive: 111011

(b) Receive: 100101

HAMMING BINARY BLOCK CODE WITH k=4 AND n=7

- In general, a block code with k information digits and block length n is called an (n,k) code.

- Thus, this example is called an (7,4) code.

- This is a very special example where we use pictures to explain the code. Other codes are NOT explainable in this way.

- All that we need to know is modulo 2 addition, ⊕:

  $0 \oplus 0 = 0, \quad 1 \oplus 0 = 1, \quad 0 \oplus 1 = 1, \quad 1 \oplus 1 = 0.$

HAMMING BINARY BLOCK CODE WITH k=4 AND n=7

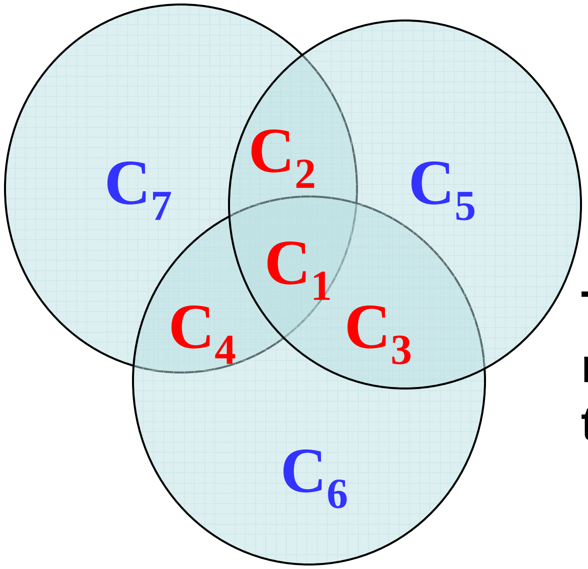
- Message digits: C_1 C_2 C_3 C_4
- Code word C_1 C_2 C_3 C_4 C_5 C_6 C_7

Parity Check Equations:

$$C_1 \oplus C_2 \oplus C_3 \oplus C_5 = 0$$
$$C_1 \oplus C_3 \oplus C_4 \oplus C_6 = 0$$
$$C_1 \oplus C_2 \oplus C_4 \oplus C_7 = 0$$

Parity Check Matrix:

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

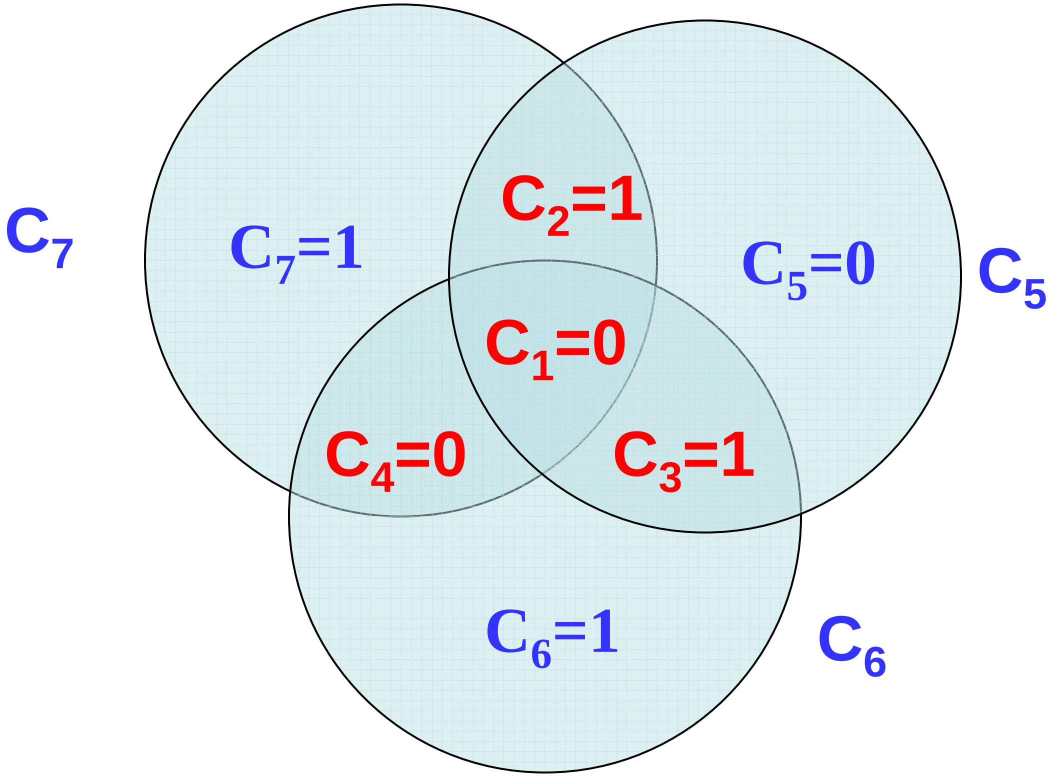


The circles represent the equations.

There is an even number of 1's in each circle.

HAMMING (7,4) CODE: ENCODING

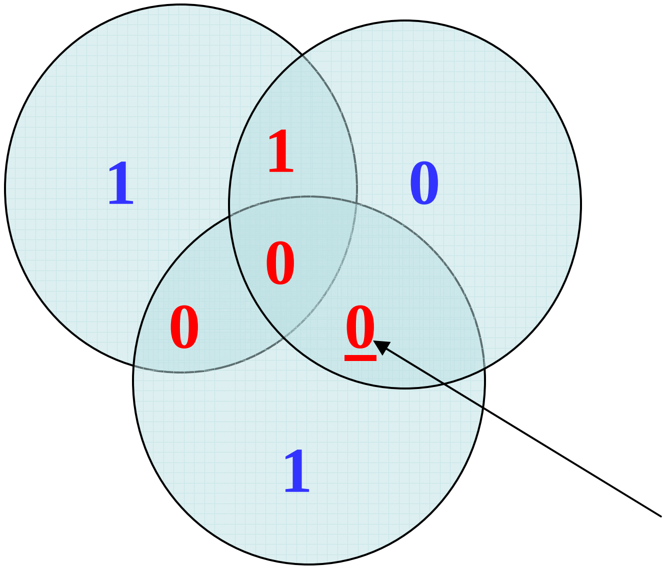
- **Message: $(C_1\ C_2\ C_3\ C_4) = (0\ \ 1\ \ 1\ \ 0)$**



- **Resultant code word: 0 1 1 0 0 1 1**

HAMMING (7,4) CODE: DECODING

- **Transmitted code word:** **0 1 1 0 0 1 1**
- **Example 1:** **Received block with one error in a message bit.** **0 1 0 0 0 1 1**



By counting 1's in each circle we find:

There is an error in right circle.

