

# RELIABLE COMPUTER SYSTEMS

DESIGN AND EVALUATION **THIRD EDITION**

DANIEL P. SIEWIOREK and ROBERT S. SWARZ



Editorial, Sales, and Customer Service Office

A K Peters, Ltd.  
63 South Avenue  
Natick, MA 01760

Copyright © 1998 by A K Peters, Ltd.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

Trademark products mentioned in the book are listed on page 890.

### Library of Congress Cataloging-in-Publication Data

Siewiorek, Daniel P.

Reliable computer systems : design and evaluation / Daniel P.  
Siewiorek, Robert S. Swarz. – 3rd ed.

p. cm.

First ed. published under title: The theory and practice of  
reliable system design.

Includes bibliographical references and index.

ISBN 1-56881-092-X

1. Electronic digital computers – Reliability. 2. Fault-tolerant  
computing. I. Swarz, Robert S. II. Siewiorek, Daniel P. Theory  
and practice of reliable system design. III. Title.

QA76.5.S537 1998

98-202237

004-dc21

CIP

Printed in the United States of America

02 01 00 99 98 10 9 8 7 6 5 4 3 2 1

### CREDITS

**Figure 1-3:** Eugene Foley, “The Effects of Microelectronics Revolution on Systems and Board Test,” *Computers*, Vol. 12, No. 10 (October 1979). Copyright © 1979 IEEE. Reprinted by permission.

**Figure 1-6:** S. Russell Craig, “Incoming Inspection and Test Programs,” *Electronics Test* (October 1980). Reprinted by permission.

*Credits are continued on pages 885-890, which are considered a continuation of the copyright page.*

# RELIABLE COMPUTER SYSTEMS

## DESIGN AND EVALUATION

THIRD EDITION

Daniel P. Siewiorek  
Carnegie Mellon University  
Pittsburgh, Pennsylvania

Robert S. Swarz  
Worcester Polytechnic Institute  
Worcester, Massachusetts



A K Peters  
Natick, Massachusetts

**To Karon and Lonnie**

*A Special Remembrance:*

During the development of this book, a friend, colleague, and fault-tolerant pioneer passed away. Dr. Wing N. Toy documented his 37 years of experience in designing several generations of fault-tolerant computers for the Bell System electronic switching systems described in Chapter 8. We dedicate this book to Dr. Toy in the confidence that his writings will continue to influence designs produced by those who learn from these pages.

# PREFACE

System reliability has been a major concern since the beginning of the electronic digital computer age. The earliest computers were constructed of components such as relays and vacuum tubes that would fail to operate correctly as often as once every hundred thousand or million cycles. This error rate was far too large to ensure correct completion of even modest calculations requiring tens of millions of operating cycles. The Bell relay computer (c. 1944) performed a computation twice and compared results; it also employed error-detecting codes. The first commercial computer, the UNIVAC I (c. 1951), utilized extensive parity checking and two arithmetic logic units (ALUs) in a match-and-compare mode. Today, interest in reliability pervades the computer industry—from large mainframe manufacturers to semiconductor fabricators who produce not only reliability-specific chips (such as for error-correcting codes) but also entire systems.

Computer designers have to be students of reliability, and so do computer system users. Our dependence on computing systems has grown so great that it is becoming difficult or impossible to return to less sophisticated mechanisms. When an airline seat selection computer “crashes,” for example, the airline can no longer revert to assigning seats from a manual checklist; since the addition of round-trip check-in service, there is no way of telling which seats have been assigned to passengers who have not yet checked in without consulting the computer. The last resort is a free-for-all rush for seats. The computer system user must be able to understand the advantages and limitations of the state-of-the-art in reliability design; determine the impact of those advantages and limitations upon the application or computation at hand; and specify the requirements for the system’s reliability so that the application or computation can be successfully completed.

The literature on reliability has been slow to evolve. During the 1950s reliability was the domain of industry, and the quality of the design often depended on the cleverness of an individual engineer. Notable exceptions are the work of Shannon [1948] and Hamming [1950] on communication through noisy (hence error-inducing) channels, and of Moore and Shannon [1956] and von Neumann [1956] on redundancy that survives component failures. Shannon and Hamming inaugurated the field of coding theory, a cornerstone in contemporary systems design. Moore, Shannon, and von Neumann laid the foundation for development and mathematical evaluation of redundancy techniques.

