IN/COSTELLO

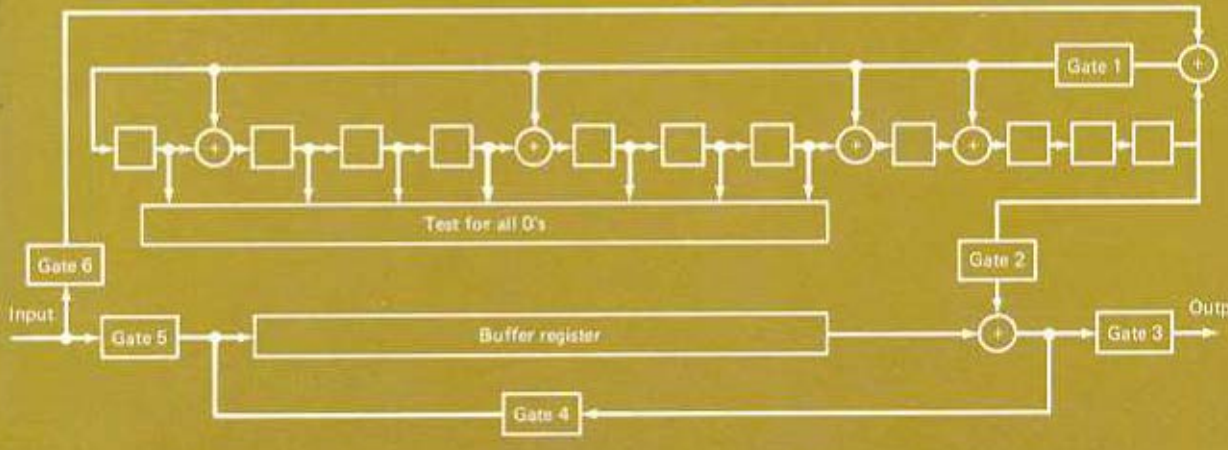
Error Control Coding: Fundamentals and Applications

QA
268
.L55
C.1

SHU LIN / DANIEL J. COSTELLO, Jr.

Error Control Coding:

Fundamentals and Applications



PRENTICE-HALL SERIES IN COMPUTER APPLICATIONS IN ELECTRICAL ENGINEERING
Franklin F. Kuo, *Editor*

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 1
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

ERROR CONTROL CODING

Fundamentals and Applications

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 2
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

PRENTICE-HALL COMPUTER APPLICATIONS IN ELECTRICAL ENGINEERING SERIES

FRANKLIN F. KUO, editor

ABRAMSON and KUO, Computer-Communication Networks
BOWERS and SEDORE, Sceptre: A Computer Program for Circuit and Systems Anal
CADZOW, Discrete Time Systems: An Introduction with Interdisciplinary Applications
CADZOW and MARTENS, Discrete-Time and Computer Control Systems
DAVIS, Computer Data Displays
FRIEDMAN and MENON, Fault Detection in Digital Circuits
HUELSMAN, Basic Circuit Theory
JENSEN and LIEBERMAN, IBM Circuit Analysis Program: Techniques and Applications
JENSEN and WATKINS, Network Analysis: Theory and Computer Methods
KLINE, Digital Computer Design
KOCHENBURGER, Computer Simulation of Dynamic Systems
KUO, (ed.) Protocols and Techniques for Data Communication Networks
KUO and MAGNUSON, Computer Oriented Circuit Design
LIN, An Introduction to Error-Correcting Codes
LIN and COSTELLO, Error Control Coding: Fundamentals and Applications
NAGLE, CARROLL, and IRWIN, An Introduction to Computer Logic
RHYNE, Fundamentals of Digitals Systems Design
SIFFERLEN and VARTANIAN, Digital Electronics with Engineering Applications
STAUDHAMMER, Circuit Analysis by Digital Computer
STOUTEMYER, PL/1 Programming for Engineering and Science

ERROR CONTROL CODING

Fundamentals and Applications

SHU LIN
University of Hawaii
Texas A&M University

DANIEL J. COSTELLO, JR.
Illinois Institute of Technology

Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 4
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

Library of Congress Cataloging in Publication Data

LIN, SHU.
Error control coding.

(Prentice-Hall computer applications in electrical engineering series)
Includes bibliographical references and index.
1. Error-correcting codes (Information theory)
1. Costello, Daniel J. . II. Title.
III. Series.
QA268.L55 001.53'9 82-5255
ISBN 0-13-283796-X AACR2

Editorial/production supervision and interior design by Anne Simpson
Cover design by Marvin Warshaw
Manufacturing buyer: Joyce Levatino

© 1983 by Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632

All rights reserved. No part of this book
may be reproduced in any form or
by any means without permission in writing
from the publisher.

Printed in the United States of America

10 9 8 7

ISBN 0-13-283796-X

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
EDITORA PRENTICE-HALL DO BRAZIL, LTDA, *Rio de Janeiro*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

3

Linear Block Codes

In this chapter basic concepts of block codes are introduced. For ease of code synthesis and implementation, we restrict our attention to a subclass of the class of all block codes, the *linear block codes*. Since in most present digital computers and digital data communication systems, information is coded in binary digits "0" or "1," we discuss only the linear block codes with symbols from the binary field GF(2). The theory developed for the binary codes can be generalized to codes with symbols from a nonbinary field in a straightforward manner.

First, linear block codes are defined and described in terms of *generator* and *parity-check* matrices. The parity-check equations for a *systematic* code are derived. Encoding of linear block codes is discussed. In Section 3.2 the concept of *syndrome* is introduced. The use of syndrome for error detection and correction is discussed. In Sections 3.3 and 3.4 we define the *minimum distance* of a block code and show that the random-error-detecting and random-error-correcting capabilities of a code are determined by its minimum distance. Probabilities of a decoding error are discussed. In Section 3.5 the *standard array* and its application to the decoding of linear block codes are presented. A general decoder based on the *syndrome decoding* scheme is given. Finally, we conclude the chapter by presenting a class of single-error-correcting linear codes.

References 1 to 4 contain excellent treatments of linear block codes.

3.1 INTRODUCTION TO LINEAR BLOCK CODES

We assume that the output of an information source is a sequence of binary digits "0" or "1." In block coding, this binary information sequence is segmented into *message* blocks of fixed length; each message block, denoted by **u**, consists of k

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 6
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

information digits. There are a total of 2^k distinct messages. The encoder, *according to certain rules*, transforms each input message **u** into a binary n-tuple **v** with $n > k$. This binary n-tuple **v** is referred to as the *code word* (or *code vector*) of the message **u**. Therefore, corresponding to the 2^k possible messages, there are 2^k code words. This set of 2^k code words is called a *block* code. For a block code to be useful, the 2^k code words must be distinct. Therefore, there should be a one-to-one correspondence between a message **u** and its code word **v**.

For a block code with 2^k code words and length n, unless it has a certain special structure, the encoding apparatus would be prohibitively complex for large k and n since it has to store the 2^k code words of length n in a dictionary. Therefore, we must restrict our attention to block codes that can be mechanized in a practical manner. A desirable structure for a block code to possess is the *linearity*. With this structure in a block code, the encoding complexity will be greatly reduced, as we will see.

Definition 3.1. A block code of length n and 2^k code words is called a *linear* (n, k) code if and only if its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuples over the field GF(2).

In fact, a binary block code is linear if and only if the modulo-2 sum of two code words is also a code word. The block code given in Table 3.1 is a (7, 4) linear code. One can easily check that the sum of any two code words in this code is also a code word.

Since an (n, k) linear code C is a k-dimensional subspace of the vector space V_n of all the binary n-tuples, it is possible to find k linearly independent code words,

TABLE 3.1 LINEAR BLOCK CODE WITH $k = 4$ AND $n = 7$

Messages	Code words
(0 0 0 0)	(0 0 0 0 0 0 0)
(1 0 0 0)	(1 1 0 1 0 0 0)
(0 1 0 0)	(0 1 1 0 1 0 0)
(1 1 0 0)	(1 0 1 1 1 0 0)
(0 0 1 0)	(1 1 1 0 0 1 0)
(1 0 1 0)	(0 0 1 1 0 1 0)
(0 1 1 0)	(1 0 0 0 1 1 0)
(1 1 1 0)	(0 1 0 1 1 1 0)
(0 0 0 1)	(1 0 1 0 0 0 1)
(1 0 0 1)	(0 1 1 1 0 0 1)
(0 1 0 1)	(1 1 0 0 1 0 1)
(1 1 0 1)	(0 0 0 1 1 0 1)
(0 0 1 1)	(0 1 0 0 0 1 1)
(1 0 1 1)	(1 0 0 1 0 1 1)
(0 1 1 1)	(0 0 1 0 1 1 1)
(1 1 1 1)	(1 1 1 1 1 1 1)

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 7
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

$\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}$ in C such that every code word $\mathbf{v}$ in C is a linear combination of these k code words, that is,

$$\mathbf{v} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \cdots + u_{k-1}\mathbf{g}_{k-1}, \tag{3.1}$$

where $u_i = 0$ or 1 for $0 \leq i < k$. Let us arrange these k linearly independent code words as the rows of a $k \times n$ matrix as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix}, \tag{3.2}$$

where $\mathbf{g}_i = (g_{i0}, g_{i1}, \ldots, g_{i,n-1})$ for $0 \leq i < k$. If $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ is the message to be encoded, the corresponding code word can be given as follows:

$$\begin{aligned} \mathbf{v} &= \mathbf{u} \cdot \mathbf{G} \\ &= (u_0, u_1, \ldots, u_{k-1}) \cdot \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} \\ &= u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \cdots + u_{k-1}\mathbf{g}_{k-1}. \end{aligned} \tag{3.3}$$

Clearly, the rows of $\mathbf{G}$ *generate* (or *span*) the (n, k) linear code C. For this reason, the matrix $\mathbf{G}$ is called a *generator* matrix for C. Note that any k linearly independent code words of an (n, k) linear code can be used to form a generator matrix for the code. It follows from (3.3) that an (n, k) linear code is completely specified by the k rows of a generator matrix $\mathbf{G}$. Therefore, the encoder has only to store the k rows of $\mathbf{G}$ and to form a linear combination of these k rows based on the input message $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$.

Example 3.1

The (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

If $\mathbf{u} = (1\ \ 1\ \ 0\ \ 1)$ is the message to be encoded, its corresponding code word, according to (3.3), would be

$$\begin{aligned} \mathbf{v} &= 1 \cdot \mathbf{g}_0 + 1 \cdot \mathbf{g}_1 + 0 \cdot \mathbf{g}_2 + 1 \cdot \mathbf{g}_3 \\ &= (1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0) + (0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0) + (1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1) \\ &= (0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1). \end{aligned}$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 8
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

A desirable property for a linear block code to possess is the *systematic structure* of the code words as shown in Figure 3.1, where a code word is divided into two parts, the message part and the redundant checking part. The message part consists of k unaltered information (or message) digits and the redundant checking part consists of $n - k$ *parity-check* digits, which are *linear sums* of the information digits. A linear block code with this structure is referred to as a *linear systematic block code.* The (7, 4) code given in Table 3.1 is a linear systematic block code, the rightmost four digits of each code word are identical to the corresponding information digits.

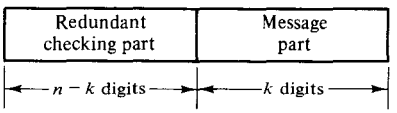


Figure 3.1 Systematic format of a code word.

A linear systematic (n, k) code is completely specified by a $k \times n$ matrix $\mathbf{G}$ of the following form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} = \left[\begin{array}{cccc|ccccc} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ p_{20} & p_{21} & \cdots & p_{2,n-k-1} & 0 & 0 & 1 & \cdots & 0 \\ & & & & & & & & \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{array}\right], \tag{3.4}$$

($\mathbf{P}$ matrix; $k \times k$ identity matrix)

where $p_{ij} = 0$ or 1. Let $\mathbf{I}_k$ denote the $k \times k$ identity matrix. Then $\mathbf{G} = [\mathbf{P}\ \mathbf{I}_k]$. Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded. The corresponding code word is

$$\begin{aligned} \mathbf{v} &= (v_0, v_1, v_2, \ldots, v_{n-1}) \\ &= (u_0, u_1, \ldots, u_{k-1}) \cdot \mathbf{G}. \end{aligned} \tag{3.5}$$

It follows from (3.4) and (3.5) that the components of $\mathbf{v}$ are

$$v_{n-k+i} = u_i \qquad \text{for } 0 \leq i < k \tag{3.6a}$$

and

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} \tag{3.6b}$$

for $0 \leq j < n - k$. Equation (3.6a) shows that the rightmost k digits of a code word $\mathbf{v}$ are identical to the information digits $u_0, u_1, \ldots, u_{k-1}$ to be encoded, and (3.6b) shows that the leftmost $n - k$ redundant digits are linear sums of the information digits. The $n - k$ equations given by (3.6b) are called *parity-check equations* of the code.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 9
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

Example 3.2

The matrix **G** given in Example 3.1 is in systematic form. Let $\mathbf{u} = (u_0, u_1, u_2, u_3)$ be the message to be encoded and let $\mathbf{v} = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word. Then

$$\mathbf{v} = (u_0, u_1, u_2, u_3) \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

By matrix multiplication, we obtain the following digits of the code word **v**:

$$\begin{aligned} v_6 &= u_3 \\ v_5 &= u_2 \\ v_4 &= u_1 \\ v_3 &= u_0 \\ v_2 &= u_1 + u_2 + u_3 \\ v_1 &= u_0 + u_1 + u_2 \\ v_0 &= u_0 + u_2 + u_3. \end{aligned}$$

The code word corresponding to the message (1 0 1 1) is (1 0 0 1 0 1 1).