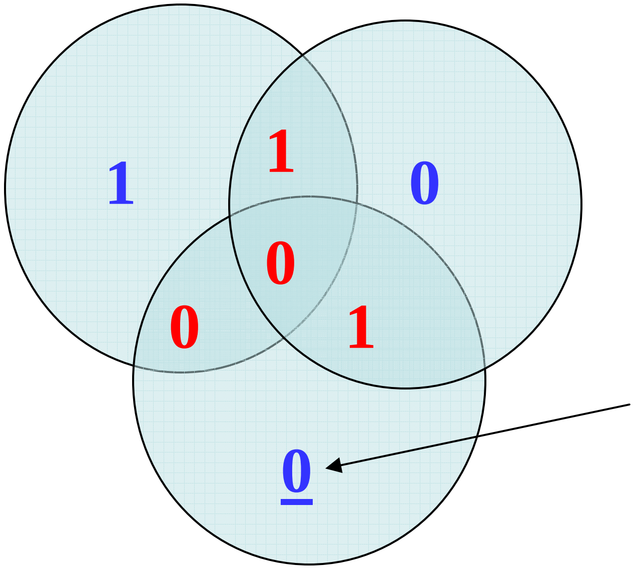
There is an error in bottom circle

There is no error in left circle.

Therefore the error is in the third digit!

HAMMING (7,4) CODE: DECODING

- **Transmitted code word:** 0 1 1 0 0 1 1
- **Example 2:** Received block with one error in parity bit: 0 1 1 0 0 <u>0</u> 1



There is <u>no error</u> in right circle.

There is <u>an error</u> in bottom circle

There is <u>no error</u> in left circle.

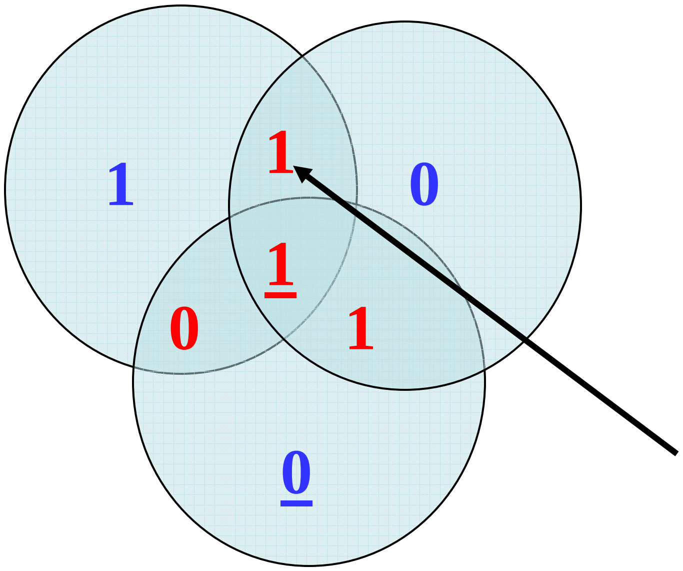
The 6th digit is in error!

HAMMING (7,4) CODE: DECODING

- **Transmitted code word:** **0 1 1 0 0 1 1**

- **Example 3: Received block with two errors:**

  1 1 1 0 0 0 1



There is an error in right circle.

There is no error in bottom circle

There is an error in left circle.

The 2nd digit is in error.

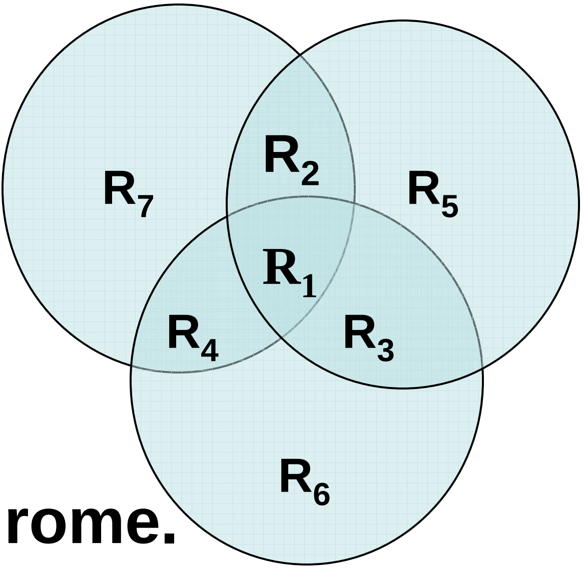
WRONG!!!

HAMMING (7,4) CODE: SYNDROME DECODING

- Let R_1 R_2 R_3 R_4 R_5 R_6 R_7 be the received block of binary digits, possibly with errors.

- Counting 1's in the circles is the same as computing the result of the following equations:

$$R_1 \oplus R_2 \oplus R_3 \oplus R_5 = S_1$$
$$R_1 \oplus R_3 \oplus R_4 \oplus R_6 = S_2$$
$$R_1 \oplus R_2 \oplus R_4 \oplus R_7 = S_3$$

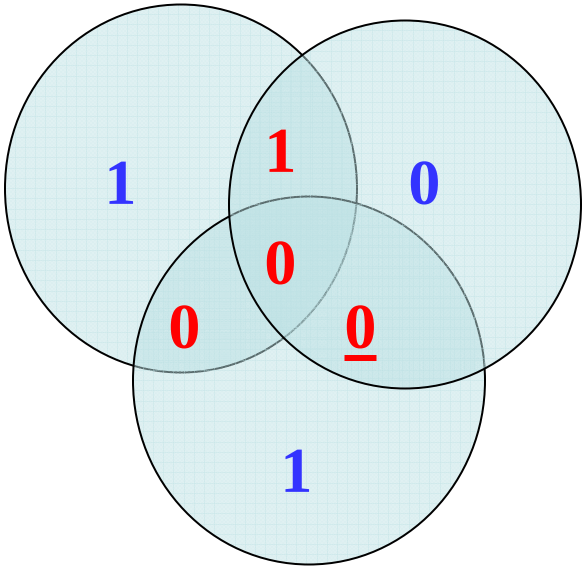


- S_1, S_2 and S_3 is called the syndrome.

