# 计算机 系统结构基础

(影印版)

## ESSENTIALS OF COMPUTER ARCHITECTURE

Douglas E. Comer





## 计算机 系统结构基础

(影印版)

### ESSENTIALS OF COMPUTER ARCHITECTURE

Douglas E. Comer



高等教育出版社

#### 图字:01-2005-0964号

#### **Essentials of Computer Architecture**

Douglas E. Comer

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签。无标签者不得销售。

English reprint edition copyright @ 2005 by PEARSON EDUCATION ASIA LIMITIED and HIGHER

EDUCATION PRESS. (Essentials of Computer Architecture from Pearson Education's edition of the Work)

Essentials of Computer Architecture, le by Douglas E. Comer, Copyright © 2005.

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc. , publishing as Pearson Education, Inc. .

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Regions of Hong Kong and Macau).

原版 ISBN:0-13-149179-2

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macao SAR).

仅限于中华人民共和国境内(但不允许在中国香港、澳门特别 行政区和中国台湾地区)销售发行。

#### 图书在版编目(CIP)数据

计算机系统结构基础 = Essentials of Computer Arc hitecture/(美)科默(Comer, D. E.). —影印本. 北京:高等教育出版社,2005.12

ISBN 7-04-017816-8

I. 计... II. 科... III. 计算机体系结构 - 高等学校 - 教材 - 英文 IV. TP303

中国版本图书馆 CIP 数据核字(2005)第 132966 号

出版发行 高等教育出版社 购书热线 010 - 58581118址 北京市西城区德外大街 4号 800 - 810 - 0598免费咨询 邮政编码 100011 http://www.hep.edu.cn 机 010 - 58581000http://www.hep.com.cn 北京蓝色畅想图书发行有限公司 网上订购 http://www.landraco.com 印 北京外文印刷厂 http://www.landraco.com.cn  $787 \times 1092 \quad 1/16$ 版 2005年12月第1版 印 25 即 次 2005年12月第1次印刷 字 数 480 000 定 38.00元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。 版权所有 侵权必究

物料号 17816-00

stand how the mechanisms operate and the consequences for programmers. Finally Part 5 covers advanced topics like parallelism, pipelming, and performance.

students can learn by doing. Almongh most lab problem focus un programmane, sur

# Preface in the first few weaks in lab wiring a few gates on he first few weaks than the conjugate of the first few apent less than fifteen dollars per students purchase their own set of thins for under twenty deligns). The first few appendix in the first few appendix in the first few appendix in the first few appendix and the first few appendix appendix appendix the first few appendix ap

This book began when I was assigned to help salvage an undergraduate computer organization course. The course had suffered years of neglect: it had been taught by a series of professors, mostly visitors, who had little or no interest or background in digital hardware, and the curriculum had deteriorated to a potpourri of topics that were only loosely related to hardware architectures. In some semesters, students spent the entire class studying Boolean Algebra, without even the slightest connection to actual hardware. In others, students learned the arcane details of one particular assembly language, without a notion of alternatives.

Is a computer organization course worth saving? Absolutely! In many Computer Science programs, the computer organization course is the only time students are exposed to fundamental concepts that explain the structure of the computer they are programming. Understanding the hardware makes it possible to construct programs that are more efficient and less prone to errors. In a broad sense, a basic knowledge of architecture helps programmers improve program efficiency by understanding the consequences of programming choices. Knowing how the hardware works can also improve the programming process by allowing programmers to pinpoint the source of bugs quickly. Finally, graduates need to understand basic architectural concepts to pass job application tests given by firms like Intel and Microsoft.

One of the steps in salvaging our architecture course consisted in looking at text-books. We discovered the texts could be divided into roughly two types: texts aimed at beginning engineering students who would go on to design hardware, and texts written for CS students that attempt to include topics from compilers, operating systems, and (in at least one case) a complete explanation of how Internet protocols operate. Neither approach seemed appropriate for a single, introductory course on the subject. We wanted a book that (1) focused on the concepts rather than engineering details (because our students are not focused on hardware design); (2) explained the subject from a programmer's point of view, and emphasized consequences for programmers; and (3) did not try to cover several courses' worth of material. When no text was found, it seemed that the only solution was to create one.