There is another useful matrix associated with every linear block code. As stated in Chapter 2, for any $k \times n$ matrix **G** with k linearly independent rows, there exists an $(n - k) \times n$ matrix **H** with $n - k$ linearly independent rows such that any vector in the row space of **G** is orthogonal to the rows of **H** and any vector that is orthogonal to the rows of **H** is in the row space of **G**. Hence, we can describe the (n, k) linear code generated by **G** in an alternate way as follows: *An n-tuple* **v** *is a code word in the code generated by* **G** *if and only if* $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$. This matrix **H** is called a *parity-check matrix* of the code. The 2^{n-k} linear combinations of the rows of matrix **H** form an $(n, n - k)$ linear code C_d. This code is the null space of the (n, k) linear code C generated by matrix **G** (i.e., for any $\mathbf{v} \in C$ and any $\mathbf{w} \in C_d$, $\mathbf{v} \cdot \mathbf{w} = 0$). C_d is called the *dual* code of C. Therefore, a parity-check matrix for a linear code C is a generator matrix for its dual code C_d.

If the generator matrix of an (n, k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form:

$$\begin{aligned} \mathbf{H} &= [\mathbf{I}_{n-k} \quad \mathbf{P}^T] \\ &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \cdots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}, \end{aligned} \tag{3.7}$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 10
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

where $\mathbf{P}^T$ is the transpose of the matrix $\mathbf{P}$. Let $\mathbf{h}_j$ be the jth row of $\mathbf{H}$. We can check readily that the inner product of the ith row of $\mathbf{G}$ given by (3.4) and the jth row of $\mathbf{H}$ given by (3.7) is

$$\mathbf{g}_i \cdot \mathbf{h}_j = p_{ij} + p_{ij} = 0$$

for $0 \leq i < k$ and $0 \leq j < n - k$. This implies that $\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}$. Also, the $n - k$ rows of $\mathbf{H}$ are linearly independent. Therefore, the $\mathbf{H}$ matrix of (3.7) is a parity-check matrix of the (n, k) linear code generated by the matrix $\mathbf{G}$ of (3.4).

The parity-check equations given by (3.6b) can also be obtained from the parity-check matrix $\mathbf{H}$ of (3.7). Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded. In systematic form the corresponding code word would be

$$\mathbf{v} = (v_0, v_1, \ldots, v_{n-k-1}, u_0, u_1, \ldots, u_{k-1}).$$

Using the fact that $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$, we obtain

$$v_j + u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} = 0 \tag{3.8}$$

for $0 \leq j < n - k$. Rearranging the equations of (3.8), we obtain the same parity-check equations of (3.6b). Therefore, an (n, k) linear code is completely specified by its parity-check matrix.

Example 3.3

Consider the generator matrix of a (7, 4) linear code given in Example 3.1. The corresponding parity-check matrix is

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

At this point, let us summarize the foregoing results: For any (n, k) linear block code C, there exists a $k \times n$ matrix $\mathbf{G}$ whose row space gives C. Furthermore, there exists an $(n - k) \times n$ matrix $\mathbf{H}$ such that an n-tuple $\mathbf{v}$ is a code word in C if and only if $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$. If $\mathbf{G}$ is of the form given by (3.4), then $\mathbf{H}$ may take the form given by (3.7), and vice versa.

Based on the equations of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily. The encoding circuit is shown in Figure 3.2, where $\rightarrow\boxed{\ }\rightarrow$ denotes a shift-register stage (e.g., a flip-flop), $\oplus$ denotes a modulo-2 adder, and $\rightarrow(p_{ij})\rightarrow$ denotes a connection if $p_{ij} = 1$ and no connection if $p_{ij} = 0$. The encoding operation is very simple. The message $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ to be encoded is shifted into the message register and simultaneously into the channel. As soon as the entire message has entered the message register, the $n - k$ parity-check digits are formed at the outputs of the $n - k$ modulo-2 adders. These parity-check digits are then serialized and shifted into the channel. We see that the complexity of the encoding circuit is linearly proportional to the block length of the code. The encoding circuit for the (7, 4) code given in Table 3.1 is shown in Figure 3.3, where the connection is based on the parity-check equations given in Example 3.2.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 11
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

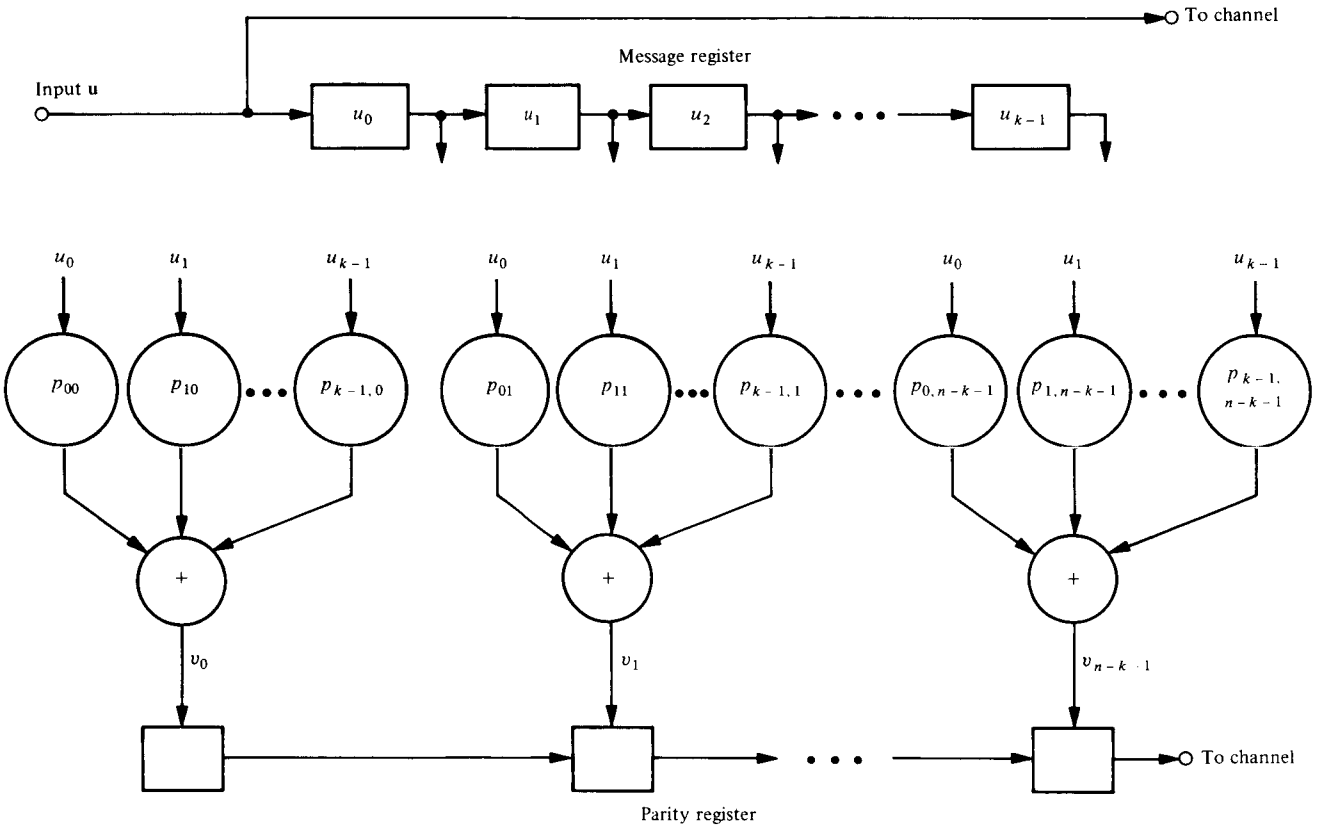


Figure 3.2 Encoding circuit for a linear systematic (n, k) code.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 12
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

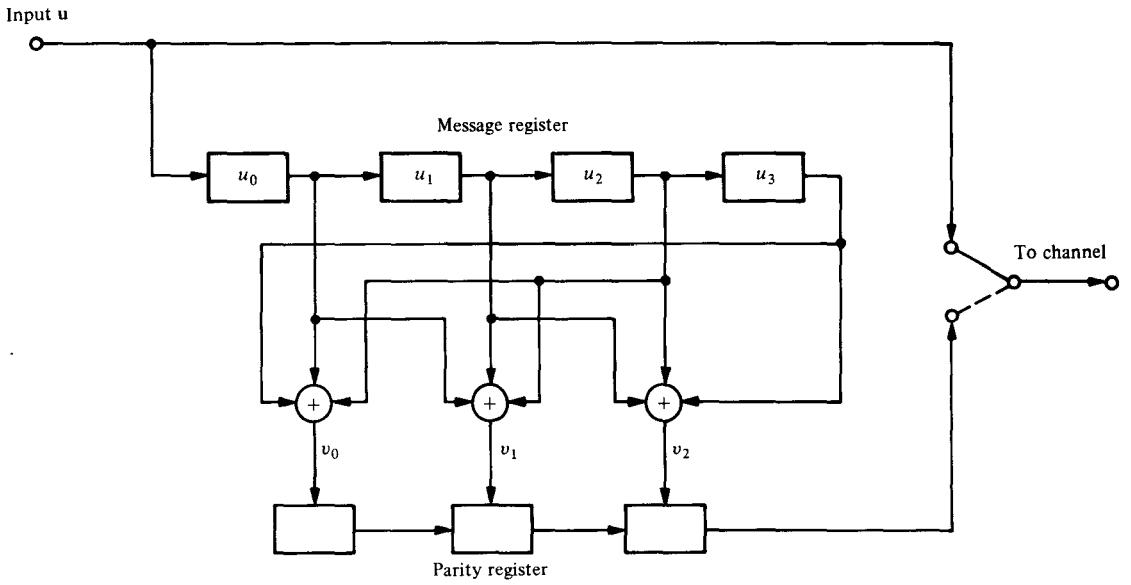


Figure 3.3 Encoding circuit for the (7, 4) systematic code given in Table 3.1.

3.2 SYNDROME AND ERROR DETECTION

Consider an (n, k) linear code with generator matrix $\mathbf{G}$ and parity-check matrix $\mathbf{H}$. Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a code word that was transmitted over a noisy channel. Let $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$ be the received vector at the output of the channel. Because of the channel noise, $\mathbf{r}$ may be different from $\mathbf{v}$. The vector sum

$$\begin{aligned} \mathbf{e} &= \mathbf{r} + \mathbf{v} \\ &= (e_0, e_1, \ldots, e_{n-1}) \end{aligned} \tag{3.9}$$

is an n-tuple where $e_i = 1$ for $r_i \neq v_i$ and $e_i = 0$ for $r_i = v_i$. This n-tuple is called the *error vector* (or *error pattern*). The 1's in $\mathbf{e}$ are the *transmission errors* caused by the channel noise. It follows from (3.9) that the received vector $\mathbf{r}$ is the vector sum of the transmitted code word and the error vector, that is,

$$\mathbf{r} = \mathbf{v} + \mathbf{e}.$$

Of course, the receiver does not know either $\mathbf{v}$ or $\mathbf{e}$. Upon receiving $\mathbf{r}$, the decoder must first determine whether $\mathbf{r}$ contains transmission errors. If the presence of errors is detected, the decoder will either take actions to locate the errors and correct them (FEC) or request for a retransmission of $\mathbf{v}$(ARQ).

When $\mathbf{r}$ is received, the decoder computes the following $(n - k)$-tuple:

$$\begin{aligned} \mathbf{s} &= \mathbf{r} \cdot \mathbf{H}^T \\ &= (s_0, s_1, \ldots, s_{n-k-1}). \end{aligned} \tag{3.10}$$

which is called the *syndrome* of $\mathbf{r}$. Then $\mathbf{s} = \mathbf{0}$ if and only if $\mathbf{r}$ is a code word, and $\mathbf{s} \neq \mathbf{0}$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 13
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

if and only if $\mathbf{r}$ is not a code word. Therefore, when $\mathbf{s} \neq \mathbf{0}$, we know that $\mathbf{r}$ is not a code word and the presence of errors has been detected. When $\mathbf{s} = \mathbf{0}$, $\mathbf{r}$ is a code word and the receiver accepts $\mathbf{r}$ as the transmitted code word. It is possible that the errors in certain error vectors are not detectable (i.e., $\mathbf{r}$ contains errors but $\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = \mathbf{0}$). This happens when the error pattern $\mathbf{e}$ is identical to a nonzero code word. In this event, $\mathbf{r}$ is the sum of two code words which is a code word, and consequently $\mathbf{r} \cdot \mathbf{H}^T = \mathbf{0}$. Error patterns of this kind are called *undetectable* error patterns. Since there are $2^k - 1$ nonzero code words, there are $2^k - 1$ undetectable error patterns. When an undetectable error pattern occurs, the decoder makes a *decoding error*. In a later section of the chapter we derive the probability of an undetected error for a BSC and show that this error probability can be made very small.

