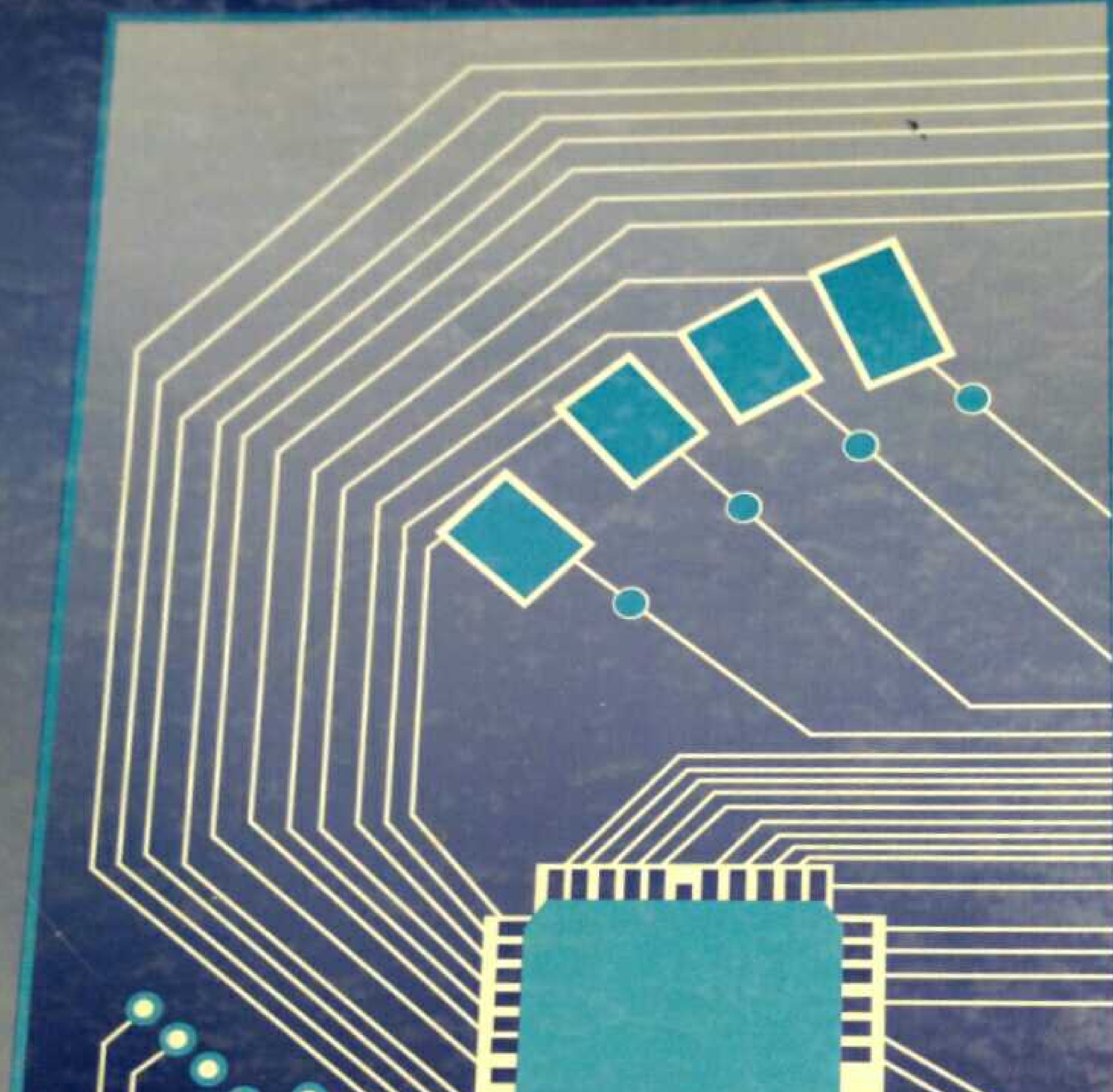
Michael J. Flynn

Computer Architecture

PIPELINED AND PARALLEL PROCESSOR DESIGN



Contents

Preface xv

Acknowledgments xvii

1 Architecture and Machines 1
1.1 Some Definitions and Terms 3
1.2 Interpretation and Microprogramming 5
1.3 The Instruction Set 9
1.4 Basic Data Types 13
1.5 Instructions 21
1.5.1 Classes of Operations 21
1.5.2 Instruction Mnemonics 25
1.5.3 General Machine Conventions 26
1.5.4 Branches 30
1.5.5 Register Sets and Addressing Modes 32
1.5.6 Instruction Code Examples 33
1.5.7 Other Instruction Set Issues 35
1.5.8 Program Size 37
1.6 Addressing and Memory 39
1.6.1 Process Addressing 39
1.6.2 System Addresses and Segmentation 41
1.6.3 Memory Space 43
1.7 Virtual to Real Mapping 45
1.8 Basic Instruction Timing 47
1.8.1 Examples of Well-mapped Machine Instruction Timing 49
1.8.2 Overlapped and Pipelined Processors 53
1.9 Conclusions 53
1.10 Historical Development of Computers 54
1.11 Annotated Bibliography 56
1.12 Problem Set 58