The text is divided into five parts. Part 1 covers the basics of digital logic, gates, and data representation. We emphasize the representation chapter because notions of two's-compliment arithmetic and ranges of integer values are essential in programming. Parts 2, 3, and 4 cover the three essential areas of architecture: processors, memories, and I/O systems. In each case, the chapters give students enough background to under-

Preface

stand how the mechanisms operate and the consequences for programmers. Finally, Part 5 covers advanced topics like parallelism, pipelining, and performance.

An Appendix describes an important aspect of the course: a hands-on lab where students can learn by doing. Although most lab problems focus on programming, students should spend the first few weeks in lab wiring a few gates on a breadboard. The equipment is inexpensive (we spent less than fifteen dollars per student on permanent equipment; students purchase their own set of chips for under twenty dollars).

We have set up a web site to accompany the book at:

#### http://www.eca.cs.purdue.edu

Rajesh Subraman has agreed to manage the site, which contains a set of class presentation materials created by the author as well as a set created by Rajesh. We invite other instructors to contribute their materials.

The text and lab exercises have been used at Purdue; students have been extremely positive about both. We received notes of thanks for the text and course. For many students, the lab is their first experience with hardware, and they are enthusiastic.

My thanks to the many individuals who contributed to the book. Bernd Wolfinger provided extensive reviews and made several important suggestions about topics and direction. Dan Ardelean, James Cernak, and Tim Korb gave detailed comments on many chapters. Dave Capka reviewed early chapters. Rajesh Subraman taught from the book and provided his thoughts about the content. In the CS 250 class at Purdue, the following students each identified one or more typos in the manuscript: Nitin Alreja, Alex Cox, David Ehrmann, Roger Maurice Elion, Andrew Lee, Stan Luban, Andrew L. Soderstrom, and Brandon Wuest.

Finally, I thank my wife, Chris, for her patient and careful editing and valuable suggestions that improve and polish each book.

dual la gnislool ni bataianoa aruoa anuacintara ruo Douglas E. Comer ant la ano

books. We thicovered the texts could be divided into roughly two types: texts aimed at beginning engineering shoots, sould be on to design hardware, and texts withter

### **About The Author**

Dr. Douglas Comer has an extensive background in computer systems, and has worked with both hardware and software. Comer's work on software spans most aspects of systems, including compilers and operating systems. He created a complete operating system, including a process manager, a memory manager, and device drivers for both serial and parallel interfaces. Comer has also implemented network protocol software and network device drivers for conventional computers and network processors. Both his operating system, Xinu, and TCP/IP protocol stack have been used in commercial products.

Comer's experience with hardware includes work with discrete components, building circuits from logic gates, and experience with basic silicon technology. He has written popular textbooks on network processor architectures, and at Bell Laboratories, Comer studied VLSI design and fabricated a VLSI chip.

Comer is a Distinguished Professor of Computer Science at Purdue University, where he develops and teaches courses and does research on computer organization, operating systems, networks, and Internets. Comer has created innovative laboratories in which students can build and measure systems such as operating systems and IP routers; all of Comer's courses include hands-on lab work. He continues to consult and lecture at universities, industries, and conferences around the world.

In addition to writing a series of internationally acclaimed technical books on computer operating systems, networks, TCP/IP, and computer technologies, Comer serves as the editor-in-chief of the journal *Software — Practice and Experience*. He is a Fellow of the ACM, a Fellow of the Purdue Teaching Academy, and a recipient of numerous awards, including a Usenix Lifetime Achievement award.

Additional information can be found at:

www.cs.purdue.edu/people/comer

and information about Comer's books can be found at:

