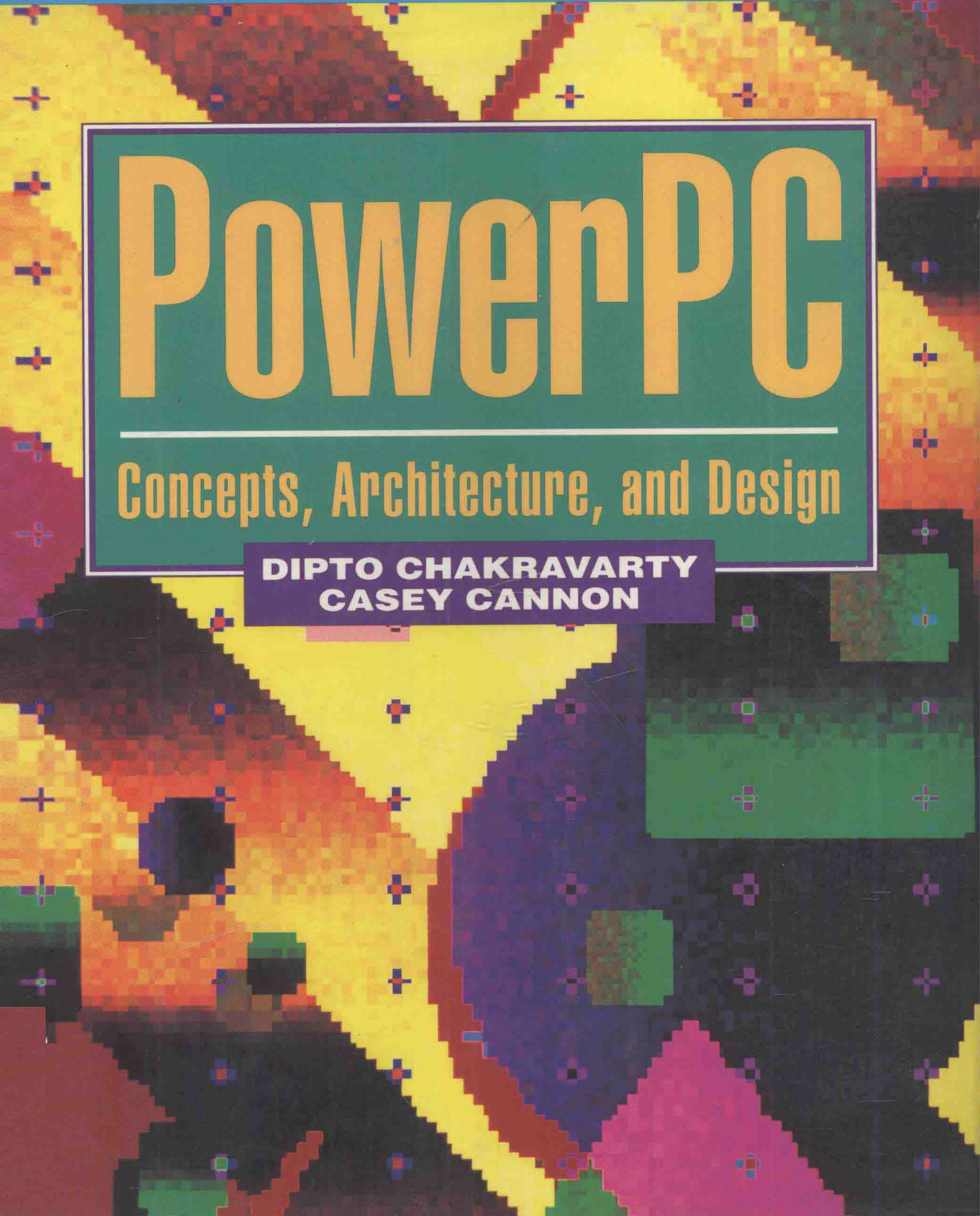
J. Ranade Workstation Series

PowerPC

Concepts, Architecture, and Design

DIPTO CHAKRAVARTY
CASEY CANNON



PowerPC

Concepts, Architecture, and Design

Dipto Chakravarty

Casey Cannon

McGraw-Hill, Inc.

