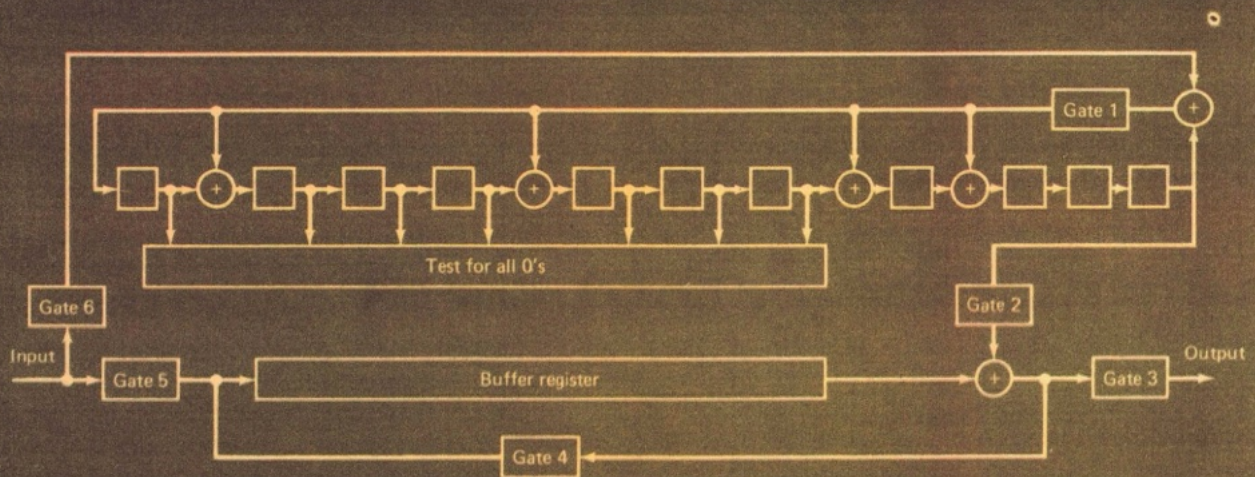


SHU LIN / DANIEL J. COSTELLO, Jr.

Error Control Coding:

Fundamentals and Applications



PRENTICE-HALL SERIES IN COMPUTER APPLICATIONS IN ELECTRICAL ENGINEERING

Franklin F. Kuo, Editor

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

IPR2025-00756, OnePlus Ex. 1011

Page 2 of 8

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)

I. 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 6 5 4

ISBN 0-13-283796-X

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
EDITORIA 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*