During the 1960s the design of reliable systems received systematic treatment in industry. Bell Telephone Laboratories designed and built an Electronic Switching System (ESS), with a goal of only two hours’ downtime in 40 years [Downing, Nowak, and Tuomenoksa, 1964]. The IBM System/360 computer family had extensive serviceability features [Carter et al., 1964]. Reliable design also found increasing use in the aerospace industry, and a triplicated computer helped man land on the moon [Cooper and Chow,

1976; Dickinson, Jackson, and Randa, 1964]. The volume of literature also increased. In 1962 a Symposium on Redundancy Techniques held in Washington, D.C., led to the first comprehensive book on the topic [Wilcox and Mann, 1962]. Later, Pierce [1965] published a book generalizing and analyzing the Quadded Redundancy technique proposed by Tryon and reported in Wilcox and Mann [1962]. A community of reliability theoreticians and practitioners was developing.

During the 1970s interest in system reliability expanded explosively. Companies were formed whose major product was a reliable system (such as Tandem). Due to the effort of Algirdas Avizienis and other pioneers, a Technical Committee on Fault Tolerant Computing (TCFTC) was formulated within the Institute of Electrical and Electronic Engineers (IEEE). Every year since 1971, the TCFTC has held an International Symposium on Fault-Tolerant Computing.

In 1982, when the first edition of *The Theory and Practice of Reliable System Design* was published, the time was ripe for a book on the design of reliable computing structures. The book was divided into two parts—the first being devoted to the fundamental concepts and theory and the second being populated with a dozen chapters that represented detailed case studies. The second edition follows the same basic structure, but is divided into three parts. Part I deals with the theory and Parts II and III with the practice of reliable design. The appendices provide detailed information on coding theory, design for testability, and the MIL-HDBK-217 component reliability model.

In recent years, the number of reliability and redundancy techniques has continued to expand, along with renewed emphasis on software techniques, application of older techniques to newer areas, and in-depth analytical evaluation to compare and contrast many techniques. In Part I, Chapters 3 and 5 have been expanded to include these new results. More case studies have been developed on the frequency and manifestation of hardware and software system failures. Chapter 2 has been updated to include summaries of this new material. Likewise, Chapter 4 has been enlarged to cover testing techniques commencing with prototypes through manufacturing, field installation, and field repair. The new additions to Part I have resulted in over a 50 percent increase in the number of references cited in the second edition over the first edition.

Part II of the second edition has undergone an even more dramatic change. In the first edition, Part II surveyed twelve different computer systems, ranging from one-of-a-kind research vehicles to mass-produced general-purpose commercial systems. The commercial systems focused on error detection and retry and represented three of the case studies. Four case studies represented one-of-a-kind research systems. Three other systems sought limited deployment in aerospace and message-switching applications. Only two of the case studies represented wider-spread deployment of fault-tolerant systems numbering in the thousands. Furthermore, each case study represented almost a unique architecture with little agreement as to the dominant approach for building fault-tolerant systems.

In the intervening years between the first and second editions, fault tolerance has established itself as a major segment of the computing market. The number of deployed fault-tolerant systems is measured in the tens of thousands. Manufacturers are

developing the third- and fourth-generation systems so that we can look back at the evolutionary trajectory of these “fault-tolerant computer families.” There has also been a convergence with respect to the system architecture of preference. While the commercial systems still depend upon error detection and retry, the high-reliability systems rely upon triplication and voting, and the high-availability systems depend upon duplication and matching. The case studies have been reduced to nine in order for more space to be devoted to technical details as well as evolutionary family growth. Two case studies represent general-purpose commercial systems, three represent research and aerospace systems, and four represent high-availability systems. The approaches used in each of these three application areas can be compared and contrasted. Of special interest are the subtle variations upon duplication and matching used by all four high-availability architectures. In total, almost 50 percent of the material in the second edition is new with respect to the first edition.

This book has three audiences. The first is the advanced undergraduate student interested in reliable design; as prerequisites, this student should have had courses in introductory programming, computer organization, digital design, and probability. In 1983, the IEEE Computer Society developed a model program in computer science and engineering. This program consisted of nine core modules, four laboratory modules, and fifteen advanced subject areas. One of those advanced subject areas was “fault-tolerant computing.” Table P-1 illustrates how this book can be used in support of the module on fault-tolerant computing.