New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

To Bapi
—D.C.
To my family for their enthusiasm
and to Don Meyer for his encouragement
—C.C.

Library of Congress Cataloging-in-Publication Data

Chakravarty, Dipto.
PowerPC : concepts, architecture, and design / Dipto Chakravarty, Casey Cannon.
p. cm. — (J. Ranade workstation series)
Includes bibliographical references.
ISBN 0-07-011192-8
1. PowerPC (Microprocessor) I. Title. II. Series.
QA76.8.P67C48 1994
004.165—dc20 94-19135
CIP

1 2 3 4 5 6 7 8 9 0 DOH/DOH 9 0 9 8 7 6 5 4

ISBN 0-07-011192-8

The sponsoring editor for this book was Jerry Papke and the production supervisor was Pamela A. Pelton. This book was set in Century Schoolbook by North Market Street Graphics.

Printed and bound by R. R. Donnelley & Sons Company.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The author and publisher have exercised care in preparing this book and the programs contained in it. They make no representation, however, that the programs are error-free or suitable for every application to which the reader may attempt to apply them. The author and publisher make no warranty of any kind, expressed or implied, including the warranties of merchantability or fitness for a particular purpose, with regard to these programs or the documentation or theory contained in this book, all of which are provided "as is." The author and publisher shall not be liable for damages in amount greater than the purchase price of this book, or in any event for incidental or consequential damages in connection with, or arising out of the furnishing, performance, or use of these programs or the associated descriptions or discussions.

Readers should test any program on their own systems and compare results with those presented in this book. They should then construct their own test programs to verify that they fully understand the requisite calling conventions and data formats for each of the programs. Then they should test the specific application thoroughly.

J. Ranade Workstation Series

CHAKRAVARTY • *Power RISC System / 6000: Concepts, Facilities, and Architecture,* 0-07-011047-6

CHAKRAVARTY/CANNON • *PowerPC: Concepts, Architecture, and Design,* 0-07-011192-8

DEROEST • *AIX for RS / 6000: System and Administration Guide,* 0-07-036439-7

HENRY/GRAHAM • *Solaris 2.X System Administrator's Guide,* 0-07-029368-6

JOHNSTON • *OS / 2 Connectivity and Networking: A Guide to Communication Manager / 2,* 0-07-032696-7

LAMB • *MicroFocus Workbench and Toolset Developer's Guide,* 0-07-036123-3

LEININGER • *UNIX Developer's Tool Kit,* 0-07-911646-9

LOCKHART • *OSF DCE: Guide to Developing Distributed Applications,* 0-07-911481-4

RANADE/ZAMIR • *C++ Primer for C Programmers,* 2/e, 0-07-051487-9

SANCHEZ/CANTON • *Graphics Programming Solutions,* 0-07-911464-4

SANCHEZ/CANTON • *High Resolution Video Graphics,* 0-07-911646-9

SANCHEZ/CANTON • *PC Programmer's Handbook,* 2/e, 0-07-54948-6

WIGGINS • *The Internet for Everyone: A Guide for Users and Providers,* 0-07-067019-8

To order or receive additional information on these or any other
McGraw-Hill titles, in the United States please call 1-800-822-8158.
In other countries, contact your local McGraw-Hill representative.