IPR2025-00756, OnePlus Ex. 1011

Page 3 of 8

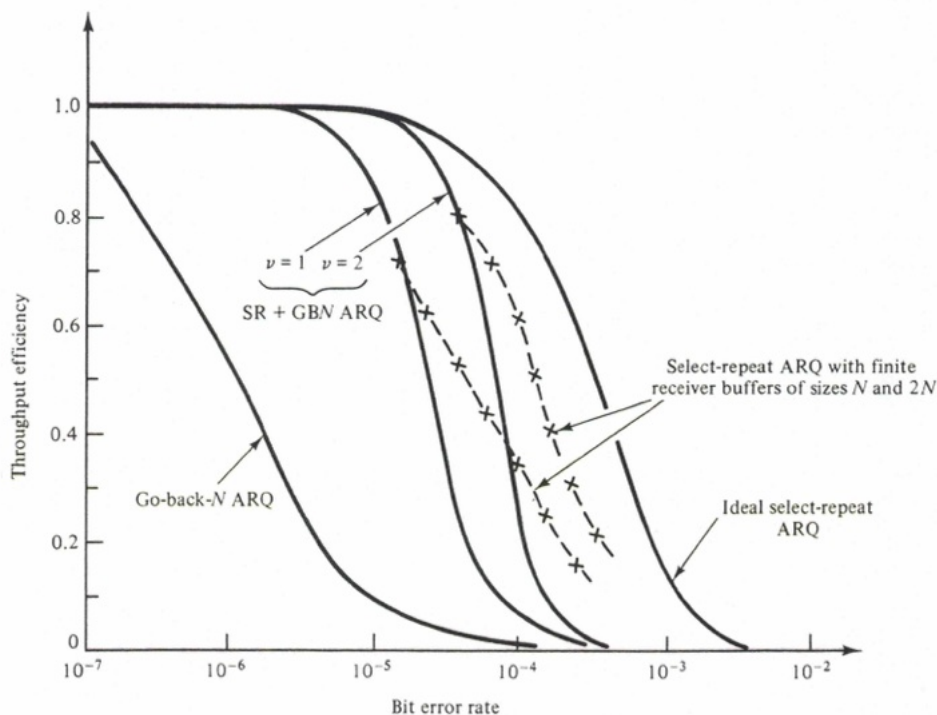


Figure 15.13 Throughput efficiencies of the SR + GBN with $v = 1$ and 2, $N = 512$ and $n = 2024$.

15.4 HYBRID ARQ SCHEMES

Comparing the two error control schemes, we see that ARQ is simple and provides high system reliability. However, ARQ systems have a severe drawback—their throughputs fall rapidly with increasing channel error rate. Systems using FEC maintain constant throughput (equal to the code rate $R = k/n$) regardless of the channel error rate. However, FEC systems have two drawbacks. First, when a received vector is detected in error, it must be decoded and the decoded message must be delivered to the user regardless of whether it is correct or incorrect. Since the probability of a decoding error is much greater than the probability of an undetected error, it is hard to achieve high system reliability with FEC. Second, to obtain high system reliability, a long powerful code must be used and a large collection of error patterns must be corrected. This makes decoding hard to implement and expensive. For these reasons, ARQ is often preferred over FEC for error control in data communication systems, such as packet-switching data networks and computer communication networks. However, in communication (or data storage) systems where return channels are not available or retransmission is not possible for some reason, FEC is the only choice.

The drawbacks in both ARQ and FEC could be overcome if two error control schemes are properly combined. Such a combination of the two basic error control

schemes is referred to as a *hybrid ARQ* [5,17]. A hybrid ARQ system consists of an FEC subsystem contained in an ARQ system. The function of the FEC portion is to reduce the frequency of retransmission by correcting the error patterns that occur most frequently. This increases the system throughput. When a less frequent error pattern occurs and is detected, the receiver requests a retransmission rather than passing the unreliably decoded message to the user. This increases the system reliability. As a result, a proper combination of FEC and ARQ would provide higher reliability than an FEC system alone and a higher throughput than the system with ARQ only. Furthermore, since the decoder is designed to correct a small collection of error patterns, it can be simple. The FEC scheme can be incorporated with any of the three basic ARQ schemes.

A straightforward hybrid ARQ scheme is to use a code, say an (n, k) linear code, which is designed for simultaneous error correction and error detection. When a received vector is detected in error, the receiver first attempts to locate and correct the errors. If the number of errors (or the length of an error burst) is within the designed error-correcting capability of the code, the errors will be corrected and the decoded message will be passed to the user or saved in a buffer until it is ready to be delivered. If an uncorrectable error pattern is detected, the receiver rejects the received vector and requests a retransmission. When the retransmitted vector is received, the receiver again attempts to correct the errors (if any). If the decoding is not successful, the receiver again rejects the received vector and asks for another retransmission. This error-correction and retransmission process continues until the vector is successfully received or decoded. For example, one may use the (1023, 923) BCH code in a hybrid ARQ system. This code has a minimum distance 21; it can be used for correcting five or fewer errors and simultaneously detecting any combination of 15 or fewer errors (and many other error patterns). If an error pattern with five or fewer errors occurs, it will be detected and corrected. If an error pattern with more than five but less than 16 errors occurs, it will be detected. In this event, the receiver will request a retransmission of the erroneous vector.

The hybrid ARQ scheme described above is referred to as the *type I hybrid ARQ scheme*. Since a code used in this scheme must be able to correct a certain collection of error patterns and simultaneously detect other error patterns, more parity-check digits are needed. This increases the overhead for each transmission and retransmission. As a result, when the channel error rate is low, it has a lower throughput than its corresponding ARQ scheme. However, when the channel error rate increases, the throughput of the ARQ scheme drops rapidly and the hybrid-ARQ scheme provides higher throughput, as shown in Figure 15.14. The type I hybrid ARQ scheme is capable of maintaining significant high throughput over a wide range of channel error rates if the designed error-correcting capability of the code is sufficiently large.