www.comerbooks.com

#### 

Preface Probables Types And Instruction Time 95 - rehydrental	xxi
Digital Lorus And Abstraction 29	
Print And Bridge 100 - 100	
Chapter 1 Introduction And Overview	1.54
Summary and the state of the st	2.5
1.1 The Importance Of Architecture 1	
1.2 Learning The Essentials 1	
1.3 Organization Of The Text 2	
1.4 What We Will Omit 3	
1.5 Terminology: Architecture And Design 3	
1.6 Summary 3	
Signed Integer 37	
	2.13
PART I Basics	
Florette strong a strong at the strong of the strong strong and the strong stro	
Chapter 2 Fundamentals Of Digital Logic	73.67
Party Agenty 1922 [active read specifically selected and another selection of the	
2.1 Introduction 7	
2.2 Electrical Terminology: Voltage And Current 7	
2.3 The Transistor 8	
2.4 Logic Gates 9	
2.5 Symbols Used For Gates 10	PART
2.6 Construction Of Gates From Transistors 11	
2.7 Example Interconnection Of Gates 12	
2.8 Multiple Gates Per Integrated Circuit 14	
2.9 The Need For More Than Combinatorial Circuits 15	
2.10 Circuits That Maintain State 15	
2.11 Transition Diagrams 16 The synthesis A samuely not	
2.12 Binary Counters 17	
2.13 Clocks And Sequences 18	
2.14 The Important Concept Of Feedback 20	
2.15 Starting A Sequence 22	
2.16 Iteration In Software Vs. Replication In Hardware 22	47
2.17 Gate And Chip Minimization 23	
2.18 Using Spare Gates 24	

2.19	Power Distribution And Heat Dissipation 24	
2.20	Timing 25	
2.21	Physical Size And Process Technologies 26	
2.22	Circuit Boards And Layers 27	
2.23	Levels Of Abstraction 27	
2.24	Summary 28	
hapte	r 3 Data And Program Representation	29
3.1	Introduction 29	
3.2	Digital Logic And Abstraction 29	
3.3	Bits And Bytes 30	
3.4	Byte Size And Possible Values 30 physical bank moltaubourne	
3.5	Binary Arithmetic 31	
3.6	Hexadecimal Notation 32	
3.7	Notation For Hexadecimal And Binary Constants 33	
3.8	Character Sets 34	
3.9	Unicode 35	
3.10	Unsigned Integers, Overflow, And Underflow 35	
3.11	Numbering Bits And Bytes 36	
3.12	Signed Integers 37	
3.13	An Example Of Two's Complement Numbers 38	
3.14		
3.15	Floating Point 40	
3.16	Special Values 42	
3.17	Range Of IEEE Floating Point Values 42	
3.18	Data Aggregates 42	
3.19	Program Representation 43	
3.20	Summary 43 Common Start Expelled Common Common Common Start Services	
	The Trendstor. 8	
	Logis Galler 9	
PART	II Processors	
	Contraction Of Guter From Transitions 14	
	Lample Interview on Of Gates 12	
Chapte	r 4 The Variety Of Processors And Computational Engines	47
	The Noed Fee More Than Conditionarial Caraline 150	
4.1	Introduction 47	
4.2	Von Neumann Architecture 47	
4.3	Definition Of A Processor 48	
4.4	The Range Of Processors 48	
4.5	Hierarchical Structure And Computational Engines 49	
4.6	Structure Of A Conventional Processor 51	
4.7	Definition Of An Arithmetic Logic Unit (ALU) 52	
	- Junion of the transfer Logic One (1100) 52	