2 Time, Area, and Instruction Sets 63
2.1 Introduction 63
2.2 Time 64
2.2.1 The Nature of a Cycle 64
2.2.2 Partitioning Instruction Execution into Cycles 65
2.2.3 Clocking Overhead and Reliable Clocking 66
2.2.4 Pipelined Processors 70
2.2.5 Optimum Pipelining 70
2.2.6 Cycle Quantization 77
2.2.7 Wave Pipelining 79
2.3 Cost-Area 83
2.3.1 Area 84
2.3.2 Data Storage 93
2.4 Technology State of the Art 99
2.5 The Economics of a Processor Project: A Study 103
2.5.1 Phase 1: Development 106
2.5.2 Phase 2: Early Manufacturing 106
2.5.3 Phase 3: Production 107
2.5.4 Phase 4: All Good Things Must Come to an End 108
2.6 Instruction Sets: Processor Evaluation Metrics 109
2.6.1 Program Execution 110
2.6.2 Instruction Set Comparisons 112
2.6.3 Invariant Effects 117
2.6.4 Code Density 118
2.6.5 Role of Registers, Evaluation Stacks, and Data Buffers 124
2.7 Conclusions 132
2.8 Some Areas for Further Research 133
2.9 Data Notes 134
2.10 Annotated Bibliography 134
2.11 Problem Set 136

3 Data: How Programs Behave 141
3.1 Introduction 141
3.2 Instruction Usage 142
3.2.1 Data Categories 142
3.2.2 Format Distribution 144
3.2.3 Operation Set Distribution 144
3.3 Process Management 150
3.3.1 Procedure Calls: User State 150
3.3.2 Calls to the System 153

3.4 Breaks in Machine Execution 156
3.4.1 Instruction Run Length 156
3.4.2 Branches 158
3.4.3 Branch Target Distribution 161
3.4.4 Condition Code Testing 163
3.4.5 Move and Arithmetic Class Operations 164
3.4.6 Register-Based Addressing 167
3.4.7 Decimal and Character Operand Length 170
3.5 Conclusions 174
3.6 Some Areas for Further Research 174
3.7 Data Notes 175
3.8 Annotated Bibliography 176
3.9 Problem Set 177

4 Pipelined Processor Design **181**
4.1 Introduction 181
4.1.1 Evolution of a Computer Processor Family 182
4.1.2 Processor Design 185
4.1.3 Organization of the Chapter 185
4.2 Approaching Pipelined Processors 186
4.2.1 Examples of Pipeline Implementations 190
4.3 Evaluating Pipelined Processor Performance 193
4.4 Design of a Pipelined Processor 213
4.4.1 Cache Access Controller 214
4.4.2 Accounting for the Effect of Buffers in a Pipelined System 215
4.4.3 Buffer Design 216
4.4.4 Designing a Buffer for a Mean Request Rate 216
4.4.5 I-Buffers Designed for Maximum Request Rates 219
4.5 Branches 222
4.5.1 Branch Elimination 225
4.5.2 Branch Speedup 225
4.5.3 Branch Prediction Strategies 226
4.5.4 Branch Target Capture: Branch Target Buffers 236
4.6 Interlocks 240
4.6.1 Decoder and Interlocks 240
4.6.2 Bypassing 241
4.6.3 Address Generation Interlocks 242
4.6.4 Execution Interlocks and Interlock Tables 243
4.7 Run-On Delay 250