HAMMING (7,4) CODE: SYNDROME DECODING

- **Resultant code word: 0 1 1 0 0 1 1**

- **Example 1: Received block with one error in a message bit. 0 1 0 0 0 1 1**



There is an error in right circle. $S_1 = 1$

There is an error in bottom circle. $S_2 = 1$

There is no error in left circle. $S_3 = 0$

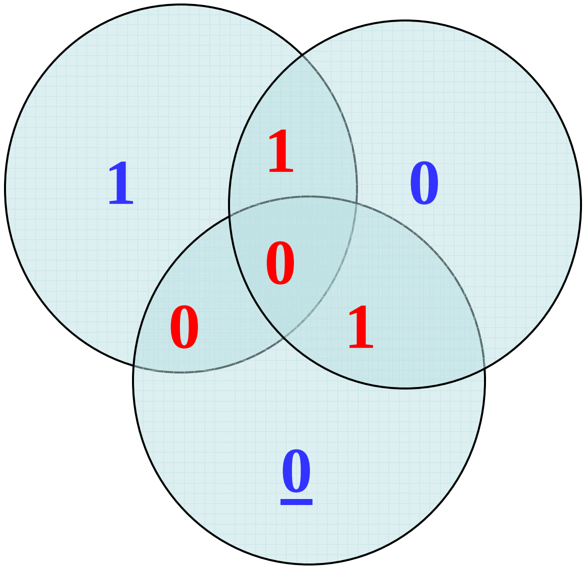
Parity Check Matrix:

$$\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ & & \uparrow & & & & \end{array}$$

HAMMING (7,4) CODE: SYNDROME DECODING

- **Transmitted code word:** 0 1 1 0 0 1 1

- **Example 2:** Received block with one error in parity bit: 0 1 1 0 0 0 1



There is no error in right circle. $S_1=0$

There is an error in bottom circle. $S_2=1$

There is no error in left circle. $S_3=0$

Parity Check Matrix:

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ & & & & & \uparrow & \end{matrix}$$

HAMMING (7,4) CODE: SYNDROME DECODING

- **Thus to correct a single error based upon the received sequence R_1, R_2, R_3, R_4, R_5, R_6, R_7:**
  - **one can first compute the syndrome S_1, S_2, S_3 ,**
  - **and then compare it with the columns of the parity check matrix.**
  - **The matching column is where the error occurred.**
- **This technique will work for any single error correcting code.**

HAMMING (7,4) CODE

- Another way of decoding is to compare the received sequence to all of the code words and choose the one that is "closest" to it, that is differs from it in the fewest number of positions.

- The list of 16 code words for this code is shown on the next slide.

- No matter how we decode, if more than one error occurs in the block of 7 digits the decoder will decode to the wrong code word.

LIST OF CODE WORDS: HAMMING (7,4) CODE

0	0	0	0	0	0	0
1	0	0	0	1	1	1
0	1	0	0	1	0	1
0	0	1	0	1	1	0
0	0	0	1	0	1	1
1	1	0	0	0	1	0
1	0	1	0	0	0	1
1	0	0	1	1	0	0
0	1	1	0	0	1	1
0	1	0	1	1	1	0
0	0	1	1	1	0	1
1	1	1	0	1	0	0
1	1	0	1	0	0	1
1	0	1	1	0	1	0
0	1	1	1	0	0	0
1	1	1	1	1	1	1

PROPERTIES OF BINARY PARITY CHECK CODES

- **An (n,k) binary parity check code (also called an (n,k) group code) is a set of code words of length n, which consist of all of the binary n-vectors which are the solutions of r = (n-k) linearly independent equations called parity check equations.**

- **Each parity check equation specifies a subset of the components of the n-vector which sum to 0, modulo 2.**

- **If one has r = (n-k) linearly independent equations, there will be some set of k of the components of the n-vectors which can be arbitrarily specified such that one can solve for the other r = (n-k) components.**

PROPERTIES OF BINARY PARITY CHECK CODES

- The k components that are specified are called information digits (or message digits) and the other r = (n-k) components are called parity digits (or redundant digits).

- Since there are a set of k binary symbols that can be chosen arbitrarily, these symbols can be filled in 2^k different ways.

- Thus the complete list of code words contains 2^k code words.

- Note that the all-zero vector always satisfies these parity check equations since any subset of the components of the all-zero vector sums to 0 modulo 2.

PROPERTIES OF BINARY PARITY CHECK CODES

- The coefficients of the r = (n-k) linearly independent parity check equations can be written as a matrix called the parity check matrix and is denoted H.

- The parity check matrix has r rows and n columns.

- The i-jth entry (ith row and jth column) in this parity check matrix, $h_{i,j}$, is equal to 1 if and only if the jth component of a code word is contained in the ith parity check equation. Otherwise it is equal to 0.

FOR HAMMING (7,4) CODE

- For the Hamming (7,4) code there were 3 equations

$$C1 \oplus C2 \oplus C3 \oplus C5 = 0$$
$$C1 \oplus C3 \oplus C4 \oplus C6 = 0$$
$$C1 \oplus C2 \oplus C4 \oplus C7 = 0.$$

Thus the parity check matrix for this code is:

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

FOR HAMMING (7,4) CODE

- For the Hamming (7,4) code there were 3 linearly independent equations

$$C1 \oplus C2 \oplus C3 \oplus C5 = 0$$

$$C1 \oplus C3 \oplus C4 \oplus C6 = 0$$

$$C1 \oplus C2 \oplus C4 \oplus C7 = 0$$

so r=3 and k=4. Thus there are $2^4 = 16$ code words in this code.

- Note that the all-zero vector is a code word.

PROPERTIES OF BINARY PARITY CHECK CODES

- Since the parity check equations are linear (modulo 2), if $\underline{C}_1$ is a solution of the equations and if $\underline{C}_2$ is a solution to the equations, then $\underline{C}_1 \oplus \underline{C}_2$ is also a solution to the equations.

- Thus the modulo 2 sum of any two code words is a code word.

- Consider the set of k distinct code words each of which had a single 1 in one of the information positions. Any of the 2^k code words can be constructed by taking a linear combination of these k code words.

- These k code words are said to be generators of the code.

PROPERTIES OF BINARY PARITY CHECK CODES

- A k row by n column matrix made up of these code words is called the generator matrix, G, of the code.

- We will assume that the components of the code words are ordered so that the first k components are the message digits. Then the rows of G can be ordered so that there is a k by k unit matrix on the left.