**TABLE P-1**  
*Mapping of the  
 book to modules  
 in Subject Area 20:  
 Fault-Tolerant  
 Computing, of the  
 1983 IEEE  
 Computer Society  
 Model  
 Undergraduate  
 Program in  
 Computer Science  
 and Engineering*

Module	Appropriate Chapter
1. Need for Fault-Tolerant Systems: Applications, fault avoidance, fault tolerance, levels of implementation elements	Ch. 1, Fundamental Concepts Ch. 3, Reliability Techniques
2. Faults and Their Manifestations: Sources, characteristics, effects, modeling	Ch. 2, Faults and Their Manifestations
3. Error Detection: Duplication, timeouts, parity checks	Ch. 3, Reliability Techniques
4. Protective Redundancy: Functional replication, information redundancy, temporal methods	Ch. 3, Reliability Techniques
5. Fault-Tolerant Software: N-version programming, recovery blocks, specification validation, proof, mutation	Ch. 3, Reliability Techniques
6. Measures of Fault Tolerance: Reliability models, coverage, availability, maintainability	Ch. 5, Evaluation Criteria Ch. 6, Financial Considerations
7. Case Studies	Introduction to Part II and further examples from Chapters 7 to 11 as time permits



The second audience is the graduate student seeking a second course in reliable design, perhaps as a prelude to engaging in research. The more advanced portions of Part I and the system examples of Part II should be augmented by other books and current research literature as suggested in Table P-2. A project, such as design of a dual system with a mean-time-to-failure that is an order of magnitude greater than nonredundant systems while minimizing life-cycle costs, would help to crystallize the material for students. An extensive bibliography provides access to the literature.

The third audience is the practicing engineer. A major goal of this book is to provide enough concepts to enable the practicing engineer to incorporate comprehensive reliability techniques into his or her next design. Part I provides a taxonomy of reliability techniques and the mathematical models to evaluate them. Design techniques are illustrated through the series of articles in Part II, which describe actual implementations of reliable computers. These articles were written by the system designers. The final chapter provides a methodology for reliable system design and illustrates how this methodology can be applied in an actual design situation (the DEC VAXft 310).

**Acknowledgments.** The authors wish to express deep gratitude to many colleagues in the fault-tolerant computing community. Without their contributions and assistance this book could not have been written. We are especially grateful to the authors of the papers who shared their design insights with us.

Special thanks go to Joel Bartlett (DEC-Western), Wendy Bartlett (Tandem), Thomas Bissett (DEC), Doug Bossen (IBM), William Bruckert (DEC), Richard Carr (Tandem), Kate Connolly (IBM), Stanley Dickstein (IBM), Dave Garcia (Tandem), Jim Gray (Tandem), Jeffrey P. Hansen (CMU), Robert Horst (Tandem), M.Y. Hsiao (IBM), Robert Jardine (Tandem), Doug Jewett (Tandem), Robert W. Kocsis (Jet Propulsion Lab.), Dan Lenoski (Tandem), Dix McGuire (Tandem), Bob Meeker (IBM), Dick Merrill (IBM), Larry Miller (IBM), Louise Nielsen (IBM), Les Parker (IBM), Frank Sera (IBM), Mandakumar Tendolkar (IBM), Liane Toy (AT&T), Wing Toy (AT&T), and Steven Webber (Stratus).

Jim Franck and John Shebell of Digital provided material and insight for Chapters 4 and 6 respectively. Jim Gray provided data on Tandem system failures that have been included in Chapter 2.

Jeff Hansen, David Lee, and Michael Schuette provide material on mathematical modeling, computer aids, and techniques. Comments from several reviewers and students were particularly helpful.

Special thanks are due to colleagues at both Carnegie-Mellon University and Digital Equipment Corporation (DEC) for providing an environment conducive to generating and testing ideas, especially Steve Director, Dean of the Engineering College, and Nico Habermann, Dean of the School of Computer Science. The entire staff of Digital Press provided excellent support for a timely production.