Based on (3.7) and (3.10), the syndrome digits are as follows:

$$\begin{aligned}
s_0 &= r_0 + r_{n-k}p_{00} + r_{n-k+1}p_{10} + \cdots + r_{n-1}p_{k-1,0} \\
s_1 &= r_1 + r_{n-k}p_{01} + r_{n-k+1}p_{11} + \cdots + r_{n-1}p_{k-1,1} \\
&\vdots \\
s_{n-k-1} &= r_{n-k-1} + r_{n-k}p_{0,n-k-1} + r_{n-k+1}p_{1,n-k-1} + \cdots + r_{n-1}p_{k-1,n-k-1}.
\end{aligned} \tag{3.11}$$

If we examine the equations above carefully, we find that the syndrome $\mathbf{s}$ is simply the vector sum of the received parity digits $(r_0, r_1, \ldots, r_{n-k-1})$ and the parity-check digits recomputed from the received information digits $r_{n-k}, r_{n-k+1}, \ldots, r_{n-1}$. Therefore, the syndrome can be formed by a circuit similar to the encoding circuit. A general syndrome circuit is shown in Figure 3.4.

Example 3.4

Consider the (7, 4) linear code whose parity-check matrix is given in Example 3.3. Let $\mathbf{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector. Then the syndrome is given by

$$\begin{aligned}
\mathbf{s} &= (s_0, s_1, s_2) \\
&= (r_0, r_1, r_2, r_3, r_4, r_5, r_6) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.
\end{aligned}$$

The syndrome digits are

$$\begin{aligned}
s_0 &= r_0 + r_3 + r_5 + r_6 \\
s_1 &= r_1 + r_3 + r_4 + r_5 \\
s_2 &= r_2 + r_4 + r_5 + r_6.
\end{aligned}$$

The syndrome circuit for this code is shown in Figure 3.5.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 14
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

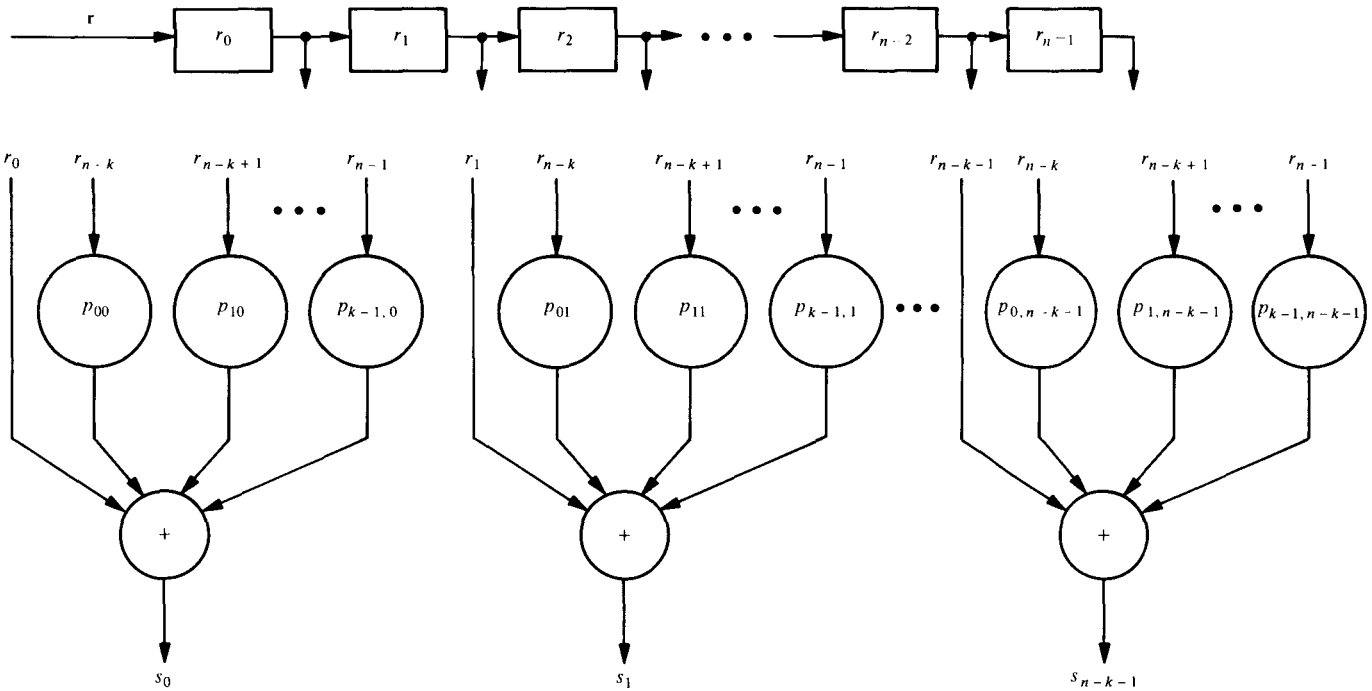


Figure 3.4 Syndrome circuit for a linear systematic (n, k) code.

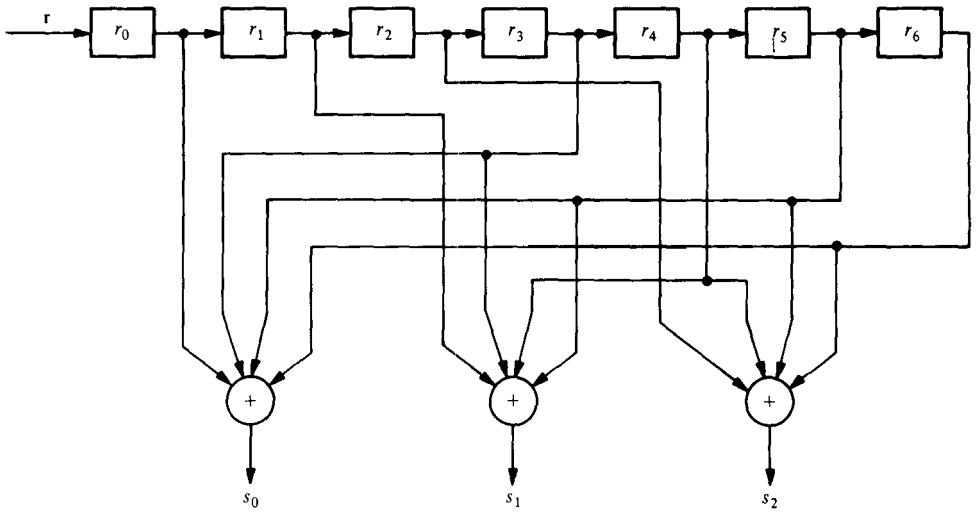


Figure 3.5 Syndrome circuit for the (7, 4) code given in Table 3.1.

The syndrome **s** computed from the received vector **r** actually depends only on the error pattern **e**, and not on the transmitted code word **v**. Since **r** is the vector sum of **v** and **e**, it follows from (3.10) that

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (\mathbf{v} + \mathbf{e})\mathbf{H}^T = \mathbf{v} \cdot \mathbf{H}^T + \mathbf{e} \cdot \mathbf{H}^T.$$

However, $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$. Consequently, we obtain the following relation between the syndrome and the error pattern:

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T. \tag{3.12}$$

If the parity-check matrix **H** is expressed in the systematic form as given by (3.7), multiplying out $\mathbf{e} \cdot \mathbf{H}^T$ yields the following linear relationship between the syndrome digits and the error digits:

$$\begin{aligned}
s_0 &= e_0 + e_{n-k}p_{00} + e_{n-k+1}p_{10} + \cdots + e_{n-1}p_{k-1,0} \\
s_1 &= e_1 + e_{n-k}p_{01} + e_{n-k+1}p_{11} + \cdots + e_{n-1}p_{k-1,1} \\
&\cdots \\
s_{n-k-1} &= e_{n-k-1} + e_{n-k}p_{0,n-k-1} + e_{n-k+1}p_{1,n-k-1} + \cdots + e_{n-1}p_{k-1,n-k-1}.
\end{aligned} \tag{3.13}$$

The syndrome digits are simply linear combinations of the error digits. Clearly, they provide information about the error digits and therefore can be used for error correction.

At this point, one would feel that any error correction scheme is a method of solving the $n - k$ linear equations of (3.13) for the error digits. Once the error pattern **e** has been found, the vector $\mathbf{r} + \mathbf{e}$ is taken as the actual transmitted code word. Unfortunately, determining the true error vector **e** is not a simple matter. This is because the $n - k$ linear equations of (3.13) do not have a unique solution but have 2^k solutions (this will be proved in Theorem 3.6). In other words, there are 2^k error

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 16
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

patterns that result in the same syndrome, and the true error pattern $\mathbf{e}$ is just one of them. Therefore, the decoder has to determine the true error vector from a set of 2^k candidates. To minimize the probability of a decoding error, the most *probable* error pattern that satisfies the equations of (3.13) is chosen as the true error vector. If the channel is a BSC, the most probable error pattern is the one that has the smallest number of nonzero digits.

The notion of using syndrome for error correction may be clarified by an example.

Example 3.5

Again, we consider the (7, 4) code whose parity-check matrix is given in Example 3.3. Let $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ be the transmitted code word and $\mathbf{r} = (1\ 0\ 0\ 1\ 0\ 0\ 1)$ be the received vector. Upon receiving $\mathbf{r}$, the receiver computes the syndrome:

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (1 \quad 1 \quad 1).$$

Next, the receiver attempts to determine the true error vector $\mathbf{e} = (e_0, e_1, e_2, e_3, e_4, e_5, e_6)$, which yields the syndrome above. It follows from (3.12) or (3.13) that the error digits are related to the syndrome digits by the following linear equations:

$$1 = e_0 + e_3 + e_5 + e_6$$
$$1 = e_1 + e_3 + e_4 + e_5$$
$$1 = e_2 + e_4 + e_5 + e_6.$$

There are $2^4 = 16$ error patterns that satisfy the equations above. They are

$$\begin{array}{llll}
(0\ 0\ 0\ 0\ 0\ 1\ 0), & (1\ 0\ 1\ 0\ 0\ 1\ 1), \\
(1\ 1\ 0\ 1\ 0\ 1\ 0), & (0\ 1\ 1\ 1\ 0\ 1\ 1), \\
(0\ 1\ 1\ 0\ 1\ 1\ 0), & (1\ 1\ 0\ 0\ 1\ 1\ 1), \\
(1\ 0\ 1\ 1\ 1\ 1\ 0), & (0\ 0\ 0\ 1\ 1\ 1\ 1), \\
(1\ 1\ 1\ 0\ 0\ 0\ 0), & (0\ 1\ 0\ 0\ 0\ 0\ 1), \\
(0\ 0\ 1\ 1\ 0\ 0\ 0), & (1\ 0\ 0\ 1\ 0\ 0\ 1), \\
(1\ 0\ 0\ 0\ 1\ 0\ 0), & (0\ 0\ 1\ 0\ 1\ 0\ 1), \\
(0\ 1\ 0\ 1\ 1\ 0\ 0), & (1\ 1\ 1\ 1\ 1\ 0\ 1).
\end{array}$$

The error vector $\mathbf{e} = (0\ 0\ 0\ 0\ 0\ 1\ 0)$ has the smallest number of nonzero components. If the channel is a BSC, $\mathbf{e} = (0\ 0\ 0\ 0\ 0\ 1\ 0)$ is the most probable error vector that satisfies the equations above. Taking $\mathbf{e} = (0\ 0\ 0\ 0\ 0\ 1\ 0)$ as the true error vector, the receiver decodes the received vector $\mathbf{r} = (1\ 0\ 0\ 1\ 0\ 0\ 1)$ into the following code word:

$$\begin{aligned}
\mathbf{v}^* &= \mathbf{r} + \mathbf{e} \\
&= (1\ 0\ 0\ 1\ 0\ 0\ 1) + (0\ 0\ 0\ 0\ 0\ 1\ 0) \\
&= (1\ 0\ 0\ 1\ 0\ 1\ 1).
\end{aligned}$$

We see that $\mathbf{v}^*$ is the actual transmitted code word. Hence, the receiver has made a correct decoding. Later we show that the (7, 4) linear code considered in this example is capable of correcting any single error over a span of seven digits; that is, if a code word is transmitted and if only one digit is changed by the channel noise, the receiver will be able to determine the true error vector and to perform a correct decoding.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 17
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

More discussion on error correction based on syndrome is given in Section 3.5. Various methods of determining the true error pattern from the $n - k$ linear equations of (3.13) are presented in later chapters.