Let P_i be the probability that a received vector will be decoded successfully. This probability depends on the designed error-correcting capability of the code being used. Replacing P by P_i in Equations (15.4), (15.5), (15.6), and (15.11), we obtain the throughput efficiencies for the type I hybrid selective-repeat ARQ with infinite receiver buffer, the type I hybrid go-back- N ARQ, the type I hybrid stop-

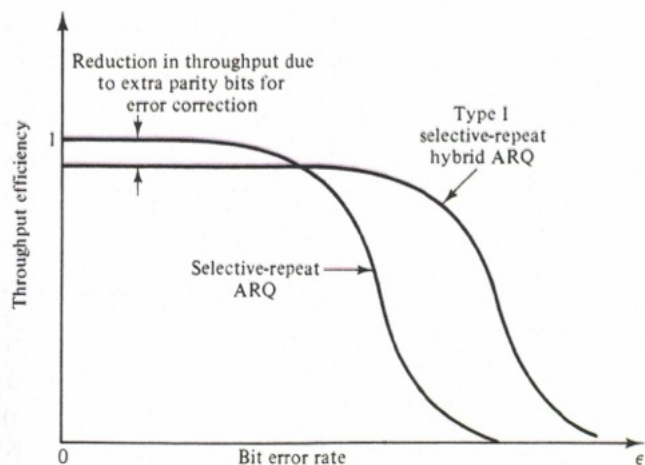


Figure 15.14 Throughput performance of the type I selective-repeat ARQ.

and-wait ARQ, and the type I hybrid selective-repeat ARQ with receiver buffer of size N , respectively.

The second type (or type II) of hybrid ARQ scheme is devised based on the concept that the parity-check digits for error correction are sent to the receiver only when they are needed [18,19]. Two linear codes are used in this type of scheme; one is a high-rate (n, k) code C_0 which is designed for error detection only, the other is an *invertible* half-rate $(2k, k)$ code C_1 which is designed for simultaneous error correction and error detection. A code is said to be invertible if, knowing only the parity-check digits of a code vector, the corresponding information digits can be uniquely determined by an *inversion process*. In the next section, a class of invertible half-rate linear codes is presented and we show that the inversion can be accomplished by a linear sequential circuit.

When a message $\mathbf{u}$ of k information digits is ready for transmission, it is encoded into a code vector $\mathbf{v} = (f(\mathbf{u}), \mathbf{u})$ of n digits based on the error-detecting code C_0 , where $f(\mathbf{u})$ denotes the $n - k$ parity-check digits. The code vector $\mathbf{v} = (f(\mathbf{u}), \mathbf{u})$ is then transmitted. At the same time, the transmitter computes the k parity-check digits, denoted by $q(\mathbf{u})$, based on the message $\mathbf{u}$ and the half-rate invertible code C_1 . Clearly, the $2k$ -tuple $(q(\mathbf{u}), \mathbf{u})$ is a code vector in C_1 . The k -bit parity block $q(\mathbf{u})$ is not transmitted but stored in the retransmission buffer of the transmitter for later use. Let $\tilde{\mathbf{v}} = (\tilde{f}(\mathbf{u}), \tilde{\mathbf{u}})$ denote the received vector corresponding to $\mathbf{v} = (f(\mathbf{u}), \mathbf{u})$. When $\tilde{\mathbf{v}}$ is received, the receiver computes the syndrome of $\tilde{\mathbf{v}}$ based on C_0 . If the syndrome is zero, $\tilde{\mathbf{u}}$ is assumed to be error-free and will be accepted by the receiver. If the syndrome is not zero, errors are detected in $\tilde{\mathbf{v}}$. The erroneous message $\tilde{\mathbf{u}}$ is then saved in the receiver buffer and a NAK is sent to the transmitter. Upon receiving this NAK, the transmitter encodes the k -bit parity block $q(\mathbf{u})$ into a code vector $\mathbf{v}^* = (f[q(\mathbf{u})], q(\mathbf{u}))$ of n bits based on C_0 , where $f[q(\mathbf{u})]$ denotes the $n - k$ parity-check digits for $q(\mathbf{u})$. This vector $\mathbf{v}^* = (f[q(\mathbf{u})], q(\mathbf{u}))$ is then transmitted (here the retransmission is a parity vector). Let $\tilde{\mathbf{v}}^* = (\tilde{f}[q(\mathbf{u})], \tilde{q}(\mathbf{u}))$ denote the received vector corresponding to $\mathbf{v}^* = (f[q(\mathbf{u})], q(\mathbf{u}))$. When $\tilde{\mathbf{v}}^*$ is received, the syndrome of $\tilde{\mathbf{v}}^*$ is computed based on C_0 . If the syndrome is zero, $\tilde{q}(\mathbf{u})$ is assumed to be error-free and