BC14BCZ

Trademarks

Alpha and VAX are trademarks of Digital Equipment Corporation.
AFS and Transarc are trademarks of Transarc Corporation.
AIXwindows and AIXwindows Interface Composer/6000 are trademarks of International Business Machines Corporation.
AIX 3278/79 Emulation/6000 is a trademark of International Business Machines Corporation.
Apple, Appletalk, and Finder are trademarks of Apple Computer Corp.
C Set ++ is a trademark of International Business Machines Corporation.
CICS is a trademark of International Business Machines Corporation
Ethernet is a trademark of Xerox Corporation.
GL and Graphics Library are trademarks of Silicon Graphics, Inc.
HP is a registered trademark of Hewlett-Packard.
HP Precision Architecture and Laserjet are trademarks of Hewlett-Packard.
InfoExplorer is a trademark of International Business Machines Corporation.
Macintosh is a trademark of Apple Computer Corp.
Microsoft is a trademark of Microsoft Corporation.
Motorola and Motorola 88110 are trademarks of Motorola, Inc.
Micro Channel is a registered trademark of International Business Machines Corporation.
MIPS is a registered trademark of MIPS Computer Systems, Inc.
MS-DOS is a trademark of Microsoft Corporation.
NETBIOS is a trademark of International Business Machines Corporation.
NFS and SunOS are trademarks of Sun Microsystems, Inc.
OPEN LOOK is a trademark of AT&T.
Open Software Foundation, OSF, OSF/Motif and Motif are trademarks of Open Software Foundation, Inc.
Pentium is a trademark of Intel Corporation.
POSIX is a trademark of the Institute of Electrical and Electronic Engineers (IEEE).
POWER Architecture is a trademark of International Business Machines Corporation.
PowerOpen is a trademark of International Business Machines Corporation.
PowerPC, PowerPC Architecture, POWERserver, POWERstation are trademarks of International Business Machines Corporation.
PowerPC 601, PowerPC 603, PowerPC 604, and PowerPC 620 are trademarks of International Business Machines Corporation.
RISC System/6000 is a trademark of International Business Machines Corporation.
RT and RT PC are trademarks of International Business Machines Corporation.
SPARC is a registered trademark of SPARC International, Inc.
Taligent is a registered trademark of Taligent, Inc.
UNIX is licensed by and is a registered trademark of UNIX System Laboratories, Inc.
Wabi is a trademark of Sun Microsystems, Inc.
X Window System is a trademark of Massachusetts Institute of Technology
X11 is a trademark of Massachusetts Institute of Technology.
XPG3 is a trademark of X/OPEN Company Limited.

Preface

This book is a general-purpose reference for the computer professionals who wish to understand the PowerPC technology, which has evolved as a result of the IBM-Motorola-Apple alliance. The text is designed to serve as a single source of reference about the PowerPC hardware and its operating environments. A layer-by-layer introduction of the hardware, middleware, and software options unveils the diverse capabilities and features of this revolutionary technology. The subtitle, *Concepts, Architecture, and Design,* is quite appropriate, as the book contains a comprehensive overview of the hardware and the software *concepts* from both user as well as system perspectives. The text introduces the hierarchical *architecture* of the PowerPC microprocessor and explains the *design* rationales for the facilities and features that enable PowerPC to achieve the paramount level of performance.

Architecture and implementation of a computer are two distinct entities. Perhaps the most popular distinction between the terms was made in a *Communications of the ACM* journal.*

Computer architecture

> . . . is defined as the attributes and behavior of a computer as seen by a machine language programmer. This definition includes the instruction set, instruction formats, operations codes, addressing modes, and all register and memory locations that may be directly manipulated by a machine language programmer.

Implementation

> . . . is defined as the actual hardware structure, logic design, and data path organization of a particular embodiment of the architecture.

Thus, architecture is a definition that describes the behavior of all possible implementations, as compared to implementation that typically references a single microprocessor. The discussions of PowerPC architecture and its implementations have been kept separate for maximum benefit to the reader. A survey of operating systems and user interfaces has been provided to present to the reader with a system-level picture of the PowerPC-based computer systems.

* *Communications of the ACM,* vol. 36, no. 2, p. 33, February 1993.

WHY THIS BOOK

The publicity generated concerning the PowerPC microprocessor has resulted in the dissemination of a wide variety of information among vendors and developers in the computer industry. What has not been disseminated is a reference that ties all the aspects of this nascent technology together in a comprehensive source of reference. *Hence, this book.*

OBJECTIVES OF THIS BOOK

The first objective of this book is to describe the principles of the PowerPC architecture (and its implementations), which evolved out of the POWER architecture. Introduced in 1990, the POWER technology used in the RISC System/6000 product line acquired a quick reputation for itself in the marketplace by virtue of its advanced RISC-based design and achievable level of performance. PowerPC is a flexible derivative of the POWER architecture and shares a wide spectrum of traits with its parent architecture.