LIST OF CODE WORDS: HAMMING (7,4) CODE

0	0	0	0	0	0	0
1	0	0	0	1	1	1
0	1	0	0	1	0	1
0	0	1	0	1	1	0
0	0	0	1	0	1	1
1	1	0	0	0	1	0
1	0	1	0	0	0	1
1	0	0	1	1	0	0
0	1	1	0	0	1	1
0	1	0	1	1	1	0
0	0	1	1	1	0	1
1	1	1	0	1	0	0
1	1	0	1	0	0	1
1	0	1	1	0	1	0
0	1	1	1	0	0	0
1	1	1	1	1	1	1

This is the generator matrix of the code.

FOR HAMMING (7,4) CODE

- In the list of 16 code words for the (7,4) Hamming code, the 16 code words can be formed by taking all of the linear combinations of the code words having a single 1 in the information positions. These were:

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

This 4-row by 7-column matrix is the generator matrix, G, of the code. Note the 4 by 4 unit matrix on the left.

PROPERTIES OF BINARY PARITY CHECK CODES

- For any (n,k) binary code, assume the parity check matrix H is of the form:

  $$H = [A \quad 1_{r,r,}]$$

  where A is an arbitrary (n-k) by k binary matrix and where $1_{r,r}$ is an r by r unit matrix.

- Then G is of the form:

  $$G = [1_{k,k} \quad A^T]$$

  where $1_{k,k}$ is a k by k unit matrix and A^T is A transpose. The proof follows.

PROPERTIES OF BINARY PARITY CHECK CODES

- **Since every code word $\underline{C}$ must satisfy the parity check equations, this says that $\underline{C}$ must satisfy the matrix vector equation:**

  $$H\,\underline{C} = \underline{0}.$$

  Here we are assuming that $\underline{C}$ and $\underline{0}$ are column vectors of dimension n and r=(n-k) respectively.

- **But since the rows of G are all code words, the H and G must satisfy the matrix equation:**

  $$H\,G^t = 0.$$

  Here G^t is the transpose of the matrix G.

PROPERTIES OF BINARY PARITY CHECK CODES

- **Proof:**

$$
\begin{aligned}
H\,G^{t} &= [A \;\; 1_{r,r,}]\,[1_{k,k} \;\; A^{T}]^{T} \\
&= [A \;\; 1_{r,r,}] \begin{bmatrix} 1_{k,k} \\ A \end{bmatrix} \\
&= A \oplus A = 0
\end{aligned}
$$

FOR HAMMING (7,4) CODE

- The parity check matrix for this code is:

$$\left[\begin{array}{cccc:ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

and the generator matrix is:

$$\left[\begin{array}{cccc:ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

PROPERTIES OF BINARY PARITY CHECK CODES

- **If $\underline{X}$ and $\underline{Y}$ are any two binary vectors of the same length, define the Hamming distance between $\underline{X}$ and $\underline{Y}$, denoted $d_H(\underline{X},\underline{Y})$, as the number of positions in which $\underline{X}$ and $\underline{Y}$ differ.**

- **For any binary vector $\underline{Z}$, define the Hamming weight of $\underline{Z}$, denoted $w_H(\underline{Z})$, as the number of 1's in the vector $\underline{Z}$.**

- **Then it is easy to see that $d_H(\underline{X},\underline{Y}) = w_H(\underline{X} \oplus \underline{Y})$.**

PROPERTIES OF BINARY PARITY CHECK CODES

- The minimum Hamming distance of a code C, denoted $d_{min}(C)$, is defined as the minimum Hamming distance between any two distinct code words in C.

- For any two code words, $\underline{C}_i$ and $\underline{C}_j$,

$$\underline{C}_i \oplus \underline{C}_j = \underline{C}_k.$$

- But then,

$$d_H(\underline{C}_i\ ,\ \underline{C}_j) = w_H(\underline{C}_i \oplus \underline{C}_j) = w_H(\underline{C}_k)$$

- Thus, $d_{min}(C)$ is equal to the minimum Hamming weight of any non-zero code word.

HAMMING (7,4) CODE

0	0	0	0	0	0	0
1	0	0	0	1	1	1
0	1	0	0	1	0	1
0	0	1	0	1	1	0
0	0	0	1	0	1	1
1	1	0	0	0	1	0
1	0	1	0	0	0	1
1	0	0	1	1	0	0
0	1	1	0	0	1	1
0	1	0	1	1	1	0
0	0	1	1	1	0	1
1	1	1	0	1	0	0
1	1	0	1	0	0	1
1	0	1	1	0	1	0
0	1	1	1	1	0	0
1	1	1	1	1	1	1

For this (7,4) Hamming code, $d_{min} = 3$.

PROPERTIES OF BINARY PARITY CHECK CODES

- **For any code that has minimum distance d_{min}:**
  - **The code can detect any pattern of $(d_{min} - 1)$ or fewer errors.**
  - **The code can fill in any pattern of $(d_{min} - 1)$ or fewer erasures.**
  - **The code can correct any pattern of $\text{int}[(d_{min} - 1)/2]$ or fewer errors.**
  - **The code can simultaneously fill in e or fewer erasures and correct t or fewer errors if:**

$$\boxed{2t + e \leq (d_{min} - 1).}$$

HAMMING (7,4) CODE

- Since the (7,4) Hamming code has minimum distance d_{min}= 3:
  - The code can detect any pattern of 2 or fewer errors. It can detect many more error patterns than that. This will be discussed later.
  - The code can correct any single error.
  - The code can fill in any pattern of 2 or fewer erasures. It can sometimes fill in 3 erasures.

PROPERTIES OF BINARY PARITY CHECK CODES

- Since every code word $\underline{C}$ must satisfy the parity check equations, then $\underline{C}$ must satisfy the equation:

$$H\,\underline{C} = \underline{0}.$$

- Assume that $\underline{C}$ is a code word that has d 1's and (n- d) 0's. Then, d columns of H must sum to $\underline{0}$.

- The smallest value of d for which this is true is d= d_{min}. Thus d_{min} columns of H sum to 0 and no fewer than d_{min} columns of H sum to 0.

- Said in another way, a code has minimum distance d_{min}, if and only if d_{min} columns of H sum to $\underline{0}$ and no fewer than d_{min} columns of H sum to $\underline{0}$.

HAMMING (7,4) CODE

- Consider the parity check matrix for the Hamming (7,4) code:

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

- No single column is $\underline{0}$ and no two columns sum to $\underline{0}$. (Two columns sum to $\underline{0}$ iff the columns are the same.)