4.8	Processor Categories And Roles 52	
4.9	Processor Technologies 54 88 moltamoral 359 dancas 60 molt	
4.10	Stored Programs 54 98 somewhat as allowed beautiful	
4.11	The Fetch-Execute Cycle 55 law and house and have account? Such and	
4.12	Clock Rate And Instruction Rate 56	
4.13	Control: Getting Started And Stopping 57 (3) to slaved land stocked 3	
4.14	Starting The Fetch-Execute Cycle 57	
4.15	Summary 58 120 00 themself to spiral and at all select	
	Operand Addressing Modes 92 22 cmg/laneary	
Chapte	r 5 Processor Types And Instruction Sets	61
5.1	Introduction 61	
5.2	Mathematical Power, Convenience, And Cost 61	
5.3	Instruction Set And Representation 62	
5.4	Opcodes, Operands, And Results 63	177
5.5	Typical Instruction Format 63	
5.6	Variable-Length Vs. Fixed-Length Instructions 63	7.3
5.7	General-Purpose Registers 64	7.4
5.8	Floating Point Registers And Register Identification 65	
5.9	Programming With Registers 65	7.6
5.10	Register Banks 66 Mg mainway Lank systlying	
5.11	Complex And Reduced Instruction Sets 67	
5.12	RISC Design And The Execution Pipeline 68	
5.13	Pipelines And Instruction Stalls 69 SOI autoMairan Shorestally	
5.14	Other Causes Of Pipeline Stalls 71 shoopsail (1) spotsmorth and	
5.15	Consequences For Programmers 71 T side W shapers 14 guide W	53.72
5.16	Programming, Stalls, And No-Op Instructions 72	
5.17	Forwarding 72 Will shooms like to maximaly	MAG
5.18	Types Of Operations 73 201 Showson MA Instrument Allegence A	
5.19	Program Counter, Fetch-Execute, And Branching 73	
5.20	Subroutine Calls, Arguments, And Register Windows 75	
5.21	An Example Instruction Set 76 Marin State Shows and Authority M.	21.15
5.22	Minimalistic Instruction Set 78	
5.23	The Principle Of Orthogonality 79 10 10 10 10 10 10 10 10 10 10 10 10 10	
5.24	Condition Codes And Conditional Branching 80	1227
5.25	Summary 80 Shilling Bellevilles Williams B. buch Endowes B. Landridges O.	7.22
	Consequences I or Progrationers 1113 methods contributed to	
	Therefore Ind. Advanced LLI responses	2.24
	er 6 Operand Addressing And Instruction Representation	83
6.1	Introduction 83	
6.2	Zero, One, Two, Or Three Address Designs 83	
6.3	Zero Operands Per Instruction 84	

6.4	One Operand Per Instruction 85	
6.5	Two Operands Per Instruction 85	
6.6	Three Operands Per Instruction 86	
6.7	Operand Sources And Immediate Values 86	
6.8	The Von Neumann Bottleneck 87	
6.9	Explicit And Implicit Operand Encoding 88	
6.10	Operands That Combine Multiple Values 89	
6.11	Tradeoffs In The Choice Of Operands 90	
6.12	Values In Memory And Indirect Reference 91	
6.13	Operand Addressing Modes 92	
6.14	Summary 93 chie mail: and bnA sequT sussessed 2 to	
Chapte	r 7 CPUs: Microcode, Protection, And Processor Modes	95
	Instruction Set And Representation 62	
7.1	Introduction 95	
7.2	A Central Processor 95	
7.3	CPU Complexity 96 managed all digital hands or digital addition.	
7.4	Modes Of Execution 97	
7.5	Backward Compatibility 97	
7.6	Changing Modes 98	
7.7	Privilege And Protection 99	
7.8	Multiple Levels Of Protection 99	
7.9	Microcoded Instructions 100 A resturant and the management	
7.10	Microcode Variations 102 & Waste motivated by A spatished	
7.11	The Advantage Of Microcode 102	
7.12	Making Microcode Visible To Programmers 103	
7.13	Vertical Microcode 103	
7.14	Horizontal Microcode 104	
7.15	Example Horizontal Microcode 105	
7.16	A Horizontal Microcode Example 107	
7.17	Operations That Require Multiple Cycles 108	
7.18	Horizontal Microcode And Parallel Execution 109	
7.19	Look-Ahead And High Performance Execution 110	
7.20	Parallelism And Execution Order 111	
7.21		
7.22	Conditional Branches And Branch Prediction 112	
7.23	Consequences For Programmers 113	
7.24	Summary 113	
	Coorand Addressing And Instruction Representations	