The second objective of this book is to depict a system-level picture of computers based on the PowerPC processor with emphasis on the operating systems, software development tools, standards, and user interfaces. An array of technologies is available today, in terms of hardware as well as software, that can be optimized by using PowerPC as its core. With a proliferation of PowerPC-based computer systems in the marketplace, end users will be faced with a choice of whether to opt for a computer that is PowerPC reference platform compliant or a computer that uses the PowerPC as its core. Both types of systems offer the power of the PowerPC microprocessor, the difference being in the varying degrees of compatibility that exist among the applications.

USES OF THIS BOOK

The intended use of this book is threefold:

- It can be used by the computer professionals working on or transitioning to the PowerPC-based development environment.
- It can be read by the general audience of the computer community wishing to get acquainted with PowerPC technology. The material delves into adequate depth to serve a novice as well as a knowledgeable user.
- It can be used as supplemental reading material in a computer system architecture course.

ORGANIZATION OF THIS BOOK

The first part of the book introduces the PowerPC in light of RISC technology. The second part explains the PowerPC architecture and discusses its available and planned implementations, including a comparative study with the POWER offerings to explain how POWER developed into the PowerPC. The third part of the book covers the user interfaces, standards, and tools. It also discusses the several operating systems that can/will run on the PowerPC. The final chapter wraps up the concepts by giving a tutorial on how to go about building one's own PowerPC platform. The contents of each of the parts are stand-alone and can be studied individually.

The book is organized into 12 chapters:

1. Presenting the PowerPC
2. RISC Technology
3. Architectural Definition
4. Processor Implementations
5. User Interfaces
6. Choice of Operating Systems
7. Development Tools
8. Supported Standards
9. Design of AIX: A PowerOpen Implementation
10. AIX Process Subsystem Internals
11. AIX File, Memory, and I/O Subsystem Internals
12. What You Need to Build a PowerPC

Chapter 1 introduces the PowerPC, discusses its evolution through the formation of the IBM-Motorola-Apple alliance, compares its standing with the Pentium, and addresses some of the important highlights such as the PowerOpen Environment, the application binary interface (ABI), and the application programming interface (API) definitions. Chapter 2 discusses the RISC technology in light of its unique traits, performance tradeoffs with CISC, pipelined implementation of the execution units, and the significance of reduced instruction set cycles. Chapter 3 explains how the layered architecture defines the varying degree of compatibility from an instruction set level, to the virtual environment level, all the way up to the operating environment level. Chapter 4 describes the implementations of the PowerPC architecture, such as the 601, 603, 604, and the 620, while contrasting them with some of the POWER implementations, the RS 1, RS .9, and RSC. Chapter 5 discusses the functionality and illustrates the leading industry standards of user interfaces including the Common Desktop Environment, Wabi, X Windows, and Macintosh Application Services.