- But there are many instance where 3 columns sum to 0: e.g., the 4th, 6th and 7th column of the parity check matrix.

- Thus d_{min} = 3 for the code.

PROPERTIES OF BINARY PARITY CHECK CODES

- **Sometimes we modify a code by adding one more parity digit, called an overall parity digit.**

- **The equation corresponding to this extra parity digit is such that the modulo 2 summation of all of the digits in the code word (including this overall parity digit) is equal to 0.**

- **The result is that the parity check matrix is modified by adding an extra row of all 1's and a column on the right of all 0's and a 1 at the bottom.**

PROPERTIES OF BINARY PARITY CHECK CODES

- **This overall parity digit insures that every code word has even Hamming weight.**

- **Thus if an overall parity digit is appended to a code that had an odd minimum Hamming distance, d_{min}, then the new code has a minimum distance (d_{min} +1).**

- **However, the new code has one more parity digit and the same number of information digits as the original code. (The new code has block length one more than the original code.)**

MODIFYING A HAMMING (7,4) CODE

- Original (7,4) code had a parity check matrix given as:

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

- The new code is an (8,4) code with parity check matrix:

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

- The new code has $d_{min} = 4$.

MODIFYING A HAMMING (7,4) CODE

- **But this parity check matrix does not have a unit matrix on the right. We can make this happen by replacing the last equation with the sum of all of the equations resulting in the parity check matrix:**

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

- **Note that d_{min} = 4 since 4 columns sum to $\underline{0}$ (e.g., the 1st, 5th, 6th and 7th) but no fewer than 4 columns sum to $\underline{0}$.**

HAMMING (8,4) CODE

- The code words in the new code are:

0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0
0	1	0	0	1	0	1	1
0	0	1	0	1	1	0	1
0	0	0	1	0	1	1	1
1	1	0	0	0	1	0	1
1	0	1	0	0	0	1	1
1	0	0	1	1	0	0	1
0	1	1	0	0	1	1	0
0	1	0	1	1	1	0	0
0	0	1	1	1	0	1	0
1	1	1	1	1	0	0	0
1	1	0	1	0	0	1	0
1	0	1	1	0	1	0	0
0	1	1	1	0	0	0	1
1	1	1	1	1	1	1	1

DECODING OF BINARY PARITY CHECK CODES

- **Assume that one is using a code with parity check matrix H and that the transmitter transmits a code word $\underline{C}$.**

- **Assume the receiver receives the binary vector $\underline{R}$ where $\underline{R} = \underline{C} \oplus \underline{E}$. Thus $\underline{E} = \underline{C} \oplus \underline{R}$.**

- **$\underline{E}$ is called the error vector and has a 1 in those positions where $\underline{C}$ and $\underline{R}$ differ (i.e., where there are errors) and 0's elsewhere.**

SYNDROME DECODING OF BINARY PARITY CHECK CODES

- **The decoder first forms the syndrome $\underline{S}$ using the parity check matrix $\underline{H}$ and $\underline{R}$ by calculating:**

$$\underline{S} = H\,\underline{R}.$$

- **Note that since $\underline{R} = \underline{C} \oplus \underline{E}$ and since $H\,\underline{C} = \underline{0}$, then**

$$\underline{S} = H\,\underline{R} = H(\underline{C} \oplus \underline{E}) = H\underline{C} \oplus H\underline{E} = \underline{0} \oplus H\underline{E}.$$

- **Thus $\underline{S} = H\underline{E}$. This says that the syndrome, $\underline{S}$, is equal to the modulo 2 summation of those columns of H where the errors occurred.**

SYNDROME DECODING OF BINARY PARITY CHECK CODES

- But there are many solutions for $\underline{E}$ to the equation

$$\underline{S} = H\underline{E}.$$

- In fact for each possible syndrome $\underline{S}$ there are 2^k different vectors $\underline{E}$ that satisfy that equation since if $\underline{E}$ is a solution so is $\underline{E} \oplus \underline{C}$ for any code word $\underline{C}$.

- For a random error channel with bit error probability $p < 0.5$, the most likely solution for $\underline{E}$ is the one that corresponds to the fewest errors. This means choosing the vector $\underline{E}$ with the fewest non-zero components.

BINARY PARITY CHECK CODES: ENCODING

- There are many circuits that are used in encoding binary parity check codes.

- For any code, if k is not too large, one can use a table of size 2^k by r, where we input the k information digits as an address and look up the r parity digits.

ENCODING THE HAMMING (7,4) CODE USING A TABLE

- Parity check matrix:

$$H= \begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

- Encoding table:

Information Digits	Parity Digits
0 0 0 0	0 0 0
0 0 0 1	0 1 1
0 0 1 0	1 1 0
0 0 1 1	1 0 1
. . . .	. . .
. . . .	. . .
1 1 1 1	1 1 1

BINARY PARITY CHECK CODES: M.L. DECODING

- If the code words are transmitted with equal apriori probability over a B.S.C. with error probability p, p < 0.5, a decoder which results in the smallest probability of word error is as follows:

  Compare the received vector R with every code word and choose the code word that differs from it in the fewest positions.

- This is like the optimal detector found in ECE 154B for the Gaussian channel but we here we use Hamming distance instead of Euclidean distance.

- Since there are 2^k code words, this is impractical if k is large.

BINARY PARITY CHECK CODES: SYNDROME DECODING

- **Assume we first compute the syndrome.**
- **For many codes, one finds the minimum Hamming weight vector E which is the solution to the equation S = HE by algebraic means.**
- **However, if the dimension of S, r, is not too big one can construct a decoding table with 2^r entries that relate the syndrome to the minimum Hamming weight error pattern, E.**
- **Such a decoder would be maximum likelihood.**

DECODING THE HAMMING (7,4) CODE USING A TABLE

- **Parity check matrix:**

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

- **Decoding table:**

Syndrome	**Error Pattern**
0 0 0	0 0 0 0 0 0 0
0 0 1	0 0 0 0 0 0 1
0 1 0	0 0 0 0 0 1 0
0 1 1	0 0 0 1 0 0 0
1 0 0	0 0 0 0 1 0 0
1 0 1	0 1 0 0 0 0 0
1 1 0	0 0 1 0 0 0 0
1 1 1	1 0 0 0 0 0 0

BINARY PARITY CHECK CODES: SYNDROME DECODING BY TABLES

- If both k and r are not too large, two tables can be used to do the entire decoding.