The professionalism of the staff at Technical Texts is deeply appreciated as they provided invaluable assistance throughout the production of the book. A special acknowledgment is also due Sylvia Dovner whose countless suggestions and attention to details contributed towards her goal of a "user friendly" book. The manuscript



**TABLE P-2**  
*Proposed structure  
 for graduate course*

Chapters	Augmentation
Ch. 1, Fundamental Concepts	
Ch. 2, Faults and Their Manifestations	Ross [1972] and/or Shooman [1968] for random variables, statistical parameter estimation ARINC [1964] for data collection and analysis
Ch. 3, Reliability and Availability Techniques	Appendix A, Peterson and Weldon [1972] for coding theory; Sellers, Hsiao, and Bearnson [1968b] for error-detection techniques <i>Proceedings of Annual IEEE International Symposium on Fault-Tolerant Computing</i> Special issues of the <i>IEEE Transactions on Computers</i> on Fault-Tolerant Computing (e.g., November 1971, March 1973, July 1974, May 1975, June 1976, June 1980, July 1982, 1986, April 1990) Special issues of <i>Computer</i> on Fault-Tolerant Computing (e.g., March 1980, July 1984, July 1990)
Ch. 4, Maintainability and Testing Techniques	Breuer and Friedman [1976] for testing; <i>Proceedings of Cherry Hill Test Conference</i> Special issues of <i>Computer</i> on Testing (e.g., October 1979) ARINC [1964] for maintenance analysis
Ch. 5, Evaluation Criteria	Ross [1972], Howard [1971], Shooman [1968], Craig [1964] for Markov models and their solutions
Ch. 6, Financial Considerations	Phister [1979]
<b>Part II</b>	October 1978 special issue of the <i>Proceedings of the IEEE</i>

provided many unforeseen “challenges,” and Sylvia’s perseverance was the glue that held the project together. That the book exists today is due in no small part to Sylvia’s efforts.

This book would not have been possible without the patience and diligence of Mrs. Laura Forsyth, who typed, retyped, and mailed the many drafts of the manuscript. Her activities as a “traffic controller” were vital to the project.

Finally, the support and understanding of our families is the central ingredient that made this book possible. From the occupation of the dining room table for weeks at a time for reorganizing text or double-checking page proofs to missing social events or soccer games, their patience and sacrifice over the last five years enabled the project to draw to a successful conclusion.

## REFERENCES\*

ARINC [1964]; Breuer and Friedman [1976]; Carter et al. [1964]; Cooper and Chow [1976]; Craig [1964]; Dickinson, Jackson, and Randa [1964]; Downing, Nowak, and Toumenoksa [1964]; Hamming [1950]; Howard [1971]; Moore and Shannon [1956]; Peterson and Weldon [1972]; Phister [1979]; Pierce [1965]; Ross [1972]; Sellers, Hsiao, and Bearnson [1968b]; Shannon [1948]; Shooman [1968]; von Neumann [1956]; Wilcox and Mann [1962].

\* For full citations of the shortened references at the end of each chapter, see References at the back of the book.

# CONTENTS

**Preface** xv

## **I THE THEORY OF RELIABLE SYSTEM DESIGN 1**

### **I FUNDAMENTAL CONCEPTS 3**

Physical Levels in a Digital System 5

Temporal Stages of a Digital System 6

Cost of a Digital System 18

Summary 21

References 21

### **2 FAULTS AND THEIR MANIFESTATIONS 22**

System Errors 24

Fault Manifestations 31

Fault Distributions 49

Distribution Models for Permanent Faults: The MIL-HDBK-217 Model 57

Distribution Models for Intermittent and Transient Faults 65

Software Fault Models 73

Summary 76

References 76

Problems 77

### **3 RELIABILITY TECHNIQUES 79**

Steven A. Elkind and Daniel P. Siewiorek

System-Failure Response Stages 80

Hardware Fault-Avoidance Techniques 84

Hardware Fault-Detection Techniques 96

Hardware Masking Redundancy Techniques 138

Hardware Dynamic Redundancy Techniques 169

Software Reliability Techniques 201

Summary 219

References 219

Problems 221

### **4 MAINTAINABILITY AND TESTING TECHNIQUES 228**

Specification-Based Diagnosis 229

Symptom-Based Diagnosis 260

Summary	268
References	268
Problems	269

## **5 EVALUATION CRITERIA 271**

Stephen McConnell and Daniel P. Siewiorek

Introduction	271
Survey of Evaluation Criteria: Hardware	272
Survey of Evaluation Criteria: Software	279
Reliability Modeling Techniques: Combinatorial Models	285
Examples of Combinatorial Modeling	294
Reliability and Availability Modeling Techniques: Markov Models	305
Examples of Markov Modeling	334
Availability Modeling Techniques	342
Software Assistance for Modeling Techniques	349
Applications of Modeling Techniques to Systems Designs	356
Summary	391
References	391
Problems	392