Chapter 6 reviews the PowerOpen Application Binary Interface and then highlights five of the 32-bit operating systems that the PowerPC platform is intended to support, including Taligent, Windows NT, Solaris, AIX, and Workplace OS. Chapter 7 provides a broad overview of the most widely used development tools for UNIX operating systems, including discussions of (among others) the XL C optimizing compiler, assembler, and debuggers. Chapter 8 discusses the compatibility, portability, and interoperability standards for the PowerPC, followed by an overview of the interconnectivity functionalities of the PowerPC. Chapters 9, 10, and 11 cover in detail the PowerOpen-compliant AIX operating system, which is based on the COSE (Common Open Software Environment) version of UNIX. Chapter 9 presents the design of AIX, with in-depth discussions on components of the kernel, structural layout and characteristic features of the kernel, internal representation of files, related kernel tables, interprocess communication mechanisms, and allied data structures. Chapter 10 explains AIX process management principles, with emphasis on process structure, process state, context switching, scheduling principles, affiliated kernel structures and their positioning in the kernel address space, the art of monitoring processes by traversing through the *kmem* (running kernel's memory), and handling of threads. Chapter 11 discusses the file, memory, I/O, and device subsystems of the AIX kernel. It begins with a detailed discussion on the AIX file system in light of its memory mapped files, journaled file system, and the logical volume manager. The memory architecture topics include the addressability of the segmented memory, followed by the virtual memory management, page replacement, and memory load control schemes. The I/O subsystem topics include asynchronous I/O and I/O pacing, followed by the device subsystem, which discusses device drivers and the object data manager. In conclusion, Chapter 12 wraps up the concepts, architecture, and design of the PowerPC by providing a description of the devices and interfaces that are recommended for designing and building a PowerPC based computer system.

The content of Chapter 1 serves as an introduction to PowerPC for everyone. The material in Chapters 2, 3, and 4 will be of maximum benefit to hardware engineers who need to know about the registers and the architectural traits of the PowerPC microprocessor. The information in Chapters 5, 6, 7, and 8 are meant for end users and system integrators/designers. Chapters 9, 10, and 11 are geared for UNIX gurus who wish to understand how the internals of the PowerOpen compliant AIX operating system works. The content of Chapter 12 provides an account of the aspects to be considered when building one's own PowerPC-based computer system.

In conclusion, this book can be thought of as a single source of information about all technical aspects of the PowerPC. Professionals requiring an immersion training in PowerPC, as well as those keen on gaining an insight into the internals of this complex system, will benefit from this book.

A few caveats need to be mentioned. No attempts have been made to cover details of implementation-specific hardware components or release-specific software components. Such attributes are likely to change over a period of time. For an implementation-specific dependency of a microprocessor, or a release-specific dependency of an operating system or software component, one is encouraged to refer to the corresponding product reference manuals. Although we have avoided predicting the future development of the hardware and software, trends in many of the characteristics are obvious. In that case, this book will serve as the baseline technical reference for future products based on the PowerPC architecture.

The IBM-Motorola-Apple alliance has resulted in the birth of the PowerPC, and, consequently, has brought the RISC technology to the desktop computing world. By blending together the cost-performance and scalable aspects of the architecture, along with the interoperable software base, the PowerPC has made the biggest impact in the personal computer industry since the original Intel-based personal computer itself.

Acknowledgments

The genesis of PowerPC has generated an enthusiastic community worldwide. The technology, which evolved as a result of the IBM-Motorola-Apple alliance, is rapidly penetrating the computer industry's relentless pursuit for cost-performance computing.

The inspiration and support for writing this book came from our colleagues and friends too numerous to mention. Of the many individuals who helped us author this book, we would like to express our gratitude to our colleagues at IBM, Motorola, and Apple who willingly answered questions about this emerging subject.

We express our sincere thanks and gratitude to each of the following people whose timely help enabled us to march to an ambitious production schedule.

Gary Leikam provided timely guidance and review of the standards encompassing the PowerPC technology.

Art Adkins provided ongoing advice and help whenever we needed. He also reviewed the information pertaining to PowerPC reference platform specification and operating systems.

Morris Grove reviewed the material on architectural definition of the PowerPC and provided useful suggestions regarding the format of the material.

John McKeeman reviewed the material on PowerPC-based embedded controllers.

Paul Lugo, Dave Thompson, and Mark Wieland reviewed the material on Wabi, Macintosh Application Services, and Common Desktop Environment.

Richard Swann helped us with the material on Windows NT.

Mark Hevesh and Sanjoy Chatterji perused the material on PowerOpen Association and PowerOpen Environment.

Jim Shaffer reviewed the material on AIX process subsystem internals and helped fine-tune the topics on processes and threads.