- The syndrome can be calculated from R by using the encoding table with 2^k entries as follows:

  1. The first k components of R are used as the address in the encoding table and the resulting parity bits are added (bit by bit modulo 2) to the last r bits of R.
  2. The result is the syndrome S.

- Then the syndrome is used as the address in the decoding table with 2^r entries and the error pattern is read from the table.

- The error pattern is then added to R to find the decoded code word.

BINARY PARITY CHECK CODES: SYNDROME DECODING

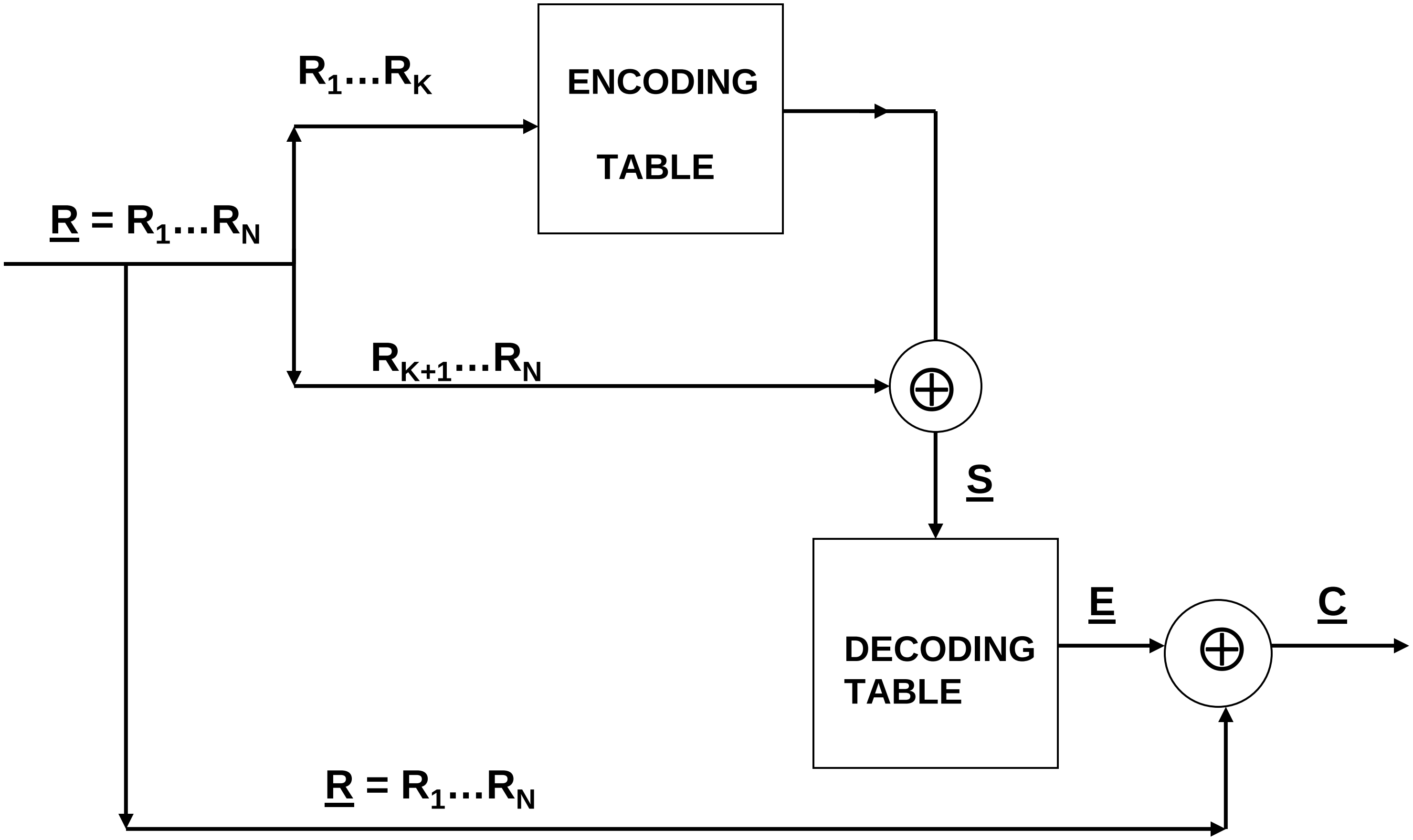
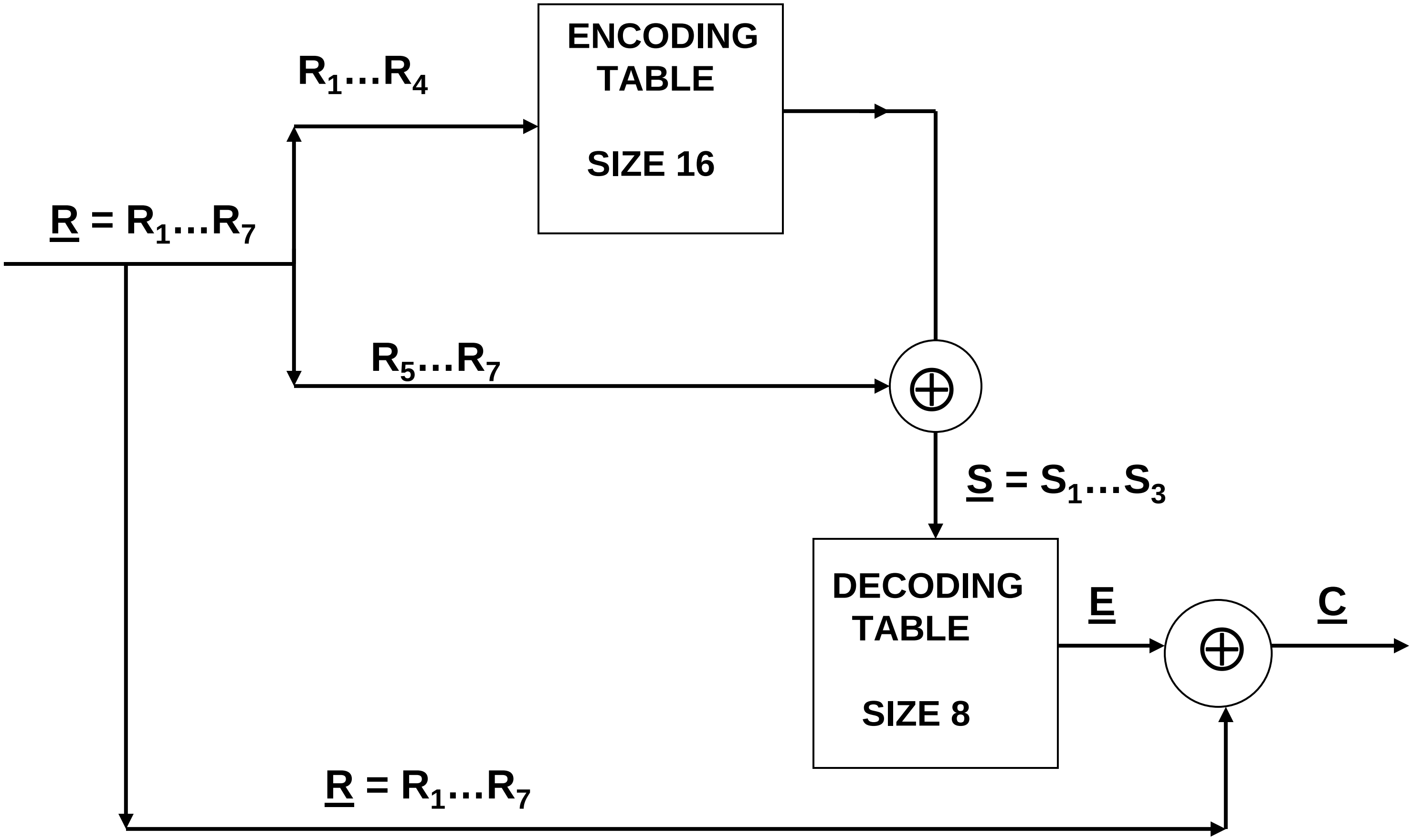