3.3 THE MINIMUM DISTANCE OF A BLOCK CODE

In this section an important parameter of a block code called the *minimum distance* is introduced. This parameter determines the random-error-detecting and random-error-correcting capabilities of a code. Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple. The *Hamming weight* (or simply weight) of $\mathbf{v}$, denoted by $w(\mathbf{v})$, is defined as the number of nonzero components of $\mathbf{v}$. For example, the Hamming weight of $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ is 4. Let $\mathbf{v}$ and $\mathbf{w}$ be two n-tuples. The *Hamming distance* (or simply distance) between $\mathbf{v}$ and $\mathbf{w}$, denoted $d(\mathbf{v}, \mathbf{w})$, is defined as the number of places where they differ. For example, the Hamming distance between $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ and $\mathbf{w} = (0\ 1\ 0\ 0\ 0\ 1\ 1)$ is 3; they differ in the zeroth, first, and third places. The Hamming distance is a metric function that satisfies the *triangle inequality*. Let $\mathbf{v}$, $\mathbf{w}$, and $\mathbf{x}$ be three n-tuples. Then

$$d(\mathbf{v}, \mathbf{w}) + d(\mathbf{w}, \mathbf{x}) \geq d(\mathbf{v}, \mathbf{x}). \tag{3.14}$$

(The proof of this inequality is left as a problem.) It follows from the definition of Hamming distance and the definition of modulo-2 addition that the Hamming distance between two n-tuples, $\mathbf{v}$ and $\mathbf{w}$, is equal to the Hamming weight of the sum of $\mathbf{v}$ and $\mathbf{w}$, that is,

$$d(\mathbf{v}, \mathbf{w}) = w(\mathbf{v} + \mathbf{w}). \tag{3.15}$$

For example, the Hamming distance between $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ and $\mathbf{w} = (1\ 1\ 1\ 0\ 0\ 1\ 0)$ is 4 and the weight of $\mathbf{v} + \mathbf{w} = (0\ 1\ 1\ 1\ 0\ 0\ 1)$ is also 4.

Given a block code C, one can compute the Hamming distance between any two distinct code words. The *minimum distance* of C, denoted $d_{\min}$, is defined as

$$d_{\min} = \min \{d(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\}. \tag{3.16}$$

If C is a linear block code, the sum of two vectors is also a code vector. It follows from (3.15) that the Hamming distance between two code vectors in C is equal to the Hamming weight of a third code vector in C. Then it follows from (3.16) that

$$\begin{aligned} d_{\min} &= \min \{w(\mathbf{v} + \mathbf{w}) : \mathbf{v}, \mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\} \\ &= \min \{w(\mathbf{x}) : \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\} \\ &\triangleq w_{\min}. \end{aligned} \tag{3.17}$$

The parameter $w_{\min} \triangleq \{w(\mathbf{x}) : \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\}$ is called the *minimum weight* of the linear code C. Summarizing the result above, we have the following theorem.

Theorem 3.1. The minimum distance of a linear block code is equal to the minimum weight of its nonzero code words.

Therefore, for a linear block code, to determine the minimum distance of the code is equivalent to determining its minimum weight. The (7, 4) code given in Table 3.1 has minimum weight 3; thus, its minimum distance is 3. Next, we prove a number

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 18
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

of theorems that relate the weight structure of a linear block code to its parity-check matrix.

Theorem 3.2. Let C be an (n, k) linear code with parity-check matrix $\mathbf{H}$. For each code vector of Hamming weight l, there exist l columns of $\mathbf{H}$ such that the vector sum of these l columns is equal to the zero vector. Conversely, if there exist l columns of $\mathbf{H}$ whose vector sum is the zero vector, there exists a code vector of Hamming weight l in C.

Proof. Let us express the parity-check matrix in the following form:

$$\mathbf{H} = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{n-1}],$$

where $\mathbf{h}_i$ represents the ith column of $\mathbf{H}$. Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a code vector of weight l. Then $\mathbf{v}$ has l nonzero components. Let $v_{i_1}, v_{i_2}, \ldots, v_{i_l}$ be the l nonzero components of $\mathbf{v}$, where $0 \leq i_1 < i_2 < \cdots < i_l \leq n-1$. Then $v_{i_1} = v_{i_2} = \cdots = v_{i_l} = 1$. Since $\mathbf{v}$ is a code vector, we must have

$$\begin{aligned}\mathbf{0} &= \mathbf{v} \cdot \mathbf{H}^T \\ &= v_0\mathbf{h}_0 + v_1\mathbf{h}_1 + \cdots + v_{n-1}\mathbf{h}_{n-1} \\ &= v_{i_1}\mathbf{h}_{i_1} + v_{i_2}\mathbf{h}_{i_2} + \cdots + v_{i_l}\mathbf{h}_{i_l} \\ &= \mathbf{h}_{i_1} + \mathbf{h}_{i_2} + \cdots + \mathbf{h}_{i_l}.\end{aligned}$$

This proves the first part of the theorem.

Now suppose that $\mathbf{h}_{i_1}, \mathbf{h}_{i_2}, \ldots, \mathbf{h}_{i_l}$ are l columns of $\mathbf{H}$ such that

$$\mathbf{h}_{i_1} + \mathbf{h}_{i_2} + \cdots + \mathbf{h}_{i_l} = \mathbf{0}. \tag{3.18}$$

Let us form a binary n-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_{n-1})$ whose nonzero components are $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$. The Hamming weight of $\mathbf{x}$ is l. Consider the product

$$\begin{aligned}\mathbf{x} \cdot \mathbf{H}^T &= x_0\mathbf{h}_0 + x_1\mathbf{h}_1 + \cdots + x_{n-1}\mathbf{h}_{n-1} \\ &= x_{i_1}\mathbf{h}_{i_1} + x_{i_2}\mathbf{h}_{i_2} + \cdots + x_{i_l}\mathbf{h}_{i_l} \\ &= \mathbf{h}_{i_1} + \mathbf{h}_{i_2} + \cdots + \mathbf{h}_{i_l}.\end{aligned}$$

It follows from (3.18) that $\mathbf{x} \cdot \mathbf{H}^T = \mathbf{0}$. Thus, $\mathbf{x}$ is a code vector of weight l in C. This proves the second part of the theorem. Q.E.D.

It follows from Theorem 3.2 that we have the following two corollaries.

Corollary 3.2.1. Let C be a linear block code with parity-check matrix $\mathbf{H}$. If no $d-1$ or fewer columns of $\mathbf{H}$ add to $\mathbf{0}$, the code has minimum weight at least d.

Corollary 3.2.2. Let C be a linear code with parity-check matrix $\mathbf{H}$. The minimum weight (or the minimum distance) of C is equal to the smallest number of columns of $\mathbf{H}$ that sum to $\mathbf{0}$.

Consider the (7, 4) linear code given in Table 3.1. The parity-check matrix of this code is

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

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 19
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

We see that all columns of **H** are nonzero and that no two of them are alike. Therefore, no two or fewer columns sum to **0**. Hence, the minimum weight of this code is at least 3. However, the zeroth, second and sixth columns sum to **0**. Thus, the minimum weight of the code is 3. From Table 3.1 we see that the minimum weight of the code is indeed 3. It follows from Theorem 3.1 that the minimum distance is 3.

Corollaries 3.2.1 and 3.2.2 are generally used to determine the minimum distance or to establish a lower bound on the minimum distance of a linear block code.

3.4 ERROR-DETECTING AND ERROR-CORRECTING CAPABILITIES OF A BLOCK CODE

When a code vector $\mathbf{v}$ is transmitted over a noisy channel, an error pattern of l errors will result in a received vector $\mathbf{r}$ which differs from the transmitted vector $\mathbf{v}$ in l places [i.e., $d(\mathbf{v}, \mathbf{r}) = l$]. If the minimum distance of a block code C is $d_{\min}$, any two distinct code vectors of C differ in at least $d_{\min}$ places. For this code C, no error pattern of $d_{\min} - 1$ or fewer errors can change one code vector into another. Therefore, any error pattern of $d_{\min} - 1$ or fewer errors will result in a received vector $\mathbf{r}$ that is not a code word in C. When the receiver detects that the received vector is not a code word of C, we say that errors are detected. Hence, a block code with minimum distance $d_{\min}$ is capable of detecting all the error patterns of $d_{\min} - 1$ or fewer errors. However, it cannot detect all the error patterns of $d_{\min}$ errors because there exists at least one pair of code vectors that differ in $d_{\min}$ places and there is an error pattern of $d_{\min}$ errors that will carry one into the other. The same argument applies to error patterns of more than $d_{\min}$ errors. For this reason, we say that the random-error-detecting capability of a block code with minimum distance $d_{\min}$ is $d_{\min} - 1$.

Even though a block code with minimum distance $d_{\min}$ guarantees detecting all the error patterns of $d_{\min} - 1$ or fewer errors, it is also capable of detecting a large fraction of error patterns with $d_{\min}$ or more errors. In fact, an (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n. This can be shown as follows. Among the $2^n - 1$ possible nonzero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero code words. If any of these $2^k - 1$ error patterns occurs, it alters the transmitted code word $\mathbf{v}$ into another code word $\mathbf{w}$. Thus, $\mathbf{w}$ will be received and its syndrome is zero. In this case, the decoder accepts $\mathbf{w}$ as the transmitted code word and thus commits an incorrect decoding. Therefore, there are $2^k - 1$ *undetectable* error patterns. If an error pattern is not identical to a nonzero code word, the received vector $\mathbf{r}$ will not be a code word and the syndrome will not be zero. In this case, error will be detected. There are exactly $2^n - 2^k$ error patterns that are not identical to the code words of an (n, k) linear code. These $2^n - 2^k$ error patterns are *detectable* error patterns. For large n, $2^k - 1$ is in general much smaller than 2^n. Therefore, only a small fraction of error patterns pass through the decoder without being detected.

Let C be an (n, k) linear code. Let A_i be the number of code vectors of weight i in C. The numbers $A_0, A_1, \ldots, A_n$ are called the *weight distribution* of C. If C is used only for error detection on a BSC, the probability that the decoder fails to detect the presence of errors can be computed from the weight distribution of C. Let $P_u(E)$ denote the probability of an undetected error. Since an undetected error occurs only

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 20
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

when the error pattern is identical to a nonzero code vector of C,

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}, \tag{3.19}$$

where p is the transition probability of the BSC. If the minimum distance of C is $d_{\min}$, then A_1 to $A_{d_{\min}-1}$ are zero.

Consider the (7, 4) code given in Table 3.1. The weight distribution of this code is $A_0 = 1$, $A_1 = A_2 = 0$, $A_3 = 7$, $A_4 = 7$, $A_5 = A_6 = 0$, and $A_7 = 1$. The probability of an undetected error is

$$P_u(E) = 7p^3(1-p)^4 + 7p^4(1-p)^3 + p^7.$$

If $p = 10^{-2}$, this probability is approximately 7×10^{-6}. In other words, if 1 million code words are transmitted over a BSC with $p = 10^{-2}$, there are on the average seven erroneous code words passing through the decoder without being detected.

If a block code C with minimum distance $d_{\min}$ is used for random-error correction, one would like to know how many errors that the code is able to correct. The minimum distance $d_{\min}$ is either odd or even. Let t be a positive integer such that

$$2t + 1 \leq d_{\min} \leq 2t + 2. \tag{3.20}$$

Next, we show that the code C is capable of correcting all the error patterns of t or fewer errors. Let $\mathbf{v}$ and $\mathbf{r}$ be the transmitted code vector and the received vector, respectively. Let $\mathbf{w}$ be any other code vector in C. The Hamming distances among $\mathbf{v}$, $\mathbf{r}$, and $\mathbf{w}$ satisfy the triangle inequality:

$$d(\mathbf{v}, \mathbf{r}) + d(\mathbf{w}, \mathbf{r}) \geq d(\mathbf{v}, \mathbf{w}). \tag{3.21}$$

Suppose that an error pattern of t' errors occurs during the transmission of $\mathbf{v}$. Then the received vector $\mathbf{r}$ differs from $\mathbf{v}$ in t' places and therefore $d(\mathbf{v}, \mathbf{r}) = t'$. Since $\mathbf{v}$ and $\mathbf{w}$ are code vectors in C, we have

$$d(\mathbf{v}, \mathbf{w}) \geq d_{\min} \geq 2t + 1. \tag{3.22}$$

Combining (3.21) and (3.22) and using the fact that $d(\mathbf{v}, \mathbf{r}) = t'$, we obtain the following inequality:

$$d(\mathbf{w}, \mathbf{r}) \geq 2t + 1 - t'.$$

If $t' \leq t$, then

$$d(\mathbf{w}, \mathbf{r}) > t.$$

The inequality above says that if an error pattern of t or fewer errors occurs, the received vector $\mathbf{r}$ is closer (in Hamming distance) to the transmitted code vector $\mathbf{v}$ than to any other code vector $\mathbf{w}$ in C. For a BSC, this means that the conditional probability $P(\mathbf{r} \mid \mathbf{v})$ is greater than the conditional probability $P(\mathbf{r} \mid \mathbf{w})$ for $\mathbf{w} \neq \mathbf{v}$. Based on the maximum likelihood decoding scheme, $\mathbf{r}$ is decoded into $\mathbf{v}$, which is the actual transmitted code vector. This results in a correct decoding and thus errors are corrected.

On the other hand, the code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector. To show this, let $\mathbf{v}$ and $\mathbf{w}$ be two code vectors in C such that

$$d(\mathbf{v}, \mathbf{w}) = d_{\min}.$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 21
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be two error patterns that satisfy the following conditions:

(i) $\mathbf{e}_1 + \mathbf{e}_2 = \mathbf{v} + \mathbf{w}$.

(ii) $\mathbf{e}_1$ and $\mathbf{e}_2$ do not have nonzero components in common places.

