

Preface

Error control coding theory has been studied for over half a century, and it is still going stronger than ever. The most recent examples are the turbo codes and the low-density parity check codes (LDPCs). Also, during these years, error control codes have been extensively applied to various digital systems, such as computer and communication systems, as an essential technique to improve system reliability. As an integral part of modern day high-speed dependable systems and semiconductor memories, high-speed parallel decoding is essential. Error control codes suitable for high-speed parallel decoding are regularly expressed and studied in parity-check matrices. For highly reliable communication systems and disk memory systems, on the other hand, serial decoding based on linear feedback shift registers (LFSRs) is used. Error control codes for serial decoding are typically expressed and studied using generator polynomials. In this book, the former codes are called *matrix codes* and the latter *polynomial codes*. So far, traditional coding theory has been studied mainly using code generator polynomials. We emphasize that the linear codes expressed in polynomials can always be expressed using parity-check matrices, but the converse is not always possible. This book focuses specifically on the design theory for matrix codes and their practical applications, which has been seriously lacking in the traditional scope of coding theory investigations.

In dependable computer systems, many types of error control codes have been applied to memory subsystems and processors in order to achieve efficient and reliable data processing and storage. Some systems could never have been realized without the application of cost-effective error control codes, mainly very large capacity, high-speed semiconductor memories, very high-density magnetic disk memories, and recent optical disk memories such as compact disc (CD) and digital versatile disc (DVD). More recently mobile digital systems have gained wide popularity, and these systems are sometimes operated under unfavorable environments where electromagnetic noise, α -particles and cosmic rays abound. Modern high-speed, high-density VLSI processors and semiconductor memories are operated at low supply voltage levels and thus low logic signal swing; they therefore are vulnerable to external disturbances that can induce transient errors. Transient errors are a dominant concern in today's digital systems. Error control

coding is the most efficient and effective way to tolerate these errors, and is expected to become ever more important in future VLSI systems.

The challenge is to choose among many different applications of error control codes. Often a new application calls for a new type of code that can be developed most efficiently to fit a new requirement. Matrix codes are far more flexible compared with polynomial codes. Parity-check matrices can be manipulated easily. Some known examples are column vector exchange in a matrix, the odd-weight-column matrix, the low-density matrix, and the rotational matrix form. These manipulations of matrices have yielded many useful codes for important applications. Polynomial codes, on the other hand, are impossible to be manipulated in a similar way for code design fine-tuning. The main reason is that the matrix code is capable of expressing various types of code functions and thus allows for very high design flexibility. In practice, such flexibility has led to excellent code designs, satisfying the various reliability requirements of the dependable systems.

This book builds on the author's previous book, *Error Control Coding for Computer Systems* (Prentice-Hall, 1989), and it likewise aims at introducing the latest developments and advances in the field. However, as was mentioned earlier, additionally the book is unique in its concentration on the treatment of matrix codes. Unlike any existing coding theory books, this book will not burden the reader with unnecessary background on polynomial algebra. The book includes only the mathematical background essential for the understanding of matrix code construction and design. Such an arrangement frees up space for the description of some fine artistry of matrix code design strategies and techniques. Matrix code designs are presented with respect to practical applications, such as high-speed semiconductor memories, mass memories of disks and tapes, logic circuits and systems, data entry systems, and distributed storage systems. Also new classes of matrix codes, such as error locating codes, spotty byte error control codes, and unequal error control codes, are presented in their practical settings. The new parallel decoding algorithm of the burst error control codes is demonstrated and further extended to the generalized parallel decoding of the codes.

Chapter 1 provides background and a preview of material covered in the subsequent chapters. First, it defines faults, errors, and failures and explains the many types of faults and errors. This is the core knowledge needed to understand what constitutes a good code. To design an efficient and effective code for a given application, it is important first to know what types of errors matter, how much the system's reliability can be improved by coding techniques, and what are the constraints on check-bit length, decoding speed, and so forth. The matrix code designing procedure is laid out in this chapter from this standpoint. The chapter concludes with a brief introduction to the competitors of the coding technique in dependable systems, namely conventional error recovery techniques and/or error masking techniques.

Chapter 2 provides the fundamental mathematical background and coding theory necessary to understand the later chapters. The chapter covers the matrix representations of well-known error control codes, such as simple parity-check codes, cyclic codes, Hamming codes, BCH codes, Reed-Solomon codes, and Fire codes. These codes are manipulated in the later chapters in examples of how matrix codes satisfy the system requirements for given applications.