TABLE LOOK UP DECODER: HAMMING (7,4) CODE



CONSTRUCTING OTHER BINARY HAMMING CODES

- One can construct single error correcting binary codes with $d_{min}=3$ having other block lengths.

- If one wants a code of block length $n=(2^r-1)$ for any integer r, the n columns of the parity matrix are chosen as the (2^r-1) distinct non-zero vectors of length r. Note that since there are r rows in the parity matrix, r is the number of parity digits in the code.

n	7	15	31	63	127	255
k	4	11	26	57	120	247
r	3	4	5	6	7	8

THE GOLAY (23,12) CODE

- This code has d_{min}=7. A parity check matrix for this code is:

$$
H= \begin{matrix}
1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0\\
0&1&1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&1&1&1&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&1&1&1&0&1&1&0&0&0&1&0&0&0&0&0&0&0\\
1&1&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&0&0&0&0\\
1&0&0&1&1&1&0&1&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0\\
1&0&1&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&1&0&1&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&1&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&1&0&1&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&1&0\\
1&1&1&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&1
\end{matrix}
$$

THE GOLAY (23,12) CODE

- Since the (23,12) Golay code has d_{min}=7, it can correct all patterns of 1, 2, or 3 errors in each code block of 23 digits.

- The decoder uses two table: one of size 2^{12} and the other of size 2^{11}.

- One can make a (24,12) code with d_{min}=8 by appending an overall parity digit to the (23,12) Golay code. To decode this (24,12) code one could use a two tables of size 2^{12}.