Obviously, we have

$$w(\mathbf{e}_1) + w(\mathbf{e}_2) = w(\mathbf{v} + \mathbf{w}) = d(\mathbf{v}, \mathbf{w}) = d_{\min}. \tag{3.23}$$

Now suppose that $\mathbf{v}$ is transmitted and is corrupted by the error pattern $\mathbf{e}_1$. Then the received vector is

$$\mathbf{r} = \mathbf{v} + \mathbf{e}_1.$$

The Hamming distance between $\mathbf{v}$ and $\mathbf{r}$ is

$$d(\mathbf{v}, \mathbf{r}) = w(\mathbf{v} + \mathbf{r}) = w(\mathbf{e}_1). \tag{3.24}$$

The Hamming distance between $\mathbf{w}$ and $\mathbf{r}$ is

$$d(\mathbf{w}, \mathbf{r}) = w(\mathbf{w} + \mathbf{r}) = w(\mathbf{w} + \mathbf{v} + \mathbf{e}_1) = w(\mathbf{e}_2). \tag{3.25}$$

Now suppose that the error pattern $\mathbf{e}_1$ contains more than t errors [i.e., $w(\mathbf{e}_1) > t$]. Since $2t + 1 \leq d_{\min} \leq 2t + 2$, it follows from (3.23) that

$$w(\mathbf{e}_2) \leq t + 1.$$

Combining (3.24) and (3.25) and using the fact that $w(\mathbf{e}_1) > t$ and $w(\mathbf{e}_2) \leq t + 1$, we obtain the following inequality:

$$d(\mathbf{v}, \mathbf{r}) \geq d(\mathbf{w}, \mathbf{r}).$$

This inequality says that there exists an error pattern of l ($l > t$) errors which results in a received vector that is closer to an incorrect code vector than to the transmitted code vector. Based on the maximum likelihood decoding scheme, an incorrect decoding would be committed.

Summarizing the results above, a block code with minimum distance $d_{\min}$ *guarantees* correcting all the error patterns of $t = \lfloor (d_{\min} - 1)/2 \rfloor$ or fewer errors, where $\lfloor (d_{\min} - 1)/2 \rfloor$ denotes the largest integer no greater than $(d_{\min} - 1)/2$. The parameter $t = \lfloor (d_{\min} - 1)/2 \rfloor$ is called the *random-error-correcting capability* of the code. The code is referred to as a t-error-correcting code. The (7, 4) code given in Table 3.1 has minimum distance 3 and thus $t = 1$. It is capable of correcting any error pattern of single error over a block of seven digits.

A block code with random-error-correcting capability t is usually capable of correcting many error patterns of $t + 1$ or more errors. For a t-error-correcting (n, k) linear code, it is capable of correcting a total 2^{n-k} error patterns, including those with t or fewer errors (this will be seen in the next section). If a t-error-correcting block code is used strictly for error correction on a BSC with transition probability p, the probability that the decoder commits an erroneous decoding is upper bounded by

$$P(E) \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1 - p)^{n-i}. \tag{3.26}$$

In practice, a code is often used for correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors. That is, when λ or fewer errors occur, the

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 22
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

code is capable of correcting them; when more than λ but fewer than $l + 1$ errors occur, the code is capable of detecting their presence without making a decoding error. For this purpose, the minimum distance $d_{\min}$ of the code is at least $\lambda + l + 1$ (left as a problem). Thus, a block code with $d_{\min} = 10$ is capable of correcting three or fewer errors and simultaneously detecting six or fewer errors.

From the discussion above, we see that random-error-detecting and random-error-correcting capabilities of a block code are determined by the code's minimum distance. Clearly, for given n and k, one would like to construct a block code with minimum distance as large as possible, in addition to the implementation considerations.

3.5 STANDARD ARRAY AND SYNDROME DECODING

In this section a scheme for decoding linear block codes is presented. Let C be an (n, k) linear code. Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{2^k}$ be the code vectors of C. No matter which code vector is transmitted over a noisy channel, the received vector $\mathbf{r}$ may be any of the 2^n n-tuples over GF(2). Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \ldots, D_{2^k}$ such that the code vector $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$. Thus, each subset D_i is one-to-one correspondence to a code vector $\mathbf{v}_i$. If the received vector $\mathbf{r}$ is found in the subset D_i, $\mathbf{r}$ is decoded into $\mathbf{v}_i$. Correct decoding is made if and only if the received vector $\mathbf{r}$ is in the subset D_i that corresponds to the actual code vector transmitted.

A method to partition the 2^n possible received vectors into 2^k disjoint subsets such that each subset contains one and only one code vector is described here. The partition is based on the linear structure of the code. First, the 2^k code vectors of C are placed in a row with the all-zero code vector $\mathbf{v}_1 = (0, 0, \ldots, 0)$ as the first (left-most) element. From the remaining $2^n - 2^k$ n-tuples, an n-tuple $\mathbf{e}_2$ is chosen and is placed under the zero vector $\mathbf{v}_1$. Now, we form a second row by adding $\mathbf{e}_2$ to each code vector $\mathbf{v}_i$ in the first row and placing the sum $\mathbf{e}_2 + \mathbf{v}_i$ under $\mathbf{v}_i$. Having completed the second row, an unused n-tuple $\mathbf{e}_3$ is chosen from the remaining n-tuples and is placed under $\mathbf{v}_1$. Then a third row is formed by adding $\mathbf{e}_3$ to each code vector $\mathbf{v}_i$ in the first row and placing $\mathbf{e}_3 + \mathbf{v}_i$ under $\mathbf{v}_i$. We continue this process until all the n-tuples are used. Then we have an array of rows and columns as shown in Figure 3.6. This array is called a *standard array* of the given linear code C.

It follows from the construction rule of a standard array that the sum of any two vectors in the same row is a code vector in C. Next, we prove some important properties of a standard array.

Theorem 3.3. No two n-tuples in the same row of a standard array are identical. Every n-tuple appears in one and only one row.

Proof. The first part of the theorem follows from the fact that all the code vectors of C are distinct. Suppose that two n-tuples in the lth rows are identical, say $\mathbf{e}_l + \mathbf{v}_i = \mathbf{e}_l + \mathbf{v}_j$ with $i \neq j$. This means that $\mathbf{v}_i = \mathbf{v}_j$, which is impossible. Therefore, no two n-tuples in the same row are identical.

It follows from the construction rule of the standard array that every n-tuple appears at least once. Now suppose that an n-tuple appears in both lth row and the

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 23
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

$$
\begin{array}{lllll}
\mathbf{v}_1 = 0 & \mathbf{v}_2 & \cdots & \mathbf{v}_i & \cdots \quad \mathbf{v}_{2^k} \\
\mathbf{e}_2 & \mathbf{e}_2 + \mathbf{v}_2 & \cdots & \mathbf{e}_2 + \mathbf{v}_i & \cdots \quad \mathbf{e}_2 + \mathbf{v}_{2^k} \\
\mathbf{e}_3 & \mathbf{e}_3 + \mathbf{v}_2 & \cdots & \mathbf{e}_3 + \mathbf{v}_i & \cdots \quad \mathbf{e}_3 + \mathbf{v}_{2^k} \\
\vdots & & & & \vdots \\
\mathbf{e}_l & \mathbf{e}_l + \mathbf{v}_2 & \cdots & \mathbf{e}_l + \mathbf{v}_i & \cdots \quad \mathbf{e}_l + \mathbf{v}_{2^k} \\
\vdots & & & & \vdots \\
\mathbf{e}_{2^{n-k}} & \mathbf{e}_{2^{n-k}} + \mathbf{v}_2 & \cdots & \mathbf{e}_{2^{n-k}} + \mathbf{v}_i & \cdots \quad \mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}
\end{array}
$$

Figure 3.6 Standard array for an (n, k) linear code.

mth row with $l < m$. Then this n-tuple must be equal to $\mathbf{e}_l + \mathbf{v}_i$ for some i and equal to $\mathbf{e}_m + \mathbf{v}_j$ for some j. As a result, $\mathbf{e}_l + \mathbf{v}_i = \mathbf{e}_m + \mathbf{v}_j$. From this equality we obtain $\mathbf{e}_m = \mathbf{e}_l + (\mathbf{v}_i + \mathbf{v}_j)$. Since $\mathbf{v}_i$ and $\mathbf{v}_j$ are code vectors in C, $\mathbf{v}_i + \mathbf{v}_j$ is also a code vector in C, say $\mathbf{v}_s$. Then $\mathbf{e}_m = \mathbf{e}_l + \mathbf{v}_s$. This implies that the n-tuple $\mathbf{e}_m$ is in the lth row of the array, which contradicts the construction rule of the array that $\mathbf{e}_m$, the first element of the mth row, should be unused in any previous row. Therefore, no n-tuple can appear in more than one row of the array. This concludes the proof of the second part of the theorem. Q.E.D.

From Theorem 3.3 we see that there are $2^n/2^k = 2^{n-k}$ disjoint rows in the standard array, and that each row consists of 2^k distinct elements. The 2^{n-k} rows are called the *cosets* of the code C and the first n-tuple $\mathbf{e}_j$ of each coset is called a *coset leader*. Any element in a coset can be used as its coset leader. This does not change the elements of the coset; it simply permutes them.

Example 3.6

Consider the (6, 3) linear code generated by the following matrix:

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

The standard array of this code is shown in Figure 3.7.

A standard array of an (n, k) linear code C consists of 2^k disjoint columns. Each column consists of 2^{n-k} n-tuples with the topmost one as a code vector in C. Let D_j denote the jth column of the standard array. Then

$$
D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \ldots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}, \tag{3.27}
$$

where $\mathbf{v}_j$ is a code vector of C and $\mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_{2^{n-k}}$ are the coset leaders. The 2^k disjoint columns $D_1, D_2, \ldots, D_{2^k}$ can be used for decoding the code C as described earlier in this section. Suppose that the code vector $\mathbf{v}_j$ is transmitted over a noisy channel. From (3.27) we see that the received vector $\mathbf{r}$ is in D_j if the error pattern caused by the channel is a coset leader. In this event, the received vector $\mathbf{r}$ will be decoded

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 24
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

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

Figure 3.7 Standard array for the (6, 3) code.

correctly into the transmitted code vector $\mathbf{v}_j$. On the other hand, if the error pattern caused by the channel is not a coset leader, an erroneous decoding will result. This can be seen as follows. The error pattern $\mathbf{x}$ caused by the channel must be in some coset and under some nonzero code vector, say in the lth coset and under the code vector $\mathbf{v}_i \neq \mathbf{0}$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + (\mathbf{v}_i + \mathbf{v}_j) = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector $\mathbf{r}$ is thus in D_s and is decoded into $\mathbf{v}_s$, which is not the transmitted code vector. This results in an erroneous decoding. Therefore, the decoding is correct if and only if the error pattern caused by the channel is a coset leader. For this reason, the 2^{n-k} coset leaders (including the zero vector $\mathbf{0}$) are called the *correctable error patterns*. Summarizing the results above, we have the following theorem:

Theorem 3.4. Every (n, k) linear block code is capable of correcting 2^{n-k} error patterns.

To minimize the probability of a decoding error, the error patterns that are most likely to occur for a given channel should be chosen as the coset leaders. For a BSC, an error pattern of smaller weight is more probable than an error pattern of larger weight. Therefore, when a standard array is formed, each coset leader should be chosen to be a vector of *least weight* from the remaining available vectors. Choosing coset leaders in this manner, each coset leader has minimum weight in its coset. As a result, the decoding based on the standard array is the minimum distance decoding (i.e., the maximum likelihood decoding). To see this, let $\mathbf{r}$ be the received vector. Suppose that $\mathbf{r}$ is found in the ith column D_i and lth coset of the standard array. Then $\mathbf{r}$ is decoded into the code vector $\mathbf{v}_i$. Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, the distance between $\mathbf{r}$ and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l). \tag{3.28}$$

Now, consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j).$$

Since $\mathbf{v}_i$ and $\mathbf{v}_j$ are two different code vectors, their vector sum, $\mathbf{v}_i + \mathbf{v}_j$, is a nonzero code vector, say $\mathbf{v}_s$. Thus,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_s). \tag{3.29}$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 25
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

Since, $\mathbf{e}_l$ and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows from (3.28) and (3.29) that

$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j).$$

This says that the received vector is decoded into a closest code vector. Hence, if each coset leader is chosen to have minimum weight in its coset, the decoding based on the standard array is the minimum distance decoding or MLD.

Let α_i denote the number of coset leaders of weight i. The numbers $\alpha_0, \alpha_1, \ldots, \alpha_n$ are called the *weight distribution* of the coset leaders. Knowing these numbers, we can compute the probability of a decoding error. Since a decoding error occurs if and only if the error pattern is not a coset leader, the error probability for a BSC with transition probability p is

$$P(E) = 1 - \sum_{i=0}^{n} \alpha_i p^i (1 - p)^{n-i}. \tag{3.30}$$

Example 3.7

Consider the (6, 3) code given in Example 3.6. The standard array for this code is shown in Figure 3.7. The weight distribution of the coset leaders is $\alpha_0 = 1$, $\alpha_1 = 6$, $\alpha_2 = 1$, and $\alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 0$. Thus,

$$P(E) = 1 - (1 - p)^6 - 6p(1 - p)^5 - p^2(1 - p)^4.$$

For $p = 10^{-2}$, we have $P(E) \approx 1.37 \times 10^{-3}$.

An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns; however, it is capable of correcting only 2^{n-k} error patterns. For large n, 2^{n-k} is a small fraction of $2^n - 2^k$. Therefore, the probability of a decoding error is much higher than the probability of an undetected error.