Our warmest thanks to the team at North Market Street Graphics, especially Virginia Carroll, Anne Friedman, Christine Furry, and Nathanael Waite,

who adhered to a tight schedule to get this book out on time. A special mention should be made of editors Gerald Papke, Jay Ranade, and Rachel Hirshfield at McGraw-Hill.

A recognition or an acknowledgment will not be enough, but it will have to do for my wife, Aloka Chakravarty, without whose support this book would not have happened.

—*D.C.*

Special thanks to my management team—Pat Birdsall, Mark Akers, and Tom Cross—for their support.

—*C.C.*

Contents

Preface xv

Acknowledgments xix

Chapter 1. Introduction 1

1.1 Overview of the PowerPC 1

1.2 PowerOpen Environment 2

1.3 PowerOpen Association 3

1.4 POWER to PowerPC Architecture 4

1.5 POWER-PowerPC Deltas 5

1.6 PowerPC Performance 7

1.6.1 Why the RISC-based PowerPC Is Faster 8

1.6.2 601 Microprocessor 8

1.6.3 603 Microprocessor 8

1.6.4 604 Microprocessor 9

1.6.5 620 Microprocessor 10

1.6.6 601 versus the Pentium 11

1.7 System Environment Overview 13

1.7.1 PowerOpen ABI and API 14

1.7.2 International Language Support 14

1.8 Summary 15

Part 1 Hardware

Chapter 2. RISC Technology 19

2.1 Evolution of RISC 19

2.2 RISC Characteristics 20

2.2.1 Load-Store Architecture 20

2.2.2 Fixed-length Instructions 20

2.2.3 Hardwired Control 21

2.2.4 Fused Instructions 21

2.2.5 Pipelined Implementation 22

2.3 Superscalar Implementation 27

2.4 RISC/CISC Tradeoffs 28

2.5 Effect of Pipelining 28

2.6 Reduced Instruction Set Cycles 29

2.7 Summary 29

Chapter 3. Architectural Definition 31

3.1 Evolutionary Road Map of PowerPC 31
3.2 The PowerPC Instruction Set 33
3.3 The 32-bit PowerPC Architecture 34
3.3.1 Instruction Set Architecture 34
3.3.2 Virtual Environment Architecture 41
3.3.3 Operating Environment Architecture 45
3.4 The 64-bit PowerPC Architecture 53
3.4.1 Instruction Set Architecture 53
3.4.2 Virtual Environment Architecture 56
3.4.3 Operating Environment Architecture 59
3.5 Timer Facilities 67
3.6 Summary 68

Chapter 4. Processor Implementations 69

4.1 Understanding the Common CPU Model 69
4.2 The POWER RS 1 Microprocessor 70
4.2.1 Organization 70
4.2.2 Instruction Cache 72
4.2.3 Data Cache 73
4.2.4 Fixed-point Unit 73
4.2.5 Floating-point Unit 74
4.2.6 Packaging 75
4.3 The POWER RS .9 Microprocessor 76
4.3.1 Organization 77
4.3.2 Instruction Cache 77
4.3.3 Data Cache 77
4.3.4 Fixed-point Unit 79
4.3.5 Floating-point Unit 79
4.3.6 Packaging 79
4.4 The POWER RSC Microprocessor 80
4.4.1 Organization 80
4.4.2 Cache 81
4.4.3 Branch Processing and Instruction Fetch Unit 82
4.4.4 Instruction Queue and Dispatch Unit 83
4.4.5 Fixed-point Unit 83
4.4.6 Floating-point Unit 83
4.4.7 Memory Management Unit 83
4.4.8 Memory Interface Unit 84
4.4.9 Sequencer Unit 84
4.4.10 Packaging 84
4.5 The PowerPC 601 Microprocessor 84
4.5.1 Pipelines 85
4.5.2 Organization 86
4.5.3 Instruction Queue and Dispatch Unit 87
4.5.4 Instruction Fetch Unit 87
4.5.5 Branch Processing Unit 87
4.5.6 Fixed-point Unit 88
4.5.7 Floating-point Unit 88
4.5.8 Memory Management Unit 88
4.5.9 Cache 89
4.5.10 Memory Queue 90
4.5.11 Bus Interface 90
4.5.12 Sequencer Unit 90