the message $\mathbf{u}$ is recovered from $\tilde{q}(\mathbf{u})$ by inversion. If the syndrome is not zero, $\tilde{q}(\mathbf{u})$ and the erroneous message $\tilde{\mathbf{u}}$ (stored in the receiver buffer) together are used for *error correction* based on the half-rate code C_1 . If the errors in $(\tilde{q}(\mathbf{u}), \tilde{\mathbf{u}})$ form a correctable error pattern, they will be corrected. The decoded message $\mathbf{u}$ is then accepted by the receiver. If the errors in $(\tilde{q}(\mathbf{u}), \tilde{\mathbf{u}})$ form a detectable but not a correctable error pattern, $\tilde{\mathbf{u}}$ is discarded and the erroneous parity block $\tilde{q}(\mathbf{u})$ is stored in the receiver buffer; also, a NAK is sent to the transmitter.

Upon receiving the second NAK for the code vector $\mathbf{v} = (f(\mathbf{u}), \mathbf{u})$, the transmitter resends $\mathbf{v} = (f(\mathbf{u}), \mathbf{u})$. When $\tilde{\mathbf{v}} = (\tilde{f}(\mathbf{u}), \tilde{\mathbf{u}})$ is received, the syndrome of $\tilde{\mathbf{v}}$ is again computed based on C_0 . If the syndrome is zero, $\tilde{\mathbf{u}}$ is assumed to be error-free and is accepted by the receiver, the erroneous parity block $\tilde{q}(\mathbf{u})$ is then discarded. If the syndrome is not zero, $\tilde{\mathbf{u}}$ and the erroneous parity block $\tilde{q}(\mathbf{u})$ (stored in the receiver buffer) together are used for error correction based on C_1 . If the errors in $(\tilde{q}(\mathbf{u}), \tilde{\mathbf{u}})$ are corrected, the decoded message $\mathbf{u}$ is then accepted by the receiver and an ACK is sent to the transmitter. However, if the errors in $(\tilde{q}(\mathbf{u}), \tilde{\mathbf{u}})$ are detectable but not correctable, $\tilde{q}(\mathbf{u})$ is discarded, $\tilde{\mathbf{u}}$ is stored in the receiver buffer, and a NAK is sent to the transmitter. The next retransmission will be the parity vector $\mathbf{v}^* = (f[q(\mathbf{u})], q(\mathbf{u}))$. Therefore, the retransmissions are *alternating repetitions* of the parity code vector $\mathbf{v}^* = (f[q(\mathbf{u})], q(\mathbf{u}))$ and the information code vector $\mathbf{v} = (f(\mathbf{u}), \mathbf{u})$. The receiver stores the received message $\tilde{\mathbf{u}}$ and the received parity block $\tilde{q}(\mathbf{u})$ alternately. The retransmissions continue until $\mathbf{u}$ is finally recovered.