Theorem 3.5. For an (n, k) linear code C with minimum distance $d_{\min}$, all the n-tuples of weight of $t = \lfloor (d_{\min} - 1)/2 \rfloor$ or less can be used as coset leaders of a standard array of C. If all the n-tuples of weight t or less are used as coset leaders, there is at least one n-tuple of weight $t + 1$ that cannot be used as a coset leader.

Proof. Since the minimum distance of C is $d_{\min}$, the minimum weight of C is also $d_{\min}$. Let $\mathbf{x}$ and $\mathbf{y}$ be two n-tuples of weight t or less. Clearly, the weight of $\mathbf{x} + \mathbf{y}$ is

$$w(\mathbf{x} + \mathbf{y}) \leq w(\mathbf{x}) + w(\mathbf{y}) \leq 2t < d_{\min}.$$

Suppose that $\mathbf{x}$ and $\mathbf{y}$ are in the same coset; then $\mathbf{x} + \mathbf{y}$ must be a nonzero code vector in C. This is impossible because the weight of $\mathbf{x} + \mathbf{y}$ is less than the minimum weight of C. Therefore, no two n-tuples of weight t or less can be in the same coset of C, and all the n-tuples of weight t or less can be used as coset leaders.

Let $\mathbf{v}$ be a minimum weight code vector of C [i.e., $w(\mathbf{v}) = d_{\min}$]. Let $\mathbf{x}$ and $\mathbf{y}$ be two n-tuples which satisfy the following two conditions:

(i) $\mathbf{x} + \mathbf{y} = \mathbf{v}$.

(ii) $\mathbf{x}$ and $\mathbf{y}$ do not have nonzero components in common places.

It follows from the definition that $\mathbf{x}$ and $\mathbf{y}$ must be in the same coset and

$$w(\mathbf{x}) + w(\mathbf{y}) = w(\mathbf{v}) = d_{\min}.$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 26
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

Suppose we choose $\mathbf{y}$ such that $w(\mathbf{y}) = t + 1$. Since $2t + 1 \leq d_{\min} \leq 2t + 2$, we have $w(\mathbf{x}) = t$ or $t + 1$. If $\mathbf{x}$ is used as a coset leader, then $\mathbf{y}$ cannot be a coset leader. Q.E.D.

Theorem 3.5 reconfirms the fact that an (n, k) linear code with minimum distance $d_{\min}$ is capable of correcting all the error patterns of $\lfloor (d_{\min} - 1)/2 \rfloor$ or fewer errors, but it is not capable of correcting all the error patterns of weight $t + 1$.

A standard array has an important property that can be used to simplify the decoding process. Let $\mathbf{H}$ be the parity-check matrix of the given (n, k) linear code C.

Theorem 3.6. All the 2^k n-tuples of a coset have the same syndrome. The syndromes for different cosets are different.

Proof. Consider the coset whose coset leader is $\mathbf{e}_l$. A vector in this coset is the sum of $\mathbf{e}_l$ and some code vector $\mathbf{v}_i$ in C. The syndrome of this vector is

$$(\mathbf{e}_l + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_l\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_l\mathbf{H}^T$$

(since $\mathbf{v}_i\mathbf{H}^T = \mathbf{0}$). The equality above says that the syndrome of any vector in a coset is equal to the syndrome of the coset leader. Therefore, all the vectors of a coset have the same syndrome.

Let $\mathbf{e}_j$ and $\mathbf{e}_l$ be the coset leaders of the jth and lth cosets, respectively, where $j < l$. Suppose that the syndromes of these two cosets are equal. Then

$$\mathbf{e}_j\mathbf{H}^T = \mathbf{e}_l\mathbf{H}^T,$$

$$(\mathbf{e}_j + \mathbf{e}_l)\mathbf{H}^T = \mathbf{0}.$$

This implies that $\mathbf{e}_j + \mathbf{e}_l$ is a code vector in C, say $\mathbf{v}_i$. Thus, $\mathbf{e}_j + \mathbf{e}_l = \mathbf{v}_i$ and $\mathbf{e}_l = \mathbf{e}_j + \mathbf{v}_i$. This implies that $\mathbf{e}_l$ is in the jth coset, which contradicts the construction rule of a standard array that a coset leader should be previously unused. Therefore, no two cosets have the same syndrome. Q.E.D.

We recall that the syndrome of an n-tuple is an $(n - k)$-tuple and there are 2^{n-k} distinct $(n - k)$-tuples. It follows from Theorem 3.6 that there is a one-to-one correspondence between a coset and an $(n - k)$-tuple syndrome. Or, there is a one-to-one correspondence between a coset leader (a correctable error pattern) and a syndrome. Using this one-to-one correspondence relationship, we can form a decoding table, which is much simpler to use than a standard array. The table consists of 2^{n-k} coset leaders (the correctable error patterns) and their corresponding syndromes. This table is either stored or wired in the receiver. The decoding of a received vector consists of three steps:

Step 1. Compute the syndrome of $\mathbf{r}$, $\mathbf{r} \cdot \mathbf{H}^T$.

Step 2. Locate the coset leader $\mathbf{e}_l$ whose syndrome is equal to $\mathbf{r} \cdot \mathbf{H}^T$. Then $\mathbf{e}_l$ is assumed to be the error pattern caused by the channel.

Step 3. Decode the received vector $\mathbf{r}$ into the code vector $\mathbf{v} = \mathbf{r} + \mathbf{e}_l$.

The decoding scheme described above is called the *syndrome decoding* or *table-lookup decoding*. In principle, table-lookup decoding can be applied to any (n, k) linear code. It results in minimum decoding delay and minimum error probability. However,

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 27
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

for large $n - k$, the implementation of this decoding scheme becomes impractical, and either a large storage or a complicated logic circuitry is needed. Several practical decoding schemes which are variations of table-lookup decoding are discussed in subsequent chapters. Each of these decoding schemes requires additional properties in a code other than the linear structure.

Example 3.8

Consider the (7, 4) linear code given in Table 3.1. The parity-check matrix, as given in Example 3.3, is

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

The code has $2^3 = 8$ cosets and, therefore, there are eight correctable error patterns (including the all-zero vector). Since the minimum distance of the code is 3, it is capable of correcting all the error patterns of weight 1 or 0. Hence, all the 7-tuples of weight 1 or 0 can be used as coset leaders. There are $\binom{7}{0} + \binom{7}{1} = 8$ such vectors. We see that, for the (7, 4) linear code considered in this example, the number of correctable error patterns guaranteed by the minimum distance is equal to the total number of correctable error patterns. The correctable error patterns and their corresponding syndromes are given in Table 3.2.

TABLE 3.2 DECODING TABLE FOR THE (7, 4) LINEAR CODE GIVEN IN TABLE 3.1

Syndrome	Coset leaders
(1 0 0)	(1 0 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0 0)
(1 1 0)	(0 0 0 1 0 0 0)
(0 1 1)	(0 0 0 0 1 0 0)
(1 1 1)	(0 0 0 0 0 1 0)
(1 0 1)	(0 0 0 0 0 0 1)

Suppose that the code vector $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ is transmitted and $\mathbf{r} = (1\ 0\ 0\ 1\ 1\ 1\ 1)$ is received. For decoding $\mathbf{r}$, we compute the syndrome of $\mathbf{r}$,

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

From Table 3.2 we find that (0 1 1) is the syndrome of the coset leader $\mathbf{e} = (0\ 0\ 0\ 0\ 1\ 0\ 0)$. Thus, (0 0 0 0 1 0 0) is assumed to be the error pattern caused by the channel, and $\mathbf{r}$ is decoded into

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 28
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

$$\begin{aligned}\mathbf{v}^* &= \mathbf{r} + \mathbf{e} \\ &= (1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1) + (0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0) \\ &= (1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1),\end{aligned}$$

which is the actual code vector transmitted. The decoding is correct since the error pattern caused by the channel is a coset leader.

Now suppose that $\mathbf{v} = (0\ 0\ 0\ 0\ 0\ 0\ 0)$ is transmitted and $\mathbf{r} = (1\ 0\ 0\ 0\ 1\ 0\ 0)$ is received. We see that two errors have occurred during the transmission of $\mathbf{v}$. The error pattern is not correctable and will cause a decoding error. When $\mathbf{r}$ is received, the receiver computes the syndrome

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (1\ \ 1\ \ 1).$$

From the decoding table we find that the coset leader $\mathbf{e} = (0\ 0\ 0\ 0\ 0\ 1\ 0)$ corresponds to the syndrome $\mathbf{s} = (1\ 1\ 1)$. As a result, $\mathbf{r}$ is decoded into the code vector

$$\begin{aligned}\mathbf{v}^* &= \mathbf{r} + \mathbf{e} \\ &= (1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0) + (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0) \\ &= (1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0).\end{aligned}$$

Since $\mathbf{v}^*$ is not the actual code vector transmitted, a decoding error is committed.

Using Table 3.2, the code is capable of correcting any single error over a block of seven digits. When two or more errors occur, a decoding error will be committed.

The table-lookup decoding of an (n, k) linear code may be implemented as follows. The decoding table is regarded as the truth table of n switching functions:

$$\begin{aligned} e_0 &= f_0(s_0, s_1, \ldots, s_{n-k-1}), \\ e_1 &= f_1(s_0, s_1, \ldots, s_{n-k-1}), \\ &\vdots \\ e_{n-1} &= f_{n-1}(s_0, s_1, \ldots, s_{n-k-1}), \end{aligned}$$

where $s_0, s_1, \ldots, s_{n-k-1}$ are the syndrome digits, which are regarded as switching variables, and $e_0, e_1, \ldots, e_{n-1}$ are the estimated error digits. When these n switching functions are derived and simplified, a combinational logic circuit with the $n - k$ syndrome digits as inputs and the estimated error digits as outputs can be realized. The implementation of the syndrome circuit has been discussed in Section 3.2. The general decoder for an (n, k) linear code based on the table-lookup scheme is shown in Figure 3.8. The cost of this decoder depends primarily on the complexity of the combinational logic circuit.

Example 3.9

Again, we consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Figure 3.5. The decoding table is given by Table 3.2. From this table we form the truth table (Table 3.3). The switching expressions for the seven error digits are

$$\begin{aligned} e_0 &= s_0 \Lambda s_1' \Lambda s_2', & e_1 &= s_0' \Lambda s_1 \Lambda s_2', \\ e_2 &= s_0' \Lambda s_1' \Lambda s_2, & e_3 &= s_0 \Lambda s_1 \Lambda s_2', \\ e_4 &= s_0' \Lambda s_1 \Lambda s_2, & e_5 &= s_0 \Lambda s_1 \Lambda s_2, \\ e_6 &= s_0 \Lambda s_1' \Lambda s_2, \end{aligned}$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 29
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

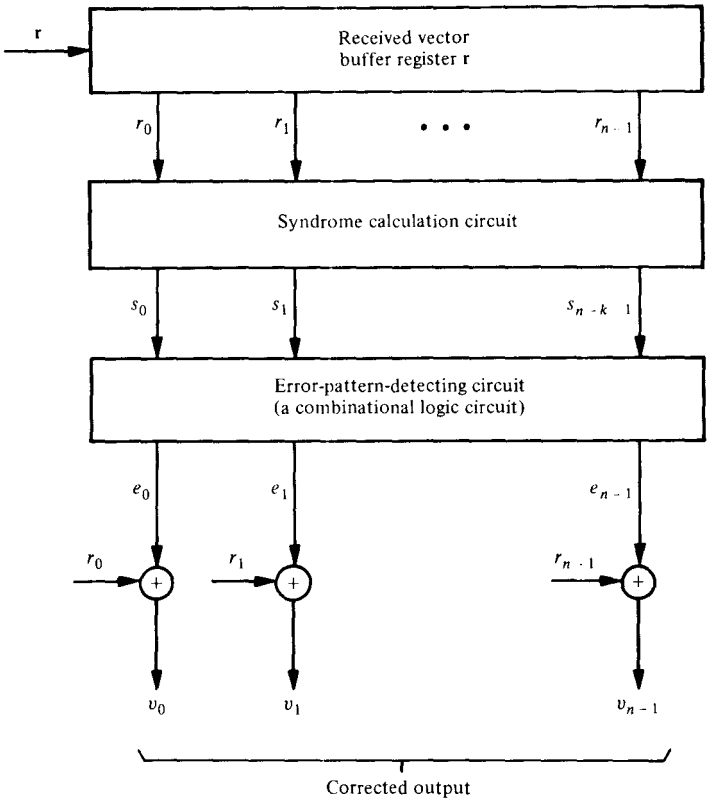


Figure 3.8 General decoder for a linear block code.