## **6 FINANCIAL CONSIDERATIONS 402**

Fundamental Concepts	402
Cost Models	408
Summary	419
References	419
Problems	420

## **II THE PRACTICE OF RELIABLE SYSTEM DESIGN 423**

Fundamental Concepts	402
General-Purpose Computing	424
High-Availability Systems	424
Long-Life Systems	425
Critical Computations	425

## **7 GENERAL-PURPOSE COMPUTING 427**

Introduction	427
Generic Computer	427
DEC	430
IBM	431
The DEC Case: <i>RAMP in the VAX Family</i>	433
Daniel P. Siewiorek	

The VAX Architecture	433
First-Generation VAX Implementations	439
Second-Generation VAX Implementations	455
References	484

**The IBM Case Part I: *Reliability, Availability, and Serviceability in IBM 308X and IBM 3090 Processor Complexes* 485**

Daniel P. Siewiorek

Technology	485
Manufacturing	486
Overview of the 3090 Processor Complex	493
References	507

**The IBM Case Part II: *Recovery Through Programming: MVS Recovery Management* 508**

C.T. Connolly

Introduction	508
RAS Objectives	509
Overview of Recovery Management	509
MVS/XA Hardware Error Recovery	511
MVS/XA Serviceability Facilities	520
Availability	522
Summary	523
Bibliography	523
Reference	523

**8 HIGH-AVAILABILITY SYSTEMS 524**

Introduction	524
AT&T Switching Systems	524
Tandem Computers, Inc.	528
Stratus Computers, Inc.	531
References	533

**The AT&T Case Part I: *Fault-Tolerant Design of AT&T Telephone Switching System Processors* 533**

W.N. Toy

Introduction	533
Allocation and Causes of System Downtime	534
Duplex Architecture	535
Fault Simulation Techniques	538
First-Generation ESS Processors	540
Second-Generation Processors	544
Third-Generation 3B20D Processor	551
Summary	572
References	573

The AT&T Case Part II: *Large-Scale Real-Time Program Retrofit Methodology in AT&T 5ESS® Switch* 574

L.C. Toy

5ESS Switch Architecture Overview 574

Software Replacement 576

Summary 585

References 586

The Tandem Case: *Fault Tolerance in Tandem Computer Systems* 586

Joel Bartlett, Wendy Bartlett, Richard Carr, Dave Garcia, Jim Gray, Robert Horst, Robert Jardine, Doug Jewett, Dan Lenoski, and Dix McGuire

Hardware 588

Processor Module Implementation Details 597

Integrity S2 618

Maintenance Facilities and Practices 622

Software 625

Operations 647

Summary and Conclusions 647

References 648

The Stratus Case: *The Stratus Architecture* 648

Steven Webber

Stratus Solutions to Downtime 650

Issues of Fault Tolerance 652

System Architecture Overview 653

Recovery Scenarios 664

Architecture Tradeoffs 665

Stratus Software 666

Service Strategies 669

Summary 670

## 9 LONG-LIFE SYSTEMS 671

Introduction 671

Generic Spacecraft 671

Deep-Space Planetary Probes 676

Other Noteworthy Spacecraft Designs 679

References 679

The *Galileo* Case: *Galileo Orbiter Fault Protection System* 679

Robert W. Kocsis

The *Galileo* Spacecraft 680

Attitude and Articulation Control Subsystem 680

Command and Data Subsystem 683

AACS/CDS Interactions 687

Sequences and Fault Protection 688

Fault-Protection Design Problems and Their Resolution 689

Summary 690

References 690

## **10 CRITICAL COMPUTATIONS 691**

Introduction 691

C.vmp 691

SIFT 693

*The C.vmp Case: A Voted Multiprocessor 694*

Daniel P. Siewiorek, Vittal Kini, Henry Mashburn, Stephen McConnel, and Michael Tsao

System Architecture 694

Issues of Processor Synchronization 699

Performance Measurements 702

Operational Experiences 707

References 709

*The SIFT Case: Design and Analysis of a Fault-Tolerant Computer for Aircraft Control 710*

John H. Wensley, Leslie Lamport, Jack Goldberg, Milton W. Green, Karl N. Levitt, P.M. Melliar-Smith, Robert E. Shostak, and Charles B. Weinstock