4.8 Miscellaneous Effects 251

4.8.1 Store in Instruction Stream Delay 251

4.9 Conclusions 259

4.10 Some Areas for Further Research 260

4.11 Data Notes 260

4.12 Annotated Bibliography 261

4.13 Problem Set 262

5 Cache Memory 265

5.1 Introduction 265

5.2 Basic Notions 266

5.3 Cache Organization 269

5.4 Cache Data 274

5.5 Adjusting the Data for Cache Organization 277

5.6 Write Policies 280

5.7 Strategies for Line Replacement at Miss Time 283

5.7.1 Fetching a Line 284

5.7.2 Line Replacement 286

5.8 Cache Environment 287

5.9 Other Types of Cache 291

5.10 Split I- and D-Caches 294

5.10.1 I- and D-Caches 294

5.10.2 Code Density Effects 297

5.11 On-Chip Caches 299

5.12 Two-Level Caches 302

5.12.1 Logical Inclusion 308

5.13 Write Assembly Cache 309

5.14 Cache References per Instruction 311

5.14.1 Instruction Traffic 311

5.14.2 Data Traffic 314

5.15 Technology-Dependent Cache Considerations 317

5.16 Virtual-to-Real Translation 322

5.16.1 Translation Lookaside Buffer (TLB) 323

5.17 Overlapping the T cycle in $V \rightarrow R$ Translation 325

5.17.1 Set Associative Caches 326

5.17.2 Virtual Caches 327

5.17.3 Real Caches Using Colored Pages 328

5.18 Studies 329

5.18.1 Actual Reference Traffic 331

5.19 Design Summary 336

5.19.1 Cache Evaluation Design Rules 336
5.19.2 Cache/TLB Excess CPI Design Rules 337
5.20 Conclusions 337
5.21 Some Areas for Further Research 338
5.22 Data Notes 338
5.23 Bibliography 340
5.24 Problem Set 341

6 Memory System Design 345
6.1 Introduction 345
6.2 The Physical Memory 349
6.2.1 The Memory Module 350
6.2.2 Error Detection and Correction 354
6.2.3 Memory Buffers 358
6.2.4 Partitioning of the Address Space 358
6.3 Models of Simple Processor–Memory Interaction 360
6.3.1 Memory Systems Design 361
6.3.2 Multiple Simple Processors 362
6.3.3 Hellerman's Model 363
6.3.4 Strecker's Model 363
6.3.5 Rau's Model 365
6.4 Processor–Memory Modeling Using Queueing Theory 365
6.4.1 Performance Models of Processor Memory Interactions 368
6.4.2 Arrival Distribution 369
6.4.3 Service Distribution 370
6.4.4 Terminology 371
6.4.5 Queue Properties 372
6.5 Open-, Closed-, and Mixed-Queue Models 374
6.5.1 The Open-Queue (Flores) Memory Model 376
6.5.2 Closed Queues 378
6.5.3 Mixed Queues 382
6.6 Waiting Time, Performance, and Buffer Size 384
6.6.1 Pipelined Processors 385
6.6.2 Designing a $M/M/1$ Buffer Given a Mean Queue Size . 389
6.6.3 Comparison of Memory Models 391
6.7 Review and Selection of Queueing Models 394
6.8 Processors with Cache 396
6.8.1 Fully and Partially Blocking Caches 397
6.8.2 Accessing a Line ($T_{\text{line access}}$) 39
6.8.3 Contention Time (T_{busy}) and Copyback Caches 40

6.8.4 I/O Effects 402
6.8.5 Performance Effects 403
6.8.6 Copyback Cache Study 404
6.8.7 Simple Write-Through Caches 407
6.8.8 Write-Through Cache Example 410
6.8.9 Shared Bus 411
6.8.10 Nonblocking Caches 413
6.8.11 Interleaved Caches 416
6.9 Conclusions 417
6.10 Some Areas for Further Research 418
6.11 Data Notes 419
6.12 Annotated Bibliography 419
6.13 Problem Set 420

7 Concurrent Processors 425
7.1 Introduction 425
7.2 Vector Processors 427
7.2.1 Vector Functional Units 428
7.2.2 Vector Instructions/Operations 437
7.2.3 Vector Processor Implementation 434
7.2.4 A Generic Vector Processor 436
7.3 Vector Memory 437
7.3.1 The Special Case of Vector Memory 437
7.3.2 Modeling Vector Memory Performance 443
7.3.3 Gamma (γ)-Binomial Model 446
7.3.4 Bypassing between Vector Instructions 449
7.4 Vector Processor Speedup 452
7.4.1 Basic Issues 452
7.4.2 Measures 455
7.5 Multiple-Issue Machines 456
7.6 Out-of-Order and Multiple-Instruction Execution 458
7.6.1 Data Dependencies 459
7.6.2 Representing Data Dependencies 461
7.6.3 Other Types of Dependencies 463
7.6.4 When and How to Detect Instruction Concurrency . . . 464
7.6.5 Two Scheduling Implementations 467
7.6.6 An Improved Scoreboard 475
7.6.7 Dealing with Out-of-Order Execution 486
7.6.8 Interleaved Caches 490
7.6.9 Branches and Speculative Execution 492