TABLE 3.3 TRUTH TABLE FOR THE ERROR DIGITS OF THE CORRECTABLE ERROR PATTERNS OF THE (7, 4) LINEAR CODE GIVEN IN TABLE 3.1

Syndromes			Correctable error patterns (coset leaders)						
s_0	s_1	s_2	e_0	e_1	e_2	e_3	e_4	e_5	e_6
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
1	1	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1

where Λ denotes the logic-AND operation and s' denotes the logic-COMPLEMENT of s. These seven switching expressions can be realized by seven 3-input AND gates. The complete circuit of the decoder is shown in Figure 3.9.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 30
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

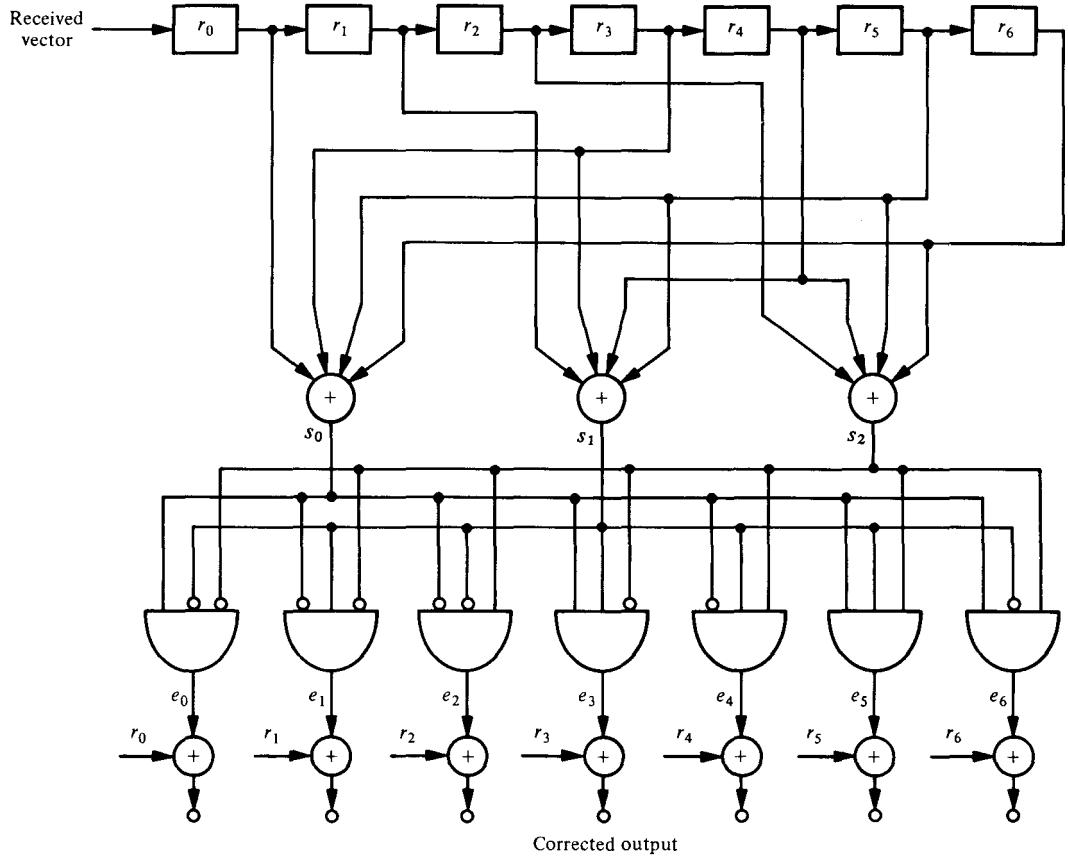


Figure 3.9 Decoding circuit for the (7, 4) code given in Table 3.1.

3.6 PROBABILITY OF AN UNDETECTED ERROR FOR LINEAR CODES OVER A BSC

If an (n, k) linear code is used only for error detection over a BSC, the probability of an undetected error, $P_u(E)$, can be computed from (3.19) if the weight distribution of the code is known. There exists an interesting relationship between the weight distribution of a linear code and the weight distribution of its dual code. This relationship often makes the computation of $P_u(E)$ much easier. Let $\{A_0, A_1, \ldots, A_n\}$ be the weight distribution of an (n, k) linear code C and let $\{B_0, B_1, \ldots, B_n\}$ be the weight distribution of its dual code C_d. Now we represent these two weight distributions in polynomial form as follows:

$$\begin{aligned} A(z) &= A_0 + A_1 z + \cdots + A_n z^n, \\ B(z) &= B_0 + B_1 z + \cdots + B_n z^n. \end{aligned} \tag{3.31}$$

Then $A(z)$ and $B(z)$ are related by the following identity:

$$A(z) = 2^{-(n-k)}(1+z)^n B\left(\frac{1-z}{1+z}\right). \tag{3.32}$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 31
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

This identity is known as the *MacWilliams identity* [5]. The polynomials $A(z)$ and $B(z)$ are called the *weight enumerators* for the (n, k) linear code C and its dual C_d. From the MacWilliams identity, we see that if the weight distribution of the dual of a linear code is known, the weight distribution of the code itself can be determined. As a result, this gives us more flexibility of computing the weight distribution of a linear code.

Using the MacWilliams identity, we can compute the probability of an undetected error for an (n, k) linear code from the weight distribution of its dual. First, we put the expression of (3.19) into the following form:

$$\begin{aligned} P_u(E) &= \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \\ &= (1-p)^n \sum_{i=1}^{n} A_i \left(\frac{p}{1-p}\right)^i. \end{aligned} \tag{3.33}$$

Substituting $z = p/(1-p)$ in $A(z)$ of (3.31) and using the fact that $A_0 = 1$, we obtain the following identity:

$$A\left(\frac{p}{1-p}\right) - 1 = \sum_{i=1}^{n} A_i \left(\frac{p}{1-p}\right)^i. \tag{3.34}$$

Combining (3.33) and (3.34), we have the following expression for the probability of an undetected error:

$$P_u(E) = (1-p)^n \left[A\left(\frac{p}{1-p}\right) - 1 \right]. \tag{3.35}$$

From (3.35) and the MacWilliams identity of (3.32), we finally obtain the following expression for $P_u(E)$:

$$P_u(E) = 2^{-(n-k)} B(1-2p) - (1-p)^n, \tag{3.36}$$

where

$$B(1-2p) = \sum_{i=0}^{n} B_i (1-2p)^i.$$

Hence, there are two ways for computing the probability of an undetected error for a linear code; often one is easier than the other. If $n - k$ is smaller than k, it is much easier to compute $P_u(E)$ from (3.36); otherwise, it is easier to use (3.35).

Example 3.10

Consider the (7, 4) linear code given in Table 3.1. The dual of this code is generated by its parity-check matrix,

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

(see Example 3.3). Taking the linear combinations of the rows of $\mathbf{H}$, we obtain the following eight vectors in the dual code:

$$\begin{array}{llll} (0\ 0\ 0\ 0\ 0\ 0\ 0), & & (1\ 1\ 0\ 0\ 1\ 0\ 1), \\ (1\ 0\ 0\ 1\ 0\ 1\ 1), & & (1\ 0\ 1\ 1\ 1\ 0\ 0), \\ (0\ 1\ 0\ 1\ 1\ 1\ 0), & & (0\ 1\ 1\ 1\ 0\ 0\ 1), \\ (0\ 0\ 1\ 0\ 1\ 1\ 1), & & (1\ 1\ 1\ 0\ 0\ 1\ 0). \end{array}$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 32
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

Thus, the weight enumerator for the dual code is $B(z) = 1 + 7z^4$. Using (3.36), we obtain the probability of an undetected error for the (7, 4) linear code given in Table 3.1,

$$P_u(E) = 2^{-3}[1 + 7(1 - 2p)^4] - (1 - p)^7.$$

This probability was also computed in Section 3.4 using the weight distribution of the code itself.

Theoretically, we can compute the weight distribution of an (n, k) linear code by examining its 2^k code words or by examining the 2^{n-k} code words of its dual and then applying the MacWilliams identity. However, for large n, k, and $n - k$, the computation becomes practically impossible. Except for some short linear codes and a few small classes of linear codes, the weight distributions for many known linear codes are still unknown. Consequently, it is very difficult, if not impossible, to compute their probability of an undetected error.

Although it is difficult to compute the probability of an undetected error for a specific (n, k) linear code for large n and k, it is quite easy to derive an upper bound on the average probability of an undetected error for the ensemble of all (n, k) linear systematic codes. As we have shown earlier, an (n, k) linear systematic code is completely specified by a matrix $\mathbf{G}$ of the form given by (3.4). The submatrix $\mathbf{P}$ consists of $k(n - k)$ entries. Since each entry p_{ij} can be either a 0 or a 1, there are $2^{k(n-k)}$ distinct matrices $\mathbf{G}$'s of the form given by (3.4). Let Γ denote the ensemble of codes generated by these $2^{k(n-k)}$ matrices. Suppose that we choose a code randomly from Γ and use it for error detection. Let C_j be the chosen code. Then the probability of C_j being chosen is

$$P(C_j) = 2^{-k(n-k)}. \tag{3.37}$$

Let A_{ji} denote the number of code words in C_j with weight i. It follows from (3.19) that probability of an undetected error for C_j is given by

$$P_u(E \mid C_j) = \sum_{i=1}^{n} A_{ji} p^i (1 - p)^{n-i}. \tag{3.38}$$

The average probability of an undetected error for a linear code in Γ is defined as

$$\mathbf{P_u(E)} = \sum_{j=1}^{|\Gamma|} P(C_j) P_u(E \mid C_j), \tag{3.39}$$

where $|\Gamma|$ denotes the number of codes in Γ. Substituting (3.37) and (3.38) into (3.39), we obtain

$$\mathbf{P_u(E)} = 2^{-k(n-k)} \sum_{i=1}^{n} p^i (1 - p)^{n-i} \sum_{j=1}^{|\Gamma|} A_{ji}. \tag{3.40}$$

A nonzero n-tuple is either contained in exactly $2^{(k-1)(n-k)}$ codes in Γ or contained in none of the codes (left as a problem). Since there are $\binom{n}{i}$ n-tuples of weight i, we have

$$\sum_{j=1}^{|\Gamma|} A_{ji} \leq \binom{n}{i} 2^{(k-1)(n-k)}. \tag{3.41}$$

Substituting (3.41) into (3.40), we obtain the following upper bound on the average probability of an undetected error for an (n, k) linear systematic code:

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 33
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

$$\begin{aligned} \mathbf{P}_u(\mathbf{E}) &\leq 2^{-(n-k)} \sum_{i=1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \\ &= 2^{-(n-k)}[1-(1-p)^n]. \end{aligned} \tag{3.42}$$

Since $[1-(1-p)^n] \leq 1$, it is clear that $\mathbf{P}_u(\mathbf{E}) \leq 2^{-(n-k)}$.

The result above says that there exist (n, k) linear codes with probability of an undetected error, $P_u(E)$, upper bounded by $2^{-(n-k)}$. In other words, there exist (n, k) linear codes with $P_u(E)$ *decreasing exponentially with the number of parity-check digits*, $n-k$. Even for moderate $n-k$, these codes have a very small probability of an undetected error. For example, let $n-k=30$. There exist (n, k) linear codes for which $P_u(E)$ is upper bounded by $2^{-30} \approx 10^{-9}$. Many classes of linear codes have been constructed for the past three decades. However, only a few small classes of linear codes have been proved to have $P_u(E)$ satisfying the upper bound $2^{-(n-k)}$. It is still not known whether the other known linear codes satisfy this upper bound. A class of linear codes that satisfies this upper bound is presented in the next section. Other codes with probability of an undetected error decreasing exponentially with $n-k$ are presented in subsequent chapters.

3.7 HAMMING CODES

Hamming codes are the first class of linear codes devised for error correction [6]. These codes and their variations have been widely used for error control in digital communication and data storage systems.

For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters:

Code length:	$n = 2^m - 1$
Number of information symbols:	$k = 2^m - m - 1$
Number of parity-check symbols:	$n - k = m$
Error-correcting capability:	$t = 1 (d_{\min} = 3)$.

The parity-check matrix $\mathbf{H}$ of this code consists of all the nonzero m-tuples as its columns. In systematic form, the columns of $\mathbf{H}$ are arranged in the following form:

$$\mathbf{H} = [\mathbf{I}_m \quad \mathbf{Q}],$$

where $\mathbf{I}_m$ is an $m \times m$ identity matrix and the submatrix $\mathbf{Q}$ consists of $2^m - m - 1$ columns which are the m-tuples of weight 2 or more. For example, let $m = 3$. The parity-check matrix of a Hamming code of length 7 can be put in the form

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

which is the parity-check matrix of the (7, 4) linear code given in Table 3.1 (see Example 3.3). Hence, the code given in Table 3.1 is a Hamming code. The columns of $\mathbf{Q}$ may be arranged in any order without affecting the distance property and weight

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 34
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

distribution of the code. In systematic form, the generator matrix of the code is

$$\mathbf{G} = [\mathbf{Q}^T \quad \mathbf{I}_{2^m-m-1}],$$

where $\mathbf{Q}^T$ is the transpose of $\mathbf{Q}$ and $\mathbf{I}_{2^m-m-1}$ is an $(2^m - m - 1) \times (2^m - m - 1)$ identity matrix.

Since the columns of $\mathbf{H}$ are nonzero and distinct, no two columns add to zero. It follows from Corollary 3.2.1 that the minimum distance of a Hamming code is at least 3. Since $\mathbf{H}$ consists of all the nonzero m-tuples as its columns, the vector sum of any two columns, say $\mathbf{h}_i$ and $\mathbf{h}_j$, must also be a column in $\mathbf{H}$, say $\mathbf{h}_l$. Thus,

$$\mathbf{h}_i + \mathbf{h}_j + \mathbf{h}_l = \mathbf{0}.$$

It follows from Corollary 3.2.2 that the minimum distance of a Hamming code is exactly 3. Hence, the code is capable of correcting all the error patterns with a single error or of detecting all the error patterns of two or fewer errors.