Chapte	r 8 Assembly Languages And Programming Paradigm	115
8.1	Introduction 115	
8.2	Characteristics Of A High-level Programming Language 115	
8.3	Characteristics Of A Low-Level Programming Language 116	
8.4	Assembly Language 117 24 moderates a group M 10 may	
8.5	Assembly Language Syntax And Opcodes 118	
8.6	Operand Order 120 Communication of State A State VI works	
8.7	Register Names 121 and a resolvent with the force	
8.8	Operand Types 122 This and pulsantes T growth burning	
8.9	Assembly Language Programming Paradigm And Idioms 122	
8.10	Assembly Code For Conditional Execution 123	
8.11	Assembly Code For A Conditional Alternative 124	
8.12	Assembly Code For Definite Iteration 124	
8.13	Assembly Code For Indefinite Iteration 125	
8.14	Assembly Code For Procedure Invocation   125	
8.15	Assembly Code For Parameterized Procedure Invocation 126	
8.16	Consequence For Programmers 127	
8.17	Assembly Code For Function Invocation 128	
8.18	Interaction Between Assembly And High-Level Languages 12	
8.19	Assembly Code For Variables And Storage 129	
8.20	Two-Pass Assembler 130	
8.21		
8.22	Summary 134 EE1 sessentials by 600 6500 and more	
	uran Of Memory Sign 135 781 on existent	
PART	TIII Memories The Little 221 Share State	
Chapte	er 9 Memory And Storage PZ   Springs M Makaza Mul-A my	137
9.1	Introduction 137	
9.2	Definition 137	
9.3	The Key Aspects Of Memory 138	
9.4	Characteristics Of Memory Technologies 138	
9.5	The Important Concept Of A Memory Hierarchy 140	
9.6	Instruction And Data Store 140	
9.7	The Fetch-Store Paradigm 141	
9.8	Summary 141 See Mark Space and See See Section 12	
	il Memory Term as age 154 ACL agest consistence of	
	er Translation Oir Address Mapping - 184 - vors 1890 voor	
	ing Arithmetic Calculation 167 [91] and a 2 L to a 2 L t	

Chapter	10 Physical Memory And Physical Addressing	143
10.1	Introduction 143	
10.2	Characteristics Of Computer Memory 143	
10.3	Static And Dynamic RAM Technologies 144	
10.4	Measures Of Memory Technology 145	
10.5	Density 146	
10.6	Separation Of Read And Write Performance 146	
10.7	Latency And Memory Controllers 146	
10.8	Synchronized Memory Technologies 147	
10.9	Multiple Data Rate Memory Technologies 148	
10.10	Examples Of Memory Technologies 148	
	Memory Organization 148 148 (September 2011) A 1867 Alexandra 1887	
	Memory Access And Memory Bus 149	
10.13	Memory Transfer Size 150 materials and plants and about a decree A	
10.14	Physical Addresses And Words 150	
10.15	Physical Memory Operations 150	
10.16	Word Size And Other Data Types 151	
10.17	An Extreme Case: Byte Addressing 151	
10.18	Byte Addressing With Word Transfers 152	
10.19	Using Powers Of Two 153 and has a salahatan and has a salahatan A	
	Byte Alignment And Programming 154001 wildings to the Court	
	Memory Size And Address Space 154	
	Programming With Word Addressing 155	
	Measures Of Memory Size 155	
	Pointers And Data Structures 156	
	Indirection And Indirect Operands 158	
	Memory Banks And Interleaving 158	
	Content Addressable Memory 159 approved by the special of the spec	
	Ternary CAM 160	
10.30	Summary 160	
10.50	Out of the second secon	
Chapter	11 Virtual Memory Technologies And Virtual Addressing	163
11.1	Introduction 163	
11.2	Definition 163	
11.3	A Virtual Example: Byte Addressing 164	
11.4	Virtual Memory Terminology 164	
11.5	An Interface To Multiple Physical Memory Systems 164	
11.6	Address Translation Or Address Mapping 166	
11.7	Avoiding Arithmetic Calculation 167	
11.8	Discontiguous Address Spaces 168	