The most important feature of the type II hybrid ARQ is the *parity retransmission* for error correction based on a half-rate invertible code C_1 . This parity-retransmission strategy can be incorporated with any of the three basic types of ARQ. It is particularly effective when it is used in conjunction with the selective-repeat ARQ. Because of the invertible property of C_1 , the message $\mathbf{u}$ can be reconstructed uniquely from the parity block $q(\mathbf{u})$ by inversion. Hence, the parity block $q(\mathbf{u})$ contains the same amount of information as the message. As a result, the overhead per transmission or retransmission is simply the number of parity-check digits, $n - k$, needed for error detection based on the (n, k) code C_0 , which is required by any ARQ scheme. Therefore, when the channel is quiet or the channel error rate is low, the type II hybrid ARQ has the same throughput efficiency as its corresponding ARQ scheme. When the channel error rate is high, the error correction provided by the half-rate code C_1 will maintain the throughput high. In Section 15.5 we present a type II hybrid ARQ that incorporates the parity-retransmission strategy with the selective-repeat ARQ with finite receiver buffer. We will show that, for a BSC, the throughput efficiency of the ideal selective-repeat ARQ with infinite receiver buffer can be achieved by a half-rate invertible code C_1 with a very small designed error-correcting capability, say $t = 3$ to 5. With a larger t , the type II hybrid selective-repeat ARQ with finite receiver buffer will be far superior to the ideal selective-repeat ARQ with infinite receiver buffer.

The decoding complexity for a type II hybrid ARQ is only slightly greater than that of a corresponding type I hybrid ARQ with the same designed error-correcting capability. The extra circuits needed for a type II hybrid ARQ scheme are an inversion circuit based on C_1 , which is simply a linear sequential circuit and an error detection circuit based on C_0 .

The disadvantage of the type I hybrid ARQ is that the overhead due to the extra parity-check digits for error correction must be included in each transmission or retransmission regardless of the channel error rate. When the channel is quiet, this represents a waste. However, the type II hybrid ARQ removes this disadvantage. It is an adaptive scheme. This scheme is particularly attractive for high-speed data communication systems where round-trip delay is large and error rate is nonstationary such as satellite communication systems.

Various hybrid ARQ schemes and their analysis can be found in References 17 to 29.

15.5 CLASS OF HALF-RATE INVERTIBLE CODES

In a type II hybrid ARQ system, C_1 is chosen as a half-rate invertible code. The invertible property facilitates the data recovery process. During a retransmission, if the parity block $q(u)$ is successfully received (no errors being detected), the message u can be reconstructed from $q(u)$ by a simple inversion process rather than by a more complicated decoding process. The inversion process also reduces the frequency of retransmission. For example, if the received message $\tilde{u}$ contains more than t errors and the received parity block $\tilde{q}(u)$ is error-free, the decoding process based on $(\tilde{q}(u), \tilde{u})$ would not be able to recover the message u . Hence, another retransmission would be required. However, taking the inverse of the error-free parity block $\tilde{q}(u)$, we will be able to recover u and thus avoid another retransmission.

In the following, a class of half-rate invertible block codes will be presented and we will show that inversion can be accomplished by a linear sequential circuit.

Let C be an (n, k) cyclic code with $n - k \leq k$. Let $g(X)$ be the generator polynomial of C with the form

$$g(X) = 1 + g_1X + g_2X^2 + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}.$$

Let

$$v(X) = v_0 + v_1X + v_2X^2 + \cdots + v_{n-1}X^{n-1}$$

be a code polynomial. In systematic form, the k leading high-order coefficients $v_{n-k}, v_{n-k+1}, \dots, v_{n-1}$ are identical to k information digits, the $n - k$ low-order coefficients $v_0, v_1, \dots, v_{n-k-1}$ are parity-check digits. Consider the set of those code vectors in C whose $2k - n$ leading high-order components $v_{2(n-k)}, v_{2(n-k)+1}, \dots, v_{n-1}$ are zeros. There are 2^{n-k} such code vectors in C . If the $2k - n$ high-order zero components are removed from these code vectors, we obtain a set of 2^{n-k} vectors of length $2(n - k)$. These vectors form a half-rate $(2n - 2k, n - k)$ shortened cyclic code C_1 (see section 4.7). This shortened cyclic code has at least the same error-correcting capability as C . We have shown in Section 4.7 that the encoding and decoding of C_1 can be accomplished by the same circuits (or with a slight modification) as employed by C .

Next, we show that the shortened cyclic code C_1 has the invertible property.

Let

$$u(X) = u_0 + u_1X + \cdots + u_{n-k-1}X^{n-k-1}$$