If we form the standard array for the Hamming code of length $2^m - 1$, all the $(2^m - 1)$-tuples of weight 1 can be used as coset leaders (Theorem 3.5). The number of $(2^m - 1)$-tuples of weight 1 is $2^m - 1$. Since $n - k = m$, the code has 2^m cosets. Thus, the zero vector $\mathbf{0}$ and the $(2^m - 1)$-tuples of weight 1 form all the coset leaders of the standard array. This says that a Hamming code corrects only the error patterns of single error and no others. This is a very interesting structure. A t-error-correcting code is called a *perfect code* if its standard array has all the error patterns of t or fewer errors and no others as coset leaders. Thus, Hamming codes form a class of single-error-correcting perfect codes. Perfect codes are rare [3]. Besides the Hamming codes, the only other nontrivial binary perfect code is the (23, 12) Golay code (see Section 5.3).

Decoding of Hamming codes can be accomplished easily with the table-lookup scheme described in Section 3.5. The decoder for a Hamming code of length $2^m - 1$ can be implemented in the same manner as that for the (7,4) Hamming code given in Example 3.9.

We may delete any l columns from the parity-check matrix $\mathbf{H}$ of a Hamming code. This deletion results in an $m \times (2^m - l - 1)$ matrix $\mathbf{H}'$. Using $\mathbf{H}'$ as a parity-check matrix, we obtain a shortened Hamming code with the following parameters:

Code length:	$n = 2^m - l - 1$
Number of information symbols:	$k = 2^m - m - l - 1$
Number of parity-check symbols:	$n - k = m$
Minimum distance:	$d_{\min} \geq 3$.

If we delete columns from $\mathbf{H}$ properly, we may obtain a shortened Hamming code with minimum distance 4. For example, if we delete from the submatrix $\mathbf{Q}$ all the columns of even weight, we obtain an $m \times 2^{m-1}$ matrix

$$\mathbf{H}' = [\mathbf{I}_m \quad \mathbf{Q}'],$$

where $\mathbf{Q}'$ consists of $2^{m-1} - m$ columns of odd weight. Since all the columns of $\mathbf{H}'$ have odd weight, no three columns add to zero. However, for a column $\mathbf{h}_i$ of weight 3 in $\mathbf{Q}'$, there exists three columns $\mathbf{h}_j$, $\mathbf{h}_l$, and $\mathbf{h}_s$ in $\mathbf{I}_m$ such that $\mathbf{h}_i + \mathbf{h}_j + \mathbf{h}_l + \mathbf{h}_s = \mathbf{0}$.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 35
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

Thus, the shortened Hamming code with $\mathbf{H}'$ as a parity-check matrix has minimum distance exactly 4.

The distance 4 shortened Hamming code can be used for correcting all error patterns of single error and simultaneously detecting all error patterns of double errors. When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1's. However, when double errors occur, the syndrome is also nonzero, but it contains even number of 1's. Based on these facts, decoding can be accomplished in the following manner:

1. If the syndrome $\mathbf{s}$ is zero, we assume that no error occurred.
2. If $\mathbf{s}$ is nonzero and it contains odd number of 1's, we assume that a single error occurred. The error pattern of a single error that corresponds to $\mathbf{s}$ is added to the received vector for error correction.
3. If $\mathbf{s}$ is nonzero and it contains even number of 1's, an uncorrectable error pattern has been detected.

A class of single-error-correcting and double-error-detecting shortened Hamming codes which is widely used for error control in computer main/or control storages is presented in Chapter 16.

The weight distribution of a Hamming code of length $n = 2^m - 1$ is known [1–4]. The number of code vectors of weight i, A_i, is simply the coefficient of z^i in the expansion of the following polynomial:

$$A(z) = \frac{1}{n+1}\{(1+z)^n + n(1-z)(1-z^2)^{(n-1)/2}\}. \tag{3.43}$$

This polynomial is the weight enumerator for the Hamming codes.

Example 3.11

Let $m = 3$. Then $n = 2^3 - 1 = 7$ and the weight enumerator for the (7, 4) Hamming code is

$$A(z) = \tfrac{1}{8}\{(1+z)^7 + 7(1-z)(1-z^2)^3\} = 1 + 7z^3 + 7z^4 + z^7.$$

Hence, the weight distribution for the (7, 4) Hamming code is $A_0 = 1$, $A_3 = A_4 = 7$, and $A_7 = 1$.

The dual code of a $(2^m - 1, 2^m - m - 1)$ Hamming code is a $(2^m - 1, m)$ linear code. This code has a very simple weight distribution; it consists of the all-zero code word and $2^m - 1$ code words of weight 2^{m-1}. Thus, its weight enumerator is

$$B(z) = 1 + (2^m - 1)z^{2^{m-1}}. \tag{3.44}$$

The duals of Hamming codes are discussed further in Chapter 7.

If a Hamming code is used for error detection over a BSC, its probability of an undetected error, $P_u(E)$, can be computed either from (3.35) and (3.43) or from (3.36) and (3.44). Computing $P_u(E)$ from (3.36) and (3.44) is easier. Combining (3.36) and (3.44), we obtain

$$P_u(E) = 2^{-m}\{1 + (2^m - 1)(1-2p)^{2^{m-1}}\} - (1-p)^{2^{m-1}}. \tag{3.45}$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 36
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

The probability $P_u(E)$ for Hamming codes does satisfy the upper bound $2^{-(n-k)} = 2^{-m}$ for $p \leq \frac{1}{2}$ [i.e., $P_u(E) \leq 2^{-m}$] [7]. This can be shown by using the expression of (3.45) (see Problem 3.21).

PROBLEMS

3.1. Consider a systematic (8, 4) code whose parity-check equations are

$$v_0 = u_1 + u_2 + u_3,$$
$$v_1 = u_0 + u_1 + u_2,$$
$$v_2 = u_0 + u_1 + u_3,$$
$$v_3 = u_0 + u_2 + u_3.$$

where u_0, u_1, u_2, and u_3 are message digits and v_0, v_1, v_2, and v_3 are parity-check digits. Find the generator and parity-check matrices for this code. Show analytically that the minimum distance of this code is 4.

3.2. Construct an encoder for the code given in Problem 3.1.

3.3. Construct a syndrome circuit for the code given in Problem 3.1.

3.4. Let $\mathbf{H}$ be the parity-check matrix of an (n, k) linear code C that has both odd- and even-weight code vectors. Construct a new linear code C_1 with the following parity-check matrix:

$$\mathbf{H}_1 = \left[\begin{array}{c|ccc} 0 & & & \\ 0 & & & \\ \cdot & & \mathbf{H} & \\ \cdot & & & \\ \cdot & & & \\ 0 & & & \\ \hline 1 & 1 & 1 \cdots & 1 \end{array}\right].$$

(Note that the last row of $\mathbf{H}_1$ consists of all 1's)

(a) Show that C_1 is an $(n + 1, k)$ linear code. C_1 is called an *extension* of C.

(b) Show that every code vector of C_1 has even weight.

(c) Show that C_1 can be obtained from C by adding an extra parity-check digit, denoted v_∞, to the left of each code vector $\mathbf{v}$ as follows: (1) if $\mathbf{v}$ has odd weight, then $v_\infty = 1$, and (2) if $\mathbf{v}$ has even weight, then $v_\infty = 0$. The parity-check digit v_∞ is called an *overall parity-check* digit.

3.5. Let C be a linear code with both even-weight and odd-weight code vectors. Show that the number of even-weight code vectors is equal to the number of odd-weight code vectors.

3.6. Consider an (n, k) linear code C whose generator matrix $\mathbf{G}$ contains no zero column. Arrange all the code vectors of C as rows of a 2^k-by-n array.

(a) Show that no column of the array contains only zeros.

(b) Show that each column of the array consists of 2^{k-1} zeros and 2^{k-1} ones.

(c) Show that the set of all code vectors with zeros in a particular component forms a subspace of C. What is the dimension of this subspace?

3.7. Prove that the Hamming distance satisfies the triangle inequality; that is, let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ be three n-tuples over GF(2), and show that

$$d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z}).$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 37
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

3.8. Prove that a linear code is capable of correcting λ or fewer errors and simultaneously detecting $l (l > \lambda)$ or fewer errors if its minimum distance $d_{\min} \geq \lambda + l + 1$.

3.9. Determine the weight distribution of the (8, 4) linear code given in Problem 3.1. Let the transition probability of a BSC be $p = 10^{-2}$. Compute the probability of an undetected error of this code.

3.10. Since the (8, 4) linear code given in Problem 3.1 has minimum distance 4, it is capable of correcting all the single-error patterns and simultaneously detecting any combination of double errors. Construct a decoder for this code. The decoder must be capable of correcting any single error and detecting any double errors.

3.11. Let Γ be the ensemble of all the binary systematic (n, k) linear codes. Prove that a nonzero binary n-tuple $\mathbf{v}$ is either contained in exactly $2^{(k-1)(n-k)}$ codes in Γ or contained in none of the codes in Γ.

3.12. The (8, 4) linear code given in Problem 3.1 is capable of correcting 16 error patterns (the coset leaders of a standard array). Suppose that this code is used for a BSC. Devise a decoder for this code based on the table-lookup decoding scheme. The decoder is designed to correct the 16 most probable error patterns.

3.13. Let C_1 be an (n_1, k) linear systematic code with minimum distance d_1 and generator matrix $\mathbf{G}_1 = [\mathbf{P}_1 \ \mathbf{I}_k]$. Let C_2 be an (n_2, k) linear systematic code with minimum distance d_2 and generator matrix $\mathbf{G}_2 = [\mathbf{P}_2 \ \mathbf{I}_k]$. Consider an $(n_1 + n_2, k)$ linear code with the following parity-check matrix:

$$\mathbf{H} = \left[\begin{array}{c|c} & \mathbf{P}_1^T \\ \mathbf{I}_{n_1+n_2-k} & \mathbf{I}_k \\ & \mathbf{P}_2^T \end{array}\right].$$

Show that this code has minimum distance at least $d_1 + d_2$.

3.14. Show that the dual code of the (8, 4) linear code C given in Problem 3.1 is identical to C. C is said to be *self-dual.*

3.15. Form a parity-check matrix for a (15, 11) Hamming code. Devise a decoder for this code.

3.16. For any binary (n, k) linear code with minimum distance (or minimum weight) $2t + 1$ or greater, show that the number of parity-check digits satisfies the following inequality:

$$n - k \geq \log_2 \left[1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}\right].$$

The inequality above gives an upper bound on the random error-correcting capability t of an (n, k) linear code. This bound is known as the *Hamming bound* [5]. [*Hint:* For an (n, k) linear code with minimum distance $2t + 1$ or greater, all the n-tuples of weight t or less can be used as coset leaders in a standard array.]

3.17. Show that the Hamming codes achieve the Hamming bound.

3.18. Show that the minimum distance $d_{\min}$ of an (n, k) linear code satisfies the following inequality:

$$d_{\min} \leq \frac{n \cdot 2^{k-1}}{2^k - 1}.$$

(*Hint:* Use the result of Problem 3.6(b). The bound above is known as the *Plotkin bound* [1–3].)

3.19. Show that there exists an (n, k) linear code with minimum distance at least d if

$$\sum_{i=1}^{d-1} \binom{n}{i} < 2^{n-k}.$$

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 38
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC

[*Hint:* Use the result of Problem 3.11 and the fact that the nonzero n-tuples of weight $d-1$ or less can be at most in

$$\left\{\sum_{i=1}^{d-1}\binom{n}{i}\right\} \cdot 2^{(k-1)(n-k)}$$

(n, k) systematic linear codes.]

3.20. Show that there exists an (n, k) linear code with minimum distance at least $d_{\min}$ which satisfies the following inequality:

$$\sum_{i=1}^{d_{\min}-1}\binom{n}{i} < 2^{n-k} \leq \sum_{i=1}^{d_{\min}}\binom{n}{i}.$$

(*Hint:* See Problem 3.19. The second inequality provides a lower bound on the minimum distance attainable with an (n, k) linear code. This bound is known as *Varsharmov–Gilbert* bound [1–3].)

3.21. Show that the probability of an undetected error for Hamming codes on a BSC with transition probability p satisfies the upper bound 2^{-m} for $p \leq \frac{1}{2}$. [*Hint:* Use the inequality $(1-2p) \leq (1-p)^2$.]

3.22. Compute the probability of an undetected error for a (15, 11) Hamming code on a BSC with transition probability $p = 10^{-2}$.

REFERENCES

1. E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.
2. W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed., MIT Press, Cambridge, Mass., 1972.
3. F. J. MacWilliams and J. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.
4. R. J. McEliece, *The Theory of Information and Coding*, Addison-Wesley, Reading, Mass., 1977.
5. F. J. MacWilliams, "A Theorem on the Distribution of Weights in a Systematic Code," *Bell Syst. Tech. J.*, 42, pp. 79–94, 1963.
6. R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell Syst. Tech. J.*, 29, pp. 147–160, April 1950.
7. S. K. Leung-Yan-Cheong and M. E. Hellman, "Concerning a Bound on Undetected Error Probability," *IEEE Trans. Inf. Theory*, IT-22, pp. 235–237, March 1976.

Security First Innovations, LLC, Exhibit 2032
Page 2032 - 39
IPR2025-01202, Intl. Bus. Machs. Corp. v. Security First Innovations, LLC