4.5.13 Multiprocessor Capabilities 90
4.5.14 Packaging 91
4.6 The PowerPC 603 Microprocessor 92
4.6.1 Pipelines 93
4.6.2 Organization 94
4.6.3 Instruction and Data Caches 95
4.6.4 Instruction Fetch and Branch Unit 95
4.6.5 Dispatcher Unit 96
4.6.6 Completion/Exception Unit 96
4.6.7 Fixed-Point Unit 97
4.6.8 Floating-Point Unit 97
4.6.9 Load/Store Unit 97
4.6.10 System Unit 97
4.6.11 Bus Interface Unit 98
4.6.12 Packaging and Power Management 98
4.7 The PowerPC 604 Microprocessor 98
4.7.1 Pipelines 98
4.7.2 Organization 98
4.7.3 Packaging 99
4.8 The PowerPC 620 Microprocessor 99
4.9 PowerPC Embedded Processors 100
4.9.1 The Embedded Controller 403 102
4.10 Summary 103

Part 2 Software

Chapter 5. User Interfaces 107

5.1 Common Desktop Environment 108
5.1.1 Overview 108
5.1.2 Getting Started with Common Desktop Environment 108
5.1.3 Common Desktop Environment Services 111
5.2 Wabi 118
5.2.1 Overview 118
5.2.2 Wabi Capabilities and Functions 119
5.2.3 Getting Started with Wabi 120
5.3 X Window System 124
5.3.1 AIXwindows Environment 126
5.3.2 AIXwindows 3-D 127
5.3.3 AIXwindows Interface Composer 127
5.4 Macintosh Application Services 128
5.4.1 Overview 128
5.4.2 Capabilities and Functions 128
5.4.3 Getting Started with Macintosh Application Services 129
5.5 Summary 131

Chapter 6. Operating Systems 133

6.1 PowerOpen Application Binary Interface 134
6.2 AIX 135
6.2.1 AIX Personal Productivity Client Configuration 135
6.2.2 Operating Environment 136
6.2.3 End-user Environment 140
6.2.4 Optimizing AIX 145

6.3 Taligent 147
6.3.1 Microkernel Paradigm 148
6.3.2 Operating Environment 149
6.3.3 End-user Environment 150
6.4 Solaris 150
6.4.1 Operating Environment 151
6.4.2 End-user Environment 152
6.5 Windows NT 153
6.5.1 End-user Environment 154
6.5.2 Operating Environment 155
6.6 Workplace OS 160
6.6.1 Operating Environment 160
6.6.2 End-user Environment 161
6.7 Summary 161

Chapter 7. Development Tools 163

7.1 Compilers 163
7.1.1 XL C Compiler 163
7.1.2 C Set ++ Compiler 165
7.1.3 AIX XL FORTRAN and Pascal Compilers 165
7.2 Assembler 166
7.3 Debuggers 167
7.3.1 adb 167
7.3.2 fsdb 168
7.3.3 dbx and xde 170
7.3.4 Kernel Debug Program 172
7.3.5 Trace Facility 174
7.4 Source Code Analysis Tools 175
7.4.1 lint 175
7.4.2 cflow 175
7.4.3 cxref 175
7.5 Lexical Analyzer—lex 176
7.6 Parser Generator—yacc 176
7.7 Pattern Matching Language 176
7.7.1 awk 176
7.7.2 sed 177
7.8 Macro Processor—m4 178
7.9 Program Modules Management 178
7.9.1 make 178
7.9.2 imake 179
7.10 Source Code Control System 180
7.11 AIX Performance Tools 181
7.11.1 Monitoring Tools 181
7.11.2 Analysis Tools 188
7.11.3 Tuning Tools 199
7.12 Summary 200

Chapter 8. Standardization and Connectivity 203

8.1 Standardization 203
8.1.1 Compatibility Standards 203
8.1.2 Portability Standards 204
8.1.3 Interoperability Standards 209