TABLE LOOK UP DECODER: GOLAY (23,12) CODE

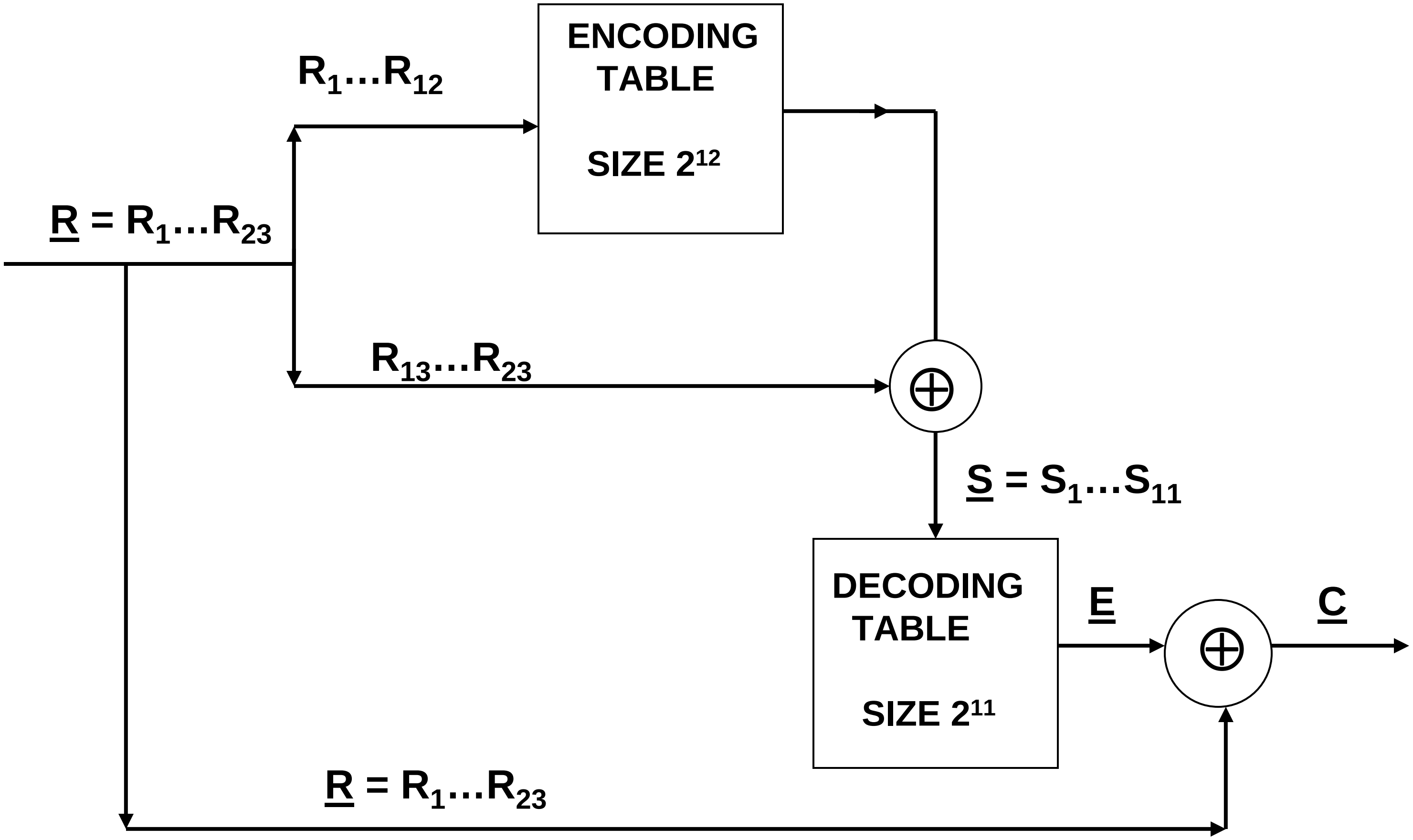
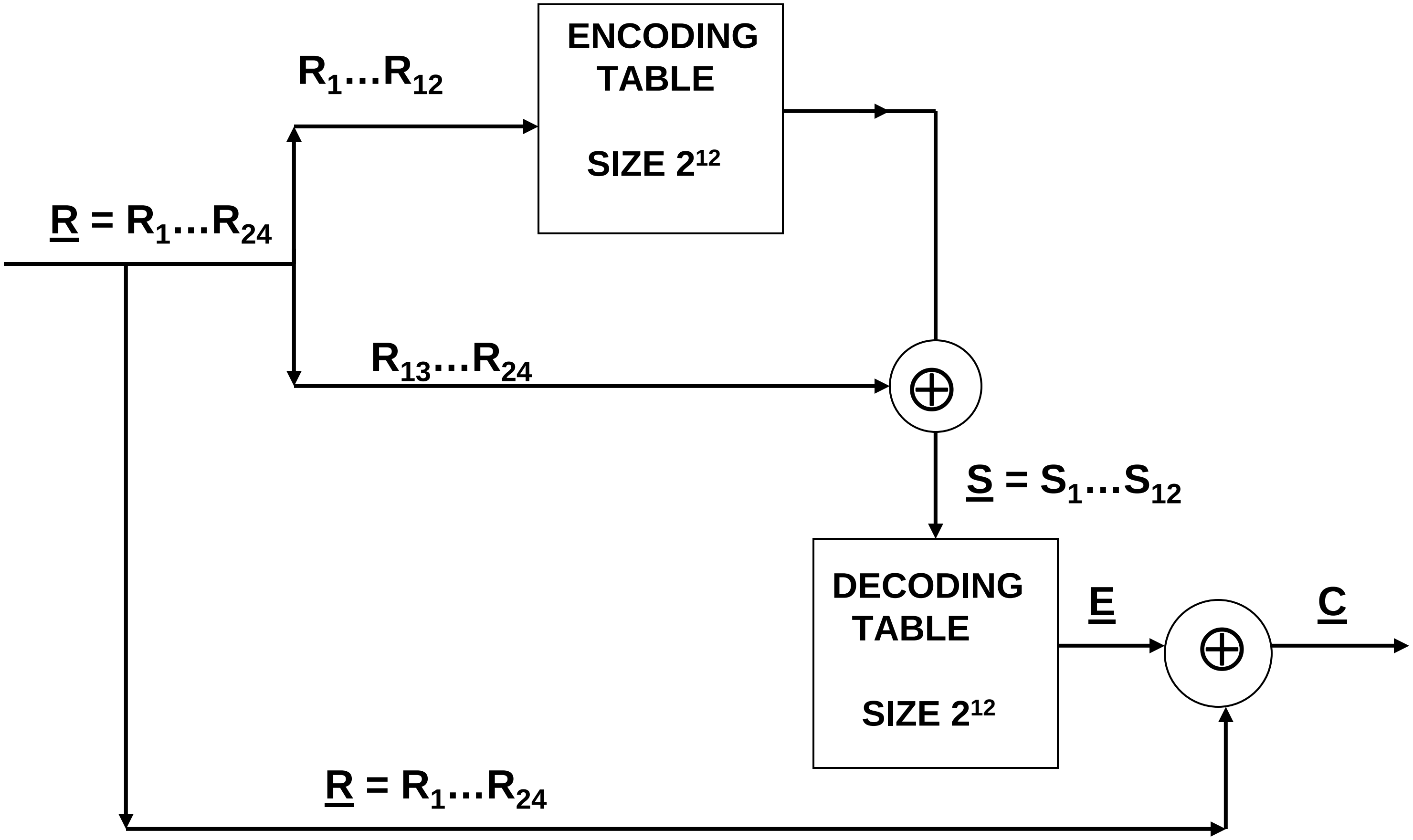


TABLE LOOK UP DECODER: GOLAY (24,12) CODE



SHORTENING BINARY PARITY CHECK CODES

- For any positive integer “a” (a< k), if one starts with an (n,k) binary parity check code with minimum distance d_{min}, one can construct an (n-a, k-a) parity check code with minimum distance at least d_{min}.

- One can do this by setting the setting the first “a” information digits to 0 and not transmitting them.

- The parity check matrix of this (n-a,k-a) code is formed from the parity check matrix of the (n,k) code by eliminating the first “a” columns.

SHORTENING BINARY PARITY CHECK CODES

- For example, if one shortens the (24,12) with d_{min}=8 by 2 digits (i.e., a=2) one would have a (22,10) code with minimum distance at least 8.

- If one shortens the code enough the minimum distance could actually increase. This is true, since the minimum distance between the remaining code words might be greater than the minimum distance between the original list of code words.

- There is no general rule, however, which predicts by how much the minimum distance would increase.

PUNCTURING BINARY PARITY CHECK CODES

- **While shortening a code reduces the number of information digits in a code, puncturing a code reduces the number of parity digits.**

- **One punctures a code by eliminating one or more of the parity equations and thus eliminating the corresponding parity digits.**

- **In general, puncturing a code reduces the minimum distance of the code but increases the code rate, R.**

ERROR DETECTION ONLY

- If one only wants to detect errors, one can compute the syndrome to see if it is all-zero.
  - If the syndrome is all-zero one assumes that no errors occurred since the received vector is a code word.
  - If the syndrome is not all zero, one knows that errors have occurred.

- The only time that the decoder will be incorrect is if the error pattern itself is a code word. Then, the syndrome will be all-zero but errors will have occurred.

- For a binary symmetric channel, if one knows the number of code words of each Hamming weight, one can write an expression for the probability of undetected error.