Motivation and Background 710

SIFT Concept of Fault Tolerance 711

The SIFT Hardware 719

The Software System 723

The Proof of Correctness 728

Summary 733

Appendix: Sample Special Specification 733

References 735

## **III A DESIGN METHODOLOGY AND EXAMPLE OF DEPENDABLE SYSTEM DESIGN 737**

### **11 A DESIGN METHODOLOGY 739**

Daniel P. Siewiorek and David Johnson

Introduction 739

A Design Methodology for Dependable System Design 739

*The VAXft 310 Case: A Fault-Tolerant System by Digital Equipment Corporation 745*

William Bruckert and Thomas Bissett

Defining Design Goals and Requirements for the VAXft 310 746

VAXft 310 Overview 747

Details of VAXft 310 Operation 756

Summary 766

**APPENDIXES 769****APPENDIX A 771***Error-Correcting Codes for Semiconductor Memory Applications:**A State-of-the-Art Review 771*

C.L. Chen and M.Y. Hsiao

Introduction 771

Binary Linear Block Codes 773

SEC-DEC Codes 775

SEC-DED-SBD Codes 778

SBC-DBD Codes 779

DEC-TED Codes 781

Extended Error Correction 784

Conclusions 786

References 786

**APPENDIX B 787***Arithmetic Error Codes: Cost and Effectiveness Studies for Application in Digital System Design 787*

Algirdas Avizienis

Methodology of Code Evaluation 787

Fault Effects in Binary Arithmetic Processors 790

Low-Cost Radix-2 Arithmetic Codes 794

Multiple Arithmetic Error Codes 799

References 802

**APPENDIX C 803***Design for Testability—A Survey 803*

Thomas W. Williams and Kenneth P. Parker

Introduction 803

Design for Testability 807

Ad-Hoc Design for Testability 808

Structured Design for Testability 813

Self-Testing and Built-In Tests 821

Conclusion 828

References 829

**APPENDIX D 831***Summary of MIL-HDBK-217E Reliability Model 831*

Failure Rate Model and Factors 831

Reference 833



**APPENDIX E 835*****Algebraic Solutions to Markov Models* 835**

Jeffrey P. Hansen

Solution of MTTF Models 837

Complete Solution for Three- and Four-State Models 838

Solutions to Commonly Encountered Markov Models 839

References 839

**GLOSSARY 841****REFERENCES 845****CREDITS 885****TRADEMARKS 890****INDEX 891**

# I THE THEORY OF RELIABLE SYSTEM DESIGN

Part I of this book presents the many disciplines required to construct a reliable computing system. Chapter 1 explains the motivation for reliable systems and provides the theoretical framework for their design, fabrication, and maintenance. It presents the hierarchy of physical levels into which a computer system is customarily partitioned and introduces the stages into which the life of a computer system is divided. Chapter 1 also provides a detailed discussion of two stages in a system's life: manufacturing and operation. Lastly, the chapter identifies several of the costs of ownership for a computer system and specifies some of the parameters that the designer must control to increase customer satisfaction.

Chapter 2 discusses errors and fault manifestations in a computer system. A review of applicable probability theory is presented as an aid to understanding the mathematics of the various fault distributions. Common techniques for matching empirical data to fault distributions, such as the maximum likelihood estimator, linear regression, and the chi-square goodness-of-fit test, are discussed. Chapter 2 also introduces methods for estimating permanent failure rates, including the MIL-HDBK-217 procedure, a widely used mathematical model of permanent faults in electronic equipment, and the life-cycle testing and data analysis approaches. It addresses the problem of finding an appropriate distribution for intermittent and transient errors by analyzing field data from computer systems of diverse manufacturers.

Chapter 3 deals with reliability techniques, or ways to improve the mean time to failure. It presents a comprehensive taxonomy of reliability and availability techniques. There is also a catalog of techniques, along with evaluation criteria for both hardware and software.

Chapter 4 deals with maintainability techniques, or ways to improve the mean time to repair of a failed computer system. It provides a taxonomy of testing and maintenance techniques, and describes ways to detect and correct sources of errors at each stage of a computer's life cycle. Specific strategies for testing during the manufacturing phase are discussed. The chapter explains several logic-level acceptance

江苏工业学院图书馆  
藏书章