11.9	Other Memory Organizations 169 hashad and bridge the	17797
11.10	Motivation For Virtual Memory 169 and saldon ground bound	12.22
11.11	Multiple Virtual Spaces And Multiprogramming 170 changes and	12.23
11.12	Multiple Levels Of Virtualization 171 and violent Agricum Mariella	12.24
11.13	Creating Virtual Spaces Dynamically 171 1 0 1710 2000 9 2000 9	12.23
11.14	Base-Bound Registers 172 2011 shoot ground sylmboxxA ts2	12.26
11.15	Changing The Virtual Space 172 23 May 2019 107 120 180 180 180 180 180 180 180 180 180 18	12,27
11.16	Virtual Memory, Base-Bound, And Protection 173	
11.17	Segmentation 174	
11.18	Demand Paging 175	
11.19	Hardware And Software For Demand Paging 175	THA
11.20	Page Replacement 176	
11.21	Paging Terminology And Data Structures 176	
11.22	Address Translation In A Paging System 177 Magin D. Magin City	
11.23	Using Powers Of Two 178	
11.24	Presence, Use, And Modified Bits 179	
11.25	Page Table Storage 180 THE TRANSPORT LANGUE BAR MORE	13.2
11.26	Paging Efficiency And A Translation Lookaside Buffer 181	13.3
11.27	Consequences For Programmers 182 000 mounts and	-9.11
11.28	Summary 183 900 responsibility and Indiana Inch Inivation	
	Self-Clocking Diate 210	13.6
	15 Programmed Attl Instinguistifusing 40% but as Loud-liv?	
Chapter	12 Caches And Caching S sughguord T and youstal southeast	8 18
	The Fundamental Idea Of Multiplexing 211 1922 postuborant	
12.1	Introduction 185 7 212 england luminal as a solved slighted	13.10
12.2	Definition 185 238 EIS ON 10 wall a rousson A	13.11
12.3	Characteristics Of A Cache 186 EIS CHARACTER STATE OF A Cache 186	
12.4	The Importance Of Caching 187	
12.5	Examples Of Caching 188	
12.6	Cache Terminology 188 assumed from A BUE back special Att	wiged
12.7	Best And Worst Case Cache Performance 189	
12.8	Cache Performance On A Typical Sequence 190	141
12.9	Cache Replacement Policy 190 215 us 6 A 10 months C	
	LRU Replacement 191 Hill Dresses Libra 2 20 June 191 AND 1915 LAND	
	Multi-level Cache Hierarchy 191 (Ches Madanto Cha A grant in good)	
	Preloading Caches 192 4 logodon and and and assure hamile	14.5
	Caches Used With Memory 192	
	TLB As A Cache 193	
	Demand Paging As A Form Of Caching 193 washing holized 19	14.6
	Physical Memory Cache 194 115 815 816 816 816	
	Write Through And Write Back 194	
	Cache Coherence 195 And Planting and Managham State Assas and	14.14
	L1, L2, and L3 Caches 196 244 055 211 A 1840 Stone-Hors-T	14.12
12.20	Sizes Of L1, L2, And L3 Caches 197 197 198 198 198 198 198 198	

V	Contents

12.21	Instruction And Data Caches 197	
	Virtual Memory Caching And A Cache Flush 198	
	Implementation Of Memory Caching 199	
	Direct Mapping Memory Cache 200 military 100 desired also made	
12.25	Using Powers Of Two For Efficiency 201	
12.26	Set Associative Memory Cache 202 STI MANIE STATE OF THE S	
12.27	Consequences For Programmers 203 sange landing and anagurals	
	Summary 204 EVI 1 - with sear Florid James - part , you all lustral V	
	Language And Mesons Controllers 146 471 mointaining Tag-	
	Demand Paging 175 751 vanablement vanish bacturelage	
PART	Horeboare And Software Fax Demand Baging MIREA was OV at VI	11.79
	Page Replacement 176, Its. responsives February to estimate	11.20
	Paging Terminology And Data Structures . [The leasting of veneral	
	13 Input/Output Concepts And Terminology	207
10.73	Using Powers Of Two 178 1821 will religiously recensive	
13.1	Introduction 207 And Mindiffied Birth 179 and 1997 and 19	11.24
13.2		
13.3		
13.4		
13.5	Serial And Parallel Data Transfers 209	
13.6		
13.7	Full-Duplex And Half-Duplex Interaction 210	
13.8	Interface Latency And Throughput 211 11000 bas and and all	Chapter
13.9	The Fundamental Idea Of Multiplexing 211	
	Multiple Devices Per External Interface 212 281 moitsubscount	
	A Processor's View Of I/O 213	
13.12	Summary 213 B 681 Cache 186 Thomas and The Importance Of Caching W. 187 Insection of the Importance Of Caching W.	
14.3	Examples Of Caching 188 851 gas-estadii gob toledi yannidi	
-	r 14 Buses And Bus Architectures 221 - Vgolonimus T. siloso	
14.1	Cache Performance On A Typical Sequence 190212   notational	
14.2	Definition Of A Bus 215 081 walled the meanages and and	
14.3	Processors, I/O Devices, And Buses 216 (21 managed and 3%)	01.21
14.4	Proprietary And Standardized Buses 216	
14.5	Shared Buses And An Access Protocol 217	
14.6	Multiple Buses   217 CONTANT AND BELLE MAINER	EL SI
14.7	A Parallel, Passive Mechanism 217 EPI SHOW A TANKLY	12.14
14.8	Physical Connections 217 and the DAM mod A & A subject Managed	
14.9	Bus Interface 218 All askay growald holistyll?	12.16
14.10	Address, Control, And Data Lines 219 1 say 4 bank daywood and W	12:17
14.11	The Fetch-Store Paradigm 220	
14.12	Fetch-Store Over A Bus 220	12,19
14.13	The Width Of A Bus 220 220 200 200 200 200 200 200 200 20	