Chapter 3 discusses the matrix code design techniques related to high-speed decoding, area efficient encoding/decoding hardware, modularized organization of encoding/decoding circuits, and so forth.

Chapters 4, 5, 6, and 7 cover topics on matrix code design for high-speed semiconductor memories. Depending on the application, the matrix code can be designed to handle bit or byte errors and in some cases a mixture of both bit and byte errors. The latter are typical errors found in large capacity semiconductor memory systems using high-density RAM chips. Chapter 4 discusses bit error control codes, such as the modified Hamming single-bit error correcting and double-bit error detecting (SEC-DED) codes, the modified double-bit error correcting BCH codes, and the memory on-chip codes. For the memory systems using byte-organized RAM chips, single-byte error correcting (*SbEC*) codes, and single-byte error correcting and double-byte error detecting (*SbEC-DbED*) codes, are presented in Chapter 5. The codes for the mixed type of bit errors and byte errors are presented in Chapter 6. Among them, a byte error detecting SEC-DED code, developed by the author and his colleague in the 1980s, has found practical application in recent workstations. Chapter 7 presents a relatively new class of byte error control codes: spotty byte error control codes. This class of codes has been specifically designed to fit the large capacity memory systems that use high-density RAM chips with wide input/output data of 8, 16, and 32 bits. Also a general class of these codes with minimum Hamming distance- d and with maximum distance separable (MDS) characteristics is presented in this chapter. The well-known Reed-Solomon codes are included in these generalized codes, which makes them practically and theoretically important. They will be quite useful for future applications.

Chapter 8 presents the generalized parallel decoding algorithm for error control codes. Initially developed for burst error control codes, this new decoding algorithm includes the conventional parallel decoding algorithm of the existing bit/byte error correcting codes. The generalized algorithm can also be used for multiple burst or byte error correcting codes. The chapter takes this new algorithm and demonstrates how the parallel decoding method can be implemented in combinational circuits. In addition the chapter addresses the important problem of glitches in parallel decoding circuits. Parallel decoding circuits depend heavily on large exclusive-OR tree circuits, which are well known to readily produce glitches. The glitches are the unwanted logic signal transitions that can generate, propagate, and accumulate in the logic circuits and then induce noise and instability on the power supply lines. The chapter explains why the glitches are generated, how they are propagated and accumulated in the circuits, and how to reduce these undesirable effects.

Chapter 9 presents a new class of codes, namely error locating codes. Error location is an error control function lying midway between error correction and error detection. An error locating code will indicate where the errors lie but not the precise erroneous digit positions. This type of codes is useful for identifying the faulty block, faulty package, or faulty chip, and thus enables fault isolation and reconfiguration. The chapter includes practical codes for memory systems to use in locating faulty packages/cards. It also provides a practical code for locating faulty chips. Both codes have the capability to correct single-bit errors, even though the codes are mainly designed for identifying the faulty areas. In addition, burst error locating codes are introduced here. The chapter concludes with a precise analysis of error locating codes with an emphasis on the code conditions and their relation between error locating codes and error correcting/detecting codes.

Chapter 10 shows yet another new class of unequal error control (UEC) codes. In many applications certain positions in a word have higher error rates or require more protection. The UEC codes can indicate the area in a word having a higher error rate with stronger error control code functions, and the area having a lower error rate with weaker error

control functions. In other words, this type of code has different code functions within a code word, depending on the area and the associated error rate. The chapter provides optimal codes with some UEC code functions. Similar codes exist in unequal error protection (UEP) codes. This type of code protects the valuable information part of a word against errors. For example, control information or address information in communication messages or computer words, or similarly pointer information in the database words, must be more protected from errors than their other parts. The chapter provides some UEP codes that protect against burst errors and also against single-bit errors. The chapter includes examples of UEC and UEP codes used in holographic memories and lossless compressed data.