7.6.10 Adaptive Speculation 493
7.6.11 Results 494
7.7 Comparing Vector and Multiple-Issue Processors 499
7.7.1 Cost Comparison 499
7.7.2 Performance Comparison 502
7.7.3 Alternative Organizations 505
7.8 Conclusions 505
7.9 Some Areas for Further Research 506
7.10 Data Notes 507
7.11 Annotated Bibliography 507
7.12 Problem Set 508

8 Shared Memory Multiprocessors **511**
8.1 Basic Issues 511
8.2 Partitioning 513
8.3 Scheduling 519
8.3.1 Run-Time Scheduling Techniques 520
8.4 Synchronization and Coherency 523
8.5 The Effects of Partitioning and Scheduling Overhead 526
8.5.1 Grain Size and Overhead 531
8.6 Types of Shared Memory Multiprocessors 532
8.7 Multithreaded or Shared Resource Multiprocessing 533
8.8 Memory Coherence in Shared Memory Multiprocessors 538
8.9 Shared-Bus Multiprocessors 541
8.9.1 Snoopy Protocols 541
8.9.2 Bus-Based Models 551
8.10 Scalable Multiprocessors 555
8.11 Directory-Based Protocols 556
8.11.1 Directory Structure 557
8.11.2 Invalidate Protocols 559
8.11.3 Update Protocols 562
8.12 Evaluating Some Systems Alternatives 563
8.13 Interconnections 569
8.14 Static Networks 571
8.14.1 Links and Nodes 572
8.15 Dynamic Networks 574
8.16 Evaluating Interconnect Networks 578
8.16.1 Direct Static vs. Indirect Dynamic 578
8.16.2 Network Dimensionality and Link-Limited Network . 585
8.17 Hotspots and Combining 588

8.18 Other Characterizations of Multiprocessors 590
8.19 Conclusions 592
8.20 Some Areas for Further Research 594
8.21 Annotated Bibliography 594
8.22 Problem Set 595

9 I/O and the Storage Hierarchy **599**
9.1 The Role of I/O 599
9.2 Evolution of I/O Systems Organization 601
9.2.1 I/O Processors/Channels 603
9.2.2 I/O System Support for Multiprocessors 604
9.3 Design of Storage Systems 605
9.3.1 Disk Technology 605
9.3.2 The Disk Device 606
9.4 Simple I/O Transactions 609
9.4.1 Multiple Servers 614
9.4.2 Single-Server Low Population (n) 615
9.4.3 Disk Modeling 618
9.4.4 Multiprogramming Models and Inverted Servers 621
9.4.5 Improving I/O Response and Capacity 630
9.5 I/O Traffic and Virtual Memory Effects 632
9.5.1 Basic I/O Request Rate 633
9.5.2 Virtual Memory I/O Traffic 635
9.5.3 Disk Cache Buffers 636
9.5.4 Concurrent Disks 638
9.5.5 Clusters of Independent Disks 643
9.5.6 Striping 644
9.5.7 Disk Arrays 646
9.5.8 Composite Configurations 648
9.6 Some Practical Considerations 652
9.7 Redundancy in Disk Arrays 654
9.8 Conclusions 657
9.9 Some Areas for Further Research 658
9.10 Data Notes 658
9.11 Annotated Bibliography 659
9.12 Problem Set 660

10 Processor Studies **663**
10.1 The Baseline Mark II 663
10.1.1 Design Assumptions 664

10.1.2 Design Alternatives 665
10.1.3 Pipeline Timing Analysis 665
10.1.4 Pipeline Penalty Analysis 671
10.1.5 Cache and Memory Analysis 676
10.1.6 Cost-Performance Analysis 677
10.2 Area Performance Analysis of Processors 683
10.2.1 The Problem 683
10.2.2 Specifications 685
10.2.3 Assumptions 688
10.2.4 The Design 688
10.2.5 Analysis 704
10.3 Study Results 717
10.4 Conclusions 718

Appendix A DTMR Cache Miss Rates **719**
A.1 Basic DTMR 719
A.2 Associativity Adjustments 719
A.3 User + System 721
A.4 Transaction-Based Systems 722
A.5 Multiprogrammed (Warm Cache) Environment 728

Appendix B SPECmark vs. DTMR Cache Performance **741**

Appendix C Modeling System Effects in Caches **743**
C.1 Cold Start Cache 743
C.2 Cache Misses in Multiprogramming Environment 744

Appendix D New DRAM Technologies **747**
D.1 Typical Performance Enhancements 747
D.2 Enhanced DRAM 748
D.3 Synchronous DRAM 748
D.4 Cache DRAM 748
D.5 Rambus DRAM 748
D.6 Ramlink DRAM 749
D.7 Chip Level Summary 749

Appendix E $M/G/1$ Queues **751**

Appendix F Some Details on Bus-Based Protocols **755**

Bibliography **765**

Index **782**