	14.14	Multiplexing 221 Tempers 085 (AMC) research countries.	
	14.15	Bus Width And Size Of Data Items 222 125 gradual 13 15	15,23
	14.16	Bus Address Space 17223 and proposed stated valued back authorize	15,24
	14.17	Potential Errors 224 SES gransal mainstage	20.2571
		Address Configuration And Sockets 225	
	14.19	Many Buses Or One Bus 226	
	14.20	Using Fetch-Store With Devices 226	
	14.21	An Example Of Device Control Using Fetch-Store 226	rietquirl0
	14.22	Operation Of An Interface 227	
	14.23	Asymmetric Assignments 228	
	14.24	Unified Memory And Device Addressing 228	
	14.25	Holes In The Address Space 1230 to hand the same has a same and the same of the same and the same and the same of the same and the same	
	14.26	Address Map 230 Address Map 23	
	14.27	Program Interface To A Bus 231 1 865 2354496 30 1590 500 T	
	14.28	Bridging Between Two Buses 232 and Adjusted wall was a signed.	
	14.29	Main And Auxiliary Buses 232	
	14.30	Consequences For Programmers 234 100 was said who gained	
	14.31	Switching Fabrics 234 man 224 months and bearing	
	14.32	Summary 235 Management of the State of the S	
		Asymchronous Na Symchronous Popproprinted Forcetions 263.	
C	hapter	15 Programmed And Interrupt-Driven I/O vil beweit at ON	237
		Run-Time I/O Ubraries 2652 years also belong malanded	
	15.1	Introduction 237 ACC CAMPAGNACIA MARKET STREET STREET	
	15.2	1/O Paradigms 237 Total among the 200 self another 90 OM	
	15.3	Programmed I/O 238 882 amunipasqQ QVI Q res 3 stT	16,17
	15.4	Synchronization 238 800 hoseless O. Mad mater 2 and applicable 2	
	15.5	Polling 239 289 Polling 269 Paglicaing 269	91.000
	15.6	Code For Polling 239 075 grantfull to nontainemental	
	15.7	Control And Status Registers 241 IVS 10 Tellin A gradient's	
	15.8	Processor Use And Polling 241 272 bight at garantell	
	15.9	First, Second, And Third Generation Computers 242	1623
		Interrupt-Driven I/O 242 MA and a particular and a supplied to the supplied to	
		A Hardware Interrupt Mechanism 243 MTS gradual of restals.	
	15.12	Interrupts And The Fetch-Execute Cycle 243	
		Handling An Interrupt 244	
		Interrupt Vectors 245	
	15.15	Initialization And Enabling And Disabling Interrupts 246	
	15.16	Preventing Interrupt Code From Being Interrupted 246	
	15.17	Multiple Levels Of Interrupts 246	
	15.18	Assignment Of Interrupt Vectors And Priorities 247	
		Dynamic Bus Connections And Pluggable Devices 248	
		The Advantage Of Interrupts 249	
	15.21	Smart Devices And Improved I/O Performance 249	