8.2 Connectivity 223
8.2.1 Connectivity with Peer UNIX Machines 223
8.2.2 Connectivity with Host Machines 226
8.2.3 Connectivity and Access to PC-DOS 228
8.3 Summary 229

Chapter 9. Design of AIX: A PowerOpen Implementation 231

9.1 Components of the Kernel 231
9.2 Functions of the Kernel 233
9.3 Kernel Services 233
9.3.1 System Calls 233
9.3.2 Kernel Facilities 235
9.4 Distinguishing Features of the AIX Kernel 241
9.5 Extending the Kernel 245
9.6 Programs, Processes, and Process Groups 246
9.7 AIX Notifiers 248
9.8 Internal Representation of Files 252
9.8.1 File Types 252
9.8.2 Inode and In-core Inodes 253
9.8.3 File Links 253
9.8.4 Files to File System Relationship 254
9.9 Buffer Cache 257
9.10 Summary 259

Chapter 10. AIX Process Subsystem Internals 261

10.1 The Difference Between a Program and a Process 261
10.2 Process Structure 262
10.3 Process-affiliated Kernel Structures 263
10.4 Process States 264
10.5 Priority Handling 266
10.6 Context Switching 268
10.7 Process Scheduling 269
10.8 The Thread Paradigm 271
10.8.1 Thread-affiliated Kernel Structures 272
10.8.2 Thread States 273
10.8.3 pids and tids 275
10.8.4 Context Switching 275
10.8.5 Scheduling 276
10.9 Process Monitoring 277
10.10 Interrupt and Exception Handling 280
10.11 Interprocess Communication 282
10.11.1 Pipes 282
10.11.2 Message Queues 283
10.11.3 Shared Memory 284
10.11.4 Semaphores 285
10.11.5 Sockets 286
10.11.6 Streams 287
10.12 Summary 290

Chapter 11. AIX File, Memory, and I/O Subsystem Internals 291

11.1 AIX File System 291
11.1.1 Physical File System 292
11.1.2 Memory Mapped Files 293
11.2 Journaled File System 295
11.2.1 Logical Volume Manager 295
11.2.2 Disk Mirroring 301
11.2.3 Bad Block Relocation 302
11.3 Memory Subsystem 303
11.3.1 Memory Addressability 303
11.3.2 Segmented Memory 304
11.3.3 Virtual Memory Management 307
11.3.4 Page Replacement 309
11.3.5 Memory Load Control 310
11.3.6 Code Pinning 311
11.4 I/O Subsystem 311
11.4.1 Asynchronous I/O 311
11.4.2 I/O Pacing 312
11.5 Device Subsystem 312
11.5.1 Device Drivers Overview 313
11.5.2 Major and Minor Numbers 313
11.5.3 Character and Block Device Drivers 313
11.5.4 Device Switch Table 313
11.5.5 Device Head and Device Handler 313
11.6 Object Data Manager 314
11.7 Summary 316

Chapter 12. What You Need to Build a PowerPC 317

12.1 Memory Subsystems 318
12.1.1 System Memory 319
12.1.2 System ROM 320
12.1.3 Nonvolatile Memory 320
12.1.4 I/O Memory 320
12.1.5 Memory Mapped System I/O 320
12.1.6 Secondary Cache 320
12.2 Storage Subsystems 320
12.2.1 Interface 321
12.2.2 Hardfile 321
12.2.3 Diskette 321
12.2.4 CD-ROM 321
12.3 Human Interface Subsystems 321
12.3.1 Alphanumeric Input Device 321
12.3.2 Pointer Device 321
12.3.3 Audio 322
12.3.4 Graphics 322
12.4 Real-Time Clock 322
12.5 Connectivity Subsystems 322
12.5.1 Serial 322
12.5.2 Parallel 323
12.5.3 Network 323
12.6 Expansion Bus Options 323
12.7 Interface Standards 323
12.7.1 SCSI 323
12.7.2 IDE 323