Chapters 11, 12, 13, and 14 present the codes for some specific systems, namely mass memories such as magnetic tapes and disks, logic circuits and systems, data entry systems, and distributed storage systems. Chapter 11 covers the codes designed specifically for mass memories such as tape memories, magnetic disk memories, and recent optical disk memories. The various modified types of Reed-Solomon codes and adaptive parity codes are presented to the tape memories and to the disk memories. Codes for recent CDs and DVDs are also introduced. Chapter 12 mentions error checking for logic systems using efficient error detecting codes. An important concept of self-checking is first introduced. The chapter then clarifies how the errors in the logic circuits and systems are detected, how the error detecting checker circuits are implemented, how the errors in the checker itself are detected, and how the self-testing checkers are implemented. Especially self-checking ALU is presented by using parity-based codes, and also self-checking design for processor systems is demonstrated. Chapter 13 presents the codes for data entry systems. In these systems, in general, nonbinary symbols are routinely used in character recognition systems, and recent two-dimensional symbols. The chapter characterizes the errors that occur in these nonbinary symbols as asymmetric errors and presents some asymmetric error control codes. These codes are basically nonlinear, and are designed by using elements in newly defined rings. Also nonsystematic nonbinary asymmetric error correcting codes are designed based on a multilevel coding method and a set-partitioning algorithm, and QR codes and two-dimensional unidirectional clustered error correcting codes are presented for two-dimensional matrix symbols. Chapter 14 provides the codes for distributed storage systems connected via networks. Codes for recent RAID systems that tolerate two disk failures are introduced, and then an efficient error recovery scheme from multiple disk failures in the distributed storage system is discussed and is implemented by using block design in combinatorial theory.

The introductory portion of the book, Chapters 1 and 2, and the parts of Chapters 3, 4, 5, 6, 8, 9, and 10, can be used as the text for a course at an advanced undergraduate level or for an introductory one-semester course at the graduate level. For graduate classes and advanced students who have the background in mathematics, logic circuits, and rudimentary knowledge of codes, the book can be used as a whole with selected topics from each of the chapters. Practicing engineers/designers will find useful discussions in Chapters 6 to 14, which demonstrate, in detail, the procedure of designing sophisticated codes in practical form. For the practicing engineer, Chapter 2 presents mathematics and coding theory, not in strict form but in introductory form, which is necessary in understanding the later chapters. Many examples, figures, exercises, and references are provided in each chapter of the book. Many attractive codes with practical code parameters and their evaluation data on decoding hardware and error detection capabilities

are fully demonstrated. These can be used by practicing engineers as a practical guide and handy reference.

My sincere appreciation goes to many people. Professors Jack K. Wolf of the University of California San Diego, Hideki Imai of the University of Tokyo, T. R. N. Rao of the University of Louisiana Lafayette, and Bella Bose of Oregon State University encouraged me to continue my research on code design theory and to write this book. Emeritus professor Yoshihiro Tohma of Tokyo Institute of Technology, Professors Takashi Nanya of the University of Tokyo, Hideo Ito of Chiba University, and Jien-Chung Lo of the University of Rhode Island gave important suggestions and valuable discussions on research for dependable systems. Recently Professor Lo also provided valuable comments on the final book and an important discussion on glitches, (i.e., logical noise) that are generated, propagated, and accumulated in large exclusive-OR tree circuits in the parallel decoder of the codes. The author's NTT colleagues, Dr. Shigeo Kaneda, now professor at Doshisha University, and Dr. Kazumitsu Matsuzawa, now professor at Kanagawa University, collaborated to develop practical codes for computer memories. Dr. Masato Kitakami, now associate professor at Chiba University, Dr. Mitsuru Hamada, now associate professor at Tamagawa University, Dr. Shuxin Jiang, Dr. Saowapa Kiattichai, Dr. Hongyuang Chen, Dr. Kazuteru Namba, Dr. Ganesan Umanesan, Dr. Haruhiko Kaneko, Dr. Kazuyoshi Suzuki, Mr. Tsuyoshi Tanaka, Mr. Toshihiko Kashiya, and Mr. Hiroyuki Ohde devoted themselves to designing the excellent codes in their master's and/or doctorate course programs at the Tokyo Institute of Technology. Much of the motivation for making the codes practical was due to discussions with many researchers and engineers in Japanese industry.

Thanks also go to art designer, Mr. Ippei Inoh, a friend of mine, who proposed and directed the marvelous idea of the front cover design. Ms. Tiki Ishizuka, a computer graphic designer, arranged the wonderful fine art of this cover. You can see "Hoh-Oh," a legendary happy bird, in the center of the front cover whose original pattern was introduced from China more than one thousand years ago to Japan, and since then appeared as an art design in Japanese art and craft products. I sincerely hope the book will bring happiness and pleasure to the reader.

At this point in a preface, I usually thank my wife, Sachiko, and my daughter's family, Sayaka, Makoto, and Asuka, for encouraging me in continuing this difficult project.

EIJI FUJIWARA

(Autumn in 2005 on the foot of Mt. Fuji)