15 22	Disast Massac (DMA) 250	
	Direct Memory Access (DMA) 250	
	Buffer Chaining 251	
	Scatter Read And Gather Write Operations 252	
	Operation Chaining 252	
13.20	Summary 253	
	16 A Programmer's View Of Devices, I/O, And Buffering	255
Chapter	To A Programmer's view of Devices, 170, And Buttering in	255
16.1	Introduction 255 352 automorphe A significant	
16.2	Definition Of A Device Driver 256 A 2010 St. M. A. COLORS M. Device Driver	
16.3	Device Independence, Encapsulation, And Hiding 256	
16.4	Conceptual Parts Of A Device Driver 257	
16.5		
16.6	Example Flow Through A Device Driver 258	
16.7	Queued Output Operations 259 SES result remitive hard mon	
16.8	Forcing An Interrupt 261 MS Transmission No. 12 Transmission No. 1	
16.9	Queued Input Operations 261	
	Devices That Support Bi-Directional Transfer 262	14.32
	Asynchronous Vs. Synchronous Programming Paradigm 263	
	Asynchrony, Smart Devices, And Mutual Exclusion 264	
	I/O As Viewed By An Application 264	
	Run-Time I/O Libraries 265	
	The Library/Operating System Dichotomy   266 TES   manualsustant	
	I/O Operations The OS Supports 267 200 TEL amelianted CIV	
	The Cost Of I/O Operations 268	
16.18	Reducing The System Call Overhead 268	
16.19	The Important Concept Of Buffering 269	
16.20	Implementation of Buffering 270	
	Flushing A Buffer 271	
16.22	Buffering On Input 272	4353
	Effectiveness Of Buffering 272	
	Buffering In An Operating System 273	
16.25	Relation To Caching 274	
	An Example: The Unix Standard I/O Library 274	
16.27	Summary 274	
	The first territory of the second sec	
	Initialization And Embling And Okability Interceptor Difference	
	Multiple Levels Of Interrupts 246	

### PART V Advanced Topics (Special Part of the Part V Advanced Topics (Special Part of the Part V Advanced Topics (Special Part of the Part V Advanced Topics (Special Part V Advanced Topics (Sp

Chapter	r 17 Parallelism	279
17.1	Introduction 279 and I amend a supply of the seg. Making the	19.4
17.2	Parallel And Pipelined Architectures 279 410 25th nontropyland	
17.3	Characterizations Of Parallelism 280	
17.4	Microscopic Vs. Macroscopic 280 zdrawski of dromatil had ON	
17.5	Examples Of Microscopic Parallelism 281	
17.6	Examples Of Macroscopic Parallelism 281	
17.7	Symmetric Vs. Asymmetric 282 metry? [9][[6][6][6][6][6][6][6][6][6][6][6][6][6	
17.8	Fine-grain Vs. Coarse-grain Parallelism 282 818 granual	
17.9	Explicit Vs. Implicit Parallelism 283	
17.10	Parallel Architectures 283	
17.11	Types Of Parallel Architectures (Flynn Classification) 283	
17.12	Single Instruction Single Data (SISD) 284	
17.13	Single Instruction Multiple Data (SIMD) 284 UTC months of the	
17.14	Multiple Instructions Multiple Data (MIMD) 286 and Instructions	20.2
17.15	Communication, Coordination, And Contention 288	
17.16	Performance Of Multiprocessors 290 of the Amanagement and	2014
17.17	Consequences For Programmers 292	
17.18	Redundant Parallel Architectures 294	20.6
17.19	Distributed And Cluster Computers 295 252 26230 26000000000000000000000000000000	
17.20	Summary 296 226 would housely back enthancement gringland	20.8
	Board-Level Architecture 325	
	Chip-Level Architecture 327	20.10
Chapte	r 18 Pipelining #54 gld A Clark Drain 128 Pipelining 1854	299
		20.12
18.1	Introduction 299 856 agranged by A. Andrews S.	
18.2	The Concept Of Pipelining 299	
18.3	Software Pipelining 301	
18.4	Software Pipeline Performance And Overhead 302	
18.5	Hardware Pipelining 303	
18.6	How Hardware Pipelining Increases Performance 303	
18.7	When Pipelining Can Be Used 306 and A well employed languist	
18.8	The Conceptual Division Of Processing 307	
18.9	Pipeline Architectures 307 Ltd. Impossible and Applied Applied.	
18.10		
18.11	Definition Of Superpipeline Architecture 308 house and sound	21.0
18.12	Summary 309	
	Introduction and account configuration 337	